

Deutsch



FUJITSU Software BS2000

# UDS/SQL V2.8

Sichern, Informieren und Reorganisieren

Benutzerhandbuch

Ausgabe März 2016

## **Kritik... Anregungen... Korrekturen...**

Die Redaktion ist interessiert an Ihren Kommentaren zu diesem Handbuch. Ihre Rückmeldungen helfen uns, die Dokumentation zu optimieren und auf Ihre Wünsche und Bedürfnisse abzustimmen.

Sie können uns Ihre Kommentare per E-Mail an [manuals@ts.fujitsu.com](mailto:manuals@ts.fujitsu.com) senden.

## **Zertifizierte Dokumentation nach DIN EN ISO 9001:2008**

Um eine gleichbleibend hohe Qualität und Anwenderfreundlichkeit zu gewährleisten, wurde diese Dokumentation nach den Vorgaben eines Qualitätsmanagementsystems erstellt, welches die Forderungen der DIN EN ISO 9001:2008 erfüllt.

cognitas. Gesellschaft für Technik-Dokumentation mbH  
[www.cognitas.de](http://www.cognitas.de)

## **Copyright und Handelsmarken**

Copyright © 2016 Fujitsu Technology Solutions GmbH.

Alle Rechte vorbehalten.

Liefermöglichkeiten und technische Änderungen vorbehalten.

Alle verwendeten Hard- und Softwarenamen sind Handelsnamen und/oder Warenzeichen der jeweiligen Hersteller.

---

# Inhalt

<b>1</b>	<b>Einleitung</b>	<b>11</b>
<b>1.1</b>	<b>Konzept der UDS/SQL-Dokumentation</b>	<b>12</b>
<b>1.2</b>	<b>Zielsetzung und Zielgruppen des Handbuchs</b>	<b>16</b>
<b>1.3</b>	<b>Konzept des Handbuchs</b>	<b>17</b>
<b>1.4</b>	<b>Änderungen gegenüber den Vorgänger-Handbüchern</b>	<b>18</b>
<b>1.5</b>	<b>Darstellungsmittel</b>	<b>20</b>
1.5.1	Warnhinweise und Hinweise	20
1.5.2	Nicht-SDF-Darstellungsmittel	20
1.5.3	SDF-Syntaxdarstellung	22
<b>1.6</b>	<b>Beispieldatenbanken</b>	<b>28</b>
<b>2</b>	<b>Datenbank aktualisieren und rekonstruieren mit BMEND</b>	<b>31</b>
<b>2.1</b>	<b>Funktionen</b>	<b>32</b>
<b>2.2</b>	<b>Anweisungen</b>	<b>33</b>
2.2.1	Regeln	33
2.2.2	Erlaubte Funktionen	34
2.2.3	Anweisungen für BMEND	35
	Realms zu einer Datenbank zuschalten (ADD-REALM)	36
	Puffergröße festlegen (ALLOCATE-BUFFER-POOL)	37
	Online-Sicherungsfähigkeit der Datenbank zurücknehmen (DISABLE-ONLINE-COPY)	38
	Datenbank als online-sicherungsfähig markieren (ENABLE-ONLINE-COPY)	39
	Eingabe der Anweisungen beenden (END)	40
	Logging bei inkonsistenter Datenbank ausschalten (KILL-LOG)	41
	Datenbank eröffnen (OPEN-DATABASE)	42
	Realms abschalten (REMOVE-REALM)	43
	Logging-Informationen ausgeben (SHOW-LOG-INFORMATION)	44
	Logging für das Datenbank-Original einschalten (START-LOG)	50

	Logging für den Datenbankbetrieb ausschalten (STOP-LOG) . . . . .	58
	Anweisung rückgängig machen (UNDO) . . . . .	59
2.2.4	AFIMs in eine Datenbank einspielen (UPDATE-DATABASE) . . . . .	60
	Kommandofolge zum Starten von BMEND . . . . .	63
<b>2.3</b>	<b>Versorgung der BMEND-Jobvariablen . . . . .</b>	<b>64</b>
<b>3</b>	<b>Konsistenz einer Datenbank prüfen mit BCHECK . . . . .</b>	<b>69</b>
<hr/>		
<b>3.1</b>	<b>Beschreibung des Prüfverfahrens . . . . .</b>	<b>70</b>
3.1.1	Prüfmodus einstellen . . . . .	70
3.1.2	Umfang der Prüfung festlegen . . . . .	72
3.1.3	Prüfung auf Kohärenz . . . . .	73
<b>3.2</b>	<b>Systemumgebung . . . . .</b>	<b>74</b>
<b>3.3</b>	<b>Ergebnisse des Summierungslaufs für einen Sortierungslauf nutzen . . . . .</b>	<b>81</b>
<b>3.4</b>	<b>Anweisungen für BCHECK . . . . .</b>	<b>82</b>
	Größe des Sortierpuffers festlegen (SORTCORE) . . . . .	83
	Prüfmodus wählen und Umfang der Prüfung festlegen (CHECK) . . . . .	84
	Kriterien für die globale Konsistenzprüfung wählen (TYPE) . . . . .	85
	Schema bezeichnen (SCHEMA NAME) . . . . .	87
	Zu prüfende Realms spezifizieren (REALM NAME) . . . . .	88
	Zu prüfende Satzarten spezifizieren (RECORD NAME) . . . . .	89
	Zu prüfende Sets spezifizieren (SET NAME) . . . . .	92
	Zu prüfende SEARCH-Keys spezifizieren (KEY REF) . . . . .	94
<b>3.5</b>	<b>Kommandofolge zum Starten von BCHECK . . . . .</b>	<b>96</b>
<b>3.6</b>	<b>Beispiele zu BCHECK . . . . .</b>	<b>97</b>
<b>3.7</b>	<b>Meldungen . . . . .</b>	<b>103</b>
3.7.1	Warnungen . . . . .	104
3.7.2	Fehlermeldungen . . . . .	104
3.7.3	Ablaufmeldungen . . . . .	105
3.7.4	Konsistenzfehlermeldungen . . . . .	110
3.7.4.1	Globale Konsistenzfehler (keine Indexwertprüfung) . . . . .	110
3.7.4.2	Globale Konsistenzfehler (Indexwertprüfung) . . . . .	120
3.7.4.3	Lokale Konsistenzfehler . . . . .	123
3.7.5	Verwendung von Auftragsschaltern . . . . .	136

<b>4</b>	<b>Schema/Subschema Information Area ausdrucken mit BPSIA</b>	<b>137</b>
<b>4.1</b>	<b>Systemumgebung</b>	<b>138</b>
<b>4.2</b>	<b>Anweisungen für BPSIA</b>	<b>139</b>
	Schema ausdrucken (DISPLAY SCHEMA)	139
	Subschema ausdrucken (DISPLAY SUBSCHEMA)	140
	Eingabe der Anweisungen beenden (END)	140
<b>4.3</b>	<b>Kommandofolge zum Starten von BPSIA</b>	<b>141</b>
<b>4.4</b>	<b>Beschreibung des SIA-Protokolls</b>	<b>142</b>
	SIA PRINT REPORT (allgemeine Informationen)	142
	REFERENCE NUMBERS	145
	AREA INFORMATION	146
	RECORD WITHIN LIST	148
	RECORD INFORMATION	149
	DBTT INFORMATION	151
	CALC INFORMATION	153
	SET INFORMATION	155
	KEY INFORMATION (NO CALC SEARCH KEYS)	159
	CALC-SEARCH-KEY INFORMATION	162
<b>4.5</b>	<b>Beschreibung des SSIA-Protokolls</b>	<b>164</b>
	SSIA PRINT REPORT (allgemeine Informationen)	164
	REFERENCE NUMBERS	166
	AREA INFORMATION	167
	RECORD INFORMATION	168
	CALC KEY INFORMATION	170
	ITEM STRING LIST	172
	KEY ITEM LIST	174
	SET INFORMATION	176
	KEY INFORMATION	179
<b>5</b>	<b>Relationale Schemainformation ausgeben mit BPSQLSIA</b>	<b>181</b>
<b>5.1</b>	<b>Überblick</b>	<b>181</b>
<b>5.2</b>	<b>Systemumgebung</b>	<b>182</b>
<b>5.3</b>	<b>Voraussetzungen für SQL-Zugriffe in den CODASYL-Definitionen</b>	<b>183</b>
<b>5.4</b>	<b>SQL-Datentypen</b>	<b>184</b>
	Alphanumerischer Datentyp	184
	Nationaler Datentyp	184
	Numerische Datentypen	185

	Strukturierte Datentypen . . . . .	186
<b>5.5</b>	<b>Anweisungen für BPSQLSIA . . . . .</b>	<b>189</b>
	Eingabe beenden (END) . . . . .	189
	Datenbank eröffnen (OPEN-DATABASE) . . . . .	190
	Subschemata auswählen (PRINT-RELATIONAL-SCHEMAINFO) . . . . .	191
<b>5.6</b>	<b>Kommandofolge zum Starten von BPSQLSIA . . . . .</b>	<b>192</b>
<b>5.7</b>	<b>Beschreibung der Ausgabe von BPSQLSIA . . . . .</b>	<b>193</b>
<b>5.8</b>	<b>Umsetzregeln . . . . .</b>	<b>194</b>
<b>5.9</b>	<b>Übersicht über die zugelassenen SQL-Zugriffe pro Basistabelle . . . . .</b>	<b>196</b>
<b>5.10</b>	<b>Beispiel . . . . .</b>	<b>197</b>
<b>6</b>	<b>Statistik der Speicherplatzbelegung ausdrucken mit BSTATUS . . . . .</b>	<b>203</b>
<b>6.1</b>	<b>Funktionen . . . . .</b>	<b>204</b>
<b>6.2</b>	<b>Systemumgebung . . . . .</b>	<b>207</b>
	Arbeitsdateien . . . . .	208
<b>6.3</b>	<b>Anweisungen für BSTATUS . . . . .</b>	<b>210</b>
	Subschema bezeichnen (SUBSCHEMA) . . . . .	211
	Realm-Statistik ausdrucken (DISPLAY REALM) . . . . .	212
	Set-Statistik ausdrucken (DISPLAY TABLE FOR SET) . . . . .	215
	Owner-Statistik ausdrucken (DISPLAY TABLE FOR OWNER) . . . . .	218
	Satzart-Statistik ausdrucken (DISPLAY RECORD) . . . . .	221
	CALC-Key-Statistik ausdrucken (DISPLAY CALC) . . . . .	224
	Satzanzahl-Statistik ausdrucken (DISPLAY RECORDNUMBER) . . . . .	228
	Eingabe der Anweisungen beenden (END) . . . . .	230
<b>6.4</b>	<b>Kommandofolge zum Starten von BSTATUS . . . . .</b>	<b>231</b>
<b>7</b>	<b>Inhalt der Realms ausdrucken mit BPRECORD . . . . .</b>	<b>233</b>
<b>7.1</b>	<b>Systemumgebung . . . . .</b>	<b>235</b>
<b>7.2</b>	<b>Allgemeine Beschreibung der Ausgabe von BPRECORD . . . . .</b>	<b>236</b>
<b>7.3</b>	<b>Anweisungen für BPRECORD . . . . .</b>	<b>239</b>
	SCHEMA bezeichnen (SCHEMA NAME) . . . . .	241
	Auszugebenden Realm angeben (REALM NAME) . . . . .	242
	Umfang der Ausgaben festlegen (PRINT) . . . . .	243
	Kennseite ausdrucken (DISPLAY PAGE) . . . . .	246

	FPA-Einträge auflisten (DISPLAY FPA) . . . . .	250
	DBTT-Einträge auflisten (DISPLAY DBTT) . . . . .	251
	CALC-Seiten ausdrucken (DISPLAY CALC) . . . . .	253
	Datenseiten ausdrucken (DISPLAY DATA) . . . . .	258
	BPRECORD-Lauf beenden (END) . . . . .	261
<b>7.4</b>	<b>Kommandofolge zum Starten von BPRECORD . . . . .</b>	<b>262</b>
<b>8</b>	<b>Online-Dienste durchführen mit der UDS-Online-Utility . . . . .</b>	<b>263</b>
<b>8.1</b>	<b>Funktionen der UDS-Online-Utility . . . . .</b>	<b>264</b>
<b>8.2</b>	<b>DML RELOCATE - Datenseiten verlagern . . . . .</b>	<b>266</b>
<b>8.3</b>	<b>DML FPASCAN - Such-Modus für Freiplatzsuche festlegen . . . . .</b>	<b>269</b>
<b>8.4</b>	<b>DML PREFRLM - Preferred-Realm für verteilbare Listen festlegen . . . . .</b>	<b>269</b>
<b>8.5</b>	<b>DML REORGPPP – Probable Position Pointers (PPPs) reorganisieren . . . . .</b>	<b>270</b>
<b>8.6</b>	<b>Kommandofolge zum Starten der UDS-Online-Utility . . . . .</b>	<b>273</b>
<b>8.7</b>	<b>Anweisungen der UDS-Online-Utility . . . . .</b>	<b>273</b>
<b>8.8</b>	<b>SDF-Anweisungen der UDS-Online-Utility . . . . .</b>	<b>274</b>
	Prozedur definieren (DECLARE-PROCEDURE) . . . . .	275
	Variable definieren (DECLARE-VARIABLE) . . . . .	276
	Prozedur löschen (DELETE-PROCEDURE) . . . . .	277
	Variable löschen (DELETE-VARIABLE) . . . . .	277
	UDS-Online-Utility beenden (END) . . . . .	277
	Prozedur ausführen (REPEAT-PROCEDURE) . . . . .	278
	Such-Modus für Freiplatzsuche festlegen (SET-FPA-SCAN-PARAMETERS) . . . . .	280
	Online-Utility-Parameter festlegen (SET-ONLINE-UTILITY-PARAMETERS ) . . . . .	281
	Preferred-Realm festlegen (SET-PREF-REALM-PARAMETERS ) . . . . .	282
	Eigenschaften einer RELOCATE-DML festlegen (SET-RELOCATE-PARAMETERS) . . . . .	283
	Eigenschaften einer REORGPPP-DML definieren (SET-REORGANIZE-PPP-PARAMETERS . . . . .	288
	Aktuell gültige Parameter für DML FPASCAN ausgeben (SHOW-FPA-SCAN-PARAMETERS) . . . . .	291
	Prozedur ausgeben (SHOW-PROCEDURE) . . . . .	291
	Aktuell gültige Parameter für DML PREFRLM ausgeben (SHOW-PREF-REALM-PARAMETERS ) . . . . .	292
	Aktuell gültige Parameter für DML RELOCATE ausgeben (SHOW-RELOCATE-PARAMETERS) . . . . .	292
	Aktuell gültige Parameter für DML REORGPPP ausgeben (SHOW-REORGANIZE-PPP-PARAMETERS) . . . . .	292

	Aktuellen Wert einer Variablen ausgeben (SHOW-VARIABLE) . . . . .	293
<b>8.9</b>	<b>Prozedur-Anweisungen der UDS-Online-Utility . . . . .</b>	<b>294</b>
	Wert zu einer Variablen addieren (ADD) . . . . .	295
	Prozedurfolge sofort beenden (BREAK) . . . . .	296
	Eingabe von Prozedur-Anweisungen beenden (END) . . . . .	296
	Prozedurfolge nach aktuellem Durchlauf beenden (EXIT) . . . . .	296
	Aktuelle Transaktion beenden (FINISH) . . . . .	297
	Startseite für Freiplatzsuche festlegen (FPASCAN) . . . . .	297
	Wert einer Variablen festlegen (MOVE) . . . . .	298
	Preferred-Realm für eine verteilbare Liste neu setzen bzw. verändern (PREFRLM) . . . . .	298
	Aktuelle Transaktion der UDS-Online-Utility starten (READY UPDATE) . . . . .	299
	Verlagerung durchführen (RELOCATE) . . . . .	299
	Kommentar einfügen (REMARK) . . . . .	300
	PPP-Reorganisation durchführen (REORGPPP) . . . . .	300
	Wert einer Variablen ausgeben (SHOW) . . . . .	301
	Wartezeit festlegen (WAIT) . . . . .	302
<b>8.10</b>	<b>Fehlerbehandlung der UDS-Online-Utility . . . . .</b>	<b>303</b>
<b>8.11</b>	<b>Vordefinierte Variablen . . . . .</b>	<b>304</b>
<b>8.12</b>	<b>Vordefinierte Standardprozeduren . . . . .</b>	<b>305</b>
<b>8.13</b>	<b>Status-Codes . . . . .</b>	<b>305</b>
<b>8.14</b>	<b>Syntax von Bedingungen . . . . .</b>	<b>306</b>
<b>8.15</b>	<b>Beispiele . . . . .</b>	<b>307</b>
	Beispiel 1: Verlagerung einer festen Anzahl von Seiten mit der Standardprozedur . . . . .	307
	Beispiel 2: Verlagerung aller möglicher Seiten mit der Standardprozedur . . . . .	307
	Beispiel 3: Vermeidung von Kollisionen . . . . .	308
	Beispiel 4: Vermeidung von Kollisionen orientiert an den gemeldeten Sperrern . . . . .	309
	Beispiel 5: Vermeidung von Kollisionen mit Warten . . . . .	310
	Beispiel 6: Information über Statuscode . . . . .	311
	Beispiel 7: Verkleinerung von Realms . . . . .	312
	Beispiel 8: Reorganisieren von PPPs . . . . .	312
<b>9</b>	<b>Datenbank reorganisieren mit BREORG . . . . .</b>	<b>313</b>
<b>9.1</b>	<b>Funktionen . . . . .</b>	<b>314</b>
<b>9.2</b>	<b>Systemumgebung . . . . .</b>	<b>315</b>
	Arbeitsdateien . . . . .	317
<b>9.3</b>	<b>Datenbanksicherung . . . . .</b>	<b>321</b>



<b>9.4</b>	<b>Anweisungen für BREORG</b> . . . . .	<b>322</b>
	Puffergröße festlegen (ALLOCATE-BUFFERPOOL) . . . . .	323
	Eingabe der Anweisungen beenden (END) . . . . .	324
	Realm-Größe ändern (MODIFY-REALM-SIZE) . . . . .	325
	Satzmengengerüst ändern (MODIFY-RECORD-POPULATION) . . . . .	328
	Datenbank eröffnen (OPEN-DATABASE) . . . . .	331
	CALC-Bereiche reorganisieren (REORGANIZE-CALC) . . . . .	332
	Alle Probable Position Pointer (PPP) in einem Realm reorganisieren (REORGANIZE-POINTERS) . . . . .	339
	Tabellen und Set-Konstrukte reorganisieren (REORGANIZE-SET) . . . . .	340
	Schema angeben (SPECIFY-SCHEMA) . . . . .	346
	Subschema angeben (SPECIFY-SUBSCHEMA) . . . . .	347
	Anweisung rückgängig machen (UNDO) . . . . .	348
<b>9.5</b>	<b>Kommandofolge zum Starten von BREORG</b> . . . . .	<b>349</b>
<b>9.6</b>	<b>Beispiele</b> . . . . .	<b>351</b>
<b>10</b>	<b>Wiederverwendung von Database Keys und Freiplatzsuche steuern mit BMODTT</b> . . . . .	<b>355</b>
<b>10.1</b>	<b>Systemumgebung</b> . . . . .	<b>356</b>
<b>10.2</b>	<b>Anweisungen für BMODTT</b> . . . . .	<b>357</b>
<b>10.3</b>	<b>Kommandofolge zum Starten von BMODTT</b> . . . . .	<b>359</b>
	<b>Fachwörter</b> . . . . .	<b>361</b>
	<b>Abkürzungen</b> . . . . .	<b>413</b>
	<b>Literatur</b> . . . . .	<b>417</b>
	<b>Stichwörter</b> . . . . .	<b>423</b>



---

# 1 Einleitung

Das **Universelle Datenbank-System** UDS/SQL ist ein Datenbanksystem für hohe Durchsatzanforderungen. Es basiert auf dem Strukturkonzept von CODASYL, geht aber in seinen Möglichkeiten weit darüber hinaus und bietet koexistent auf dem gleichen Datenbestand das Relationenmodell an.

Zur Auswertung und Änderung der Daten stehen COBOL-DML, CALL-DML und SQL (ISO-konform) zur Verfügung. COBOL-DML-Anweisungen sind in die COBOL-Sprache integriert, die CALL-DML kann aus jeder Programmiersprache aufgerufen werden, SQL-Anweisungen können innerhalb von DRIVE-Programmen angewendet oder über eine ODBC-Schnittstelle genutzt werden.

UDS/SQL verhindert durch wirksame, flexibel einsetzbare Schutzmechanismen unberechtigte Zugriffe auf die Datenbank und garantiert Vertraulichkeit, Integrität und Verfügbarkeit. Diese Mechanismen sind mit dem Transaktionsmonitor openUTM abgestimmt.

Das Datensicherungskonzept von UDS/SQL schützt die Datenbestände wirkungsvoll vor Zerstörung und Verlust. Dabei werden UDS/SQL- eigene Mechanismen wie Logging veränderter Information mit BS2000-Funktionen wie DRV (Dual Recording by Volume) kombiniert.

Unter Einsatz des Zusatzproduktes UDS-D können Datenbestände in BS2000-Rechnernetzen verarbeitet werden. UDS/SQL garantiert dabei die netzweite Konsistenz der Daten. In Verbindung mit openUTM-D bzw. openUTM (Unix/Linux/Windows) lässt sich verteilte Transaktionsverarbeitung sowohl in BS2000-Rechnernetzen als auch im Verbund von BS2000 und anderen Betriebssystemen realisieren. UDS/SQL kann als Datenbank in Client-Server-Lösungen über SQL-Gateway bzw. über ODBC-Server eingesetzt werden.

UDS/SQL bietet durch seine Architekturmerkmale (z. B. Multitasking, Multithreading, DB-Cache) und durch seine vielseitigen Strukturierungsmöglichkeiten einen sehr hohen Durchsatz.

## 1.1 Konzept der UDS/SQL-Dokumentation

Dem Abschnitt „Wegweiser durch die Handbuchreihe“ entnehmen Sie, welche Handbücher und welche Teile daraus Ihrem Informationsbedürfnis entsprechen. Ein Fachwortverzeichnis liefert Kurzdefinitionen der im Text benutzten Fachwörter.

Außer über das Inhaltsverzeichnis können Sie die Antworten auf Ihre Fragen gezielt über das Stichwortverzeichnis und über Kolumnentitel nachschlagen.

### Wegweiser durch die Handbuchreihe

Das Datenbanksystem UDS/SQL ist im Wesentlichen in fünf Handbüchern dokumentiert:

- UDS/SQL Entwerfen und Definieren
- UDS/SQL Anwendungen programmieren
- UDS/SQL Aufbauen und Umstrukturieren
- UDS/SQL Datenbankbetrieb
- UDS/SQL Sichern, Informieren und Reorganisieren

**Weitere Handbücher** zu UDS/SQL und Zusatzprodukten finden Sie auf [Seite 15](#).

Als Einstieg dient Ihnen das Handbuch „[Entwerfen und Definieren](#)“, Kapitel 2 und 3; hier werden erläutert:

- die Gründe für den Einsatz von Datenbanken
- das Datenbankmodell der CODASYL
- das Relationenmodell unter Berücksichtigung von SQL
- eine Abgrenzung der Modelle
- die Koexistenz der verschiedenen Datenbankmodelle bei einer UDS/SQL-Datenbank
- die charakteristischen Eigenschaften von UDS/SQL

Der weitere Umgang mit den Handbüchern richtet sich nach Ihren Vorkenntnissen und Aufgaben. Die [Tabelle 1](#) hilft Ihnen dabei, den richtigen Weg durch die Handbücher zu finden.

### Beispiele

Angenommen, Ihre Aufgabe ist es, in COBOL-DML zu programmieren, so finden Sie in der zweiten Zeile der [Tabelle 1](#) unter „Aufgaben des Anwenders“ die Spalte „COBOL/CALL-DML Programm“. Im Handbuch „[Entwerfen und Definieren](#)“ brauchen Sie dann für Ihre Arbeit folgende Kapitel:

Allgemeines	E = zum Einstieg
Schema-DDL	D = zur Detailinformation

SSL D = zur Detailinformation

Subschema-DDL L = zum Lernen der Funktionen

Welche Kapitel Sie aus den weiteren Handbüchern brauchen, erfahren Sie in der gleichen Spalte.

Wenn Sie dagegen als Datenbankadministrator für den Datenbankbetrieb zuständig sind, orientieren Sie sich bitte in der Spalte „Verwalten und Bedienen“.

Inhalt der fünf Haupthandbücher	Aufgaben des Anwenders							
	Entwerfen und Definieren	COBOL/ CALL-DML Programm.	SQL- Programmieren	Aufbauen und Umstrukt.	Verwalten und Bedienen	Arbeiten mit openUTM	Arbeiten mit IQS	Arbeiten mit UDS-D

**Handbuch UDS/SQL Entwerfen und Definieren**

Einleitung	E	-	-	-	-	E	E	-
Allgemeines	E	E	E	E	E	E	-	-
Entwurf der Datenbank	E	-	-	-	-	-	-	-
Schema-DDL	L	D	-	L	L	-	-	-
SSL	L	D	-	L	L	-	-	-
Subschema DDL	L	L	-	L	L	-	-	-
relationales Schema	L	-	D	-	-	-	-	-
Aufbau der Seiten	D	-	-	D	D	-	-	-
Aufbau der Sätze und Tabellen	D	-	-	D	D	-	-	-
Nachschlageteil	S	-	-	S	-	-	-	-

**Handbuch UDS/SQL Anwendungen programmieren**

Einleitung	-	E	-	-	-	E	E	-
Einführung	-	E	-	-	-	-	-	-
Transaktionskonzept	-	L	-	L	L	D	D	-
Currency-Tabelle	-	L	-	L	L	-	-	-
Funktionen der DML	D	L	-	L	-	-	-	-
Anwenden der DML	-	L	-	D	-	-	-	-
Nachschlageteil COBOL-DML	-	L	-	-	-	-	-	-
Nachschlageteil CALL-DML	-	L	-	-	-	-	-	-
Testen von DML-Funktionen mit DMLTEST	-	L	-	-	-	-	-	-

Tabelle 1: Wegweiser durch die Handbücher

Inhalt der fünf Haupthandbücher	Aufgaben des Anwenders							
	Entwerfen und Definieren	COBOL/ CALL-DML Programm.	SQL- Program- mieren	Aufbauen und Umstrukt.	Verwalten und Bedienen	Arbeiten mit openUTM	Arbeiten mit IQS	Arbeiten mit UDS-D

**Handbuch UDS/SQL Aufbauen und Umstrukturieren**

Einleitung	-	-	-	E	-	E	E	-
Überblick	-	-	-	E	E	-	-	-
Datenbank aufbauen	-	-	-	L	-	-	-	-
Zugriffsberechtigungen festlegen	-	-	-	L	-	-	-	-
Daten speichern und entladen	D	-	-	L	-	D	-	-
Datenbank umstrukturieren	D	-	-	L	-	-	-	-
Datenbankobjekte umbenennen	D	-	-	L	-	-	-	-
Datenbank umstellen	D	-	-	L	-	-	-	-

**Handbuch UDS/SQL Datenbankbetrieb**

Einleitung	-	-	-	-	E	E	E	-
Der Database Handler	-	-	-	-	L	-	-	D
Ladeparameter des DBH	-	-	-	-	L	-	-	D
Administration	-	-	-	-	L	-	-	D
Hochverfügbarkeit	-	-	-	-	E	-	-	-
Ressourcen-Erweiterung und Umorganisation im laufenden Betrieb	D	-	-	-	E	-	-	-
Datenbank sichern und wiederherstellen im Fehlerfall	D	-	-	D	L	D	-	D
Leistungsoptimierung	-	-	-	-	D	-	-	D
Nutzung der BS2000-Funktionalität	-	-	-	-	D	-	-	-
Der SQL-Vorgang	-	-	-	-	L	-	-	-
UDSMON	-	-	-	-	D	-	-	-
Einsatz von IQS	-	-	-	L	D	-	D	-
Einsatz von UDS-D	D	D	-	D	D	D	-	D
Funktionscodes der DML-Anweisungen	-	D	-	-	D	-	-	-

Tabelle 1: Wegweiser durch die Handbücher

(Teil 2 von 3)

Inhalt der fünf Haupthandbücher	Aufgaben des Anwenders							
	Entwerfen und Definieren	COBOL/ CALL-DML Programm.	SQL- Programmieren	Aufbauen und Umstrukt.	Verwalten und Bedienen	Arbeiten mit openUTM	Arbeiten mit IQS	Arbeiten mit UDS-D

**Handbuch UDS/SQL Sichern, Informieren und Reorganisieren**

Einleitung	-	-	-	-	E	E	E	-
Datenbank aktualisieren und rekonstruieren	D	-	-	D	L	D	-	-
Konsistenz einer Datenbank prüfen	-	-	-	-	L	-	-	-
Datenbankinformationen ausgeben	D	-	-	D	L	-	-	-
Online-Dienste durchführen	D	-	-	D	L	-	-	-
Datenbank reorganisieren	D	-	-	D	L	-	-	-
Wiederverwendung von freigewordenen Database Keys steuern	D	-	-	D	L	-	-	-

**Weitere Handbücher**

UDS/SQL Meldungen	D	D	D	D	D	D	D	D
UDS/SQL Taschenbuch	S	S	-	S	S	S	S	S
IQS	-	-	-	D	D	-	L	-
ADILOS	-	-	-	-	D	-	L	-
KDBS	-	L	-	D	-	-	-	-
SQL für UDS/SQL Sprachbeschreibung	-	-	D	-	D	-	-	-

Tabelle 1: Wegweiser durch die Handbücher

(Teil 3 von 3)

- E dient als Einstieg, wenn Sie bisher noch nichts mit UDS/SQL zu tun hatten
- L in diesen Teilen der Handbücher steht das Lernen der Funktionen im Vordergrund
- D hier können Sie hineinschauen, wenn Sie Detailinformationen suchen
- S dient zum Nachschlagen von Syntaxregeln bei der praktischen Arbeit

**Was Sie noch über die Handbücher wissen sollten**

Literaturverweise finden Sie in Kurzform im Text. Finden Sie im Text z.B. (siehe Handbuch „Anwendungen programmieren“, CONNECT), so müssen Sie unter dem Stichwort CONNECT im Handbuch „Anwendungen programmieren“ nachschauen. Der vollständige Handbuchtitel steht im Literaturverzeichnis.

### **UDS/SQL Meldungen**

Das Handbuch enthält alle Meldungen, die UDS/SQL ausgibt. Die Meldungen sind aufsteigend nach Nummern oder bei einigen Dienstprogrammen alphabetisch sortiert.

### **UDS/SQL Taschenbuch**

Das UDS/SQL-Taschenbuch enthält alle Übersichten zu den UDS/SQL-Funktionen und Formaten.

### **SQL für UDS/SQL Sprachbeschreibung**

Das Handbuch beschreibt den SQL-DML-Sprachumfang von UDS/SQL. Neben UDS/SQL-spezifischen Erweiterungen umfasst der beschriebene Sprachumfang die dynamische SQL als wesentliche Erweiterung der SQL-Norm.

## **1.2 Zielsetzung und Zielgruppen des Handbuchs**

Das Handbuch ist für den Datenbankadministrator bestimmt, der die Aufgabe hat, Datenbanken zu aktualisieren und zu rekonstruieren, Daten zu reorganisieren und Datenbanken auf Konsistenz zu prüfen.

Der Datenbankadministrator sollte alle Schritte des Datenbankentwurfs (Datenbankdesign, Schema-Erstellen, Subschema-Erstellen und SSL-Erstellen) kennen und wissen, wie er DB-Anwenderprogramme erstellt.

Außerdem muss er über gute BS2000-Kenntnisse verfügen, das Transaktionskonzept von UDS/SQL und das allgemeine Sicherungskonzept von UDS/SQL kennen (siehe Handbuch „[Datenbankbetrieb](#)“) sowie grundlegende Kenntnisse über die Dateien einer UDS/SQL-Datenbank und die UDS/SQL-Dienstprogramme haben (siehe Handbuch „[Aufbauen und Umstrukturieren](#)“, Überblick über die Dateien und Realms einer UDS/SQL-Datenbank).



## 1.3 Konzept des Handbuchs

### Was enthält dieses Handbuch?

Dieses Handbuch beschreibt Verwaltungs- und Bedienungsarbeiten, die für einen reibungslosen Betrieb der Datenbank notwendig sind:

- Aktualisieren und Rekonstruieren der Datenbank
- Prüfen der Konsistenz der Datenbank
- Ausgeben von Datenbankinformationen
- Reorganisieren der Datenbank
- Steuern der Wiederverwendung von frei gewordenen Database Keys

Beispiele helfen Ihnen, dies nachzuvollziehen.

### Wie finden Sie sich im Handbuch zurecht?

Dem [Abschnitt „Konzept der UDS/SQL-Dokumentation“](#) entnehmen Sie, welche Handbücher und welche Teile daraus Ihrem Informationsbedürfnis entsprechen. Ein Fachwortverzeichnis liefert Kurzdefinitionen der im Text benutzten Fachwörter.

Außer über das Inhaltsverzeichnis können Sie die Antworten auf Ihre Fragen gezielt über das Stichwortverzeichnis und über Kolumnentitel nachschlagen.

### Readme-Datei

Funktionelle Änderungen der aktuellen Produktversion und Nachträge zu diesem Handbuch entnehmen Sie bitte ggf. der produktspezifischen Readme-Datei.

Readme-Dateien stehen Ihnen online bei dem jeweiligen Produkt zusätzlich zu den Produkthandbüchern unter <http://manuals.ts.fujitsu.com> zur Verfügung. Alternativ finden Sie Readme-Dateien auch auf der Softbook-DVD.

#### *Informationen unter BS2000*

Wenn für eine Produktversion eine Readme-Datei existiert, finden Sie im BS2000-System die folgende Datei:

```
SYSRME.<product>.<version>.<lang>
```

Diese Datei enthält eine kurze Information zur Readme-Datei in deutscher oder englischer Sprache (<lang>=D/E). Die Information können Sie am Bildschirm mit dem Kommando `/SHOW-FILE` oder mit einem Editor ansehen.

Das Kommando `/SHOW-INSTALLATION-PATH INSTALLATION-UNIT=<product>` zeigt, unter welcher Benutzerkennung die Dateien des Produkts abgelegt sind.

*Ergänzende Produkt-Informationen*

Aktuelle Informationen, Versions-, Hardware-Abhängigkeiten und Hinweise für Installation und Einsatz einer Produktversion enthält die zugehörige Freigabemitteilung. Solche Freigabemitteilungen finden Sie online unter <http://manuals.ts.fujitsu.com>.

## 1.4 Änderungen gegenüber den Vorgänger-Handbüchern

In der folgenden **Tabelle 2** sind die wichtigsten Änderungen der Version UDS/SQL V2.8 gegenüber der Version V2.7 aufgeführt. Außerdem wird jeweils das Handbuch und das Kapitel genannt, in dem die Änderung beschrieben wird. Wird ein Thema in mehr als einem Handbuch beschrieben, dann wird zuerst das Handbuch aufgeführt, in dem das Thema vollständig beschrieben wird. In der Spalte „Handbuch“ bedeuten die Einträge:

- |     |                              |     |   |
|-----|------------------------------|-----|---|
| ENT | Entwerfen und Definieren     | DBB | Datenbankbetrieb                        |
| ANW | Anwendungen programmieren    | SIR | Sichern, Informieren und Reorganisieren |
| AUF | Aufbauen und Umstrukturieren | MEL | Meldungen                               |

Thema	Handbuch	Kapitel
<b>UDSMON-Utility: Verbesserungen bei Transaktionszeit und DB COUNTERS</b>		
Für die Ausgabe auf Terminal und Drucker: In der UDS/SQL-Monitor-Maske wird COUNTER, die Einheit für die Ausgabe der AVG TRANSACTION TIME, präzisiert auf Sekunden und Millisekunden für das Monitoring von kurzen Transaktionen.	DBB	11
Neues DISPLAY DBCOUNTERS-Kommando in UDSMON für die Ausgabe von Datenbankzählern.	DBB	11
<b>BSTATUS-Utility: Begrenzung des TABLE STATISTICS FOR OWNER IN SET</b>		
Verbesserte DISPLAY TABLE FOR OWNER-Anweisung, um eine Begrenzung von TABLE STATISTICS FOR OWNER IN SET auf Sätze von bestimmten Ownern oder Satzbereiche zu ermöglichen.	SIR	6
Neue Meldungen des Dienstprogramms BSTATUS	MEL	3
<b>Neue Meldung des Dienstprogramms BPRECORD 2553</b> falls der Wert 0 als Startwert in RSQ range definiert wird.	MEL	3

Table 2: Änderungen in V2.8 gegenüber V2.7

Thema	Handbuch	Kapitel
<b>Datenbankbetrieb: Die Anzahl der DML-Anweisungen und die Anzahl der Ein- und Ausgaben werden pro Datenbank gezählt.</b>	DBB	4
	MEL	2
<b>BOUTLOAD-Utility: Ausgabe im CSV-Format</b>	AUF	5
	MEL	3
COPY-RECORD-Anweisung: Neuer Operand CSV-OUTPUT	AUF	5
Neues Ausgabeformat CSV	AUF	5
<b>Online-Utility – Probable Position Pointers (PPP) reorganisieren</b>		
Neue DML REORGPPP - PPPs reorganisieren	SIR	8
Neue SDF-Anweisungen: SET-REORGANIZE-PPP-PARAMETERS, SHOW-REORGANIZE-PPP-PARAMETERS	SIR	8
Neue Prozedur-Anweisung REORGPPP	SIR	8
Neue vordefinierte Variablen: REORG-PPP-CURRENT, REORG-PPP-LOCKED, REORG-PPP-PAGES	SIR	8
Neue vordefinierte Standardprozedur *STDREPPP	SIR	8
Neues Beispiel „Reorganisieren von PPPs“	SIR	8
Neue Statuscodes mit Fortschrittshinweisen der Online-Utility REORGPPP und neue Fehlercodes	ANW	10

Table 2: Änderungen in V2.8 gegenüber V2.7



### Allgemeine Änderung

Die bisherige Bezeichnung BS2000/OSD-BC des BS2000-Grundausbaus ändert sich und lautet ab Version V10.0: BS2000 OSD/BC.

## 1.5 Darstellungsmittel

In diesem Abschnitt finden Sie die Erläuterung der Piktogramme für Warnhinweise und Hinweise sowie die Zeichenerklärung der Metasprache, wie sie zur Beschreibung von Syntaxregeln benutzt wird.

### 1.5.1 Warnhinweise und Hinweise

	Hinweis auf besonders wichtige Informationen
 <b>VORSICHT!</b>	Warnhinweis

### 1.5.2 Nicht-SDF-Darstellungsmittel

Sprachelement	Erklärung	Beispiel
<u>SCHLÜSSELWORT</u>	Schlüsselwörter sind durch Großbuchstaben mit Unterstreichung dargestellt. Sie müssen mindestens die unterstrichenen Teile des Schlüsselwortes angeben.	DATABASE- <u>KEY</u> <u>MANUAL</u>
WAHLWORT	Wahlwörter sind durch Großbuchstaben ohne Unterstreichung dargestellt. Wenn Sie Wahlwörter weglassen, hat das keinen Einfluss auf die Bedeutung einer Klausel.	NAME IS ALLOWED PAGES
<i>variable</i>	Variable sind mit kursiven Kleinbuchstaben dargestellt. Bei der Benutzung eines Formats, in dem eine Variable erscheint, müssen Sie einen aktuellen Wert an ihre Stelle setzen.	<i>fieldname</i> <i>literal-3</i> <i>ganzzahl</i>
{ Entweder } { oder }	Genau einen der eingeklammerten Ausdrücke müssen Sie angeben. Eingerückte Zeilen gehören zum vorhergehenden Ausdruck. Die Klammer geben Sie nicht an.	{ <u>CALC</u> } { <u>INDEX</u> }  { <u>VALUE IS</u> } { <u>VALUES ARE</u> }
[wahlweise]	Den eingeklammerten Ausdruck dürfen Sie weglassen. UDS/SQL benutzt dann Standardwerte. Die Klammern selbst geben Sie nicht an.	[IS <i>ganzzahl</i> ]  [ <u>WITHIN</u> <i>realmname</i> ]

Tabelle 3: Zeichen der Metasprache

(Teil 1 von 2)

Sprachelement	Erklärung	Beispiel
... oder ,...	Den unmittelbar vorstehenden Ausdruck können Sie wahlweise mehrmals wiederholen. Die beiden Sprachelemente unterscheiden Wiederholungen mit Leerzeichen oder mit Komma als Trennzeichen.	<i>fieldname,...</i>  { <u>SEARCH</u> KEY.....}...
..... oder . .	Kennzeichnet Auslassungen aus Gründen der Übersichtlichkeit. Bei der Benutzung der Formate sind diese Auslassungen nicht erlaubt.	<u>SEARCH</u> KEY IS ..... <u>RECORD</u> NAME . .
⋮	Den Punkt müssen Sie angeben, gefolgt von mindestens einem Leerzeichen. Die Unterstreichung geben Sie nicht an.	<u>SET SECTION</u> .  03 <i>fieldname</i> ..... ⋮
Zwischenraum	Bedeutet, dass Sie mindestens ein Leerzeichen angeben müssen.	<u>USING</u> <u>CALC</u>

Tabelle 3: Zeichen der Metasprache

(Teil 2 von 2)

Alle übrigen Zeichen wie ( ) , . ; „ “ = sind keine Metazeichen:  
Sie müssen sie so angeben, wie sie im Format dargestellt sind.

### 1.5.3 SDF-Syntaxdarstellung

Diese Syntaxbeschreibung basiert auf der SDF-Version 4.7. Die Syntax der SDF-Kommando-/Anweisungssprache wird im Folgenden in 3 Tabellen erklärt.

#### Tabelle 4: Metasyntax

In den Kommando-/Anweisungsformaten werden bestimmte Zeichen und Darstellungsformen verwendet, deren Bedeutung in [Tabelle 4](#) erläutert wird.

#### Tabelle 5: Datentypen

Variable Operandenwerte werden in SDF durch Datentypen dargestellt. Jeder Datentyp repräsentiert einen bestimmten Wertevorrat. Die Anzahl der Datentypen ist beschränkt auf die in [Tabelle 5](#) beschriebenen Datentypen.

Die Beschreibung der Datentypen gilt für alle Kommandos und Anweisungen. Deshalb werden bei den entsprechenden Operandenbeschreibungen nur noch Abweichungen von [Tabelle 5](#) erläutert.

#### Tabelle 6: Zusätze zu Datentypen

Für den Datentyp integer enthält [Tabelle 6](#) außerdem kursiv gesetzte Einheiten, die nicht Bestandteil der Syntax sind. Sie dienen lediglich als Lesehilfe.

Die Beschreibung der Zusätze zu den Datentypen gilt für alle Kommandos und Anweisungen. Deshalb werden bei den entsprechenden Operandenbeschreibungen nur noch Abweichungen von [Tabelle 6](#) erläutert.

Kennzeichnung	Bedeutung	Beispiele
GROSSBUCHSTABEN	Großbuchstaben bezeichnen Schlüsselwörter. Einige Schlüsselwörter beginnen mit *	OPEN DATABASE  COPY-NAME = <u>*NONE</u>
=	Das Gleichheitszeichen verbindet einen Operandennamen mit den dazugehörigen Operandenwerten.	CONFIGURATION-NAME = <name 1..8>
< >	Spitze Klammern kennzeichnen Variablen, deren Wertevorrat durch Datentypen und ihre Zusätze beschrieben wird ( <a href="#">Tabellen 24</a> und <a href="#">27</a> ).	DATABASE = <dbname>

Tabelle 4: Metasyntax

(Teil 1 von 2)

Kennzeichnung	Bedeutung	Beispiele
<u>Unterstreich</u>	Der Unterstrich kennzeichnet den Standardwert eines Operanden.	SCHEMA-NAME = <u>*STD</u>
/	Der Schrägstrich trennt alternative Operandenwerte.	CMD = <u>*ALL</u> / <dal-cmd>
(...)	Runde Klammern kennzeichnen Operandenwerte, die eine Struktur einleiten.	*KSET-FORMAT(...)
Einrückung	Die Einrückung kennzeichnet die Abhängigkeit zu dem jeweils übergeordneten Operanden.	USER-GROUP-NAME = *KSET-FORMAT(...) *KSET-FORMAT(...)   HOST = <host>
	Der Strich kennzeichnet zusammengehörende Operanden einer Struktur. Sein Verlauf zeigt Anfang und Ende einer Struktur an. Innerhalb einer Struktur können weitere Strukturen auftreten. Die Anzahl senkrechter Striche vor einem Operanden entspricht der Strukturtiefe.	USER-GROUP-NAME = *ALL-EXCEPT(...) *ALL-EXCEPT(...)   NAME = *KSET-FORMAT(...)   *KSET-FORMAT(...)   HOST = <host>   ...
,	Das Komma steht vor weiteren Operanden der gleichen Strukturstufe.	,SPACE = <u>STD</u>
list-poss(n):	Aus den list-poss folgenden Operandenwerten kann eine Liste gebildet werden. Ist (n) angegeben, können maximal n Elemente in der Liste vorkommen. Enthält die Liste mehr als ein Element, muss sie in runde Klammern eingeschlossen werden.	NAME = list-poss(30): <subschemaname>

Tabelle 4: Metasyntax

(Teil 2 von 2)

Datentyp	Zeichenvorrat	Besonderheiten
alog-seq-nr	0..9	1..9 Zeichen
appl	A..Z 0..9 \$,#,@  Strukturkennzeichen: Bindestrich	1..8 Zeichen Zeichenfolge, die in mehrere durch Bindestrich getrennte Teilzeichenfolgen gegliedert sein kann; erstes Zeichen A..Z oder \$, #, @ Wenn weniger als 8 Zeichen angegeben werden, wird intern mit Unterstrich auf 8 Zeichen aufgefüllt.
catid	A..Z 0..9	1..4 Zeichen; darf nicht mit der Zeichenfolge PUB beginnen
copyname	A..Z 0..9	1..7 Zeichen beginnend mit A..Z
c-string	EBCDIC-Zeichen	1..4 Zeichen ist in Hochkommata einzuschließen; der Buchstabe C kann vorangestellt werden; Hochkommata innerhalb des c-string müssen verdoppelt werden
csv-dateiname	A..Z 0..9 Strukturkennzeichen: Bindestrich	1..30 Zeichen ist in Hochkomma einzuschließen
dal-cmd	A..Z 0..9 Bindestrich	1..64 Zeichen
date	0..9 Strukturkennzeichen: Bindestrich	Angabe eines Datums Eingabeformat: jjjj-mm-tt jjjj : Jahr; wahlweise 2- oder 4-stellig mm : Monat tt : Tag
dbname	A..Z 0..9	1..17 Zeichen beginnend mit A..Z
device	A..Z 0..9 \$,#,@  Strukturkennzeichen: Bindestrich	5..8 Zeichen beginnend mit A..Z oder 0..9 Zeichenfolge, die in mehrere durch Bindestrich getrennte Teilzeichenfolgen gegliedert sein kann und die einem im System verfügbaren Gerät entspricht. In der Dialogführung zeigt SDF die zulässigen Operandenwerte an. Hinweise zu möglichen Geräten sind der jeweiligen Operandenbeschreibung zu entnehmen.

Tabelle 5: Datentypen

(Teil 1 von 3)



Datentyp	Zeichenvorrat	Besonderheiten
host	A..Z 0..9 \$,#,@  Strukturkennzeichen: Bindestrich	1..8 Zeichen Zeichenfolge, die in mehrere durch Bindestrich getrennte Teilzeichenfolgen gegliedert sein kann; erstes Zeichen A..Z oder \$, #, @; wenn weniger als 8 Zeichen angegeben werden, wird intern mit Unterstrich auf 8 Zeichen aufgefüllt
integer	0..9,+,-	+ bzw. - kann nur erstes Zeichen sein.
kset	A..Z 0..9 \$,#,@  Strukturkennzeichen: Bindestrich	1..8 Zeichen Zeichenfolge, die in mehrere durch Bindestrich getrennte Teilzeichenfolgen gegliedert sein kann; erstes Zeichen A..Z oder \$, #, @; wenn weniger als 8 Zeichen angegeben werden, wird intern mit Unterstrich auf 8 Zeichen aufgefüllt.
name	A..Z 0..9 \$,#,@	1..8 Zeichen darf nicht nur aus 0..9 bestehen oder darf nicht mit einer Ziffer beginnen.
realmname	A..Z 0..9  Strukturkennzeichen: Bindestrich	1..30 Zeichen Zeichenfolge, die in mehrere durch Bindestrich getrennte Teilzeichenfolgen gegliedert sein kann; erstes Zeichen: A..Z
realmref	0..9	1..3 Zeichen
recordname	A..Z 0..9  Strukturkennzeichen: Bindestrich	1..30 Zeichen Zeichenfolge, die in mehrere durch Bindestrich getrennte Teilzeichenfolgen gegliedert sein kann; erstes Zeichen: A..Z Bei Satzarten mit Searchkey wird empfohlen, max. 26 Zeichen lange Namen zu verwenden, da ansonsten der implizit gebildete Setname (SYS_...) entsprechend der Begrenzung der Namenslänge von Sets gekürzt wird.
recordref	0..9	1..3 Zeichen
schemaname	A..Z 0..9  Strukturkennzeichen: Bindestrich	1..30 Zeichen Zeichenfolge, die in mehrere durch Bindestrich getrennte Teilzeichenfolgen gegliedert sein kann; erstes Zeichen: A..Z

Tabelle 5: Datentypen

(Teil 2 von 3)

Datentyp	Zeichenvorrat	Besonderheiten
setname	A..Z 0..9  Strukturkennzeichen: Bindestrich	1..30 Zeichen Zeichenfolge, die in mehrere durch Bindestrich getrennte Teilzeichenfolgen gegliedert sein kann; erstes Zeichen A..Z
structured-name	A...Z 0...9 \$, #, @ Bindestrich	alphanumerische Zeichenfolge, die in mehrere durch Bindestrich getrennte Teilzeichenfolgen gegliedert sein kann; erstes Zeichen: A...Z oder \$, #, @
subschemaname	A..Z 0..9  Strukturkennzeichen: Bindestrich	1..30 Zeichen Zeichenfolge, die in mehrere durch Bindestrich getrennte Teilzeichenfolgen gegliedert sein kann; erstes Zeichen: A..Z
time	0..9 Strukturkennzeichen: Doppelpunkt	Angabe einer Tageszeit  Eingabeformat: $\left. \begin{array}{l} hh:mm:ss \\ hh:mm \\ hh \end{array} \right\}$  hh, mm, ss: bei Stunden, Minuten und Sekunden können führende Nullen weggelassen werden
userid	A..Z  0..9 \$, #, @	1..8 Zeichen beginnend mit A..Z oder \$, #, @ BPRIVACY: Wenn weniger als 8 Zeichen angegeben werden, wird intern mit Unterstrich auf 8 Zeichen aufgefüllt.
volume	A..Z 0..9 \$, #, @	1..6 Zeichen beginnend mit A..Z oder 0..9
x-string	Sedezimal: 00..FF	1..8 Zeichen ist in Hochkommata einzuschließen; der Buchstabe X muss vorangestellt werden; die Anzahl der Zeichen darf ungerade sein.

Tabelle 5: Datentypen

(Teil 3 von 3)

Zusatz	Bedeutung
<i>x..y unit</i>	beim Datentyp integer: Intervallangabe x      Mindestwert, der für integer erlaubt ist. x ist eine ganze Zahl, die mit einem Vorzeichen versehen werden darf. y      Maximalwert, der für integer erlaubt ist. y ist eine ganze Zahl, die mit einem Vorzeichen versehen werden darf. <i>unit</i> nur bei integer: zusätzliche Einheiten. Folgende Angaben werden verwendet: <i>Mbyte</i> <i>Kbyte</i> <i>seconds</i>

Tabelle 6: Zusätze zu Datentypen

## 1.6 Beispieldatenbanken

Die Datenbanken VERSAND und KUNDEN sind Grundlage für die meisten Beispiele und Abläufe der Dienstprogramme in diesem Handbuch.

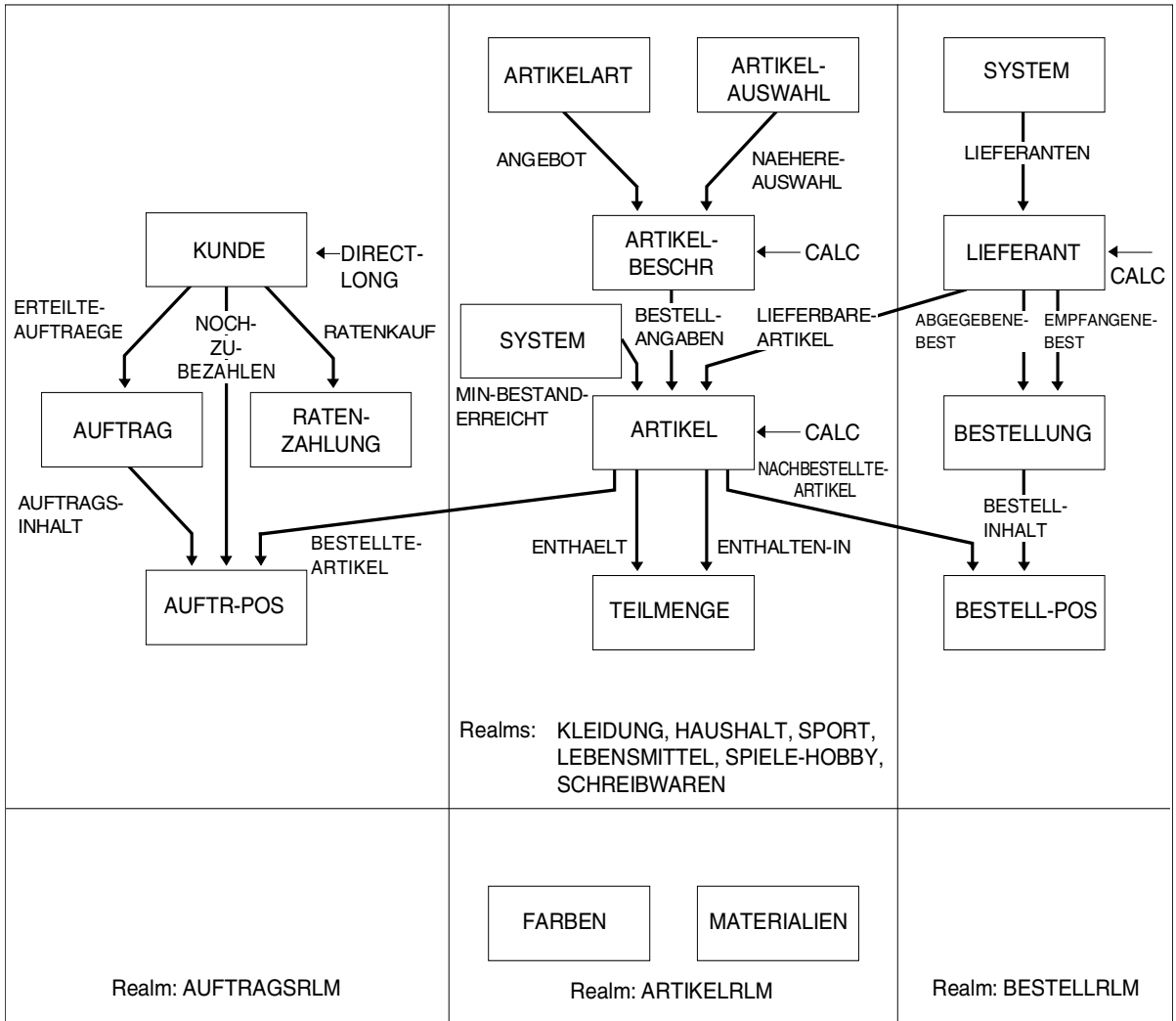


Bild 1: Datenbank VERSAND mit dem Schemanamen ARTIKELVERSAND

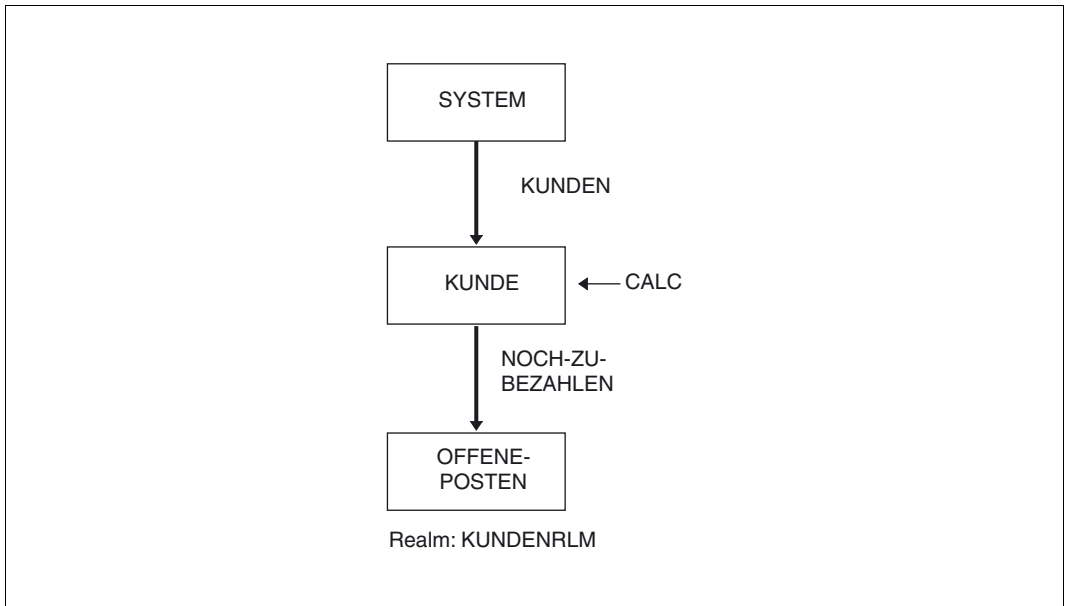


Bild 2: Datenbank KUNDEN mit dem Schemanamen KUNDENDATEI

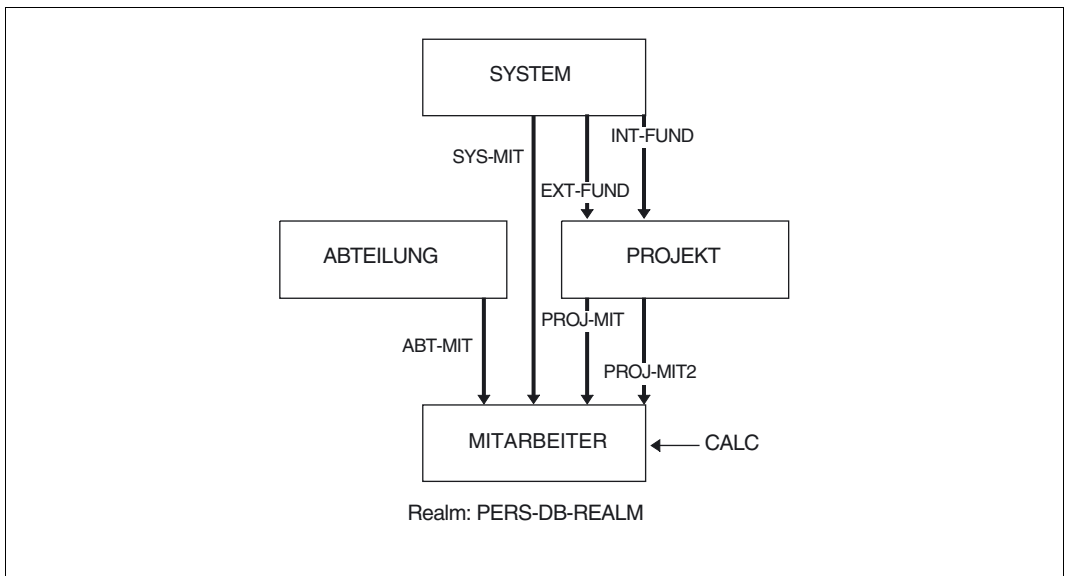


Bild 3: Datenbank PERSONAL mit dem Schemanamen PERS-DB



---

## 2 Datenbank aktualisieren und rekonstruieren mit BMEND

Im Rahmen des UDS/SQL-Sicherungskonzeptes (siehe Handbuch „[Datenbankbetrieb](#)“) erfüllt das Dienstprogramm BMEND die Funktion Datenbank aktualisieren und rekonstruieren.

Mit BMEND ist es möglich, die gesamte Datenbank, einzelne oder ausgewählte Realms mit einer ALOG-Datei bzw. mit mehreren ALOG-Dateien zu aktualisieren. BMEND können Sie sowohl für die Originaldatenbank, für die Schattendatenbank als auch für abgeschaltete Realms verwenden.

Die ALOG-Dateien fahren Sie mit der Anweisung UPDATE-DATABASE nach.

Die BMEND-Anweisung SHOW-LOG-INFORMATION liefert Ihnen die Angaben, mit welchen ALOG-Dateien die einzelnen Realms zu aktualisieren sind, bis sie einen konsistenten Stand der gesamten Datenbank erreicht haben, d.h. diese ALOG-Dateien müssen Sie für das Nachfahren zur Verfügung stellen.

Sie erhalten auch Informationen darüber, ob in der Sequenz der ALOG-Dateien eine Logging-Lücke existiert oder eine ALOG-Datei inkonsistent beendet wurde. In solchen Fällen ist eine Aktualisierung nur möglich bis zu diesem Zeitpunkt oder erst mit einer Sicherung nach diesem Zeitpunkt.

## 2.1 Funktionen

Das Dienstprogramm BMEND bietet folgende Funktionen:

- Nachfahren von Datenbanken und Schattendatenbanken (bzw. einzelner Realms davon)
- Zuschalten und Abschalten von Realms
- Einschalten und Ausschalten der Online-Sicherungsfähigkeit einer Datenbank
- Einschalten und Ausschalten AFIM-Logging
- Auskunftsfunktion über den Zustand von Datenbanken, Schattendatenbanken und ALOG-Dateien
- Nachfahren von abgeschalteten Realms parallel zum DBH-Betrieb wie auch Aufrufen der Auskunftsfunktion für Realms oder ALOG-Dateien

BMEND berücksichtigt beim Start ggf. eine zugewiesene UDS/SQL-Pubset-Deklaration (siehe Handbuch „[Datenbankbetrieb](#)“, Pubset-Deklarations-Jobvariable). Eine fehlerhafte Zuweisung führt zum Programmabbruch.



## 2.2 Anweisungen

### 2.2.1 Regeln

Sie können falsch eingegebene Anweisungen korrigieren.

Anweisungen des gleichen Typs (außer SHOW-LOG-INFORMATION) werden zusammengefasst und nur einmal ausgeführt.

Widersprechen sich die Angaben bzgl. der Funktion bzw. des Objektes (z.B. START-LOG/STOP-LOG oder ADD-REALM/REMOVE-REALM), so gilt immer die letzte Angabe.

Alle gültigen Anweisungen werden nach der Anweisung END ausgeführt. Ausgenommen sind die Anweisungen ALLOCATE-BUFFER-POOL, OPEN-DATABASE und UNDO:

- Die Anweisung ALLOCATE-BUFFER-POOL muss als erste Anweisung gegeben werden.
- Die Anweisung OPEN-DATABASE ist nur erlaubt, wenn vorher kein ADD-FILE-LINK LINK-NAME=DATABASE angegeben wurde.
- Mit der Anweisung UNDO - oder mit der inversen Funktion, wenn vorhanden - können Sie jede richtig eingegebene Anweisung zurücknehmen.

Alle anderen Anweisungen können Sie in beliebiger Reihenfolge eingeben. Die Ausführung der Anweisungen wird in dieser Reihenfolge durchgeführt:

- ALLOCATE-BUFFER-POOL
- OPEN-DATABASE
- UPDATE-DATABASE
- ADD-REALM
- REMOVE-REALM
- START-LOG
- STOP-LOG
- KILL-LOG
- ENABLE-ONLINE-COPY
- DISABLE-ONLINE-COPY
- SHOW-LOG-INFORMATION

## 2.2.2 Erlaubte Funktionen

Der Umfang der erlaubten Funktionen hängt ab von folgenden Fragen:

- Originaldatenbank oder Schattendatenbank?
- Wird die Datenbank oder die Schattendatenbank vom DBH prozessiert?
- Ist die Datenbank konsistent?
- Ist das AFIM-Logging eingeschaltet?

Nicht erlaubte Funktionen sind aus der SDF-Maske ausgeblendet.

Die Menge der erlaubten Funktionen kann sich während eines BMEND-Laufs ändern, weil BMEND-Anweisungen Konsistenz und AFIM-Logging beeinflussen können.

Die Ausgabe von Konsistenzinformationen ist immer möglich.

Die Anweisung KILL-LOG ist nur auf inkonsistenten Datenbanken zulässig. Die Anweisung ist erforderlich, um einen Warmstart ohne Logging durchzuführen.

### Liste der Sonderfälle

- Originaldatenbank mit aktivem DBH:  
UPDATE-DATABASE  
Nachfahren parallel zu einer Session auf Realms erlaubt, die gemäß DBDIR-AK0 abgeschaltet sind
- inkonsistente Originaldatenbank mit AFIM-Logging ohne aktiven DBH:  
ADD-REALM  
REMOVE-REALM  
START-LOG  
STOP-LOG  
werden abgewiesen, wenn vorher keine Anweisung UPDATE-DATABASE angegeben wurde, um eine konsistente Datenbank zu erzeugen
- konsistente Originaldatenbank ohne AFIM-Logging ohne aktiven DBH:  
ENABLE-ONLINE-COPY  
erst dann erlaubt, wenn vorher die Anweisung START-LOG angegeben wurde
- Schattendatenbank mit aktivem DBH:  
UPDATE-DATABASE  
Nachfahren parallel zu einer Session auf Realms erlaubt, die gemäß DBDIR-AK0 abgeschaltet sind

- inkonsistente Schattendatenbank ohne aktiven DBH:

ADD-REALM

REMOVE-REALM

wird abgewiesen, wenn vorher keine Anweisung UPDATE-DATABASE angegeben wurde, um eine konsistente Datenbank zu erzeugen

### 2.2.3 Anweisungen für BMEND

Anweisung	Bedeutung
ADD-REALM	Realms zu einer Datenbank zuschalten
ALLOCATE-BUFFER-POOL	Puffergröße festlegen
DISABLE-ONLINE-COPY	Online-Sicherungsfähigkeit der Datenbank zurücknehmen
ENABLE-ONLINE-COPY	Datenbank als online-sicherungsfähig markieren
END	Eingabe der Anweisungen beenden
KILL-LOG	Logging bei inkonsistenter Datenbank ausschalten
OPEN-DATABASE	Datenbank eröffnen
REMOVE-REALM	Realms abschalten
SHOW-LOG-INFORMATION	Logging-Informationen ausgeben
START-LOG	Logging für das Datenbank-Original einschalten
STOP-LOG	Logging für den Datenbankbetrieb ausschalten
UNDO	Anweisung rückgängig machen
UPDATE-DATABASE	AFIMs in eine Datenbank einspielen

Tabelle 7: Anweisungen für BMEND

Die Anweisungen von BMEND sind im Folgenden in alphabetischer Reihenfolge beschrieben.

## Realms zu einer Datenbank zuschalten (ADD-REALM)

Die Anweisung ADD-REALM ermöglicht es, einen Realm oder mehrere Realms zuzuschalten. Die Realms müssen konsistent sein und zum aktuellen DBDIR passen.

```
ADD-REALM
```

```
REALM-NAME = *ALL / *ALL-EXCEPT(...) / list-poss(30): <realmname>
```

```
  *ALL-EXCEPT(...)
```

```
    | NAME = list-poss(30): <realmname>
```

### **REALM-NAME = \*ALL**

Alle abgeschalteten Realms werden zugeschaltet.

### **REALM-NAME = \*ALL-EXCEPT(...)**

Mit Ausnahme der angegebenen Realms werden alle abgeschalteten Realms zugeschaltet.

### **NAME = list-poss(30): <realmname>**

Name derjenigen Realms, die nicht zugeschaltet werden sollen.

### **REALM-NAME = list-poss(30): <realmname>**

Alle angegebenen Realms werden zugeschaltet.



Die Realms DBDIR und DBCOM können nicht zugeschaltet werden. Sie werden abgewiesen.

## Puffergröße festlegen (ALLOCATE-BUFFER-POOL)

Die Anweisung ALLOCATE-BUFFER-POOL legt die Größe des verwendeten Buffer Pools in Mbyte fest.

Diese Anweisung kann entfallen (Nutzung des Standardwertes) oder muss als erste Anweisung gegeben werden.

ALLOCATE-BUFFER-POOL
BUFFER-SIZE = *STD / <integer 1..2000>

### **BUFFER-SIZE = \*STD**

Die Standard-Größe des Buffer Pools wird mit 1 Mbyte festgelegt.

### **BUFFER-SIZE = <integer 1..2000>**

Die Größe des Buffer Pools muss in den angegebenen Grenzen liegen. Der Maximalwert ist abhängig von der Version des Betriebssystems und dem Arbeitsspeicherausbau der Anlage.



Der Wert von ADDRESS-SPACE-LIMIT muss größer als der hier angegebene Wert sein. Er kann vom Systembetreuer mit dem Kommando MODIFY-USER-ATTRIBUTES eingestellt werden.

## Online-Sicherungsfähigkeit der Datenbank zurücknehmen (DISABLE-ONLINE-COPY)

Für alle Realms der Datenbank wird das Merkmal der Online-Sicherungsfähigkeit zurückgenommen.

Die Anweisung ist auch für eine Schattendatenbank erlaubt.

DISABLE-ONLINE-COPY

Diese Anweisung hat keine Operanden.

## Datenbank als online-sicherungsfähig markieren (ENABLE-ONLINE-COPY)

Alle Realms der Datenbank werden als online-sicherungsfähig gekennzeichnet.

Die Anweisung ist auch für eine Schattendatenbank erlaubt.

Das Markieren der Datenbank als online-sicherungsfähig ist bei einem Original nur möglich, wenn das AFIM-Logging eingeschaltet ist oder vorher die Anweisung START-LOG gegeben wurde. Die Anweisung START-LOG wiederum braucht eine konsistente Datenbank, sodass möglicherweise auch eine Anweisung UPDATE-DATABASE notwendig ist.

Zum Erstellen von Online-Sicherungen von Datenbanken bzw. einzelnen Realms sollten Sie grundsätzlich die entsprechenden Anweisungen von HSMS bzw. ARCHIVE verwenden.

Bei Verwendung von COPY-FILE zur Erzeugung einer Online-Kopie wird vom System nicht geprüft, ob für die betreffende Datenbank alle Voraussetzungen für die "Online Sicherungsfähigkeit" erfüllt sind. Sie müssen in diesem Fall dafür sorgen, dass das AFIM-Logging eingeschaltet ist, damit später die Online-Kopie durch Nachfahren konsistent gemacht werden kann.

Die Erzeugung von Online-Kopien während des Ablaufs ändernder Dienstprogramme ist generell nicht zulässig.

Die Online-Sicherungsfähigkeit einer Datenbank ist sowohl in den UDS/SQL-Verwaltungsdaten als auch in den DVS-Katalogeinträgen der Datenbank-Dateien vermerkt. Beim Kopieren einer Datenbank, sei es mit COPY-FILE oder mit HSMS/ARCHIVE, gehen die Angaben zur Online-Sicherungsfähigkeit im DVS-Katalog abhängig von den verwendeten Parametern evtl. verloren. Daher müssen Sie vor dem Einsatz einer Datenbankkopie, bei deren Original die Online-Sicherungsfähigkeit eingeschaltet war, für die Konsistenz dieser Eigenschaft sorgen, z.B. durch einen erneuten Lauf des Dienstprogramms BMEND mit Anweisung ENABLE-ONLINE-COPY.

ENABLE-ONLINE-COPY

Diese Anweisung hat keine Operanden.

## Eingabe der Anweisungen beenden (END)

Die Eingabe der Anweisungen wird beendet. Die Ausführung wird gestartet.

Die Anweisung END kann durch die Anweisung UNDO nicht zurückgenommen werden.

END

Diese Anweisung hat keine Operanden.



## Logging bei inkonsistenter Datenbank ausschalten (KILL-LOG)

Mit der Anweisung KILL-LOG wird das AFIM-Logging abgeschaltet, um einen Warmstart ohne ALOG-Datei durchzuführen (z.B. wegen Hardware-Fehler auf der ALOG-Datei). Die ALOG-Datei wird beim Warmstart dazu verwendet, die AFIMs aus der RLOG-Datei aufzunehmen, die noch nicht in der ALOG-Datei sind.

Für einen Datenbank-Warmstart reicht die RLOG-Datei aus.

Die Kennzeichnung der Online-Sicherungsfähigkeit wird ebenfalls durch die Anweisung KILL-LOG zurückgenommen.

Die Anweisung KILL-LOG ist nur auf einer inkonsistenten Datenbank zulässig.

KILL-LOG

Diese Anweisung hat keine Operanden.



Wird der Warmstart ohne ALOG-Datei durchgeführt, entsteht eine Logging-Lücke. Es ist daher sinnvoll, nach dem Warmstart die Datenbank zu sichern und vor erneuten Änderungen das Logging wieder einzuschalten.

## Datenbank eröffnen (OPEN-DATABASE)

Mit der Anweisung OPEN-DATABASE legen Sie die Datenbank fest, die mit den nachfolgenden Anweisungen bearbeitet werden soll.

```
OPEN-DATABASE
```

```
DATABASE-NAME = <dbname>
```

```
,COPY-NAME = *NONE / <copyname>
```

```
,USER-IDENTIFICATION = *OWN / <userid>
```

### **DATABASE-NAME = <dbname>**

Name der Datenbank. Sie können nur eine Datenbank bearbeiten, die in Ihrer eigenen Kennung liegt. Eine Datenbank aus einer fremden Kennung kann nur von der TSOS-Kennung des Systemverwalters bearbeitet werden.

### **COPY-NAME = \*NONE**

Es wird das Datenbank-Original bearbeitet.

### **COPY-NAME = <copyname>**

Es wird die Schattendatenbank mit dem angegebenen Kopienamen bearbeitet.

### **USER-IDENTIFICATION = \*OWN**

Die Datenbank liegt in der eigenen Kennung.

### **USER-IDENTIFICATION = <userid>**

Die Angabe einer fremden Datenbank-Kennung ist nur unter der TSOS-Kennung erlaubt.



Ist die Datenbank über ADD-FILE-LINK LINK-NAME=DATABASE zugewiesen, so ist die Anweisung OPEN-DATABASE unzulässig.

## Realms abschalten (REMOVE-REALM)

Mit der Anweisung REMOVE-REALM können konsistente Realms abgeschaltet werden.

```
REMOVE-REALM
```

```
REALM-NAME = *ALL-EXCEPT(...) / list-poss(30): <realmname>
```

```
  *ALL-EXCEPT(...)
```

```
    | NAME = list-poss(30): <realmname>
```

### **REALM-NAME = \*ALL-EXCEPT(...)**

Mit Ausnahme der angegebenen Realms werden alle zugeschalteten Realms abgeschaltet.

**Name = list-poss(30): <realmname>**

Name derjenigen Realms, die nicht abgeschaltet werden sollen.

**REALM-NAME = list-poss(30): <realmname>**

Alle angegebenen Realms werden abgeschaltet.



Die Realms DBDIR und DBCOM können nicht abgeschaltet werden. Sie werden abgewiesen.

## Logging-Informationen ausgeben (SHOW-LOG-INFORMATION)

Mit der Anweisung SHOW-LOG-INFORMATION können folgende Informationen ausgegeben werden:

- Zustand von Realms der zugewiesenen Datenbank im Hinblick auf das notwendige Logintervall zum Erreichen eines gemeinsamen Konsistenzpunktes
- Eigenschaften der Datenbank „mit Logging bzw. ohne Logging“ und „online-sicherungsfähig“
- Informationen über bis zu 63 ALOG-Dateien (Historie, Folge Nummern absteigend)

Bei den ausgegebenen Zeiten handelt es sich um die lokale Uhrzeit.

Darüberhinaus werden Informationen für Nachfahrenvorgänge in einer Jobvariablen hinterlegt, wenn diese mit LINK-NAME=JVBMEND eingerichtet ist (siehe [Abschnitt „Versorgung der BMEND-Jobvariablen“ auf Seite 64](#)).

Die Anweisung SHOW-LOG-INFORMATION kann allein auch bei laufendem DBH-Betrieb angegeben werden.

```
SHOW-LOG-INFORMATION
```

```
REALM-NAME = *ALL / *ALL-EXCEPT(...) / list-poss(30): <realmname>
```

```
  *ALL-EXCEPT(...)
```

```
    | NAME = list-poss(30): <realmname>
```

```
,LOG-FILE = *STD / *NONE / <alog-seq-nr>
```

```
,OUTPUT = list-poss: *SYSLST / *SYSOUT
```

### **REALM-NAME = \*ALL**

Zu jedem Realm der zugewiesenen Datenbank werden Informationen ausgegeben.

### **REALM-NAME = \*ALL-EXCEPT(...)**

Mit Ausnahme der angegebenen Realms werden zu jedem Realm Informationen ausgegeben.

**NAME = list-poss(30): <realmname>**

Namen der Realms, für die keine Informationen ausgegeben werden sollen.

### **REALM-NAME = list-poss(30): <realmname>**

Zu jedem angegebenen Realm werden Informationen ausgegeben.

### **LOG-FILE = \*STD**

Informationen ab der aktuellen ALOG-Datei, d.h. der im DBDIR angegebenen, werden ausgegeben.

**LOG-FILE = \*NONE**

Es werden keine Informationen zu ALOG-Dateien ausgegeben.

**LOG-FILE = <alog-seq-nr>**

Informationen ab der angegebenen ALOG-Datei werden ausgegeben.

**OUTPUT = list-poss: \*SYSLST / \*SYSOUT**

Die Informationen werden ausgegeben.

**\*SYSLST**

Log-Information wird auf SYSLST ausgegeben.

**\*SYSOUT**

Log-Information wird auf SYSOUT ausgegeben.

**Meldungstexte**

Es werden folgende Informationen ausgegeben:

## 1. Angaben zur bearbeiteten Datenbank

```
***** LOG INFORMATION FOR DATABASE $userid.dbname[.copyname]
```

*dbname*

Name der zugewiesenen Datenbank

*copyname*

wird ausgegeben, falls eine Schattendatenbank zugewiesen wurde

## 2. Auflistung der spezifizierten Realms mit Angabe der ALOG SEQ NR (z.B. unterschiedlich bei Sicherungsbeginn oder bei Sicherungsende einer Online-Sicherung)

```
***** LOG INTERVAL OF SPECIFIED REALMS
```

REALM NAME	ALOG SEQ NR		CONSISTENT
	BEGIN	END	
<i>realmname-1</i>	<i>alog-seq-nr</i>	<i>alog-seq-nr</i>	YES/NO
...			
<i>realmname-n</i>	<i>alog-seq-nr</i>	<i>alog-seq-nr</i>	YES/NO

*realmname*

Namen der angegebenen Realms

## ALOG SEQ NR

BEGIN: *alog-seq-nr* des Realms zu Beginn des Sicherungsvorganges (der Online-Sicherung)  
 END: *alog-seq-nr* des Realms am Ende des Sicherungsvorganges (der Online-Sicherung)

## CONSISTENT

YES: Das Konsistenzdatum am Beginn und Ende des Sicherungsvorganges stimmt überein und es ist kein SYSTEM BREAK gesetzt  
 NO: Der Realm stellt einen von einem Konsistenzpunkt verschiedenen Stand dar (das Konsistenzdatum zu Beginn und am Ende des Sicherungsvorganges differieren oder SYSTEM BREAK ist gesetzt)

## 3. Angabe des notwendigen Logintervalls für das Nachfahren

```
***** TO MAKE THE SPECIFIED REALMS CONSISTENT, THE FOLLOWING
LOG FILES ARE NECESSARY:
FROM ALOG SEQ NR alog-seq-nr1 TO ALOG SEQ NR alog-seq-nr2
OR FROM LOG INTERVAL BEGIN datetime-1 TO LOG INTERVAL END datetime-2
```

*alog-seq-nr1*

Ab dieser ALOG SEQ NR muss die Nachfahrfunktion starten. Dieser Wert wird in einer Jobvariablen gesichert.

*alog-seq-nr2*

Dieser Wert stellt die DEADLINE dar, die angegeben werden muss, um alle spezifizierten Realms auf einen gemeinsamen Konsistenzstand zu bringen.

*datetime-1*

Zeitpunkt der ersten Protokollierung

*datetime-2*

Zeitpunkt der letzten Protokollierung

Wenn nichts nachzufahren ist, erscheint diese Ausgabe nicht.

## 4. Angabe des Logging-Zustandes mit Dateiträgerinformationen; diese Meldung wird zu Schattendatenbanken nicht ausgegeben.

```
***** SUPPORTS OF ACTUAL LOG FILE:
  DEFAULT SUPPORT:{VOLUME = vol-1 DEVICE = dev-1
                  [...] /
                  PVS ID = {DEFAULT PVS / catid}}
  RESERVE SUPPORT:{VOLUME = vol-4 DEVICE = dev-4
                  [...] /
                  PVS ID = {DEFAULT PVS / catid}}
```

### 5. Ausgabe der Logging-Geschichte

\*\*\*\*\* INFORMATION ABOUT LOG HISTORY:

ALOG SEQ NR	LOG INTERVAL		AFIM	BACKOUT	LOGGING
	BEGIN	END			
<i>alog-seq-nr</i>	<i>time-1</i>	<i>time-2</i>	[*]	[*]	[{*/?}]

#### ALOG SEQ NR

Bei der über LOG-FILE-Operand spezifizierten Folgennummer beginnend werden Informationen zu maximal 63 ALOG-Dateien ausgegeben (Folgennummern absteigend)

#### LOG INTERVAL

BEGIN: *time-1* Zeitpunkt der ersten Protokollierung  
 END: *time-2* Zeitpunkt der letzten Protokollierung

Ist dabei der Zeitpunkt nicht eindeutig, wird Sommerzeit angenommen und eine Warnung ausgegeben.

AFIM \*: In der ALOG-Datei sind AFIMs protokolliert  
 ̀: Lücke innerhalb der AFIM-Protokollierung

#### BACKOUT BFIM

\*: In der ALOG-Datei sind BFIMs protokolliert  
 ̀: In der ALOG-Datei sind keine BFIMs protokolliert

#### LOGGING GAP

\*: LOG INTERVAL END ist kleiner als LOG INTERVAL BEGIN der nächst höheren ALOG-Datei bzw. kleiner als BACK UP DATA des DBDIR (LOG-FILE=\*STD)  
 ? : Eine Fortführung der Protokollierung in der ALOG-Datei mit der nächst höheren Folgennummer kann nicht untersucht werden (LOG-FILE=*alog-seq-nr*)  
 ̀: Protokollierung zur nächsthöheren ALOG-Datei wurde ohne Unterbrechung fortgesetzt

Kann die im LOG-FILE-Operanden spezifizizierte ALOG-Datei nicht gelesen werden, so wird an Stelle der Tabelle eine Fehlermeldung ausgegeben.

### 6. Angaben zur Online-Sicherungsfähigkeit

\*\*\*\*\* ONLINE COPIES BY ARCHIVE ARE [NOT] ALLOWED

*Beispiel*

```

/ASSIGN-SYSDTA TO-FILE=*SYSCMD
/ADD-FILE-LINK LINK-NAME=DATABASE,FILE-NAME=VERSAND.DBDIR
/SELECT-PRODUCT-VERSION PRODUCT-NAME=UDS-SQL,VERSION=02.8A00
/START-UDS-BMEND
***** START          BMEND          (UDS/SQL V2.8 0000 )    2015-06-28    09:27:53
//SHOW-LOG-INFORMATION REALM-NAME=*ALL,LOG-FILE=4,OUTPUT=*SYSOUT
SYSTEM_BREAK OCCURRED IN REALM DATABASE-DIRECTORY
***** INCONSISTENT DATABASE DIRECTORY
SYSTEM_BREAK OCCURRED IN REALM BESTELLRLM
SYSTEM_BREAK OCCURRED IN REALM LEBENSMITTEL
SYSTEM_BREAK OCCURRED IN REALM ARTIKELRLM
***** INCONSISTENT DATABASE DIRECTORY
        FUNCTION ADD NOT AVAILABLE
        FUNCTION REMOVE NOT AVAILABLE
        FUNCTION START NOT AVAILABLE
        FUNCTION STOP NOT AVAILABLE

//END
***** BEGIN          FUNCTION SHOW LOG INFORMATION AT 09:27:54
***** LOG INFORMATION FOR DATABASE $XXXXXXXX.VERSAND _____ 1.
***** LOG INTERVAL OF SPECIFIED REALMS _____ 2.

                !      ALOG SEQ NR      !      !
REALM-NAME      ! BEGIN      ! END      ! CONSISTENT !
-----+-----+-----+-----+
DATABASE-DIRECTORY      !      2 !      2 !      NO      !
DATABASE-COMPILER-REALM !      1 !      1 !      YES     !
AUFTRAGSRLM            !      1 !      1 !      YES     !
BESTELLRLM             !      2 !      2 !      NO      !
KLEIDUNG               !      1 !      1 !      YES     !
HAUSHALT               !      1 !      1 !      YES     !
SPORT                 !      1 !      1 !      YES     !
LEBENSMITTEL           !      2 !      2 !      NO      !
SPIELE-HOBBY           !      1 !      1 !      YES     !
SCHREIBWAREN           !      1 !      1 !      YES     !
ARTIKELRLM             !      2 !      2 !      NO      !
***** TO MAKE THE SPECIFIED REALMS CONSISTENT, THE FOLLOWING LOG FILES ARE _____ 3.
        NECESSARY :
        FROM ALOG SEQ NR      2 TO ALOG SEQ NR      2
        OR FROM LOG INTERVAL BEGIN 20150628092730 TO LOG INTERVAL END
        20150628092730
***** LOG MODE : AFIM LOGGING _____ 4.
***** SUPPORTS OF ACTUAL LOG FILE :
        DEFAULT SUPPORT :PVS ID = DEFAULT PVS
        RESERVE SUPPORT :PVS ID = :SQL2:

```



\*\*\*\*\* INFORMATION ABOUT LOG HISTORY : \_\_\_\_\_ 5.

ALOG SEQ NR!	LOG INTERVAL	! AFIM	!BACKOUT!	LOGGING !
!	BEGIN ! END	!	! BFIM !	GAP !
4	!20150628092742!20150628092743!	* !	!	? !
3	!20150628092741!20150628092742!	* !	!	!
2	!20150628092730!20150628092741!	* !	!	!
1	!20150628092729!20150628092730!	* !	!	!

\*\*\*\*\* ONLINE COPIES BY ARCHIVE ARE ALLOWED \_\_\_\_\_ 6.

\*\*\*\*\* NORMAL END FUNCTION SHOW LOG INFORMATION AT 09:27:54

\*\*\*\*\* DIAGNOSTIC SUMMARY OF BMEND

NO WARNINGS  
 NO ERRORS  
 NO SYSTEM-ERRORS

\*\*\*\*\* END OF DIAGNOSTIC SUMMARY

\*\*\*\*\* NR OF DATABASE ACCESSES : 51

\*\*\*\*\* NORMAL END BMEND (UDS/SQL V2.8 0000 ) 2015-06-28 09:27:54

## Logging für das Datenbank-Original einschalten (START-LOG)

Mit der Anweisung START-LOG werden das AFIM-Logging für die Originaldatenbank eingeschaltet und eventuell neue Datenträger vereinbart, auf denen die weiteren ALOG-Dateien eingerichtet werden sollen.

### START-LOG

```
DEFAULT-SUPPORT = *PUBLIC(...) / *UNCHANGED / list-poss(15): *PRIVATE(...)
```

```
*PUBLIC(...)
```

```
  | CATID = *STD / *OWN / <catid>
  | ,VOLUME-SET = *STD / <catid>
  | ,VOLUME = *STD / list-poss(15): <volume>
```

```
*PRIVATE(...)
```

```
  | VOLUME = list-poss(15): <volume>
  | ,DEVICE = <device>
```

```
,RESERVE-SUPPORT = *PUBLIC(...) / *UNCHANGED / list-poss(15): *PRIVATE(...)
```

```
*PUBLIC(...)
```

```
  | CATID = *STD / *OWN / <catid>
  | ,VOLUME-SET = *STD / <catid>
  | ,VOLUME = *STD / list-poss(15): <volume>
```

```
*PRIVATE(...)
```

```
  | VOLUME = list-poss(15): <volume>
  | ,DEVICE = <device>
```

```
,SPACE = *STD / *RELATIVE(...) / *UNCHANGED
```

```
*RELATIVE(...)
```

```
  | PRIMARY-ALLOCATION = <integer 192..50331645>
  | ,SECONDARY-ALLOCATION = <integer 576..32767>
```

```
,USER-ACCESS = *OWNER-ONLY / *ALL-USERS
```

```
,RESET-LOG-POOL = *NO / *YES
```

**DEFAULT-SUPPORT = \*PUBLIC (...)**

Eine neu anzulegende ALOG-Datei wird auf gemeinschaftlicher Platte eingerichtet, wenn Sie nicht selbst ein Kommando CREATE-FILE für die neue ALOG-Datei gegeben haben.



Nur für einen der folgenden Operanden CATID, VOLUME-SET und VOLUME ist eine Angabe ungleich \*STD erlaubt.

Die Angaben werden nicht auf Verträglichkeit mit der für den BMEND-Lauf aktuellen UDS/SQL-Pubset-Deklaration geprüft. Aber Sie müssen beachten, dass die hier getroffenen Festlegungen mit den UDS/SQL-Pubset-Deklarationen der späteren Einsatzumgebungen (DBH-Betrieb, Dienstprogramme) verträglich sein müssen.

**CATID = \*STD / \*OWN / <catid>**

Bestimmt die Katalogkennung des Pubsets.

**CATID = \*STD**

Die Katalogkennung wird aus den mit dem Operanden VOLUME oder VOLUME-SET gemachten Angaben vom System bestimmt. Falls für beide Operanden keine Angabe gemacht oder \*STD angegeben wurde, wird die Standard-Katalogkennung der Konfigurationskennung genutzt.

**CATID = \*OWN**

Die Standard-Katalogkennung der Konfigurationskennung wird genutzt.

**CATID = <catid>**

Die angegebene Katalogkennung wird genutzt.

**VOLUME-SET = \*STD / <catid>**

Bestimmt den Volume-Set eines SM-Pubsets, auf dem die ALOG-Datei eingerichtet werden soll. Nichtprivilegierte Benutzer können Volume-Sets nur dann explizit angeben, wenn sie die Berechtigung zur physikalischen Allokierung auf dem betroffenen Pubset besitzen.

**VOLUME-SET = \*STD**

Der Volume-Set für ALOG-Dateien auf einem SM-Pubset wird vom System ausgewählt.

**VOLUME-SET = <catid>**

Explizite Angabe des Volume-Sets, auf dem die ALOG-Datei eingerichtet werden soll.

**VOLUME = \*STD / list-poss(15): <volume>**

Angabe der gemeinschaftlichen Platten, auf den die ALOG-Datei eingerichtet werden soll.

**VOLUME = \*STD**

Die Platten, auf denen die ALOG-Datei eingerichtet wird, werden vom System ausgewählt.

**VOLUME = list-poss(15): <volume>**

Eine neu anzulegende ALOG-Datei wird auf den angegebenen Platten angelegt. Diese können einem SF-Pubset oder einem Volume-Set eines SM-Pubsets zugeordnet sein. Es können maximal 15 VSNs angegeben werden, die verschieden sein müssen. Die VSN kann in PUB-Notation (PUBpxx) oder in Punkt-Notation (pp[pp].[xy]z) angegeben werden.

Alle angegebenen Platten müssen demselben Volume-Set angehören, d. h. sie müssen dieselbe Katalogkennung haben.

Für nichtprivilegierte Benutzer ist diese Angabe nur erlaubt, wenn sie die Berechtigung zur physikalischen Allokierung von gemeinschaftlichem Speicherplatz auf dem betroffenen Pubset haben.

**DEFAULT-SUPPORT = list-poss(15): \*PRIVATE (...)**

Eine neu anzulegende ALOG-Datei wird auf der angegebenen Platte bzw. auf den angegebenen Platten eingerichtet, wenn Sie nicht selbst ein Kommando CREATE-FILE für die neue ALOG-Datei gegeben haben. Sie können maximal 15 Platten angeben.

**VOLUME = list-poss(15): <volume>**

Angaben der privaten Platten, auf denen ALOG-Dateien angelegt werden können.

**DEVICE = <device>**

Definieren des Gerätetyps der privaten Platten.

**DEFAULT-SUPPORT = \*UNCHANGED**

Die Werte bleiben unverändert. \*UNCHANGED ist nur möglich, wenn das Logging bereits eingeschaltet ist.

**RESERVE-SUPPORT = \*PUBLIC (...)**

Die ALOG-Datei wird auf gemeinschaftlicher Platte eingerichtet, wenn Sie eine neu anzulegende ALOG-Datei nicht eingerichtet haben und diese auch nicht auf den mit DEFAULT-SUPPORT festgelegten Datenträgern eingerichtet werden kann.



Nur für einen der folgenden Operanden CATID, VOLUME-SET und VOLUME ist eine Angabe ungleich \*STD erlaubt.

Die Angaben werden nicht auf Verträglichkeit mit der für den BMEND-Lauf aktuellen UDS/SQL-Pubset-Deklaration geprüft. Aber Sie müssen beachten, dass die hier getroffenen Festlegungen mit den UDS/SQL-Pubset-Deklarationen der späteren Einsatzumgebungen (DBH-Betrieb, Dienstprogramme) verträglich sein müssen.

**CATID = \*STD**

Die Katalogkennung wird aus den mit dem Operanden VOLUME oder VOLUME-SET gemachten Angaben vom System bestimmt. Falls für beide Operanden keine Angabe gemacht oder \*STD angegeben wurde, wird die Standard-Katalogkennung der Konfigurationskennung genutzt.

**CATID = \*OWN**

Die Standard-Katalogkennung der Konfigurationskennung wird genutzt.

**CATID = <catid>**

Die angegebene Katalogkennung wird genutzt.

**VOLUME-SET = \*STD / <catid>**

Bestimmt den Volume-Set eines SM-Pubsets, auf dem die ALOG-Datei eingerichtet werden soll. Nichtprivilegierte Benutzer können Volume-Sets nur dann explizit angeben, wenn sie die Berechtigung zur physikalischen Allokierung auf dem betroffenen Pubset besitzen.

**VOLUME-SET = \*STD**

Der Volume-Set für ALOG-Dateien auf einem SM-Pubset wird vom System ausgewählt.

**VOLUME-SET = <catid>**

Explizite Angabe des Volume-Sets, auf dem die ALOG-Datei eingerichtet werden soll.

**VOLUME = \*STD / list-poss(15): <volume>**

Angabe der gemeinschaftlichen Platten, auf den die ALOG-Datei eingerichtet werden soll.

**VOLUME = \*STD**

Die Platten, auf denen die ALOG-Datei eingerichtet wird, werden vom System ausgewählt.

**VOLUME = list-poss(15): <volume>**

Eine neu anzulegende ALOG-Datei wird auf den angegebenen Platten angelegt. Diese können einem SF-Pubset oder einem Volume-Set eines SM-Pubsets zugeordnet sein. Es können maximal 15 VSNs angegeben werden, die verschieden sein müssen. Die VSN kann in PUB-Notation (PUBp<sub>xx</sub>) oder in Punkt-Notation (pp[pp].[xy]z) angegeben werden.

Alle angegebenen Platten müssen demselben Volume-Set angehören, d. h. sie müssen dieselbe Katalogkennung haben.

Für nichtprivilegierte Benutzer ist diese Angabe nur erlaubt, wenn sie die Berechtigung zur physikalischen Allokierung von gemeinschaftlichem Speicherplatz auf dem betroffenen Pubset haben.

**RESERVE-SUPPORT = list-poss(15): \*PRIVATE (...)**

Eine neu einzurichtende ALOG-Datei wird auf der angegebenen Platte bzw. den angegebenen Platten angelegt, wenn Sie die ALOG-Datei nicht eingerichtet haben und diese auch nicht auf den mit DEFAULT-SUPPORT festgelegten Datenträgern eingerichtet werden kann.

**VOLUME = list-poss(15): <volume>**

Angeben der privaten Platten, auf denen ALOG-Dateien angelegt werden können.

**DEVICE = <device>**

Definieren des Gerätetyps der privaten Platten.

**RESERVE-SUPPORT = \*UNCHANGED**

Die Werte bleiben unverändert. UNCHANGED ist nur möglich, wenn das Logging bereits eingeschaltet ist.



Ist der Operand DEFAULT mit dem gleichen Wert besetzt wie der Operand RESERVE, so ist das ein Fehler. Dieser Fehler wird nicht bei der Syntaxanalyse der START-LOG-Anweisung sondern erst beim Anlegen von ALOG-Dateien festgestellt. Es besteht dann keine Möglichkeit, auf ein Ersatzmedium auszuweichen.

**SPACE = \*STD**

Die neu anzulegende ALOG-Datei erhält als Primärzuweisung den Wert 192, als Sekundärzuweisung den Wert 576.

**SPACE = \*RELATIVE (...)**

Die neu anzulegende ALOG-Datei erhält als Primärzuweisung und Sekundärzuweisung die angegebenen Werte.

(Dabei muss der Wert  $\geq 192$  und  $\leq 50331645$  für Primärzuweisung bzw.  $\geq 576$  und  $\leq 32767$  für Sekundärzuweisung sein.)

**PRIMARY-ALLOCATION = <integer 192..50331645>**

Anzahl der PAM-Seiten für die Primärzuweisung.

**SECONDARY-ALLOCATION = <integer 576..32767>**

Anzahl der PAM-Seiten für die Sekundärzuweisung.

**SPACE = \*UNCHANGED**

Die Werte bleiben unverändert. \*UNCHANGED ist nur möglich, wenn das Logging bereits eingeschaltet ist.

**USER-ACCESS = \*OWNER-ONLY**

Legt fest, dass die ALOG-Datei nur von der Benutzerkennung angesprochen werden darf, in der sie angelegt ist.

**USER-ACCESS = \*ALL-USERS**

Legt fest, dass die ALOG-Datei auch von anderen Benutzerkennungen angesprochen werden darf.

**RESET-LOG-POOL = \*NO**

Die neu anzulegende ALOG-Datei wird angelegt mit der um 1 erhöhten ALOG SEQ NR aus dem Act-Key-0 des DBDIR.

**RESET-LOG-POOL = \*YES**

Die ALOG-Datei beginnt mit der ALOG SEQ NR = 1.

Mit diesem Operanden kann z.B. der Log Pool einer duplizierten Datenbank - beginnend mit ALOG SEQ NR = 1 - eingerichtet werden.



Sie können bis zu 15 Platten in einem PRIVATE-Operanden angeben und bis zu 15 Ausprägungen des PRIVATE-Operanden. Ist die Gesamtzahl der Platten, die Sie in einer Anweisung angeben, jedoch größer als 15, so gelten die letzten 15 Angaben.

Mehrere Ausprägungen des PRIVATE-Operanden sind somit nur sinnvoll, wenn die Platten verschiedenen Gerätetypen zugeordnet sind.

Kann auf eine ALOG-Datei nicht zugegriffen werden, weil sie z.B. gelöscht wurde, so wird von den Dienstprogrammen eine neue ALOG-Datei mit einer um 1 erhöhten ALOG-Nummer angelegt. Wenn in der gelöschten ALOG-Datei keine relevanten Informationen liegen (Konsistenzpunkt wurde nicht verlassen), dann kann auch über eine so entstandene Logging-Lücke hinweg in 2 Schritten (1. bis zur Lücke, 2. nach der Lücke) nachgefahren werden.

*Beispiel*

1. Einschalten des AFIM-Logging und der Online-Sicherungsfähigkeit, neu anzulegende ALOG-Dateien werden auf verschiedenen gemeinschaftlichen Platten angelegt.

```

/ASSIGN-SYSDTA TO-FILE=*SYSCMD
/SELECT-PRODUCT-VERSION PRODUCT-NAME=UDS-SQL,    VERSION=02.8A00
/START-UDS-BMEND
***** START    BMEND    (UDS/SQL V2.8 0000 ) 2015-09-18    09:27:29
//ALLOCATE-BUFFER-POOL BUFFER-SIZE=*STD
//OPEN-DATABASE DATABASE-NAME=VERSAND
***** DATABASE ORIGINAL WITHOUT AFIM LOGGING
        FUNCTION ENABLE NOT AVAILABLE
        FUNCTION KILL NOT AVAILABLE
***** CONSISTENT    DATABASE DIRECTORY
//START-LOG DEFAULT-SUPPORT=*PUBLIC(CATID=*OWN), -
//          RESERVE-SUPPORT=*PUBLIC(CATID=SQL2), -
//          SPACE=*STD, RESET-LOG-POOL=*NO
***** LOGGING WILL BE ACTIVATED
        FUNCTION ENABLE AVAILABLE FROM NOW ON
//ENABLE-ONLINE-COPY
//END
***** BEGIN          FUNCTION START LOGGING AT 09:27:29
ALOG FILE CREATED ACCORDING TO DEFAULT-SUPPORT
***** NORMAL    END FUNCTION START LOGGING AT 09:27:29
***** BEGIN          FUNCTION ENABLE ONLINE COPY AT 09:27:29
***** ONLINE COPY FOR DATABASE $XXXXXXXXX.VERSAND ALLOWED
***** NORMAL    END FUNCTION ENABLE ONLINE COPY AT 09:27:29

***** DIAGNOSTIC SUMMARY OF BMEND

                NO WARNINGS
                NO ERRORS
                NO SYSTEM-ERRORS

***** END OF DIAGNOSTIC SUMMARY
***** NR OF DATABASE ACCESSES :          147
***** NORMAL END    BMEND    (UDS/SQL V2.8 0000 ) 2015-09-18    09:27:29

```



2. Zurücksetzen der ALOG-SEQ-NR auf 1 und Angabe von drei Platten vom gleichen Gerätetyp, auf denen die nachfolgenden ALOG-Dateien neu eingerichtet werden sollen.

```

/ASSIGN-SYSDTA TO-FILE=*SYSCMD
/SELECT-PRODUCT-VERSION PRODUCT-NAME=UDS-SQL,VERSION=02.8A00
/START-UDS-BMEND
//OPEN-DATABASE DATABASE-NAME=dbname
//START-LOG
//          DEFAULT=*PRIVATE(VOLUME=(G3200A,G3200B,G3200C),-
//                                DEVICE=D3468)                ,-
//          RESERVE=*PRIVATE(VOLUME=G3400A
//                                DEVICE=D3468)                ,-
//          RESET-LOG-POOL=*YES
//END
***** BEGIN FUNCTION START LOGGING AT timestamp

***** NORMAL END FUNCTION START LOGGING AT timestamp

```

3. Vorgabe von drei Platten verschiedener Gerätetypen, auf denen die nachfolgenden ALOG-Dateien neu eingerichtet werden sollen.

```

/ASSIGN-SYSDTA TO-FILE=*SYSCMD
/SELECT-PRODUCT-VERSION PRODUCT-NAME=UDS-SQL,VERSION=02.8A00
/START-UDS-BMEND
//OPEN-DATABASE DATABASE-NAME=dbname
//START-LOG DEFAULT=(*PRIVATE(VOLUME=G3030M,                ,-
//                                DEVICE=D3468)                ,-
//                                *PRIVATE(VOLUME=(G2065C,G2065D)
//                                DEVICE=D5804)                ),-
//          RESERVE=*PUBLIC
//END
***** BEGIN FUNCTION START LOGGING AT timestamp

***** NORMAL END FUNCTION START LOGGING AT timestamp

```

*timestamp* gibt den aktuellen Zeitpunkt an.

## Logging für den Datenbankbetrieb ausschalten (STOP-LOG)

Mit der Anweisung STOP-LOG wird das Logging ausgeschaltet. Vor dem Ausschalten des Loggings nimmt BMEND für alle Realms eine mögliche Kennzeichnung der Online-Sicherungsfähigkeit zurück.

STOP-LOG

Diese Anweisung hat keine Operanden.

## Anweisung rückgängig machen (UNDO)

Die zuletzt eingegebene, korrekte Anweisung (außer UNDO selbst) wird nicht ausgeführt.

Mit einer weiteren Anweisung UNDO wird die vorletzte Anweisung (außer UNDO) nicht ausgeführt usw.

Die Anweisung UNDO setzt keine Anweisung ALLOCATE-BUFFER-POOL oder END zurück.

UNDO

Diese Anweisung hat keine Operanden.

## AFIMs in eine Datenbank einspielen (UPDATE-DATABASE)

Mit der Anweisung UPDATE-DATABASE können Sie Realms nachfahren durch Einspielen von AFIMs aus ALOG-Dateien.

Die Realms sind zuvor durch Kopieren, Einspielen von ARCHIVE-Sicherungen oder Umkatalogisieren bereitgestellt worden. Sie repräsentieren einen älteren Stand der Datenbank.

Abgeschaltete Realms können auch bearbeitet werden, wenn BMEND parallel zum DBH läuft.

Inkonsistente Realms müssen mit der Anweisung UPDATE-DATABASE konsistent gemacht werden, bevor sie zugeschaltet oder abgeschaltet werden.

### UPDATE-DATABASE

```

REALM-NAME = *ALL / *ALL-EXCEPT(...) / list-poss(30): <realmname>
  *ALL-EXCEPT(...)
    | NAME = list-poss(30): <realmname>
,DEADLINE = *STD / *BREAK-POINT / <alog-seq-nr> / *TIME-STAMP(...)
  *TIME-STAMP(...)
    | DATE = <date>
    | ,TIME = <time>
,DELETE = *NO / *YES

```

### **REALM-NAME = \*ALL**

Alle Realms werden nachgefahren.

### **REALM-NAME = \*ALL-EXCEPT(...)**

Mit Ausnahme der angegebenen Realms werden alle Realms nachgefahren.

#### **NAME = list-poss(30): <realmname>**

Name der Realms, die nicht nachgefahren werden sollen.

### **REALM-NAME = list-poss(30): <realmname>**

Alle angegebenen Realms werden nachgefahren.

### **DEADLINE = \*STD**

Die Datenbank wird bis zum Ende der letzten abgeschlossenen ALOG-Datei nachgefahren. Ein eventuell in der aktuellen, nicht abgeschlossenen ALOG-Datei verzeichneter Konsistenzpunkt kann mit DEADLINE = \*STD nicht erreicht werden.

**DEADLINE = \*BREAK-POINT**

Alle ALOG-Dateien einschließlich der letzten, aktuellen und möglicherweise inkonsistenten ALOG-Datei werden nachgefahren. Somit ist das Nachfahren bis zum Zeitpunkt eines Absturzes möglich. Falls die letzte ALOG-Datei inkonsistent ist, ist anschließend ein Warmstart der Datenbank nötig.

Realms der Schattendatenbank werden nur mit abgeschlossenen ALOG-Dateien nachgefahren.

**DEADLINE = <alog-seq-nr>**

Nummer der ALOG-Datei, bis zu der einschließlich AFIMs eingespielt werden. Es werden nur abgeschlossene ALOG-Dateien nachgefahren. (Führende Nullen bei der Angabe der *alog-seq-nr* können weggelassen werden.)

**DEADLINE = \*TIME-STAMP(...)**

Die Datenbank wird bis einschließlich zur letzten abgeschlossenen ALOG-Datei nachgefahren, deren LOG\_INTERVAL\_END kleiner oder gleich \*TIME-STAMP(...) ist.

**DATE = <date>**

Datum, das den Nachfahrvorgang begrenzt.

**TIME = <time>**

Uhrzeit, die den Nachfahrvorgang begrenzt; ist dabei der Zeitpunkt nicht eindeutig, wird Sommerzeit angenommen und eine Warnung ausgegeben.

**DELETE = \*NO**

Die eingespielten ALOG-Dateien bleiben auf Platte stehen.

**DELETE = \*YES**

Mit Ausnahme der aktuellen Datei werden die eingespielten ALOG-Dateien nach dem Nachfahren automatisch gelöscht, falls bei dieser Anweisung kein Fehler aufgetreten ist.

Der DBDIR und der DBCOM werden genauso behandelt wie jeder andere Realm. Bei den Angaben \*ALL und \*ALL-EXCEPT wird implizit der Realm DBCOM mitangesprochen.

Liegen nicht alle Realms vor, die durch REALM-NAME spezifiziert sind, so werden die fehlenden Realms nicht bearbeitet. Außerdem unterbleibt ein Löschen der ALOG-Datei, falls es gefordert ist.

Die Angabe des Operanden DEADLINE = \*BREAK-POINT ist bei der Schattendatenbank nicht sinnvoll. In diesem Fall wird \*BREAK-POINT wie \*STD behandelt.

Die Folgennummer der nächsten, zum Erreichen der gewünschten DEADLINE erforderlichen ALOG-Datei wird nach dem Einspielen jeder einzelnen ALOG-Datei in einer Jobvariablen gesichert (siehe [Abschnitt „Versorgung der BMEND-Jobvariablen“ auf Seite 64](#)).

Enthält die ALOG-Dateien-Sequenz, die nachzufahren ist, eine Logging-Lücke, so endet an diesem Punkt der Nachfahrvorgang.

Es wird eine Warnung ausgegeben, wenn die DEADLINE (Angabe von *alog-seq-nr* bzw. \*TIME-STAMP(...)) nicht erreicht wurde.

Ein Realm ungleich DBDIR kann nur dann nachgefahren werden, wenn er in der Update-Session angesprochen wurde oder wenn er zum DBDIR passt und der DBDIR ebenfalls nachgefahren wird.

### Statistik- und Endeausgaben

Am Ende des BMEND-Laufes werden für den SUMMARY-REPORT interne Zähler ausgewertet und ausgegeben.

```
***** DIAGNOSTIC SUMMARY OF BMEND

      { NO }   WARNINGS
      {   }   ERRORS
      { anz }  SYSTEM ERROR

***** END OF DIAGNOSTIC SUMMARY
***** NR OF DATABASE ACCESSES:  anzahl
```

## 2.2.4 Kommandofolge zum Starten von BMEND

Die hier beschriebene Kommandofolge geht davon aus, dass UDS/SQL mit IMON installiert wurde (siehe Abschnitt „START-Kommandos der UDS/SQL-Programme“ in Kapitel 2 des Handbuchs „[Aufbauen und Umstrukturieren](#)“).

```
01 [/CREATE-JV JV-NAME=JOBVAR,PROTECTION=*STD]
02 [/SET-JV-LINK LINK-NAME=JVBMEND,JV-NAME=JOBVAR]
03 [/ADD-FILE-LINK LINKNAME=DATABASE,
    FILE-NAME=[ :catid: ][ $userid. ] dbname.DBDIR]
04 /SELECT-PRODUCT-VERSION PRODUCT-NAME=UDS-SQL,VERSION=version,SCOPE=*TASK
05 /START-UDS-BMEND
06 [//OPEN-DATABASE DATABASE-NAME=dbname
    [ ,COPYNAME=*NONE/copyname]
    [ ,USER-IDENTIFICATION=*OWN/userid]
07 bmend-anweisungen
08 //END
```

01, 02 Einrichten einer Jobvariablen

03, 06 Sie müssen eine der beiden Anweisungen angeben.

04 Die Version des Dienstprogramm wird ausgewählt.  
Die Angabe der Version wird generell empfohlen, da mehrere UDS/SQL-Versionen parallel installiert sein können.

05 Aliasnamen für den Aufruf sind START-UDS-REPAIR und BMEND.

Beispiele zum Einsatz von BMEND finden Sie im Handbuch „[Datenbankbetrieb](#)“.

## 2.3 Versorgung der BMEND-Jobvariablen

Um einen automatischen Sicherungsbetrieb und Recovery-Betrieb zu realisieren, versorgt das Dienstprogramm BMEND eine Jobvariable. Diese Jobvariable können andere Programme bzw. Prozeduren zur Steuerung verwenden.

Die Jobvariable wird versorgt, wenn eine Jobvariable mit LINK-NAME=JVBMEND eingerichtet ist.

BMEND benutzt den Inhalt der Jobvariablen nicht als Eingabe, sondern aktualisiert nur am Ende bestimmter Funktionen relevante Werte.

Die Versorgung dieser Jobvariablen gliedert sich - intern - in zwei Teile:

1. Initialisieren durch SHOW-LOG-INFORMATION
2. Aktualisieren durch UPDATE-DATABASE

Durch das Initialisieren der Jobvariablen (Anweisung SHOW-LOG-INFORMATION) werden erste Entscheidungshilfen gegeben, um Recovery-Maßnahmen einzuleiten.

Sie als Anwender müssen anschließend entscheiden, ob die gelieferten Informationen ausreichend sind. Möglicherweise müssen Sie die Initialisierung wiederholen, indem Sie in der Anweisung SHOW-LOG-INFORMATION den Operanden LOG-FILE modifizieren.

Die Jobvariable enthält folgende Informationsbereiche:

- Zustand des bearbeiteten DBDIR
- Zustand eines Log Pool-Ausschnitts (allgemein)
- Zustand eines Log Pool-Ausschnitts bzgl. AFIM-Logging
- Folge Nummern für weitere Einspielvorgänge

### Anweisungsfolge

```
/CREATE-JV JV-NAME=JOBVAR, PROTECTION=*STD
/SET-JV-LINK LINK-NAME=JVBMEND, JV-NAME=JOBVAR
/ADD-FILE-LINK LINK-NAME=DATABASE, FILE-NAME=VERSAND.DBDIR
/SELECT-PRODUCT-VERSION PRODUCT-NAME=UDS-SQL, VERSION=version, SCOPE=*TASK
/START-UDS-BMEND
//SHOW-LOG-INFORMATION LOGFILE=*STD, OUTPUT=*SYSOUT
//END
.
.
```



**Struktur der Jobvariablen**

DISPL

( 0) DBDIR_DATA	beschreibt Zustand des DBDIR, auf den zugegriffen wird; wird von UPDATE gelöscht
( 0) ALOG_SEQ_NR	ALOG_SEQ_NR aus DBDIR
( 4) CONSISTENCY_DATA	BACK_UP_DATA aus DBDIR; Zeitpunkt der letzten Änderung in der Datenbank
( 18) CONSISTENT	CONSISTENCY / C'Y' oder C'N' C'N': wenn SYSTEM_BREAK in AK0 oder AKn gesetzt oder wenn AK0 ungleich AKn ist
( 19) FILLER	
( 20) COMMON_LOG_DATA	gibt Grenzen der untersuchten ALOG-Dateien-Sequenz an; ältester und jüngster Eintrag der Historie wird von UPDATE gelöscht
( 20) HIGHEST_ALOG_SEQ_NR	höchste Folgennummer, deren Historie in der zugewiesenen ALOG-Datei hinterlegt ist
( 24) LOWEST_ALOG_SEQ_NR	niedrigste Folgennummer, deren Historie in der zugewiesenen ALOG-Datei hinterlegt ist
( 28) LOG_POOL_PART_END_DATA	YYYYMMDDHHMMSS; LOG_INTERVAL_END der im LOG-FILE-Operanden zugewiesenen ALOG-Datei
( 42) LOG_POOL_PART_BEGIN_DATA	YYYYMMDDHHMMSS
( 56) AFIM_LOG_DATA	Angaben zum jüngsten Log-Intervall mit lückenloser AFIM-Protokollierung aus der Historie; falls untersuchter Bereich ohne AFIMs, dann gleich 0; wird von UPDATE gelöscht
( 56) UPPER_ALOG_SEQ_NR	höchste Folgennummer des untersuchten Log Pool-Ausschnitts mit AFIMs
( 60) LOWER_ALOG_SEQ_NR	niedrigste Folgennummer des untersuchten Log Pool-Ausschnitts mit AFIMs
( 64) UPPER_ALOG_DATA	LOG_INTERVAL_END der ALOG-Datei
( 78) LOWER_ALOG_DATA	LOG_INTERVAL_BEGIN der ALOG-Datei

( 92) BACKOUT_LOG_DATA	gibt Grenzen des jüngsten lückenlosen BACKOUT-Bereichs an
( 92) UPPER ALOG_SEQ_NR	reserviert für zukünftige Erweiterungen
( 96) LOWER ALOG_SEQ_NR	reserviert für zukünftige Erweiterungen
(100) UPPER ALOG_DATA	reserviert für zukünftige Erweiterungen
(114) LOWER ALOG_DATA	reserviert für zukünftige Erweiterungen
(128) NEXT_SEQ_NR	enthält Folgennummer der als nächstes einzuspielenden ALOG-Datei
(128) UPDATE_START_SEQ_NR	kleinste ALOG-Nummer aller betrachteten Re- alms initialisiert durch SHOW-LOG; wird nach Einspielen von allen AFIMs einer ALOG-Datei um 1 erhöht; beim Erreichen von DEADLINE auf 0 gesetzt
(132) RESET_START_SEQ_NR	
(136) CHAR_TYPE_LOG_DATA	Angabe in abdruckbarer Darstellung
(136) DBNAME	Name der bearbeiteten Datenbank
(153) COPYNAME	CHAR (8); COPYNAME der bearbeiteten Schattendaten- bank
(161) ALOG_SEQ_CHAR	CHAR (9); ALOG_SEQ_NR aus DBDIR
(170) UPDATE_START_SEQ_CHAR	CHAR (9); UPDATE_START_SEQ_NR
(179) END	

## Ergebnisse der Initialisierung

Zur Initialisierung werden die Informationen aus der ALOG BOTTOM PAGE der zugewiesenen ALOG-Datei und aus dem DBDIR genutzt.

Die Differenz zwischen niedrigster und höchster Folgenummer der ALOG-Dateien-Sequenz kann auch geringer als 63 sein. Diese Situation ist möglich bei geringer Anzahl von ALOG-Dateien sowie nach einem inkonsistenten Wechsel der ALOG-Datei (alte ALOG-Datei nicht mehr im Zugriff).

Wird von einer **Originaldatenbank** ausgegangen, erhalten Sie durch die Initialisierung der Jobvariablen folgende Informationen:

- Konsistenz des Original-DBDIR
- Zeitpunkt der letzten Änderung des DBDIR
- höchste Folgenummer der ALOG-Datei-Sequenz
- das jüngste Intervall mit AFIM-Logging wird ausgegeben (Grenzwerte als Folgenummern und Zeitstempel)

Der Zeitpunkt einer maximalen DEADLINE ist hinterlegt worden (LOG END des jüngsten AFIM-Loggingintervalls).

Mit dem gelieferten Wert können Sie für die Funktion UPDATE einen geeigneten Sicherungsbestand einspielen.

Da nur die letzten 63 Logdateien untersucht wurden, ist nicht immer erkennbar, ob der Zeitraum des lückenlosen Loggings weiter in die Vergangenheit zurückreicht.

Muss die ALOG-Datei-Sequenz weiter zurück untersucht werden, so ist eine erneute Initialisierung der Jobvariablen erforderlich. Dabei ist es sinnvoll, mit der niedrigsten *alog-seq-nr* der untersuchten Sequenz zu starten, um eine Überlappung der Ausschnitte zu erhalten.

Wurde eine **Schattendatenbank** zugewiesen, so können Sie aus dem DBDIR keine Informationen gewinnen, mit welcher Folgenummer der ALOG-Dateien die Datenbank zuletzt prozessiert worden ist. Die höchste Folgenummer der belegten ALOG-Dateien können Sie nur ermitteln, wenn Sie die Originaldatenbank zuweisen. Bei Schattendatenbanken kann nur die Folgenummer zum Zeitpunkt der Sicherung des DBDIR ausgegeben werden.

Die Jobvariablenfelder werden bei einer Schattendatenbank genauso versorgt wie bei der Zuweisung eines Originals.

### Aktualisierungen durch UPDATE-DATABASE

Durch die Anweisung UPDATE-DATABASE werden alle Werte der Jobvariablen, die den DBDIR bzw. das BACKOUT-Logging betreffen, gelöscht.

Wurde eine ALOG-Datei eingespielt, wird die Folgenummer der ALOG-Datei eingetragen, die als nächste eingespielt werden soll. Ist DEADLINE erreicht, wird binär Null eingetragen.

### Anweisungsfolge

```
/CREATE-JV JV-NAME=JOBVAR,PROTECTION=*STD
/SET-JV-LINK LINK-NAME=JVBMEND,JV-NAME=JOBVAR
/ADD-FILE-LINK LINK-NAME=DATABASE,FILE-NAME=VERSAND.DBDIR
/SELECT-PRODUCT-VERSION PRODUCT-NAME=UDS-SQL,VERSION=version,SCOPE=*TASK
/START-UDS-BMEND
//UPDATE-DATABASE REALM-NAME=*ALL,DEADLINE=*STD,DELETE=*NO
//END
.
.
```

---

## 3 Konsistenz einer Datenbank prüfen mit BCHECK

Für einen ordnungsgemäßen Datenbankbetrieb müssen die physischen Strukturen der Datenbank unbedingt korrekt sein. Verletzungen der physischen Konsistenz auf Grund von Systemfehlern können jedoch nie ganz ausgeschlossen werden. Solche Verletzungen werden im Gegensatz zu Inkonsistenzen aus DBH-Sicht als Konsistenzfehler bezeichnet. Mit dem UDS/SQL-Dienstprogramm BCHECK sind Sie in der Lage, aus gegebenem Anlass oder im Rahmen der periodischen Datensicherung die physischen Strukturen Ihrer Datenbank zu überprüfen und so evtl. vorhandene Konsistenzfehler frühzeitig aufzudecken. Da BCHECK jeden Fehler genau lokalisieren kann, ist es möglich, eine fehlerhafte Datenbank zu korrigieren und so Folgefehler zu vermeiden.

BCHECK prüft eine UDS/SQL-Datenbank nach dem Redundanzprinzip. Zunächst sind dies die vorgefundenen Daten einer Seite einerseits und die vorgegebenen physischen Strukturen sowie die Metadaten im DBDIR in Form der SIA andererseits (lokale Konsistenz). Anschließend prüft BCHECK die Konsistenz logisch zusammengehöriger Systemdaten, die in unterschiedlichen Seiten liegen, ebenfalls unter Zuhilfenahme der SIA (globale Konsistenz).

Prüfungen der logischen Konsistenz der Benutzerdaten sind nicht möglich, da UDS/SQL keine Informationen über Referenzdaten bzw. die Gültigkeit von Benutzerdaten in der Datenbank vorliegen.

Prüfen lassen sich die Benutzerrealms der Datenbank, die PRIVACY-AND-IQF-Datenbank im DBDIR und die Compilerdatenbank im DBCOM. BCHECK prüft nur Realms auf Platte.

Sie müssen nicht immer die gesamte Datenbank prüfen lassen. Es ist auch möglich, die Prüfbreite auf einzelne Realms, Satzarten, Sets oder SEARCH-Keys und auf Konsistenzkriterien einzuschränken und so Zeit einzusparen. Ebenso können Sie die Prüftiefe einschränken, indem Sie die Verwaltungsstrukturen zum Abspeichern von Sätzen bzw. Schlüsselwerten von ASC-Keys bzw. DESC-Keys oder SEARCH-Keys von der Prüfung ausschließen.

Ab UDS/SQL V2.6 nutzt BCHECK beim Einlesen der Datenbankseiten I/O-Transferrängen über 32 K. Dabei wird mit dem Minimum der möglichen I/O-Transferrängen aller Platten, auf denen die betreffenden Datenbankrealms liegen, gearbeitet.

## 3.1 Beschreibung des Prüfverfahrens

Um die Zahl der Datenbankzugriffe zu optimieren, prüft BCHECK die Datenbank in einem einmaligen Durchgang.

Zunächst prüft BCHECK lokal in den angegebenen Realms:

- die Act-Key-0-Seite und die Act-Key-N-Seite,
- FPA-Seiten, DBTT-Ankerseiten und DBTT-Seiten,
- den Seitenkopf aller Datenbankseiten,
- den formalen Aufbau von CALC- und Tabellenseiten einschließlich Listen der Stufe 0,
- in direkten CALC-Seiten die Übereinstimmung der Schlüsselwerte in den Sätzen mit denen der CALC-Tabellenzeile,
- in den Seitenindex-Einträgen die Distanzen zu den Sätzen und
- in den Tabellenseiten die Sortierfolge der Schlüssel bzw. der Satzfolgennummern

Er prüft somit die inneren Seitenstrukturen der Realms.

Anschließend prüft BCHECK die globalen, d.h. seitenübergreifenden Beziehungen der angegebenen Prüfobjekte. Dazu liest er von den zu prüfenden Satzarten, Sets und SEARCH-Keys die zugehörigen DBTT-Einträge, SCD-Einträge, Tabellenköpfe, Tabellen-Einträge und Sätze und bildet aus dieser Information so genannte Prüfsätze.

### 3.1.1 Prüfmodus einstellen

Für die folgende Fehleranalyse bestimmen Sie den Modus, in welchem BCHECK die Prüfsätze weiterverarbeitet. Sie können wählen zwischen der *summarischen Prüfung* und der *Sortierungsprüfung*.

Bei beiden Modi nutzt BCHECK den Umstand aus, dass in den verschiedenen Seitenarten (DBTT-Seiten, Datenseiten etc.) die Informationen über die Prüfobjekte redundant gespeichert sind. Beispielsweise lässt sich aus einer DBTT-Seite zu jedem DB-Key die Seitenadresse des Satzes ermitteln. DB-Key und Seitenadresse stehen, wenn die Datenbank konsistent ist, auch in der Datenseite, in der der Satz gespeichert ist. Das bedeutet, dass aus den verschiedenen Datenbankseiten für jedes Prüfobjekt zwei gleiche Prüfsätze erzeugt werden, die, je nachdem woher sie stammen, ein positives Vorzeichen bzw. ein negatives Vorzeichen erhalten.

BCHECK verwendet zwei Verfahren, um Konsistenzfehler in der Datenbank festzustellen:

### 1. Zählerverfahren

BCHECK richtet drei Zähler ein:

- in einem zählt er, ob genauso viele positive wie negative Prüfsätze existieren, was bei einer konsistenten Datenbank der Fall ist;
- in den beiden anderen addiert er die auf 6 byte zusammengefassten Prüfsätze bzw. das Quadrat dieser Werte.

Diese Technik, die auf der Theorie der fehlererkennenden Codes beruht, gewährleistet mit hoher Genauigkeit, dass ein vorhandener Fehler dadurch erkannt wird, dass die zusammengehörigen Summen nicht gleich sind.

Das Zählerverfahren kann Konsistenzfehler in der Datenbank zwar erkennen und eingrenzen, aber nicht genau lokalisieren. Dieses Verfahren wird auch als summarische Prüfung bezeichnet.

### 2. Sortierverfahren

BCHECK sammelt alle Prüfsätze in einer Datei und sortiert sie, sodass jeweils zusammengehörige Mengen von Prüfsätzen nebeneinander stehen müssen, wenn die Datenbank korrekt ist. Diese Mengen vergleicht er dann der Reihe nach; bei Ungleichheit ist an der betreffenden Stelle der Datenbank ein Konsistenzfehler.

Das bedeutet, BCHECK kann bei diesem Prüfverfahren Konsistenzfehler nicht nur erkennen, sondern auch lokalisieren, sodass die Konsistenzfehler behoben werden können. Durch das Sortieren der Prüfsätze arbeitet dieses Verfahren allerdings mit einem erheblich größeren Zeitaufwand als das Zählerverfahren.

## Summarische Prüfung

In Abhängigkeit von BCHECK-Anweisungen verwendet BCHECK das Zählerverfahren, das Sortierverfahren bzw. innerhalb der summarischen Prüfung implizit eine Kombination aus beiden. Da in der summarischen Prüfung das Sortierverfahren nur für eine begrenzte Menge von Prüfsätzen verwendet wird, arbeitet die summarische Prüfung mit geringem Zeitaufwand und lässt sich sehr gut bei der täglichen Sicherung einsetzen. Werden globale summarische Konsistenzfehler gemeldet, so müssten Sie mit der Sortierungsprüfung die fehlerhafte Stelle lokalisieren.

## Sortierungsprüfung

Hierbei verwendet BCHECK nur das Sortierverfahren. Ohne weitere Vorkehrungen arbeitet diese Prüfung mit einem bedeutend größeren Zeitaufwand als die summarische Prüfung.

Diesen Zeitaufwand können Sie erheblich verringern, indem Sie eine Inkrementalprüfung durchführen, ein Verfahren, das im Folgenden beschrieben ist und im Übrigen auch für die summarische Prüfung anwendbar ist.

### 3.1.2 Umfang der Prüfung festlegen

Neben dem Prüfmodus können Sie noch den Umfang der Prüfung festlegen. BCHECK kennt eine *Totalprüfung* und eine *Inkrementalprüfung*.

#### **Totalprüfung**

BCHECK prüft die Datenbank in ihrer Gesamtheit auf Konsistenz.

Sie können die Originaldatenbank oder eine Schattendatenbank prüfen lassen.

Eine Totalprüfung müssen Sie auf jeden Fall dann machen, wenn Sie keine konsistente Sicherung Ihrer Datenbank haben bzw. diese bisher nicht überprüft haben.

Ebenso ist eine Totalprüfung notwendig, wenn Sie Ihre Datenbank umstrukturiert oder reorganisiert haben und von der veränderten Datenbank noch keine konsistente Sicherung haben bzw. diese noch nicht überprüft haben.

#### **Inkrementalprüfung**

BCHECK prüft nur die geänderten Seiten bezüglich einer Schattendatenbank eines älteren Standes. Die geänderten Seiteninhalte kann BCHECK sich beschaffen, indem er die ältere Schattendatenbank mit der Originaldatenbank oder mit einer jüngeren Schattendatenbank vergleicht. Die ältere Schattendatenbank benötigt nur dann den DBDIR, wenn dieser geändert wurde.

Eine Inkrementalprüfung spart gegenüber der Totalprüfung erheblich Zeit ein, da BCHECK in den meisten Fällen nicht alle Seiten der Datenbank prüfen muss. Dem Mehraufwand des doppelten Lesens jeder nichtleeren Seite steht der Minderaufwand des Prüfens nur der geänderten Seiten gegenüber und - im SORTING-Modus - eine im Allgemeinen wesentlich kleinere Menge zu sortierender Daten.

Im SUMMING-Modus sind die Performanceunterschiede zwischen Totalprüfung und Inkrementalprüfung normalerweise nicht sehr groß; allgemeine Regeln über die Performance der verschiedenen Prüfmodi in Kombination mit dem Umfang der Prüfung lassen sich aber nicht aufstellen.



### 3.1.3 Prüfung auf Kohärenz

BCHECK prüft, ob die Realms, die Sie als zusammengehörig angegeben haben, auch wirklich derselben Version angehören, d.h. kohärent sind. Dazu liest er von jedem angegebenen Realm die im DBDIR eingetragene interne Versionsnummer und den Zeitpunkt der letzten Änderung (siehe Konsistenz-Zeitstempel) und vergleicht diese Werte mit den im Realm selbst eingetragenen Werten. Nur die Realms, bei denen BCHECK Übereinstimmung mit dem DBDIR feststellt, sind kohärent.

Nach Dienstprogrammläufen mit BALTER oder BREORG ist die Kohärenz innerhalb einer Inkrementalprüfung nicht gegeben.

## 3.2 Systemumgebung

### Auswirkungen auf den Datenbankbetrieb

Greift BCHECK bei dem Prüflauf auf die Originaldatenbank zu, so spricht man von einem Online-Prüflauf, der parallel zu einem SHARED-RETRIEVAL-Datenbankbetrieb möglich ist.

Wenn BCHECK nicht auf die Originaldatenbank zugreift, spricht man von einem Offline-Prüflauf, der parallel zu jedem Datenbankbetrieb stattfinden kann.

Prüfläufe können zueinander parallel ablaufen.

Vor jedem BCHECK-Prüflauf müssen Sie die zu prüfende Datenbank bzw. Schattendatenbank mit folgendem Kommando zuweisen:

```
/ADD-FILE-LINK LINK-NAME=DATABASE,FILE-NAME=dbname.DBDIR[.copyname]
```

BCHECK berücksichtigt beim Start ggf. eine zugewiesene UDS/SQL-Pubset-Deklaration (siehe Handbuch „[Datenbankbetrieb](#)“, Pubset-Deklarations-Jobvariable). Eine fehlerhafte Zuweisung führt zum Programmabbruch.

### Kohärenzprüfung

BCHECK prüft bei jedem Prüflauf die Datenbanken auf Kohärenz, auf deren DBDIR er zugreifen kann. Er prüft auf Kohärenz:

- **bei Totalprüfung:**  
die Originaldatenbank bzw. die Schattendatenbank
- **bei Inkrementalprüfung Original  $\uparrow\uparrow$  Schattendatenbank:**  
die Originaldatenbank und die Schattendatenbank, falls deren DBDIR vorhanden ist
- **bei Inkrementalprüfung Schattendatenbank-neu  $\uparrow\uparrow$  Schattendatenbank-alt:**  
die neuere Schattendatenbank und die ältere Schattendatenbank, falls deren DBDIR vorhanden ist

Im Folgenden sind die bei den verschiedenen Prüfläufen benötigten Datenbankdateien dargestellt:

*Totalprüfung Original*

Original jedes zu prüfenden Realm

Original des DBDIR

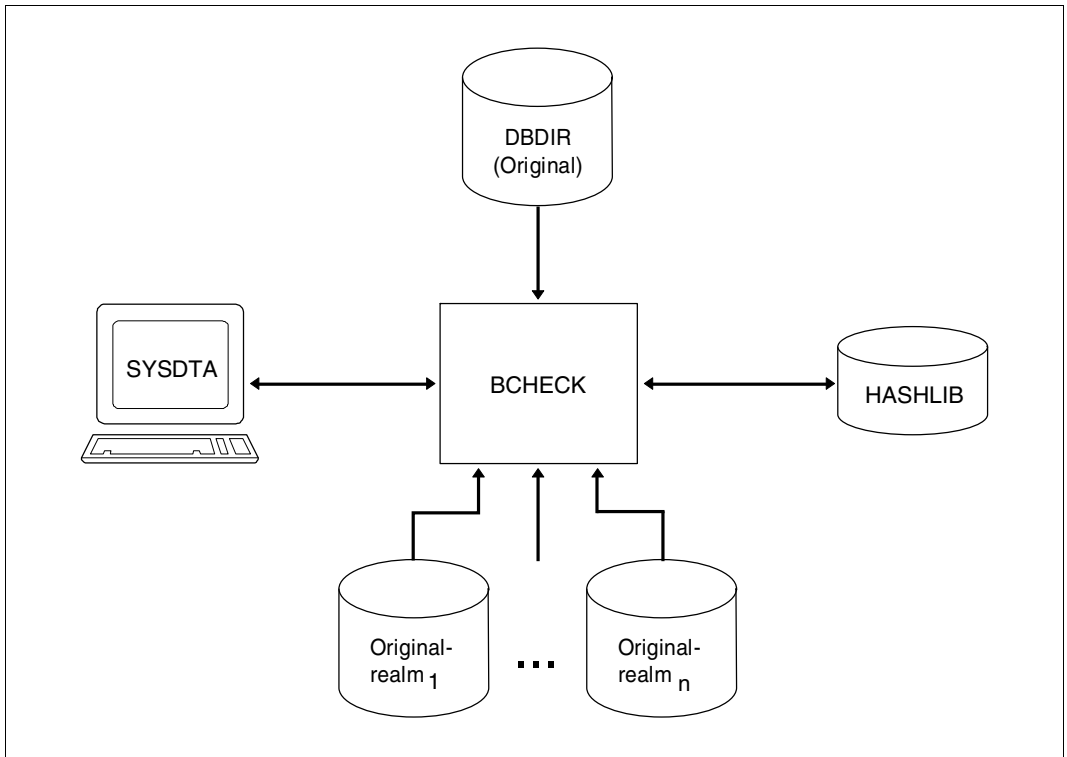


Bild 4: Systemumgebung bei einer Totalprüfung von Originalrealms

*Totalprüfung Schattendatenbank*

zu prüfende Realms der Schattendatenbank

DBDIR der Schattendatenbank

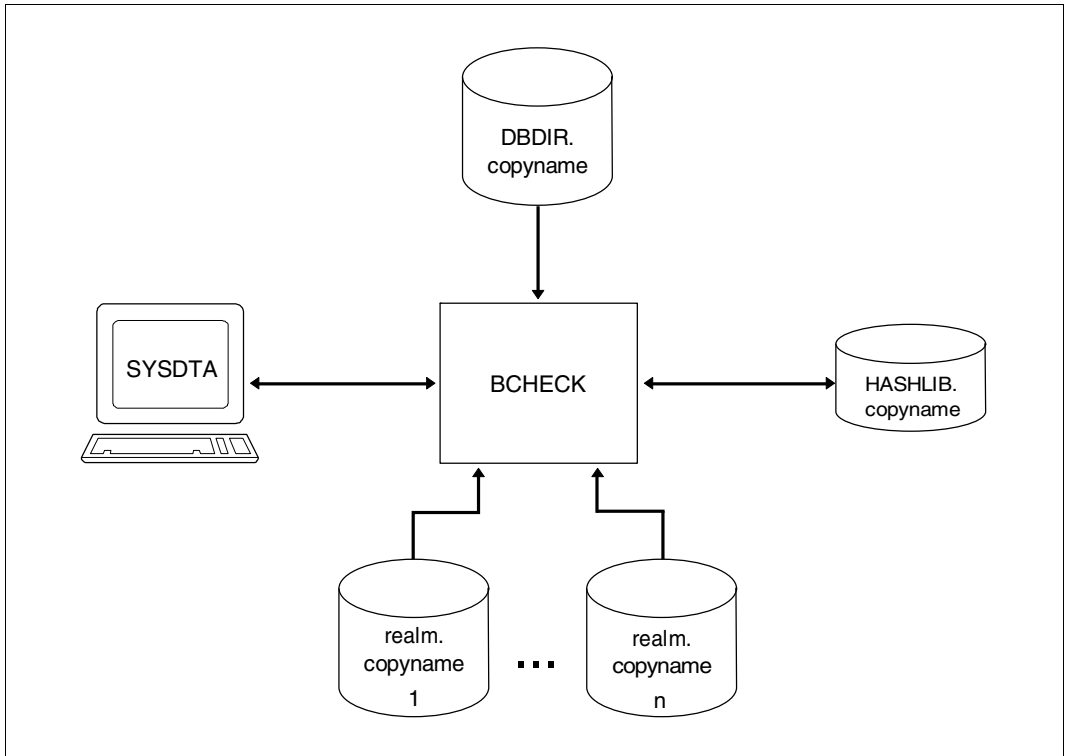


Bild 5: Systemumgebung bei einer Totalprüfung der Schattendatenbank

*Inkrementalprüfung Original ↔ Schattendatenbank*

Original jedes zu prüfenden Realm

Original des DBDIR

Realms der Schattendatenbank

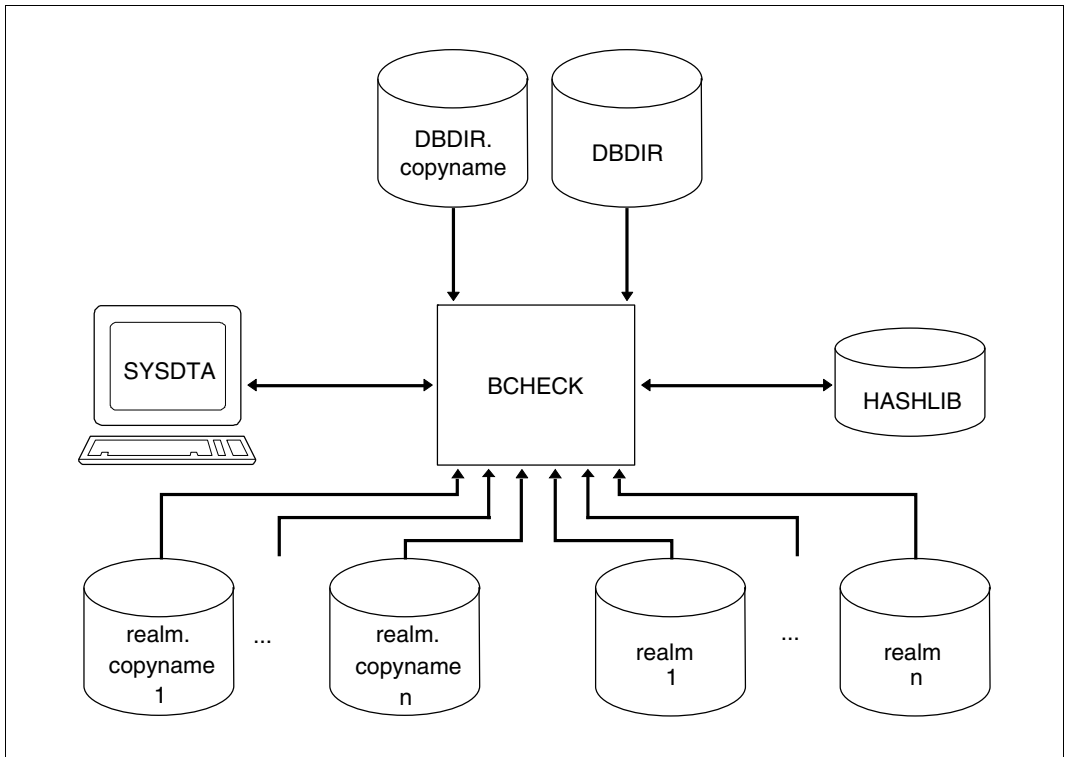
evtl. DBDIR der Schattendatenbank (siehe [Seite 74](#))

Bild 6: Systemumgebung bei einer Inkrementalprüfung Original ↔ Schattendatenbank

*Inkrementalprüfung Schattendatenbank-neu ↔ Schattendatenbank-alt*

die zu prüfenden Realms jeder Schattendatenbank

DBDIR zur älteren Schattendatenbank oder beide DBDIR's (siehe [Seite 74](#))

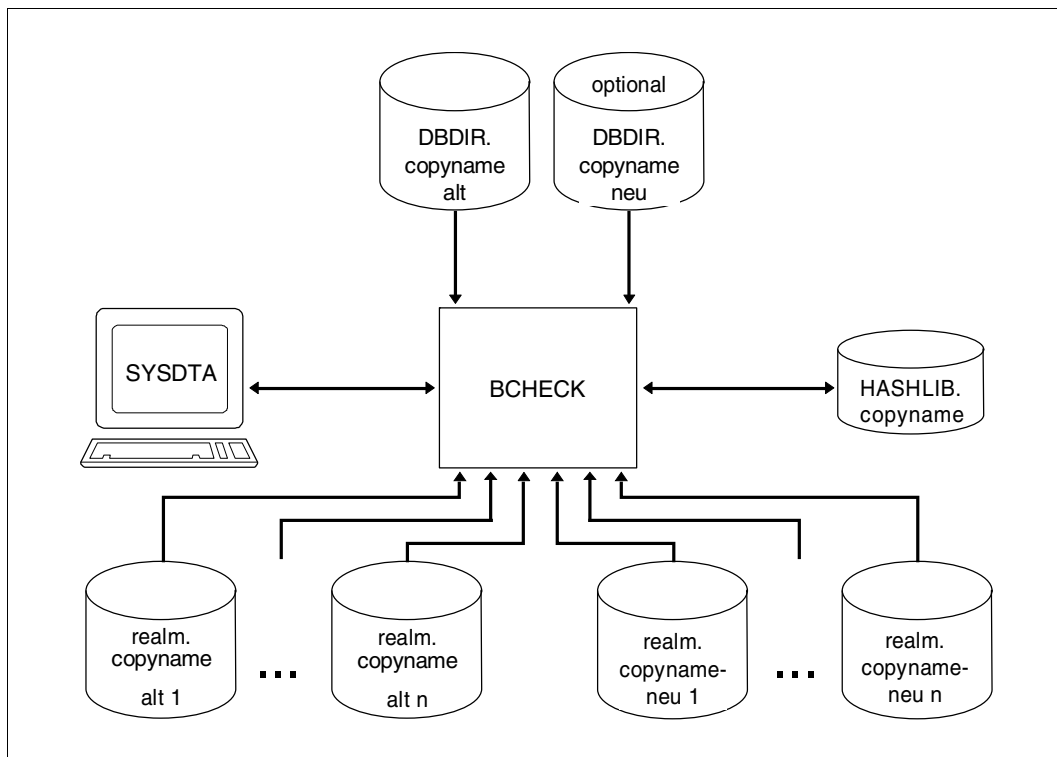


Bild 7: Systemumgebung bei einer Inkrementalprüfung Schattendatenbank ↔ Schattendatenbank

## Benötigte Arbeitsdateien

BCHECK benötigt für einen Prüflauf zwei Arbeitsdateien, die er automatisch unter der Benutzerkennung, unter der Sie BCHECK gestartet haben, auf gemeinschaftlicher Platte anlegt und nach normaler Beendigung des Laufs wieder löscht. Die Dateien haben standardmäßig die Dateikettungsnamen SCRTCH1 und SORTWK:

### SCRTCH1

benötigt BCHECK bei jedem Prüflauf zum Speichern eines Seitenverzeichnisses.

### SORTWK

benötigt der von BCHECK benutzte SORT bei einer Sortierungsprüfung und beim Prüfen von Indexwerten zum Sammeln und Sortieren der Prüfsätze. Siehe auch Handbuch „[SORT \(BS2000\)](#)“.

Wollen Sie die beiden Arbeitsdateien explizit einrichten, so müssen diese folgende Eigenschaften besitzen:

#### *Arbeitsdatei-1*

Dateikettungsname SCRTCH1

Zugriffsmethode=SAM, feste Satzlänge

Das Mengengerüst der zwischenspeichernden Daten ergibt sich aus der Formel:

$$\textit{seitenanzahl} \times 16 \text{ Bytes}$$

#### *seitenanzahl*

Anzahl relevanter Seiten aller zu prüfenden Realms, d.h. Summe der Realm-Größen in Datenbankseiten, abzüglich der Act-Key-0-Seiten, FPA-Seiten und leeren Seiten.

Die Primärzuweisung für die Arbeitsdatei-1 sollte sich am Mengengerüst der zwischenspeichernden Daten orientieren. Es sollte immer eine angemessene Sekundärzuweisung erfolgen für den Fall, dass der Speicherplatz erweitert werden muss.

*Arbeitsdatei-2*

Dateikettungsname SORTWK

Zugriffsmethode=PAM

Das Mengengerüst der zu sortierenden Daten ist bei Totalprüfung mit den beiden folgenden Formeln ermittelbar:

- Formel RSQ-Prüfung:

$26 \times \text{anzahl-prüfsätze}$  Bytes

- Formel Indexwertprüfung und Schlüsselwertprüfung:

$\text{schlüssellänge} \times \text{anzahl-prüfsätze}$  Bytes

Sie können das Mengengerüst bei einem SORTING-Lauf ermitteln, indem Sie zunächst CHECK SUMMING ausführen, die verdoppelte Summe der unter „DIAGNOSTIC SUMMARY OF BCHECK“ ausgewiesenen Prüfobjekte aus RECORD/TABLE-OCCURRENCES, CHAIN-SET-MEMBERSHIPS, REFERENCES BETWEEN TABLE-OCCURRENCES und REFERENCES FROM TABLES TO MEMBER-RECORDS bilden und diesen Wert als *anzahl-prüfsätze* in der Formel RSQ-Prüfung nutzen.

Bei einem SORTING-Lauf mit Schlüsselwertprüfung addieren Sie auf den so ermittelten Wert noch den Wert aus der Formel Schlüsselwertprüfung, wobei Sie als *anzahl-prüfsätze* den doppelten Wert von USERKEYS BETWEEN TABLES AND MEMBER-RECORDS einsetzen.

Bei einem SORTING-Lauf mit Indexwertprüfung addieren Sie auf den bisher ermittelten Wert noch den Wert aus der Formel Indexwertprüfung, wobei Sie als *anzahl-prüfsätze* den doppelten Wert von REFERENCES BETWEEN TABLE-OCCURRENCES einsetzen.

Als Schlüssellänge nutzen Sie jeweils die mittlere, mit der Satzanzahl gewichtete Schlüssellänge aller Schlüssel, wobei Sie bei der Indexwertprüfung für Schlüssel mit Duplikaten jeweils zusätzlich 6 Byte berücksichtigen müssen. Zur Vereinfachung können Sie mit der Nutzung der maximalen Schlüssellänge und evtl. zusätzlichen 6 Byte, falls Duplikate erlaubt sind, sicherstellen, dass es zu keinem Ressourcenengpass bei der Sortierung kommt.

Bei Inkrementalprüfungen bezieht sich das Mengengerüst auf die Änderungen, die gegenüber dem Vergleichsstand erfolgt sind.

Die Arbeitsdatei-2 wird vom SORT benötigt, wenn der Arbeitsspeicher, der durch die Anweisung SORTCORE beeinflusst werden kann, nicht ausreicht. Die Primärzuweisung sollte sich daher am Mengengerüst der zu sortierenden Daten orientieren. Es sollte immer eine angemessene Sekundärzuweisung erfolgen, für den Fall, dass der Speicherplatz erweitert werden muss.



### 3.3 Ergebnisse des Summierungslaufs für einen Sortierungslauf nutzen

#### Interne Ergebnisse der summarischen Prüfung nutzen

Bei Feststellung von globalen summarischen Konsistenzfehlern legt BCHECK eine Ausgabedatei an mit dem Namen:

`UTI.tsn.zeitstempel.BCHECK`

*tsn* Prozessfolgennummer, vierstellig

*zeitstempel*

Tag und Uhrzeit, an dem die Datei erstellt wird

Format: *tthhmmss*

Diese Datei enthält interne Ergebnisse des Summierungslaufs, die BCHECK als Information für einen nachfolgenden Sortierungslauf auswerten kann.

Damit BCHECK die Ausgabedatei des Summierungslaufs auswertet, müssen Sie vor dem Sortierungslauf der Ausgabedatei `UTI.tsn.zeitstempel.BCHECK` den Dateikettungsnamen BCHECK zuweisen und in der CHECK-Anweisung des Sortierungslaufs GENERATE SORTING angeben.

Sind keine brauchbaren Daten für einen SORTING-Lauf in der UTI-Datei, so wird die Datei bei aufgetretenen globalen Fehlern gelöscht.

#### Ausgabeprotokoll auswerten

In einem Summierungslauf schreibt BCHECK auf SYSLST:

- alle Meldungen und bei Feststellung von globalen summarischen Konsistenzfehlern
- die Kommandofolge für einen auf den Summierungslauf abgestimmten Sortierungslauf (mit Ausnahme der CREATE-FILE- und ADD-FILE-LINK-Kommandos für die beiden Arbeitsdateien SORTWK und SCRTCH1).

#### SYSLST-Datei aufbereiten

Wenn Sie vor einem Summierungslauf SYSLST einer Datei zuweisen, können Sie mit dem EDT die Ausgaben von BCHECK aufbereiten. Die aufbereitete Datei mit den generierten SORTING-Anweisungen können Sie für den Sortierungslauf nutzen.

BCHECK erkennt innerhalb des Summierungslaufs in welchen Satzarten, Sets oder Keys und zu welchen Konsistenzkriterien Fehler vorhanden sind. Die generierten SORTING-Anweisungen enthalten nur diese Objekte und die entsprechenden TYPE-Klauseln. Die Realm-Auswahl wird auf das notwendigste eingeschränkt.

### 3.4 Anweisungen für BCHECK

Anweisung	Bedeutung
SORTCORE	wahlweise; Größe des Sortierpuffers festlegen
CHECK	Prüfmodus wählen und Umfang der Prüfung festlegen
TYPE	wahlweise; Konsistenzkriterien spezifizieren
SCHEMA NAME	Schema bezeichnen
REALM NAME	zu prüfende Realms spezifizieren
RECORD NAME	wahlweise; zu prüfende Satzarten spezifizieren
SET NAME	wahlweise; zu prüfende Sets spezifizieren
KEY REF	wahlweise; zu prüfende SEARCH-Keys spezifizieren
END	Eingabe der Anweisungen beenden

Tabelle 8: Anweisungen für BCHECK

#### BCHECK-Anweisungen im Dialog eingeben

Die BCHECK-Anweisungen bestehen aus:

- den Steueranweisungen  
SORTCORE, CHECK, TYPE und SCHEMA
- den Objektauswahanweisungen  
REALM, RECORD, SET und KEY.

Sie können die Anweisungen innerhalb der beiden Gruppen in beliebiger Reihenfolge eingeben. Vor der ersten Anweisung zur Objektauswahl müssen Sie die Anweisung CHECK eingeben. Haben Sie ein Objekt eingegeben, können Sie keine weiteren Steueranweisungen mehr eingeben und gegebene Steueranweisungen auch nicht mehr korrigieren.

Anweisungen können Sie im Dialog korrigieren und wieder eingeben. Beachten Sie aber, dass Syntaxfehler in den den Objektauswahanweisungen folgenden Schlüsselwörtern gegebenenfalls zur Generierung von Standardwerten führen, die nicht mehr korrigiert werden können.

## Größe des Sortierpuffers festlegen (SORTCORE)

---

[SORTCORE IS *n*]

---

*n* Anzahl der Hauptspeicherseiten für den Sortierpuffer in Einheiten von 4-Kbyte; der Wertebereich von *n* ist durch den SORT vorgegeben.

Standardwert: 150

Bei der Sortierungsprüfung verwendet BCHECK zum Sortieren der Prüfsätze das BS2000-Dienstprogramm SORT (siehe Handbuch "[SORT \(BS2000\)](#)").

Mit der SORTCORE-Anweisung bestimmen Sie die Größe des Hauptspeicherplatzes für den Sortierpuffer des Dienstprogramms SORT (siehe Handbuch "[SORT \(BS2000\)](#)"). Sie können sie wahlweise bei der Sortierungsprüfung angeben.

Das Mengengerüst der zu sortierenden Daten ist das gleiche, das der Größe der Arbeitsdatei-2 zugrunde liegt (siehe "[Arbeitsdatei-2](#)" auf Seite 80).

## Prüfmodus wählen und Umfang der Prüfung festlegen (CHECK)

---

```
CHECK { [[GENERATE ]SORTING] } [[ AGAINST COPY NAME IS copyname ]  
      [SUMMING]
```

---

### GENERATE SORTING

Sortierungsprüfung mit Auswerten der internen Ergebnisse eines vorangegangenen Summierungslaufs

### SORTING

Sortierungsprüfung

### SUMMING

summarische Prüfung

### AGAINST COPY

Inkrementalprüfung

### *copyname*

Kopienname der Schattendatenbank

### Standardwert:

SUMMING

Mit der Anweisung CHECK legen Sie fest, ob BCHECK eine Sortierungsprüfung oder eine summarische Prüfung und eine Totalprüfung oder eine Inkrementalprüfung durchführen soll.

Sie müssen die CHECK-Anweisung innerhalb der Steueranweisungen angeben.



Hat eine vorausgegangene Prüfung lokale Konsistenzfehler entdeckt, so sollten diese vor einer Sortierungsprüfung behoben werden, sonst werden eventuell sehr viele globale Konsistenzfehler gemeldet, die lediglich auf die lokalen Inkonsistenzen zurückzuführen sind.

## Kriterien für die globale Konsistenzprüfung wählen (TYPE)

---


$$\text{[TYPE IS } \left. \begin{array}{l} \text{ALL [EXCEPT } type\text{-nr-1[}, type\text{-nr-2]...]} \\ 0 \\ type\text{-nr-1[}, type\text{-nr-2]...} \end{array} \right\}]$$


---

**ALL** BCHECK prüft nach allen Konsistenzkriterien.

*type-nr*

Nummer des Konsistenzkriteriums *type-nr*=1...11 (siehe [Tabelle 9](#))

**ALL EXCEPT** *type-nr-1[,type-nr-2]...*

BCHECK prüft nach allen Konsistenzkriterien außer den nach EXCEPT aufgelisteten.

**0**

BCHECK prüft nur lokal, d.h. innerhalb einer Seite. Seitenübergreifende Beziehungen der angegebenen Prüfobjekte werden nicht geprüft.

*type-nr-1[,type-nr-2]...*

BCHECK prüft nur nach den angegebenen Konsistenzkriterien.

Standardwert:

ALL

BCHECK kennt folgende Konsistenzkriterien:

<i>type-nr</i>	Konsistenzkriterium
1	korrekter Verweis vom Act-Key in einer DBTT-Spalte auf den Satz bzw. auf die höchste bzw. erste Tabellenseite
2	korrekte Verkettung der Sätze in einem Set MODE IS CHAIN einschließlich des Ownersatzes
3	korrekte Verkettung der Sätze in einem Set MODE IS CHAIN LINKED TO PRIOR einschließlich des Ownersatzes
4	korrekte Verkettung zwischen der ersten Tabellenseite und der letzten Tabellenseite auf Stufe 0
5	korrekte Verkettung der Tabellenseiten zwischen den Stufen
6	korrekte Verkettung der Tabellenseiten zwischen den Stufen bezüglich ihrer Indexwerte
7	korrekte Verkettung der Tabellenseiten innerhalb einer Stufe
8	korrekter Verweis von den Tabelleneinträgen der Stufe 0 auf die Sätze und Übereinstimmung der Schlüsselwerte in Tabelle und Satz

Tabelle 9: Konsistenzkriterien von BCHECK

(Teil 1 von 2)

type-nr	Konsistenzkriterium
9	korrekter Verweis von den indirekten CALC-Tabelleneinträgen auf die Sätze und Übereinstimmung der Schlüsselwerte in Tabelle und Satz
10	korrekte Verkettung einer CALC-Tabellen-Überlaufkette
11	korrekte Verkettung einer DUPLIKAT-Tabellen-Überlaufkette

Tabelle 9: Konsistenzkriterien von BCHECK

(Teil 2 von 2)

Die folgende Tabelle zeigt, welche Konsistenzkriterien bei den jeweiligen Prüfobjekten geprüft werden.

BCHECK - Anweisung		Typ	Konsistenzkriterium														
			0 1	0 2	0 3	0 4	0 5	0 6	0 7	0 8	0 9	1 0	1 1				
RECORD	WITH LOCATION CHECK oder WITH KEYVALUE CHECK	nicht CALC	X														
		direct CALC	X												X		
		indirect CALC	X											X	X		
	WITHOUT LOCATION ..	alle															
SET	WITH INDEX CHECK	Tabelle	X			X	X	X	X	X							
	WITHOUT INDEX CHECK	Tabelle	X			X	X		X	X							
		CHAIN		X													
		CHAIN PRIOR			X												
		SYSTEM-Set CHAIN	X	X													
		SYSTEM-Set CH-PR	X		X												
KEY	WITH INDEX CHECK	Tabelle	X			X	X	X	X	X							
		Duplikattabelle	X			X	X	X	X	X						X	
	WITHOUT INDEX CHECK	Tabelle	X			X	X		X	X							
		Duplikattabelle	X			X	X		X	X							X
		indirect CALC												X	X		

Tabelle 10: Zuordnung Prüfobjekt - Konsistenzkriterium



Es wird empfohlen, immer „ALL“ anzugeben. Nur so ist gewährleistet, dass die Prüfung vollständig ist.

Nicht vorhandene DB-Strukturen ignoriert BCHECK auch dann, wenn die entsprechenden Konsistenzkriterien angegeben sind, einen Performancegewinn durch Weglassen der Kriterien gibt es nicht.

## Schema bezeichnen (SCHEMA NAME)

---

SCHEMA NAME IS *schemaname*

---

*schemaname*

Name des Schemas, dessen SIA BCHECK zum Prüfen der Datenbank verwenden soll; für *schemaname* können Sie Folgendes angeben:

- *benutzerschemaname* zum Prüfen von Benutzerrealms
- COMPILER-SCHEMA zum Prüfen des DBCOM
- PRIVACY-AND-IQF-SCHEMA zum Prüfen des DBDIR

Die SCHEMA-Anweisung nennt die SIA, aus der BCHECK die Schema-Information zum Prüfen der Datenbank holen soll. Wird die SCHEMA-Anweisung nicht angegeben, so wird die Benutzerdatenbank geprüft.



Beim Prüfen des DBDIR prüft BCHECK die Satzart SSIA-RECORD wegen ihrer besonderen Struktur (spanned records) nicht.

## Zu prüfende Realms spezifizieren (REALM NAME)

---

```
REALM NAME IS { [ALL [ EXCEPT realmname-1, ... ] ]
                realmname-2, ... }
```

---

### *realmname*

Name eines Realm der Datenbank; geben Sie an

- für Benutzerrealms:  
den in der Schema-DDL per AREA-Klausel definierten Realm-Namen
- für das Database Directory: DATABASE-DIRECTORY
- für den Database Compiler Realm: DATABASE-COMPILER-REALM

**ALL** BCHECK prüft alle nicht-temporären Benutzerrealms, die im DBDIR als EXISTENT und SWITCHED-ON gekennzeichnet sind.

Die Realms werden in aufsteigender Folge der Realm-Nummern geprüft.

**ALL EXCEPT** *realmname-1,...*

Bedeutung wie bei ALL; BCHECK schließt aber die aufgeführten Realms von der Prüfung aus

*realmname-2,...*

Namen der Realms - einzeln aufgeführt -, die BCHECK prüfen soll

Werden mehrere REALM-Anweisungen zwischen den Steueranweisungen und der Anweisung END angegeben, so prüft BCHECK in der angegebenen Reihenfolge. Es muss mindestens ein Realm angegeben werden.

Alle HASH-Routinen, die zu CALC-Seiten in den angegebenen Realms gehören, müssen im Modul UDSHASH eingetragen und in der HASHLIB vorhanden sein, sonst wird eine weitere Prüfung abgelehnt.



## Zu prüfende Satzarten spezifizieren (RECORD NAME)

---

```
[RECORD NAME IS { ALL [ EXCEPT satzname-1, ... ] } [ { WITH } { LOCATION CHECK } ] ]
{ satzname-2, ... } [ { WITHOUT } { KEYVALUE CHECK } ] ]
```

---

### *satzname*

Name einer Satzart des angegebenen Schemas.

*satzname* muss in dem angegebenen Realm bzw. in den angegebenen Realms liegen.

**ALL** BCHECK prüft alle Satzarten des genannten Schemas, die in den zu prüfenden Realms enthalten sind

**ALL EXCEPT *satzname-1*,...**

Bedeutung wie bei ALL; BCHECK schließt aber die aufgeführten Satzarten von der Prüfung aus

*satzname-2*,...

Namen der Satzarten - einzeln aufgeführt -, die BCHECK prüfen soll

**WITH LOCATION CHECK**

BCHECK prüft zusätzlich alle Verwaltungsstrukturen, die mit dem Abspeichern einer Satzart zusammenhängen, d.h. in der DBTT die Verweise auf die Sätze. Es findet keine Schlüsselwertprüfung statt.

Bei Satzarten, die mit LOCATION MODE IS CALC definiert sind, prüft BCHECK Folgendes:

- das Ergebnis des Hashverfahrens (nur für Primärseiten und erste Überlaufseiten),
- die Verkettung der Überlaufseiten,
- die interne Struktur der CALC-Tabellen und
- die Verweise der CALC-Tabelleneinträge auf die Sätze

**WITH KEYVALUE CHECK**

BCHECK prüft zusätzlich alle Schlüsselfelder eines Satzes, die in irgend einem Zugriffspfad vorkommen. WITH KEYVALUE CHECK schließt die Funktionalität von WITH LOCATION CHECK mit ein.

WITH KEYVALUE CHECK ist ab UDS/SQL V2.6 der Standardwert.

**WITHOUT LOCATION CHECK**

BCHECK prüft lediglich lokale Informationen der Satzarten, d.h. er prüft, ob die Satzarten in der DBTT plausibel sind und ob die Satzlängen mit den Angaben in der SIA übereinstimmen, erzeugt aber keine Prüfsätze.

**WITHOUT KEYVALUE CHECK**

Diese Klausel ist nicht erlaubt.

Mit der RECORD-Anweisung nennen Sie BCHECK die zu prüfenden Satzarten. Sie können sie an beliebiger Stelle, beliebig oft, nach den Steueranweisungen angeben. Doppelte Angaben ignoriert BCHECK.

### **Hinweise zu CALC-Keys:**

Existieren in einem Realm zwei oder mehr CALC-Keys (indirekte CALC-Keys oder CALC-Search-Keys) mit gleicher Schlüssellänge, so kann BCHECK diese nicht global prüfen, falls Sie nicht alle CALC-Keys in diesem Realm mit dieser Schlüssellänge angegeben haben. Wählen Sie einen solchen CALC-Key zur Prüfung aus, so lehnt BCHECK dessen globale Prüfung ab, führt aber für den betreffenden Hashbereich eine lokale Prüfung durch. Da er zu dieser lokalen Prüfung alle CALC-Seiten heranzieht, die CALC-Keys von der Länge des zu prüfenden CALC-Keys enthalten, kann es vorkommen, dass BCHECK lokale Konsistenzfehler in CALC-Seiten meldet, die zu keinem von Ihnen angegebenen CALC-Key gehören.

Die Schlüsselwertprüfung indirekter CALC-Tabellen für Konsistenzkriterium 9 kann im Allgemeinen ebenfalls nicht satzartgenau erfolgen. Um zu vermeiden, dass in einer konsistenten Datenbank - insbesondere im Summing-Modus - vermeintliche Inkonsistenzen ausgewiesen werden, wird empfohlen, immer alle Satzarten in die Prüfung einzubeziehen.

In der Sortierungsprüfung können im Allgemeinen Konsistenzfehler bei CALC-Keys nicht lokalisiert werden. BCHECK kann nur den betroffenen Realm, RSQ und Schlüssellänge feststellen.

Das Auftreten von nicht erlaubten Duplikaten eines CALC-Keys in verschiedenen Seiten einer Hash-Überlaufkette kann BCHECK nicht feststellen.

### **Hinweise zur Schlüsselwertprüfung:**

Fehlende SET- bzw. KEY-Klauseln, in denen eine zu prüfende Satzart Member ist, führen zum Unterlassen der Prüfung für den betreffenden Zugriffspfad, insbesondere auch zum Unterlassen einer gewünschten Schlüsselwertprüfung.

Um Schlüsselfelder in einem Satz zu positionieren, benötigt BCHECK ein Subschema, das die betroffene Satzart bzw. die betroffene Membersatzart samt Set vollständig enthält. Sollen Schlüsselwerte für eine Satzart geprüft werden, die in keinem Subschema vorkommen, so wird eine Warnung ausgegeben und die Prüfung in den betroffenen Satzarten wird dann ohne Schlüsselwertprüfung durchgeführt.

Die Metadaten zur Schlüsselwertprüfung werden immer für alle Satzarten erzeugt. Das hat zur Folge, dass die Warnung möglicherweise auch dann ausgegeben wird, wenn eine der Satzarten nicht geprüft werden soll.

**Hinweise zu verteilbaren Listen:**

Bei der Prüfung eines indirekten CALC-Keys oder eines CALC-Search-Keys, der zu einer verteilbaren Liste gehört, müssen immer alle Realms angegeben werden, auf denen die Sätze der zugehörigen Membersatzart liegen können.

## Zu prüfende Sets spezifizieren (SET NAME)

---

```
[SET NAME IS { ALL [ EXCEPT setname-1, ... ] } [ { WITH } ] [ { WITHOUT } ] INDEX CHECK]]
```

---

### *setname*

Name eines Sets des angegebenen Schemas.

*setname* muss in dem angegebenen Realm bzw. in den angegebenen Realms liegen.

**ALL** BCHECK prüft alle Sets des genannten Schemas und deren Ownersatzart bzw. Membrosatzart bzw. Tabellen, die in den zu prüfenden Realms enthalten sind

**ALL EXCEPT *setname-1*,...**

Bedeutung wie bei ALL;

BCHECK schließt aber die aufgeführten Sets von der Prüfung aus

*setname-2*,...

Namen der Sets - einzeln aufgeführt -, die BCHECK prüfen soll

**WITH INDEX CHECK**

Indexwerte (Schlüsselwert und Satzfolgennummer bzw. Schlüsselwert oder Satzfolgennummer, abhängig von der Art der Tabelle) prüfen.

WITH INDEX CHECK bezieht sich nur auf die Konsistenzkriterien 5, 6, 7 und 11, und nicht auf die Schlüsselwertprüfung der Konsistenzkriterien 8 und 9.

Tabellenprüfungen:

BCHECK prüft in Sort-Key-Tabellen, mehrstufigen Adresslisten und mehrstufigen Listen die Indexwerte auf Korrektheit der Ungleichheitsbeziehungen, die innerhalb der Ketten von Tabellenseiten gleicher Stufe und zwischen den einzelnen Tabellenseiten bestehen müssen. Das bedeutet:

- BCHECK prüft bei Ketten von Tabellenseiten gleicher Stufe, ob bei der Vorwärtsverkettung der niedrigste Indexwert der Nachfolgesseite größer ist als der höchste Indexwert der betrachteten Seite und umgekehrt bei der Rückwärtsverkettung (bei mehrstufigen Listen erst ab Stufe > 0)
- BCHECK prüft bei Tabellenseiten verschiedener Stufen, ob der Indexwert eines Tabelleneintrags einer Stufe > 0 stets größer bzw. gleich ist als der größte Indexwert in der Tabelle nächstniedriger Stufe, auf die er verweist, und kleiner ist als der kleinste Indexwert der darauffolgenden Tabelle (bei mehrstufigen Listen erst ab Stufe > 1)

**WITHOUT INDEX CHECK**

BCHECK prüft keine Indexwerte

Mit der SET-Anweisung nennen Sie BCHECK die zu prüfenden Sets. BCHECK prüft bei Sets:

- die Owner-Member-Beziehung über die Set-Connection-Data (SCD),
- die Verkettung der Membersätze bei MODE IS CHAIN,
- bei einstufigen und mehrstufigen Listen, Sort-Key-Tabellen und einstufigen und mehrstufigen Adresslisten:
  - den Tabellenkopf,
  - die Adressverkettung der Tabellenseiten gleicher Stufe,
  - bei mehrstufigen Tabellen die Adressverweise zwischen den Tabellenseiten verschiedener Stufen und
  - die Verweise der Tabelleneinträge der untersten Stufe auf die Sätze.

Bei jedem Set, den BCHECK prüft, prüft er automatisch auch die zugehörige Ownersatzart und Membersatzart ohne LOCATION CHECK. Sie müssen diese Satzarten daher nur dann in der RECORD-Anweisung nennen, wenn Sie die Prüftiefe auf WITH LOCATION CHECK ausdehnen wollen.

Die SET-Anweisung können Sie an beliebiger Stelle, beliebig oft, nach den Steueranweisungen angeben. Doppelte Angaben ignoriert BCHECK.



Implizite Sets, d.h. SEARCH-Keys auf Satzartebene, und zu einem Set gehörige SEARCH-Keys prüft BCHECK nur, wenn Sie dies mit der KEY-Anweisung verlangen.

Zum Prüfen der Indexwerte (WITH INDEX CHECK) sortiert BCHECK die Prüfsätze auch bei einer summarischen Prüfung.

Indexwerte prüft BCHECK nur bei einer Totalprüfung.

Das Prüfen der Indexwerte beinhaltet nicht das Prüfen der Sortierfolge in Ketten und das Prüfen von einstufigen Listen.

Bei der Prüfung einer verteilbaren Liste müssen immer alle Realms angegeben werden, auf denen die Sätze der zugehörigen Membersatzart liegen können.

## Zu prüfende SEARCH-Keys spezifizieren (KEY REF)

---

```
[KEY REF IS { ALL [ EXCEPT keyref-1, ... ] } [ { WITH } ] [ { WITHOUT } ] INDEX CHECK ]
```

---

*keyref* Schlüsselnummer eines SEARCH-Keys. Sie können sie dem SIA PRINT REPORT entnehmen (siehe [Seite 142](#)).

ALL BCHECK prüft von dem genannten Schema alle SEARCH-Keys und die jeweilige Satzart, auf der der SEARCH-Key definiert ist, soweit sie in den zu prüfenden Realms enthalten sind

ALL EXCEPT *keyref-1,...*

Bedeutung wie bei ALL.

BCHECK schließt aber die aufgeführten SEARCH-Keys von der Prüfung aus

*keyref-2,..*

Schlüsselnummern der SEARCH-Keys - einzeln aufgeführt -, die BCHECK prüfen soll

WITH INDEX CHECK

Indexwerte prüfen bei mehrstufigen SEARCH-Key-Tabellen (siehe [Seite 92](#))

WITH INDEX CHECK bezieht sich nur auf die Konsistenzkriterien 5, 6, 7 und 11, und nicht auf die Schlüsselwertprüfung der Konsistenzkriterien 8 und 9.

WITHOUT INDEX CHECK

BCHECK prüft keine Indexwerte

Mit der KEY-Anweisung nennen Sie BCHECK die zu prüfenden SEARCH-Keys. Die SEARCH-Keys dürfen auf Satzartebene oder Setebene als INDEX-SEARCH-Key oder CALC-SEARCH-Key definiert sein. BCHECK prüft auch Duplikat-Tabellen. Die Satzart, auf der ein SEARCH-Key definiert ist, prüft BCHECK automatisch mit, allerdings ohne LOCATION CHECK.

Zum Prüfen von SEARCH-Keys führt BCHECK dieselben Prüfungen durch wie

- für CALC-SEARCH-Keys:  
beim LOCATION CHECK von Satzarten, die mit LOCATION MODE IS CALC definiert sind (siehe [Seite 89](#))
- für INDEX-SEARCH-Keys:  
bei den mehrstufigen Tabellen eines Sets (siehe [Seite 92](#)).

Bei Duplikat-Tabellen prüft BCHECK zusätzlich

- im Tabellenkopf den Verweis auf den Duplikat-Kopf auf Plausibilität,
- den Duplikat-Kopf,
- die Schlüsselwerte in den Tabellenindex-Einträgen einer Tabellenseite auf aufsteigende Reihenfolge,
- die Verweise auf die Tabellenzeilen (DB-Key-Listen) aus den Tabellenindex-Einträgen auf Plausibilität,
- die Verkettung der Überlaufseiten einer Duplikat-Tabelle und
- die Satzfolgennummern einer Tabellenzeile auf aufsteigende Reihenfolge.

Sie können die KEY-Anweisung angeben an beliebiger Stelle, beliebig oft, nach den Steueranweisungen. Doppelte Angaben ignoriert BCHECK.



Sie können ASC-Keys bzw. DESC-Keys in der KEY-Anweisung nicht angeben. Sie werden automatisch mit geprüft, wenn der zugehörige Set in der SET-Anweisung angegeben ist.

Geben Sie in der KEY-Anweisung die Nummer eines ASC-Keys bzw. DESC-Keys an, so weist BCHECK die Prüfung dieses Schlüssels zurück.

Für CALC-SEARCH-Keys mit gleicher Schlüssellänge gilt dieselbe Einschränkung wie für indirekte CALC-Keys (siehe [Seite 89](#)).

Das Auftreten von nicht erlaubten Duplikaten eines CALC-SEARCH-Keys in verschiedenen Seiten einer Hash-Überlaufkette kann BCHECK nicht feststellen.

BCHECK sortiert zum Prüfen der Indexwerte (WITH INDEX CHECK) die Prüfsätze auch bei einer summarischen Prüfung.

BCHECK prüft Indexwerte nur bei einer Totalprüfung.

Bei der Prüfung von Search-Keys oder CALC-Search-Keys, die zu einer verteilbaren Liste gehören, müssen immer alle Realms angegeben werden, in denen die Sätze der zugehörigen Membersatzart liegen können.

### 3.5 Kommandofolge zum Starten von BCHECK

Die hier beschriebene Kommandofolge geht davon aus, dass UDS/SQL mit IMON installiert wurde (siehe Abschnitt „START-Kommandos der UDS/SQL-Programme“ in Kapitel 2 des Handbuchs „[Aufbauen und Umstrukturieren](#)“).

Abhängig vom Prüflauf benötigen Sie folgende Kommandos zum Starten von BCHECK:

```

01  [/CREATE-FILE FILE-NAME=arbeitsdatei-1 ...
      /ADD-FILE-LINK LINK-NAME=SCRATCH1,FILE-NAME=arbeitsdatei-1,
      ACCESS-METHOD=*SAM ]
02  [/CREATE-FILE FILE-NAME=arbeitsdatei-2 ...
      /ADD-FILE-LINK LINK-NAME=SORTWK,FILE-NAME=arbeitsdatei-2,
      ACCESS-METHOD=*UPAM ]
03  /ADD-FILE-LINK LINK-NAME=DATABASE,
      FILE-NAME=[:catid:][$userid.]dbname.DBDIR[.copyname]
04  [/ADD-FILE-LINK LINK-NAME=BCHECK,
      FILE-NAME=[:catid:]UTI.tsn.zeitstempel.BCHECK ]
05  [/ASSIGN-SYSLST TO=dateiname]
06  /SELECT-PRODUCT-VERSION PRODUCT-NAME=UDS-SQL,VERSION=version,SCOPE=*TASK
07  /START-UDS-BCHECK
08  bcheck-anweisungen
09  END
10  [/ASSIGN-SYSLST TO=*PRIMARY]

```

- 03 Hier ist die Angabe von *:catid:* zulässig (siehe Handbuch „[Datenbankbetrieb](#)“).
- 04 Geben Sie bei einem Sortierungslauf an, wenn BCHECK die Ausgabedatei des Summierungslaufs *UTI.tsn.zeitstempel*.BCHECK auswerten soll.
- 05 Geben Sie bei einem Summierungslauf an, wenn Sie die Protokollierung für einen nachfolgenden Sortierungslauf nutzen wollen.
- 06 Die Version des Dienstprogramms wird ausgewählt.  
Die Angabe der Version wird generell empfohlen, da mehrere UDS/SQL-Versionen parallel installiert sein können.
- 07 Das UDS/SQL-Dienstprogramm kann auch mit dem Aliasnamen BCHECK gestartet werden.



### 3.6 Beispiele zu BCHECK

Die Datenbank KUNDEN (siehe [Seite 29](#)) wurde umstrukturiert und sieht danach folgendermaßen aus:

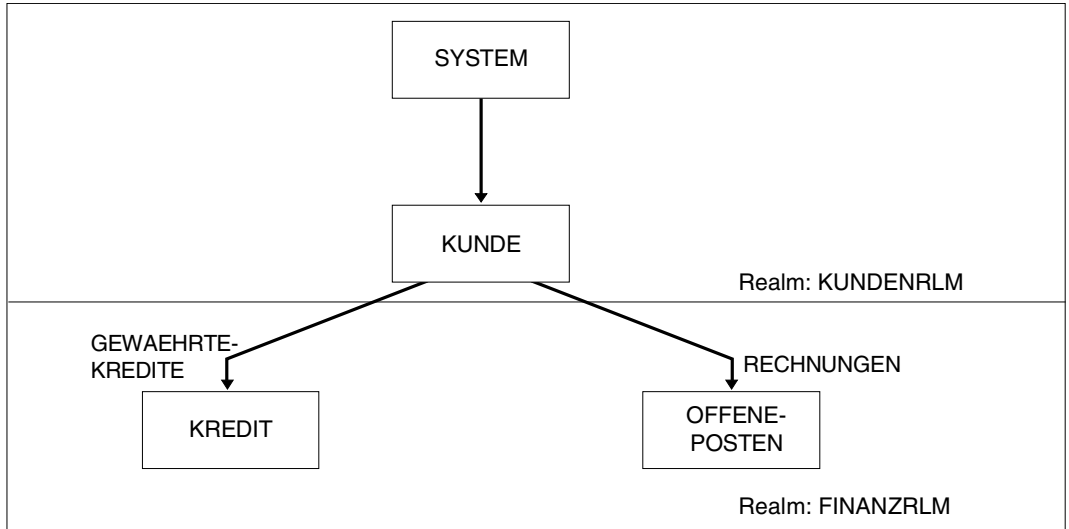


Bild 8: Datenbank KUNDEN nach dem Umstrukturieren

Nachdem einige Zeit mit der umstrukturierten Datenbank KUNDEN gearbeitet wurde, soll sie mit BCHECK auf Konsistenz geprüft werden. Folgende Stände existieren:

- Schattendatenbank unmittelbar nach dem Umstrukturieren: *copyname.NACHUM*
- Originaldatenbank (neuerer Stand als NACHUM)

*Beispiel 1*

Die Schattendatenbank wird nach dem Umstrukturieren (NACHUM) komplett summarisch geprüft.

```
/ADD-FILE-LINK LINK-NAME=DATABASE,FILE-NAME=KUNDEN.DBDIR.NACHUM
/SELECT-PRODUCT-VERSION PRODUCT-NAME=UDS-SQL, VERSION=02.8A00
/START-UDS-BCHECK
***** START          BCHECK          (UDS/SQL V2.8 0000 )    2015-06-28  11:25:59
CHECK SUMMING
SCHEMA KUNDENKARTEI
REALM ALL
RECORD ALL
SET ALL WITH INDEX CHECK
KEY ALL WITH INDEX CHECK
END
***** FOR INDEX CHECKS A SORTING MUST BE PERFORMED EVEN FOR THE SUMMING
PROCEDURE

***** THE INPUT IS CORRECT, THE CHECK RUN IS STARTING NOW
***** ANALYSING NEW STATE OF REALM KUNDENRLM FROM FILE
      :SQL2:$XXXXXXXX.KUNDEN.KUNDENRLM.NACHUM
          100 NON EMPTY PAGES HAVE BEEN ANALYSED
***** ANALYSING NEW STATE OF REALM FINANZRLM FROM FILE
      :SQL2:$XXXXXXXX.KUNDEN.FINANZRLM.NACHUM
          6 NON EMPTY PAGES HAVE BEEN ANALYSED
***** SORTING   STARTED          DATE AND TIME  2015-06-28  11:25:59
***** SORTING COMPLETED        DATE AND TIME  2015-06-28  11:25:59

***** DIAGNOSTIC SUMMARY OF BCHECK

          NO WARNINGS
          NO ERRORS
          NO SYSTEM-ERRORS

11379 LOCAL CHECKS HAVE BEEN DONE
      126 RECORD/TABLE-OCCURRENCES HAVE BEEN CHECKED AGAINST DBTT
      68 CHAIN-SET-MEMBERSHIPS HAVE BEEN CHECKED
      52 REFERENCES BETWEEN TABLE-OCCURRENCES HAVE BEEN CHECKED
      72 REFERENCES FROM TABLES TO MEMBER-RECORDS HAVE BEEN CHECKED
      72 USERKEYS BETWEEN TABLES AND MEMBER-RECORDS HAVE BEEN CHECKED

          NO EASY (MINOR) LOCAL CONSISTENCY ERRORS
          NO FATAL (SERIOUS, STRUCTURAL) LOCAL CONSISTENCY ERRORS

          NO GLOBAL CONSISTENCY ERRORS
          NO GLOBAL INDEX CHECK HAS BEEN DONE
```

```

***** END OF DIAGNOSTIC SUMMARY
***** NR OF DATABASE ACCESSES :           26
***** NORMAL END   BCHECK      (UDS/SQL V2.8 0000 )   2015-06-28  11:25:59

```

### Beispiel 2

Die Originaldatenbank wird gegen die laut BCHECK-Prüflauf fehlerfreie Schattendatenbank nach der Umstrukturierung (NACHUM) geprüft. Bei dieser summarischen Inkrementalprüfung prüft BCHECK beide Datenbanken auch auf Kohärenz.

```

/ADD-FILE-LINK LINK-NAME=DATABASE,FILE-NAME=KUNDEN.DBDIR
/SELECT-PRODUCT-VERSION PRODUCT-NAME=UDS-SQL,VERSION=02.8A00
/START-UDS-BCHECK
***** START      BCHECK      (UDS/SQL V2.8 0000 )   2015-06-28  11:36:10
CHECK SUMMING AGAINST COPY NAME NACHUM
SCHEMA KUNDENKARTEI
REALM ALL
RECORD ALL
SET ALL
KEY ALL
END

***** THE INPUT IS CORRECT, THE CHECK RUN IS STARTING NOW
***** ANALYSING NEW STATE OF REALM KUNDENRLM FROM FILE
      :SQL2:$XXXXXXXX.KUNDEN.KUNDENRLM
      AGAINST OLD STATE FROM FILE :SQL2:$XXXXXXXX.KUNDEN.KUNDENRLM.NACHUM
      50 CHANGED PAGES HAVE BEEN ANALYSED
***** ANALYSING NEW STATE OF REALM FINANZRLM FROM FILE
      :SQL2:$XXXXXXXX.KUNDEN.FINANZRLM
      AGAINST OLD STATE FROM FILE :SQL2:$XXXXXXXX.KUNDEN.FINANZRLM.NACHUM
      2 CHANGED PAGES HAVE BEEN ANALYSED

+++++ GLOBAL CONSISTENCY ERRORS IN TYPE-NR :           1

+++++ GLOBAL CONSISTENCY ERRORS IN REF-NR  :           2

***** DIAGNOSTIC SUMMARY OF BCHECK

      NO WARNINGS
      NO ERRORS
      NO SYSTEM-ERRORS

14474 LOCAL CHECKS HAVE BEEN DONE
      246 RECORD/TABLE-OCCURRENCES HAVE BEEN CHECKED AGAINST DBTT
      136 CHAIN-SET-MEMBERSHIPS HAVE BEEN CHECKED
      102 REFERENCES BETWEEN TABLE-OCCURRENCES HAVE BEEN CHECKED
      144 REFERENCES FROM TABLES TO MEMBER-RECORDS HAVE BEEN CHECKED
      144 USERKEYS BETWEEN TABLES AND MEMBER-RECORDS HAVE BEEN CHECKED

```

```

NO EASY (MINOR) LOCAL CONSISTENCY ERRORS
NO FATAL (SERIOUS, STRUCTURAL) LOCAL CONSISTENCY ERRORS

+++++          GLOBAL CONSISTENCY ERRORS. DO A SORTING CHECK
                NO GLOBAL INDEX CHECK HAS BEEN DONE

***** END OF DIAGNOSTIC SUMMARY
***** NR OF DATABASE ACCESSES :          48
***** NORMAL END   BCHECK      (UDS/SQL V2.8 0000 )    2015-06-28  11:36:10

```

### Ausschnitte aus dem SYSLST-Protokoll:

```

.
.
/REMARK  START GENERATED BCHECK-STATEMENTS
/ADD-FILE-LINK LINK-NAME=BCHECK,          -
/  FILE-NAME=$XXXXXXXX.UTI.1DP5.30125022.BCHECK
/ADD-FILE-LINK LINK-NAME=DATABASE,       -
/  FILE-NAME=$XXXXXXXX.KUNDEN.DBDIR
/SELECT-PRODUCT-VERSION PRODUCT-NAME=UDS-SQL,VERSION=XX.XXXX
/SET-JV-LINK LINK-NAME=UDSPS01,JV-NAME=UDS-PUB-DECL
/START-UDS-BCHECK
CHECK GENERATE SORTING AGAINST COPY NACHUM
TYPE      IS 1
SCHEMA NAME IS KUNDENKARTEI
REALM  NAME IS KUNDENRLM
REALM  NAME IS FINANZRLM
RECORD NAME IS KUNDE                      WITH  LOCATION CHECK
SET    NAME IS GEWAehrTE-KREDITE          WITHOUT INDEX CHECK
KEY    REF IS 1                          WITHOUT INDEX CHECK
END
.
.
.

```

Nachdem in der summarischen Prüfung Fehler festgestellt wurden, werden diese mit einer Sortierprüfung lokalisiert. Die Anweisungsfolge für eine eingeschränkte Sortierprüfung ist in der SYSLST-Ausgabe der summarischen Prüfung nach REMARK... aufgeführt.

*Beispiel 3*

Es wird eine eingeschränkte Sortierprüfung durchgeführt, um die Fehler genauer zu lokalisieren. Dazu werden die Anweisungen verwendet, die im Protokoll der summarischen Prüfung nach REMARK... aufgeführt sind.

```

/ADD-FILE-LINK BCHECK, $XXXXXXXX.UTI.1DP5.30125022.BCHECK
/ADD-FILE-LINK LINK-NAME=DATABASE, FILE-NAME=KUNDEN.DBDIR
/SELECT-PRODUCT-VERSION PRODUCT-NAME=UDS-SQL, VERSION=02.8A00
/SET-JV-LINK LINK-NAME=UDSPS01, JV-NAME=UDS-PUB-DECL
/START-UDS-BCHECK
***** START          BCHECK          (UDS/SQL V2.8 0000 )    2015-06-28   11:36:10
CHECK GENERATE SORTING AGAINST COPY NACHUM
TYPE          IS 1
SCHEMA NAME IS KUNDENKARTEI
REALM NAME IS KUNDENRLM
REALM NAME IS FINANZRLM
RECORD NAME IS KUNDE                               WITH LOCATION CHECK
SET NAME IS GENAEHRTE-KREDITE                       WITHOUT INDEX CHECK
KEY REF IS 1                                         WITHOUT INDEX CHECK
END

```

```

***** THE INPUT IS CORRECT, THE CHECK RUN IS STARTING NOW.
***** ANALYSING NEW STATE OF REALM KUNDENRLM FROM FILE
      :SQL2:$XXXXXXXX.KUNDEN.KUNDENRLM
      AGAINST OLD STATE FROM FILE :SQL2:$XXXXXXXX.KUNDEN.KUNDENRLM.NACHUM
      50 CHANGED BLOCKS HAVE BEEN ANALYSED.
***** ANALYSING NEW STATE OF REALM FINANZRLM FROM FILE
      :SQL2:$XXXXXXXX.KUNDEN.FINANZRLM
      AGAINST OLD STATE FROM FILE :SQL2:$XXXXXXXX.KUNDEN.FINANZRLM.NACHUM
      2 CHANGED BLOCKS HAVE BEEN ANALYSED.
.
.
.
***** DIAGNOSTIC SUMMARY OF BCHECK

```

```

NO WARNINGS
NO ERRORS
NO SYSTEM-ERRORS

```

```

10062 LOCAL CHECKS HAVE BEEN DONE
144 RECORD/TABLE-OCCURRENCES HAVE BEEN CHECKED AGAINST DBTT
0 CHAIN-SET-MEMBERSHIPS HAVE BEEN CHECKED
0 REFERENCES BETWEEN TABLE-OCCURRENCES HAVE BEEN CHECKED
0 REFERENCES FROM TABLES TO MEMBER-RECORDS HAVE BEEN CHECKED

```

```

NO EASY (MINOR) LOCAL CONSISTENCY ERRORS
NO FATAL (SERIOUS, STRUCTURAL) LOCAL CONSISTENCY ERRORS

```

```
+++++          1 GLOBAL CONSISTENCY ERRORS

***** END OF DIAGNOSTIC SUMMARY
***** NR OF DATABASE ACCESSES :          67
***** NORMAL END   BCHECK      (UDS/SQL V2.8 0000 )   2015-06-28  11:36:10
```

### Ausschnitte aus dem SYSLST-Protokoll:

```
.
.
.
***** THE FOLLOWING CHANGE OF DATABASE IS NOT CONSISTENT
CHECK ELEMENT : 0002 00000021 0002 FFFF 03000066 01 00 0000 00000000
FOR OWNER-DBKEY 0002-000000000021 DBTT-COL 2
      A TABLE-OCCURRENCE AT TABLE-LEVEL MAX. WAS ADDED TO PAGE 03000066
      WITH NO CORRESPONDING CHANGE OF THE REFERENCE IN THE DBTT.
.
.
.
```

## 3.7 Meldungen

Die Meldungen, die BCHECK ausgibt, lassen sich in vier Gruppen unterteilen:

- **Warnungen**

Warnungen beziehen sich auf Umstände, die einen ordnungsgemäßen Ablauf des BCHECK nicht beeinträchtigen. Möglicherweise werden aber nicht alle vom Anwender angegebenen Objekte geprüft. Es wird der Auftragsschalter 30 gesetzt.

- **Fehlermeldungen (Error, System-Error)**

Fehlermeldungen beziehen sich auf Umstände, die einen ordnungsgemäßen Ablauf des BCHECK beeinträchtigen, verhindern oder zu einem vorzeitigen Abbruch des BCHECK führen. Es wird der Auftragsschalter 31 gesetzt.

- **Ablaufmeldungen**

Ablaufmeldungen geben einen Hinweis über den Ablauf von BCHECK und werden auf SYSOUT und SYSLST ausgegeben.

- **Konsistenzfehlermeldungen**

Konsistenzfehlermeldungen weisen auf Konsistenzfehler in der untersuchten Datenbank hin. Sie werden auf SYSLST ausgegeben. Es werden die Auftragsschalter 26 (MINOR LOCAL), 27 (SERIOUS LOCAL und STRUCTURAL LOCAL) und/oder 28 (GLOBAL) gesetzt.

### 3.7.1 Warnungen

Nach dem Abschluss der Eingabe führt BCHECK eine Analyse zwischen den spezifizierten Objekten und den Objekten des Schemas durch.

Ergibt diese Analyse, dass für die geforderten Prüfungen nicht alle notwendigen Realms spezifiziert sind, und nur eine Teilprüfung möglich ist, so erfolgt die entsprechende Warnung.

Warnungen innerhalb eines BCHECK-Laufes mit der Anweisung „CHECK GENERATE SORTING“, die auf eine fehlende Spezifizierung zu prüfender Objekte verweisen, können ignoriert werden, da eine ordnungsgemäße Prüfung bereits beim zugehörigen „CHECK SUMMIMG“ - Lauf erfolgt ist.

Umstände, die zu Warnungen führen, sind zum Beispiel:

- Ein indirekter CALC-Bereich soll geprüft werden und nicht alle CALC-Bereiche des Realm zu Keys mit gleicher Schlüssellänge sind spezifiziert worden.
- Ein Set oder ein Key soll geprüft werden und nicht alle Realms sind angegeben, in denen die Ownersätze bzw. Membersätze oder Tabellen liegen.
- Eine Satzart soll geprüft werden, und nicht alle Realms mit Sätzen dieser Satzart sind angegeben.

### 3.7.2 Fehlermeldungen

Führt die unter Warnungen genannte Analyse der spezifizierten Objekte zu der Erkenntnis, dass angegebene Objekte nicht geprüft werden können, so wird der BCHECK-Lauf nach der Ausgabe entsprechender Fehlermeldungen abgebrochen.

Umstände, die zu Fehlermeldungen führen, sind zum Beispiel:

- Es ist kein Realm spezifiziert.
- Ein Set oder Key soll geprüft werden und die Realms für Ownersätze bzw. Membersätze oder Tabellen sind nicht angegeben.
- Eine Satzart, Set oder Key soll geprüft werden und der DBTT-Realm ist nicht angegeben.

Diese und weitere Fehlermeldungen, die nicht auf Grund der Benutzereingabe entstehen, siehe Handbuch „[Meldungen](#)“. Zu den Fehlermeldungen gehören auch eventuelle Systemfehler.



### 3.7.3 Ablaufmeldungen

#### Meldungen zum Ablauf während der Analysephase

CHECK OF INDICES IS IMPOSSIBLE WITHOUT TOTAL CHECK OF THE DATABASE: THE INDEX CLAUSE IS IGNORED.

##### Bedeutung

Innerhalb einer Inkrementalprüfung kann das Konsistenzkriterium 6 nicht überprüft werden (korrekte Verkettung der Tabellenseiten zwischen den Stufen bezüglich ihrer Indexwerte).

FOR INDEX CHECKS A SORTING MUST BE PERFORMED EVEN FOR THE SUMMING PROCEDURE.

##### Bedeutung

Bei einer Indexwertprüfung müssen, auch bei einer summarischen Prüfung, die Prüfsätze sortiert werden. Innerhalb der Eingabeanalyse wird festgestellt, dass das Konsistenzkriterium 6 zu untersuchen ist (korrekte Verkettung der Tabellenseiten zwischen den Stufen bezüglich ihrer Indexwerte).

\*\*\*\*\* THE INPUT IS CORRECT, THE CHECK RUN IS STARTING NOW.

##### Bedeutung

Die Eingabeanalyse ergab keine Fehlermeldungen. Die eigentliche Konsistenzprüfung beginnt.

#### Meldungen zum augenblicklich verarbeiteten Realm

\*\*\*\*\* ANALYSING NEW STATE OF REALM *realmname* FROM FILE *filename anzahl* NON EMPTY BLOCKS HAVE BEEN ANALYSED.

##### Bedeutung

Totalprüfung mit statistischem Hinweis. Sind keine DBTT-Ankerseiten, keine DBTT-Seiten, keine CALC-Seiten und nur leere Datenseiten vorhanden, so sind nur Verwaltungsdaten im Realm zu prüfen. In diesem Fall wird nur diese Prüfung durchgeführt, es wird aber keine Meldung für den Realm ausgegeben.

\*\*\*\*\* ANALYSING NEW STATE OF REALM *realmname* FROM FILE *filename* AGAINST OLD STATE FROM FILE *filename anzahl* CHANGED BLOCKS HAVE BEEN ANALYSED.

##### Bedeutung

Inkrementalprüfung ORIG/COPY 11 COPY mit statistischem Hinweis

## Meldungen zur Sortierungsphase

\*\*\*\*\* SORTING STARTED DATE AND TIME JJJJ-MM-TT HH:MM:SS

\*\*\*\*\* SORTING COMPLETED DATE AND TIME JJJJ-MM-TT HH:MM:SS

### Bedeutung

Diese Meldungen werden jeweils unmittelbar vor und nach dem Sortieren aller Prüfsätze auf SYSOUT und SYSLST ausgegeben. Zwischen den beiden Meldungen werden folgende Aktionen ausgeführt:

- DB-Zugriff mit den lokalen Konsistenzprüfungen und dem Erzeugen der Prüfsätze
- Sortieren der Prüfsätze
- Auswerten der Prüfsätze und ggf. Generieren der Konsistenzfehlermeldungen

## Meldungen zur Konsistenzprüfung bei summarischer Prüfung

```
+++++ GLOBAL CONSISTENCY ERRORS IN {TYPE-NR} {type-nr}
                                     {REF-NR} {ref-nr}
```

```
[ - SYSTEM-ANCHOR-DBKEY = 0000-rsq          ]
[ - KEY-LENGTH           = key-length       ]
[ - DBTT-COL-NR         = dbtt-spalten-nr  ]
[ - AREA-REF            = area-ref         ]
```

*type-nr* bezeichnet die Konsistenzkriterien, zu denen BCHECK Konsistenzfehler entdeckt hat (siehe [Seite 85](#))

*ref-nr* bezeichnet die fehlerhaften Satzarten zu den Konsistenzkriterien 1-8

*type-nr* = 1      Satzart zur fehlerhaften DBTT

*type-nr* = 2-8    Ownersatzart für die zugehörige Setbeziehung

*0000-rsq*

wird ausgegeben bei *ref-nr* = 0; bezeichnet den DB-Key eines Ankersatzes

*key-length*

wird ausgegeben bei Konsistenzkriterium 9 und 10; bezeichnet die Schlüssellänge des CALC-Keys

*dbtt-spalten-nr*

wird ausgegeben bei Konsistenzkriterium 11; bezeichnet die Spaltennummer der DBTT

*area-ref*

wird ausgegeben bei Konsistenzkriterien 9-11

*type-nr* = 9,10    Realm, in dem sich die indirekte CALC-Tabelle befindet

*type-nr* = 11     Realm, in dem sich die Duplikat-Tabelle befindet.

**SUMMARY-Report**

\*\*\*\*\* DIAGNOSTIC SUMMARY OF BCHECK

{NO/ *anzahl*} WARNINGS           bezieht sich auf die Warnungen  
 {NO/ *anzahl*} ERRORS            und  
 {NO/ *anzahl*} SYSTEM-ERRORS   bezieht sich auf die Fehlermeldungen

Konnte eine Konsistenzprüfung durchgeführt werden, erfolgen weitere Statistikmeldungen:

*anzahl* RECORD/TABLE ...

Anzahl der geprüften Verweise vom Act-Key in einer DBTT-Spalte auf den Satz bzw. auf die höchste Tabellenseite bzw. erste Tabellenseite (Konsistenzkriterium 1, siehe [Seite 85](#))

*anzahl* CHAIN-SET-...

Anzahl der geprüften Sätze

- in einem Set MODE IS CHAIN einschließlich des Ownersatzes (Konsistenzkriterium 2)
- in einem Set MODE IS CHAIN LINKED TO PRIOR einschließlich des Ownersatzes (Konsistenzkriterium 3)

*anzahl* REFERENCES BETWEEN ...

Anzahl der geprüften Verweise zwischen

- der ersten Tabellenseite und der letzten Tabellenseite auf Stufe 0 (Konsistenzkriterium 4)
- der Tabellenseiten zwischen den Stufen (Konsistenzkriterium 5)
- der Tabellenseiten zwischen den Stufen bezüglich ihrer Indexwerte (Konsistenzkriterium 6)
- den Tabellenseiten innerhalb einer Stufe (Konsistenzkriterium 7)
- einer CALC-Tabellen-Überlaufkette (Konsistenzkriterium 10)
- einer Duplikat-Tabellen-Überlaufkette (Konsistenzkriterium 11)

*anzahl* REFERENCES FROM ...

Anzahl der geprüften Verweise von

- Tabelleneinträgen der Stufe 0 auf die Sätze (Konsistenzkriterium 8) und
- den indirekten Tabelleneinträgen auf die Sätze (Konsistenzkriterium 9)

*anzahl* LOCAL CHECKS HAVE BEEN DONE ...

Anzahl der insgesamt erfolgten lokalen Prüfungen in allen zu prüfenden Realms.

*anzahl* USERKEYS BETWEEN TABLES AND MEMBER-RECORDS HAVE BEEN CHECKED

Anzahl derjenigen unter REFERENCES FROM TABLES ... ausgegebenen Prüfungen, die mit Schlüsselwertprüfung durchgeführt wurden.

- **lokale Prüfung**

$$\left\{ \begin{array}{l} \text{NO} \\ \text{anzahl} \end{array} \right\} \left\{ \begin{array}{l} \text{EASY (MINOR)} \\ \text{FATAL (SERIOUS, STRUCTURAL)} \end{array} \right\} \text{LOCAL CONSISTENCY ERRORS.}$$

Anzahl der Konsistenzfehler, die BCHECK bei dem Prüflauf in der Datenbank entdeckt hat

**EASY (MINOR) LOCAL CONSISTENCY ERRORS**

weniger wichtige Konsistenzfehler, d.h. lokale Fehler, die den Datenbankbetrieb nicht behindern

**FATAL (SERIOUS, STRUCTURAL) LOCAL CONSISTENCY ERRORS**

schwerwiegende Konsistenzfehler, d.h. lokale Fehler, die behoben werden müssen, da sie einen korrekten Datenbankbetrieb behindern oder verhindern

SERIOUS bezeichnet einen Fehler, der in der Regel nur ein einzelnes in einer Seite vorhandenes Objekt betrifft.

Bei STRUCTURAL sind mehrere Objekte einer Seite betroffen oder die Seite als Ganzes; als Folge kann eine große Zahl von globalen Konsistenzfehlern nur für diese eine Seite gemeldet werden.

- **summarische Prüfung**

$$\left\{ \begin{array}{l} \text{NO GLOBAL CHECK HAS BEEN DONE} \\ \text{NO GLOBAL CONSISTENCY ERRORS} \\ \text{GLOBAL CONSISTENCY ERRORS. DO A SORTING CHECK} \end{array} \right\}$$

Diese Meldungen beziehen sich auf die Konsistenzkriterien 1-5 und 7-11, siehe [Seite 85](#).

$$\left\{ \begin{array}{l} \text{NO GLOBAL INDEX CHECK HAS BEEN DONE} \\ \left\{ \begin{array}{l} \text{NO} \\ \text{anzahl} \end{array} \right\} \text{GLOBAL CONSISTENCY ERRORS IN INDEX CHECK} \end{array} \right\}$$

Diese Meldungen beziehen sich nur auf das Konsistenzkriterium 6 ([korrekte Verkettung der Tabellenseiten zwischen den Stufen bezüglich ihrer Indexwerte](#)).

- **Sortierungsprüfung**

```
{ NO GLOBAL CHECK HAS BEEN DONE }
{ { NO } GLOBAL CONSISTENCY ERRORS }
{ { anzahl } }
```

Anzahl der globalen Konsistenzfehler bezüglich der Konsistenzkriterien 1-11, siehe [Seite 85](#).

\*\*\*\*\*END OF DIAGNOSTIC SUMMARY

**Bedeutung**

Ende des SUMMARY-Reports

### 3.7.4 Konsistenzfehlermeldungen

Hat BCHECK lokale oder globale Konsistenzfehler erkannt, so meldet er diese auf SYSLST, globale Konsistenzfehler allerdings nur bei einer Sortierungsprüfung. Außerdem setzt er einen Auftragschalter (siehe [Abschnitt „Verwendung von Auftragschaltern“ auf Seite 136](#)).

#### 3.7.4.1 Globale Konsistenzfehler (keine Indexwertprüfung)

Aufbau des Meldungsschlüssels:

Meldungen zu globalen Konsistenzfehlern beginnen mit einer fünfstelligen Nummer, eingeleitet mit „G“ für „Global“ und gefolgt von zwei zweistelligen Zahlen:

Die erste Zahl bezeichnet das Konsistenzkriterium der TYPE-Anweisung. Dabei wird bei der Darstellung des Konsistenzkriteriums 1 noch unterschieden zwischen DBTT-Spalte 0 (Wert 00) und DBTT-Spalte > 0 (Wert 01).

Die zweite Zahl bedeutet folgendes:

- 00 Existenzsatz aus der Originaldatenbank (bzw. der neueren Schattendatenbank bei Inkrementalprüfung)
- 01 Verweissatz aus der Originaldatenbank (bzw. der neueren Schattendatenbank bei Inkrementalprüfung)
- 02 Existenzsatz aus der (älteren) Schattendatenbank bei Inkrementalprüfung
- 03 Verweissatz aus der (älteren) Schattendatenbank bei Inkrementalprüfung

Fehlermeldungen zu globalen Konsistenzfehlern, die nicht mit der Indexwertprüfung zusammenhängen, beginnen mit der Überschriftzeile

```
THE FOLLOWING CHANGE OF DATABASE IS NOT CONSISTENT
```

Entdeckt BCHECK bezüglich einer eigentlich konsistenten, globalen Beziehung zusätzliche Einträge (z.B. zwei Satzeinträge mit gleichem DB-Key, der DBTT-Eintrag verweist auf einen von den beiden), so protokolliert BCHECK die konsistente Beziehung im Anschluss an die Konsistenzfehlermeldung. Der Überschriftzeile:

```
THE FOLLOWING DATABASE CHANGES MAY BE CONNECTED WITH THE CONSISTENCY ERROR
```

folgen in dem Fall die entsprechenden Meldungen ohne den Zusatz:

```
WITH NO CORRESPONDING CHANGE . . . . .
```

Im Folgenden sind alle Meldungen zu globalen Konsistenzfehlern - außer denen zur Indexwertprüfung - erläutert. Beachten Sie dabei, dass Meldungen mit dem Text

..... DELETED FROM .....

nur bei einer Inkrementalprüfung ausgegeben werden. Diese Meldungen beziehen sich immer auf die ältere Schattendatenbank. Die zu diesen Meldungen angegebenen Erklärungen hingegen beziehen sich auf das Original (bzw. die neuere Schattendatenbank) und sind **nur sinnvoll**, wenn die ältere Schattendatenbank konsistent ist.

Falls die ältere Schattendatenbank nicht konsistent sein sollte, gilt für Konsistenzfehler in dieser Schattendatenbank die Erklärung der entsprechenden Meldung mit ADDED TO.

G0000 FOR DBKEY *dbkey* THE RECORD-OCCURRENCE WAS ADDED TO PAGE *act-key* WITH NO CORRESPONDING CHANGE OF A REFERENCE FROM DBTT-COL-0.

**Bedeutung**

Der DBTT-Eintrag für den Satz *dbkey* in der Seite *act-key* ist falsch.

G0001 FOR DBKEY *dbkey* AN ENTRY WAS ADDED TO DBTT-COL-0 POINTING TO PAGE *act-key* WITH NO CORRESPONDING CHANGE OF LOCATION OF THE RECORD-OCCURRENCE.

**Bedeutung**

Für den Satz *dbkey* existiert ein DBTT-Eintrag, der auf die Seite *act-key* verweist. Der Satz ist in dieser Seite nicht enthalten.

G0002 FOR DBKEY *dbkey* THE RECORD-OCCURRENCE WAS DELETED FROM PAGE *act-key* WITH NO CORRESPONDING CHANGE OF A REFERENCE FROM THE DBTT-COL-0.

**Bedeutung**

Für den Satz *dbkey* existiert ein DBTT-Eintrag, der auf die Seite *act-key* verweist. Der Satz ist in dieser Seite nicht enthalten.

G0003 FOR DBKEY *dbkey* AN ENTRY WAS DELETED FROM DBTT-COL-0 POINTING TO PAGE *act-key* WITH NO CORRESPONDING CHANGE OF LOCATION OF THE RECORD-OCCURRENCE.

**Bedeutung**

Der DBTT-Eintrag für den Satz *dbkey* in der Seite *act-key* ist falsch.

G0100 FOR OWNER-DBKEY *dbkey* DBTT-COL *colnr* A TABLE-OCCURRENCE AT TABLE LEVEL  $\{levnr|MAX\}$  WAS ADDED TO PAGE *act-key-1* (POINTING TO NEXT HIGHER TABLE IN PAGE *act-key-2*) WITH NO CORRESPONDING CHANGE OF THE REFERENCE IN THE  $\{DBTT|SYSTEM-RECORD\}$ .

**Bedeutung**

Bei einer Tabelle, die durch den Ownersatz *dbkey* und die DBTT-Spaltennummer *colnr* eindeutig bestimmt ist, fehlt zu einer Tabellenseite mit Stufe *levnr* in der Seite *act-key-1* der Verweis von der DBTT oder dem Ankersatz. Bei Tabellenseiten der höchsten Stufe steht MAX an Stelle von *levnr*.

G0101 FOR OWNER-DBKEY *dbkey* DBTT-COL *colnr* AN ENTRY REFERRING TO A TABLE-OCCURRENCE AT LEVEL *levnr* WAS ADDED TO THE {DBTT|SYSTEM-RECORD} POINTING TO A PAGE *act-key-1* (WITH THE REFERENCE COMING FROM PAGE *act-key-2*) WITH NO CORRESPONDING CHANGE OF LOCATION OF THE TABLE-OCCURRENCE POINTED AT.

**Bedeutung**

Bei der Tabelle, die durch den Ownersatz *dbkey* und die DBTT-Spaltennummer *colnr* eindeutig bestimmt ist, fehlt eine Tabellenseite der Stufe *levnr* in der Seite *act-key-1*, auf die von der DBTT oder vom Ankersatz verwiesen wird.

G0102 FOR OWNER-DBKEY *dbkey* DBTT-COL *colnr* A TABLE-OCCURRENCE AT TABLE LEVEL {*levnr*|MAX} WAS DELETED FROM PAGE *act-key-1* (POINTING TO NEXT HIGHER TABLE IN PAGE *act-key-2*) WITH NO CORRESPONDING CHANGE OF THE REFERENCE IN THE {DBTT|SYSTEM-RECORD}.

**Bedeutung**

Bei einer Tabelle, die durch den Ownersatz *dbkey* und die DBTT-Spaltennummer *colnr* eindeutig bestimmt ist, fehlt eine Tabellenseite der Stufe *levnr* in der Seite *act-key-1*, auf die von der DBTT oder vom Ankersatz verwiesen wird. Bei Tabellenseiten der höchsten Stufe steht MAX an Stelle von *levnr*.

G0103 FOR OWNER-DBKEY *dbkey* DBTT-COL *colnr* AN ENTRY REFERRING TO A TABLE-OCCURRENCE AT LEVEL *levnr* WAS DELETED FROM THE TABLE AT NEXT HIGHER LEVEL POINTING TO A PAGE *act-key-1* (WITH THE REFERENCE COMING FROM PAGE *act-key-2*) WITH NO CORRESPONDING CHANGE OF LOCATION OF THE TABLE-OCCURRENCE POINTED AT.

**Bedeutung**

Bei der mehrstufigen Tabelle, die durch den Ownersatz *dbkey* und die DBTT-Spaltennummer *colnr* eindeutig bestimmt ist, fehlt zu einer Tabellenseite mit Stufe *levnr* in der Seite *act-key-1* der Verweis von der DBTT oder dem Ankersatz.

G0200

G0300 FOR OWNER-DBKEY *dbkey-1* SET-REF *setref* A RECORD-OCCURRENCE WAS ADDED TO THE OWNER'S SET CHAIN WITH MEMBER-DBKEY *dbkey-2* (POINTING TO PRIOR DBKEY *dbkey-3*) WITH NO CORRESPONDING CHANGE OF A FORWARD POINTER IN THE SET CHAIN.

**Bedeutung**

Bei dem Set *setref* existiert in der Set-Occurrence des Owners kein Verweis auf den Satz *dbkey-2*. Bei Rückwärtsverkettung ist der Membersatz *dbkey-3*, der den Verweis enthalten müsste, mit angegeben.

G0201

G0301 FOR OWNER-DBKEY *dbkey-1* SET-REF *setref* A FORWARD POINTER WAS ADDED TO THE OWNER'S SET CHAIN POINTING TO MEMBER-DBKEY *dbkey-2* (FROM DBKEY *dbkey-3*) WITH NO CORRESPONDING CHANGE OF THE MEMBER-RECORD POINTED AT.

**Bedeutung**

Bei dem Set *setref* existiert in der Set-Occurrence des Owners *dbkey-1* ein Verweis auf den Satz *dbkey-2*, der aber kein Mitglied der Kette ist. Bei Rückwärtsverkettung ist der Membersatz *dbkey-3*, der den Verweis enthält, mit angegeben.



G0202

G0302

FOR OWNER-DBKEY *dbkey-1* SET-REF *setref* A RECORD-OCCURRENCE WAS DELETED FROM THE OWNER'S SET CHAIN WITH MEMBER-DBKEY *dbkey-2* (POINTING TO PRIOR DBKEY *dbkey-3*) WITH NO CORRESPONDING CHANGE OF A FORWARD POINTER IN THE SET CHAIN.

**Bedeutung**

Bei dem Set *setref* existiert in der Set-Occurrence des Owners *dbkey-1* ein Verweis auf den Satz *dbkey-2*, der aber kein Mitglied der Kette ist. Bei Rückwärtsverkettung ist der Membersatz *dbkey-3*, der den Verweis enthält, mit angegeben.

G0203

G0303

FOR OWNER-DBKEY *dbkey-1* SET-REF *setref* A FORWARD POINTER WAS DELETED FROM THE OWNER'S SET CHAIN POINTING TO MEMBER-DBKEY *dbkey-2* (FROM DBKEY *dbkey-3*) WITH NO CORRESPONDING CHANGE OF THE MEMBER-RECORD POINTED AT.

**Bedeutung**

Bei dem Set *setref* existiert in der Set-Occurrence des Owners kein Verweis auf den Satz *dbkey-2*. Bei Rückwärtsverkettung ist der Membersatz *dbkey-3*, der den Verweis enthalten müsste, mit angegeben.

G0400

FOR OWNER-DBKEY *dbkey* DBTT-COL *colnr* A LAST TABLE-OCCURRENCE WAS ADDED TO PAGE *act-key* WITH NO CORRESPONDING CHANGE OF THE REFERENCE IN THE TOP-TABLE.

**Bedeutung**

Bei der mehrstufigen Tabelle, die durch den Ownersatz *dbkey* und die DBTT-Spaltennummer *colnr* eindeutig bestimmt ist, fehlt zu der letzten Tabellenseite der untersten Stufe in der Seite *act-key* der zugehörige Verweis aus der Tabellenseite der höchsten Stufe.

Oder:

Bei der flachen Tabelle, die durch den Ownersatz *dbkey* und die DBTT-Spaltennummer *colnr* eindeutig bestimmt ist, fehlt zu der letzten Tabellenseite (*act-key*) der zugehörige Verweis aus der ersten Tabellenseite.

G0401

FOR OWNER-DBKEY *dbkey* DBTT-COL *colnr* THE REFERENCE TO THE LAST TABLE WAS ADDED TO THE TOP-TABLE POINTING TO PAGE *act-key* WITH NO CORRESPONDING CHANGE OF LOCATION OF THE LAST TABLE-OCCURRENCE.

**Bedeutung**

Bei der mehrstufigen Tabelle, die durch den Ownersatz *dbkey* und die DBTT-Spaltennummer *colnr* eindeutig bestimmt ist, beinhaltet die Seite *act-key*, auf die im Tabellenkopf der höchsten Stufe verwiesen wird, die letzte Tabellenseite der untersten Stufe nicht.

Oder:

Bei der flachen Tabelle, die durch den Ownersatz *dbkey* und die DBTT-Spaltennummer *colnr* eindeutig bestimmt ist, beinhaltet die Seite *act-key*, auf die im Tabellenkopf der ersten Seite verwiesen wird, die letzte Tabellenseite nicht.

G0402 FOR OWNER-DBKEY *dbkey* DBTT-COL *colnr* A LAST TABLE-OCCURRENCE WAS DELETED FROM PAGE *act-key* WITH NO CORRESPONDING CHANGE OF THE REFERENCE IN THE TOP-TABLE.

**Bedeutung**

Bei der mehrstufigen Tabelle, die durch den Ownersatz *dbkey* und die DBTT-Spaltennummer *colnr* eindeutig bestimmt ist, beinhaltet die Seite *act-key*, auf die im Tabellenkopf der höchsten Stufe verwiesen wird, die letzte Tabellenseite der untersten Stufe nicht.

Oder:

Bei der flachen Tabelle, die durch den Ownersatz *dbkey* und die DBTT-Spaltennummer *colnr* eindeutig bestimmt ist, beinhaltet die Seite *act-key*, auf die im Tabellenkopf der ersten Seite verwiesen wird, die letzte Tabellenseite nicht.

G0403 FOR OWNER-DBKEY *dbkey* DBTT-COL *colnr* THE REFERENCE TO THE LAST TABLE WAS DELETED FROM THE TOP-TABLE POINTING TO PAGE *act-key* WITH NO CORRESPONDING CHANGE OF LOCATION OF THE LAST TABLE-OCCURRENCE.

**Bedeutung**

Bei der mehrstufigen Tabelle, die durch den Ownersatz *dbkey* und die DBTT-Spaltennummer *colnr* eindeutig bestimmt ist, fehlt zu der letzten Tabellenseite der untersten Stufe in der Seite *act-key* der zugehörige Verweis aus der Tabellenseite der höchsten Stufe.

Oder:

Bei der flachen Tabelle, die durch den Ownersatz *dbkey* und die DBTT-Spaltennummer *colnr* eindeutig bestimmt ist, fehlt zu der letzten Tabellenseite (*act-key*) der zugehörige Verweis aus der ersten Tabellenseite.

G0500 FOR OWNER-DBKEY *dbkey* DBTT-COL *colnr* A TABLE-OCCURRENCE AT TABLE LEVEL {*levnr*|MAX} WAS ADDED TO PAGE *act-key-1* (POINTING TO NEXT HIGHER TABLE IN PAGE *act-key-2*) WITH NO CORRESPONDING CHANGE OF THE REFERENCE IN THE TABLE AT NEXT HIGHER LEVEL.

**Bedeutung**

Bei einer mehrstufigen Tabelle, die durch den Ownersatz *dbkey* und die DBTT-Spaltennummer *colnr* eindeutig bestimmt ist, fehlt zu einer Tabellenseite mit Stufe *levnr* in der Seite *act-key-1* der Verweis von der nächsthöheren Stufe (Tabelleneintrag in Seite *act-key-2*). Bei Tabellenseiten der höchsten Stufe steht MAX an Stelle von *levnr*.

G0501 FOR OWNER-DBKEY *dbkey* DBTT-COL *colnr* AN ENTRY REFERRING TO A TABLE-OCCURRENCE AT LEVEL *levnr* WAS ADDED TO THE TABLE AT NEXT HIGHER LEVEL POINTING TO A PAGE *act-key-1* (WITH THE REFERENCE COMING FROM PAGE *act-key-2*) WITH NO CORRESPONDING CHANGE OF LOCATION OF THE TABLE-OCCURRENCE POINTED AT.

**Bedeutung**

Bei der mehrstufigen Tabelle, die durch den Ownersatz *dbkey* und die DBTT-Spaltennummer *colnr* eindeutig bestimmt ist, fehlt eine Tabellenseite der Stufe *levnr* in der Seite *act-key-1*, auf die von der nächsthöheren Stufe (Tabelleneintrag in der Seite *act-key-2*) verwiesen wird.

- G0502 FOR OWNER-DBKEY *dbkey* DBTT-COL *colnr* A TABLE-OCCURRENCE AT TABLE LEVEL {*levnr*|MAX} WAS DELETED FROM PAGE *act-key-1* (POINTING TO NEXT HIGHER TABLE IN PAGE *act-key-2*) WITH NO CORRESPONDING CHANGE OF THE REFERENCE IN THE TABLE AT NEXT HIGHER LEVEL.
- Bedeutung**  
Bei einer mehrstufigen Tabelle, die durch den Ownersatz *dbkey* und die DBTT-Spaltennummer *colnr* eindeutig bestimmt ist, fehlt eine Tabellenseite der Stufe *levnr* in der Seite *act-key-1*, auf die von der nächsthöheren Stufe (Tabelleneintrag in der Seite *act-key-2*) verwiesen wird. Bei Tabellenseiten der höchsten Stufe steht MAX an Stelle von *levnr*.
- G0503 FOR OWNER-DBKEY *dbkey* DBTT-COL *colnr* AN ENTRY REFERRING TO A TABLE-OCCURRENCE AT LEVEL *levnr* WAS DELETED FROM THE {DBTT|SYSTEM-RECORD} POINTING TO A PAGE *act-key-1* (WITH THE REFERENCE COMING FROM PAGE *act-key-2*) WITH NO CORRESPONDING CHANGE OF LOCATION OF THE TABLE-OCCURRENCE POINTED AT.
- Bedeutung**  
Bei der Tabelle, die durch den Ownersatz *dbkey* und die DBTT-Spaltennummer *colnr* eindeutig bestimmt ist, fehlt zu einer Tabellenseite mit Stufe *levnr* in der Seite *act-key-1* der Verweis von der DBTT oder dem Ankersatz.
- G0700 FOR OWNER-DBKEY *dbkey* DBTT-COL *colnr* A TABLE-OCCURRENCE AT TABLE LEVEL *levnr* WAS ADDED TO PAGE *act-key-1* POINTING TO PRIOR TABLE IN PAGE *act-key-2* WITH NO CORRESPONDING CHANGE OF A FORWARD POINTER IN THE TABLE CHAIN.
- Bedeutung**  
Bei der Tabelle, die durch den Ownersatz *dbkey* und die DBTT-Spaltennummer *colnr* eindeutig bestimmt ist, fehlt auf gleicher Stufe *levnr* zur Rückwärtsverkettung in der Seite *act-key-1* die zugehörige Vorwärtsverkettung aus der Seite *act-key-2*.
- G0701 FOR OWNER-DBKEY *dbkey* DBTT-COL *colnr* A TABLE-OCCURRENCE AT TABLE LEVEL *levnr* WAS ADDED TO PAGE *act-key-1* POINTING TO NEXT-TABLE IN PAGE *act-key-2* WITH NO CORRESPONDING CHANGE OF A BACKWARD POINTER IN THE TABLE CHAIN.
- Bedeutung**  
Bei der Tabelle, die durch den Ownersatz *dbkey* und die DBTT-Spaltennummer *colnr* eindeutig bestimmt ist, fehlt auf gleicher Stufe *levnr* zur Vorwärtsverkettung in der Seite *act-key-1* die zugehörige Rückwärtsverkettung aus der Seite *act-key-2*.
- G0702 FOR OWNER-DBKEY *dbkey* DBTT-COL *colnr* A TABLE-OCCURRENCE AT TABLE LEVEL *levnr* WAS DELETED FROM PAGE *act-key-1* POINTING TO PRIOR TABLE IN PAGE *act-key-2* WITH NO CORRESPONDING CHANGE OF A FORWARD POINTER IN THE TABLE CHAIN.
- Bedeutung**  
Bei der Tabelle, die durch den Ownersatz *dbkey* und die DBTT-Spaltennummer *colnr* eindeutig bestimmt ist, fehlt auf gleicher Stufe *levnr* zur Vorwärtsverkettung in der Seite *act-key-2* die zugehörige Rückwärtsverkettung in der Seite *act-key-1*.

G0703 FOR OWNER-DBKEY *dbkey* DBTT-COL *colnr* A TABLE-OCCURRENCE AT TABLE LEVEL *levnr* WAS DELETED FROM PAGE *act-key-1* POINTING TO NEXT-TABLE IN PAGE *act-key-2* WITH NO CORRESPONDING CHANGE OF A BACKWARD POINTER IN THE TABLE CHAIN.

**Bedeutung**

Bei der Tabelle, die durch den Ownersatz *dbkey* und die DBTT-Spaltennummer *colnr* eindeutig bestimmt ist, fehlt auf gleicher Stufe *levnr* zur Rückwärtsverkettung in der Seite *act-key-2* die zugehörige Vorwärtsverkettung aus der Seite *act-key-1*.

G0800 FOR OWNER-DBKEY *dbkey-1* DBTT-COL *colnr* THE MEMBERSHIP-INDICATOR WAS ADDED TO THE RECORD WITH MEMBER-DBKEY *dbkey-2* (LOCATED IN PAGE *act-key*) WITH NO CORRESPONDING CHANGE OF A POINTER IN SOME BOTTOM TABLE. CONTENT OF KEYVALUE OR TABLE ENTRY: *keyvalue*.

**Bedeutung**

In der SEARCH-Key- bzw. Sort-Key-Tabelle, die durch den Ownersatz *dbkey-1* und die DBTT-Spaltennummer *colnr* eindeutig bestimmt ist, existiert für den Satz *dbkey-2* ein Tabelleneintrag auf Stufe 0, obwohl dieser Satz kein Member des zugehörigen Sets ist.

G0801 FOR OWNER-DBKEY *dbkey-1* DBTT-COL *colnr* AN ENTRY WAS ADDED TO A BOTTOM TABLE POINTING TO MEMBER-RECORD WITH DBKEY *dbkey-2* (WITH THE TABLE BEING LOCATED IN PAGE *act-key*) WITH NO CORRESPONDING CHANGE OF THE RECORD-OCCURRENCE. CONTENT OF KEYVALUE OR TABLE ENTRY: *keyvalue*.

**Bedeutung**

In der SEARCH-Key- bzw. Sort-Key-Tabelle, die durch den Ownersatz *dbkey-1* und die DBTT-Spaltennummer *colnr* eindeutig bestimmt ist, existiert für den Satz *dbkey-2* ein Tabelleneintrag auf Stufe 0 (gespeichert in der Seite *act-key*), obwohl dieser Satz kein Member des zugehörigen Sets ist.

G0802 FOR OWNER-DBKEY *dbkey-1* DBTT-COL *colnr* THE MEMBERSHIP-INDICATOR WAS DELETED FROM THE RECORD WITH MEMBER-DBKEY *dbkey-2* (LOCATED IN PAGE *act-key*) WITH NO CORRESPONDING CHANGE OF A POINTER IN SOME BOTTOM TABLE. CONTENT OF KEYVALUE OR TABLE ENTRY: *keyvalue*.

**Bedeutung**

In der SEARCH-Key- bzw. Sort-Key-Tabelle, die durch den Ownersatz *dbkey-1* und die DBTT-Spaltennummer *colnr* eindeutig bestimmt ist, fehlt zum Membersatz *dbkey-2* (gespeichert in der Seite *act-key*) der zugehörige Tabelleneintrag auf Stufe 0.

G0803 FOR OWNER-DBKEY *dbkey-1* DBTT-COL *colnr* AN ENTRY WAS DELETED FROM A BOTTOM TABLE POINTING TO A MEMBER-RECORD WITH *dbkey-2* (WITH THE TABLE BEING LOCATED IN PAGE *act-key*) WITH NO CORRESPONDING CHANGE OF THE RECORD-OCCURRENCE. CONTENT OF KEYVALUE OR TABLE ENTRY: *keyvalue*.

**Bedeutung**

In der SEARCH-Key- bzw. Sort-Key-Tabelle, die durch den Ownersatz *dbkey-1* und die DBTT-Spaltennummer *colnr* eindeutig bestimmt ist, fehlt zum Membersatz *dbkey-2* der zugehörige Tabelleneintrag auf Stufe 0 (gespeichert in der Seite *act-key*).

- G0900 A RECORD-OCCURRENCE WITH RSQ *rsq* WHICH HAS AN (INDIRECT) CALC-(SEARCH-)KEY WITH KEYLENGTH *keylength* WAS ADDED TO THE PAGE *act-key* WITH NO CORRESPONDING CHANGE IN THE CALC-TABLE. CONTENT OF KEYVALUE OR TABLE ENTRY: *keyvalue*
- Bedeutung**  
In der CALC-Tabelle des indirekten CALC-Keys bzw. CALC-SEARCH-Keys mit Schlüssellänge *keylength* fehlt der Verweis auf den in der Seite *act-key* gespeicherten Satz mit Satzfolgennummer *rsq*.
- G0901 FOR AN (INDIRECT) CALC-(SEARCH-)KEY WITH KEYLENGTH *keylength* AN ENTRY WITH RSQ *rsq* WAS ADDED TO A CALC TABLE AT PAGE *act-key* OF THE DATABASE WITH NO CORRESPONDING CHANGE IN THE RELATED RECORD-OCCURRENCE. CONTENT OF KEYVALUE OR TABLE ENTRY: *keyvalue*
- Bedeutung**  
In der CALC-Tabelle des indirekten CALC-Keys bzw. CALC-SEARCH-Keys mit Schlüssellänge *keylength* verweist in der Seite *act-key* der Eintrag mit Satzfolgennummer *rsq* auf einen nicht-vorhandenen Satz.
- G0902 A RECORD-OCCURRENCE WITH RSQ *rsq* WHICH HAS AN (INDIRECT) CALC-(SEARCH-)KEY WITH KEYLENGTH *keylength* WAS DELETED FROM THE PAGE *act-key* WITH NO CORRESPONDING CHANGE IN THE CALC TABLE. CONTENT OF KEYVALUE OR TABLE ENTRY: *keyvalue*
- Bedeutung**  
In der CALC-Tabelle des indirekten CALC-Keys bzw. CALC-SEARCH-Keys mit Schlüssellänge *keylength* verweist der Eintrag mit Satzfolgennummer *rsq* auf einen nicht-vorhandenen Satz.
- G0903 FOR AN (INDIRECT) CALC-(SEARCH-)KEY WITH KEYLENGTH *keylength* AN ENTRY WITH RSQ *rsq* WAS DELETED FROM A CALC TABLE AT PAGE *act-key* OF THE DATABASE WITH NO CORRESPONDING CHANGE IN THE RELATED RECORD-OCCURRENCE. CONTENT OF KEYVALUE OR TABLE ENTRY: *keyvalue*
- Bedeutung**  
In der CALC-Tabelle des indirekten CALC-Keys bzw. CALC-SEARCH-Keys mit Schlüssellänge *keylength* fehlt der Verweis auf den gespeicherten Satz mit Satzfolgennummer *rsq*.
- G1000 IN A HASH OVERFLOW CHAIN FOR KEYLENGTH *keylength* A POINTER TO PRIOR OVERFLOW PAGE *act-key-1* WAS ADDED TO PAGE *act-key-2* WITH NO CORRESPONDING CHANGE OF THE FORWARD POINTER IN THE OVERFLOW CHAIN
- Bedeutung**  
In einer Hash-Überlaufkette des CALC-Keys mit Schlüssellänge *keylength* existiert in der Seite *act-key-2* eine Rückwärtsverketzung zur Seite *act-key-1* ohne zugehörige Vorwärtsverketzung in der Kette der Überlaufseiten.

G1001 IN A HASH OVERFLOW CHAIN FOR KEYLENGTH *keylength* A POINTER TO NEXT OVERFLOW PAGE *act-key-1* WAS ADDED TO PAGE *act-key-2* WITH NO CORRESPONDING CHANGE OF THE BACKWARD POINTER IN THE OVERFLOW CHAIN.

**Bedeutung**

In einer Hash-Überlaufkette des CALC-Keys mit Schlüssellänge *keylength* existiert in der Seite *act-key-2* eine Vorwärtsverkettung zur Seite *act-key-1* ohne zugehörige Rückwärtsverkettung in der Kette der Überlaufseiten.

G1002 IN A HASH OVERFLOW CHAIN FOR KEYLENGTH *keylength* A POINTER TO PRIOR OVERFLOW PAGE *act-key-1* WAS DELETED FROM PAGE *act-key-2* WITH NO CORRESPONDING CHANGE OF THE FORWARD POINTER IN THE OVERFLOW CHAIN.

**Bedeutung**

In einer Hash-Überlaufkette des CALC-Keys mit Schlüssellänge *keylength* existiert in der Seite *act-key-1* eine Vorwärtsverkettung zur Seite *act-key-2* ohne zugehörige Rückwärtsverkettung in der Kette der Überlaufseiten.

G1003 IN A HASH OVERFLOW CHAIN FOR KEYLENGTH *keylength* A POINTER TO NEXT OVERFLOW PAGE *act-key-1* WAS DELETED FROM PAGE *act-key-2* WITH NO CORRESPONDING CHANGE OF THE BACKWARD POINTER IN THE OVERFLOW CHAIN.

**Bedeutung**

In einer Hash-Überlaufkette des CALC-Keys mit Schlüssellänge *keylength* existiert in der Seite *act-key-1* eine Rückwärtsverkettung zur Seite *act-key-2* ohne zugehörige Vorwärtsverkettung in der Kette der Überlaufseiten.

G1100 IN THE OVERFLOW CHAIN OF A DUPLICATE TABLE WITH MAIN LEVEL *act-key-1* DBTT-COL *colnr* A BACKWARD POINTER TO PAGE *act-key-2* WAS ADDED TO PAGE *act-key-3* WITH NO CORRESPONDING CHANGE IN THE PREDECESSOR.

**Bedeutung**

In der Überlaufkette einer Duplikat-Tabelle, mit der Grundstufe in der Seite *act-key-1* und DBTT-Spaltennummer *colnr*, existiert in der Seite *act-key-3* eine Rückwärtsverkettung zur Seite *act-key-2*, ohne die zugehörige Vorwärtsverkettung in der Kette der Überlaufseiten.

G1101 IN THE OVERFLOW CHAIN OF A DUPLICATE TABLE WITH MAIN LEVEL *act-key-1* DBTT-COL *colnr* A FORWARD POINTER TO PAGE *act-key-2* WAS ADDED TO PAGE *act-key-3* WITH NO CORRESPONDING CHANGE IN THE SUCCESSOR.

**Bedeutung**

In der Überlaufkette einer Duplikat-Tabelle, mit der Grundstufe in der Seite *act-key-1* und DBTT-Spaltennummer *colnr*, existiert in der Seite *act-key-3* eine Vorwärtsverkettung zur Seite *act-key-2*, ohne die zugehörige Rückwärtsverkettung in der Kette der Überlaufseiten.

G1102 IN THE OVERFLOW CHAIN OF A DUPLICATE TABLE WITH MAIN LEVEL *act-key-1* DBTT-COL *colnr* A BACKWARD POINTER TO PAGE *act-key-2* WAS DELETED FROM *act-key-3* WITH NO CORRESPONDING CHANGE IN THE PREDECESSOR.

**Bedeutung**

In der Überlaufkette einer Duplikat-Tabelle, mit der Grundstufe in der Seite *act-key-1* und DBTT-Spaltennummer *colnr*, existiert in der Seite *act-key-2* eine Vorwärtsverkettung zur Seite *act-key-3*, ohne die zugehörige Rückwärtsverkettung in der Kette der Überlaufseiten.

G1103 IN THE OVERFLOW CHAIN OF A DUPLICATE TABLE WITH MAIN LEVEL *act-key-1* DBTT-COL *colnr* A FORWARD POINTER TO PAGE *act-key-2* WAS DELETED FROM PAGE *act-key-3* WITH NO CORRESPONDING CHANGE IN THE SUCCESSOR.

**Bedeutung**

In der Überlaufkette einer Duplikat-Tabelle, mit der Grundstufe in der Seite *act-key-1* und DBTT-Spaltennummer *colnr*, existiert in der Seite *act-key-2* eine Rückwärtsverkettung zur Seite *act-key-3*, ohne die zugehörige Vorwärtsverkettung in der Kette der Überlaufseiten.

### 3.7.4.2 Globale Konsistenzfehler (Indexwertprüfung)

Fehler in den Indexwert-Beziehungen zwischen zwei verketteten Tabellenseiten einer mehrstufigen Tabelle bzw. zwischen den Seiten der Überlaufkette einer Duplikat-Tabelle meldet BCHECK mit der Überschrift:

THE FOLLOWING CHANGE OF DATABASE IS NOT CONSISTENT DUE TO WRONG KEY VALUE RELATIONS IN MATCHING CHECK-RECORDS

Waren die zu vergleichenden Indexwerte unterschiedlich lang, meldet BCHECK in der Überschrift stattdessen:

THE FOLLOWING CHANGE OF DATABASE IS NOT CONSISTENT DUE TO DIFFERENT KEY LENGTHS IN MATCHING CHECK-RECORDS

BCHECK führt in so einem Fall die eigentliche Indexwertprüfung nicht durch.

Die Meldungen der Indexwertprüfung sind im Folgenden erläutert.

\*\*\*THE FOLLOWING DATABASE CHANGES MAY BE CONNECTED WITH THE CONSISTENCY ERROR:

G0500 FOR OWNER-DBKEY *dbkey* DBTT-COL *colnr* A TABLE-OCCURRENCE AT TABLE LEVEL *levnr* WAS ADDED TO PAGE *act-key-1* POINTING TO NEXT HIGHER TABLE IN PAGE *act-key-2*.

CONTENT OF KEYVALUE OR TABLE ENTRY:

*keyvalue-2*  
*rsq-2*

#### Bedeutung

In der mehrstufigen Tabelle, die durch den Ownersatz *dbkey* und die DBTT-Spaltennummer *colnr* eindeutig bestimmt ist, ist in der Tabellenseite *act-key-2* der Stufe *levnr+1* der Indexwert *keyvalue-1 rsq-1* eines Eintrags kleiner als der höchste Indexwert *keyvalue-2 rsq-2* in der Tabellenseite *act-key-1* der nächstniedrigeren Stufe *levnr*, auf die er verweist.

G0501 FOR OWNER-DBKEY *dbkey* DBTT-COL *colnr* AN ENTRY REFERRING TO A TABLE-OCCURRENCE AT LEVEL *levnr* WAS ADDED TO THE TABLE AT NEXT HIGHER LEVEL POINTING TO A PAGE *act-key-1* WITH THE REFERENCE COMING FROM PAGE *act-key-2* WITH NO CORRESPONDING CHANGE OF LOCATION OF THE TABLE-OCCURRENCE POINTED AT.

CONTENT OF KEYVALUE OR TABLE ENTRY:

*keyvalue-1*  
*rsq-1*



\*\*\*THE FOLLOWING DATABASE CHANGES MAY BE CONNECTED WITH THE CONSISTENCY ERROR:

G0600 IN AN INDEXED TABLE FOR OWNER-DBKEY *dbkey* DBTT-COL *colnr* AN ENTRY AT TABLE LEVEL *levnr+1* WAS ADDED TO PAGE *act-key-3* POINTING TO A TABLE AT NEXT-LOWER LEVEL *act-key-2* WITH NO CORRESPONDING CHANGE OF THE SUCCESSOR OF THE REFERENCED TABLE

CONTENT OF KEYVALUE OR TABLE ENTRY:  
*keyvalue-2*  
*rsq-2*

### **Bedeutung**

In der mehrstufigen Tabelle, die durch den Ownersatz *dbkey* und die DBTT-Spaltennummer eindeutig bestimmt ist, ist in der Tabellenseite *act-key-3* der Stufe *levnr+1* der Indexwert *keyvalue-2 rsq-2* eines Eintrags nicht kleiner als der niedrigste Indexwert *keyvalue-1 rsq-1* der Nachfolgeseite *act-key-1* derjenigen Tabellenseite *act-key-2*, auf die der Eintrag verweist.

G0601 IN AN INDEXED TABLE FOR OWNER-DBKEY *dbkey* DBTT-COL *colnr* AN ENTRY AT TABLE LEVEL *levnr* WAS ADDED TO PAGE *act-key-1* POINTING TO ITS PREDECESSOR IN THE TABLE CHAIN *act-key-2* WITH NO CORRESPONDING CHANGE OF THE TABLE AT NEXT HIGHER LEVEL.

CONTENT OF KEYVALUE OR TABLE ENTRY:  
*keyvalue-1*  
*rsq-1*

\*\*\*THE FOLLOWING DATABASE CHANGES MAY BE CONNECTED WITH THE CONSISTENCY ERROR:

G0700 FOR OWNER-DBKEY *dbkey* DBTT-COL *colnr* A TABLE-OCCURRENCE AT TABLE LEVEL *levnr* WAS ADDED TO PAGE *act-key-2* POINTING TO PRIOR TABLE IN PAGE *act-key-1*

CONTENT OF KEYVALUE OR TABLE ENTRY:  
*keyvalue-2*  
*rsq-2*

### **Bedeutung**

In der Tabellenkette, die durch den Ownersatz *dbkey* und die DBTT-Spaltennummer *colnr* eindeutig bestimmt ist, ist der höchste Indexwert *keyvalue-1 rsq-1* in der Tabellenseite *act-key-1* nicht kleiner als der niedrigste Indexwert *keyvalue-2 rsq-2* in der Nachfolgeseite *act-key-2* der Tabellenkette gleicher Stufe *levnr*.

G0701 FOR OWNER-DBKEY *dbkey* DBTT-COL *colnr* A TABLE-OCCURRENCE AT TABLE LEVEL *levnr* WAS ADDED TO PAGE *act-key-1* POINTING TO NEXT-TABLE IN PAGE *act-key-2* WITH NO CORRESPONDING CHANGE OF A BACKWARD POINTER IN THE TABLE CHAIN.

CONTENT OF KEYVALUE OR TABLE ENTRY:  
*keyvalue-1*  
*rsq-1*

G1100 IN THE OVERFLOW CHAIN OF A DUPLICATE-TABLE WITH MAIN LEVEL *act-key-1* DBTT-COL *colnr* A BACKWARD POINTER TO PAGE *act-key-2* WAS ADDED TO PAGE *act-key-3* WITH NO CORRESPONDING CHANGE IN THE PREDECESSOR.

CONTENT OF RSQ:  
*rsq-1*

\*\*\*THE FOLLOWING DATABASE CHANGES MAY BE CONNECTED WITH THE CONSISTENCY ERROR:

G1101 IN THE OVERFLOW CHAIN OF A DUPLICATE-TABLE WITH MAIN LEVEL *act-key-1* DBTT-COL *colnr* A FORWARD POINTER TO PAGE *act-key-3* WAS ADDED TO PAGE *act-key-2*.

CONTENT OF RSQ:  
*rsq-2*

### **Bedeutung**

In der Überlaufkette der Duplikat-Tabelle mit der Grundstufe in der Seite *act-key-1* und DBTT-Spaltennummer *colnr* ist der höchste RSQ-Wert *rsq-2* der Duplikat-Tabelle in der Seite *act-key-2* nicht kleiner als der niedrigste RSQ-Wert *rsq-1* der Nachfolgeseite *act-key-3* der Duplikat-Tabelle.

### 3.7.4.3 Lokale Konsistenzfehler

Aufbau des Meldungsschlüssels:

Meldungen zu lokalen Konsistenzfehlern beginnen mit einer fünfstelligen Nummer, eingeleitet mit "L" für "Local" und gefolgt von einer vierstelligen Zahl, die keine semantische Bedeutung hat.

L0027 STRUCTURAL CONSISTENCY ERROR: FREE-SPACE.DISPL IN THE PAGE HEADER DOES NOT OCCUR AS SMALLEST RECORD-DISPL IN A PAGE INDEX

#### **Bedeutung**

Das FPA-Displacement im Seitenkopf ist nicht als kleinstes Record-Displacement in mindestens einem Seitenindex vorhanden.

L0033 SERIOUS CONSISTENCY ERROR: FPA VALUE FOR {OLD/NEW} {ACT-KEY-0/DBTT ANCHOR PAGE/DBTT PAGE/CALC PAGE/FPA PAGE/TABLE PAGE/DUPLICATE TABLE OVERFLOW PAGE} *act-key* CLAIMS NO FULL PAGE, VALUE = *fpa-wert*

#### **Bedeutung**

Der zentrale FPA-Wert für eine ACT-KEY-0-Seite, DBTT-Ankerseite, DBTT-Seite, CALC-Seite, FPA-Seite, Tabellen-Seite oder Duplicate-Table-Overflow-Seite ist ungleich 0.

L0049 MINOR CONSISTENCY ERROR: BNR OF PPP IN TABLE ENTRY *te-nr* IS WRONG

#### **Bedeutung**

Die Seitennummer im (indirekten CALC-) Tabelleneintrag ist gleich 0 , wobei die Realm-Ref 0 ist.

L0056 SERIOUS CONSISTENCY ERROR: BNR OF PPP IN TABLE ENTRY *te-nr* IS WRONG

#### **Bedeutung**

Die Seitennummer im (indirekten CALC-) Tabelleneintrag ist gleich 0 , wobei die Realm-Ref 0 ist.

L0057 STRUCTURAL CONSISTENCY ERROR: DB-KEY-LIST IN DUPLICATE TABLE FOR TABLE ENTRY *te-nr* HAS AN INVALID LENGTH

#### **Bedeutung**

Die Länge des Duplicate-Table-Eintrags ist kein Vielfaches von 3 bei einer Seitenlänge von 2048 byte bzw. 6 bei einer Seitenlänge von 4000 bzw. 8096 byte.

L0070 SERIOUS CONSISTENCY ERROR: FPA VALUES FOR {DBTT/CALC}-PAGE(S) BETWEEN *seitennummer* AND *seitennummer* INDICATE EMPTY PAGES OR ARE NOT RECOGNIZED AS {DBTT/CALC}-PAGE(S), REALM IS *realmname*

**Bedeutung**

Der zentrale FPA-Wert für die DBTT- Seiten oder CALC-Seiten steht auf X'07EC' bei einer Seitenlänge von 2048 byte, X'0F8C' bei einer Seitenlänge von 4000 byte bzw. X'1F8C' bei einer Seitenlänge von 8096 byte, oder das Layout der Seite entspricht nicht der einer DBTT-Seite oder CALC-Seite.

L0081 STRUCTURAL CONSISTENCY ERROR: TABLE FOR PAGE INDEX *pageindex* IN TABLE LIST DOES NOT OCCUPY A WHOLE PAGE

**Bedeutung**

Die Liste ist allein in der Seite vorgesehen, aber der zentrale FPA-Wert oder die Anzahl der Seitenindex-Einträge oder die Anzahl der für die Liste reservierten Einträge stehen im Widerspruch dazu.

L0084 SERIOUS CONSISTENCY ERROR: FIRST MEMBER DB-KEY *dbkey* IS INVALID (SET-REF = *setref*, SCD-DISPL = *displacement*)

**Bedeutung**

Fehlerhafter DB-Key des ersten Membersatzes in der SCD des Owners.

L0085 SERIOUS CONSISTENCY ERROR: LAST MEMBER DB-KEY *dbkey* IS INVALID (SET-REF = *setref*, SCD-DISPL = *displacement*)

**Bedeutung**

Fehlerhafter DB-Key des letzten Membersatzes in der SCD des Owners.

L0086 SERIOUS CONSISTENCY ERROR: NEXT MEMBER DB-KEY *dbkey* IS INVALID (SET-REF = *setref*, SCD-DISPL = *displacement*)

**Bedeutung**

Fehlerhafter DB-Key des folgenden Membersatzes in der SCD eines Members.

L0087 SERIOUS CONSISTENCY ERROR: PRIOR MEMBER DB-KEY *dbkey* IS INVALID (SET-REF = *setref*, SCD-DISPL = *displacement*)

**Bedeutung**

Fehlerhafter DB-Key des vorangehenden Membersatzes in der SCD eines Members.

L0088 SERIOUS CONSISTENCY ERROR: OWNER SEQUENCE NUMBER *nummer* IS INVALID (SET-REF = *setref*, SCD-DISPL = *displacement*)

**Bedeutung**

In der SCD ist die RSQ des Owners größer als der maximal zulässige Wert.

- L0089      SERIOUS CONSISTENCY ERROR: RECORD-LENGTH IS INVALID OR PAGE INDEX DISPL WRONG
- Bedeutung**  
Die Satzlänge in der SIA unterscheidet sich von der lokal in der Seite berechneten Länge oder das Displacement im Seitenindex ist falsch.
- L0090      SERIOUS CONSISTENCY ERROR: COMPRESSION FORMAT IS INVALID
- Bedeutung**  
Die Satzlänge in der SIA und die lokal in der Seite berechnete Länge eines Compression-Records sind unterschiedlich.
- L0091      SERIOUS CONSISTENCY ERROR: SET RELATION IS NOT CONSISTENT (SET-REF = *setref*, SCD-DISPL = *displacement*)
- Bedeutung**  
Die SCD stimmt mit der Setbeschreibung in der SIA nicht überein.
- L0092      MINOR CONSISTENCY ERROR: PPP *act-key* IS INVALID, REALM IS WRONG (SET-REF = *setref*, SCD-DISPL = *displacement*)
- Bedeutung**  
Der angegebene Pointer enthält entweder eine Realm-Nummer, die nicht im Schema definiert ist, oder er zeigt in einen Realm, in dem der Satz nicht liegen darf (Fehlergewicht EASY (MINOR) LOCAL ERROR).
- L0093      SERIOUS CONSISTENCY ERROR: ACT-KEY *act-key* IN SYSTEM-RECORD IS INVALID, COL-NR=*col-nr*
- Bedeutung**  
Der Act-Key in einer DBTT-Spalte des SYSTEM-Records und der Act-Key der Seite unterscheiden sich bzw. der Verweis auf eine Tabelle ist nicht vorhanden.
- L0094      SERIOUS CONSISTENCY ERROR: THE DB-KEYS IN THE OWNER-RECORD POINTING TO ITS SET DON'T MATCH (SET-REF = *setref*, SCD-DISPL = *displacement*)
- Bedeutung**  
In der SCD eines Owners eines leeren Sets ist entweder nur der DB-Key des ersten Membersatzes oder nur der DB-Key des letzten Membersatzes gleich dem eigenen DB-Key.
- L0095      SERIOUS CONSISTENCY ERROR: DB-KEY OF THE RECORD IS INVALID
- Bedeutung**  
Der DB-Key des SYSTEM-Records und der aus der SIA (Anchor) unterscheiden sich.
- L0098      STRUCTURAL CONSISTENCY ERROR: NUMBER OF ALLOWED ENTRIES = 0. THE CALC TABLE WILL NOT BE ANALYZED
- Bedeutung**  
Der Wert der Maximalanzahl von Einträgen in der Tabelle ist 0, d.h. die Tabelle kann nicht verwendet werden.

L0099 NOT CONSISTENT CALC HEADER: *calcheader*

– L0063 MINOR CONSISTENCY ERROR: NUMBER OF PRESENT ENTRIES = 0 IN CALC OVERFLOW PAGE

**Bedeutung**

Die Anzahl der Tabelleneinträge in einer CALC-Overflow-Seite ist gleich 0.

– L0096 SERIOUS CONSISTENCY ERROR: NUMBER OF ACTUAL ENTRIES *wert* DIFFERS FROM NUMBER OF PAGE INDEX ENTRIES *wert*

**Bedeutung**

Die Anzahl der vorhandenen Tabelleneinträge und die ausgegebene Anzahl der Seiten-Indizes in der direkten CALC-Seite sind unterschiedlich.

– L0122 STRUCTURAL CONSISTENCY ERROR: NUMBER OF ACTUAL ENTRIES *wert* IS TOO BIG

**Bedeutung**

Die Anzahl der vorhandenen Tabelleneinträge ist größer als die Anzahl der vorgesehenen Einträge.

– L0125 STRUCTURAL CONSISTENCY ERROR: PAGE NUMBERS OF SOME POINTERS ARE TOO BIG, MAXIMUM PAGE NUMBER IS *max*

**Bedeutung**

Der Vorwärts- oder Rückwärts-Zeiger im CALC-Tabellenkopf ist größer als der maximal zulässige Wert.

– L0158 SERIOUS CONSISTENCY ERROR: PRIMARY BUCKET *bnr* IN REALM *realmname* HAS A BACKWARD POINTER

**Bedeutung**

Der Rückwärts-Zeiger im CALC-Tabellenkopf in einem Primär-Bucket ist ungleich 0.

– L0163 SERIOUS CONSISTENCY ERROR: KEYLENGTH AND/OR NUMBER OF TABLE ENTRIES DO NOT CORRESPOND WITH LENGTH OF TABLE

**Bedeutung**

Die mittels ‚FREE SPACE.DISPL‘ aus der Seiten-Control-Information und Einträgen des CALC-Tabellenkopfs berechneten Tabellenlängen sind unterschiedlich.

L0100 SERIOUS CONSISTENCY ERROR: FPA VALUE OF {OLD/NEW} PAGE *act-key* IS IMPOSSIBLE

**Bedeutung**

Der FPA-Wert ist größer als die Seitenlänge minus der Länge der Seiten-Control-Information.

- L0101 MINOR CONSISTENCY ERROR: THE {OLD/NEW} FPA VALUE "*wert*" IS WRONG, PAGE HEADER CONTENTS *pageheader*
- Bedeutung**
1. Eine Liste belegt eine ganze Seite, hat jedoch einen Verschnitt.
  2. In der zentralen FPA-Verwaltung wird weniger Freiplatz angegeben als tatsächlich vorhanden ist.
  3. Der Verschnitt ist größer als ein Eintrag und zentraler und lokaler FPA-Wert sind unterschiedlich.
- Der Seitenkopf (20 Bytes) wird ausgegeben
- L0102 DUMP OF {OLD/NEW} PAGE CONTENTS:
- Bedeutung**  
Ausdruck des Seiteninhalts der „alten“ Schattendatenbank bzw. der „neuen“ Schattendatenbank oder des Originals, da dieser nicht weiter geprüft wird.
- L0103 STRUCTURAL CONSISTENCY ERROR: THE PAGE HEADER IS NOT CONSISTENT *erläuterung*
- WRONG ACT-KEY
- Bedeutung**  
Act-Key der Seite ist falsch.
- sonstige Meldungsergänzung
- Bedeutung**  
Fehler in den Bytes 5-20, der Name des Feldes erscheint in *erläuterung*.
- L0104 SERIOUS CONSISTENCY ERROR: THE PAGE INDEX IS NOT CONSISTENT *erläuterung*
- Bedeutung**  
Siehe unter „[L0105 STRUCTURAL CONSISTENCY ERROR: THE PAGE INDEX IS NOT CONSISTENT \*erläuterung\*“ auf Seite 127.](#)
- L0105 STRUCTURAL CONSISTENCY ERROR: THE PAGE INDEX IS NOT CONSISTENT *erläuterung*
- CALC-REC IN NON-CALC-PAGE
- Bedeutung**  
Das Status-Byte im Seitenindex in einer Seite außerhalb des CALC-Bereiches zeigt den Wert 3 an.
- DBTT-COL = 0 FOR TABLE-REC
- Bedeutung**  
Die DBTT-Column-Nr im Seitenindex eines Tabellensatzes zeigt den Wert 0 an.
- DISPL IMPOSSIBLE
- Bedeutung**  
Das Displacement im Seitenindex hat einen ungültigen Wert

– DISPL IMPOSSIBLE FOR TABLE REC

**Bedeutung**

Das Displacement im Seitenindex des Tabellensatzes ist größer als der maximal zulässige Wert

– IMPOSSIBLE LENGTH

**Bedeutung**

Unmögliche Länge des Satzes oder der Tabelle.

– MISSING BLOCK INDEX FOR EXPECTED LIST TABLE HEADER

**Bedeutung**

Es existiert ein List-Satz ohne zugehörigen List-Header

– NON-CALC-REC IN CALC PAGE

**Bedeutung**

Das Status-Byte im Seitenindex einer Seite aus dem CALC-Bereich zeigt einen Wert  $\neq 3$  an.

– REC-REFS FOR CALC RECORDS NOT EQUAL

**Bedeutung**

Die Rec-Refs in den Seitenindizes der CALC-Records sind unterschiedlich.

– RSQ IS 0

**Bedeutung**

Die RSQ im Seitenindex zeigt den Wert 0 an.

– STATUS-BYTE AND REC-REF DON'T MATCH

**Bedeutung**

Das Status-Byte im Seitenindex zeigt einen spanned record an, aber die Rec-Ref ist  $\neq 1$ .

– WRONG DBTT-COL FOR GIVEN TABLE-REC

**Bedeutung**

Die DBTT-Column-Nr eines SYSTEM-Records ist größer als der maximal zulässige Wert.

– WRONG DBTT-COL OR STATUS

**Bedeutung**

Das Status-Byte im Seitenindex zeigt einen Wert  $\neq 3$  an, aber die DBTT-Column-Nr zeigt einen Wert größer 0 an.

– WRONG REC-REF

**Bedeutung**

Die Rec-Ref im Seitenindex ist verfälscht oder größer als der maximal zulässige Wert.



– WRONG REC-REF FOR EXPECTED LIST RECORD

**Bedeutung**

Falsche Rec-Ref eines List-Satzes

– WRONG RSQ FOR GIVEN REC-REF

**Bedeutung**

Die RSQ-Nr eines SYSTEM-Records ist größer als der maximal zulässige Wert.

– WRONG STATUS-BYTE

**Bedeutung**

1. Das Status-Byte im Seitenindex ist nicht 0,1,2,3,4.
2. Das Status-Byte im Seitenindex eines SYSTEM-Records, der keine Tabelle ist, zeigt einen Wert  $\neq 0$  an.

– WRONG STATUS-BYTE FOR EXPECTED LIST-RECORD

**Bedeutung**

Zu wenige List-Sätze zum zugehörigen List-Header vorhanden

L0106 SERIOUS CONSISTENCY ERROR: THE TABLE APPEARS TWICE IN THE SAME PAGE

**Bedeutung**

Der DB-Key eines Tabellensatzes ist in mindestens zwei Seitenindizes vermerkt.

L0120 NOT CONSISTENT TABLE HEADER: *tableheader*

– L0121 STRUCTURAL CONSISTENCY ERROR: NUMBER OF TABLE ENTRIES DOES NOT CORRESPOND WITH LENGTH OF TABLE

**Bedeutung**

Die Tabellenlänge und die mittels der Anzahl der reservierten Einträge und Eintragslänge berechnete Tabellenlänge sind unterschiedlich.

– L0122 STRUCTURAL CONSISTENCY ERROR: NUMBER OF ACTUAL ENTRIES *wert* IS TOO BIG

**Bedeutung**

Die Anzahl der vorhandenen Tabelleneinträge ist größer als die Anzahl der vorgesehenen Einträge.

– L0123 STRUCTURAL CONSISTENCY ERROR: TABLE-DESCRIPTION-BYTE DIFFERS FROM SIA

**Bedeutung**

Der im Tabellenkopf und in der SIA vermerkte Tabellentyp sind unterschiedlich.

– L0124 STRUCTURAL CONSISTENCY ERROR: INDEX LEVEL FOR A NON-INDEXED TABLE

**Bedeutung**

Eine Tabelle, die keine Indextabelle ist, hat einen Level größer 0 im Tabellenkopf vermerkt.

- L0125 STRUCTURAL CONSISTENCY ERROR: PAGE NUMBERS OF SOME POINTERS ARE TOO BIG, MAXIMUM PAGE NUMBER IS *max*  
**Bedeutung**  
Im Tabellenkopf existiert mindestens ein Zeiger, der größer als der maximal zulässige Wert ist.
- L0126 SERIOUS CONSISTENCY ERROR: RSQ *rsq* IN TABLE ENTRY *te-nr* IS TOO BIG, MAXIMUM RSQ IS *max*  
**Bedeutung**  
Die RSQ im (CALC-) Tabelleneintrag ist größer als der maximal zulässige Wert.
- L0127 NOT CONSISTENT DUPLICATE TABLE HEADER: *dtableheader*
- L0128 STRUCTURAL CONSISTENCY ERROR: DUPLICATE TABLE HEADER IS OUT OF TABLE RANGE  
**Bedeutung**  
Der Duplicate-Table-Header liegt nicht im Speicherbereich der Tabelle.
- L0129 STRUCTURAL CONSISTENCY ERROR: FREE-TABLE-SPACE IS WRONG  
**Bedeutung**  
Der Wert FREE-TABLE.SPACE im Duplicate-Table-Header einer Overflow-Seite und der berechnete Freiplatz sind unterschiedlich.
- L0168 STRUCTURAL CONSISTENCY ERROR: POINTER TO PRIOR OVERFLOW PAGE IN LEVEL-0 TABLE NOT ALLOWED  
**Bedeutung**  
Der Rückwärts-Zeiger im Duplicate-Table-Header einer Primärseite ist ungleich 0.
- L0169 STRUCTURAL CONSISTENCY ERROR: POINTER TO PRIOR OVERFLOW PAGE IN DTOB MISSING  
**Bedeutung**  
Der Rückwärts-Zeiger in einem Duplicate-Table-Header einer Overflow-Seite ist 0.
- L0170 STRUCTURAL CONSISTENCY ERROR: POINTER TO NEXT OVERFLOW PAGE IN A PAGE WITH MORE THAN ONE DATABASE KEY LIST NOT ALLOWED  
**Bedeutung**  
Der Vorwärts-Zeiger im Duplicate-Table-Header ist ungleich 0, obwohl noch mindestens eine weitere Duplicate-Table in der Seite existiert.
- L0172 STRUCTURAL CONSISTENCY ERROR: FREE SPACE IN DTOB NOT ALLOWED BETWEEN PAGE INDEX AND DUPLICATE TABLE HEADER  
**Bedeutung**  
Der Duplicate-Table-Header einer Overflow-Seite liegt nicht unmittelbar hinter dem Seiten-Index.

L0130 STRUCTURAL CONSISTENCY ERROR: DB-KEY-LIST IN DUPLICATE TABLE FOR TABLE ENTRY *te-nr* IS OUT OF TABLE RANGE

**Bedeutung**

Der Duplicate-Table-Eintrag liegt nicht im Speicherbereich der Duplicate-Table.

L0132 SERIOUS CONSISTENCY ERROR: USER KEY OF TABLE ENTRY *te-nr* IS NOT IN SEQUENCE

**Bedeutung**

1. Die CALC-Schlüssel sind nicht aufsteigend sortiert.
2. In einer sortierten Tabelle liegt der Tabelleneintrag nicht in der richtigen Reihenfolge vor.

L0133 SERIOUS CONSISTENCY ERROR: TABLE WAS STORED INTO WRONG REALM

**Bedeutung**

Die Tabelle ist in einem Realm, der der SIA-Angabe widerspricht.

L0134 SERIOUS CONSISTENCY ERROR: REALM-REF OF PPP IN TABLE ENTRY *te-nr* IS WRONG

**Bedeutung**

Die Realm-Ref im (indirekten CALC-) Tabelleneintrag ist gleich 0 oder größer als der maximal zulässige Wert.

L0135 INCORRECT *act-key-0/act-key-N* OF REALM *realmname*:

- L0137 MINOR CONSISTENCY ERROR: REALMNAME *realmname* DIFFERS FROM SIA

**Bedeutung**

Der Realm-Name in der Act-Key-0-Seite oder der Act-Key-N-Seite und in der SIA stimmen nicht überein.

- L0138 SERIOUS CONSISTENCY ERROR: SYSTEM-BREAK HAD OCCURRED

**Bedeutung**

Das Systembreak-Bit in der Act-Key-0-Seite oder der Act-Key-N-Seite des DBDIR ist gesetzt.

- L0139 SERIOUS CONSISTENCY ERROR: FPA-VALUE FOR *act-key-0/act-key-N* CLAIMS NO FULL PAGE, VALUE = *fpa-wert*

**Bedeutung**

Der FPA-Wert für die Act-Key-0-Seite oder die Act-Key-N-Seite ist ungleich 0.

L0151 SERIOUS CONSISTENCY ERROR: USER KEY OF TABLE ENTRY *te-nr* IS A NOT ALLOWED DUPLICATE

**Bedeutung**

1. Der CALC-Schlüssel des Eintrags existiert bereits in der gleichen Seite und Duplikate sind nicht erlaubt.
2. In einer Tabelle, die keine Duplicate-Table ist, existiert ein Eintrag bereits, und Duplikate sind nicht erlaubt.

L0152 SERIOUS CONSISTENCY ERROR: ACT-KEY *act-key* IN DBTT ENTRY *dbttentry* COLUMN *column-nr* IS WRONG

**Bedeutung**

In einer Spalte des DBTT-Eintrags ist die Realm-Ref = 0 (Satz gelöscht), aber die Seitennummer größer als 1.

- L0153 DBTT ENTRY FOR NON EXISTING RECORD

**Bedeutung**

In einer Spalte größer 0 ist die Realm-Ref eines gelöschten Satzes ungleich 0.

- L0154 REALM-REF NOT ALLOWED IN SCHEMA

**Bedeutung**

Die Realm-Ref des DBTT-Eintrags ist größer als der maximal zulässige Wert.

- L0155 REALM-REF INVALID FOR RECORD OR TABLE

**Bedeutung**

Die Realm-Ref des DBTT-Eintrags identifiziert einen Realm, der nicht zur Speicherung freigegeben ist.

- L0156 PAGE NUMBER TOO BIG, MAXIMUM PAGE NUMBER IS *max*

**Bedeutung**

Die Seitennummer im DBTT-Eintrag ist größer als der maximal zulässige Wert.

L0159 SERIOUS CONSISTENCY ERROR: INDEX POINTER OF TABLE ENTRY *te-nr* DOES NOT POINT TO A PAGE INDEX

**Bedeutung**

Das Displacement zum Seitenindex im direkten CALC-Tabelleneintrag liegt nicht im Speicherbereich der Seitenindizes.

L0160 SERIOUS CONSISTENCY ERROR: USER CALC KEY OF TABLE ENTRY *te-nr* WAS STORED INTO WRONG BUCKET (CORRECT BUCKET: *bucket-nr*)

**Bedeutung**

Ein CALC-Record in einem Primär-Bucket oder 1. Overflow-Bucket ist im falschen Bucket abgespeichert.

L0162 SERIOUS CONSISTENCY ERROR: RSQ *rsq* OF TABLE ENTRY *te-nr* FOR EQUAL USER KEYS NOT IN SEQUENCE

**Bedeutung**

1. Bei gleichem CALC-Schlüssel sind die RSQs nicht aufsteigend sortiert.
2. In einer Tabelle, die keine Duplicate-Table ist, existiert die RSQ mehrfach.

L0171 SERIOUS CONSISTENCY ERROR: MAIN LEVEL POINTER IN DTOB *act-key* REFERS PAGE *seitennummer des main levels* WHICH IS NO LEVEL-0 DUPLICATE TABLE BLOCK TABLE PAGE WITH ONE INDEX ENTRY

**Bedeutung**

In dem Main-Level einer Duplicate-Table sind mindestens zwei Sätze gespeichert.

L0173 SERIOUS CONSISTENCY ERROR: RSQS OF DB-KEY-LIST IN DUPLICATE TABLE FOR TABLE ENTRY *te-nr* NOT IN SEQUENCE

**Bedeutung**

Die RSQs in der Duplicate-Table sind nicht aufsteigend sortiert.

L0176 SERIOUS CONSISTENCY ERROR: RSQ OF TABLE ENTRY *te-nr* DIFFERS FROM RSQ OF PAGE INDEX

**Bedeutung**

In einem direkten CALC-Bereich wird auf einen Seitenindex mit einer abweichenden RSQ verwiesen.

L0177 SERIOUS CONSISTENCY ERROR: BNR OF MAIN LEVEL POINTER IN DTOB *act-key* IS IMPOSSIBLE

**Bedeutung**

Die Seitennummer des Main-Levels einer Duplicate-Table-Overflow-Seite ist größer als der maximal mögliche Wert.

L0179 SERIOUS CONSISTENCY ERROR: THE {OLD/NEW} FPA VALUE "*wert*" IS WRONG, PAGE HEADER CONTENTS *pageheader*

**Bedeutung**

1. Der zentrale FPA-Wert für eine Act-Key-0-Seite, DBTT-Seite oder CALC-Seite ist ungleich 0.
2. Eine Liste belegt eine ganze Seite, der Freiplatz ist kleiner einem Eintrag, aber der Verschnitt ist größer 8 byte bei einer Seitenlänge von 2048 byte bzw. größer 12 byte bei einer Seitenlänge von 4000 bzw. 8096 byte.
3. Der zentrale FPA-Wert ist größer als der in der Seite berechnete Wert.

L0180 STRUCTURAL CONSISTENCY ERROR: DBTT ANCHOR PAGE *act-key* NOT CONSISTENT: VALUE "*content*" FROM *itemname* IS WRONG

**Bedeutung**

In einer DBTT-Ankerseite wurde ein Fehler entdeckt. Der falsche Inhalt des zugrundeliegenden Feldes wird zusammen mit dem Act-Key der DBTT-Ankerseite ausgegeben.

L0181 STRUCTURAL CONSISTENCY ERROR: DBTT ANCHOR PAGES NOT CONSISTENT: VALUE "*content*" FROM *itemname* IS WRONG, RECORD IS *recordname*

**Bedeutung**

Beim Prozessieren von DBTT-Ankerseiten wurde ein Fehler entdeckt. Der falsche Inhalt des zugrundeliegenden Feldes wird zusammen mit dem Satzartnamen ausgegeben.

L0182 STRUCTURAL CONSISTENCY ERROR: DBTT ANCHOR PAGES NOT CONSISTENT: THE DBTT PAGE(S) BETWEEN *pagenumber1* AND *pagenumber2* OF RECORD *recordname1* ARE NOT SEPARATED FROM THE DBTT PAGE(S) BETWEEN *pagenumber3* AND *pagenumber4* OF RECORD *recordname2*

**Bedeutung**

Nach dem Sortieren von in den DBTT-Ankerseiten verwalteten DBTT-Bereichen gemäß dem Act-Key der jeweils ersten DBTT-Seite wurde festgestellt, dass sich zwei DBTT-Bereiche überschneiden.

L0183 STRUCTURAL CONSISTENCY ERROR: DBTT ANCHOR PAGE *act-key* NOT CONSISTENT: VALUES "*content*" FROM *itemnames* DON'T MATCH

**Bedeutung**

In einer DBTT-Ankerseite wurde ein Fehler entdeckt. Der nicht zusammenpassende Inhalt der zwei zugrundeliegenden Felder wird zusammen mit dem Act-Key der DBTT-Ankerseite ausgegeben.

L0184 STRUCTURAL CONSISTENCY ERROR: DBTT ANCHOR PAGES NOT CONSISTENT: VALUES "*content*" FROM *itemnames* DON'T MATCH, RECORD IS *recordname*

**Bedeutung**

Beim Prozessieren von DBTT-Ankerseiten wurde ein Fehler entdeckt. Der nicht zusammenpassende Inhalt der zwei zugrundeliegenden Felder wird zusammen mit dem Satzartnamen ausgegeben.

L0187 SERIOUS CONSISTENCY ERROR: TABLE HEADER NOT CONSISTENT: {ACTKEY STRUCTURE/TOP-OR-FIRST-INDICATOR} NOT ALLOWED

**Bedeutung**

Table-Header im ACTKEY-Format/mit der Top-or-First-Anzeige ist nicht erlaubt.

L0188 SERIOUS CONSISTENCY ERROR: TABLE HEADER NOT CONSISTENT: LEVEL\_BACK\_KEY = 0, BUT TOP-OR-FIRST-INDICATOR IS NOT ON

**Bedeutung**

Wenn im ACTKEY-Format Level-Back = 0 ist, muss die Top-or-First-Anzeige gesetzt sein

L0189 SERIOUS CONSISTENCY ERROR: TABLE HEADER NOT CONSISTENT: TOP-OR-FIRST-INDICATOR IS ON, BUT FORWARD-KEY <> 0 OR LEVEL-BACK-KEY <> 0

**Bedeutung**

Wenn die Top-or-First-Anzeige gesetzt ist, muss der Vorwärts-Verweis und Level-Back = 0 sein

L0190      SERIOUS CONSISTENCY ERROR: TABLE HEADER NOT CONSISTENT: INVALID AREA-REF IN  
{FORWARD/BACKWARD}-KEY

**Bedeutung**

Im ACTKEY des Vorwärts- bzw. Rückwärts-Verweises ist eine AREA-REF eingetragen, die grösser als die maximale AREA-REF der Datenbank ist.

L0206      SERIOUS CONSISTENCY ERROR: COMPRESSION ENTRY *c-e* DOES NOT MATCH WITH RECORD-  
LENGTH

**Bedeutung**

Die im Compression-Eintrag des Satzes vermerkten Displacements widersprechen der Satzlänge.

### 3.7.5 Verwendung von Auftragsschaltern

BCHECK setzt beim Auftreten von Fehlern Auftragsschalter (mittels BS2000-Kommando bzw. Makro MODIFY-JOB-SWITCHES). Sie können diese Auftragsschalter in Ihren Prozeduren nutzen, sollten aber nur für den Fall einen Steuerungsautomatismus vorsehen, dass kein Auftragsschalter gesetzt ist. Jeder andere Fall erfordert eine Sichtung des SYSLST-Protokolls.

Die folgende Tabelle beschreibt, welche Auftragsschalter BCHECK setzt und welche Maßnahmen erforderlich sind:

Art des Fehlers	Auftragsschalter	Maßnahme
WARNING	30	siehe Handbuch „ <a href="#">Meldungen</a> “
ERROR	31	siehe Handbuch „ <a href="#">Meldungen</a> “
SYSTEM-ERROR	31	siehe Handbuch „ <a href="#">Meldungen</a> “
EASY (MINOR) LOCAL CONSISTENCY ERROR	26	Fehlerunterlagen an Diagnose schicken; Datenbankbetrieb ist ohne Einschränkung möglich
FATAL (SERIOUS, STRUCTURAL) LOCAL CONSISTENCY ERROR	27	Fehlerunterlagen an Diagnose schicken; Datenbankbetrieb ist je nach Art des Konsistenzfehlers nach Rücksprache mit Diagnose ggf. möglich
GLOBAL CONSISTENCY ERROR	28	Fehlerunterlagen an Diagnose schicken; Datenbankbetrieb ist je nach Art des Konsistenzfehlers nach Rücksprache mit Diagnose ggf. möglich

Tabelle 11: Verwendung von Auftragsschaltern bei BCHECK



---

## 4 Schema/Subschema Information Area ausdrucken mit BPSIA

BPSIA druckt das Schema (SIA-Protokoll) bzw. ein Subschema (SSIA-Protokoll) in tabellarischer Form aus.

Das SIA-Protokoll kann für Sie z.B. nützlich sein:

- beim Einspeichern von Sätzen in die Datenbank mit BINILOAD,
- beim Ausdrucken bestimmter Übersichten mit BSTATUS oder
- beim Ausdrucken des Datenbankinhalts mit BPRECORD

Ein SSIA-Protokoll hingegen stellt für den Programmierer der DB-Anwendungen eine hilfreiche Unterlage dar.

Mit BPSIA können Sie sich ausdrucken lassen:

- das Benutzerschema und die Benutzersubschemaschemata
- das Compiler-Schema und das Compiler-Subschema
- das PRIVACY-AND-IQF-Schema und das PRIVACY-AND-IQF-Subschema

BPSIA liest zum Ausdrucken eines Schemas die entsprechende Schema-Information-Area (SIA) und zum Ausdrucken eines Subschemas die entsprechende Subschema-Information-Area (SSIA) aus dem DBDIR und protokolliert diese tabellarisch aufbereitet.

Einen Ausdruck der Benutzer-SIA bzw. einer Benutzer-SSIA erhalten Sie in gleicher Form auch, wenn Sie beim Erzeugen der SIA mit dem Dienstprogramm BGSIA bzw. beim Erzeugen der SSIA mit dem Dienstprogramm BGSSIA per DISPLAY-Anweisung den entsprechenden Ausdruck anfordern.

Zusätzlich zur Ausgabe auf SYSLST können Sie die Daten auch im CSV-Format in eine Datei ausgeben lassen. Mit dem CSV-Format wird die Weiterverarbeitung von Daten in anderen Systemumgebungen (z.B. bei Spreadsheet-Anwendungen) erleichtert. Die Ausgabe im CSV-Format ist im Handbuch „[Datenbankbetrieb](#)“ im Abschnitt „Datenbankinformationen systemneutral ausgeben“ beschrieben.

## 4.1 Systemumgebung

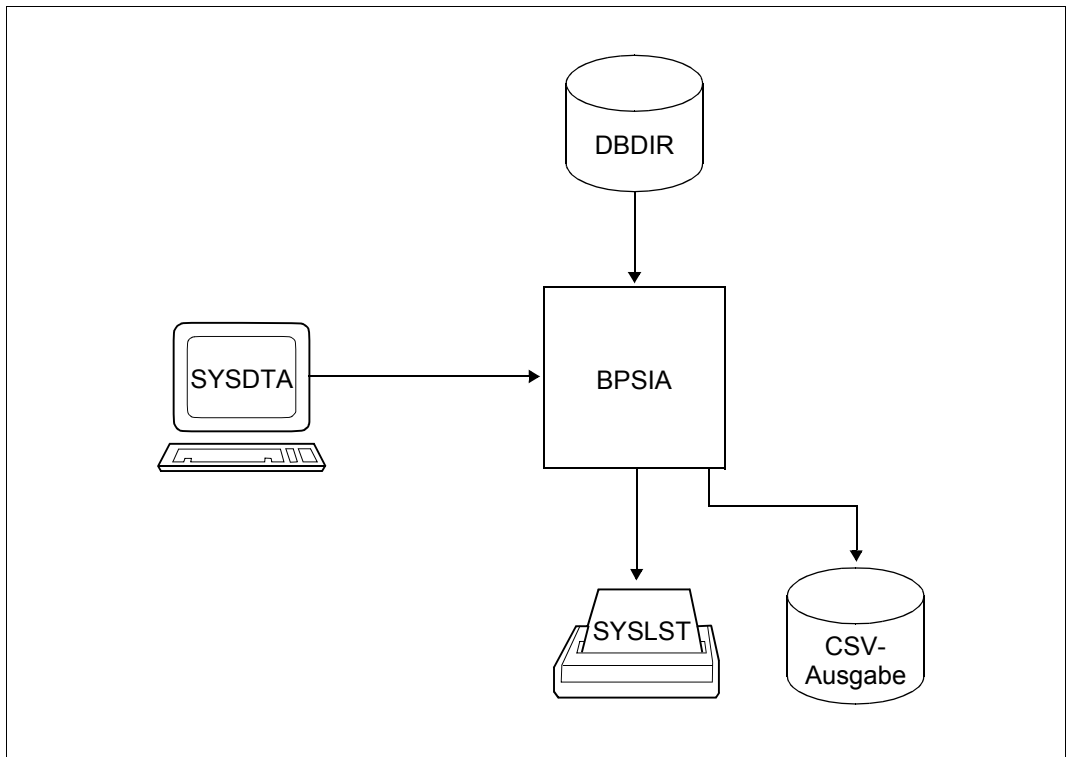


Bild 9: Systemumgebung von BPSIA

**i** BPSIA kann auch online gestartet werden, das DBDIR ist dann als inkonsistent gekennzeichnet. Es wird die Warnung ausgegeben, dass das SYSTEM-BREAK-BIT im DBDIR gesetzt ist.

Wenn Sie BPSIA online ausführen, müssen Sie damit rechnen, dass die ausgegebenen Daten nicht aktuell sind, da der DBH noch nicht alle Daten aus dem Puffer in die Datenbank geschrieben hat. Um weitgehend aktuelle Daten zu erhalten, sollten Sie unmittelbar vor dem BPSIA-Lauf mit dem DAL-Kommando CHECKPOINT oder NEW RLOG eine Aktualisierung der Datenbank erzwingen. Aber auch danach sind noch Abweichungen der BPSIA-Ausgabe vom tatsächlichen Stand der Datenbank möglich, falls parallel ein Update durchgeführt wird.

BPSIA berücksichtigt beim Start ggf. eine zugewiesene UDS/SQL-Pubset-Deklaration (siehe Handbuch „[Datenbankbetrieb](#)“, Pubset-Deklarations-Jobvariable). Eine fehlerhafte Zuweisung führt zum Programmabbruch.

## 4.2 Anweisungen für BPSIA

Anweisung	Bedeutung
DISPLAY SCHEMA	ein Schema ausdrucken
DISPLAY SUBSCHEMA	ein Subschema ausdrucken
END	Eingabe der Anweisungen beenden

Tabelle 12: Anweisungen für BPSIA

Beide DISPLAY-Anweisungen sind wahlweise anzugeben. Sie dürfen sie auch mehrfach angeben.

Die Anweisungen können über mehrere Zeilen gehen. Jede Zeile ist aber auf 72 Stellen beschränkt. Ein Fortsetzungszeichen ist bei mehrzeiliger Schreibweise nicht erforderlich.

### Schema ausdrucken (DISPLAY SCHEMA)

---

```
[DISPLAY [IN CSV [csv-dateiname]] SCHEMA schemaname]
```

---

#### IN CSV

BPSIA gibt die Daten zusätzlich im CSV-Format aus.

##### *csv-dateiname*

Name der Datei, in die die Daten im CSV-Format ausgegeben werden sollen. Die Angabe von *csv-dateiname* ist bei der ersten IN CSV-Anweisung einer BPSIA-Ausführung zwingend (z.B. DISPLAY IN CSV 'BPSIA.CSV' ...).

Eine ausführliche Beschreibung der Ausgabe im CSV-Format finden Sie im Handbuch „[Datenbankbetrieb](#)“ im Abschnitt „Datenbankinformationen systemneutral ausgeben“.

##### *schemaname*

Name des Schemas, dessen SIA auszudrucken ist; Sie können angeben:

*benutzerschemaname*

COMPILER-SCHEMA

PRIVACY-AND-IQF-SCHEMA

## Subschema ausdrucken (DISPLAY SUBSCHEMA)

---

[DISPLAY [IN CSV [*csv-dateiname*]] SUBSCHEMA *subschemaname*]

---

### IN CSV

BPSIA gibt die Daten zusätzlich im CSV-Format aus.

*csv-dateiname*

Name der Datei, in die die Daten im CSV-Format ausgegeben werden sollen. Die Angabe von *csv-dateiname* ist bei der ersten IN CSV-Anweisung einer BPSIA-Ausführung zwingend (z.B. DISPLAY IN CSV 'BPSIA.CSV' ...).

Eine ausführliche Beschreibung der Ausgabe im CSV-Format finden Sie im Handbuch „[Datenbankbetrieb](#)“ im Abschnitt „Datenbankinformationen systemneutral ausgeben“.

*subschemaname*

Name des Subschemas, dessen SSIA auszudrucken ist; Sie können angeben:

*benutzersubschemaname*

COMPILER-SUBSCHEMA

PRIVACY-AND-IQF-SS

## Eingabe der Anweisungen beenden (END)

---

END

---

### 4.3 Kommandofolge zum Starten von BPSIA

Die hier beschriebene Kommandofolge geht davon aus, dass UDS/SQL mit IMON installiert wurde (siehe Abschnitt „START-Kommandos der UDS/SQL-Programme“ in Kapitel 2 des Handbuchs „[Aufbauen und Umstrukturieren](#)“).

```
01 /ADD-FILE-LINK LINK-NAME=DATABASE,  
    FILE-NAME=[ :catid: ] [ $userid. ] dbname.DBDIR [ .copyname ]  
02 /SELECT-PRODUCT-VERSION PRODUCT-NAME=UDS-SQL, VERSION=version, SCOPE=*TASK  
03 /START-UDS-BPSIA  
04 bpsia-anweisungen  
05 END
```

- 01 Hier ist die Angabe von *:catid:* zulässig (siehe Handbuch „[Datenbankbetrieb](#)“).
- 02 Die Version des Dienstprogramms wird ausgewählt.  
Die Angabe der Version wird generell empfohlen, da mehrere UDS/SQL-Versionen parallel installiert sein können.
- 03 Das UDS/SQL-Dienstprogramm kann auch mit dem Aliasnamen BPSIA gestartet werden.
- 04 Ein Punkt wird als Endekriterium betrachtet. Hinter dem Punkt kann wieder eine Anweisung stehen.

## 4.4 Beschreibung des SIA-Protokolls

Das SIA-Protokoll stellt einen tabellarisch aufbereiteten Abdruck einer SIA dar. Sie können ihm die wichtigsten Informationen über ein Schema entnehmen.

### SIA PRINT REPORT (allgemeine Informationen)

#### *Beispiel*

```
*** SIA PRINT REPORT ***  
  
DATABASE ID           = XXXXXXXX  
DATABASE NAME        = VERSAND  
DATABASE-LAYOUT-VERSION = 004.00  
SCHEMA NAME          = ARTIKELVERSAND  
SCHEMA TYPE          = USER SCHEMA  
SIA VALIDATION DATE  = 2015-06-28 11:26:15  
LENGTH OF SIA        = 6588  
MAXIMUM AREA REF     = 12  
MAXIMUM RECORD REF   = 15  
MAXIMUM SET REF      = 34  
MAXIMUM KEY REF      = 19  
TCUA LENGTH          = 3424  
MAXIMUM RECORD LENGTH = 580  
MAXIMUM ENTRY LENGTH = 52  
MAXIMUM NR MEMBERSHIPS = 4  
MAXIMUM SPLIT PARAMETER = 5  
LENGTH KEY-BIT-STRING = 0  
LENGTH PHYSICAL BLOCK = 4000  
FPA-ENTRIES MAIN-BLOCK = 1990
```

Unter der Überschrift SIA PRINT REPORT druckt BPSIA folgende allgemeinen Informationen aus:

**DATABASE ID**

Kennung, in der die Datenbank gespeichert ist

**DATABASE NAME**

Name der Datenbank

**DATABASE-LAYOUT-VERSION**

Layout-Version der Datenbank

**SCHEMA NAME**

Name des Schemas

**SCHEMA TYPE**

Art des Schemas

**SIA VALIDATION DATE**

Validierungsdatum des Schemas mit Datum und Uhrzeit

**LENGTH OF SIA**

Länge der SIA

**MAXIMUM AREA REF**

höchste Nummer eines Realm Ihrer Datenbank;  
sie stimmt nach einer Umstrukturierung nicht unbedingt mit der Anzahl der Realms überein

**MAXIMUM RECORD REF**

höchste Nummer einer Satzart Ihrer Datenbank;  
sie stimmt nach einer Umstrukturierung nicht unbedingt mit der Anzahl der Satzarten überein

**MAXIMUM SET REF**

höchste Nummer eines Sets Ihrer Datenbank;  
sie stimmt nach einer Umstrukturierung nicht unbedingt mit der Anzahl der Sets überein

**MAXIMUM KEY REF**

höchste Nummer eines ASC-Keys bzw. DESC-Keys oder SEARCH-Keys Ihrer Datenbank;  
sie stimmt nach einer Umstrukturierung nicht unbedingt mit der Anzahl der entsprechenden Schlüssel überein

**TCUA LENGTH**

Länge der Transaction Currency Area eines Subschemas

**MAXIMUM RECORD LENGTH**

Länge der längsten Satzart des Schemas einschließlich der Set-Connection-Data (SCD)

**MAXIMUM ENTRY LENGTH**

Länge des größten Schlüssels des Schemas (CALC-Key, ASC-Key bzw. DESC-Key oder SEARCH-Key), erhöht um 7 byte und aufgerundet auf ein ganzzahliges Vielfaches von vier

**MAXIMUM NR MEMBERSHIPS**

die größte Anzahl Sets des Schemas, in denen die gleiche Satzart als Membersatzart enthalten ist

**MAXIMUM SPLIT PARAMETER**

höchster angegebener Wert in der DYNAMIC REORGANIZATION-Klausel der SSL

**LENGTH KEY-BIT-STRING**

Kontroll-Information für die MODIFY-Funktion; bei Angabe 0 werden 4 byte reserviert

**LENGTH PHYSICAL BLOCK**

Länge der Datenbankseiten

**FPA-ENTRIES MAIN-BLOCK**

Anzahl möglicher FPA-Einträge pro FPA-Basis-Seite



## REFERENCE NUMBERS

*Beispiel*

```

*** REFERENCE NUMBERS ***

11 AREAS : 1 3 4 5 6 7 8 9 10 11
           12

15 RECORDS : 1 2 3 4 5 6 7 8 9 10
             11 12 13 14 15

34 SETS : 1 2 3 4 5 6 7 8 9 10
          11 12 13 14 15 16 17 18 19 20
          21 22 23 24 25 26 27 28 29 30
          31 32 33 34

19 KEYS : 1 2 3 4 5 6 7 8 9 10
          11 12 13 14 15 16 17 18 19

```

Unter der Überschrift REFERENCE NUMBERS druckt BPSIA aus, wie viele Realms, Satzarten, Sets und Schlüssel in dem Schema definiert sind, und listet die jeweiligen Nummern auf.

Die ausgegebenen Nummern entsprechen dem aktuellen Stand der Datenbank. Sie müssen nicht unbedingt den unter der Überschrift SIA PRINT REPORT ausgedruckten Maximalwert enthalten (siehe [Seite 142](#)).

### AREAS

- 1: Database Directory
- 2: Database Compiler Realm
- 3,...: Benutzerrealms

Der Database Compiler Realm gehört zur Compiler-Datenbank; daher wird seine Nummer beim Ausdrucken einer Benutzer-SIA nicht mit aufgelistet.

### RECORDS

- 1: SSIA-RECORD
- 2,...: Benutzersatzarten

## AREA INFORMATION

### Beispiel

\*\*\* AREA INFORMATION \*\*\*

REF	AREA-NAME	TEMP	D/T	FPA-BEGIN <sup>1</sup> REF	BNR	ENTRIES	EXTENTS	INCR -ACT	CURRENT BNR	FREE NR-PAGES	FULL NR-PAGES	PART-FILLED NR-PAGES	FIL%	SCAN	REUSE
1	DATABASE-DIRECTORY	D		1	1	100	0	YES	48	51	42	7	76	1	
3	AUFTRAGSRML	D		3	1	38	0	YES	13	24	13	1	2	1	
4	BESTELLRLM	D		4	1	60	0	YES	24	33	26	1	2	1	
5	KLEIDUNG	D		5	1	54	0	YES	20	32	10	12	65	1	
6	HAUSHALT	D		6	1	24	0	YES	7	16	8	0	-	1	
7	SPORT	D		7	1	46	0	YES	8	37	9	0	-	1	
8	LEBENSMITTEL	D		8	1	18	0	YES	8	8	5	5	92	1	
9	SPIELE-HOBBY	D		9	1	44	0	YES	8	35	9	0	-	1	
10	SCHREIBWAREN	D		10	1	24	0	YES	6	17	7	0	-	1	
11	ARTIKELRLM	D		11	1	62	0	YES	46	14	44	4	9	1	
12	SUCHRLM	*	D						0	0				1	

Unter der Überschrift AREA INFORMATION druckt BPSIA Informationen über die Realms der Datenbank aus:

REF Nummer eines Realm

AREA-NAME

Name des Realm

TEMP Kennzeichnung, ob es sich um einen Temporären Realm handelt

\*: Temporärer Realm

D/T Kennzeichnung, ob der Realm auf Platte oder Band liegt,

D: Platte (Disc)

T: Band (Tape)

FPA-BEGIN

Adresse (Realm-Nummer (REF) und Seitennummer (BNR)) der ersten Seite der Freiplatzverwaltung in diesem Realm

ENTRIES

Gesamtanzahl der Datenbankseiten dieses Realm

EXTENTS

Anzahl der FPA-Extents dieses Realms

<sup>1</sup> Die halbfett gedruckten Felder enthalten vor Ablauf des Dienstprogramms BFORMAT noch den Wert ,0'.

**INCR-ACT**

Kennzeichnung der Online-Erweiterbarkeit dieses Realms

YES: Für den Realm ist die Online-Erweiterbarkeit aktiviert.

NO: Für den Realm ist die Online-Erweiterbarkeit nicht aktiviert.

Die zuverlässige Information über die Online-Erweiterbarkeit ist im Act-key-0 des jeweiligen Realms abgelegt und kann mit der BPRECORD-Anweisung DISPLAY PAGE ZERO ausgegeben werden. In der SIA ist die Information nur zuverlässig verfügbar, wenn der DBH das letzte ändernde Programm auf der Datenbank war.

**CURRENT BNR**

Nummer der Seite, in der zuletzt freier Speicherplatz gefunden wurde

0: auf den Realm wurde bisher nicht zugegriffen

**FREE NR-PAGES**

Anzahl völlig freier Seiten

**FULL NR-PAGES**

Anzahl voller Seiten

**PART-FILLED NR-PAGES**

Anzahl teilweise gefüllter Seiten mit durchschnittlichem Füllgrad FILL%

**SCAN** gibt an, welcher Such-Modus für die Freiplatzsuche verwendet wurde

1: First Scan (Platz für neu einzuspeichernde Daten wird im freien Bereich am Ende des Realms gesucht).

2: Second Scan (Platz für neu einzuspeichernde Daten wird ab dem Beginn des Realms gesucht).

**REUSE**

Kennzeichen, ob für diesen Realm die BMODTT-Anweisung SET REUSE-FREE-SPACE gegeben worden ist

\_ nicht angegeben

\* angegeben



Beim Ausgeben eines Benutzer-Schemas oder Compiler-Schemas ist es möglich, dass die Werte FPA-BEGIN und ENTRIES für den DBDIR (Zeile 1) nicht aktuell sind. Das kann z.B. nach einem BREORG-Lauf der Fall sein.

**RECORD WITHIN LIST***Beispiel*

\*\*\* RECORD WITHIN LIST \*\*\*

REF	AREA-NAME	NR-WITHIN	LIST OF RECORDS			
1	DATABASE-DIRECTORY	1	1			
3	AUFTRAGSRLM	4	2	3	4	5
4	BESTELLRLM	3	13	14	15	
5	KLEIDUNG	4	6	7	8	9
6	HAUSHALT	5	6	7	8	9 10
7	SPORT	5	6	7	8	9 10
8	LEBENSMITTEL	4	6	7	8	9
9	SPIELE-HOBBY	4	6	7	8	9
10	SCHREIBWAREN	4	6	7	8	9
11	ARTIKELRLM	2	11	12		
12	SUCHRLM	0				

Unter der Überschrift RECORD WITHIN LIST druckt BPSIA aus, welche Satzarten in den einzelnen Realms liegen:

REF Nummer eines Realm

AREA-NAME

Name des Realm

NR-WITHIN

Anzahl der Satzarten, die in dem Realm gespeichert sind

LIST OF RECORDS

Nummern der Satzarten

## RECORD INFORMATION

*Beispiel*

\*\*\* RECORD INFORMATION \*\*\*

REF	RECORD-NAME	LOC-MODE	LENGTH	SYSINFO	COMPR	IMPL-SET	LIST-SET	LOC-VIA	OPT-CLAIM
1	SSIA-RECORD		0	0					0
2	KUNDE	DIR-LG	116	48					588
3	AUFTRAG		17	6				1	0
4	AUFTR-POS		54	46					0
5	RATENZAHLUNG		44	18		28			0
6	ARTIKELART		29	4		29			0
7	ARTIKELAUSWAHL		25	0		30			0
8	ARTIKELBESCHR	CALC	580	20	V				0
9	ARTIKEL	CALC	215	128		31	7	7	0
10	TEILMENGE		58	56					0
11	FARBEN		22	0		32			0
12	MATERIALIEN		21	0		33			0
13	LIEFERANT	CALC	167	37					0
14	BESTELLUNG		50	40					0
15	BESTELL-POS		36	28			17	17	0

Unter der Überschrift RECORD INFORMATION druckt BPSIA eine Tabelle aus mit den wichtigsten Informationen über die einzelnen Satzarten:

REF Nummer einer Satzart

RECORD-NAME  
Name der Satzart

LOC-MODE

Art der LOCATION MODE-Klausel der DDL

CALC: LOCATION MODE IS CALC wurde definiert

DIRECT: LOCATION MODE IS DIRECT wurde definiert

DIR-LG: LOCATION MODE IS DIRECT-LONG wurde definiert

LENGTH

Gesamtlänge der Satzart einschließlich der Set-Connection-Data (SCD)

SYSINFO

Länge der Set-Connection-Data (in byte)

COMPR

Kennzeichnung, ob die Sätze der Satzart komprimiert gespeichert werden

\*: komprimierte Speicherung

V: die Satzart enthält ein variables Feld

**IMPL-SET**

Nummer des impliziten Sets dieser Satzart, wenn ein SEARCH-Key auf Satzartebene definiert wurde

**LIST-SET**

Nummer eines mit MODE IS LIST definierten Sets, in dem die Satzart als Member-satzart enthalten ist

**LOC-VIA**

Nummer eines Sets, für den MODE IS LIST oder PLACEMENT OPTIMIZATION definiert wurde und in dem die Satzart als Membersatzart enthalten ist

**OPT-CLAIM**

Anzahl der Bytes, die beim Speichern eines Satzes dieser Satzart reserviert werden (nur von Bedeutung, wenn die vorliegende Satzart als Ownersatzart in einem Set enthalten ist, für den PLACEMENT OPTIMIZATION oder ATTACHED TO OWNER definiert wurde) .

Der Wert gibt die Anzahl Bytes an, die zum Abspeichern des Owners, der erwarteten Member sowie der Verwaltungsinformationen und Tabellen erforderlich sind.

## DBTT INFORMATION

Unter der Überschrift DBTT INFORMATION druckt BPSIA zwei Tabellen aus:

- Die erste Tabelle enthält Informationen über die Database Key Translation Tables (DBTTs) der einzelnen Satzarten des Schemas.
- Die zweite Tabelle listet die Sets auf, in denen die Satzarten des Schemas Member-satzart bzw. Ownersatzart sind.

### Beispiel

\*\*\* DBTT INFORMATION \*\*\*

REF	RECORD-NAME	ANCHOR		CURRENT	NR	EXTENTS	LENGTH	PER-BLOCK	NO REUSE
1	SSIA-RECORD	1	2 <sup>1</sup>	11	995	0	4	995	
2	KUNDE	3	2	1	331	0	12	331	
3	AUFTRAG	3	4	1	497	0	8	497	
4	AUFTR-POS	3	6	1	1990	0	4	995	
5	RATENZAHLUNG	3	9	1	995	0	4	995	
6	ARTIKELART	11	2	4	497	0	8	497	
7	ARTIKELAUSSWAHL	11	4	5	497	0	8	497	
8	ARTIKELBESCHR	11	6	13	497	0	8	497	
9	ARTIKEL	11	8	63	995	0	4	995	
10	TEILMENGE	6	5	1	995	0	4	995	
11	FARBEN	11	10	25	995	0	4	995	
12	MATERIALIEN	11	12	10	995	0	4	99	
13	LIEFERANT	4	2	1	662	0	12	331	
14	BESTELLUNG	4	18	1	497	0	8	497	
15	BESTELL-POS	4	20	1	995	0	4	995	

REF	RECORD-NAME	OWNERSHIPS				MEMBERSHIPS			
2	KUNDE	1	3	4					
3	AUFTRAG	2				1			
4	AUFTR-POS					2	3	13	
5	RATENZAHLUNG					28	4		
6	ARTIKELART	5				29			
7	ARTIKELAUSSWAHL	6				30			
8	ARTIKELBESCHR	7				5	6		
9	ARTIKEL	9	10	13	14	31	7	8	12
10	TEILMENGE					9	10		
11	FARBEN					32			
12	MATERIALIEN					33			
13	LIEFERANT	12	15	16		11			
14	BESTELLUNG	17				15	16		
15	BESTELL-POS					14	17		

<sup>1</sup> Die halbfett gedruckten Felder enthalten vor Ablauf des Dienstprogramms BFORMAT noch den Wert ‚0‘.

REF Nummer einer Satzart;

RECORD-NAME  
Name der Satzart

ANCHOR  
Erste Seite der DBTT-Ankertabelle (Realm-Nummer und Seitennummer)

CURRENT  
Satzfolgennummer des zuletzt gespeicherten Satzes dieser Satzart

NR Anzahl der Sätze, die, auf Grund der aktuellen DBTT-Größe, von dieser Satzart maximal gespeichert werden können

EXTENTS  
Anzahl der momentan existierenden DBTT-Extents

LENGTH  
Länge eines DBTT-Eintrags

PER-BLOCK  
Anzahl der DBTT-Einträge, die auf einer Seite Platz haben

NO REUSE  
\*: Database Keys gelöschter Sätze können nicht wieder verwendet werden.  
┆ Database Keys gelöschter Sätze können wieder verwendet werden.

OWNERSHIPS  
Nummern der Sets, in denen die Satzart Owner ist

MEMBERSHIPS  
Nummern der Sets, in denen die Satzart Member ist



Beim Ausgeben eines Benutzer-Schemas oder Compiler-Schemas ist es möglich, dass die Werte BEGIN, NR und LENGTH nicht aktuell sind. Das kann z.B. nach einem BREORG-Lauf der Fall sein.



## CALC INFORMATION

### Beispiel

\*\*\* CALC INFORMATION \*\*\*

REC				PPP-BITS	PHYSICAL_CALC_INFO						
REF	RECORD-NAME	DIR	DUPL	LENGTH	HASH-REF	O	M	TAB	FIRST-BUCKET	NR-BUCKETS	
8	ARTIKELBESCHR		*	40					5	<b>2</b> <sup>1</sup>	3
									6	<b>2</b>	2
									7	<b>2</b>	3
									8	<b>2</b>	1
									9	<b>2</b>	3
									10	<b>2</b>	2
9	ARTIKEL			10					5	<b>5</b>	3
									6	<b>4</b>	1
									7	<b>5</b>	3
									8	<b>3</b>	1
									9	<b>5</b>	3
									10	<b>4</b>	2
13	LIEFERANT		*	35					4	<b>5</b>	13

Unter der Überschrift CALC INFORMATION druckt BPSIA Informationen aus über die mit LOCATION MODE IS CALC definierten Satzarten:

#### REC REF

Nummer einer Satzart

#### RECORD-NAME

Name der Satzart

DIR Kennzeichnung, ob ein direkter Hashbereich oder indirekter Hashbereich angelegt wurde

\*: direkter Hashbereich

DUPL Kennzeichnung, ob Schlüsselwerte mehrfach vorkommen dürfen oder nicht

\*: Duplikate zugelassen

#### LENGTH

Gesamtlänge des CALC-Keys

#### HASH-REF

Kennzeichnung einer benutzereigenen Hashroutine. Wenn keine Kennzeichnung ausgedruckt ist, wird die Standard-Hashroutine von UDS/SQL verwendet

<sup>1</sup> Die halbfett gedruckten Felder enthalten vor Ablauf des Dienstprogramms BFORMAT noch den Wert ,0'.

## PPP-BITS

\* bedeutet bei

- O durch Verlagerung von Ownersätzen z.B. mit BREORG sind Probable Position Pointer (PPP), die auf diese Ownersätze verweisen, mit hoher Wahrscheinlichkeit nicht mehr aktuell
- M durch Verlagerung von Membersätzen z.B. mit BREORG sind Probable Position Pointer, die auf diese Membersätze verweisen, mit hoher Wahrscheinlichkeit nicht mehr aktuell
- TAB durch Verlagerung von Tabellen z.B. mit BREORG sind Probable Position Pointer, die auf diese Tabellen verweisen, mit hoher Wahrscheinlichkeit nicht mehr aktuell

Probable Position Pointer können auch vorliegen, ohne dass diese Anzeigen gesetzt sind, weil der DBH bei Verlagerung einzelner Sätze diese Bits nicht pflegt. Die Anzeigen werden in der Regel gesetzt, wenn durch Dienstprogramme alle Sätze verlagert wurden.

Ein BREORG-Lauf ist eventuell sinnvoll.

## PHYSICAL CALC INFO

physische Informationen über den Hashbereich:

## FIRST-BUCKET

Adresse (Realm-Nummer und Seitennummer) der ersten CALC-Seite des Primärbereichs

## NR-BUCKETS

Anzahl der für den Primärbereich reservierten CALC-Seiten

Die Informationen unter der Überschrift PHYSICAL CALC INFO wiederholt BPSIA für jeden Realm, in dem ein Hashbereich der Satzart liegt.

## SET INFORMATION

Unter der Überschrift SET INFORMATION druckt BPSIA zwei Tabellen aus mit Informationen über die im Schema definierten Sets.

### Beispiel

\*\*\* SET INFORMATION \*\*\*

REF	SET-NAME	TYPE	MODE	ORDER	INSERT	REMOVE	SOS	INIT	INCR	T-AREA	P-AREA
1	ERTEILTE-AUFTRAEGE		PTRAY	SORT	AUTO	TRAN	OWN	10	1		
2	AUFTR-INHALT		PTRAY	SORT	AUTO	PERM	CUR	0	1		
3	NOCH-ZU-BEZAHLN		CHAIN	LAST	AUTO	TRAN	OWN	0	1		
4	RATENKAUF		CHAIN	LAST	AUTO	PERM	OWN	0	1		
5	ANGEBOT		PTRAY	SORT	AUTO	PERM	CUR	100	5		
6	NAEHERE-AUSWAHL		PTRAY	SORT	AUTO	PERM	CUR	100	20		
7	BESTELLANGABEN		LIST	SORT	AUTO	PERM	CUR	15	1		
8	MIN-BESTAND-ERREICHT	SING	CH-PR	SORT	MANL	TRAN		0	1		
9	ENTHAELT		CHAIN	NEXT	AUTO	PERM	OWN	10	1		
10	ENTHALTEN-IN		CH-PR	NEXT	AUTO	PERM	OWN	0	1		
11	LIEFERANTEN	SING	PTRAY	SORT	AUTO	PERM		0	1		
12	LIEFERBARE-ARTIKEL		PTRAY	SORT	AUTO	PERM	CUR	500	1		
13	BESTELLTE-ARTIKEL		CHAIN	LAST	AUTO	PERM	OWN	0	1		
14	NACHBESTELLTE-ARTIKEL		CHAIN	LAST	AUTO	PERM	OWN	0	1		
15	ABGEBEBENE-BEST		CHAIN	LAST	AUTO	PERM	CUR	0	1		
16	EMPFANGENE-BEST		CHAIN	FIRST	MANL	PERM	CUR	0	1		
17	BESTELL-INHALT		LIST	NEXT	AUTO	PERM	CUR	20	1		
18	ERGEBNISSET	DYN	PTRAY	SORT	MANL	TRAN		0	1		
19	LIMITED-SET	DYN	PTRAY	SORT	MANL	TRAN		0	1		
20	IQL-DYN1	DYN	PTRAY	SORT	MANL	TRAN		0	1		
21	IQL-DYN2	DYN	PTRAY	SORT	MANL	TRAN		0	1		
22	IQL-DYN3	DYN	PTRAY	SORT	MANL	TRAN		0	1		
23	IQL-DYN4	DYN	PTRAY	SORT	MANL	TRAN		0	1		
24	IQL-DYN5	DYN	PTRAY	SORT	MANL	TRAN		0	1		
25	IQL-DYN6	DYN	PTRAY	SORT	MANL	TRAN		0	1		
26	IQL-DYN7	DYN	PTRAY	SORT	MANL	TRAN		0	1		
27	IQL-DYN8	DYN	PTRAY	SORT	MANL	TRAN		0	1		
28	SYS_RATENZAHLUNG	IMPL		SORT	AUTO	PERM		0	1		
29	SYS_ARTIKELART	IMPL		SORT	AUTO	PERM		0	1		
30	SYS_ARTIKELAUSWAHL	IMPL		SORT	AUTO	PERM		0	1		
31	SYS_ARTIKEL	IMPL		SORT	AUTO	PERM		0	1		
32	SYS_FARBN	IMPL		SORT	AUTO	PERM		0	1		
33	SYS_MATERIALIEN	IMPL		SORT	AUTO	PERM		0	1		
34	IMPLICIT_RESULT_SET	DYN	PTRAY	SORT	MANL	TRAN		0	1		

Die erste Tabelle enthält folgende Einträge:

REF Nummer eines Sets

SET-NAME

Name des Sets

TYPE Art des Sets

DYN: dynamischer Set

IMPL: impliziter Set

SING: SYSTEM-Set (singulärer Set)

.....: Standard Set

MODE

Set-Modus

CHAIN: Kette

CH-PR: Kette mit Rückwärtsverkettung (CHAIN LINKED TO PRIOR)

LIST: Liste

PTRAY: Adressliste (Pointer-Array)

.....: impliziter Set

ORDER

Sortierfolge innerhalb der Set-Occurrences des Sets

INSERT

Einhängen neuer Membersätze in den Set

AUTO: AUTOMATIC

MANL: MANUAL

REMOVE

Art der Set-Mitgliedschaft

PERM: permanent (MANDATORY Member)

TRAN: transient (OPTIONAL Member)

SOS Auswahl der Set-Occurrence (nur bei nicht-singulären Sets)

CUR: THRU CURRENT OF SET

OWN: THRU LOCATION MODE OF OWNER

INIT Angabe über die ursprüngliche Belegung der Set-Occurrence entsprechend der POPULATION-Klausel für diesen Set

INCR Anzahl der Einträge, um die eine Set-Occurrence zu erweitern ist entsprechend der INCREASE-Klausel für diesen Set

T-AREA

Realm-Nummer des Tabellenteils (Seiten mit Stufen >0) einer über mehrere Realms verteilbaren Liste („Tabellenrealm“).

## P-AREA

Realm-Nummer des Realms, in dem der DBH aktuell freie Seiten zur Einspeicherung von Membersätzen einer über mehrere Realms verteilbaren Liste sucht („Preferred-Realm“).

SET	OWNER	MEMBER	LNK	RSQ	OWN	MEM	OWN	MEM	OWN	MEM	ANCHOR-ACT	ANCHOR-DBK		
REF SET_NAME	OWNER	MEMBER	LNK	RSQ	OWN	MEM	OWN	MEM	OWN	MEM	ANCHOR-ACT	ANCHOR-DBK		
1 ERTEILTE-AUFTRAEGE	2	3		0	0	0	0	6	3					
2 AUFTR-INHALT	3	4		0	0	0	0	6		3				
3 NOCH-ZU-BEZAHLTEN	2	4		12	0	6	24	18	4	13				
4 RATENKAUF	2	5		12	24	0	24	18						
5 ANGEBOT	6	8		0	0	4	4	6		6				
6 NAEHERE-AUSWAHL	7	8		0	0	10	0	6						
7 BESTELLANGABEN	8	9 *		0	0	84	4	10		8				
8 MIN-BESTAND-ERREICHT	0	9		0	8	94	24	24	11	12	5	9	0	1
9 ENTHAELT	9	10 *		12	0	0	12	22	10	10				
10 ENTHALTEN-IN	9	10 *		24	12	22	24	34	13					
11 LIEFERANTEN	0	13		0	0	36	0	1	28		4	23	0	2
12 LIEFERBARE-ARTIKEL	13	9 *		0	0	118	0	10	15					
13 BESTELLTE-ARTIKEL	9	4 *		12	36	24	24	22	14					
14 NACHBESTELLTE-ARTIKEL	9	15 *		12	60	0	24	22		17				
15 ABGEGEBENE-BEST	13	14		12	0	4	24	18	16	16				
16 EMPFANGENE-BEST	13	14		12	24	22	12	18						
17 BESTELL-INHALT	14	15		0	0	22	4	6						
18 ERGEBNISSET	0	0		0	0	0	0	0	19		12	8	0	0
19 LIMITED-SET	0	0		0	0	0	0	0	20		12	8	0	0
20 IQL-DYN1	0	0		0	0	0	0	0	21		12	8	0	0
21 IQL-DYN2	0	0		0	0	0	0	0	22		12	8	0	0
22 IQL-DYN3	0	0		0	0	0	0	0	23		12	8	0	0
23 IQL-DYN4	0	0		0	0	0	0	0	24		12	8	0	0
24 IQL-DYN5	0	0		0	0	0	0	0	25		12	8	0	0
25 IQL-DYN6	0	0		0	0	0	0	0	26		12	8	0	0
26 IQL-DYN7	0	0		0	0	0	0	0	27		12	8	0	0
27 IQL-DYN8	0	0		0	0	0	0	0		12	8	0	0	0
28 SYS_RATENZAHLUNG	0	5		0	0	0	0	0	29	4	3	12	0	3
29 SYS_ARTIKELART	0	6		0	0	0	0	0	30		5	10	0	4
30 SYS_ARTIKELAUSSWAHL	0	7		0	0	0	0	0	31		11	32	0	5
31 SYS_ARTIKEL	0	9		0	0	0	0	0	32	7	5	11	0	6
32 SYS_FARBEN	0	11		0	0	0	0	0	33		11	33	0	7
33 SYS_MATERIALIEN	0	12		0	0	0	0	0			11	36	0	8
34 IMPLICIT_RESULT_SET	0	0		0	0	0	0	0			0	0	0	0

Die zweite Tabelle enthält folgende Einträge:

SET REF Nummer eines Sets

SET-NAME

Name des Sets

OWNER

Nummer der Ownersatzart bei nicht-singulären Sets

0: druckt BPSIA aus bei singulären Sets

MEMBER

Nummer der Membersatzart

0: druckt BPSIA aus bei dynamischen Sets

PHYS LNK

Kennzeichnung für zusätzliche Verkettung der Membersätze mit dem jeweiligen Ownersatz (physical linked to owner)

\*: zusätzliche Verkettung definiert

OWNER RSQ

Innerhalb der Set-Connection-Data der Membersatzart:

Distanz des Feldes, mit der Satzfolgennummer des Ownersatzes, zum Anfang der Set-Connection-Data für diesen Set (siehe Handbuch „[Entwerfen und Definieren](#)“)

SCD-DISPL

Distanz der Set-Connection-Data des Sets zum Satzbeginn:

OWN in der Ownersatzart

MEM der Membersatzart

SCD-LNGTH

Länge der Set-Connection-Data des Sets:

OWN in der Ownersatzart

MEM in der Membersatzart

CHAIN-LNK

Nummer des nächsten Sets mit:

OWN der gleichen Ownersatzart

MEM der gleichen Membersatzart

ANCHOR-ACT

Adresse (Realm-Nummer und Seitennummer) des Ankersatzes eines SYSTEM-Sets

Bei dynamischen Sets zeigt die Realm-Nummer auf den Temporären Realm, da dynamische Sets in Temporären Realms gespeichert werden.

ANCHOR-DBK

Database Key des Ankersatzes von ANCHOR-ACT

(Nummer der Satzart = 0)

## KEY INFORMATION (NO CALC SEARCH KEYS)

### Beispiel

\*\*\* KEY INFORMATION (NO CALC-SEARCH KEYS) \*\*\*

SET	KEY	DUPL	DUPL	LIST	PPP-BITS	DBTT	SSIA	OWNER	TABLE-ACT	KEY									
REF	SET-NAME	REF	TYPE	LNTH	NOT	TABLE	TABLE	INDEX	ATT	SET	O	M	TAB	SPLIT	COLUM	DISPL	DISPL	AREA	BNR <sup>1</sup>
1	ERTEILTE-AUFTRAEGE	1	ASC	4	*		*	*	*						2	1	0	0	
		2	SEARCH	6			*	*							2	2	28	0	
2	AUFTR-INHALT	3	ASC	2	*		*	*							2	1	0	0	
3	NOCH-ZU-BEZAHLN				*										2	0	0		
4	RATENKAUF				*										2	0	0		
5	ANGEBOT	4	ASC	40			*	*							5	1	0	0	11
6	NAEHERE-AUSWAHL	5	ASC	40			*	*							5	1	0	0	11
7	BESTELLANGABEN	6	ASC	4	*		*	*	*						2	1	0	0	0
8	MIN-BESTAND-ERREICHT	7	ASC	10	*		*	*							2	1	0	5	<b>8</b>
9	ENTHAELT				*										2	0	0		
10	ENTHALTEN-IN				*										2	0	0		
11	LIEFERANTEN	8	ASC	35	*		*	*							2	1	0	4	<b>22</b>
12	LIEFERBARE-ARTIKEL	9	ASC	40			*	*							5	1	0	0	
		10	SEARCH	1	*		*	*							2	2	28	0	
13	BESTELLTE-ARTIKEL				*										2	0	0		
14	NACHBESTELLTE-ARTIKEL				*										2	0	0		
15	ABGEGEBENE-BEST				*										2	0	0		
16	EMPFANGENE-BEST				*										2	0	0		
17	BESTELL-INHALT				*		*		*						2	1	0	0	0
18	ERGEBNISSET		ASC	1	*		*	*							2	1	0	12	<b>0</b>
19	LIMITED-SET		ASC	1	*		*	*							2	1	0	12	<b>0</b>
20	IQL-DYN1		ASC	1	*		*	*							2	1	0	12	<b>0</b>
21	IQL-DYN2		ASC	1	*		*	*							2	1	0	12	<b>0</b>
22	IQL-DYN3		ASC	1	*		*	*							2	1	0	12	<b>0</b>
23	IQL-DYN4		ASC	1	*		*	*							2	1	0	12	<b>0</b>
24	IQL-DYN5		ASC	1	*		*	*							2	1	0	12	<b>0</b>
25	IQL-DYN6		ASC	1	*		*	*							2	1	0	12	<b>0</b>
26	IQL-DYN7		ASC	1	*		*	*							2	1	0	12	<b>0</b>
27	IQL-DYN8		ASC	1	*		*	*							2	1	0	12	<b>0</b>
28	SYS_RATENZAHLUNG	11	SEARCH	6	*		*	*							2	1	0	3	<b>11</b>
30	SYS_ARTIKELAUSSWAHL	13	SEARCH	25			*	*							5	1	0	11	<b>31</b>
33	SYS_MATERIALIEN	18	SEARCH	1	*		*	*							5	1	0	11	<b>34</b>
		19	SEARCH	20	*		*	*							5	2	28	11	<b>35</b>
34	IMPLICIT_RESULT_SET		ASC	1	*		*	*							0	1	0	12	<b>0</b>

Unter der Überschrift KEY INFORMATION (NO CALC SEARCH KEYS) druckt BPSIA Informationen aus über die im Schema auf Satzartebene und Setebene definierten Schlüssel. Die Tabelle enthält keine Informationen über CALC-Keys und CALC-SEARCH-Keys (siehe [Seite 153](#) bzw. [Seite 162](#)).

<sup>1</sup> Die halbfett gedruckten Felder enthalten vor Ablauf des Dienstprogramms BFORMAT noch den Wert ,0'.

## SET REF

Nummer des Sets, zu dem der Schlüssel gehört

## SET-NAME

Name des Sets

## KEY REF

Nummer des Schlüssels

## TYPE Art des Schlüssels

ASC: ASCENDING-Key

DESC: DESCENDING-Key

SEARCH: SEARCH-Key (USING INDEX)

DBKEY: sortiert nach Satzfolgennummern

## LNGTH

Gesamtlänge des Schlüsselfeldes

## DUPL NOT

Kennzeichnung, ob gleiche Schlüsselwerte vorkommen dürfen oder nicht

\*: Duplikate nicht erlaubt

## DUPL TABLE

Kennzeichnung, ob eine Duplikattabelle angelegt wurde oder nicht

\*: Duplikattabelle (TYPE IS DATABASE-KEY-LIST)

┆: keine Duplikattabelle (TYPE IS REPEATED-KEY)

## TABLE

Kennzeichnung, ob eine Tabelle erzeugt wurde oder nicht

\*: Tabelle angelegt

## INDEX

Kennzeichnung, ob es sich um eine einstufige oder mehrstufige Tabelle handelt

\*: mehrstufige Tabelle

ATT Kennzeichnung, ob die Tabelle einer Set-Occurrence in der Nähe des Owners gespeichert werden soll

\*: ATTACHED TO OWNER

## LIST SET

Kennzeichnung, ob die Set-Occurrence-Tabelle als Liste angelegt ist

\*: Liste



## PPP-BITS

\* bedeutet bei

- O durch Verlagerung von Ownersätzen z.B. mit BREORG sind Probable Position Pointer (PPP), die auf diese Ownersätze verweisen, mit hoher Wahrscheinlichkeit nicht mehr aktuell
- M durch Verlagerung von Membersätzen z.B. mit BREORG sind Probable Position Pointer, die auf diese Membersätze verweisen, mit hoher Wahrscheinlichkeit nicht mehr aktuell
- TAB durch Verlagerung von Tabellen z.B. mit BREORG sind Probable Position Pointer, die auf diese Tabellen verweisen, mit hoher Wahrscheinlichkeit nicht mehr aktuell

Probable Position Pointer können auch vorliegen, ohne dass diese Anzeigen gesetzt sind, weil der DBH bei Verlagerung einzelner Sätze diese Bits nicht pflegt. Die Anzeigen werden in der Regel gesetzt, wenn durch Dienstprogramme alle Sätze verlagert wurden.

Ein BREORG-Lauf ist eventuell sinnvoll.

SPLIT Seitenanzahl für die Reorganisation laut REORGANIZATION-Klausel

2: Standardwert; wird auch ausgegeben, wenn keine Tabelle existiert

## DBTT COLUM

Spaltennummer in der DBTT der Ownersatzart, in der die Adresse der Tabelle eingetragen ist

0: keine Tabelle vorhanden

## SSIA DISPL

Distanz der Schlüssel-Beschreibung in der SSIA zum Beginn aller Schlüssel-Informationen für diesen Set

## OWNER DISPL

Innerhalb der Set-Connection-Data für diesen Set in der Ownersatzart:

Distanz des Feldes mit der Adresse der Tabelle. Es ist nur vorhanden, wenn die Tabelle mit dem Zusatz WITH PHYSICAL LINK definiert wurde

## TABLE-ACTKEY

bei Standard Sets:

AREA: Nummer des Realm, in dem die Tabelle gespeichert wird (DETACHED WITHIN); 0: Tabelle wird im Realm des Ownersatzes gespeichert (DETACHED oder ATTACHED)

BNR: keine Bedeutung bei Standard Sets

bei nicht Standard Sets: Adresse der Tabelle

AREA: Realm-Nummer

BNR: Seitennummer; 0 bei dynamischen Sets

## CALC-SEARCH-KEY INFORMATION

### Beispiel

\*\*\* CALC-SEARCH-KEY INFORMATION \*\*\*

SET REF	SET-NAME	KEY REF	DUPL LNGL	HASH NOT	PPP-BITS REF O M TAB	SSIA DISPL	PHYSICAL_CALC_INFO FIRST-BUCKET	NR-BUCKETS
29	SYS_ARTIKELART	12	25			0 11	14	1
31	SYS_ARTIKEL	14	8	*		0 11	15	3
		15	40			68 11	18	11
32	SYS_FARBEN	16	20	*		0 11	29	1
		17	2	*		28 11	30	1

Unter der Überschrift CALC-SEARCH-KEY-INFORMATION druckt BPSIA Informationen aus über SEARCH-Keys, die mit USING CALC auf Satzartebene oder Setebene (bei SYSTEM-Sets) definiert wurden:

#### SET REF

Nummer des Sets (impliziter Set oder SYSTEM-Set)

#### SET-NAME

Name des Sets

#### KEY REF

Nummer des Schlüssels

#### LNGL

Gesamtlänge des Schlüssels

#### DUPL NOT

Kennzeichnung, ob gleiche Schlüsselwerte vorkommen dürfen oder nicht

\*: Duplikate nicht erlaubt

#### HASH REF

Kennzeichnung der verwendeten Hashroutine

\_: Standard-Hashroutine von UDS/SQL

> 0: Nummer der benutzereigenen Hashroutine

#### PPP-BITS

\* bedeutet bei

O durch Verlagerung von Ownersätzen z.B. mit BREORG sind Probable Position Pointer (PPPs), die auf diese Ownersätze verweisen, mit hoher Wahrscheinlichkeit nicht mehr aktuell

M durch Verlagerung von Membersätzen z.B. mit BREORG sind Probable Position Pointer, die auf diese Membersätze verweisen, mit hoher Wahrscheinlichkeit nicht mehr aktuell

TAB durch Verlagerung von Tabellen z.B. mit BREORG sind Probable Position Pointer, die auf diese Tabellen verweisen, mit hoher Wahrscheinlichkeit nicht mehr aktuell

Probable Position Pointer können auch vorliegen, ohne dass diese Anzeigen gesetzt sind, weil der DBH bei Verlagerung einzelner Sätze diese Bits nicht pflegt. Die Anzeigen werden in der Regel gesetzt, wenn durch Dienstprogramme alle Sätze verlagert wurden.

Ein BREORG-Lauf ist eventuell sinnvoll.

#### SSIA DISPL

Distanz dieser Schlüssel-Beschreibung in der SSIA zum Beginn aller Schlüssel-Informationen für diesen Set

#### PHYSICAL CALC INFO

physische Informationen über den Hashbereich

#### FIRST-BUCKET

Adresse der ersten CALC-Seite, die für die Tabelleneinträge reserviert wurde

#### NR-BUCKETS

Anzahl der für die Tabelleneinträge reservierten CALC-Seiten

## 4.5 Beschreibung des SSIA-Protokolls

Das SSIA-Protokoll stellt einen tabellarisch aufbereiteten Abdruck eines Subschemas dar.

### SSIA PRINT REPORT (allgemeine Informationen)

*Beispiel*

```
*** SSIA PRINT REPORT ***  
  
DATABASE NAME      = VERSAND  
  
SCHEMA NAME        = ARTIKELVERSAND  
  
SUBSCHEMA NAME     = ADMIN  
  
SIA VALIDATION DATE = 2015-06-28 11:26:15  
  
LENGTH OF SSIA     = 4296  
  
TCUA LENGTH        = 2952  
  
CRA LENGTH         = 200  
  
CRR LENGTH         = 336  
  
CRS LENGTH         = 2240  
  
MAXIMUM AREA REF   = 12  
  
MAXIMUM RECORD REF = 15  
  
MAXIMUM SET REF    = 34  
  
NR AREAS           = 10  
  
NR RECORDS         = 14  
  
NR SETS            = 34
```

Unter der Überschrift SSIA PRINT REPORT druckt BPSIA folgende allgemeinen Informationen aus:

**DATABASE NAME**  
Name der Datenbank

**SCHEMA NAME**  
Name des Schemas

## SUBSCHEMA NAME

Name des Subschemas

## SIA VALIDATION DATE

Validierungsdatum des zugehörigen Schemas mit Datum und Uhrzeit

## LENGTH OF SSIA

Länge der SSIA

## TCUA LENGTH

Länge der Transaction Currency Area des Subschemas

## CRA LENGTH

Länge der CURRENT-OF-AREA-Tabelle innerhalb der TCUA

## CRR LENGTH

Länge der CURRENT-OF-RECORD-Tabelle innerhalb der TCUA

## CRS LENGTH

Länge der CURRENT-OF-SET-Tabelle innerhalb der TCUA

## MAXIMUM AREA REF

höchste Realm-Nummer innerhalb des Subschemas

## MAXIMUM RECORD REF

höchste Satzartnummer innerhalb des Subschemas

## MAXIMUM SET REF

höchste Setnummer innerhalb des Subschemas

## NR AREAS

Anzahl der Benutzerrealms innerhalb des Subschemas

## NR RECORDS

Anzahl der Satzarten innerhalb des Subschemas

## NR SETS

Anzahl der Sets innerhalb des Subschemas

## REFERENCE NUMBERS

*Beispiel*

```
*** REFERENCE NUMBERS ***  
  
AREAS   :    3    4    5    6    7    8    9   10   11   12  
  
RECORDS :    2    3    4    5    6    7    8    9   10   11  
          12   13   14   15  
  
SETS    :    1    2    3    4    5    6    7    8    9   10  
          11   12   13   14   15   16   17   18   19   20  
          21   22   23   24   25   26   27   28   29   30  
          31   32   33   34  
  
KEYS    :    1    2    3    4    5    6    7    8    9   10  
          11   12   13   14   15   16   17   18   19
```

Unter der Überschrift REFERENCE NUMBERS listet BPSIA die Nummern der im Subschema enthaltenen Realms, Satzarten, Sets und Schlüssel auf.

Die ausgedruckten Nummern entsprechen jeweils dem aktuellen Stand der Datenbank. Sie müssen nicht unbedingt den unter der Überschrift SSIA PRINT REPORT aufgeführten Maximalwert enthalten.

## AREA INFORMATION

### *Beispiel*

\*\*\* AREA INFORMATION \*\*\*

REF	AREA-NAME	CRA-DISPL
3	AUFTRAGSRLM	0
4	BESTELLRLM	20
5	KLEIDUNG	40
6	HAUSHALT	60
7	SPORT	80
8	LEBENSMITTEL	100
9	SPIELE-HOBBY	120
10	SCHREIBWAREN	140
11	ARTIKELRLM	160
12	SUCHRLM	180

Unter der Überschrift AREA INFORMATION druckt BPSIA Informationen aus über die CURRENT-OF-AREA-Tabelle der TCUA.

Alle übrigen Realminformationen enthält das SIA-Protokoll.

REF Nummer eines Realm

AREA-NAME

Name des Realm

CRA-DISPL

Distanz des zugehörigen Realm-Eintrags in der CURRENT-OF-AREA-Tabelle der TCUA zum Anfang der Tabelle

## RECORD INFORMATION

*Beispiel*

\*\*\* RECORD INFORMATION \*\*\*

REC REF	RECORD-NAME	REC DISPL	CRR DISPL	NR KEYS	DB-KEY-LOCATION		AREA-WITHIN-LIST								
					IMPL SET	REC DISPL	ITEM DISPL	AREA-ID DISPL	LIST-OF-AREAS						
2	KUNDE	0	0	0		0	60		3						
3	AUFTRAG	72	24	2					3						
4	AUFTR-POS	88	48	1					3						
5	RATENZAHLUNG	96	72	1	28				3						
6	ARTIKELART	128	96	1	29			1056	5	6	7	8	9		
									10						
7	ARTIKELAUSSWAHL	160	120	1	30			1086	5	6	7	8	9		
									10						
8	ARTIKELBESCHR	192	144	3				1116	5	6	7	8	9		
									10						
9	ARTIKEL	752	168	7	31			1146	5	6	7	8	9		
									10						
10	TEILMENGE	840	192	0				1176	6	7					
11	FARBEN	848	216	2	32				11						
12	MATERIALIEN	872	240	2	33				11						
13	LIEFERANT	896	264	2					4						
14	BESTELLUNG	1032	288	0					4						
15	BESTELL-POS	1048	312	0					4						

Unter der Überschrift RECORD INFORMATION druckt BPSIA Informationen aus über die Satzarten des Subschemas:

REC-REF

Nummer einer Satzart

RECORD-NAME

Name der Satzart

REC DISPL

Distanz der Satzart innerhalb der UWA zum Anfang der RECORD AREA

CRR DISPL

Distanz des Satzarteintrags in der CURRENT-OF-RECORD-Tabelle der TCUA zum Anfang der Tabelle



**NR KEYS**

Anzahl der für die Satzart definierten Schlüssel;  
die auf Setebene definierten Schlüssel zählt BPSIA nur dann mit, wenn sie in einem Set des Subschemas enthalten sind, in dem die Satzart als Membersatzart vorkommt

**IMPL SET**

Nummer des impliziten Sets, wenn ein SEARCH-Key auf Satzartebene definiert wurde

**DB-KEY-LOCATION**

druckt BPSIA nur aus, wenn im Schema eine Satzart vorkommt, die mit LOCATION MODE IS DIRECT oder DIRECT-LONG definiert ist:

**REC DISPL**

Distanz des Bereichs, der den Database Key enthält, zum Anfang der RECORD AREA; ist das Database-Key-Feld in der Satzart enthalten, so ist dieser Bereich der Satzbereich selbst

**ITEM DISPL**

Distanz des Database-Key-Feldes relativ zu der unter REC DISPL angegebenen Distanz

**AREA-WITHIN-LIST**

Informationen über die Realms, in denen Sätze der Satzart gespeichert werden können:

**AREA-ID DISPL**

Distanz des AREA-ID-Feldes der WITHIN-Klausel zum Anfang der RECORD AREA

**LIST-OF-AREAS**

Nummern der Realms, in denen - laut Subschema-Beschreibung - die Sätze der Satzart gespeichert werden können

## CALC KEY INFORMATION

*Beispiel*

\*\*\* CALC KEY INFORMATION \*\*\*

REF	RECORD-NAME	NR-ITEMS	ITEM-REF	REC-DISPL	LENGTH	TYPE	NEXT-SET	NEXT-KEY
8	ARTIKELBESCHR	1	6	26	40	4	5	4
9	ARTIKEL	3	0	128	6	5	8	7
			6	134	2	5	7	6
			54	182	2	5	7	6
13	LIEFERANT	2	0	37	5	5	11	8
			5	42	30	4	11	8

Unter der Überschrift CALC KEY INFORMATION druckt BPSIA eine Tabelle aus mit Informationen über die im Subschema enthaltenen CALC-Keys (LOCATION MODE IS CALC):

REF Nummer der zugehörigen Satzart

RECORD-NAME

Name der Satzart

NR-ITEMS

Anzahl der Felder, aus denen sich der CALC-Key zusammensetzt

ITEM-REF

Distanz eines CALC-Key-Feldes innerhalb der Satzart entsprechend dem Subschemaformat

┆ : das Feld ist im Subschemaformat der Satzart nicht enthalten

REC-DISPL

Distanz eines CALC-Key-Feldes innerhalb des Datenbanksatzes (Schemaformat einschließlich Set-Connection-Data)

LENGTH

Länge des CALC-Key-Feldes

TYPE Typ des Feldes:

0: Database-Key-Feld

1: numerisches Feld (gepackt)

2: binäres Feld

4: alphanumerisches Feld

5: numerisches Feld (entpackt)

8: Gleitpunkt-Feld

15: unterschiedliche Feldtypen (nur bei zusammengesetztem CALC-Key)

**NEXT-SET**

Nummer des nächsten Sets im Subschema, in dem dieses Feld als Schlüsselfeld definiert ist

**NEXT-KEY**

Nummer des Schlüssels von NEXT-SET

Die Einträge ITEM-REF bis NEXT-KEY wiederholen sich entsprechend der Anzahl der CALC-Key-Felder (NR-ITEMS).

## ITEM STRING LIST

Unter der Überschrift ITEM STRING LIST druckt BPSIA eine Tabelle aus über die Beziehungen zwischen dem Subschemaformat und dem Schemaformat einer Satzart.

Ein „ITEM STRING“ besteht aus einer Reihe von Feldern, die im Subschemaformat der Satzart in der gleichen lückenlosen Reihenfolge definiert sind wie im Schemaformat. Ist das Subschemaformat der Satzart mit dem Schemaformat identisch, so besteht der Subschemasatz aus einem Item-String.

### Beispiel

\*\*\* ITEM STRING LIST \*\*\*

REC REF	RECORD-NAME	COMPL REC	USER-REC DISPL	DB-REC DISPL	LENGTH
2	KUNDE	*	0	48	68
3	AUFTRAG	*	0	6	11
4	AUFTR-POS	*	0	46	8
5	RATENZAHLUNG	*	0	18	26
6	ARTIKELART	*	0	4	25
7	ARTIKELAUWAHL	*	0	0	25
8	ARTIKELBESCHR	*	58	78 V	502
			0	20	58
9	ARTIKEL	*	0	128	87
10	TEILMENGE	*	0	56	2
11	FARBEN	*	0	0	22
12	MATERIALIEN	*	0	0	21
13	LIEFERANT	*	0	37	130
14	BESTELLUNG	*	0	40	10
15	BESTELL-POS	*	0	28	8

Die Tabelle enthält folgende Einträge:

REC REF

Nummer einer Satzart

RECORD-NAME

Name der Satzart

COMPL REC

Kennzeichnung, ob der Subschemasatz mit dem Schemasatz identisch ist;

\*: identisch

USER-REC DISPL

Distanz eines Item-String im Subschemasatz zum Anfang des Subschemasatzes  
(in absteigender Folge der Distanzen)

**DB-REC DISPL**

Distanz des Item-String zum Anfang des Schemasatzes einschließlich der Set-Connection-Data

**LENGTH**

Länge des Item-String im Subschemasatz

V: steht vor der Länge ein V, so handelt es sich um ein variables Feld

Die Einträge USER-REC DISPL bis LENGTH wiederholen sich entsprechend der Anzahl der Item-Strings im Subschemasatz.

## KEY ITEM LIST

*Beispiel*

\*\*\* KEY ITEM LIST \*\*\*

REF	RECORD-NAME	ITEM-REF	REC-DISPL	LENGTH	SET-REF	KEY-REF
3	AUFTRAG	0	6	4	1	1
		4	10	2	1	2
		6	12	2	1	2
		8	14	2	1	2
4	AUFTR-POS	0	46	2	2	3
		20	38	2	28	11
5	RATENZAHLUNG	22	40	2	28	11
		24	42	2	28	11
		0	4	25	29	12
7	ARTIKELAUWAHL	0	0	25	30	13
8	ARTIKELBESCHR	6	26	40	CALC KEY ITEM	
					5	4
					6	5
9	ARTIKEL	0	128	6	CALC KEY ITEM	
					8	7
		6	134	2	CALC KEY ITEM	
					7	6
					8	7
		8	136	40	12	9
					31	15
		48	176	4	31	14
		52	180	2	31	14
		54	182	2	CALC KEY ITEM	
			7	6		
			8	7		
			31	14		
86	214	1	12	10		
11	FARBEN	0	0	2	32	17
		2	2	20	32	16
12	MATERIALIEN	0	0	1	33	18
		1	1	20	33	19
13	LIEFERANT	0	37	5	CALC KEY ITEM	
					11	8
		5	42	30	CALC KEY ITEM	
			11	8		

Unter der Überschrift KEY ITEM LIST druckt BPSIA eine Tabelle aus mit Informationen über alle Schlüsselfelder der Satzarten des Subschemas:

REF Nummer der zugehörigen Satzart

RECORD-NAME

Name der zugehörigen Satzart

ITEM-REF

Distanz des Schlüsselfeldes zum Anfang des Subschemasatzes

REC-DISPL

Distanz des Schlüsselfeldes zum Anfang des Schemasatzes einschließlich Set-Connection-Data

LENGTH

Länge des Schlüsselfeldes

SET-REF

Nummern (in aufsteigender Reihenfolge) der Sets, in denen das Schlüsselfeld ebenfalls enthalten ist;

CALC KEY ITEM

kennzeichnet CALC-Key-Felder der LOCATION MODE-Klausel, da diese als nicht setbezogene Schlüssel keine Set- und Schlüsselnummern haben

KEY-REF

Nummer des Schlüssels, zu dem das Schlüsselfeld gehört

## SET INFORMATION

*Beispiel*

\*\*\* SET INFORMATION \*\*\*

SET REF SET-NAME	NR KEYS	CRS DISPL	CRS-SORT KEY-DISPL	SOS-OWNER-INFO		(DIR)	(CALC)					SET-READY-LIST	
				LOC OWN	ALIAS	REC DISPL	ITEM DISPL	NR ITEMS	ITEM REF	DB-REC DISPL	ITEM LENGTH		ITEM TYPE
1 ERTEILTE-AUFTRAEGE	2	0	0	DIR		0	60						3
2 AUFTR-INHALT	1	80	4										3
3 NOCH-ZU-BEZAHLN	0	160	0	DIR		0	60						3
4 RATENKAUF	0	240	0	DIR		0	60						3
5 ANGEBOT	1	320	6										5 6 7 8
													10 11
6 NAEHERE-AUSWAHL	1	400	46										5 6 7 8
													10 11
7 BESTELLANGABEN	1	480	86										5 6 7 8
													10 11
8 MIN-BESTAND-ERREICHT	1	560	90										5 6 7 8
													10 11
9 ENTHAELT	0	640	0	CALC		752		3	0	128	6	5	5 6 7 8
									6	134	2	5	10 11
									54	182	2	5	
10 ENTHALTEN-IN	0	720	0	CALC	*	1056		3	150	128	6	5	5 6 7 8
									156	134	2	5	10 11
									158	182	2	5	
11 LIEFERANTEN	1	800	100										4
12 LIEFERBARE-ARTIKEL	2	880	135										4 5 6 7
													9 10 11
13 BESTELLTE-ARTIKEL	0	960	0	CALC		752		3	0	128	6	5	3 5 6 7
									6	134	2	5	9 10 11
									54	182	2	5	
14 NACHBESTELLTE-ARTIKEL	0	1040	0	CALC		752		3	0	128	6	5	4 5 6 7
									6	134	2	5	9 10 11
									54	182	2	5	
15 ABGEBEBENE-BEST	0	1120	0										4
16 EMPFANGENE-BEST	0	1200	0										4
17 BESTELL-INHALT	0	1280	0										4
.													
.													
.													
28 SYS_RATENZAHLUNG	1	0	0										3
29 SYS_ARTIKELART	1	0	0										5 6 7 8
													10 11
30 SYS_ARTIKELAUSSWAHL	1	0	0										5 6 7 8
													10 11
31 SYS_ARTIKEL	2	0	0										5 6 7 8
													10 11
32 SYS_FARBN	2	0	0										11
33 SYS_MATERIALIEN	2	0	0										11
34 IMPLICIT_RESULT_SET	0	2160	0										



Unter der Überschrift SET INFORMATION druckt BPSIA eine Tabelle aus mit Informationen über alle Sets des Subschemas:

**SET-REF**

Nummer eines Sets

**SET-NAME**

Name des Sets

**NR KEYS**

Anzahl der Schlüssel des zum Set gehörenden Membersatzes

**CRS DISPL**

Distanz des Set-Eintrags in der CURRENT-OF-SET-Tabelle der TCUA zum Anfang der Tabelle

**CRS-SORT KEY-DISPL**

Distanz des ASC-Key-Eintrags bzw. DESC-Key-Eintrags zum Anfang der KEY AREA in der TCUA

**SOS-OWNER-INFO**

Informationen über den Owner des Sets, wenn in der DDL die SET OCCURRENCE SELECTION-Klausel mit THRU LOCATION MODE OF OWNER spezifiziert wurde:

**LOC OWN:**

LOCATION MODE der Ownersatzart

CALC: mit Hashroutine

DIR: DIRECT bzw. DIRECT-LONG, d.h. mit Database Key

**ALIAS**

\*: ALIAS-Klausel definiert

**REC DISPL**

Distanz der Satzart zum Anfang der RECORD AREA, in der das Feld zum Auffinden des Owner liegt.

Verwenden Sie *bezeichner* bzw. haben Sie die ALIAS-Klausel angegeben, so ist dies ein impliziter Satz, der alle implizit definierten Felder enthält.

(DIR) nur wenn der LOCATION MODE des Owners DIRECT oder DIRECT-LONG ist:

**ITEM DISPL**

Distanz des Database-Key-Feldes, relativ zur Angabe unter REC DISPL

**(CALC)**

nur wenn der LOCATION MODE des Owners CALC ist:

**NR ITEMS**

Anzahl der Felder, aus denen sich der betreffende CALC-Key zusammensetzt

**ITEM REF**

Distanz des CALC-Key-Feldes innerhalb des Subschemasatzes;

Wurde eine ALIAS-Klausel angegeben:

Distanz des ALIAS-Feldes innerhalb des Satzes für implizit definierte Datenfelder

**DB-REC-DISPL**

Distanz des CALC-Key-Feldes im Schemasatz einschließlich Set-Connection-Data

**ITEM LENGTH**

Länge des CALC-Key-Feldes bzw. des ALIAS-Feldes

**ITEM TYPE**

Typ des CALC-Key-Feldes bzw. des ALIAS-Feldes (siehe [Seite 170](#))

**SET READY LIST**

Nummern der Realms, die beim Zugriff über den Set angesprochen werden können

## KEY INFORMATION

*Beispiel*

\*\*\* KEY INFORMATION \*\*\*

SET REF SET-NAME	KEY REF	NEXT KEY	NR DESC	ITEM ITEMS	DB-REC REF	DISPL DISPL	ITEM LENGTH	ITEM TYPE	KEY-CHAIN SET	KEY
1 ERTEILTE-AUFTRAEGE	1	28	1	0	6	4	5			
	2	0	3	4	10	2	5			
				6	12	2	5			
				8	14	2	5			
2 AUFTR-INHALT	3	0	1	0	46	2	5			
5 ANGEBOT	4	0	1	6	26	40	4	6	5	
6 NAEHERE-AUSWAHL	5	0	1	6	26	40	4			
7 BESTELLANGABEN	6	0	2	6	134	2	5	8	7	
				54	182	2	5	8	7	
8 MIN-BESTAND-ERREICHT	7	0	3	0	128	6	5			
				6	134	2	5			
				54	182	2	5	31	14	
11 LIEFERANTEN	8	0	2	5	42	30	4			
				0	37	5	5			
12 LIEFERBARE-ARTIKEL	9	28	1	8	136	40	4	31	15	
	10	0	1	86	214	1	4			
28 SYS_RATENZAHLUNG	11	0	3	20	38	2	5			
				22	40	2	5			
				24	42	2	5			
29 SYS_ARTIKELART	12	0	1	0	4	25	4			
30 SYS_ARTIKELAUSSWAHL	13	0	1	0	0	25	4			
31 SYS_ARTIKEL	14	68	3	48	176	4	5			
				52	180	2	5			
				54	182	2	5			
				15	0	1	8	136	40	4
32 SYS_FARBEN	16	28	1	2	2	20	4			
	17	0	1	0	0	2	5			
33 SYS_MATERIALIEN	18	28	1	0	0	1	4			
	19	0	1	1	1	20	4			

Unter der Überschrift KEY INFORMATION druckt BPSIA eine Tabelle aus mit Informationen über alle Schlüssel des Subschemas, außer denen der LOCATION MODE-Klausel:

SET REF

Nummer des zugehörigen Sets

SET-NAME

Name des zugehörigen Sets

**KEY REF**

Nummer des Schlüssels

**NEXT KEY**

Distanz innerhalb der SSIA zur nächsten Schlüssel-Beschreibung dieses Sets

0: der aktuelle Schlüssel ist der letzte oder einzige

**DESC**

Kennzeichnung, ob es sich um einen ASCENDING-Key oder DESCENDING-Key handelt

\*: DESCENDING-Key

**NR ITEMS**

Anzahl der Felder, aus denen sich der Schlüssel zusammensetzt

**ITEM REF**

Distanz jedes Feldes innerhalb des Subschemasatzes

65535: das Feld ist im Subschemaformat des Satzes nicht enthalten

**DB-REC DISPL**

Distanz jedes Feldes innerhalb des Schemasatzes einschließlich Set-Connection-Data

**ITEM LENGTH**

Länge des Feldes

**ITEM TYPE**

Typ des Feldes (siehe [Seite 170](#))

**KEY-CHAIN**

wenn das Feld in weiteren Sets als Schlüsselfeld definiert ist:

SET: Nummer des nächsten Sets mit dem gleichen Schlüsselfeld

KEY: Nummer dieses Schlüssels

---

## 5 Relationale Schemainformation ausgeben mit BPSQLSIA

Es ist möglich, auf die Daten einer UDS/SQL-Datenbank auch nach relationalen Gesichtspunkten zuzugreifen.

BPSQLSIA druckt die relationale Schemainformation eines bestehenden UDS/SQL-Subschemas aus, das gemäß dem CODASYL-Konzept definiert wurde. Die relationale Schemainformation dient als Programmierunterlage für den SQL-Anwender.

### 5.1 Überblick

Relationale Zugriffe können u.a. über die SQL-Schnittstelle von DRIVE V2.1 (siehe Handbuchreihe „[DRIVE/WINDOWS \(BS2000\)](#)“) erfolgen.

Um den SQL-Anwender in dieser Arbeitsweise zu unterstützen, druckt BPSQLSIA eine relationale Sicht für bestehende UDS/SQL-Datenstrukturen aus, die nach dem CODASYL-Konzept definiert wurden. Diese relationale Beschreibung der Datenstrukturen wird im folgenden **relationale Schemainformation** genannt.

Das CODASYL-Schema bleibt jedoch unverändert erhalten und kann von CODASYL-Anwendungen weiter benutzt werden.

Die relationale Schemainformation enthält alle notwendigen Angaben wie Tabellennamen und Feldbeschreibungen, damit der SQL-Anwender eine CODASYL-Datenbank auch relational bearbeiten kann. Sie zeigt auch, ob ein bestehendes CODASYL-Subschema vollständig relational bearbeitet werden kann, oder ob Zugriff eventuell eingeschränkt ist.

BPSQLSIA erzeugt pro CODASYL-Subschema eine relationale Schemainformation.

## 5.2 Systemumgebung

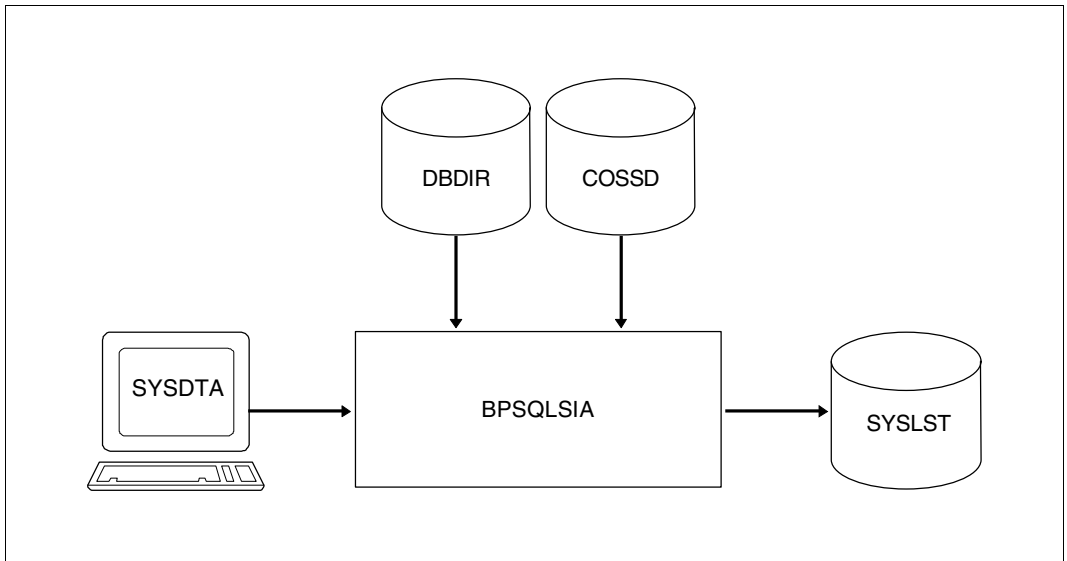


Bild 10: Systemumgebung von BPSQLSIA

BPSQLSIA ist parallel zum Datenbankbetrieb ablauffähig und wiederanlauffähig.

Wenn Sie mit dem Anwendungsbereich DATABASE arbeiten, können Sie BPSQLSIA mit dem BS2000-Kommando START-UDS-PRINT-SQLSIA oder dem Aliasnamen BPSQLSIA aufrufen.

Die UDS/SQL-Usersyntaxdatei setzt die SDF-Benutzerführung auf EXPERT. Sie können die Benutzerführung umschalten mit dem Kommando:

```
MODIFY-SDF-OPTIONS GUIDANCE=*EXPERT/*NO/*MAXIMUM/*MEDIUM/*MINIMUM
```

BPSQLSIA berücksichtigt beim Start ggf. eine zugewiesene UDS/SQL-Pubset-Deklaration (siehe Handbuch „[Datenbankbetrieb](#)“, Pubset-Deklarations-Jobvariable). Eine fehlerhafte Zuweisung führt zum Programmabbruch.

## 5.3 Voraussetzungen für SQL-Zugriffe in den CODASYL-Definitionen

Damit ein CODASYL-Subschema vollständig relational mit SQL bearbeitet werden kann, muss es folgende Bedingungen erfüllen:

Bedingung	SQL-Einschränkung, falls Bedingung nicht erfüllt
Der Temporäre Realm muss vorhanden sein	Kein SQL-Zugriff erlaubt
Alle Sets, deren Membersatzarten im Subschema liegen, müssen ebenfalls im Subschema liegen	Kein SQL-Zugriff auf die betroffene Satzart erlaubt
Alle ASCENDING-/DESCENDING-/CALC- und SEARCH-Keys müssen vollständig im Subschema enthalten sein	Kein SQL-Zugriff auf die betroffene Satzart erlaubt
Ein Set darf weder mit ORDER IS NEXT oder ORDER IS PRIOR definiert sein	Weder INSERT noch UPDATE auf die Membersatzart des Sets erlaubt
Eine Satzart darf nicht über mehrere Realms verteilt sein	Kein INSERT auf die betroffene Satzart erlaubt
Eine Satzart darf nicht mit LOCATION MODE IS DIRECT oder DIRECT-LONG definiert sein	Kein INSERT auf die betroffene Satzart erlaubt
Eine Satzart darf nicht mit der SSL-Klausel COMPRESSION definiert sein	Kein UPDATE auf die betroffene Satzart erlaubt
Eine Satzart darf kein Feld variabler Länge enthalten	Kein SQL-Zugriff auf die betroffene Satzart erlaubt
Eine Satzart darf kein nationales Feld (Unicode) enthalten	Kein SQL-Zugriff auf die betroffene Satzart erlaubt
Eine Satzart darf keine gepackten oder ungepackten numerischen Felder enthalten, für die bezüglich Anzahl der Speicherstellen und Skalenfaktor Folgendes gilt: Anzahl Speicherstellen > 15 oder Skalenfaktor < 0 oder Skalenfaktor > Anzahl Speicherstellen Ein positiver Skalenfaktor gibt die Anzahl der Stellen rechts vom Dezimalpunkt an, ein negativer Skalenfaktor gibt an, wie viel Nullen UDS/SQL beim Rechnen an den Feldinhalt anfügen muss.	Kein SQL-Zugriff auf das betroffene Feld erlaubt und kein INSERT auf die betroffene Satzart erlaubt.

Tabelle 13: Voraussetzungen für SQL-Zugriffe

## 5.4 SQL-Datentypen

Folgende Datentypen werden unterschieden:

- alphanumerischer Datentyp
- nationaler Datentyp
- numerische Datentypen
  - Festpunkt-Datentypen: DECIMAL, NUMERIC
  - Ganzzahlige Datentypen: INTEGER, SMALLINT

Daneben gibt es Datentypen, die sich aus diesen elementaren Datentypen zusammensetzen, sogenannte

- strukturierte Datentypen

Die folgende tabellarische Übersicht informiert über erlaubten Inhalt eines Satzelements, Wertebereich und Länge eines Satzelements bei den verschiedenen Datentypen.

### Alphanumerischer Datentyp

Datentyp	erlaubter Inhalt eines Satzelements, Wertebereich	Länge eines Satzelements in Byte
CHARACTER[(n)]	beliebige EBCDIC-Zeichen, wie Ziffern, Buchstaben oder Sonderzeichen $n$ = Anzahl der Zeichen $1 \leq n \leq 255$  <i>Beispiel</i> VORNAME CHARACTER(20)	$n$ Voreinstellung: $n=1$

### Nationaler Datentyp

Datentyp	erlaubter Inhalt eines Satzelements, Wertebereich	Länge eines Satzelements in Byte
NCHAR[(n)]	beliebige Unicode- bzw. NATIONAL-Zeichen $n$ = Anzahl der Zeichen $1 \leq n \leq 127$  <i>Beispiel</i> NACHNAME NCHAR(20)	$2n$ Voreinstellung: $n=1$



## Numerische Datentypen

Datentyp	erlaubter Inhalt eines Satzelements, Wertebereich	Länge eines Satzelements in Byte
DEC[IMAL]([n[,m]])	positive oder negative Festpunktzahl (gepackt) mit Vorzeichen. $n$ entspricht der Anzahl der Ziffern, davon sind $m$ Nachkommastellen. $1 \leq n \leq 15, 0 \leq m \leq n$  <i>Beispiel</i> ZWISCHENSUM DECIMAL(6,4)	$\frac{n+1}{2}$ aufgerundet  Voreinstellung: $n=15; m=0$
NUMERIC([n[,m]])	positive oder negative Festpunktzahl (ungepackt) mit $n$ Ziffern, davon sind $m$ Nach- kommastellen. $1 \leq n \leq 15, 0 \leq m \leq n$  <i>Beispiel</i> INSGESAMT NUMERIC(8,2)	$n$ Voreinstellung: $n=8; m=0$
SMALLINT	ganze Zahl im Bereich von -32768 bis 32767.  <i>Beispiel</i> SEITE SMALLINT	2
INTEGER	ganze Zahl im Bereich von -2147483648 bis 2147483647.  <i>Beispiel</i> ZAEHLER INTEGER	4

## Strukturierte Datentypen

Satzelemente strukturierten Datentyps bestehen wieder aus Satzelementen.

Satzelemente strukturierten Datentyps sind

- Vektoren,
- Strukturen und
- Vektoren mit strukturierten Elementen.

Ein strukturiertes Satzelement kann als gesamtes Satzelement angesprochen werden, oder Sie können einzelne Satzelemente daraus in SQL-Anweisungen ansprechen.

### Vektoren

Ein Vektor ist ein Satzelement von strukturiertem Datentyp, das sich aus einer festen Anzahl von Komponenten mit gleichem Datentyp zusammensetzt. Bei einem Vektor  $A$  kann eine einzelne Ausprägung in der Form  $A(l)$  oder ein Ausprägungsbereich in der Form  $A(l..m)$  bzw. der gesamte Ausprägungsbereich in der Form  $A$  oder  $A(1..n)$  angesprochen werden ( $n = \text{Zahl der Ausprägungen}$ ;  $1 \leq l \leq m \leq n$ ).

#### Beispiel

FREMDSPRACHE (3) CHARACTER(10)

└─ Anzahl der Ausprägungen

Der Vektor *FREMDSPRACHE* enthält 3 Ausprägungen der Länge 10 von alphanumerischem Datentyp.

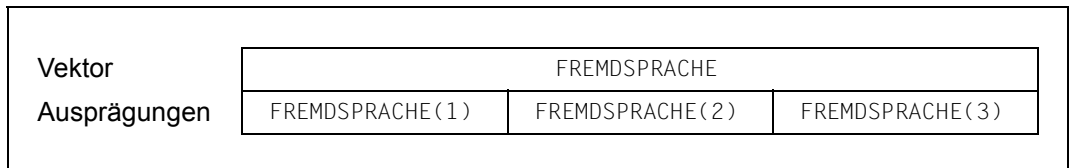


Bild 11: Strukturierter Datentyp - Vektor

## Struktur

Eine Struktur ist eine Zusammenfassung von Satzelementen.

Eine Struktur kann folgende Elemente enthalten:

- einfache Satzelemente von nicht-strukturiertem Datentyp,
- Vektoren,
- Strukturen oder
- Vektoren mit strukturierten Elementen.

### Beispiel

ADRESSE	STRUCTURE	Struktur
STRASSE	CHARACTER(20)	einfaches Satzelement
PLZ	NUMERIC(4)	einfaches Satzelement
ORT	CHARACTER(20)	einfaches Satzelement

Die Struktur *ADRESSE* besteht aus Satzelementen von alphanumerischem und numerischem Datentyp.

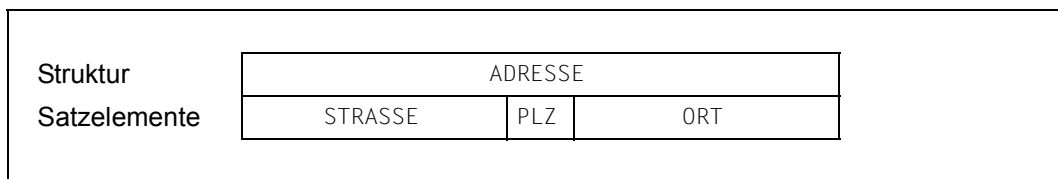


Bild 12: Strukturierter Datentyp - Struktur

## Nationale Struktur

Die nationale Struktur stellt einen Sonderfall einer Struktur dar. Sie darf ausschließlich folgende nationale Elemente enthalten:

- einfache Satzelemente von nationalem Datentyp,
- Vektoren aus Komponenten von nationalem Datentyp,
- nationale Strukturen oder
- Vektoren aus nationalen Strukturen.

Eine nationale Struktur wird als gesamtes Satzelement behandelt wie ein Satzelement von nationalem Datentyp.

*Beispiel*

NAME	NCHARSTRU	nationale Struktur
VORNAME	NCHAR(20)	einfaches Satzelement
NACHNAME	NCHAR(20)	einfaches Satzelement

Die Struktur *NAME* besteht aus Satzelementen von nationalem Datentyp.

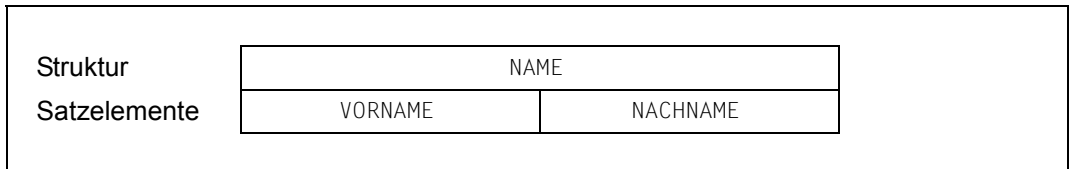


Bild 13: Strukturierter Datentyp - Nationale Struktur

**Vektor mit strukturierten Elementen**

Ein Vektor mit strukturierten Elementen ist eine Struktur mit Wiederholungsfaktor. Der Wiederholungsfaktor gibt an, wieviele Ausprägungen der Struktur zusammengefasst werden.

*Beispiel*

ADRESSE	STRUCTURE(2)	Vektor mit strukturierten Elementen
STRASSE	CHARACTER(20)	Satzelement
PLZ	NUMERIC(4)	Satzelement
ORT	CHARACTER(20)	Satzelement

Die Struktur *ADRESSE* kommt zwei Mal vor und ist deshalb ein Vektor mit strukturierten Elementen. Die Satzelemente *STRASSE*, *PLZ* und *ORT* kommen ebenfalls zwei Mal vor.

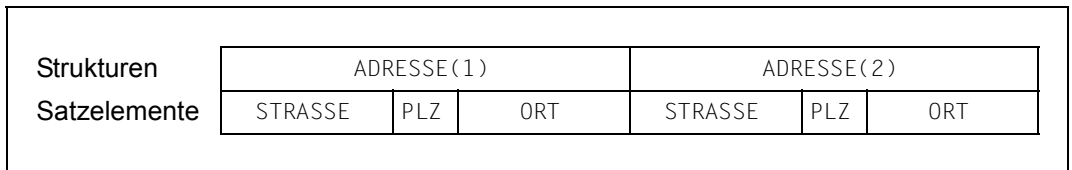


Bild 14: Strukturierter Datentyp - Vektor mit strukturierten Elementen

## 5.5 Anweisungen für BPSQLSIA

Anweisung	Bedeutung
END	Eingabe beenden
OPEN-DATABASE	Datenbank eröffnen
PRINT-RELATIONAL-SCHEMAINFO	Subschemata auswählen

Tabelle 14: Anweisungen für BPSQLSIA

Die Anweisungen von BPSQLSIA sind im Folgenden in alphabetischer Reihenfolge beschrieben.

### Eingabe beenden (END)

Mit dieser Anweisung beenden Sie die Eingabe und starten den Programmablauf.

END

Diese Anweisung hat keine Operanden.

## Datenbank eröffnen (OPEN-DATABASE)

Die Anweisung OPEN-DATABASE müssen Sie als erste Anweisung angeben, wenn Sie die Datenbank nicht zugewiesen haben mit

```
/ADD-FILE-LINK LINK-NAME=DATABASE ,
      FILE-NAME=[ :catid:][ $userid.]dbname.DBDIR[.copyname]
```

Wenn Sie die Anweisung ADD-FILE-LINK gegeben haben, wird die Anweisung OPEN-DATABASE als fehlerhaft abgewiesen, bzw. sie wird in der SDF-Maske nicht angeboten.

Der Dateikettungsname bleibt so lange gültig, bis er durch das Kommando REMOVE-FILE-LINK aufgehoben wird. Die Anweisung OPEN-DATABASE gilt jedoch nur bis zum Ende des BPSQLSIA-Laufes.

OPEN-DATABASE

DATABASE-NAME = <dbname>

,COPY-NAME = \*NONE / <copyname>

,USER-IDENTIFICATION = \*OWN / <userid>

### DATABASE-NAME = <dbname>

Name der Datenbank, mit der Sie arbeiten wollen.

### COPY-NAME = \*NONE

Es wird das Datenbank-Original bearbeitet.

### COPY-NAME = <copyname>

Es wird die Datenbankkopie mit dem angegebenen Kopienamen bearbeitet.

### USER-IDENTIFICATION = \*OWN

BPSQLSIA läuft in der gleichen Kennung, in der die Datenbank katalogisiert ist.

### USER-IDENTIFICATION = <userid>

Benutzerkennung, in der die Datenbank katalogisiert ist. Die Benutzerkennung wird ohne das \$-Zeichen angegeben.



Sie können eine Datenbankkopie zuweisen mit

```
/ADD-FILE-LINK LINK-NAME=DATABASE ,
      FILE-NAME=[ :catid:][ $userid.]dbname.DBDIR[.copyname]
```

Sie können den Kopienamen aber auch in der Anweisung OPEN-DATABASE angeben.

## Subschemata auswählen (PRINT-RELATIONAL-SCHEMAINFO)

In einem BPSQLSIA-Lauf können Sie bis zu 30 Subschemata explizit angeben. Bei PRINT-RELATIONAL-SCHEMAINFO \*ALL erzeugt BPSQLSIA jedoch ungeachtet der Anzahl für alle Subschemata der Datenbank relationale Schemainformationen. BPSQLSIA gibt die relationalen Schemainformationen in der Reihenfolge aus, in der die Subschemata im COSSD stehen, auch wenn im PRINT-Kommando eine andere Reihenfolge angegeben ist.

Die PRINT-Anweisung dürfen Sie mehrmals angeben.

```
PRINT-RELATIONAL-SCHEMAINFO
```

```
SUBSCHEMA-NAME = *ALL / *ALL-EXCEPT(...) / list-poss(20): <subschemaname>
```

```
  *ALL-EXCEPT(...)
```

```
    | NAME = list-poss(20): <subschemaname>
```

### **SUBSCHEMA-NAME = \*ALL**

Für alle Subschemata der Datenbank wird eine relationale Schemainformation erzeugt. Alle weiteren PRINT-Anweisungen werden ignoriert.

### **SUBSCHEMA-NAME = \*ALL-EXCEPT(...)**

Es wird für alle Subschemata der Datenbank eine relationale Schemainformation erzeugt, außer für die nach \*ALL-EXCEPT genannten Subschemata.

**NAME = list-poss(20): <subschemaname>**

Name der Subschemata, für die keine relationale Schemainformation erzeugt wird.

### **SUBSCHEMA-NAME = list-poss(20): <subschemaname>**

Es wird für die genannten Subschemata eine relationale Schemainformation erzeugt.

## 5.6 Kommandofolge zum Starten von BPSQLSIA

Die hier beschriebene Kommandofolge geht davon aus, dass UDS/SQL mit IMON installiert wurde (siehe Abschnitt „START-Kommandos der UDS/SQL-Programme“ in Kapitel 2 des Handbuchs „[Aufbauen und Umstrukturieren](#)“).

```
01 [/ADD-FILE-LINK LINK-NAME=DATABASE,  
    FILE-NAME=[ :catid: ][ $userid. ] dbname.DBDIR[ .copyname ]]  
02 /SELECT-PRODUCT-VERSION PRODUCT-NAME=UDS-SQL, VERSION=version. SCOPE=*TASK  
03 /START-UDS-BPSQLSIA  
04 [/OPEN-DATABASE DATABASE-NAME = ...]  
05 //PRINT-Anweisungen  
06 //END
```

- 01, 04 Sie müssen eine der beiden Zuweisungen für die Datenbank verwenden.
- 02 Die Version des Dienstprogramms wird ausgewählt.  
Die Angabe der Version wird generell empfohlen, da mehrere UDS/SQL-Versionen parallel installiert sein können.
- 03 BPSQLSIA kann von jeder beliebigen Kennung aus aufgerufen werden. Das UDS/SQL-Dienstprogramm kann auch mit den Aliasnamen BPSQLSIA oder START-UDS-PRINT-SQLSIA gestartet werden.



## 5.7 Beschreibung der Ausgabe von BPSQLSIA

BPSQLSIA gibt nach SYSLST aus:

1. Informationen zu den Basistabellen  
(Überschrift: INFORMATION ABOUT RELATIONAL SCHEMA *subschemaname*);  
darin ist enthalten:
  - Beschreibung der Felder jeder Basistabelle mit Feldname, Datentyp, Nullwertbedingung, Defaultwert und Zusatzinformationen. Zusatzinformationen können sein: PRIMARY KEY SYSTEMDEFINED, UNIQUE oder REFERENCES...
  - Zusammenstellung aller eindeutigen Schlüssel auf Setebene und aller eindeutigen Schlüssel auf Satzartebene, die aus mehr als einem Feld bestehen (UNIQUE-Übersicht).
  - Zusammenstellung aller einfachen und zusammengesetzten Schlüssel (INDEX-Übersicht).  
Für Schlüssel (z.B. Compound Key) ist die INDEX-Ausgabe zweigeteilt: in der linken Spalte stehen die Indizes, die SQL verwenden kann; in der rechten Spalte stehen die Indizes, wie sie in UDS/SQL definiert sind (z.B. Auflösung in Items).  
Auf Schlüssel, die nicht vollständig ins Subschema übernommen wurden, wird hingewiesen. Fehlende Schlüsselfelder werden mit drei Fragezeichen gekennzeichnet.
2. Tabelle aller Einschränkungen  
(Überschrift: SHORT INFORMATION ABOUT RELATIONAL SCHEMA);  
darin ist enthalten:
  - Eine Meldung, ob das Subschema mit SQL bearbeitet werden kann oder nicht.
  - falls das Subschema mit SQL bearbeitet werden kann, eine Übersicht über die zugelassenen SQL-Zugriffe pro Basistabelle unter der Überschrift: SHORT INFORMATION ABOUT TABLES.
3. Meldungen, wenn eine oder mehrere Bedingungen erfüllt sind, die die Zugriffsmöglichkeiten mit SQL einschränken (siehe [Seite 183](#))  
(Überschrift: DIAGNOSTIC SUMMARY FOR SUBSCHEMANAME *subschemaname*)

## 5.8 Umsetzregeln

Die CODASYL-Definitionen werden nach folgenden Regeln in eine relationale Schema-information umgesetzt.

Nr.	CODASYL-Subschema	relationale Schemabeschreibung
1	Satzart	Basistabelle gekennzeichnet mit TABLE .... Name der Basistabelle: <i>satzartname</i>
2	Satzart, die in mindestens einem Set Ownersatzart ist.	Primärschlüssel: <i>satzartname_</i> vom Datentyp INTEGER Zusatzinformation: PRIMARY KEY SYSTEMDEFINED Nullwertbedingung: NOT NULL
3	Feld einer Satzart	Feld der Basistabelle vom entsprechenden Datentyp mit Nullwertbedingung: NOT NULL Defaultwert: 0 für numerische Felder '_' für alphanumerische Felder
4	Datengruppe, Wiederholungsgruppe	Datentyp STRUCTURE
5	Wiederholungsfaktor	( <i>wiederholungsfaktor</i> )
6	Feld vom Typ DATABASE-KEY	Feld der Basistabelle mit dem gleichen Namen Datentyp: INTEGER Defaultwert: 0
7	Feld vom Typ DATABASE-KEY-LONG	Feld der Basistabelle mit dem gleichen Namen Datentyp: CHARACTER ; Länge 8 Zusatzinformation: ATTRIBUTE <i>feldname</i> IS DEFINED AS DATABASE-KEY-LONG; Standardwert ist X'00..00'
8	Einer oder mehrere SYSTEM-Sets, die nicht MANDATORY AUTOMATIC definiert sind	eine Basistabelle mit Namen SYSTEM und Primärschlüssel SYSTEM_ vom Datentyp INTEGER
9	Setbeziehung	Fremdschlüssel in der Basistabelle, die der Membersatzart entspricht Feldname: <i>setname_</i> Datentyp: INTEGER Referenzbedingung: REFERENCES <i>ownersatzartname</i>
10	Set: MANDATORY AUTOMATIC	Fremdschlüssel mit der Nullwertbedingung: NOT NULL

Tabelle 15: Umsetzregeln für BPSQLSIA

(Teil 1 von 2)

Nr.	CODASYL-Subschema	relationale Schemabeschreibung
11	Set: MANDATORY MANUAL	Fremdschlüssel mit der Nullwertbedingung: NOT NULL ON UPDATE Defaultwert: NULL
12	Set: OPTIONAL AUTOMATIC	Fremdschlüssel mit der Nullwertbedingung: NOT NULL ON INSERT
13	Set: OPTIONAL MANUAL	Fremdschlüssel ohne Nullwertbedingung Defaultwert: NULL
14	Schlüssel, bestehend aus einem Feld oder aus mehreren Feldern z.B. für zusammengesetzte Schlüssel	Zusatzinformation:  linke Spalte: INDEX ([setname_]gruppenname)  rechte Spalte: INDEX ([setname_]feld-1,...,feld-n)
15	eindeutiger Schlüssel auf Satzartebene bestehend aus einem Feld	Zusatzinformation UNIQUE für das Schlüsselfeld
16	eindeutiger Schlüssel auf Setebene bestehend aus einem Feld bei einem SYSTEM-Set, der MANDATORY AUTOMATIC definiert ist.	Zusatzinformation UNIQUE für das Schlüsselfeld
17	eindeutiger Schlüssel auf Satzartebene bestehend aus mehreren Feldern	Zusatzinformation UNIQUE für die entsprechende Basistabelle: UNIQUE (feld-1,...,feld-n)
18	eindeutiger Schlüssel auf Setebene, bestehend aus mehreren Feldern bei einem SYSTEM-Set, der MANDATORY AUTOMATIC definiert ist	Zusatzinformation UNIQUE für die der Member-satzart entsprechenden Basistabelle: UNIQUE (feld-1,...,feld-n)
19	eindeutiger Schlüssel auf Setebene, bestehend aus einem oder mehreren Feldern für einen Set, der mit MANDATORY AUTOMATIC definiert ist oder für einen SYSTEM-Set, der nicht mit MANDATORY AUTOMATIC definiert ist.	Zusatzinformation UNIQUE für die der Member-satzart entsprechenden Basistabelle: UNIQUE (setname_ feld-1,...,feld-n), wobei setname_ das zugehörige Fremdschlüssel-attribut ist.

Tabelle 15: Umsetzregeln für BPSQLSIA

(Teil 2 von 2)

Der Name des CODASYL-Subschemas wird zum Namen des relationalen Schemas. Wenn Namen im CODASYL-Schema Bindestriche enthalten, so erhalten die abgeleiteten Namen in der relationalen Schemainformation dafür Tiefstriche. Bedingungsnamen (Stufennummer 88) werden in der relationalen Schemainformation nicht ausgegeben.

Beispiele für die einzelnen Regeln finden Sie ab Seite 197.

Die Bedeutung der relationalen Begriffe und Konzepte finden Sie mit Beispielen im Handbuch „[SQL für UDS/SQL](#)“ erklärt.

## 5.9 Übersicht über die zugelassenen SQL-Zugriffe pro Basistabelle

BPSQLSIA gibt pro bearbeitetem CODASYL-Subschema folgende Übersicht aus:

\*\*\* SHORT INFORMATION ABOUT TABLES

TABLE	RET	INS	UPD	ATR
<i>basistabelle-1</i>	y/n	y/n	y/n	y/n
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
<i>basistabelle-n</i>	y/n	y/n	y/n	y/n

RET: y: Auf die Basistabelle ist der SQL-Retrieval-Zugriff SELECT erlaubt.  
n: Auf die Basistabelle ist kein SQL-Zugriff erlaubt.

INS y: Auf die Basistabelle ist der SQL-Zugriff INSERT erlaubt.  
n: Auf die Basistabelle ist der SQL-Zugriff INSERT nicht erlaubt.

UPD: y: Auf die Basistabelle ist der SQL-Zugriff UPDATE erlaubt.  
n: Auf die Basistabelle ist der SQL-Zugriff UPDATE nicht erlaubt.

ATR: y: Auf alle Felder der Basistabelle ist der SQL-Zugriff erlaubt.  
n: In der Basistabelle gibt es mindestens ein Feld, auf das kein SQL-Zugriff möglich ist.

## 5.10 Beispiel

### Ablauf von BPSQLSIA

```

/ADD-FILE-LINK LINK-NAME=DATABASE, FILE-NAME=PERSONAL.DBDIR
/SELECT-PRODUCT-VERSION PRODUCT-NAME=UDS-SQL, VERSION=02.8A00
/START-UDS-BPSQLSIA
***** START      BPSQLSIA      (UDS/SQL V2.8 0000 )    2015-06-28    11:26:05
//PRINT-RELATIONAL-SCHEMAINFO SUBSCHEMA-NAME=PERSONAL-DB
//END

***** DIAGNOSTIC SUMMARY FOR SUBSCHEMA PERSONAL-DB

                NO ERRORS
+++++          4 WARNINGS

***** END OF DIAGNOSTIC SUMMARY
***** NR OF DATABASE ACCESSES :           11
***** NORMAL END  BPSQLSIA      (UDS/SQL V2.8 0000 )    2015-06-28    11:26:05

```

### DDL des CODASYL-Subschemas

Die Nummern sind die Nummern der Umsetzregeln und verweisen auf die entsprechenden Teile in der relationalen Schemainformation.

```

SCHEMA PERS-DB.
AREA PERS-DB-REALM.
AREA TEMPO TEMPORARY.

```

1/2) — RECORD NAME IS ABTEILUNG WITHIN PERS-DB-REALM.

```

3) ———— 02 BEZEICHNUNG          PIC X(30).
           02 STANDORT            PIC X(30).
           02 EXTERN              PIC X.
           02 ABTEILUNG-MGR       TYPE IS BIN 31.

```

RECORD NAME IS MITARBEITER

15) ———— LOCATION MODE IS CALC USING PERSONAL-NR  
DUPLICATES ARE NOT ALLOWED

14) ———— SEARCH KEY IS M-NAME  
USING CALC  
DUPLICATES ARE ALLOWED

17) ———— SEARCH KEY IS PLZ,STADT,STRASSE  
USING INDEX

DUPLICATES ARE NOT ALLOWED.

```

5) 03 M-NAME          PIC X(30).
    03 VORNAME        PIC X(30) OCCURS 5.
    03 M-ALTER        TYPE IS BIN 15.
    03 FAM-STAND      PIC X.

4) 03 KINDER          OCCURS 10.
    04 K-NAME         PIC X(30).
    04 K-ALTER        TYPE IS BIN 15.
    03 ADRESSE        OCCURS 2.
    04 PLZ            PIC X(5).
    04 STADT          PIC X(15).
    04 STRASSE        PIC X(30).
    03 PERSONAL-NR    TYPE IS BIN 31.
    03 TAETIGKEIT     PIC X(10).
    03 GEHALT         PIC S9(8)V9(2).
    03 ZUSCHLAEGE     PIC S9(8)V9(2).
    03 MGR-NR         TYPE IS BIN 31.

```

RECORD NAME IS PROJEKT WITHIN PERS-DB-REALM.

```

02 PROJ-BEZ          PIC X(30).
02 BUDGET            PIC S9(10)V9(2).
02 PROJ-MGR         TYPE IS BIN 31.

9) SET NAME IS ABT-MIT
   ORDER IS FIRST
   OWNER IS ABTEILUNG.

10) MEMBER IS MITARBEITER
   MANDATORY AUTOMATIC
   SEARCH KEY IS MGR-NR
   USING INDEX DUPLICATES NOT ALLOWED

19) SEARCH KEY IS TAETIGKEIT,GEHALT
   USING INDEX DUPLICATES NOT ALLOWED
   SELECTION CURRENT.

   SET NAME IS PROJ-MIT
   ORDER IS FIRST
   OWNER IS PROJEKT.
   MEMBER IS MITARBEITER

13) OPTIONAL MANUAL
   SELECTION CURRENT.

8) SET NAME IS INT-FUND
   ORDER IS FIRST
   OWNER IS SYSTEM.
   MEMBER IS PROJEKT

12) OPTIONAL AUTOMATIC

```

```

SEARCH KEY IS PROJ-BEZ
      USING CALC DUPLICATES NOT ALLOWED.

SET NAME IS PROJ-MIT-2
ORDER IS FIRST
OWNER IS PROJEKT.
MEMBER IS MITARBEITER

11) ————— MANDATORY MANUAL
      SELECTION CURRENT.

SET NAME IS EXT-FUND
ORDER IS FIRST
OWNER IS SYSTEM.
MEMBER IS PROJEKT
      OPTIONAL MANUAL.

SET NAME IS SYS-MIT
ORDER IS FIRST
OWNER IS SYSTEM.
MEMBER IS MITARBEITER
      MANDATORY AUTOMATIC

16) ————— SEARCH KEY GEHALT
      USING CALC DUPLICATES NOT ALLOWED

18) ————— SEARCH KEY M-ALTER,FAM-STAND
      USING INDEX DUPLICATES NOT ALLOWED.

```

### Ausgabe von BPSQLSIA auf SYSLST

Die Nummern sind die Nummern der Umsetzregeln und verweisen auf die entsprechenden Teile in der DDL.

INFORMATION ABOUT RELATIONAL SCHEMA PERSONAL\_DB

1) — TABLE ABTEILUNG

	ATTRIBUTE	TYPE	NOT NULL	DEFAULT	
2) —	ABTEILUNG_	INTEGER	NOT NULL		PRIMARY KEY SYSTEMDEFINED
3) —	BEZEICHNUNG	CHARACTER ( 30)	NOT NULL	' '	
	STANDORT	CHARACTER ( 30)	NOT NULL	' '	
	EXTERN	CHARACTER ( 1)	NOT NULL	' '	
	ABTEILUNG_MGR	INTEGER	NOT NULL	0	

TABLE MITARBEITER

	ATTRIBUTE	TYPE	NOT NULL	DEFAULT
5) —	M_NAME	CHARACTER ( 30)	NOT NULL	' '
	VORNAME	( 5) CHARACTER ( 30)	NOT NULL	' '
	M_ALTER	SMALLINT	NOT NULL	0

4) —	FAM_STAND	CHARACTER ( 1)	NOT NULL	' '	
	KINDER	( 10) STRUCTURE		' '	
	K_NAME	CHARACTER ( 30)	NOT NULL	' '	
	K_ALTER	SMALLINT	NOT NULL	0	
	ADRESSE	( 2) STRUCTURE		' '	
	PLZ	CHARACTER ( 5)	NOT NULL	' '	
	STADT	CHARACTER ( 15)	NOT NULL	' '	
	STRASSE	CHARACTER ( 30)	NOT NULL	' '	
	PERSONAL_NR	INTEGER	NOT NULL	0	UNIQUE
	TAETIGKEIT	CHARACTER ( 10)	NOT NULL	' '	
16) —	GEHALT	NUMERIC (10, 2)	NOT NULL	0	UNIQUE
	ZUSCHLAEGE	NUMERIC (10, 2)	NOT NULL	0	
	MGR_NR	INTEGER	NOT NULL	0	
9/10) —	ABT_MIT_	INTEGER	NOT NULL		REFERENCES ABTEILUNG
13) —	PROJ_MIT_	INTEGER		NULL	REFERENCES PROJEKT
11) —	PROJ_MIT_2_	INTEGER	NOT NULL ON UPDATE	NULL	REFERENCES PROJEKT

- 17) — UNIQUE (PLZ,  
STADT,  
STRASSE)
- UNIQUE (ABT\_MIT\_,  
MGR\_NR)
- 19) — UNIQUE (ABT\_MIT\_,  
TAETIGKEIT,  
GEHALT)
- 18) — UNIQUE (M\_ALTER,  
FAM\_STAND)

INDEX TO BE USED BY SQL                      INDEX DEFINITION IN UDS

15) — INDEX (PERSONAL\_NR)

WARNING 4018 \*\*\* INDEX CAN BE USED ONLY WITHIN 'IN PREDICATE' OR WITHIN  
'COMPARISON PREDICATE' WITH 'EQUALS OPERATOR'

14) — INDEX (M\_NAME)

WARNING 4018 \*\*\* INDEX CAN BE USED ONLY WITHIN 'IN PREDICATE' OR WITHIN  
'COMPARISON PREDICATE' WITH 'EQUALS OPERATOR'

INDEX (ADRESSE)                                      INDEX (PLZ,  
STADT,  
STRASSE)

INDEX (ABT\_MIT\_,  
MGR\_NR)    INDEX (ABT\_MIT\_,  
TAETIGKEIT,  
GEHALT)

INDEX (GEHALT)

WARNING 4018 \*\*\* INDEX CAN BE USED ONLY WITHIN 'IN PREDICATE' OR WITHIN  
'COMPARISON PREDICATE' WITH 'EQUALS OPERATOR'

INDEX (M\_ALTER,  
FAM\_STAND)

TABLE PROJEKT

ATTRIBUTE	TYPE	NOT NULL	DEFAULT	
PROJEKT_	INTEGER	NOT NULL		PRIMARY KEY SYSTEMDEFINED
PROJ_BEZ	CHARACTER ( 30)	NOT NULL	' '	
BUDGET	NUMERIC (12, 2)	NOT NULL	0	
PROJ_MGR	INTEGER	NOT NULL	0	



```

12) —— INT_FUND_          INTEGER          NOT NULL ON INSERT  REFERENCES SYSTEM
        EXT_FUND_          INTEGER          NULL              REFERENCES SYSTEM

```

```

UNIQUE (INT_FUND_,
        PROJ_BEZ)

```

```

INDEX TO BE USED BY SQL          INDEX DEFINITION IN UDS

```

```

INDEX (INT_FUND_,
        PROJ_BEZ)

```

```

WARNING 4018 *** INDEX CAN BE USED ONLY WITHIN 'IN PREDICATE' OR WITHIN
'COMPARISON PREDICATE' WITH 'EQUALS OPERATOR'

```

TABLE SYSTEM

	ATTRIBUTE	TYPE	NOT NULL	DEFAULT		
8) ——	SYSTEM_	INTEGER	NOT NULL	0	PRIMARY KEY	SYSTEMDEFINED

\*\*\* SHORT INFORMATION ABOUT RELATIONAL SCHEMA

RELATIONAL SCHEMA CAN BE PROCESSED WITH SQL

\*\*\* SHORT INFORMATION ABOUT TABLES

RET = Y : TABLE CAN BE PROCESSED WITH SQL  
 = N : TABLE CANNOT BE PROCESSED WITH SQL  
 INS = N : NO INSERT ALLOWED ON TABLE  
 UPD = N : NO UPDATE ALLOWED ON TABLE  
 ATR = Y : ALL ATTRIBUTES CAN BE PROCESSED WITH SQL

TABLE	RET	INS	UPD	ATR
ABTEILUNG	Y	Y	Y	Y
MITARBEITER	Y	Y	Y	Y
PROJEKT	Y	Y	Y	Y

\*\*\*\*\* DIAGNOSTIC SUMMARY FOR SUBSCHEMA PERSONAL-DB

```

          NO ERRORS
+++++    4 WARNINGS

```

\*\*\*\*\* END OF DIAGNOSTIC SUMMARY



---

## 6 Statistik der Speicherplatzbelegung ausdrucken mit BSTATUS

BSTATUS druckt in tabellarischer Form Übersichten aus über den Zustand Ihrer Realms.

Sie enthalten Informationen über:

- den freien Speicherplatz in den Realms
- die DBTT-Größen und die Anzahl der freien DBTT-Einträge bzw. der belegten DBTT-Einträge
- den Füllgrad von Tabellen und den Speicherplatz, den diese Tabellen belegen
- die Ausnutzung der angelegten Hashbereiche und die Anzahl der Überlaufseiten
- die Verteilung der Satzarten auf die Realms

Die Übersichten ermöglichen es Ihnen somit, den für Ihre Realms reservierten Speicherplatz optimal zu nutzen.

Zusätzlich zur Ausgabe auf SYSLST können Sie die Daten auch im CSV-Format in eine Datei ausgeben lassen. Mit dem CSV-Format wird die Weiterverarbeitung von Daten in anderen Systemumgebungen (z.B. bei Spreadsheet-Anwendungen) erleichtert. Die Ausgabe im CSV-Format ist im Handbuch „[Datenbankbetrieb](#)“ im Abschnitt „Datenbankinformationen systemneutral ausgeben“ beschrieben.

## 6.1 Funktionen

Der Speicherplatz, den die Daten in Ihrer Datenbank belegen, schwankt im Laufe der Verarbeitung abhängig von den DB-Anwendungen, die Sätze in die Datenbank einspeichern oder aus ihr löschen.

Mit BSTATUS können Sie sich eine Übersicht über den belegten Speicherplatz erstellen und damit die Speicherplatzbelegung vollständig kontrollieren. Das gibt Ihnen die Möglichkeit

- den Speicherplatz, den die Realms Ihrer Datenbank belegen, jederzeit an die aktuellen Notwendigkeiten anzupassen, sodass nicht mehr Speicherplatz belegt wird als notwendig ist;
- den Ablauf von DB-Anwendungen vorzubereiten, die eine Reihe neuer Sätze in die Datenbank einspeichern, indem Sie den Datenbankelementen (Realms, Tabellen), die zum Speichern neuer Sätze zu klein geworden sind, mehr Speicherplatz zuweisen.

Sie können online auf die Originaldatenbank zugreifen, d.h. parallel zum Datenbankbetrieb, oder auf eine Schattendatenbank.



Wenn Sie BSTATUS online ausführen, müssen Sie damit rechnen, dass die ausgegebenen Daten nicht aktuell sind, da der DBH noch nicht alle Daten aus dem Puffer in die Datenbank geschrieben hat. Um weitgehend aktuelle Daten zu erhalten, sollten Sie unmittelbar vor dem BSTATUS-Lauf mit dem DAL-Kommando CHECKPOINT oder NEW RLOG eine Aktualisierung der Datenbank erzwingen. Aber auch danach sind noch Abweichungen der BSTATUS-Ausgabe vom tatsächlichen Stand der Datenbank möglich, falls parallel ein Update durchgeführt wird.

Im Einzelnen können Sie den von BSTATUS ausgedruckten Tabellen folgende Informationen entnehmen:

- Realm-Statistik - den pro Realm belegten Speicherplatz bzw. freien Speicherplatz:
  - Größe des Realm in Seiten
  - Anzahl freier Seiten
  - Anzahl teilweise gefüllter Seiten
  - Anzahl gefüllter Seiten
  - Anzahl aller freien Bytes in dem Realm

- Set-Statistik - den pro Set von Tabellen belegten Speicherplatz:
  - Anzahl der Set-Occurrences
  - Anzahl der gespeicherten Membersätze in der kleinsten und größten Set-Occurrence und durchschnittliche Anzahl der Membersätze aller Set-Occurrences

Für jede Tabelle des Sets wird ausgegeben:

  - Spaltennummern in der DBTT des Owners, in der die Adressen der Tabellen stehen
  - Füllgrad der Indexstufe 0 (Grundstufe)
  - Füllgrad aller Indexstufen ohne Grundstufe
  - Maximale und durchschnittliche Anzahl der Indexstufen ohne Grundstufe
  - Anzahl der Set-Occurrences, bei denen eine Reorganisation mit BREORG zur Reduzierung der Indexstufen führt
  
- Owner-Statistik - für einen Owner den pro Set von Tabellen belegten Speicherplatz:
  - Anzahl der Membersätze
  - Für jede Set-Occurrence-Tabelle des Owners die Spaltennummer in der DBTT, in der die Adresse der Tabelle steht
  - Füllgrad der Indexstufe 0 (Grundstufe)
  - Füllgrad aller Indexstufen ohne Grundstufe
  - Anzahl der Indexstufen ohne Grundstufe
  - Angabe, ob die Anzahl der Indexstufen mit BREORG reduziert werden kann.
  
- Satzart-Statistik - die pro Satzart belegten DBTT-Einträge bzw. freien DBTT-Einträge:
  - Anzahl der belegten DBTT-Einträge, d.h. Anzahl gespeicherter Satzexemplare
  - höchste vergebene Satzfolgenummer
  - höchste mögliche Satzfolgenummer, d.h. Anzahl der Sätze, die von dieser Satzart maximal gespeichert werden können
  - Füllgrad der DBTT in Prozent

- CALC-Key-Statistik - den pro Hashbereich von Primärseiten und Überlaufseiten belegten Speicherplatz:
  - Anzahl der reservierten Primärseiten
  - Anzahl der Sätze (bei einem direkten Hashbereich) bzw. Anzahl der Adressverweise (bei einem indirekten Hashbereich), die noch eingefügt werden können
  - Anzahl leerer Primärseiten
  - Füllgrad der Primärseiten
  - Anzahl der Überlaufseiten
  - Anzahl der Sätze bzw. Adressverweise in den Überlaufseiten
  - Füllgrad der Überlaufseiten
  - den Tiefenfaktor, d.h. die durchschnittliche Zahl der zum Auffinden eines Satzes benötigten Zugriffe
  
- Satzanzahl-Statistik - die pro Satzart in einem Realm gespeicherten Sätze:
  - für einen oder mehrere angegebene Realms: Anzahl der Sätze, die pro Satzart in diesen Realms gespeichert sind
  - für eine oder mehrere angegebene Satzarten: Anzahl der pro Realm gespeicherten Sätze der angegebenen Satzarten

Mit BSTATUS lassen sich auch Statistiken ausgeben über die Speicherplatzbelegung

- des Database Directorys (DBDIR) bzw.
- des Database Compiler Realm (DBCOM).

## 6.2 Systemumgebung

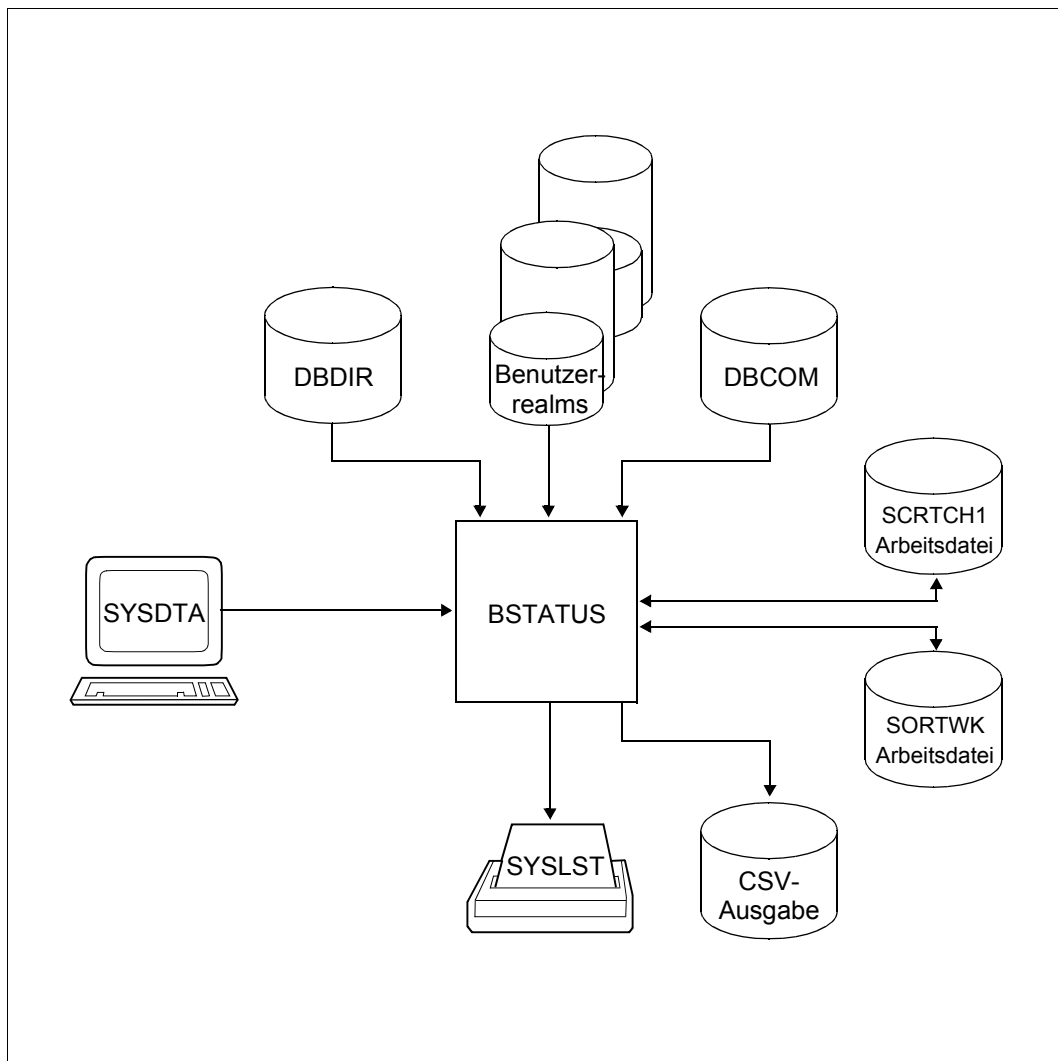


Bild 15: Systemumgebung von BSTATUS

BSTATUS berücksichtigt beim Start ggf. eine zugewiesene UDS/SQL-Pubset-Deklaration (siehe Handbuch „[Datenbankbetrieb](#)“, Pubset-Deklarations-Jobvariable). Eine fehlerhafte Zuweisung führt zum Programmabbruch.

## Arbeitsdateien

BSTATUS benötigt für die Set-Statistik und Owner-Statistik zum Sortieren zwei Arbeitsdateien, die BSTATUS automatisch auf gemeinschaftlicher Platte unter der Benutzerkennung einrichtet, unter der Sie BSTATUS gestartet haben. Die Arbeitsdateien haben standardmäßig die Dateikettungsnamen SCRTCH1 und SORTWK.

### SCRTCH1

benötigt BSTATUS, um das Ergebnis der Set-Statistik zwischen zu speichern, wenn die Ausgabe der Set-Statistik und der Owner-Statistik gewünscht wird.

### SORTWK

benötigt der von BSTATUS benutzte SORT für die Sortierung interner Auswertungssätze (siehe auch Handbuch „[SORT \(BS2000\)](#)“).

Wollen Sie die Arbeitsdateien explizit einrichten, so müssen diese die folgenden Eigenschaften besitzen:

#### *Arbeitsdatei-1*

Dateikettungsname SCRTCH1

Zugriffsmethode=SAM

Das Mengengerüst der zwischenspeichernden Daten ergibt sich aus der Formel:

$$132 * (setanzahl + schlüsselanzahl) \text{ Bytes}$$

#### *setanzahl*

Anzahl der Sets im auszuwertenden Subschema

#### *schlüsselanzahl*

Anzahl der Schlüssel im auszuwertenden Subschema

Die Primärzuweisung für die Arbeitsdatei-1 sollte sich am Mengengerüst der zwischenspeichernden Daten orientieren. Es sollte immer eine angemessene Sekundärzuweisung erfolgen für den Fall, dass der Speicherplatz erweitert werden muss.



*Arbeitsdatei-2*

Die Arbeitsdatei-2 wird vom SORT benötigt, wenn der virtuelle Speicher für die Vorsortierung nicht ausreicht. Die Primärzuweisung sollte sich am Mengengerüst der zu sortierenden Daten orientieren unter Berücksichtigung des von SORT empfohlenen Sicherheitsfaktors (siehe die Beschreibung der Arbeitsdateien im Handbuch „[SORT \(BS2000\)](#)“). Es sollte immer eine angemessene Sekundärzuweisung erfolgen, für den Fall, dass der Speicherplatz erweitert werden muss.

Dateikettungsname SORTWK

Zugriffsmethode=PAM

Das Mengengerüst der zu sortierenden Daten ergibt sich aus der Formel:

$$16 * \text{anzahl-sortsätze Bytes}$$

*anzahl-sortsätze*

Anzahl der in der Tabellen-Statistik zu bearbeitenden Sätze.

Richten Sie die beiden Arbeitsdateien nicht selbst ein, so richtet BSTATUS sie mit folgenden Namen und Größen ein:

UTI.SAMWORK.*tsn.zeitstempel.nnnn* (33,33)

UTI.*tsn.SORTWK* (120,120)

*tsn* Prozessfolgennummer des aktuellen Prozesses

*zeitstempel*

Zeitpunkt (*jjjjmmddhhmmss*), zu dem die Datei erzeugt wurde

*nnnn* vierstellige, fortlaufende Nummer

Nach normaler Beendigung des Laufes werden von BSTATUS angelegte Arbeitsdateien mit ihren Dateikettungsnamen wieder gelöscht. Explizit von Ihnen eingerichtete Arbeitsdateien werden nicht gelöscht und die Dateikettungsnamen nicht freigegeben.

## 6.3 Anweisungen für BSTATUS

Anweisung	Bedeutung
SUBSCHEMA	Subschema bezeichnen
DISPLAY REALM	Realm-Statistik ausdrucken
DISPLAY TABLE FOR SET	Set-Statistik ausdrucken
DISPLAY TABLE FOR OWNER	Owner-Statistik ausdrucken
DISPLAY RECORD	Satzart-Statistikausdrucken
DISPLAY CALC	CALC-Key-Statistik ausdrucken
DISPLAY RECORDNUMBER	Satzanzahl-Statistik ausdrucken
END	Eingabe der Anweisungen beenden

Tabelle 16: Anweisungen für BSTATUS

Alle DISPLAY-Anweisungen sind wahlweise. Sie können sie in beliebiger Reihenfolge beliebig oft angeben.

Die Anweisungen können über mehrere Zeilen gehen. Jede Zeile ist aber auf 72 Stellen beschränkt. Ein Fortsetzungszeichen ist bei mehrzeiliger Schreibweise nicht erforderlich.

Alle Anweisungen von BSTATUS können Sie mit einem Punkt (.) abschließen.

## Subschema bezeichnen (SUBSCHEMA)

---

SUBSCHEMA IS *subschemaname*

---

*subschemaname*

Name des Subschemas, für das Statistiken auszudrucken sind.

Sie können Folgendes angeben:

- *benutzersubschemaname* für Statistiken über Benutzerrealms
- COMPILER-SUBSCHEMA für Statistiken über den DBCOM
- PRIVACY-AND-IQF-SS für Statistiken über den DBDIR

Die SUBSCHEMA-Anweisung müssen Sie als erste Anweisung eingeben. Alle Realms, Satzarten und Sets, deren Statistiken BSTATUS ausdrucken soll, müssen im angegebenen Subschema enthalten sein.

## Realm-Statistik ausdrucken (DISPLAY REALM)

---

```
DISPLAY [IN CSV [csv-dateiname]] REALM STATISTICS FOR { realmname-1,... }  
ALL
```

---

### IN CSV

BSTATUS gibt die Daten zusätzlich im CSV-Format aus.

#### *csv-dateiname*

Name der Datei, in die die Daten im CSV-Format ausgegeben werden sollen. Die Angabe von *csv-dateiname* ist bei der ersten IN CSV-Anweisung einer BSTATUS-Ausführung zwingend (z.B. DISPLAY IN CSV 'BSTATUS.CSV' ...).

Eine ausführliche Beschreibung der Ausgabe im CSV-Format finden Sie im Handbuch „[Datenbankbetrieb](#)“ im Abschnitt „Datenbankinformationen systemneutral ausgeben“.

#### *realmname*

Name eines Realm, dessen Realmstatistik auszudrucken ist.

Sie geben an für

- Benutzerrealms: den in der Schema-DDL per AREA-Klausel definierten Realm-Namen
- das DBDIR: DATABASE-DIRECTORY
- den DBCOM: DATABASE-COMPILER-REALM

ALL BSTATUS druckt die Realm-Statistik aller Realms aus, die im angegebenen Subschema enthalten sind

*Beispiel*

DISPLAY REALM STATISTICS FOR ALL

SPACE USED PER REALM

R E A L M S	NR OF EMPTY PAGES	MAX NR OF CONTIGUOUS EMPTY PAGES	NR PAGES WITH FILLING PERCENTAGES BETWEEN					NR FULL USER PAGES	TOTAL NR OF PAGES	FREE OCTADS TOTAL
			0- 19	20- 39	40- 59	60- 79	80- 99			
AUFTRAGSRLM	22	22	1	0	0	0	0	10	36	91 520
BESTELLRLM	33	33	1	0	0	0	0	23	60	135K
KLEIDUNG	21	20	4	0	0	0	8	6	42	101K
HAUSHALT	15	14	0	0	0	0	0	6	24	59 700
SPORT	36	36	0	0	0	0	0	5	44	143K
LEBENSMITTEL	7	6	0	0	0	0	5	3	18	29 630
SPIELE-HOBBY	36	36	0	0	0	0	0	5	44	143K
SCHREIBWAREN	17	17	0	0	0	0	0	4	24	67 660
ARTIKELRLM	1982	1982	3	1	0	0	0	41	2062	7 903K
SUCHRLM			T E M P O R A R Y R E A L M							
T O T A L S	2 169		9	1	0	0	13	103	2 354	8 674K

Für Temporäre Realms gibt BSTATUS keine Statistik aus.

**REALMS**

Namen der Realms

**NR OF EMPTY PAGES**

Anzahl der freien Seiten

**MAX NR OF CONTIGUOUS EMPTY PAGES**

maximale Anzahl zusammenhängender freier Seiten

**NR PAGES WITH FILLING PERCENTAGES BETWEEN**

Anzahl der teilweise gefüllten Seiten, eingeteilt in Gruppen mit dem angegebenen Füllgrad (ohne FPA-Seiten, Act-Key-0-Seiten und Act-Key-N-Seiten)

## NR FULL USER PAGES

Anzahl der gefüllten Seiten (ohne FPA-Seiten, Act-Key-0-Seiten und Act-Key-N-Seiten)

## TOTAL NR OF PAGES

Größe des Realm in Seiten (mit FPA-Seiten, Act-Key-0-Seiten und Act-Key-N-Seiten)

## FREE OCTADS TOTAL

Anzahl aller freien Bytes pro Realm

K: ist die Zahl größer als 100.000 byte, so rundet BSTATUS auf ein Vielfaches von 1000 =1 kbyte

M: ist die Zahl größer als 100.000 kbyte, so rundet BSTATUS auf ein Vielfaches von 1000 kbyte =1 Mbyte

## TOTALS

geben Sie in der DISPLAY REALM-Anweisung ALL an, so druckt BSTATUS in der Zeile TOTALS die Summe aller Spalten aus.

Die Gesamtsumme freier Bytes errechnet er dabei vor dem Runden der Zahlen für die einzelnen Realms.

## Set-Statistik ausdrucken (DISPLAY TABLE FOR SET)

---

DISPLAY [IN CSV [*csv-dateiname*]] TABLE STATISTICS FOR SET

$$\left. \begin{array}{l} \textit{setname-1}, \dots \\ *ALL [ EXCEPT \textit{setname-1}, \dots ] \end{array} \right\}$$


---

### IN CSV

BSTATUS gibt die Daten zusätzlich im CSV-Format aus.

*csv-dateiname*

Name der Datei, in die die Daten im CSV-Format ausgegeben werden sollen. Die Angabe von *csv-dateiname* ist bei der ersten IN CSV-Anweisung einer BSTATUS-Ausführung zwingend (z.B. DISPLAY IN CSV 'BSTATUS.CSV' ...).

Eine ausführliche Beschreibung der Ausgabe im CSV-Format finden Sie im Handbuch „[Datenbankbetrieb](#)“ im Abschnitt „Datenbankinformationen systemneutral ausgeben“.

*setname*

Name eines Sets, dessen Statistik auszudrucken ist

\*ALL die Statistik aller Sets des Subschemas ist auszudrucken

\*ALL EXCEPT *setname-1*,...

die Statistik aller Sets, ausgenommen der mit *setname* angegebenen, ist auszudrucken

*Beispiel*

DISPLAY TABLE STATISTICS FOR SET \*ALL

SET IDENTIFICATION	OWNER OCCURRENCE	MEMBER OCCURRENCE			TABLE STATISTICS						
		MIN	MAX	AVG	COL	FILLING RATIO		LEVEL	NR	NR OCCURRENCES TO REORGANIZE	
					NR	LEV=0	LEV>0	MAX	AVG		
ERTEILTE-AUFTRAEGE					***** EMPTY SET *****						
ANGEBOT	4	1	8	3	1	5	0	0	0,0	0	
NAEHERE-AUSWAHL	5	1	6	2	1	4	0	0	0,0	0	
BESTELLANGABEN	13	1	7	4	1	33	0	0	0,0	0	
MIN-BESTAND-ERREICHT	SINGULAR			0	1	0	0		0	0	
ENTHAELT					***** EMPTY SET *****						
ENTHALTEN-IN					***** EMPTY SET *****						
LIEFERANTEN	SINGULAR			1	1	2	0		0	0	
LIEFERBARE-ARTIKEL	1	63	63	63	1	80	0	0	0,0	0	
					2	10	0		0,0		
BESTELL-INHALT					***** EMPTY SET *****						
ERGEBNISSET					***** DYNAMIC SET *****						
SYS_RATENZAHLUNG	SINGULAR			0	1	1	0		0	0	
SYS_ARTIKELART	SINGULAR			4	**** CHAIN SET WITHOUT TABLES ****						0
SYS_ARTIKELAUSSWAHL	SINGULAR			5	1	5	0		0	0	
SYS_ARTIKEL	SINGULAR			63	**** CHAIN SET WITHOUT TABLES ****						0
SYS_FARBEN	SINGULAR			25	**** CHAIN SET WITHOUT TABLES ****						0
SYS_MATERIALIEN	SINGULAR			10	1	3	0		0	0	
					2	8	0		0		
IMPLICIT_RESULT_SET					***** DYNAMIC SET *****						



BSTATUS gibt keine Statistik aus für dynamische Sets und für Sets, für die keine Set-Occurrences existieren. Diese werden mit \*EMPTY SET\* angezeigt.

#### SET IDENTIFICATION

Name des Sets im Schema

#### OWNER OCCURRENCE

Anzahl der Set-Occurrences des Sets

SINGULAR: SYSTEM-Set

#### MEMBER OCCURRENCE

Anzahl der gespeicherten Membersätze des Sets:

MIN: Anzahl der Membersätze der kleinsten Set-Occurrence

MAX: Anzahl der Membersätze der größten Set-Occurrence

AVG: durchschnittliche Anzahl der Membersätze pro Set

#### TABLE STATISTICS

Informationen über die Tabellen des Sets

(Adresslisten, Sort-Key-Tabellen oder SEARCH-Key-Tabellen)

Hat der Set keine Tabellen, wird der Text ausgegeben:

\*\*\*\* CHAIN SET WITHOUT TABLES \*\*\*\*

#### COL NR

Spaltennummern der DBTT des Owners, in der die jeweilige Adresse der Tabelle eingetragen ist

#### FILLING RATIO

Füllgrad der Tabellen, ausgedrückt als Prozentsatz aller für die Tabelle reservierten Bytes;

LEV=0: durchschnittlicher Füllgrad der Tabellen auf der Indexstufe 0 (Grundstufe)

0: Tabellen werden angelegt, es sind aber keine Membersätze gespeichert (leere Set-Occurrence). Bei Duplikattabellen ist der Füllgrad immer größer als 0, auch bei leeren Set-Occurrences.

LEV>0: durchschnittlicher Füllgrad der Tabellen aller Indexstufen (Grundstufe)

#### LEVEL NR

Anzahl der Indexstufen des Sets (ohne Grundstufe):

MAX: Höchste Indexstufe, die bei einer Tabelle einer Set-Occurrence auftritt

AVG: durchschnittliche Anzahl der Indexstufen des Sets

#### NR OCCURRENCES TO REORGANIZE

Anzahl der Set-Occurrences, bei denen Sie mit BREORG die Anzahl der Indexstufen reduzieren können. Dies gilt für eine Reorganisation mit der Anweisung

REORGANIZE SET NAME IS *setname* FILLING IS 100 PERCENT

## Owner-Statistik ausdrucken (DISPLAY TABLE FOR OWNER)

---

```

DISPLAY [IN CSV [csv-dateiname]] TABLE STATISTICS FOR OWNER IN SET
                                     { setname-1[rsq-selection-1] ,... }
                                     { *ALL[ EXCEPT setname-1,...] }

```

---

### IN CSV

BSTATUS gibt die Daten zusätzlich im CSV-Format aus.

#### *csv-dateiname*

Name der Datei, in die die Daten im CSV-Format ausgegeben werden sollen. Die Angabe von *csv-dateiname* ist bei der ersten IN CSV-Anweisung einer BSTATUS-Ausführung zwingend (z.B. DISPLAY IN CSV 'BSTATUS.CSV' ...).

Eine ausführliche Beschreibung der Ausgabe im CSV-Format finden Sie im Handbuch „Datenbankbetrieb“ im Abschnitt „Datenbankinformationen systemneutral ausgeben“.

#### *setname*

Name eines Sets, für dessen Owner eine Statistik ausgegeben werden soll

#### *rsq-selection*

*rsq-selection* = { ALL RSQS / RSQ {*rsq-1*[ IO *rsq-2*]} ,... }

$1 \leq \text{RSQ} \leq 2^{24} \cdot 2^{16} - 1$

Für Datenbanken mit einer Seitenlänge von 2048 Bytes, falls es sich um eine Ownersatzart in einem Set handelt.

$1 \leq \text{RSQ} \leq 231 - 1$

Für Datenbanken mit einer Seitenlänge von 4000/8096 Bytes.

\*ALL die Statistik ist über alle Ownersatzarten des Subschemas auszudrucken

\*ALL EXCEPT *setname-1*,...

die Statistik für alle Sets, ausgenommen der mit *setname* angegebenen, ist auszudrucken

Die DISPLAY TABLE FOR OWNER-Anweisung druckt eine Statistik aus über den Speicherplatz, den die Tabellen der Ownersätze eines Sets belegen.



Der Umfang der Ausgabe kann sehr groß werden, da er im Gegensatz zu den anderen Anweisungen nicht nur vom Mengengerüst der Metadaten abhängt, sondern auch vom Mengengerüst der Benutzerdaten.

*Beispiel*

1. DISPLAY TABLE STATISTICS FOR OWNER IN SET BESTELLANGABEN RSQ 1 TO 5

OWNER DBK		MEMBER OCCURRENCE	COL NR	FILLING RATIO	LEV=0	LEV>0	NR	REORG
8:	1	7	1	47	0	0	-	
8:	2	6	1	41	0	0	-	
8:	3	2	1	14	0	0	-	
8:	4	2	1	14	0	0	-	
8:	5	2	1	14	0	0	-	

Diese Tabelle druckt BSTATUS für jeden Set getrennt aus.

**OWNER DBK**

Database Key des Ownersatzes im Format: (*ref.ref.rsq*)

**MEMBER OCCURRENCE**

Anzahl der Member der durch OWNER DBK spezifizierten Set-Occurrence;

0: keine Membersätze gespeichert, Tabelle aber bereits angelegt

**TABLE STATISTICS**

Informationen über alle Tabellen der Set-Occurrence

(Adressliste, Liste oder Sort-Key-Tabelle und SEARCH-Key-Tabellen)

**COL NR**

Spaltennummern der DBTT des Owners, in der die jeweilige Adresse der Tabellen eingetragen ist

FILLING RATIO

Füllgrad der Tabelle, ausgedruckt als Prozentsatz aller für die Tabelle reservierten Bytes;

LEV=0: Füllgrad der Tabelle auf der Indexstufe 0 (Grundstufe);  
 0: Tabellen werden angelegt, es sind aber keine Membersätze gespeichert. Bei Duplikattabellen ist der Füllgrad immer größer als 0, auch bei leeren Set-Occurrences.

LEV>0: Füllgrad der Tabelle aller Indexstufen (ohne Grundstufe)

LEV NR

Anzahl der Indexstufen (ohne Grundstufe)

REORG

gibt an, ob Sie die Anzahl der Indexstufen durch Reorganisieren reduzieren können.

YES: die Anzahl der Indexstufen lässt sich mit dem Dienstprogramm BREORG mit der Anweisung REORGANIZE SETNAME IS *setname* FILLING IS 100 PERCENT reduzieren

NO: die Anzahl der Indexstufen lässt sich nicht verringern

- : die Tabelle ist einstufig (Grundstufe), d.h. Sie können nur die Belegung der Tabelle erhöhen

- Falls ein Benutzer als Owner RSQ oder eine Reihe von RSQs festlegt, obwohl diese RSQs nicht in der Datenbank existieren, bekommt der Benutzer eine Nachricht:

DISPLAY TABLE FOR OWNER IN SET BESTELLANGABEN RSQ 2

NO MATCHING RSQS: There is no record with RSQ 2 in the database.

```

*-----*-----*-----*-----*
|                                     | TABLE STATISTICS | | | | |
|                                     | *-----*-----* |
| OWNER DBK | MEMBER OCCUR- | COL | FILLING RATIO | LEV | REORG |
|                                     | RENCE | NR | LEV=0 | LEV>0 | NR |
*-----*-----*-----*-----*
| ***** N O M A T C H I N G R S Q S ***** | | |
*-----*-----*-----*-----*
    
```

## Satzart-Statistik ausdrucken (DISPLAY RECORD)

---

```
DISPLAY [IN CSV [csv-dateiname]] RECORD STATISTICS FOR { satzname-1,... }  
{ ALL }
```

---

### IN CSV

BSTATUS gibt die Daten zusätzlich im CSV-Format aus.

#### *csv-dateiname*

Name der Datei, in die die Daten im CSV-Format ausgegeben werden sollen. Die Angabe von *csv-dateiname* ist bei der ersten IN CSV-Anweisung einer BSTATUS-Ausführung zwingend (z.B. DISPLAY IN CSV 'BSTATUS.CSV' ...).

Eine ausführliche Beschreibung der Ausgabe im CSV-Format finden Sie im Handbuch „[Datenbankbetrieb](#)“ im Abschnitt „Datenbankinformationen systemneutral ausgeben“.

#### *satzname*

Name einer im angegebenen Subschema enthaltenen Satzart, für die die Satzart-Statistik auszudrucken ist

ALL BSTATUS druckt für alle im angegebenen Subschema enthaltenen Satzarten die Satzart-Statistik aus

*Beispiel*

DISPLAY RECORD STATISTICS FOR ALL

STATUS PER RECORD TYPE

RECORDS	D B T T						
	NR OF ENTRIES USED	HIGHEST RSQ USED	HIGHEST RSQ POSSIBLE	FILLING RATIO %	EXTENDIBLE	NR OF DBTT EXTENTS	HIGHEST USED EXTENT
KUNDE	0	0	331	0,0	SCAN	0	0
AUFTRAG	0	0	497	0,0	SCAN	0	0
AUFTR-POS	0	0	1990	0,0	SCAN	0	0
RATENZAHLUNG	0	0	995	0,0	SCAN	0	0
ARTIKELART	4	4	497	0,8	SCAN	0	0
ARTIKELAUSWAHL	5	5	497	1,0	SCAN	0	0
ARTIKELBESCHR	13	13	497	2,6	SCAN	0	0
ARTIKEL	63	63	995	6,3	SCAN	0	0
TEILMENGE	0	0	995	0,0	SCAN	0	0
FARBEN	25	25	995	2,5	SCAN	0	0
MATERIALIEN	10	10	995	1,0	SCAN	0	0
LIEFERANT	1	1	662	0,1	SCAN	0	0
BESTELLUNG	0	0	497	0,0	SCAN	0	0
BESTELL-POS	0	0	995	0,0	SCAN	0	0
T O T A L	121						

## RECORDS

Namen der Satzarten

DBTT für jede Satzart druckt BSTATUS die wesentlichen Informationen aus der DBTT aus:

## NR OF ENTRIES USED

Anzahl der belegten DBTT-Einträge, d.h. Anzahl der derzeit gespeicherten Sätze der Satzart

## HIGHEST RSQ USED

höchste vergebene Satzfolgenummer; wurden Sätze gelöscht, so kann diese Nummer höher sein als die Anzahl der derzeit gespeicherten Sätze

## HIGHEST RSQ POSSIBLE

höchste mögliche Satzfolgenummer, d.h. Anzahl der Sätze, die von dieser Satzart in der derzeitigen DBTT maximal gespeichert werden können

## FILLING RATIO %

Füllgrad der DBTT, ausgedrückt als Prozentanteil der belegten DBTT-Einträge

## EXTENDIBLE:

Angabe, ob die Satzart erweiterbar ist:

NO nicht erweiterbar

SCAN erweiterbar mit Parameter SCAN=YES

NOSCAN erweiterbar mit Parameter SCAN=NO

## NR OF DBTT EXTENTS:

Anzahl der momentan vorhandenen DBTT-Extents

## HIGHEST USED EXTENT:

Anzahl der momentan vorhandenen DBTT-Extents abzüglich der von hinten lückenlos gezählten völlig leeren DBTT-Extents. Diese Information kann für eine DBTT-Verkleinerung mittels BREORG nützlich sein.

## TOTAL

Die Gesamtsumme der Anzahl der belegten DBTT-Einträge wird ausgegeben.

Bei allen Angaben werden die im Falle der BMODTT-Anweisung KEEP gesperrten DBTT-Einträge nicht mitgezählt; als „belegt“ gelten nur diejenigen, für die Sätze gespeichert sind.

## CALC-Key-Statistik ausdrucken (DISPLAY CALC)

DISPLAY [IN CSV [*csv-dateiname*]] CALC KEY STATISTICS FOR

$$\left. \begin{array}{l} \left\{ \begin{array}{l} \text{RECORD} \\ \text{SEARCHKEY} \end{array} \right\} \left\{ \begin{array}{l} \{ \textit{satzname-1, \dots} \} \\ \{ \textit{keyref-1, \dots} \} \end{array} \right\} \left\{ \begin{array}{l} \text{ALL} \\ \text{ALL} \end{array} \right\} \end{array} \right\} \text{IN REALM} \left\{ \begin{array}{l} \{ \textit{realmname-1, \dots} \} \\ \text{ALL} \end{array} \right\}$$

### IN CSV

BSTATUS gibt die Daten zusätzlich im CSV-Format aus.

*csv-dateiname*

Name der Datei, in die die Daten im CSV-Format ausgegeben werden sollen. Die Angabe von *csv-dateiname* ist bei der ersten IN CSV-Anweisung einer BSTATUS-Ausführung zwingend (z.B. DISPLAY IN CSV 'BSTATUS.CSV' ...).

Eine ausführliche Beschreibung der Ausgabe im CSV-Format finden Sie im Handbuch „[Datenbankbetrieb](#)“ im Abschnitt „Datenbankinformationen systemneutral ausgeben“.

### RECORD

BSTATUS druckt die CALC-Key-Statistik für Satzarten aus, die mit LOCATION MODE IS CALC definiert wurden

*satzname*

Name einer Satzart, für die BSTATUS die CALC-Key-Statistik ausdrucken soll

ALL

BSTATUS druckt die CALC-Key-Statistik für alle mit LOCATION MODE IS CALC definierten Satzarten aus, die im angegebenen Subschema enthalten sind

### SEARCHKEY

BSTATUS druckt die CALC-Key-Statistik aus für Hashbereiche von CALC-SEARCH-Keys, die auf Satzart- oder Setebene definiert wurden

*keyref*

Nummer des CALC-SEARCH-Keys, dessen CALC-Key-Statistik BSTATUS ausdrucken soll.

Die Nummer können Sie dem SIA PRINT REPORT entnehmen (siehe [Seite 162](#))



**ALL**

BSTATUS druckt die CALC-Key-Statistik aus für alle Hashbereiche von CALC-SEARCH-Keys, die im Subschema enthalten sind

**REALM**

BSTATUS druckt die CALC-Key-Statistik für den oder die Realms aus

*realmname*

Name eines Realm, für den BSTATUS die CALC-Key-Statistik ausdrucken soll

**ALL**

BSTATUS druckt die CALC-Key-Statistik aus für alle Realms

*Beispiel*

DISPLAY CALC SEARCHKEY 12,14,15,16,17 IN REALM ARTIKELRLM.

STATUS OF CALC KEY BUCKETS - PER CALC KEY  
R E A L M : ARTIKELRLM

C A L C   K E Y S		NR OF PRIMARY BUCKETS	RECORDS/ POINTERS THAT CAN BE ADDED	NR OF EMPTY BUCKETS	FILLING % PRIMARY BUCKETS	NR OF OVERFLOW BUCKETS	RECORDS/ POINTERS IN OVERFLOW	FILLING % OVERFLOW BUCKETS	DEPTH FACTOR FOR RECORD
KEY-REF:	12	1	109	0	4	0	0	-	2,00
KEY-REF:	14	3	597	0	10	0	0	-	2,00
KEY-REF:	15	11	806	4	7	0	0	-	2,00
KEY-REF:	16	1	107	0	19	0	0	-	2,00
KEY-REF:	17	1	305	0	8	0	0	-	2,00

BSTATUS druckt die CALC-Key-Statistik für jeden Realm getrennt aus.

**RECORDS**

nur, wenn Sie DISPLAY CALC RECORD angeben:

Namen der Satzarten, für die BSTATUS die CALC-Key-Statistik ausdrückt

**CALC KEYS**

KEY REF: *keyref*

nur, wenn Sie DISPLAY CALC SEARCHKEY angeben:

Nummern der CALC-SEARCH-Keys, für die BSTATUS die CALC-Key-Statistik ausdrückt

**NR OF PRIMARY BUCKETS**

Anzahl der für den Hashbereich reservierten Primärseiten

## RECORDS/POINTERS THAT CAN BE ADDED

Anzahl der Sätze (bei einem direkten Hashbereich) bzw. der Adressverweise (bei einem indirekten Hashbereich), die in den Primärseiten des Hashbereichs noch gespeichert werden können

## NR OF EMPTY BUCKETS

Anzahl der leeren Primärseiten

## FILLING % PRIMARY BUCKETS

Füllgrad der Primärseiten

## NR OF OVERFLOW BUCKETS

Anzahl der angelegten Überlaufseiten

## RECORDS/POINTERS IN OVERFLOW

Anzahl der Sätze (bei einem direkten Hashbereich) bzw. der Adressverweise (bei einem indirekten Hashbereich) in den Überlaufseiten

## FILLING % OVERFLOW BUCKETS

Füllgrad der Überlaufseiten

## DEPTH FACTOR FOR RECORD

Tiefenfaktor für den Zugriff auf einen Satz, d.h. durchschnittliche Anzahl der zum Auffinden eines Satzes benötigten Zugriffe

BSTATUS berechnet den Tiefenfaktor nach folgenden Formeln:

- direkter Hashbereich:

$$t = \sum_{i=1}^n \frac{(\text{satzanzahl in seite}_i) \times \text{rang}_i}{\text{gesamte satzanzahl}}$$

- indirekter Hashbereich:

$$t = \sum_{i=1}^n \frac{(\text{anzahl adressverweise in seite}_i) \times \text{rang}_i}{\text{gesamtanzahl adressverweise}} + 1$$

$n$  Anzahl aller Seiten des Hashbereiches (Primärseiten und Überlaufseiten)

- rang<sub>i</sub>* Rang der i-ten CALC-Seite in einer Kette von Primärseiten und Überlaufseiten  
Primärseite: Rang 1  
1. Überlaufseite: Rang 2  
2. Überlaufseite: Rang 3  
etc.
- t* Tiefenfaktor
- gesamte satzanzahl*  
Anzahl aller Sätze, die in den Primärseiten und Überlaufseiten des Hashbereiches gespeichert sind
- gesamtanzahl adressverweise*  
Anzahl aller Adressverweise, die in den Primärseiten und Überlaufseiten des Hashbereiches gespeichert sind

## Satzanzahl-Statistik ausdrucken (DISPLAY RECORDNUMBER)

DISPLAY [IN CSV [*csv-dateiname*]] RECORDNUMBER STATISTICS FOR

$$\left. \begin{array}{l} \text{REALM} \left\{ \begin{array}{l} \textit{realmname-1}, \dots \\ \text{ALL} \end{array} \right\} \\ \text{RECORD} \left\{ \begin{array}{l} \textit{satzname-1}, \dots \\ \text{ALL} \end{array} \right\} \end{array} \right\}$$

### IN CSV

BSTATUS gibt die Daten zusätzlich im CSV-Format aus.

*csv-dateiname*

Name der Datei, in die die Daten im CSV-Format ausgegeben werden sollen. Die Angabe von *csv-dateiname* ist bei der ersten IN CSV-Anweisung einer BSTATUS-Ausführung zwingend (z.B. DISPLAY IN CSV 'BSTATUS.CSV' ...).

Eine ausführliche Beschreibung der Ausgabe im CSV-Format finden Sie im Handbuch „[Datenbankbetrieb](#)“ im Abschnitt „Datenbankinformationen systemneutral ausgeben“.

### REALM

BSTATUS druckt für jede Satzart, die in dem angegebenen Realm bzw. in den angegebenen Realms vorkommen kann, die Anzahl der dort gespeicherten Sätze aus

*realmname*

Name eines Realm, für den BSTATUS die Satzanzahl-Statistik ausdrucken soll

ALL

BSTATUS druckt die Satzanzahl-Statistik für alle Realms des Subschemas aus

### RECORD

BSTATUS druckt für jede angegebene Satzart die Anzahl der Sätze aus, die in den Realms, in denen die Satzart vorkommen kann, gespeichert sind

*satzname*

Name einer Satzart, für die BSTATUS die Satzanzahl-Statistik ausdrucken soll

ALL

BSTATUS druckt die Satzanzahl-Statistik für alle Satzarten des Subschemas aus

*Beispiel*

DISPLAY RECORDNUMBER STATISTICS FOR REALM ALL.

NUMBER OF RECORDS PER REALM

* RECORDS	* REALM REFS									
	3	4	5	6	7	8	9	10		
KUNDE	0	-	-	-	-	-	-	-	-	-
AUFTRAG	0	-	-	-	-	-	-	-	-	-
AUFTR-POS	0	-	-	-	-	-	-	-	-	-
RATENZAHLUNG	0	-	-	-	-	-	-	-	-	-
ARTIKELART	-	-	1	0	0	3	0	0		
ARTIKELAUSSWAHL	-	-	2	0	0	3	0	0		
ARTIKELBESCHR	-	-	8	0	0	5	0	0		
ARTIKEL	-	-	55	0	0	8	0	0		
TEILMENGE	-	-	-	0	0	-	-	-		
FARBEN	-	-	-	-	-	-	-	-		
MATERIALIEN	-	-	-	-	-	-	-	-		
LIEFERANT	-	1	-	-	-	-	-	-		
BESTELLUNG	-	0	-	-	-	-	-	-		
BESTELL-POS	-	0	-	-	-	-	-	-		

- AREA REF 3 = AUFTRAGSRM
- AREA REF 4 = BESTELLRLM
- AREA REF 5 = KLEIDUNG
- AREA REF 6 = HAUSHALT
- AREA REF 7 = SPORT
- AREA REF 8 = LEBENSMITTEL
- AREA REF 9 = SPIELE-HOBBY
- AREA REF 10 = SCHREIBWAREN

## RECORDS

Namen der Satzarten

## REALM REFS

Nummern der Realms; für jeden Realm legt BSTATUS eine Spalte an, in der er die Anzahl der in dem Realm gespeicherten Sätze auflistet

0: die Satzart kann in dem Realm enthalten sein, es ist jedoch kein Satz dort gespeichert

-: die Satzart kann in dem Realm nicht vorkommen

## AREA REF

Zuordnung der Realm-Nummern zu den Realm-Namen

**Eingabe der Anweisungen beenden (END)**

---

END

---

## 6.4 Kommandofolge zum Starten von BSTATUS

Die hier beschriebene Kommandofolge geht davon aus, dass UDS/SQL mit IMON installiert wurde (siehe Abschnitt „START-Kommandos der UDS/SQL-Programme“ in Kapitel 2 des Handbuchs „[Aufbauen und Umstrukturieren](#)“).

```
01 /ADD-FILE-LINK LINK-NAME=DATABASE,  
    FILE-NAME=[ :catid: ] [ $userid. ] dbname.DBDIR [ .copyname ]  
02 /SELECT-PRODUCT-VERSION PRODUCT-NAME=UDS-SQL, VERSION=version. SCOPE=*TASK  
03 /START-UDS-BSTATUS  
04 SUBSCHEMA IS subschemaname  
05 display-anweisungen  
06 END
```

- 01 Hier ist die Angabe von *:catid:* zulässig (siehe Handbuch „[Datenbankbetrieb](#)“).
- 02 Die Version des Dienstprogramms wird ausgewählt.  
Die Angabe der Version wird generell empfohlen, da mehrere UDS/SQL-Versionen parallel installiert sein können.
- 03 Das UDS/SQL-Dienstprogramm kann auch mit dem Aliasnamen BSTATUS gestartet werden.





---

## 7 Inhalt der Realms ausdrucken mit BPRECORD

BPRECORD erstellt einen Abzug Ihrer Datenbank.

Mit dessen Hilfe können Sie ermitteln:

- aus der Kennseite das letzte Änderungsdatum eines Realm und die Systembreak-Informationen
- aus den FPA-Einträgen die Lage und Größe freier Bereiche in den Realms (wichtig für das Reorganisieren eines Realm)
- aus den DBTTs die freien Satzfolgennummern (wichtig bei LOCATION MODE IS DIRECT bzw. DIRECT-LONG)
- aus dem Abdruck von Hashbereichen die Verteilung der Sätze über den Hashbereich (Grundlage für das Verbessern eigener Hashroutinen)
- aus dem Abdruck von Datensätzen und Tabellen die Verbindung von Datensätzen mit Tabellen (Hilfsmittel für die Fehlersuche in DB-Anwendungen).

Mit BPRECORD können Sie sich von Ihrer Datenbank den Inhalt der Benutzerrealms, des Database Directorys (DBDIR) oder des Database Compiler Realm (DBCOM) ausdrucken.

Einzelne Teile eines Realm können Sie dabei nach zwei Gesichtspunkten auswählen:

logisch      hierbei wählen Sie beispielsweise eine Satzart, bestimmte Sätze einer Satzart, eine Tabelle o. Ä.

physisch    hierbei wählen Sie beispielsweise den Realm, bestimmte Seitenarten, bestimmte Seitennummern o. Ä.

Beide Auswahlmöglichkeiten dürfen Sie kombinieren. Außerdem können Sie bei der Ausgabe von CALC-Seiten und Datenseiten wählen, ob BPRECORD den Seitenkopf (PAGE INFO), die Seitenindex-Einträge (PAGE INDEX) oder die Set-Connection-Data (SCD) der Sätze mit ausgibt (siehe [Seite 245](#)).

Aus Gründen des Datenschutzes kann BPRECORD nur unter der Kennung des Datenbankadministrators aufgerufen werden.

Zusätzlich zur Ausgabe auf SYSLST können Sie die Daten auch im CSV-Format in eine Datei ausgeben lassen. Mit dem CSV-Format wird die Weiterverarbeitung von Daten in anderen Systemumgebungen (z.B. bei Spreadsheet-Anwendungen) erleichtert. Die Ausgabe im CSV-Format ist im Handbuch „[Datenbankbetrieb](#)“ im Abschnitt „Datenbankinformationen systemneutral ausgeben“ beschrieben.



Wenn Sie BPRECORD online ausführen, müssen Sie damit rechnen, dass die ausgegebenen Daten nicht aktuell sind, da der DBH noch nicht alle Daten aus dem Puffer in die Datenbank geschrieben hat. Um weitgehend aktuelle Daten zu erhalten, sollten Sie unmittelbar vor dem BPRECORD-Lauf mit dem DAL-Kommando CHECKPOINT oder NEW RLOG eine Aktualisierung der Datenbank erzwingen. Aber auch danach sind noch Abweichungen der BPRECORD-Ausgabe vom tatsächlichen Stand der Datenbank möglich, falls parallel ein Update durchgeführt wird.

## 7.1 Systemumgebung

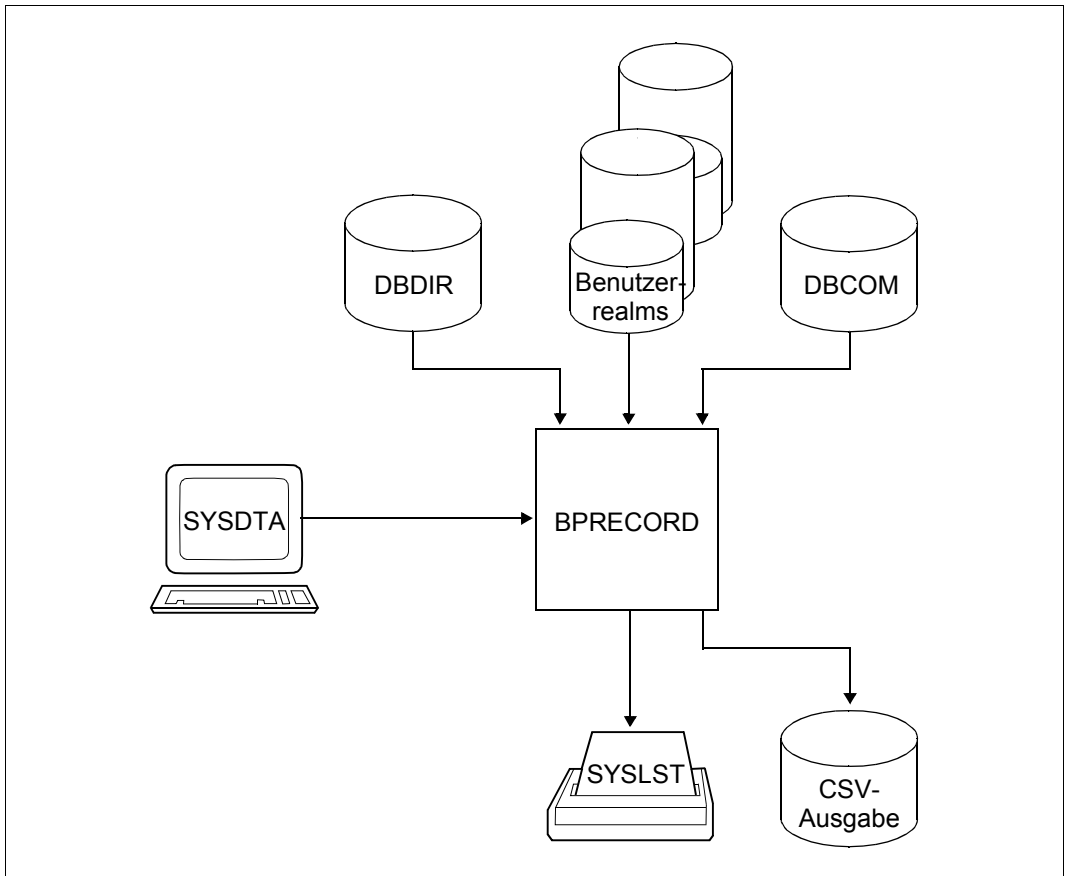


Bild 16: Systemumgebung von BPRECORD

BPRECORD berücksichtigt beim Start ggf. eine zugewiesene UDS/SQL-Pubset-Deklaration (siehe Handbuch „[Datenbankbetrieb](#)“, Pubset-Deklarations-Jobvariable). Eine fehlerhafte Zuweisung führt zum Programmabbruch.

## 7.2 Allgemeine Beschreibung der Ausgabe von BPRECORD

Jede der fünf verschiedenen Seitenarten, in die ein Realm eingeteilt ist, kann BPRECORD einzeln oder kombiniert ausdrucken. Dabei wählt er folgende Reihenfolge:

1.

\*\*\*\*\* ACTUAL-KEY=0 OF REALM < :SQL2:\$XXXXXXXX.VERSAND.ARTIKELRLM

PHYS REC LENGTH : 4000

DATASET INFO :

TOTAL NR PAGES: 2062  
 FPA BASE BEGIN: AREA REF: 11  
 BNR: 1  
 NR PAGES IN FPA BASE: 1990

FPA EXTENTS

NR BNR  
 1 47  
 MAX NR PAGES IN EXTENT: 63680

HIGHEST PAGE NR FOR FORMATTING: 2061

CREATE DATA: DATE : 20150628 TIME: 092708

BACK UP DATA: DATE : 20150628 TIME: 092747

REALM VERSION NR: 3

SYSTEM BREAK:

OCCURRED: 0  
 ADMIN USERID: \$XXXXXXXX  
 CONFNAME: BREORG

FILE NAME: :SQL2:\$XXXXXXXX.VERSAND.ARTIKELRLM

REALM LAYOUT VERSION: 004.00

UDS VERSION: V2.8

INCR NR PAGES: 64

INCR MIN PAGES: 0

2.

\*\*\*\*\* F.P.A. ENTRIES OF REALM < :SQL2:\$XXXXXXXX.VERSAND.ARTIKELRLM

	REALM	REF	11	BNR	1	ACT	KEY	X'0B000001'		
0.....29:	0	0	0	0	0	0	0	0	0	0
30.....39:	0	0	3960	3964	0	0	3956	0	0	0
40.....49:	0	0	0	0	0	0	2800	0	0	0
50.....69:	0	0	0	0	0	0	0	0	0	0
70.....79:	0	0	0	0	0	0	0	0	0	3980
80.....1989:	3980	3980	3980	3980	3980	3980	3980	3980	3980	3980
			REALM	REF	11	BNR	47	ACT <td>KEY <td>X'0B00002F'</td> </td>	KEY <td>X'0B00002F'</td>	X'0B00002F'
1990.....2059:	3980	3980	3980	3980	3980	3980	3980	3980	3980	3980
2060.....2061:	3980	0								

3.

```
***** D.B.T.T. ENTRIES OF REALM < :SQL2:$XXXXXXXXX.VERSAND.ARTIKELRLM > *****
===== RECORD REF      6, NAME: < ARTIKELART > =====
----- REALM REF  11  BNR      3  ACT KEY  X'0B000003' -----

      RSQ      1/X'00000001': ( 5,      12/X'0500000C') ( 11,      37/X'0B000025')
      RSQ      2/X'00000002': ( 8,      4/X'08000004') ( 11,      39/X'0B000027')
      RSQ      3/X'00000003': ( 8,      5/X'08000005') ( 11,      41/X'0B000029')
      RSQ      4/X'00000004': ( 8,      6/X'08000006') ( 11,      44/X'0B00002C')
RSQ      5/X'00000005'- 11/X'0000000B': ( 0,      0/X'00000000') ( 0,      0/X'00000000')
```

4.

```
***** CALC KEY BUCKETS OF REALM < :SQL2:$XXXXXXXXX.VERSAND.ARTIKELRLM > *****
===== CALC SEARCH KEY, KEY REF  14, SET NAME < SYS_ARTIKEL > =====
----- REALM REF  11  BNR      16  ACT KEY  X'0B000010' -----

PAGE INFO:  TYPE 0  FREE SPACE  SIZE  10,  DISPL  30  NR OF PAGE INDICES  0  DISPL TO CALC TABLE HEADER  30

CALC KEY TABLE:  MAX ENTRIES  220  ACT ENTRIES  25  OVERFLOW BUCKET  NEXT  0,  PRIOR  0

-1-
(  1) (00000000) F2F3F0F1 F0F7F3F6 23010736
      RSQ      1  PPP  5,  13
-2-
(  1) (00000000) F2F3F0F1 F0F7F4F2 23010742
      RSQ      4  PPP  5,  13
-3-
(  1) (00000000) F2F3F0F1 F0F7F4F8 23010748
      RSQ      7  PPP  5,  13
.
.
.
```

5.

```
***** DATA / TABLE PAGES OF REALM < :SQL2:$XXXXXXXXX.VERSAND.KLEIDUNG > *****
===== REALM REF  5  BNR      19  ACT KEY  X'05000013' -----

PAGE INFO:  TYPE 0  FREE SPACE  SIZE  2081,  DISPL  2209  NR OF PAGE INDICES  9  DISPL TO END OF PAGE  4000

LOGICAL RECORDS:
-2- PAGE INDEX:  DB_KEY  9,      48  COL-NR  0  LIST REC  DISPL  3767
(  1) (00000000) 00090000 00000030 05000013 00090000 00000030 05000013 00090000 00000030
(  33) (00000020) 05000013 00090000 00000030 05000013 00090000 00000030 05000013 00090000
(  65) (00000040) 00000030 05000013 00090000 00000030 05000013 00000000 000A0500 0012FF00
(  97) (00000060) 00000000 00000000 00000000 00000000 00000000 00000000 00000001 04000007
( 129) (00000080) F8F3F5F9 F2F8F0F9 E360E2C8 C9D9E360 D2D3C5C9 C4404040 40404040 40404040 83592809T-SHIRT-KLEID
( 161) (000000A0) 40404040 40404040 40404040 40404040 F2F3F7F1 F0F2F3F6 0020700C 0023900C 23710236
( 193) (000000C0) 00000500 000C050C 00000499 950C0000 00000000 050C00
```

Der Aufbau und die Bedeutung der verschiedenen Seitenarten ist im Handbuch „Entwerfen und Definieren“ beschrieben.

SCHEMA NAME *schemaname*

Name des Schemas, dessen Realms BPRECORD anschließend ausdrückt

ACTUAL-KEY-0 OF REALM *realmname*

Kennseite des Realm *realmname*; BPRECORD druckt wahlweise die Kennseite im Klartext aus

F.P.A. ENTRIES OF REALM *realmname*

Einträge der Freiplatzverwaltung (Free Place Administration) des Realm *realmname*; BPRECORD druckt wahlweise alle FPA-Einträge oder nur die FPA-Einträge bestimmter Seiten aus

REALM REF *realmref* BNR *snr*

vor jedem Seiteninhalt druckt BPRECORD die Adresse der Seite aus

*realmref*: Nummer des Realm

*snr*: Seitennummer

D.B.T.T. ENTRIES OF REALM *realmname*

DBTT-Einträge des Realm *realmname*; BPRECORD druckt wahlweise aus:

- alle DBTTs,
- nur die DBTT einer bestimmten Satzart oder
- nur die DBTT-Einträge bestimmter Sätze einer Satzart

RECORD REF *refref*, NAME *satzname*:

mit dieser Überschrift beginnt die DBTT einer Satzart

*refref*: Nummer der Satzart

*satzname*: Name der Satzart

CALC KEY BUCKETS OF REALM *realmname*

direkte CALC-Seiten bzw. indirekte CALC-Seiten (Primärseiten und Überlaufseiten) des Realm *realmname*

CALC RECORD, REC REF *refref*, RECORD NAME *satzname*

direkte CALC-Seiten bzw. indirekte CALC-Seiten der Satzart *satzname*

CALC SEARCH KEY, KEY REF *keyref*, SET NAME *setname*

CALC-Seiten des CALC-SEARCH-Keys *keyref*

*keyref*: Nummer des Schlüssels

*setname*: Name des zugehörigen Sets

DATA/TABLE PAGES OF REALM *realmname*

Datenseiten des Realm *realmname*, die Sätze (außer CALC-Sätzen) oder Tabellen bzw. Tabelleneinträge enthalten.

BPRECORD druckt nur nicht leere Seiten. Um zu entscheiden, ob eine Seite leer ist, prüft BPRECORD die FPA-Einträge.

CALC-Seiten und Datenseiten sortiert er nach aufsteigenden Seitennummern.

## 7.3 Anweisungen für BPRECORD

Anweisung	Bedeutung
SCHEMA NAME	Name des Schemas, das den auszugebenden Realm enthält (wahlweise)
REALM NAME	Name des auszugebenden Realm (immer erforderlich)
PRINT	Umfang der Ausgaben festlegen (wahlweise)
DISPLAY PAGE	Act-Key-0-Seite ausdrucken (wahlweise)
DISPLAY FPA	FPA-Einträge auflisten (wahlweise)
DISPLAY DBTT	DBTT-Einträge auflisten (wahlweise)
DISPLAY CALC	CALC-Seiten ausdrucken (wahlweise)
DISPLAY DATA	Datenseiten ausdrucken (wahlweise)
END	den BPRECORD-Lauf beenden (immer erforderlich)

Tabelle 17: Anweisungen für BPRECORD

Alle zu einem Realm gehörenden DISPLAY-Anweisungen sammelt BPRECORD und sortiert sie, um doppelte Druckausgaben zu vermeiden.

Die Anweisungen können über mehrere Zeilen gehen. Jede Zeile ist aber auf 72 Stellen beschränkt. Ein Fortsetzungszeichen ist bei mehrzeiliger Schreibweise nicht erforderlich.



PRINT-Anweisungen und DISPLAY-Anweisungen sollten Sie - zur besseren Übersicht - durch ein Semikolon voneinander trennen.

Ein Punkt als Trennsymbol bedeutet den Abschluss der zu einem Realm gehörenden Anweisungen. BPRECORD erwartet danach eine neue REALM-Anweisung oder die END-Anweisung.

Nach SCHEMA-NAME/REALM-NAME ist mindestens eine DISPLAY-Anweisung notwendig.

*Beispiel für eine Anweisungsfolge*

```

SCHEMA NAME IS schemaname.
REALM NAME IS realmname-1.
PRINT WITH SCD;
DISPLAY DATA PAGES ALL PAGES ALL TABLES;
DISPLAY DBTT OF ALL RECORDS.
REALM NAME IS realmname-2.
PRINT WITH PAGEINDEX;
DISPLAY FPA OF ALL PAGES;
DISPLAY CALC PAGES ALL PAGES ALL RECORDS.
END

```

### Physische Auswahl (Seitenauswahl)

Die Syntaxelemente *seitenauswahl* und *rsq-auswahl* werden in mehreren Anweisungen genutzt:

---


$$seitenauswahl := \left\{ \begin{array}{l} \underline{ALL} \ \underline{PAGES} \\ \underline{PAGE} \ \{snr-1[ \ \underline{TO} \ snr-2]\}, \dots \end{array} \right\}$$


---

#### ALL PAGES

alle Seiten der mit der logischen Auswahl festgelegten Menge

#### PAGE *snr-1*,...

Liste mit Seitennummern

#### PAGE {*snr-1 TO snr-2*},...

Bereich von Seitennummer *snr-1* bis Seitennummer *snr-2*, usw.

### Logische Auswahl (RSQ-Auswahl)

---


$$rsq-auswahl := \left\{ \begin{array}{l} \underline{ALL} \ \underline{RSQS} \\ \underline{RSQ} \ \{rsq-1[ \ \underline{TO} \ rsq-2]\}, \dots \end{array} \right\}$$


---

#### ALL RSQS

alle Satzfolgennummern

#### RSQ *rsq-1*,...

Liste mit Satzfolgennummern

#### RSQ {*rsq-1 TO rsq-2*},...

Bereich von Satzfolgennummer *rsq-1* bis Satzfolgennummer *rsq-2*, usw.



## SCHEMA bezeichnen (SCHEMA NAME)

---

SCHEMA NAME IS *schemaname*.

---

### *schemaname*

Name des Schemas, das die Beschreibung des auszugebenden Realm bzw. der auszugebenden Realms enthält; für *schemaname* können Sie Folgendes angeben:

- *benutzerschemaname* für die Ausgabe eines Benutzerrealms
- COMPILER-SCHEMA für die Ausgabe des DBCOM
- PRIVACY-AND-IQF-SCHEMA für die Ausgabe des DBDIR

Standardwert:

Benutzerschema

Die SCHEMA-Anweisung ist wahlweise anzugeben. Geben Sie sie an, so muss sie die erste Anweisung für BPRECORD sein und darf nur einmal angegeben werden.

BPRECORD greift auf die SIA des angegebenen Schemas zu und entnimmt ihr alle für den Zugriff auf die Datenbank notwendigen Informationen.

## Auszugebenden Realm angeben (REALM NAME)

---

`REALM NAME IS realmname.`

---

*realmname*

Name des auszugebenden Realm; Sie geben an für

- Benutzerrealms: den in der Schema-DDL per AREA-Klausel definierten Realm-Namen
- das DBDIR: DATABASE-DIRECTORY
- den DBCOM: DATABASE-COMPILER-REALM

Die Anweisung REALM müssen Sie mindestens einmal angeben, Sie dürfen sie auch mehrmals angeben. Beachten Sie dabei aber, dass alle Anweisungen PRINT und DISPLAY, die sich auf einen Realm beziehen, unmittelbar auf die entsprechende REALM-Anweisung folgen müssen! Die erste REALM-Anweisung muss unmittelbar auf die SCHEMA-Anweisung folgen oder, wenn diese fehlt, als erste Anweisung für BPRECORD eingegeben werden (siehe [Seite 239](#)).

## Umfang der Ausgaben festlegen (PRINT)

---

```
PRINT [ { WITH } PAGEINFO [ { WITH } PAGEINDEX ]
      [ { WITHOUT } SCD ] [ DBTT { DEC }
      [ { WITHOUT } { HEX }
      [ { WITHOUT } { BOTH } ] ] ;
```

---

### PAGEINFO

Seitenkopf

### PAGEINDEX

Seitenindex-Einträge

SCD Set-Connection-Data

PAGEINFO, PAGEINDEX und SCD gelten für die Ausgabe von CALC- und Datenseiten.

DEC dezimal

HEX sedezimal

BOTH dezimal und sedezimal

DEC, HEX und BOTH gelten für die Ausgabe von DBTTs.

Standardwerte:

WITHOUT und DEC

Die PRINT-Anweisung können Sie wahlweise angeben. Sie bezieht sich auf alle DISPLAY-Anweisungen, die Sie zwischen zwei REALM-Anweisungen bzw. zwischen einer REALM-Anweisung und der END-Anweisung eingeben.

Geben Sie die PRINT-Anweisung mehrfach an, so ist die letzte gültig.

*Beispiel*

PRINT WITH PAGEINFO WITH PAGEINDEX WITH SCD

```
***** SCHEMA NAME < ARTIKELVERSAND > *****
***** DATA / TABLE PAGES OF REALM < :SQL2:$XXXXXXX.VERSAND.KLEIDUNG > *****
----- REALM REF 5 BNR 18 ----- ACT KEY X'05000012' -----
```

PAGE INFO: TYPE 0 — FREE SPACE SIZE 1963, DISPL 2103 — NR OF PAGE INDICES 10 — DISPL TO END OF PAGE 4000

LOGICAL RECORDS:

```
-2- PAGE INDEX: DB_KEY 9, 41 — COL-NR 0 — LIST REC — DISPL 3767
( 1) (00000000) 00090000 00000029 05000012 00090000 00000029 05000012 00090000 00000029
( 33) (00000020) 05000012 00090000 00000029 05000012 00090000 00000029 05000012 00090000
( 65) (00000040) 00000029 05000012 00090000 00000029 05000012 00000000 00090500 0011FF00
( 97) (00000060) 00000000 00000000 00000000 00000000 00000000 00000000 00000001 04000007
( 129) (00000080) F8F3F1F2 F1F3F1F1 D2D3C5C9 C440C1E4 E240D1C5 D9E2C5E8 60D2D9C5 D7D740D4 83121311KLEID AUS JERSEY-KREPP M
( 161) (000000A0) C9E340D1 C1C3D2C5 40404040 40404040 F2F3F6F1 F1F1F3F6 0020700C 0023900C IT JACKE 23611136
( 193) (000000C0) 00000500 000C050C 00000499 950C0000 00000000 050C00
```

```
-3- PAGE INDEX: DB_KEY 9, 42 — COL-NR 0 — LIST REC — DISPL 3552
( 1) (00000000) 00090000 0000002A 05000012 00090000 0000002A 05000012 00090000 0000002A
( 33) (00000020) 05000012 00090000 0000002A 05000012 00090000 0000002A 05000012 00090000
( 65) (00000040) 0000002A 05000012 00090000 0000002A 05000012 00000000 00090500 0011FF00
( 97) (00000060) 00000000 00000000 00000000 00000000 00000000 00000000 00000001 04000007
( 129) (00000080) F8F3F1F2 F1F3F1F1 D2D3C5C9 C440C1E4 E240D1C5 D9E2C5E8 60D2D9C5 D7D740D4 83121311KLEID AUS JERSEY-KREPP M
( 161) (000000A0) C9E340D1 C1C3D2C5 40404040 40404040 F2F3F6F1 F1F1F3F8 0020700C 0023900C IT JACKE 23611138
( 193) (000000C0) 00000500 000C050C 00000499 950C0000 00000000 050C00
```

```
-4- PAGE INDEX: DB_KEY 9, 43 — COL-NR 0 — LIST REC — DISPL 3337
( 1) (00000000) 00090000 0000002B 05000012 00090000 0000002B 05000012 00090000 0000002B
( 33) (00000020) 05000012 00090000 0000002B 05000012 00090000 0000002B 05000012 00090000
( 65) (00000040) 0000002B 05000012 00090000 0000002B 05000012 00000000 00090500 0011FF00
( 97) (00000060) 00000000 00000000 00000000 00000000 00000000 00000000 00000001 04000007
( 129) (00000080) F8F3F1F2 F1F3F1F1 D2D3C5C9 C440C1E4 E240D1C5 D9E2C5E8 60D2D9C5 D7D740D4 83121311KLEID AUS JERSEY-KREPP M
( 161) (000000A0) C9E340D1 C1C3D2C5 40404040 40404040 F2F3F6F1 F1F1F4F0 0020700C 0023900C IT JACKE 23611140
( 193) (000000C0) 00000500 000C050C 00000499 950C0000 00000000 050C00
```

.  
.  
.

[ Seitenkopf

[ Seitenindex-Eintrag

[ Set Connection Data (SCD)

[ Datensatz

**PAGE INFO (Seitenkopf)**

Bei allen Seiten - außer DBTT-Seiten und FPA-Seiten - enthalten die ersten 20 byte den Seitenkopf. Davon druckt BPRECORD aus:

TYPE Art der Seite

0: Datenseite oder CALC-Seite

1: Act-Key-0-Seite

FREE SPACE

freier Speicherplatz in der Seite:

SIZE

Länge des freien Speicherplatzes (in byte)

DISPL

Distanz zum ersten freien Byte

NO OF PAGE INDICES

Anzahl der Seitenindex-Einträge

DISPL TO END OF PAGE

Länge der Seite (in byte)

**PAGE INDEX (Seitenindex-Eintrag)**

Seitenindex-Einträge ermöglichen das Auffinden eines Satzes bzw. einer Tabelle innerhalb der Seite. Sie kommen vor in Datenseiten und direkten CALC-Seiten. BPRECORD druckt davon aus:

DB-KEY

Database Key eines Satzes in der Form: *refref, rsq*

COL-NR

Spaltennummer in der DBTT

=0: Datensatz (LOGICAL RECORD oder CALC KEY REC)

>0: Tabellensatz (TABLE REC)

DISPL Distanz zum Datensatz

**SCD (Set-Connection-Data)**

Die Erklärung der Set-Connection-Data finden Sie im Handbuch „[Entwerfen und Definieren](#)“.

## Kennseite ausdrucken (DISPLAY PAGE)

---

```
DISPLAY [IN CSV [csv-dateiname]] PAGE ZERO;
```

---

Die DISPLAY-PAGE-ZERO-Anweisung druckt die ersten 108 byte der Kennseite (Act-Key-0-Seite) im Klartext aus. Sie können Sie wahlweise angeben. Wiederholte Eingabe dieser Anweisung für einen Realm ignoriert BPRECORD.

### IN CSV

BPRECORD gibt die Daten zusätzlich im CSV-Format aus.

#### *csv-dateiname*

Name der Datei, in die die Daten im CSV-Format ausgegeben werden sollen. Die Angabe von *csv-dateiname* ist bei der ersten IN CSV-Anweisung einer BPRECORD-Ausführung zwingend (z.B. DISPLAY IN CSV 'BPRECORD.CSV' ...).

Eine ausführliche Beschreibung der Ausgabe im CSV-Format finden Sie im Handbuch „[Datenbankbetrieb](#)“ im Abschnitt „Datenbankinformationen systemneutral ausgeben“.

### Beispiel

```
DISPLAY PAGE ZERO
```

```
***** ACTUAL-KEY=0 OF REALM < :SQL2:$XXXXXXXX.VERSAND.ARTIKELRLM
PHYS REC LENGTH :      4000
DATASET INFO :
TOTAL NR PAGES:          2062
FPA BASE BEGIN:          AREA REF:      11
                        BNR:            1
NR PAGES IN FPA BASE:    1990
FPA EXTENTS
      NR      BNR
      1      47
MAX NR PAGES IN EXTENT:  63680

HIGHEST PAGE NR FOR FORMATTING:  2061
CREATE DATA:  DATE :      20150628  TIME:    112614
BACK UP DATA:  DATE :      20150628  TIME:    112642
REALM VERSION NR:      3
SYSTEM BREAK:
OCCURRED:      0
ADMIN USERID:  $XXXXXXXX
CONFNAME:      BREORG
FILE NAME:      :SQL2:$XXXXXXXX.VERSAND.ARTIKELRLM
REALM LAYOUT VERSION:  004.00
UDS VERSION:      V2.8
INCR NR PAGES:      64
INCR MIN PAGES:      0
```

## PHYS REC LENGTH

Seitenlänge (in byte)

## DATASET INFO

allgemeine Informationen über den Realm:

## FPA BASE

Angaben zur FPA-Basis

## AREA REF

Nummer des Realms, für den diese Freiplatzverwaltung in der FPA-Basis gilt.

## BNR

Seitennummer der ersten FPA-Seite

## TOTAL NR PAGES

Anzahl der insgesamt im Realm verwalteten Seiten

## NR PAGES FPA BASE

Anzahl der über die FPA-Basis verwalteten Seiten des Realms

## FPA EXTENTS

Angaben zu allen FPA-Extents des Realms

## NR

Nummer des FPA-Extents

## BNR

Nummer der ersten Seite des FPA-Extents

## MAX NR PAGES IN EXTENT

Anzahl der maximal in einem FPA-Extent verwalteten Datenseiten

## HIGHEST PAGE NR FOR FORMATTING

Nummer der Datenbankseite, bis zu der bei einer Realm-Erweiterung eine Formatierung erfolgt

## nnnnnnnn

Nummer der Datenbankseite, bis zu der bei einer Realm-Erweiterung eine Formatierung erfolgt.

Ist der Wert identisch mit dem Wert von TOTAL NR PAGES, werden bei einer Realm-Erweiterung neu hinzukommende Seiten nicht formatiert.

Ist der Wert größer als der Wert von TOTAL NR PAGES, werden bei einer Realm-Erweiterung neu hinzukommende Seiten bis einschließlich der genannten formatiert.

Bei 2 KB Datenbanken müssen aus Gründen der Datensicherheit noch weitere Bedingungen erfüllt sein, damit auf die Formatierung neu hinzukommender Seiten verzichtet werden kann.

## UNKNOWN

Bei einer Realm-Erweiterung werden alle neu hinzukommenden Seiten formatiert.

## CREATE DATA

Erstellungszeitpunkt des Realm:

## DATE

Erstellungsdatum im Format: *jjjjmmtt*

## TIME

Erstellungsuhrzeit im Format: *hhmmss*

## BACK UP DATA

Zeitpunkt der letzten Änderung (Format siehe CREATE DATA)

## DATE

letztes Änderungsdatum

## TIME

Uhrzeit der letzten Änderungen

## REALM VERSION NR

interne Versionsnummer des Realm; sie wird nur von ändernden Dienstprogrammen erhöht (z.B. BALTER, BCHANGE, BOUTLOAD, BREORG)

## SYSTEM BREAK

kennzeichnet, ob der Realm ordnungsgemäß geschlossen wurde

## OCCURRED

0: letzte Session ordnungsgemäß abgeschlossen

1: letzte Session abgebrochen, d.h. der Realm ist u.U. inkonsistent

## ADMIN USERID

Kennung, von der aus die Datenbank verwaltet wird

## CONFNAME

Konfigurationsname oder Dienstprogrammname, unter dem zuletzt auf die Datenbank zugegriffen wurde

## FILE NAME

vollständiger Dateiname des Realm

## DATABASE LAYOUT VERSION

Versionsnummer der vorliegenden Datenbankstruktur im Format nnn.nn  
(wird für den DBDIR ausgegeben)

## REALM LAYOUT VERSION

Versionsnummer der vorliegenden Realm-Layout-Struktur im Format nnn.nn



**UDS VERSION**

UDS/SQL-Versionsnummer, unter der die Datenbank letztmalig vom DBH geändert wurde. Das Feld dient der korrekten Durchführung eines Warmstarts.

**INCR**

Angaben über die Online-Erweiterbarkeit des Realms. Diese Angaben werden für den DBDIR und die Benutzerrealms ausgegeben.

Die Angaben werden nur für Realms ausgegeben, für die mit dem DAL-Kommando ACT INCR die Online-Erweiterbarkeit aktiviert wurde.

Die Ausgabe der Angaben erfolgt auch, wenn die Online-Erweiterbarkeit deaktiviert wurde.

**NR PAGES**

Anzahl der Seiten, die dem Realm bei einer Online-Erweiterung hinzugefügt werden.

**MIN PAGES**

Anzahl der freien Seiten, bei deren Unterschreitung eine Online-Erweiterung des betreffenden Realms durch den DBH angestoßen wird.

## FPA-Einträge auflisten (DISPLAY FPA)

`DISPLAY [IN CSV [csv-dateiname]] FPA OF seitenauswahl;`

### IN CSV

BPRECORD gibt die Daten zusätzlich im CSV-Format aus.

#### *csv-dateiname*

Name der Datei, in die die Daten im CSV-Format ausgegeben werden sollen. Die Angabe von *csv-dateiname* ist bei der ersten IN CSV-Anweisung einer BPRECORD-Ausführung zwingend (z.B. DISPLAY IN CSV 'BPRECORD.CSV' ...).

Eine ausführliche Beschreibung der Ausgabe im CSV-Format finden Sie im Handbuch „Datenbankbetrieb“ im Abschnitt „Datenbankinformationen systemneutral ausgeben“.

#### *seitenauswahl*

siehe „[Physische Auswahl \(Seitenauswahl\)](#)“ auf Seite 240


Die DISPLAY FPA-Anweisung druckt die FPA-Einträge eines Realm ganz oder teilweise aus. Sie können sie wahlweise angeben.

Geben Sie mehrere FPA-Anweisungen für einen Realm an, so fasst BPRECORD sie intern zu einer Eingabe zusammen.


### Beispiel

DISPLAY FPA OF ...

```
***** F.P.A. ENTRIES OF REALM < :SQL2:$XXXXXXXXX.VERSAND.ARTIKELRLM
----- REALM REF 11 BNR 1 ----- ACT KEY X'0B000001' -----
0.....29: 0 0 0 0 0 0 0 0 0 0
30.....39: 0 0 3960 3964 0 0 3956 0 0 0
40.....49: 0 0 0 0 0 0 2800 0 0 0
50.....69: 0 0 0 0 0 0 0 0 0 0
70.....79: 0 0 0 0 0 0 0 0 0 3980
80.....1989: 3980 3980 3980 3980 3980 3980 3980 3980 3980 3980
----- REALM REF 11 BNR 47 ----- ACT KEY X'0B00002F' -----
1990.....2059: 3980 3980 3980 3980 3980 3980 3980 3980 3980 3980
2060.....2061: 3980 0
```



Seitennummern



freie Bytes pro Seite  
0 Seite gefüllt  
1 ... 3979 Seite teilweise gefüllt (Wert bei 4 Kbyte)  
3980 Seite leer

Unzulässige FPA-Werte werden mit XX gekennzeichnet.

## DBTT-Einträge auflisten (DISPLAY DBTT)

---

```
DISPLAY [IN CSV [csv-dateiname]] DBTT OF { ALL RECORDS
                                             RECORD satzname FOR rsq-auswahl };
```

---

### IN CSV

BPRECORD gibt die Daten zusätzlich im CSV-Format aus.

#### *csv-dateiname*

Name der Datei, in die die Daten im CSV-Format ausgegeben werden sollen. Die Angabe von *csv-dateiname* ist bei der ersten IN CSV-Anweisung einer BPRECORD-Ausführung zwingend (z.B. DISPLAY IN CSV 'BPRECORD.CSV' ...).

Eine ausführliche Beschreibung der Ausgabe im CSV-Format finden Sie im Handbuch „[Datenbankbetrieb](#)“ im Abschnitt „Datenbankinformationen systemneutral ausgeben“.

### ALL RECORDS

die DBTTs aller Satzarten ausdrucken, die in dem angegebenen Realm enthalten sind

#### *satzname*

Name einer Satzart, deren DBTT auszudrucken ist (bzw. deren DBTT-Einträge auszudrucken sind)

#### *rsq-auswahl*

siehe „[Logische Auswahl \(RSQ-Auswahl\)](#)“ auf Seite 240

Die DISPLAY DBTT-Anweisung können Sie wahlweise und auch mehrfach angeben. Mehrere gleiche DISPLAY DBTT-Anweisungen ignoriert BPRECORD.



Geben Sie in der DISPLAY DBTT-Anweisung mehrere Satzfolgennummern an, so müssen Sie diese in aufsteigender Reihenfolge angeben!

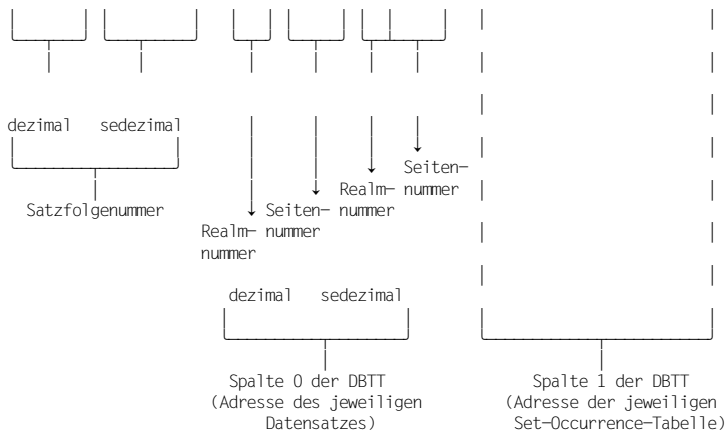
DBTT-Einträge können Sie dezimal und sedezeimal, dezimal oder sedezeimal auflisten lassen (siehe PRINT-Anweisung).

*Beispiel*

DISPLAY DBTT OF RECORD ARTIKELART FOR RSQ 1 TO 11 (mit PRINT DBTT BOTH)

```
***** D.B.T.T. ENTRIES OF REALM < :SQL2:$XXXXXXX.VERSAND.ARTIKELRLM
===== RECORD REF      6, NAME: < ARTIKELART                > =====
----- REALM REF  11  BNR      3  ACT KEY  X'0B000003' -----

RSQ      1/X'00000001': ( 5,      12/X'0500000C') ( 11,      37/X'0B000025')
RSQ      2/X'00000002': ( 8,      4/X'08000004') ( 11,      39/X'0B000027')
RSQ      3/X'00000003': ( 8,      5/X'08000005') ( 11,      41/X'0B000029')
RSQ      4/X'00000004': ( 8,      6/X'08000006') ( 11,      44/X'0B00002C')
RSQ      5/X'00000005'- 11/X'0000000B': ( 0,      0/X'00000000') ( 0,      0/X'00000000')
```



## CALC-Seiten ausdrucken (DISPLAY CALC)

DISPLAY [IN CSV [*csv-dateiname*]] CALC PAGES *seitenauswahl*

$$\left. \begin{array}{l} \text{ALL} \left[ \begin{array}{l} \text{RECORDS} \\ \text{CALC\_SEARCHKEYS} \end{array} \right] \\ \text{ONLY} \left[ \begin{array}{l} \text{RECORD } \textit{satzname} \\ \text{CALC\_SEARCHKEY } \textit{keyref} \end{array} \right] \textit{rsq-auswahl} \end{array} \right\} ;$$

### IN CSV

BPRECORD gibt die Daten zusätzlich im CSV-Format aus.

*csv-dateiname*

Name der Datei, in die die Daten im CSV-Format ausgegeben werden sollen. Die Angabe von *csv-dateiname* ist bei der ersten IN CSV-Anweisung einer BPRECORD-Ausführung zwingend (z.B. DISPLAY IN CSV 'BPRECORD.CSV' ...).

Eine ausführliche Beschreibung der Ausgabe im CSV-Format finden Sie im Handbuch „[Datenbankbetrieb](#)“ im Abschnitt „Datenbankinformationen systemneutral ausgeben“.

*seitenauswahl*

siehe „[Physische Auswahl \(Seitenauswahl\)](#)“ auf Seite 240

Auch außerhalb der Seitenauswahl liegende Überlauf-Seiten werden mit den primären CALC-Seiten ausgegeben.

ALL alle mit *seitenauswahl* ausgewählten CALC-Seiten des Realm ausdrucken

ALL RECORDS

die mit *seitenauswahl* ausgewählten CALC-Seiten aller, mit LOCATION CALC definierter Satzarten ausdrucken

ALL CALC SEARCHKEYS

die mit *seitenauswahl* ausgewählten CALC-Seiten aller CALC-SEARCH-Keys ausdrucken

ONLY RECORD *satzname*

aus den mit *seitenauswahl* ausgewählten CALC-Seiten alle Sätze oder bestimmte (*rsq-auswahl*) Sätze ausdrucken

*satzname*

Name einer mit LOCATION CALC definierten Satzart

ONLY CALC SEARCH-KEY *keyref*

aus den mit *seitenauswahl* ausgewählten CALC-Seiten des CALC-SEARCH-Keys *keyref* alle CALC-Index-Einträge oder bestimmte (*rsq-auswahl*) CALC-Index-Einträge ausdrucken

*keyref*

Schlüsselnummer eines CALC-SEARCH-Keys  
(siehe „CALC-SEARCH-KEY INFORMATION“ auf Seite 162)

*rsq-auswahl*

siehe „Logische Auswahl (RSQ-Auswahl)“ auf Seite 240

Beispiel 1

DISPLAY CALC PAGES PAGE 16 ALL CALC SEARCHKEYS

```
***** CALC KEY BUCKETS OF REALM < :SQL2:$XXXXXXXXX.VERSAND.ARTIKELRLM > *****
===== CALC SEARCH KEY, KEY REF 14, SET NAME < SYS_ARTIKEL > =====
----- REALM REF 11 BNR 16 ----- ACT KEY X'0B00010' -----
```

PAGE INFO:		TYPE 0	FREE SPACE	SIZE	10,	DISPL	30	NR OF PAGE INDICES	0	DISPL TO CALC TABLE HEADER	30
CALC KEY TABLE:		MAX ENTRIES	220	ACT ENTRIES	25	OVERFLOW BUCKET	NEXT	0,	PRIOR	0	
-1-	( 1)	(00000000)	F2F3F0F1	F0F7F3F6						23010736	
		RSQ	1	PPP 5,	13						
-2-	( 1)	(00000000)	F2F3F0F1	F0F7F4F2						23010742	
		RSQ	4	PPP 5,	13						
-3-	( 1)	(00000000)	F2F3F0F1	F0F7F4F8						23010748	
		RSQ	7	PPP 5,	13						
-4-	( 1)	(00000000)	F2F3F2F1	F0F7F3F8						23210738	
		RSQ	8	PPP 5,	14						
-5-	( 1)	(00000000)	F2F3F2F1	F0F7F4F0						23210740	
		RSQ	9	PPP 5,	14						
-6-	( 1)	(00000000)	F2F3F2F1	F0F7F4F6						23210746	
		RSQ	12	PPP 5,	14						
-7-	( 1)	(00000000)	F2F3F3F1	F0F8F3F8						23310838	
		RSQ	21	PPP 5,	15						

Beispiel 2

DISPLAY CALC PAGES ALL PAGES ONLY RECORD ARTIKELBESCHR ALL RSQS  
(mit LOCATION CALC definierte Satzart)

```
***** SCHEMA NAME < ARTIKELVERSAND > *****
***** CALC KEY BUCKETS OF REALM < :SQL2:$XXXXXXX.VERSAND.KLEIDUNG *****
===== CALC RECORD, REC REF 8, RECORD NAME < ARTIKELBESCHR > =====
----- REALM REF 5 BNR 2 ----- ACT KEY X'05000002'
```

PAGE INFO: TYPE 0 — FREE SPACE SIZE 20, DISPL 40 — NR OF PAGE INDICES 0 — DISPL TO CALC TABLE HEADER 40

CALC KEY TABLE: MAX ENTRIES 79 — ACT ENTRIES 1 — OVERFLOW BUCKET NEXT 0, PRIOR 0

-1-

```
( 1) (00000000) D2D3C5C9 C440C1E4 E240D1C5 D9E2C5E8 60D2D9C5 D7D74040 40404040 40404040 KLEID AUS JERSEY-KREPP
( 33) (00000020) 40404040 40404040
      RSQ      7 — PPP 5, 15
```

----- REALM REF 5 BNR 3 ----- ACT KEY X'05000003'

PAGE INFO: TYPE 0 — FREE SPACE SIZE 20, DISPL 40 — NR OF PAGE INDICES 0 — DISPL TO CALC TABLE HEADER 40

CALC KEY TABLE: MAX ENTRIES 79 — ACT ENTRIES 4 — OVERFLOW BUCKET NEXT 0, PRIOR 0

-1-

```
( 1) (00000000) D2D3C5C9 C440C1E4 E240C6D3 C9C5E2E2 C5D5C4C5 D440D1C5 D9E2C5E8 40404040 KLEID AUS FLIESENDEM JERSEY
( 33) (00000020) 40404040 40404040
      RSQ      8 — PPP 5, 16
```

-2-

```
( 1) (00000000) D2D3C5C9 C440C1E4 E240D1C5 D9E2C5E8 60D2D9C5 D7D740D4 C9E340D1 C1C3D2C5 KLEID AUS JERSEY-KREPP MIT JACKE
( 33) (00000020) 40404040 40404040
      RSQ      9 — PPP 5, 17
```

.  
.
  
.

----- REALM REF 5 BNR 4 ----- ACT KEY X'05000004'

PAGE INFO: TYPE 0 — FREE SPACE SIZE 20, DISPL 40 — NR OF PAGE INDICES 0 — DISPL TO CALC TABLE HEADER 40

CALC KEY TABLE: MAX ENTRIES 79 — ACT ENTRIES 3 — OVERFLOW BUCKET NEXT 0, PRIOR 0

-1-

```
( 1) (00000000) D7D3C9E2 E2C5C560 D2D3C5C9 C440D4C9 E340D1C1 C3D2C540 40404040 40404040 PLISSEE-KLEID MIT JACKE
( 33) (00000020) 40404040 40404040
      RSQ      6 — PPP 5, 14
```

-2-

```
( 1) (00000000) D7D6D3D6 60D2D3C5 C9C44040 40404040 40404040 40404040 40404040 POLO-KLEID
( 33) (00000020) 40404040 40404040
      RSQ     11 — PPP 5, 19
```

-3-

```
( 1) (00000000) E9E6C5C9 E3C5C9D3 C9C7C5E2 40E2D6D4 D4C5D9D2 D3C5C9C4 40404040 40404040 ZWEITEILIGES SOMMERKLEID
( 33) (00000020) 40404040 40404040
      RSQ      2 — PPP 5, 13
```

## CALC RECORD

Hashbereich einer mit LOCATION CALC definierten Satzart

REC REF

Nummer der Satzart

RECORD NAME

Name der Satzart

## CALC SEARCH KEY

Hashbereich eines mit USING CALC definierten SEARCH-Keys

KEY REF

Nummer des Schlüssels

SET NAME

Name des Sets, zu dem dieser Schlüssel gehört

## PAGE INFO

Seitenkopf (siehe PRINT-Anweisung);

er enthält bei CALC-Seiten an Stelle von DISPL TO END OF PAGE:

DISPL TO CALC TABLE HEADER

Distanz zum Kopf der CALC-Key-Tabelle

## LOGICAL RECORDS

Ausdruck der Datensätze, wahlweise jeweils mit:

PAGE INDEX

Seitenindex-Eintrag (siehe PRINT-Anweisung)

SCD

Set-Connection-Data (siehe PRINT-Anweisung)



## CALC KEY TABLE

CALC-Key-Tabelle;

vom Kopf der CALC-Key-Tabelle druckt BPRECORD Folgendes aus:

## MAX ENTRIES

Anzahl der maximal möglichen Einträge

## ACT ENTRIES

Anzahl der derzeitigen Einträge

## OVERFLOW BUCKET

Verkettung mit den Überlaufseiten:

## NEXT n

Seitennummer der nächsten Überlaufseite

0: keine Überlaufseite vorhanden

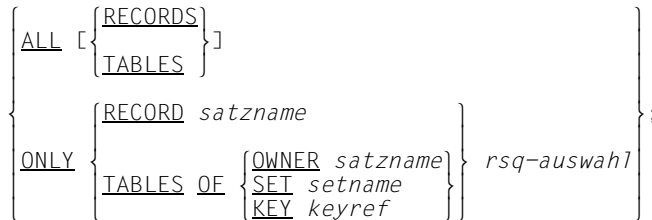
## PRIOR m

Seitennummer der vorhergehenden Seite

0: Primärseite

## Datenseiten ausdrucken (DISPLAY DATA)

`DISPLAY [IN CSV [csv-dateiname]] DATA PAGES seitenauswahl`



### IN CSV

BPRECORD gibt die Daten zusätzlich im CSV-Format aus.

*csv-dateiname*

Name der Datei, in die die Daten im CSV-Format ausgegeben werden sollen. Die Angabe von *csv-dateiname* ist bei der ersten IN CSV-Anweisung einer BPRECORD-Ausführung zwingend (z.B. DISPLAY IN CSV 'BPRECORD.CSV' ...).



Bei Vorhandensein von nationalen Feldern enthält die Ausgabe Daten im UTF-16-Format.

Eine ausführliche Beschreibung der Ausgabe im CSV-Format finden Sie im Handbuch „Datenbankbetrieb“ im Abschnitt „Datenbankinformationen systemneutral ausgeben“.

*seitenauswahl*

siehe „[Physische Auswahl \(Seitenauswahl\)](#)“ auf Seite 240

Auch außerhalb der Seitenauswahl liegende Überlauf-Seiten werden mit den primären Tabellenseiten ausgegeben.

ALL alle mit *seitenauswahl* ausgewählten Datenseiten des Realm ausdrucken

ALL RECORDS

aus den mit *seitenauswahl* ausgewählten Datenseiten alle Satzarten ausdrucken

ALL TABLES

aus den mit *seitenauswahl* ausgewählten Datenseiten alle Tabellen ausdrucken

ONLY RECORD *satzname*

aus den mit *seitenauswahl* ausgewählten Datenseiten alle oder bestimmte (*rsq-auswahl*) Sätze der Satzart *satzname* ausdrucken

*satzname*

Name einer Satzart, die nicht mit LOCATION MODE CALC definiert wurde

ONLY TABLES OF

aus den mit *seitenauswahl* ausgewählten Datenseiten die Tabellen oder bestimmte (*rsq-auswahl*) Tabelleneinträge ausdrucken:

OWNER *satzname*  
der Ownersatzart *satzname*

SET *setname*  
des Sets *setname*

KEY *keyref*  
des Schlüssels mit der Nummer *keyref* (siehe Seite 159).

*rsq-auswahl*  
siehe „Logische Auswahl (RSQ-Auswahl)“ auf Seite 240. Bei SYSTEM-Sets ist eine RSQ-Auswahl nicht sinnvoll und wird gegebenenfalls ignoriert.

Beispiel 1

DISPLAY DATA PAGES PAGE 19 ALL RECORDS  
(nur Datensätze)

```
***** DATA / TABLE PAGES OF REALM < :SQL2:$XXXXXXX.VERSAND.KLEIDUNG > *****
----- REALM REF 5 BNR 19 ACT KEY X'05000013' -----
```

PAGE INFO: TYPE 0 — FREE SPACE SIZE 2081, DISPL 2209 — NR OF PAGE INDICES 9 — DISPL TO END OF PAGE 4000

LOGICAL RECORDS:

```
-2- PAGE INDEX: DB_KEY 9, 48 — COL-NR 0 — LIST REC — DISPL 3767
( 1) (00000000) 00090000 00000030 05000013 00090000 00000030 05000013 00090000 00000030
( 33) (00000020) 05000013 00090000 00000030 05000013 00090000 00000030 05000013 00090000
( 65) (00000040) 00000030 05000013 00090000 00000030 05000013 00000000 000A0500 0012FF00
( 97) (00000060) 00000000 00000000 00000000 00000000 00000000 00000000 00000001 04000007
( 129) (00000080) F8F3F5F9 F2F8F0F9 E360E2C8 C9D9E360 D2D3C5C9 C4404040 40404040 40404040 83592809T-SHIRT-KLEID
( 161) (000000A0) 40404040 40404040 40404040 40404040 F2F3F7F1 F0F2F3F6 0020700C 0023900C 23710236
( 193) (000000C0) 00000500 000C050C 00000499 950C0000 00000000 050C00
```

```
-3- PAGE INDEX: DB_KEY 9, 49 — COL-NR 0 — LIST REC — DISPL 3552
( 1) (00000000) 00090000 00000031 05000013 00090000 00000031 05000013 00090000 00000031
( 33) (00000020) 05000013 00090000 00000031 05000013 00090000 00000031 05000013 00090000
( 65) (00000040) 00000031 05000013 00090000 00000031 05000013 00000000 000A0500 0012FF00
( 97) (00000060) 00000000 00000000 00000000 00000000 00000000 00000000 00000001 04000007
( 129) (00000080) F8F3F5F9 F2F8F0F9 E360E2C8 C9D9E360 D2D3C5C9 C4404040 40404040 40404040 83592809T-SHIRT-KLEID
( 161) (000000A0) 40404040 40404040 40404040 40404040 F2F3F7F1 F0F2F3F8 0020700C 0023900C 23710238
( 193) (000000C0) 00000500 000C050C 00000499 950C0000 00000000 050C00
```

```
-4- PAGE INDEX: DB_KEY 9, 50 — COL-NR 0 — LIST REC — DISPL 3337
( 1) (00000000) 00090000 00000032 05000013 00090000 00000032 05000013 00090000 00000032
( 33) (00000020) 05000013 00090000 00000032 05000013 00090000 00000032 05000013 00090000
( 65) (00000040) 00000032 05000013 00090000 00000032 05000013 00000000 000A0500 0012FF00
( 97) (00000060) 00000000 00000000 00000000 00000000 00000000 00000000 00000001 04000007
( 129) (00000080) F8F3F5F9 F2F8F0F9 E360E2C8 C9D9E360 D2D3C5C9 C4404040 40404040 40404040 83592809T-SHIRT-KLEID
( 161) (000000A0) 40404040 40404040 40404040 40404040 F2F3F7F1 F0F2F4F0 0020700C 0023900C 23710240
( 193) (000000C0) 00000500 000C050C 00000499 950C0000 00000000 050C00
```

Beschreibung von PAGE INFO, PAGE INDEX und SCD, siehe Seite 245.

*Beispiel 2*

DISPLAY DATA PAGES ALL PAGES ALL TABLES  
(nur Tabellen)

\*\*\*\*\* DATA / TABLE PAGES OF REALM < :SQL2:\$XXXXXXXXX.VERSAND.ARTIKELRLM \*\*\*\*\*  
REALM REF 11 BNR 31 ACT KEY X'0B00001F'

PAGE INFO: TYPE 0 — FREE SPACE SIZE 0, DISPL 32 — NR OF PAGE INDICES 1 — DISPL TO END OF PAGE 4000

LOGICAL RECORDS:

-1-	PAGE INDEX:	DB_KEY	0,	5	— COL-NR	1	— TABLE REC	— DISPL	32			
	TABLE:	MAX ENTRIES	112	—	TABLE NEXT	0	—	TABLE DESCR X'40'	—	NEXT HIGHER LEVEL	0	
		ACT ENTRIES	5	—	PRIOR	0	—	LEVEL NR	0	—	LAST ENTRY	31
-1-		RSQ	2	—	PPP	8,	4					
(	1)	(00000000)	C2C1E8C5	D9C9E2C3	C8C5E240	C2C9C5D9	40404040	40404040	40		BAYERISCHES BIER	
-2-		RSQ	1	—	PPP	5,	12					
(	1)	(00000000)	C5D3C5C7	C1D5E3C5	40D2D3C5	C9C4E4D5	C7404040	40404040	40		ELEGANTE KLEIDUNG	
-3-		RSQ	5	—	PPP	8,	6					
(	1)	(00000000)	D1D6C7C8	E4D9E340	40404040	40404040	40404040	40404040	40		JOGHURT	
-4-		RSQ	4	—	PPP	5,	18					
(	1)	(00000000)	D4D6C4C9	E2C3C8C5	40D2D3C5	C9C4E4D5	C7404040	40404040	40		MODISCHE KLEIDUNG	
-5-		RSQ	3	—	PPP	8,	5					
(	1)	(00000000)	E9C9E3D9	D6D5C5D5	40D3C9D4	D6D5C1C4	C5404040	40404040	40		ZITRONEN LIMONADE	

[ **Tabellenkopf**

[ zugehörige Tabelleneinträge

**TABLE**

Tabelle; vom Kopf der Tabelle druckt BPRECORD Folgendes aus:

**MAX ENTRIES**

Anzahl der maximal möglichen Einträge

**ACT ENTRIES**

Anzahl der derzeitigen Einträge

**TABLE**

Verkettung der Tabellenseiten

**NEXT n**

Seitennummer der nächsten Tabellenseite

**PRIOR m**

Seitennummer der vorhergehenden Tabellenseite;

0: keine vorhergehende Tabellenseite bzw. nachfolgende Tabellenseite vorhanden

## TABLE DESCR

Beschreibung der Tabelle

Bit  $2^7$  = 1: Liste

Bit  $2^6$  = 1: mehrstufige Tabelle

Bit  $2^5$  = 1: Tabelle ATTACHED TO OWNER

Bit  $2^4$  = 1: Duplikattabelle

Bit  $2^3$  = 1: Tabelle im ACTKEY-Format

Bit  $2^2$  = 1: Tabelle im ACTKEY-Format mit Verkettung zur letzten Seite

## LEVEL NR

Stufe der Tabelle

## NEXT HIGHER LEVEL

Seitennummer der Seite der nächsthöheren Stufe

## LAST ENTRY

Seitennummer der letzten Seite der Grundstufe



Bei Tabellenköpfen im ACTKEY-Format werden die Verkettungen NEXT, PRIOR, NEXT HIGHER LEVEL und LAST ENTRY im achtstelligen Hexadezimalformat ausgegeben. Nicht vorhandene Verkettungen (z.B. LAST ENTRY in Seiten der Grundstufe und PRIOR in Seiten der höchsten Stufe) bleiben in der Ausgabe frei. Bei Tabellenköpfen im BNR-Format geben die Verkettungen dezimal die jeweilige Blocknummer im gleichen Realm an. (vgl. Aufbau der Tabellen im Handbuch [Entwerfen und Definieren](#)).

## BPRECORD-Lauf beenden (END)

---

END

---

## 7.4 Kommandofolge zum Starten von BPRECORD

Die hier beschriebene Kommandofolge geht davon aus, dass UDS/SQL mit IMON installiert wurde (siehe Abschnitt „START-Kommandos der UDS/SQL-Programme“ in Kapitel 2 des Handbuchs „[Aufbauen und Umstrukturieren](#)“).

```
01 /ADD-FILE-LINK LINK-NAME=DATABASE,  
    FILE-NAME=[ :catid: ][ $userid. ] dbname.DBDIR[ .copyname ]  
02 /SELECT-PRODUCT-VERSION PRODUCT-NAME=UDS-SQL, VERSION=version. SCOPE=*TASK  
03 /START-UDS-BPRECORD  
04 bprecord-anweisungen  
05 END
```

- 01 Hier ist die Angabe von *:catid:* zulässig (siehe Handbuch „[Datenbankbetrieb](#)“).
- 02 Die Version des Dienstprogramms wird ausgewählt.  
Die Angabe der Version wird generell empfohlen, da mehrere UDS/SQL-Versionen parallel installiert sein können.
- 03 Das UDS/SQL-Dienstprogramm kann auch mit dem Aliasnamen BPRECORD gestartet werden.

---

## 8 Online-Dienste durchführen mit der UDS-Online-Utility

Die UDS-Online-Utility bietet die Möglichkeit, einige Funktionen zur Verdichtung von Datenbeständen, die sonst teilweise mit vergleichbaren Funktionen durch Dienstprogramme realisiert sind, online (d.h. während des DBH-Betriebs) durchzuführen.

Offline können Sie die Verdichtung von Datenbeständen folgendermaßen erreichen:

- mit BPGSIZE (siehe Handbuch „[Aufbauen und Umstrukturieren](#)“) oder
- durch Entladen und Neuladen der Daten mit BOUTLOAD/BINILOAD (siehe Handbuch „[Aufbauen und Umstrukturieren](#)“) oder
- durch Neuaufbau von Tabellen mit BREORG (siehe [Kapitel „Datenbank reorganisieren mit BREORG“ auf Seite 313](#))

Mit der UDS-Online-Utility können Sie die Verlagerung und Verdichtung von Datensätzen und Tabellen gezielt in einzelnen Realms im laufenden Betrieb durchführen. Außerdem können ganze Datenseiten, die Sätze einer verteilbaren Liste enthalten, im laufenden Betrieb von einem Realm in einen anderen verlagert werden.

Bei der Verdichtung von Datenbeständen mit BPGSIZE bzw. BOUTLOAD/BINILOAD werden Sperren von DBTT-Einträgen (vgl. Anweisung KEEP des Dienstprogrammes BMODTT) aufgehoben; bei der Verdichtung mit der UDS-Online-Utility bleiben diese Sperren bestehen.

Mit der UDS-Online-Utility können Sie die Freiplatzsuche im Realm verändern und damit die vergleichbaren Funktionen SET und RESET des Dienstprogrammes BMODTT (siehe [Seite 355](#)) auch im laufenden Betrieb aufrufen und wirksam machen.

Für verteilbare Listen können Sie mit der UDS-Online-Utility im laufenden Betrieb den Preferred-Realm ändern, d.h. den Realm, in dem der DBH aktuell freie Seiten zur Einspeicherung von Sätzen sucht.

Die UDS-Online-Utility bietet die Möglichkeit die Probable Position Pointers (PPP) während des laufenden Betriebs auf die neuesten Werte zu aktualisieren.

## 8.1 Funktionen der UDS-Online-Utility

Mit der UDS-Online-Utility haben Sie die Möglichkeit, Änderungen an physischen Eigenschaften der in den Datenbanken enthaltenen Daten im normalen DBH-Betrieb durchzuführen. Die Inhalte der Benutzerdaten sind dabei in der UDS-Online-Utility weder sichtbar noch änderbar.

Alle angeforderten Änderungen werden durch den DBH in Transaktionen durchgeführt, die vergleichbar zu normalen Benutzertransaktionen sind und deren Steuerung allein der UDS-Online-Utility obliegt. Die UDS-Online-Utility verhält sich insofern wie ein ganz normales Anwenderprogramm.

Die UDS-Online-Utility bietet folgende Funktionen in Form von DMLs an, die die Datenbank selbst oder deren Bearbeitung verändern:

- Mit der Online-DML RELOCATE verlagern Sie Sätze und Tabellen in einem einzelnen Realm. Dabei wird im Allgemeinen die Belegungsdichte im vorderen Teil des Realms erhöht. Im hinteren Teil des Realms entstehen dafür zusammenhängende freie Daten-seiten. Des Weiteren können Sie einen Realm entlasten durch Verlagern von Seiten einer verteilbaren Liste in einen anderen Realm. Eine Beschreibung der DML RELOCATE finden Sie im [Abschnitt „DML RELOCATE - Datenseiten verlagern“ auf Seite 266](#).
- Mit der Online-DML FPASCAN verändern Sie die Einstellungen zur Freiplatzsuche im Realm im laufenden Betrieb. Eine Beschreibung der DML FPASCAN finden Sie im [Abschnitt „DML FPASCAN - Such-Modus für Freiplatzsuche festlegen“ auf Seite 269](#).
- Mit der Online-DML PREFRLM verändern Sie die Einstellung des Preferred-Realm für eine verteilbare Liste. Eine Beschreibung der DML PREFRLM finden Sie im [Abschnitt „DML PREFRLM - Preferred-Realm für verteilbare Listen festlegen“ auf Seite 269](#).
- Mit der Online-DML REORGPPP können Sie PPPs in einem einzigen Realm während des laufenden Betriebs auf die neuesten Werte aktualisieren. Eine Beschreibung der DML REORGPPP finden Sie im [Abschnitt „DML REORGPPP – Probable Position Pointers \(PPPs\) reorganisieren“ auf Seite 270](#).

Die Funktionen RELOCATE, FPASCAN und PREFRLM beziehen sich in einem Prozedurlauf der UDS-Online-Utility immer auf eine Datenbank, die durch das Subschema eindeutig identifiziert wird, und auf einen Quellrealm. Sollen in einem Prozedurlauf der UDS-Online-Utility mehrere Funktionen (RELOCATE, FPASCAN und PREFRLM) ausgeführt werden, dann muss bei allen Funktionen der Subschema-Name und der Realm-Name jeweils identisch sein.

Um die angeforderten Änderungen in den Transaktionen der UDS-Online-Utility durchzuführen, müssen im DBH Ressourcen konkurrierend zu den weiteren, parallel ablaufenden Anwendungsprogrammen genutzt werden (Sperrern, I/O-Puffer,...). Um negative Auswir-



kungen auf diese Anwendungen zu vermeiden, sollten Sie die UDS-Online-Utility in Zeiten einsetzen, in denen die Normallast und Änderungen in dem von der UDS-Online-Utility bearbeitenden Realm eher gering sind.

Mit den Anweisungen der UDS-Online-Utility können Sie den Umfang der in einer Transaktion durchzuführenden Änderungen steuern, sodass Sie die Behinderung von parallel ablaufenden Anwendungsprogrammen sehr gering halten können. Auch die Abfolge der einzelnen Transaktionen der UDS-Online-Utility können Sie durch gezielte Wartezustände so steuern, dass Sie Behinderungen für parallel ablaufende Anwendungsprogramme weiter vermindern können.

Insgesamt ist es kein Ziel, die mit der UDS-Online-Utility angeforderten Änderungen in möglichst kurzer Zeit durchzuführen. Vielmehr ist es das Ziel, diese in möglichst kleinen Einheiten und mit möglichst geringer Auswirkung auf parallele Anwendungen abzuwickeln.

Die einzelnen Aufträge der UDS-Online-Utility an den DBH liefern Informationen über die durchgeführten Änderungen. Diese Informationen können Sie wiederum für die weitere Steuerung der UDS-Online-Utility nutzen.

Zu einem Zeitpunkt können mehrere Instanzen der UDS-Online-Utility aktiv sein, die Änderungen in unterschiedlichen Realms durchführen.

Bitte beachten Sie, dass durch die Aktionen der UDS-Online-Utility der Umfang der ALOG-Daten evtl. deutlich größer wird und berücksichtigen Sie dies, wenn Sie den Einsatz der UDS-Online-Utility planen.

## 8.2 DML RELOCATE - Datenseiten verlagern

Mit der Online-DML RELOCATE verlagern Sie im laufenden Betrieb Datensätze und kleine Tabellen gezielt innerhalb eines Realms. Es wird, soweit möglich, in freie Bereiche am Anfang des Realms verlagert. Dabei wird auch freier Platz in nicht vollständig belegten Seiten ausgenutzt. Durch die Verlagerung entstehen im hinteren Bereich des Realms freie Seiten. Eine RELOCATE-DML bezieht sich immer auf einen Benutzerrealm, der, wie auch die entsprechenden DBTT-Realms, im DBH im Update-Modus zugeschaltet sein muss.

Je nach Ausprägung des RELOCATE werden Sätze und Tabellen verlagert, die in nicht vollständig belegten Seiten liegen oder ganze Datenseiten, die Teile mehrseitiger Tabellen enthalten.

Folgendes wird nicht verlagert:

- vollständig belegte Seiten, wenn sie FPA-Einträge oder DBTT-Einträge enthalten
- DBTT-Ankerseiten
- CALC-Seiten
- Ankersätze von SYSTEM-Sets (Ankersätze enthalten zu DBTT-Seiten vergleichbare Informationen).

### Ablauf der Verlagerung

Die Verlagerung wird folgendermaßen abgearbeitet:

Vom Ende des Realms her wird nach einer geeigneten Datenseite gesucht. Für die Sätze und Tabellen einer solchen Seite bzw. für die ganze Seite wird vom Beginn des Realms freier Platz gesucht, in den die Sätze und Tabellen einzeln verlagert werden.

Zusammengehörige Daten, die durch entsprechende SSL-Deklarationen gekennzeichnet sind, werden nach Möglichkeit auch wieder in ein und derselben Zielseite abgespeichert (Erhaltung von Clustern).

Falls Seiten einer verteilbaren Liste in einen anderen Realm verlagert werden sollen, werden die benötigten freien Seiten im Zielrealm (TARGET-REALM) gesucht.

Die Anzahl der Seiten, die mit einer RELOCATE-DML nach Möglichkeit durch Verlagerung geleert werden sollen, geben Sie über die Anweisungen der UDS-Online-Utility vor (Parameter PAGES-PER-DML der SDF-Anweisung SET-RELOCATE-PARAMETERS, siehe [Abschnitt „Eigenschaften einer RELOCATE-DML festlegen \(SET-RELOCATE-PARAMETERS\)“ auf Seite 283](#)).

Falls es bei der Verlagerung zu Kollisionen mit anderen Transaktionen kommt, müssen Sie damit rechnen, dass die RELOCATE-DML die Verlagerung vorzeitig abbricht, um die Behinderung für parallele Anwendungen möglichst gering zu halten. In der Transaktion bereits durchgeführte Verlagerungen bleiben dabei im Allgemeinen in dem neuen Zustand erhalten.

## Fortschrittsdaten der Verlagerung

Der DBH führt in einem Sessionabschnitt interne Fortschrittsdaten über die bereits durch Verlagerung entleerten Seiten am Ende des Quellrealms (Quellpegel) sowie über die bereits aufgefüllten Seiten am Beginn des Zielrealms (Zielpegel), sodass in mehreren aufeinanderfolgenden Online-Utility-Läufen aufwandsarm die Verlagerung fortgesetzt werden kann.

Quell- und Zielpegel werden zur Information in der UDS-Online-Utility als Variablen bereitgestellt (Variable RELOC\_ORIGIN für Quellpegel und Variable RELOC\_DESTINATION für Zielpegel) und enthalten die Blocknummern der jeweiligen Datenbankseiten, bei denen bei weiteren Verlagerungen wieder aufgesetzt wird.

## Initialisierung von Quell- und Zielpegel

Im Allgemeinen führt die erste Verlagerungstransaktion einer Folge von Verlagerungstransaktionen zur Initialisierung des Quellpegels auf die letzte nicht leere Seite am Ende des Realms und des Zielpegels auf die erste Seite am Anfang des Realms - also zunächst auf den ACTKEY0. Nach der Verarbeitung einer erfolgreichen RELOCATE-DML wird auf die Seiten positioniert, bei denen eine nachfolgende RELOCATE-DML ihre Arbeit fortsetzt. Diese Werte werden jeweils als Quell- und Zielpegel in der UDS-Online-Utility bereitgestellt.

Die Fortschrittsdaten einer Verlagerung werden in der Datenbank nicht für spätere Sessionabschnitte festgehalten. Sie gehen auch mit dem Abschalten des entsprechenden Realms verloren. Nach einem Zuschalten des Realms sind somit keine Fortschrittsdaten aus vorherigen Sessionabschnitten mehr verfügbar. Die nächste Verlagerungstransaktion kann nur mit erneuter Initialisierung von Quell- und Zielpegel erfolgen.

Es ist möglich, nach Neustart der UDS-Online-Utility gezielt eine Verlagerung in einem nicht unterbrochenen Sessionabschnitt fortzusetzen. Mit dem Parameter INITIALIZE der SDF-Anweisung SET-RELOCATE-PARAMETERS steuern Sie dabei die Initialisierung von Quell- und Zielpegel:

- INITIALIZE=\* ANY  
Da die UDS-Online-Utility evtl. nichts von dem Verlust der Fortschrittsdaten mitbekommen hat, bietet INITIALIZE=\* ANY die Möglichkeit, die Verlagerungstransaktion so auszuführen, dass evtl. noch vorhandene Fortschrittsdaten weiter genutzt werden, ansonsten aber eine Initialisierung stattfindet.
- INITIALIZE=\*NO  
INITIALIZE=\*NO bietet die Möglichkeit, die Neuinitialisierung explizit zu verhindern, wenn der Realm nach der letzten Verlagerungstransaktion abgeschaltet und dann wieder zugeschaltet worden ist. Es ist ja nicht auszuschließen, dass zwischen dem Abschalten und erneuten Zuschalten des Realms an diesem Änderungen vorgenommen worden sind, die eine Fortsetzung der Verlagerung nicht mehr sinnvoll macht.

– INITIALIZE=\*YES

Es ist aber auch möglich, mit INITIALIZE=\*YES eine erneute Initialisierung der Verlagerung anzustoßen. Dabei werden die Pegel initialisiert wie oben beschrieben. Diese Initialisierung wird in der ersten entsprechenden Transaktion wirksam. Alle folgenden Transaktionen einer UDS-Online-Utility werden mit INITIALIZE=\*ANY ausgeführt. Dabei ist es unerheblich, ob diese Transaktionen in einer oder mehreren aufeinanderfolgenden Prozedurfolgen (SDF-Anweisung REPEAT-PROCEDURE) ausgeführt werden.

Die Initialisierung von Quell- und Zielpegel führt implizit auch zu einer Durchführung der DML FPASCAN mit \*REUSE. Der explizite Aufruf von FPASCAN schadet aber nicht. Mit der DML FPASCAN wird die Seite, ab der für neu einzuspeichernde Sätze und Tabellen Platz gesucht wird, auf den Beginn des Realms bzw. auf den Zielpegel gelegt, damit parallele Neueinspeicherungen, die durch die DMLs MODIFY und STORE von Anwendungsprogrammen verursacht sind, nicht hinter dem Quellpegel der Verlagerung eingespeichert werden. Eine Beschreibung von FPASCAN finden Sie im [Abschnitt „DML FPASCAN - Such-Modus für Freiplatzsuche festlegen“ auf Seite 269](#)

### Ende der Verlagerung

Eine Folge von Verlagerungstransaktionen, bei denen Quellrealm und Zielrealm gleich sind, wird intern beendet, wenn der Quellpegel kleiner ist als der Zielpegel. Dies wird der UDS-Online-Utility bekannt gemacht und führt dazu, dass keine weiteren Verlagerungstransaktionen auf diesem Realm mehr durchgeführt werden, da diese nichts mehr zu tun haben und immer die gleiche Information über Quell- und Zielpegel zurückmelden.

### Füllgrad, bis zu dem Seiten verlagert werden sollen

Im Allgemeinen kann bei der Verlagerung erst nach dem Lesen der Datenseite entschieden werden, ob die in der Seite enthaltenen Daten verlagert werden können. Mit dem Parameter SKIP-ABOVE-FILLING der SDF-Anweisung SET-RELOCATE-PARAMETERS können Sie Seiten von der Verlagerung ausschließen, die einen bestimmten, prozentualen Füllgrad übertreffen (siehe [Abschnitt „Eigenschaften einer RELOCATE-DML festlegen \(SET-RELOCATE-PARAMETERS\)“ auf Seite 283](#)). Der Standardwert 100 bezieht alle verlagerebaren Seiten ein. Mit einem kleineren Wert können Sie beispielsweise die Verlagerung auf dünn besetzte Seiten beschränken. Eine Angabe unter 100 benutzt der DBH auch, um anhand der Freiplatzverwaltungsdaten Seiten von der Verlagerung auszuschließen. Mit dieser Optimierung können Sie somit auch unnötiges Lesen von Datenseiten vermeiden.

## Verhalten bei gesperrten Quellseiten

Mit dem Parameter CLASH-HANDLING der SDF-Anweisung SET-RELOCATE-PARAMETERS steuern Sie, wie der DBH verfahren soll, wenn er auf eine gesperrte Quellseite trifft (siehe [Abschnitt „Eigenschaften einer RELOCATE-DML festlegen \(SET-RELOCATE-PARAMETERS\)“ auf Seite 283](#)). Bei der Standardeinstellung BREAK-DML wird die DML unter Beibehaltung der bereits erfolgten Verlagerungen abgebrochen. Die Steuerung in der UDS-Online-Utility kann auf diese Situation reagieren und den nächsten Verlagerungsauftrag nach einer gewissen Wartezeit wiederholen. Es gibt aber auch die Möglichkeit, mit SKIP-PAGE eine gesperrte Seite jeweils zu überspringen. Ein Warten wird bei der Abarbeitung der einzelnen RELOCATE-DML nicht angeboten, weil hierdurch die Gefahr von Deadlocks entstehen könnte, die parallele Anwendungen negativ beeinflussen könnten.

## 8.3 DML FPASCAN - Such-Modus für Freiplatzsuche festlegen

Mit der Online-DML FPASCAN und dem Parameter SEARCH-MODE der SDF-Anweisung SET-FPA-SCAN-PARAMETERS legen Sie den Such-Modus für die Freiplatzsuche fest, d.h. die Seite in einem Realm, ab der für neu einzuspeichernde Sätze und Tabellen, die durch die DMLs MODIFY und STORE von Anwendungsprogrammen verursacht sind, Platz gesucht wird:

- SEARCH-MODE=\*REUSE legt die Startseite für die Freiplatzsuche auf den Beginn des Realms („Second Scan“). Damit wird eine Möglichkeit geboten, freien Platz, der z.B. durch Löschen von Sätzen oder Tabellen frei geworden ist, bei einer Neueinspeicherung ab sofort wieder zu nutzen.
- Mit SEARCH-MODE=\*NOREUSE wechselt der DBH beim Erreichen des nächsten Konsistenzpunktes in den „First Scan“. Bei diesem Such-Modus wird Platz für neu einzuspeichernde Daten im freien Bereich am Ende des Realms gesucht, d.h. ab der ersten Seite des zusammenhängenden Bereichs freier Seiten am Ende des Realms.

## 8.4 DML PREFRLM - Preferred-Realm für verteilbare Listen festlegen

Mit der Online-DML PREFRLM legen Sie den Preferred-Realm für eine verteilbare Liste fest, d.h. den Realm, in dem der DBH aktuell freie Seiten zur Einspeicherung von Sätzen sucht. Die dazu erforderlichen Parameter SET-NAME (der eine verteilbare Liste bezeichnen muss) und PREFERRED-REALM-NAME vereinbaren Sie vorher mit der SDF-Anweisung SET-PREF-REALM-PARAMETERS.

## 8.5 DML REORGPPP – Probable Position Pointers (PPPs) reorganisieren

Die Online-DML REORGPPP wird verwendet, um PPPs in Sätzen und Tabellen speziell innerhalb eines Realms während eines Live-Vorgangs zu reorganisieren.

Eine REORGPPP-DML verweist immer auf einen Benutzerrealm, der wie auch die entsprechenden DBTT-Realms im Aktualisierungsmodus im DBH angefügt werden muss.

Die folgenden Seiten werden bei der PPP-Reorganisation übersprungen:

- FPA-Seiten und DBTT-Seiten
- DBTT-Ankerseiten
- Freie Seiten

### Prozedur der PPP-Reorganisation

Alle Seiten im angegebenen Benutzerrealm (außer übersprungene Seiten, siehe oben) werden sequenziell gelesen, und alle PPPs werden auf diesen Seiten aktualisiert.

Um PPPs zu aktualisieren, geben Sie die Anzahl der Seiten an, die soweit wie möglich mit einer REORGPPP-DML über die Anweisungen des UDS-Online-Hilfsprogramms verarbeitet werden sollen

(Parameter PAGES-PER-DML der SDF-Anweisung SET-REORGANIZE-PPP-PARAMETERS, siehe [Abschnitt „Eigenschaften einer REORGPPP-DML definieren \(SET-REORGANIZE-PPP-PARAMETERS“ auf Seite 288.](#)



Wenn während der PPP-Reorganisation Konflikte mit anderen Transaktionen auftreten, bricht die Online-DML REORGPPP den Vorgang vorzeitig ab, um die Behinderung einer parallelen Anwendung auf ein Minimum zu reduzieren. PPPs, die bei der Transaktion bereits reorganisiert wurden, werden aktualisiert beibehalten.

### Fortschrittsdaten der PPP-Reorganisation

In jedem Session-Abschnitt speichert der DBH interne Fortschrittsdaten für die Seitennummer im angegebenen Realm, in dem PPPs bereits reorganisiert wurden. Auf diese Weise kann die Reorganisation problemlos in mehreren gleichzeitigen Online-Hilfsprogramm-Ausführungen fortgesetzt werden.

Die aktuelle Seitennummer wird als die Variable REORG-PPP-CURRENT bereitgestellt, um das UDS-Online-Hilfsprogramm zu informieren. Sie enthält die Blocknummer der relevanten Datenbankseite, auf der gegebenenfalls die weitere PPP-Reorganisation fortgesetzt wird.

## Initialisierung des Prozesses der PPP-Reorganisation

Die erste REORGPPP-DML beginnt mit der Seitennummer 0, d. h. am Anfang des Realms. Nachdem eine erfolgreiche REORGPPP-DML verarbeitet wurde, wird die Position auf die Seitennummer gesetzt, auf der eine nachfolgende REORGPPP-DML den Vorgang fortsetzt. Dieser Wert wird als die aktuelle Seitennummer im UDS-Online-Hilfsprogramm bereitgestellt.

Die Fortschrittsdaten für die Reorganisation werden nicht für spätere Session-Abschnitte in der Datenbank gespeichert. Sie gehen verloren, wenn der betroffene Realm getrennt wird. Nachdem der Realm verbunden wurde, sind Fortschrittsdaten aus früheren Session-Abschnitten deshalb nicht mehr verfügbar. Die nächste Transaktion zur PPP-Reorganisation kann nur durchgeführt werden, wenn die aktuelle Seitennummer neu initialisiert wird.

Nach dem Neustart des UDS-Online-Hilfsprogramms kann die Reorganisation in einem Session-Abschnitt fortgesetzt werden, der nicht unterbrochen wurde. Verwenden Sie in diesem Fall den Parameter INITIALIZE der SDF-Anweisung SET-REORGANIZE-PPP-PARAMETERS, um die Initialisierung der aktuellen Seitennummer zu kontrollieren:

- INITIALIZE=\* ANY  
Da das UDS-Online-Hilfsprogramm möglicherweise nicht erkannt hat, dass die Fortschrittsdaten verloren gegangen sind, ermöglicht INITIALIZE=\* ANY die Ausführung der REORGPPP-Transaktion unter Verwendung beliebiger noch vorhandener Fortschrittsdaten. Wenn keine Fortschrittsdaten vorhanden sind, findet eine Initialisierung statt.
- INITIALIZE=\*NO  
Über INITIALIZE=\*NO kann eine Initialisierung explizit verhindert werden, wenn der Realm getrennt und dann nach der letzten REORGPPP-Transaktion wieder verbunden wurde.
- INITIALIZE=\*YES  
Es kann jedoch auch INITIALIZE=\*YES verwendet werden, um die Initialisierung des Reorganisationsprozesses wieder zu starten. In diesem Fall wird die anfängliche Seitennummer initialisiert. Diese Initialisierung wird bei der ersten entsprechenden Transaktion wirksam. Alle nachfolgenden Transaktionen eines UDS-Online-Hilfsprogramms werden mit INITIALIZE=\*ANY ausgeführt. Ob diese Transaktionen in einer oder in mehreren aufeinander folgenden Prozedursequenzen (SDF-Anweisung REPEAT-PROCEDURE) ausgeführt werden, ist nicht relevant.

## Ende der PPP-Reorganisation

Eine Sequenz von REORGPPP-Transaktionen wird intern beendet, wenn die aktuelle Seitennummer die letzte Seitennummer in dem Realm erreicht. Dies wird dem UDS-Online-Hilfsprogramm gemeldet. Dadurch werden keine weiteren Transaktionen zur Reorganisation in diesem Realm ausgeführt.

## Verhalten bei gesperrten Seiten

Über den Parameter CLASH-HANDLING der SDF-Anweisung SET-REORGANIZE-PPP-PARAMETERS können Sie das Verhalten des DBH bei einer gesperrten Seite kontrollieren (siehe [Abschnitt „Eigenschaften einer REORGPPP-DML definieren \(SET-REORGANIZE-PPP-PARAMETERS“ auf Seite 288“](#)).

- Mit dem Standardwert BREAK-DML wird die DML abgebrochen und die PPP-Reorganisationen, die bereits stattgefunden haben, werden beibehalten. Der Controller im UDS-Online-Hilfsprogramm kann auf diese Situation reagieren und den nächsten Auftrag zur Reorganisation nach einer bestimmten Wartezeit wiederholen.
- SKIP-PAGE bietet die Möglichkeit, eine gesperrte Seite zu überspringen.

Es besteht keine Möglichkeit zu warten, während die einzelne REORGPPP-DML verarbeitet wird, weil dabei die Gefahr von Deadlocks besteht, die sich negativ auf parallele Anwendungen auswirken können.



## 8.6 Kommandofolge zum Starten der UDS-Online-Utility

Die UDS-Online-Utility wird folgendermaßen gestartet:

```
/SELECT-PRODUCT-VERSION PRODUCT-NAME=UDS-SQL,VERSION=...  
/START-UDS-ONLINE-UTILITY ...
```

Die UDS-Online-Utility muss in der Kennung der zu bearbeitenden Datenbank ablaufen. Eine Verteilung der Aufträge über UDS-D ist nicht möglich.

Der Programmname, mit dem die UDS-Online-Utility in den Ausgaben von DAL bzw. im UDS-Monitor erscheint, ist \$UDSOUTI.

## 8.7 Anweisungen der UDS-Online-Utility

Die Eingabeschnittstelle der UDS-Online-Utility ist aufgeteilt in SDF-Anweisungen und Prozedur-Anweisungen.

Die SDF-Anweisungen werden einmalig und sofort ausgeführt. Sie steuern das Verhalten der UDS-Online-Utility selbst. Mit ihnen wird die Transaktionsverarbeitung vorbereitet.

Die Prozedur-Anweisungen werden unter der Steuerung des Anwenders zum Ablauf gebracht. Mit ihnen wird die Transaktionsverarbeitung im Detail festgelegt.

## 8.8 SDF-Anweisungen der UDS-Online-Utility

Folgende SDF-Anweisungen werden für die UDS-Online-Utility bereitgestellt:

Anweisung	Bedeutung
DECLARE-PROCEDURE	Prozedur-Deklaration eröffnen
DECLARE-VARIABLE	Variable definieren
DELETE-PROCEDURE	Prozedur löschen
DELETE-VARIABLE	Variable löschen
END	UDS-Online-Utility beenden
REPEAT-PROCEDURE	Prozedur ausführen
SET-FPA-SCAN-PARAMETERS	Such-Modus für Freiplatzsuche festlegen
SET-ONLINE-UTILITY-PARAMETERS	Online-Utility-Parameter festlegen
SET-PREFERRED-REALM-PARAMETERS	Preferred-Realm für eine verteilbare Liste festlegen
SET-RELOCATE-PARAMETERS	Eigenschaften einer RELOCATE-DML festlegen
SET-REORGANIZE-PPP-PARAMETERS	Eigenschaften einer REORGPPP-DML festlegen
SHOW-FPA-SCAN-PARAMETERS	Aktuell gültige Parameter für DML FPASCAN ausgeben
SHOW-PREF-REALM-PARAMETERS	Aktuell gültige Parameter für DML PREFRLM ausgeben
SHOW-PROCEDURE	Prozedur ausgeben
SHOW-RELOCATE-PARAMETERS	Aktuell gültige Parameter für DML RELOCATE ausgeben
SHOW-REORGANIZE-PPP-PARAMETERS	Aktuell gültige Parameter für DML REORGPPP ausgeben
SHOW-VARIABLE	Aktuellen Wert einer Variablen ausgeben

Tabelle 18: SDF-Anweisungen für UDS-Online-Utility

Die Anweisungen der UDS-Online-Utility sind im Folgenden in alphabetischer Reihenfolge beschrieben.

## Prozedur definieren (DECLARE-PROCEDURE)

Die Anweisung DECLARE-PROCEDURE definiert eine Prozedur.

Die UDS-Online-Utility speichert Prozedur-Deklaration intern für eine spätere Verarbeitung ab. Fehlerhafte Prozedur-Anweisungen werden erst beim Ablauf der Prozedur erkannt.

```
DECLARE-PROCEDURE
```

```
PROCEDURE-NAME = <structured-name 1..20>
```

```
,CODE = <c-string 1..1800> / <filename> / *SYSDTA
```

### **PROCEDURE-NAME = <structured-name 1..20>**

Name der Prozedur, die gespeichert werden soll. Der Prozedurname darf bis zu 20 Zeichen lang sein und muss eindeutig sein. Die Namen von vordefinierten Standardprozeduren dürfen nicht verwendet werden.

### **CODE = <c-string 1..1800>**

Vollständiger Code der Prozedur.

### **CODE = <filename>**

Name der Datei, die die Prozedur-Anweisungen enthält. Diese Datei kann entweder eine SAM-Datei sein oder eine ISAM-Datei mit Standardparametern (RECFORM=V, KEYLEN=8, KEY- POS=5).

### **CODE = \*SYSDTA**

Die Prozedur-Anweisungen sollen von SYSDTA eingelesen werden.

## Variable definieren (DECLARE-VARIABLE)

Mit DECLARE-VARIABLE definieren Sie eine Variable, die in Prozedur-Anweisungen und Bedingungen verwendet werden kann.

Namen von vordefinierten Variablen (siehe [Abschnitt „Vordefinierte Variablen“ auf Seite 304](#)) dürfen in der Anweisung DECLARE-VARIABLE nicht verwendet werden. Einige der vordefinierten Variablen sind vom Typ C (z.B. RCODE).

```
DECLARE-VARIABLE
```

```
VARIABLE-NAME = <structured-name 1..20>
,TYPE = *STRING(...) / *INTEGER
    *STRING
    |   LENGTH = <integer 1..20>
,INITIAL-VALUE = <integer 1..16777215> / <c-string> / *STD / *NONE
```

### **VARIABLE-NAME = <structured-name 1..20>**

Name der Variable. Namen von vordefinierten Variablen (siehe [Abschnitt „Vordefinierte Variablen“ auf Seite 304](#)) dürfen nicht verwendet werden.

### **TYPE =**

Typ der Variablen.

### **TYPE = \*STRING(...)**

Bei der Variablen handelt es sich um eine Textvariable.

#### **LENGTH = <integer 1..20>**

Länge der Textvariablen. Der Standardwert ist 5.

### **TYPE = \*INTEGER**

Bei der Variablen handelt es sich um eine ganzzahlige Variable.

### **INITIAL-VALUE =**

Wert, mit dem die Variable vorbelegt werden soll.

### **INITIAL-VALUE = <integer 1..16777215>**

Die Variable soll mit einer Integer-Zahl initialisiert werden.

### **INITIAL-VALUE = <c-string>**

Die Variable soll mit einem String initialisiert werden.

### **INITIAL-VALUE = \*STD**

INITIAL-VALUE \*STD belegt eine Variable vom Typ \*STRING mit Blanks und eine Variable vom Typ INTEGER mit 0.

**INITIAL-VALUE = \*NONE**

Bei INITIAL-VALUE \*NONE ist der Initialisierungswert undefiniert.

**Prozedur löschen (DELETE-PROCEDURE)**

Die Anweisung DELETE-PROCEDURE löscht eine vorher deklarierte und abgespeicherte Prozedur. Eine vordefinierte Standardprozedur (siehe [Abschnitt „Vordefinierte Standardprozeduren“ auf Seite 305](#)) kann nicht gelöscht werden.

DELETE-PROCEDURE
PROCEDURE-NAME = <structured-name 1..20>

**PROCEDURE-NAME= <structured-name 1..20>**

Name der Prozedur, die gelöscht werden soll.

**Variable löschen (DELETE-VARIABLE)**

Die Anweisung DELETE-VARIABLE löscht eine vorher deklarierte Variable. Eine vordefinierte Variable (siehe [Abschnitt „Vordefinierte Variablen“ auf Seite 304](#)) kann nicht gelöscht werden.

DELETE-VARIABLE
VARIABLE-NAME = <structured-name 1..20>

**VARIABLE-NAME = <structured-name 1..20>**

Name der Variablen, die gelöscht werden soll.

**UDS-Online-Utility beenden (END)**

Die Anweisung END beendet die UDS-Online-Utility.

END

## Prozedur ausführen (REPEAT-PROCEDURE)

Die Anweisung REPEAT-PROCEDURE führt eine vorher deklarierte und abgespeicherte Prozedur bzw. eine vordefinierte Standardprozedur aus.

Entsprechend der in der Prozedur enthaltenen DML FPASCAN bzw. RELOCATE muss vorher die zugehörige Anweisung SET-FPA-SCAN-PARAMETERS bzw. SET-RELOCATE-PARAMETERS ausgeführt worden sein. Es werden die Werte der letzten zugehörigen Anweisung verwendet.

REPEAT-PROCEDURE
PROCEDURE-NAME = *STDRELOC / *STDFPASCAN / *STDREPPP / <structured-name 1..20> ,CYCLE-LIMIT = <integer 1..16777215> / *MAX

### PROCEDURE-NAME =

Name der Prozedur, die ausgeführt werden soll.

### PROCEDURE-NAME = \*STDRELOC

Die Standardprozedur \*STDRELOC (siehe [Abschnitt „Vordefinierte Standardprozeduren“ auf Seite 305](#)) soll ausgeführt werden.

### PROCEDURE-NAME = \*STDFPASCAN

Die Standardprozedur \*STDFPASCAN (siehe [Abschnitt „Vordefinierte Standardprozeduren“ auf Seite 305](#)) soll ausgeführt werden.

### PROCEDURE-NAME = \*STDREPPP

Die Standardprozedur \*STDREPPP (siehe [Abschnitt „Vordefinierte Standardprozeduren“ auf Seite 305](#)) soll ausgeführt werden.

### PROCEDURE-NAME = <structured-name 1..20>

Name der vorher deklarierten und abgespeicherten Prozedur, die ausgeführt werden soll.

### CYCLE-LIMIT =

Gibt die Maximalzahl der Durchläufe der Prozedur an.

### CYCLE-LIMIT = <integer 1..16777215>

Maximalzahl der Durchläufe der Prozedur. Der Standardwert ist 1. Bei einem ungültigen Wert ( $\leq 0$ ) wird der Standardwert genutzt. Bei einer reinen FPASCAN-Transaktion macht ein Abweichen vom Standardwert im Allgemeinen keinen Sinn; die UDS-Online-Utility führt aber keine entsprechende Prüfung durch.

**CYCLE-LIMIT = \*MAX**

Es wird der höchste mögliche Wert ( $2^{24-1}$ ) für die Prozedur-Durchläufe eingesetzt. Dies entspricht quasi einer unbegrenzten Anzahl von Prozedurdurchläufen, sodass ohne Störungen die möglichen Verlagerungen eines Realms in einem Online-Utility-Lauf durchgeführt werden können.

## Such-Modus für Freiplatzsuche festlegen (SET-FPA-SCAN-PARAMETERS)

Mit SET-FPA-SCAN-PARAMETERS legen Sie fest, welcher Such-Modus bei der Freiplatzsuche in einem Realm verwendet wird (First Scan oder Second Scan).

```
SET-FPA-SCAN-PARAMETERS
```

```
SUBSCHEMA-NAME = <structured-name 1..30>
```

```
,REALM-NAME = <structured-name 1..30>
```

```
,SEARCH-MODE = *REUSE / *NOREUSE
```

### **SUBSCHEMA-NAME = <structured-name 1..30>**

Name des Subschemas, in dem der Such-Modus festgelegt werden soll. Es sind nur Namen von Benutzersubschemas erlaubt.

### **REALM-NAME = <structured-name 1..30>**

Name des Realms, in dem der Such-Modus festgelegt werden soll. Der Realm-Name muss ein gültiger Name im definierten Subschema sein. Es sind nur Namen von Benutzerrealms erlaubt. Die entsprechende Prüfung wird bei der Ausführung der Prozedur-Anweisung FPASCAN durchgeführt (siehe [Abschnitt „Startseite für Freiplatzsuche festlegen \(FPASCAN\)“ auf Seite 297](#)).

### **SEARCH-MODE =**

Bestimmt den Such-Modus, d.h. die Seite in einem Realm, ab der für neu einzuspeichernde Sätze und Tabellen, die durch die DMLs MODIFY und STORE von Anwendungsprogrammen verursacht sind, Platz gesucht wird.

### **SEARCH-MODE = \*REUSE**

Es wird ab dem Beginn des Realms nach Platz für neu einzuspeichernde Sätze und Tabellen gesucht (Second Scan). Die Änderung der Speichersuche wird bei \*REUSE sofort wirksam.

### **SEARCH-MODE = \*NOREUSE**

Es wird am Realm-Ende hinter der ersten Seite des zusammenhängenden Bereichs freier Seiten nach Platz für neu einzuspeichernde Sätze und Tabellen gesucht (First Scan). Die Änderung der Speichersuche wird bei \*NOREUSE zunächst vorgemerkt und beim Erreichen des nächsten Konsistenzpunktes wirksam.



## Online-Utility-Parameter festlegen (SET-ONLINE-UTILITY-PARAMETERS )

Mit SET-ONLINE-UTILITY-PARAMETERS wird festgelegt, ob eine Verbindung zum Independent DBH oder zum Linked-in DBH hergestellt werden soll.

Diese SDF-Anweisung ist zwingend die erste zu nutzende Anweisung eines UDS-Online-Utility-Einsatzes. Sie kann danach nicht noch einmal genutzt werden.

```
SET-ONLINE-UTILITY-PARAMETERS
```

```
DBH = *INDEPENDENT / *LINKED-IN  
,CONFIGURATION-NAME = <structured-name 1..17>
```

### **DBH =**

Gibt an, ob eine Verbindung zum Independent DBH oder zum Linked-in DBH hergestellt werden soll.

### **DBH = \*INDEPENDENT**

Es wird eine Verbindung zum Independent DBH hergestellt.

### **DBH = \*LINKED-IN**

Es wird eine Verbindung zum Linked-in DBH hergestellt. Die Möglichkeit, den Linked-in DBH zu nutzen, widerspricht zunächst den eigentlich mit der UDS-Online-Utility verfolgten Verfügbarkeitszielen. Es ist aber durchaus denkbar, dass in konkreten Einsatzfällen der schnellen und kompakten Abwicklung von Funktionen der UDS-Online-Utility der Vorzug gegeben wird, etwa weil eine aus anderen Gründen veranlasste Unterbrechung der Independent Session genutzt werden soll.

### **CONFIGURATION-NAME = <structured-name 1..17>**

Name der Datenbank-Konfiguration, die verwendet werden soll.

Der Konfigurationsname wird syntaktisch geprüft. Danach werden die Konnektierungsmodule nachgeladen und eine Konnektierung zur angegebenen Session wird vorgenommen. Wenn diese Konnektierung scheitert, beendet sich die UDS-Online-Utility mit einer entsprechenden Meldung.

## Preferred-Realm festlegen (SET-PREF-REALM-PARAMETERS )

Mit SET-PREF-REALM-PARAMETERS werden die Parameter für die DML PREFRLM gesetzt. Der Preferred-Realm für eine verteilbare Liste wird neu gesetzt bzw. verändert.

SET-PREF-REALM-PARAMETERS
SUBSCHEMA-NAME= <structured-name 1..30> ,SET-NAME= <structured-name 1..30> ,PREFERRED-REALM-NAME= <structured-name 1..30>

### **SUBSCHEMA-NAME = <structured-name 1..30>**

Subschema-Name. Es sind nur Namen von Benutzersubschemas erlaubt.

### **SET-NAME= <structured-name 1..30>**

Set, dessen Preferred-Realm geändert werden soll. Der Set muss ein gültiger Name im definierten Subschema sein und eine verteilbare Liste sein.

### **PREFERRED-REALM-NAME= <structured-name 1..30>**

Name des neuen Realms, der für den Set SET-NAME als Preferred-Realm gelten soll. Der Realm-Name muss ein gültiger Name im definierten Subschema sein und in der WITHIN-Klausel der DDL bei der Deklaration der Membersatzart angegeben sein.

Die entsprechenden Prüfungen werden erst bei Ausführung der DMLs der UDS-Online-Utility durchgeführt.

## Eigenschaften einer RELOCATE-DML festlegen (SET-RELOCATE-PARAMETERS)

Mit SET-RELOCATE-PARAMETERS werden die Parameter gesetzt, die nötig sind, um eine RELOCATE-DML durchführen zu können.

Die Verlagerung erfolgt in einer Folge von Verlagerungstransaktionen, die in einem oder in mehreren aufeinanderfolgenden Online-Utility-Läufen durchgeführt werden. Der DBH pflegt dabei intern transaktionsübergreifende Informationen über den Fortschritt der Verlagerung (z.B. Quell- und Zielpegel). Mit dem Ende der Session bzw. des Sessionabschnitts oder beim Abschalten der Datenbank bzw. des entsprechenden Realms geht diese Information verloren.

### SET-RELOCATE-PARAMETERS

```

SUBSCHEMA-NAME = <structured-name 1..30>
,REALM-NAME = <structured-name 1..30>
,RELOCATE-TYPE = *RECORD-PAGES(...) / *BASE-LEVEL-TABLE-PAGES(...) /
                *INDEX-LEVEL-TABLE-PAGES(...) / *DISTRIBUTABLE-TABLE-PAGES(...)
*RECORD-PAGES
  INITIALIZE= *ANY / *YES / *NO
  ,PAGES-PER-DML= <integer 1..16777215>
  ,SKIP-ABOVE-FILLING= <integer 1..100>
  ,CLASH-HANDLING= *BREAK-DML / *SKIP-PAGE
*BASE-LEVEL-TABLE-PAGES
  INITIALIZE= *ANY / *YES / *NO
  ,PAGES-PER-DML= <integer 1..16777215>
  ,CLASH-HANDLING= *BREAK-DML / *SKIP-PAGE
*INDEX-LEVEL-TABLE-PAGES
  INITIALIZE= *ANY / *YES / *NO
  ,PAGES-PER-DML= <integer 1..16777215>
*DISTRIBUTABLE-TABLE-PAGES
  INITIALIZE= *ANY / *YES / *NO
  ,PAGES-PER-DML= <integer 1..16777215>
  ,CLASH-HANDLING= *BREAK-DML / *SKIP-PAGE
  ,SET-NAME= <structured-name 1..30>
  ,TARGET-REALM-NAME= <structured-name 1..30>

```

**SUBSCHEMA-NAME = <structured-name 1..30>**

Subschema-Name. Es sind nur Namen von Benutzersubschemas erlaubt.

**REALM-NAME = <structured-name 1..30>**

Realm-Name. Der Realm-Name muss ein gültiger Name im definierten Subschema sein. Es sind nur Namen von Benutzerrealms erlaubt. Namen von temporären Realms sind nicht erlaubt.

In dem Subschema müssen auch alle Realms enthalten sein, die DBTTs enthalten, deren Einträge auf zu verlagernde Sätze und Tabellen in dem Realm verweisen. Die entsprechende Prüfung erfolgt erst bei Ausführung der DMLs. Dagegen müssen Satzarten und Sets des Realms nicht vollständig im angegebenen Subschema enthalten sein. Die Verlagerung von Sätzen und Tabellen erfolgt insofern unabhängig von den Angaben in dem Subschema.

**RELOCATE-TYPE =**

Relocate-Typ. legt die Variante der Verlagerung fest.

**RELOCATE-TYPE = \*RECORD-PAGES(...)**

Verlagert Datensätze und kleine Tabellen innerhalb eines Realms.

**INITIALIZE =**

Legt das Verhalten der RELOCATE-DML fest, wenn die Verlagerung in einem Realm auf mehrere Online-Utility-Läufe verteilt ist.

Immer dann, wenn bei INITIALIZE = \*ANY oder \*YES eine Initialisierung von Quell- und Zielpegel erfolgt, wird implizit die Freiplatzsuche auf den Beginn des Realms positioniert; dies entspricht einem Aufruf von FPASCAN mit SEARCH-MODE=\*REUSE. Insbesondere werden nach einer vollständigen Verlagerung Quell- und Zielpegel noch nicht automatisch neu initialisiert. Die weitere Durchführung von Verlagerungstransaktionen mit INITIALIZE = \*NO oder INITIALIZE = \*ANY wird also immer die Rückkehrinformation bekommen, dass nichts mehr zu tun ist.

**INITIALIZE = \*ANY**

Quell- und Zielpegel werden abhängig von bestimmten Bedingungen neu initialisiert:

- Wurde in einem vorangegangenen Online-Utility-Lauf bereits eine Verlagerung begonnen, dann wird diese Verlagerung fortgesetzt. Quell- und Zielpegel werden nicht neu initialisiert.
- Wurde in einem vorangegangenen Online-Utility-Lauf in diesem Sessionabschnitt bzw. in dem Zeitraum seit dem letzten Zuschalten des betreffenden Realms noch keine Verlagerung begonnen, dann führt die erste Verlagerungstransaktion implizit zu einer Initialisierung von Quell- und Zielpegel.

**INITIALIZE = \*YES**

Quell- und Zielpegel werden auf jeden Fall neu initialisiert.

Bei der ersten Verlagerungstransaktion wird auf jeden Fall eine Initialisierung von Quell- und Zielpegel vorgenommen. Die Verlagerung beginnt somit von neuem.

Diese Einstellung können Sie insbesondere dann nutzen, wenn vorher mit CLASH-HANDLING=\*SKIP-PAGE Zugriffskonflikte mit anderen Anwendertransaktionen dazu geführt haben, dass verlagerbare Seiten übersprungen wurden, um deren Verlagerung nunmehr nachzuholen.

**INITIALIZE = \*NO**

Quell- und Zielpegel werden nicht neu initialisiert.

Eine in einem vorangegangenen Online-Utility-Lauf bereits begonnene Verlagerung wird fortgesetzt. Wurde noch keine Verlagerung begonnen, wird ebenfalls keine Initialisierung der Quell- und Zielpegel durchgeführt.

Wenn Sie eine Verlagerung in einem Realm in mehreren unabhängigen Online-Utility-Läufen durchführen wollen, können Sie mit INITIALIZE=\*NO verhindern, dass eine eigentlich beabsichtigte Fortsetzung zu einem Neubeginn führt. Es könnte ja sein, dass zwischen zwei Online-Utility-Läufen offline Aktivitäten auf dem Realm durchgeführt worden sind, die eine weitere Verlagerung - nunmehr neu initialisiert - nicht mehr sinnvoll machen.

**PAGES-PER-DML= <integer 1..16777215>**

Gibt an, wieviele Quellseiten mit einer RELOCATE-DML maximal geleert werden sollen (Standardwert: 1).

**SKIP-ABOVE-FILLING = <integer 1..100>**

Legt einen Prozentwert fest, der angibt, wieviel die Quellseite höchstens gefüllt sein darf, damit sie geleert wird. Der Standardwert 100 bezieht alle verlagerbaren Seiten ein. Mit einem kleineren Wert können Sie die Verlagerung auf mäßig gefüllte Seiten beschränken. Eine Angabe unter 100 benutzt der DBH auch, um anhand der Freiplatzverwaltungsdaten Seiten von der Verlagerung auszuschließen. Mit dieser Optimierung können Sie somit auch unnötiges Lesen von Datenseiten vermeiden.

**CLASH-HANDLING =**

Legt fest, wie sich die RELOCATE-DML verhält, wenn eine durch eine andere Transaktion gesperrte Quellseite als nächste bearbeitet werden soll.

Beim CLASH-HANDLING wird keine Möglichkeit angeboten, während der Verarbeitung der RELOCATE-DML auf das Ende des Konflikts zu warten, weil hierdurch in ungünstigen Situationen Deadlocks entstehen können und somit Beeinträchtigungen von Anwendungen möglich wären, die parallel zur UDS-Online-Utility ablaufen.

**CLASH-HANDLING = \*BREAK-DML**

Die RELOCATE-DML wird abgebrochen, evtl. in dieser DML oder in vorigen DMLs derselben Transaktion schon durchgeführte Verlagerungen bleiben aber erhalten. Mit der nächsten RELOCATE-DML - in der Regel in einer neuen Transaktion - evtl. nach einer Verzögerung in der UDS-Online-Utility, wird erneut versucht, diese Seite zu verlagern.

Bei CLASH-HANDLING=\*BREAK-DML sind alle gewünschten Seiten verlagert, wenn Quell- und Zielpegel aufeinandertreffen. Bei auftretenden Konflikten kann es aber sein, dass dazu zusätzliche RELOCATE-DMLs benötigt werden.

**CLASH-HANDLING = \*SKIP-PAGE**

Die Seite wird ausgelassen. Es wird auch nicht versucht, in einer Folge-DML die Verlagerung dieser Seite zu wiederholen. Wurden innerhalb einer RELOCATE-DML allerdings bereits zu viele Quell- oder Zielseiten ausgelassen, wird die DML immer abgebrochen.

Bei CLASH-HANDLING=\*SKIP-PAGE wird im Allgemeinen mit einer RELOCATE-DML die gewünschte Anzahl von Seiten (Parameter PAGES-PER-DML) verlagert. Es kann aber sein, dass dann innerhalb von einem RELOCATE-Zyklus nicht alle gewünschten Seiten verlagert wurden. In diesem Fall kann mit einer Folgeaktion (INITIALIZE=\*YES) die Verlagerung in einem späteren Zeitraum abgeschlossen werden, wenn z.B. entsprechende Konflikte unwahrscheinlicher sind.

**RELOCATE-TYPE = \*BASE-LEVEL-TABLE-PAGES(...)**

Verlagert Stufe-0-Seiten mehrstufiger Tabellen innerhalb eines Realms, verschiebt also ganze Stufe-0-Seiten einer Tabelle an den freien Anfang eines Realms.

**INITIALIZE =**

Siehe „[INITIALIZE =](#)“ auf Seite 284.

**PAGES-PER-DML= <integer 1..16777215>**

Gibt an, wieviele Quellseiten mit einer RELOCATE-DML maximal geleert werden sollen (Standardwert: 1).

**CLASH-HANDLING =**

Siehe „[CLASH-HANDLING =](#)“ auf Seite 285.

**RELOCATE-TYPE = \*INDEX-LEVEL-TABLE-PAGES(...)**

Verlagert Stufe-N-Seiten mehrstufiger Tabellen innerhalb eines Realms, verschiebt also Stufe-N-Seiten einer Tabelle an den freien Anfang eines Realms.



Eine Verlagerung (RELOCATE) wird immer innerhalb einer Transaktion durchgeführt. Die aktuelle Transaktion der UDS-Online-Utility wird mit der Anweisung READY UPDATE an den DBH eröffnet. Dies gilt für alle Relocate-Typen, außer \*INDEX-LEVEL-TABLE-PAGES. Hierfür ist ein READY EXCLUSIVE UPDATE zwingend, damit eine problemlose Verlagerung und Aktualisierung der nötigen Verweise erfolgen kann.

**INITIALIZE =**

Siehe „[INITIALIZE =](#)“ auf Seite 284.

**PAGES-PER-DML= <integer 1..16777215>**

Gibt an, wieviele Quellseiten mit einer RELOCATE-DML maximal geleert werden sollen (Standardwert: 1).

**RELOCATE-TYPE = \*DISTRIBUTABLE-TABLE-PAGES(...)**

Verlagert Stufe-0-Seiten einer verteilbaren Liste von einem Realm in einen anderen Realm.

**INITIALIZE =**

Siehe „[INITIALIZE =](#)“ auf Seite 284.

**PAGES-PER-DML= <integer 1..16777215>**

Gibt an, wieviele Quellseiten mit einer RELOCATE-DML maximal geleert werden sollen (Standardwert: 1).

**CLASH-HANDLING =**

Siehe „[CLASH-HANDLING =](#)“ auf Seite 285.

**SET-NAME= <structured-name 1..30>**

Legt fest, für welche verteilbare Liste die Verlagerung stattfinden soll.

Der angegebene Set muss eine verteilbare Liste sein, sonst findet keine Verschiebung statt. In diesem Fall wird eine Fehlermeldung ausgegeben und der Ablauf wird abgebrochen.

Die entsprechenden Prüfungen werden erst bei Ausführung der DMLs der UDS-Online-Utility durchgeführt.

**TARGET-REALM-NAME= <structured-name 1..30>**

Legt den Zielrealm fest, also den Realm, in den verlagert werden soll.

Dieser Realm muss sich von dem Quellrealm (Parameter REALM-NAME) unterscheiden und muss im eingestellten Subschema (Parameter SUBSCHEMA-NAME) bekannt sein und zur verteilbaren Liste (WITHIN-Klausel der DDL-RECORD-Deklaration der Membersatzart) gehören.

Die entsprechenden Prüfungen werden erst bei Ausführung der DMLs der UDS-Online-Utility durchgeführt.

## Eigenschaften einer REORGPPP-DML definieren (SET-REORGANIZE-PPP-PARAMETERS)

SET-REORGANIZE-PPP-PARAMETERS legt die Parameter fest, die benötigt werden, um die Ausführung einer REORGPPP-DML zu ermöglichen. Die PPP-Reorganisation findet in einer Sequenz von Reorganisationsaktionen statt, die in einer oder mehreren aufeinander folgenden Online-Hilfsprogramm-Ausführungen ausgeführt werden. Hier behält der DBH die übergreifenden Transaktionsinformationen zum Fortschritt der PPP-Reorganisation, z. B. die aktuelle Seitennummer, intern bei. Diese Informationen gehen verloren, wenn die Session oder der Session-Abschnitt endet oder wenn die Datenbank des betroffenen Realms getrennt wird.

SET-REORGANIZE-PPP-PARAMETERS	
SUBSCHEMA-NAME	= <structured-name 1..30>
,REALM-NAME	= <structured-name 1..30>
,INITIALIZE	= * <u>ANY</u> / *YES / *NO
,PAGES-PER-DML	= <integer 1..16777215>
,CLASH-HANDLING	= * <u>BREAK-DML</u> / *SKIP-PAGE

### **SUBSCHEMA-NAME = <structured-name 1..30>**

Name des Subschemas. Nur Namen von Benutzersubschemas sind zulässig.

### **REALM-NAME = <structured-name 1..30>**

Name des Realms. Der Realm-Name muss ein gültiger Name im definierten Subschema sein. Nur Namen des Benutzerrealms sind zulässig. Namen von temporären Realms sind nicht zulässig. Das Subschema muss auch alle Realms enthalten, die die DBTTs enthalten, deren Einträge auf Sätze und Tabellen in dem Realm verweisen, in dem PPPs (Probable Position Pointer) aktualisiert werden müssen. Die entsprechende Prüfung findet nur statt, wenn die DMLs ausgeführt werden. Satzarten und Realm-Sets müssen dagegen nicht vollständig in dem angegebenen Subschema enthalten sein. Die Reorganisation von Pointern in Sätzen und Tabellen findet insoweit unabhängig von den Spezifikationen im Subschema statt.



**INITIALIZE =**

Definiert das Verhalten der REORGPPP-DML, wenn die PPP-Reorganisation in einem Realm über mehrere Online-Hilfsprogramm-Ausführungen verteilt ist.

Die Initialisierung der aktuellen Seite findet statt, wenn INITIALIZE = \*ANY oder \*YES. Insbesondere wird die aktuelle Seitennummer nach einer vollständigen PPP-Reorganisation nicht automatisch neu initialisiert. Eine weitere Ausführung von Aktionen zur PPP-Reorganisation über INITIALIZE = \*NO oder INITIALIZE = \*ANY führt zur Rückgabe der Information, dass es nichts mehr zu tun gibt.

**INITIALIZE = \*ANY**

Die aktuelle Seitennummer wird nach den folgenden Bedingungen neu initialisiert:

- Wenn die PPP-Reorganisation bereits in einer früheren Hilfsprogramm-Routine-Ausführung gestartet wurde, wird diese Reorganisation fortgesetzt. Die aktuelle Seitennummer wird nicht neu initialisiert.
- Wenn keine PPP-Reorganisation in einer früheren Online-Hilfsprogramm-Ausführung in diesem Session-Abschnitt oder im Zeitraum seit der letzten Verbindung des betroffenen Realms gestartet wurde, führt die erste Aktion zur PPP-Reorganisation implizit zur Initialisierung der aktuellen Seitennummer.

**INITIALIZE = \*YES**

Die aktuelle Seitennummer wird immer neu initialisiert.

Die PPP-Reorganisation beginnt entsprechend erneut.

Sie können diese Einstellung verwenden, um PPP auf diesen Seiten retroaktiv neu zu organisieren. Dies vor allem, wenn ein früherer Parameter CLASH-HANDLING=\*SKIP-PAGE anzeigt, dass Zugriffskonflikte mit anderen Benutzertransaktionen zum Überspringen bestimmter Seiten geführt haben.

**INITIALIZE = \*NO**

Die aktuelle Seitennummer wird nicht neu initialisiert.

Eine PPP-Reorganisation, die bereits in der früheren Online-Hilfsprogramm-Ausführung gestartet wurde, wird fortgesetzt. Wenn keine PPP-Reorganisation gestartet wurde, wird die aktuelle Seitennummer nicht initialisiert.

**PAGES-PER-DML = <integer 1..16777215>**

Legt die maximale Anzahl an Seiten fest, auf denen PPPs mit einer REORGPPP-DML aktualisiert werden müssen. Standardwert: 1

**CLASH-HANDLING =**

Definiert das Verhalten der REORGPPP-DML, wenn eine von einer anderen Transaktion gesperrte Seite als Nächstes verarbeitet werden muss.

CLASH-HANDLING bietet nicht die Möglichkeit zu warten, bis der Konflikt endet, während die REORGPPP-DML verarbeitet wird, weil dies in ungünstigen Situationen zu Deadlocks führen und Anwendungen beeinträchtigen kann, die parallel zum UDS-Online-Hilfsprogramm ausgeführt werden.

**CLASH-HANDLING = \*BREAK-DML**

Die REORGPPP-DML wird abgebrochen, aber alle PPP-Reorganisationen, die bereits in dieser DML oder in früheren DMLs derselben Transaktion ausgeführt wurden, werden beibehalten. Mit der nächsten REORGPPP-DML, normalerweise in einer neuen Transaktion und wenn möglich nach einer Verzögerung im UDS-Online-Hilfsprogramm, wird erneut versucht, die PPP-Reorganisation auf dieser Seite auszuführen.

Wenn CLASH-HANDLING=\*BREAK-DML, werden PPPs auf allen erforderlichen Seiten aktualisiert, wenn die letzte Seite im Realm erreicht wird. Bei Konflikten können jedoch zusätzliche REORGPPP-DMLs für diesen Zweck erforderlich sein.

**CLASH-HANDLING = \*SKIP-PAGE**

Die Seite wird übersprungen. Bei einer nachfolgenden DML wird nicht versucht, die PPP-Reorganisation für diese Seite zu wiederholen. Wenn jedoch zu viele Seiten in einer REORGPPP-DML übersprungen werden, wird die DML abgebrochen.

Wenn CLASH-HANDLING=\*SKIP-PAGE, wird die erforderliche Anzahl an Seiten (Parameter PAGES-PER-DML) generell mit einer REORGPPP-DML bearbeitet. Es kann jedoch passieren, dass nicht alle erforderlichen Seiten in einem REORGPPP-Zyklus bearbeitet werden. In diesem Fall kann eine Folgeaktion (INITIALIZE=\*YES) verwendet werden, um die PPP-Reorganisation später abzuschließen, wenn zum Beispiel entsprechende Konflikte weniger wahrscheinlich sind.

## Aktuell gültige Parameter für DML FPASCAN ausgeben (SHOW-FPA-SCAN-PARAMETERS)

Mit SHOW-FPA-SCAN-PARAMETERS werden die aktuell gültigen Parameter für die DML FPASCAN angezeigt.

SHOW-FPA-SCAN-PARAMETERS

## Prozedur ausgeben (SHOW-PROCEDURE)

Mit SHOW-PROCEDURE wird einmal eine Prozedur ausgegeben. Es können auch Standardprozeduren ausgegeben werden.

SHOW-PROCEDURE

PROCEDURE-NAME = <structured-name 1..20> / *STDRELOC / *STDFPASCAN / *STDREPPP
--

### PROCEDURE-NAME =

Name der Prozedur, die ausgegeben werden soll.

### PROCEDURE-NAME = <structured-name 1..20>

Name der benutzerdefinierten Prozedur, die ausgegeben werden soll.

### PROCEDURE-NAME = \*STDRELOC

Die Standardprozedur \*STDRELOC (siehe [Abschnitt „Vordefinierte Standardprozeduren“ auf Seite 305](#)) soll ausgegeben werden.

### PROCEDURE-NAME = \*STDFPASCAN

Die Standardprozedur \*STDFPASCAN (siehe [Abschnitt „Vordefinierte Standardprozeduren“ auf Seite 305](#)) soll ausgegeben werden.

### PROCEDURE-NAME = \*STDREPPP

Die Standardprozedur \*STDREPPP (siehe [Abschnitt „Vordefinierte Standardprozeduren“ auf Seite 305](#)) soll ausgegeben werden.

## **Aktuell gültige Parameter für DML PREFRLM ausgeben (SHOW-PREF-REALM-PARAMETERS )**

Mit SHOW-PREF-REALM-PARAMETERS werden die aktuell gültigen Parameter für die DML PREFRLM ausgegeben.

SHOW-PREF-REALM-PARAMETERS

## **Aktuell gültige Parameter für DML RELOCATE ausgeben (SHOW-RELOCATE-PARAMETERS)**

Mit SHOW-RELOCATE-PARAMETERS werden alle Parameter der DML RELOCATE ausgegeben. Alle irrelevanten Parameter für den gerade eingestellten RELOCATE-TYPE haben den Wert "UNDEFINED".

SHOW-RELOCATE-PARAMETERS

## **Aktuell gültige Parameter für DML REORGPPP ausgeben (SHOW-REORGANIZE-PPP-PARAMETERS)**

Mit SHOW-REORGANIZE-PPP-PARAMETERS werden die aktuell gültigen Parameter für DML REORGPPP ausgegeben.

SHOW-REORGANIZE-PPP-PARAMETERS

## Aktuellen Wert einer Variablen ausgeben (SHOW-VARIABLE)

Mit SHOW-VARIABLE wird einmal der aktuelle Wert einer Variablen ausgegeben.

Bei der Ausführung der Anweisung wird eine S-Variable gleichen Namens typgerecht mit dem Wert der ausgegebenen Variablen versorgt. Wenn die S-Variable noch nicht existiert, wird sie mit Standardparametern (z.B. sichtbar auf Prozedurebene) angelegt.

SHOW-VARIABLE
VARIABLE-NAME = <structured-name 1..20> / RCODE / STATUSCODE / RELOC-DESTINATION / RELOC-FREED-PAGES / RELOC-LOCKED-PAGES / RELOC-ORIGIN / RELOC-ORIGIN-LOCKS / RELOC-DEST-LOCKS / REORG-PPP-CURRENT / REORG-PPP-PAGES / REORG-PPP-LOCKED

**VARIABLE-NAME= <structured-name 1..20>**

Name der Variablen, deren Wert ausgegeben werden soll.

**VARIABLE-NAME = RCODE / STATUSCODE / RELOC-DESTINATION / RELOC-FREED-PAGES / RELOC-LOCKED-PAGES / RELOC-ORIGIN / RELOC-ORIGIN-LOCKS / RELOC-DEST-LOCKS / REORG-PPP-CURRENT / REORG-PPP-PAGES / REORG-PPP-LOCKED**

Name einer vordefinierten Variable (siehe [Abschnitt „Vordefinierte Variablen“ auf Seite 304](#)).

## 8.9 Prozedur-Anweisungen der UDS-Online-Utility

Im Folgenden sind die Prozedur-Anweisungen aufgelistet, die die UDS-Online-Utility als DML- und Prozeduraufträge abwickelt und die in Textform im Parameter CODE der DECLARE-PROCEDURE-Anweisung übergeben werden (siehe [Abschnitt „Prozedur definieren \(DECLARE-PROCEDURE\)“ auf Seite 275](#)).

Anweisung	Bedeutung
<u>ADD</u>	Wert zu einer Variablen addieren
<u>BREAK</u>	Prozedurfolge sofort beenden
<u>END</u>	Eingabe von Prozedur-Anweisungen beenden
<u>EXIT</u>	Prozedurfolge nach aktuellem Durchlauf beenden
<u>FINISH</u>	Aktuelle Transaktion beenden
<u>FPASCAN</u>	Startseite für Freiplatzsuche festlegen
<u>MOVE</u>	Wert einer Variablen festlegen
<u>PREFRLM</u>	Preferred-Realm für eine verteilbare Liste neu setzen bzw. verändern
<u>READY</u> <u>[EXCLUSIVE]</u> <u>UPDATE</u>	Aktuelle Transaktion der UDS-Online-Utility starten
<u>RELOCATE</u>	Verlagerung durchführen
<u>REMARK</u> bzw. * <sub>~</sub>	Kommentar einfügen
<u>REORGPPP</u>	PPP-Reorganisation durchführen
<u>SHOW</u>	Wert einer Variablen ausgeben
<u>WAIT</u>	Wartezeit festlegen

Tabelle 19: Prozedur-Anweisungen der UDS-Online-Utility

## Wert zu einer Variablen addieren (ADD)

---

$$\text{ADD } name, \left\{ \begin{array}{l} value \\ variable \end{array} \right\} [ , condition]$$

---

*name*

Name der Variablen, zu der ein Wert addiert werden soll. Die Variable *name* muss vom Typ numerisch sein.

*value*

Wert, der zu der Variablen *name* addiert werden soll

*variable*

Variable, deren Wert zu der Variablen *name* addiert werden soll. Die Variable *variable* muss vom Typ numerisch sein.

*condition*

Bedingung, unter der die Anweisung ADD ausgeführt werden soll. Eine Beschreibung der Syntax von Bedingungen finden Sie im [Abschnitt „Syntax von Bedingungen“ auf Seite 306](#).

Mit der Anweisung ADD wird in der vorher definierten Variable *name* der Wert *value* oder der Wert der Variablen *variable* aufaddiert. Die Variablen müssen vom Typ numerisch sein. Bei Angabe einer Bedingung muss diese den Wert *true* haben. Ohne Angabe einer Bedingung wird immer addiert.

## Prozedurfolge sofort beenden (BREAK)

---

BREAK *condition*

---

*condition*

Bedingung, unter der die Anweisung BREAK ausgeführt werden soll. Eine Beschreibung der Syntax von Bedingungen finden Sie im [Abschnitt „Syntax von Bedingungen“ auf Seite 306](#).

Die Anweisung BREAK beendet die mit REPEAT-PROCEDURE gestartete Prozedurfolge sofort, wenn die Bedingung *condition* den Wert *true* hat. Es werden keine weiteren Prozedur-Anweisungen in dem somit letzten Durchlauf der Prozedur mehr ausgeführt. BREAK kann insbesondere dazu genutzt werden, die Prozedurbearbeitung zu beenden, wenn ein READY nicht erfolgreich ist.

## Eingabe von Prozedur-Anweisungen beenden (END)

---

END

---

Die Anweisung END beendet eine mit DECLARE-PROCEDURE gestartete Eingabefolge von Prozedur-Anweisungen über CODE=\*SYSDTA. Bei Eingabe über CODE=*c-string* oder *<filename>* wird das Ende der Prozedur durch das Ende des c-strings bzw. durch das Dateiende erkannt.

## Prozedurfolge nach aktuellem Durchlauf beenden (EXIT)

---

EXIT *condition*

---

*condition*

Bedingung, unter der die Anweisung EXIT ausgeführt werden soll. Eine Beschreibung der Syntax von Bedingungen finden Sie im [Abschnitt „Syntax von Bedingungen“ auf Seite 306](#).

Die Anweisung EXIT beendet die mit REPEAT-PROCEDURE gestartete Prozedurfolge nach Beendigung des aktuellen Durchlaufs, wenn die Bedingung *condition* den Wert *true* hat.



---

## Aktuelle Transaktion beenden (FINISH)

---

FINISH [WITH CANCEL]

---

Mit der Anweisung FINISH wird eine FINISH-DML an den DBH geschickt, das die aktuelle Transaktion der UDS-Online-Utility beendet. Die DML liefert einen Statuscode zurück, der in der vordefinierten Variablen RCODE bereitsteht (siehe [Abschnitt „Vordefinierte Variablen“ auf Seite 304](#)).

Nach einer DML FPASCAN ist die Angabe WITH CANCEL wirkungslos (siehe [Abschnitt „Startseite für Freiplatzsuche festlegen \(FPASCAN\)“ auf Seite 297](#)).

---

## Startseite für Freiplatzsuche festlegen (FPASCAN)

---

FPASCAN

---

Mit der Anweisung FPASCAN wird eine FPASCAN-DML an den DBH geschickt, das die Startseite für die Freiplatzsuche festlegt. Die Startseite ist in der SDF-Anweisung SET-FPA-SCAN-PARAMETERS angegeben (siehe [Abschnitt „Such-Modus für Freiplatzsuche festlegen \(SET-FPA-SCAN-PARAMETERS\)“ auf Seite 280](#)).

Wird bei SET-FPA-SCAN-PARAMETERS SEARCH-MODE=\*REUSE angegeben, so wird freier Platz vom Beginn des Realms gesucht (Suchmodus „Second Scan“).

Bei SEARCH-MODE=\*NOREUSE wird benötigter Speicherplatz in dem zusammenhängend freien Bereich am Realmende gesucht und PLACEMENT OPTIMIZATION berücksichtigt (Suchmodus „First Scan“). Die Änderung der Speichersuche wird bei \*REUSE sofort wirksam und bei \*NOREUSE zunächst vorgemerkt und beim Erreichen des nächsten Konsistenzpunktes wirksam. Diese Aktionen werden auch durch ein nachfolgendes explizites und internes FINISH WITH CANCEL nicht mehr zurückgenommen. Insofern wird WITH CANCEL ignoriert.

Die DML liefert einen Statuscode zurück, der in vordefinierten Variablen RCODE bereitsteht (siehe [Abschnitt „Vordefinierte Variablen“ auf Seite 304](#)).

## Wert einer Variablen festlegen (MOVE)

---

$$\text{MOVE } name, \left\{ \begin{array}{l} value \\ variable \end{array} \right\} [, condition]$$

---

*name*

Name der Variablen, deren Wert festgelegt werden soll.

*value*

Wert, der in der Variablen *name* eingetragen werden soll

*variable*

Variable, deren Wert in der Variablen *name* eingetragen werden soll.

*condition*

Bedingung, unter der die Anweisung MOVE ausgeführt werden soll. Eine Beschreibung der Syntax von Bedingungen finden Sie im [Abschnitt „Syntax von Bedingungen“ auf Seite 306](#).

Mit der Anweisung MOVE wird in der vorher deklarierten Variable *name* der Wert *value* eingetragen oder der Wert der vordefinierten oder deklarierten Variablen *variable* übernommen. Bei Angabe einer Bedingung *condition* muss diese den Wert *true* haben, damit die Prozedur-Anweisung ausgeführt wird. Ohne Angabe einer Bedingung wird immer der entsprechende Wert eingetragen. Bei einer Wertübernahme aus einer anderen Variablen müssen die Typen der Variablen zusammenpassen.

## Preferred-Realm für eine verteilbare Liste neu setzen bzw. verändern (PREFRLM)

---

PREFRLM

---

Für jede verteilbare Liste wird beim Datenbankaufbau ein Preferred-Realm vergeben, in den der DBH neue Stufe-0-Seiten dieser Liste ablegt.

Mit der Anweisung PREFRLM wird eine DML an den DBH geschickt, das den Preferred-Realm für eine verteilbare Liste neu setzt bzw. verändert.

Die Parameter für die Anweisung PREFRLM werden mit der SDF-Anweisung SET-PREF-REALM-PARAMETERS bestimmt.

---

## Aktuelle Transaktion der UDS-Online-Utility starten (READY UPDATE)

---

READY [EXCLUSIVE] UPDATE

---

Mit der Anweisung READY UPDATE wird eine READY-DML an den DBH geschickt, das die aktuelle Transaktion der UDS-Online-Utility startet.

Der Name des Realms, der bearbeitet werden soll, muss vor der Ausführung mit der Anweisung SET-RELOCATE-PARAMETERS angegeben worden sein (siehe [Abschnitt „Eigenschaften einer RELOCATE-DML festlegen \(SET-RELOCATE-PARAMETERS\)“ auf Seite 283](#)). Die Initialisierung einer Folge von Verlagerungstransaktionen orientiert sich an dem Parameter INITIALIZE.

Die Angabe von EXCLUSIVE ist für den RELOCATE-TYPE=\*INDEX-LEVEL-TABLE-PAGES bei einer Verlagerungstransaktion zwingend. Die Angabe EXCLUSIVE ist für die restlichen RELOCATE-Typen nicht erlaubt und wird mit der folgenden Fehlermeldung abgewiesen: USAGE MODE EXCLUSIVE NOT ALLOWED .

Die Angabe des Parameters UPDATE ist bei einer Verlagerungstransaktion zwingend. Die DML liefert einen Statuscode zurück, der in der vordefinierten Variablen RCODE bereitsteht (siehe [Abschnitt „Vordefinierte Variablen“ auf Seite 304](#)).

---

## Verlagerung durchführen (RELOCATE)

---

RELOCATE

---

Mit der Anweisung RELOCATE wird eine RELOCATE-DML an den DBH geschickt, das die Verlagerung von Sätzen und Tabellen durchführt.

Die Menge der mit einer DML durchzuführenden Verlagerungen wird durch den Parameter PAGES-PER-DML in der Anweisung SET-RELOCATE-PARAMETERS bestimmt (siehe [Abschnitt „Eigenschaften einer RELOCATE-DML festlegen \(SET-RELOCATE-PARAMETERS\)“ auf Seite 283](#)).

Die DML liefert einen Statuscode zurück, der in den vordefinierten Variablen RCODE und STATUSCODE bereitsteht (siehe [Abschnitt „Vordefinierte Variablen“ auf Seite 304](#)). Daneben sind bei normaler Beendigung der DMLs auch die vordefinierten Variablen RELOCDESTINATION, RELOC-ORIGIN, RELOC-FREED-PAGES und RELOC-LOCKED-PAGES versorgt, mit denen u.a. ermittelt werden kann, in welchem Umfang die Verlagerung fortgeschritten ist.

## Kommentar einfügen (REMARK)

---

```
{REMARK  
*  
}
```

---

Eine Anweisung, die mit REMARK oder '\*' beginnt, wird in der Ausführung als Kommentar interpretiert und bei der Ausführung einer Prozedur überlesen.

Eine einzelne Kommentar-Anweisung darf maximal 75 Zeichen umfassen. Das Zeichen ';' ist im Kommentar nur erlaubt, wenn es in Hochkommata eingeschlossen ist, weil es ansonsten das Kommandoende anzeigt.

## PPP-Reorganisation durchführen (REORGPPP)

---

REORGPPP

---

Die REORGPPP-Anweisung sendet eine REORGPPP-DML an den DBH, der die PPP-Reorganisation ausführt.

Die Anzahl an Seiten, auf denen eine PPP-Reorganisation mit einer DML ausgeführt werden muss, wird durch den Parameter PAGES-PER-DML in der Anweisung SET-REORGANIZE-PPP-PARAMETERS festgelegt.

Die DML gibt einen Statuscode zurück, der in den vordefinierten Variablen RCODE und STATUSCODE verfügbar ist (siehe [Abschnitt „Vordefinierte Variablen“ auf Seite 304](#)). Wenn die DML normal endet, stellen die vordefinierten Variablen REORG-PPP-CURRENT, REORG-PPP-PAGES und REORG-PPP-LOCKED daneben Werte bereit, über die zum Beispiel der Fortschritt der PPP-Reorganisation kontrolliert werden kann.

Wenn alle Realms in der Datenbank auf diese Weise bearbeitet und PPPs auf allen Seiten erfolgreich aktualisiert werden, müssen die neuen Seiten bei einer Online-Realm-Erweiterung nicht formatiert werden. Alle Benutzerrealms müssen deshalb in dem Subschema enthalten sein.

Hierdurch wird die Leistung erheblich verbessert.

## Wert einer Variablen ausgeben (SHOW)

---

`SHOW name[,condition]`

---

*name*

Name der Variablen, deren Wert ausgegeben werden soll.

*condition*

Bedingung, unter der die Anweisung SHOW ausgeführt werden soll. Eine Beschreibung der Syntax von Bedingungen finden Sie im [Abschnitt „Syntax von Bedingungen“ auf Seite 306](#).

Mit der Anweisung SHOW kann der aktuelle Wert einer vordefinierten oder explizit definierten Variablen ausgegeben werden, ggf. abhängig von einer erfüllten Bedingung. Bei Angabe einer Bedingung muss diese den Wert *true* haben, damit die Prozedur-Anweisung ausgeführt wird. Ohne Angabe einer Bedingung wird immer der entsprechende Wert ausgegeben.

Bei der Ausführung der Anweisung wird eine S-Variable gleichen Namens typgerecht mit dem Wert der ausgegebenen Variablen versorgt. Wenn die S-Variable noch nicht existiert, wird sie mit Standardparametern (z.B. sichtbar auf Prozedurebene) angelegt.

## Wartezeit festlegen (WAIT)

---

`WAIT n[,condition]`

---

*n*

Anzahl Sekunden, die gewartet werden soll.

*condition*

Bedingung, unter der die Anweisung WAIT ausgeführt werden soll. Eine Beschreibung der Syntax von Bedingungen finden Sie im [Abschnitt „Syntax von Bedingungen“ auf Seite 306](#).

Mit der Anweisung WAIT wird *n* Sekunden gewartet, ggf. abhängig von einer erfüllten Bedingung. Bei Angabe einer Bedingung muss diese den Wert *true* haben, damit die Prozedur-Anweisung ausgeführt wird. Ohne Angabe einer Bedingung wird immer gewartet.

## 8.10 Fehlerbehandlung der UDS-Online-Utility

Die UDS-Online-Utility wertet die Status-Codes der Prozedurkommandos aus, die an den DBH geschickt werden und reagiert teilweise darauf.

1. Sofern anhand des Status-Codes erkennbar ist, dass eine Transaktion beendet wurde, werden nachfolgende DBH-Kommandos bis zum nächsten READY innerhalb des gleichen Prozedurzyklus nicht ausgeführt.
2. Die vordefinierten Variablen RCODE und STATUSCODE (siehe [Abschnitt „Vordefinierte Variablen“ auf Seite 304](#)) werden unmittelbar vor der Ausführung eines DBH-Kommandos initialisiert.
3. Folgende Status-Codes führen zum Abbruch der Prozedur:

020, 022, 042, 092, 093, 099, 103, 113, 123, 131, 132, 134, 136, 137, 142, 144, 145, 146, 151, 152, 154, 155, 161, 163, 164, 165, 166, 782, 783, 784, 785, 786, 789, 805, 901, 950, 954.

In all diesen Fällen liegt eine Störung vor, die bei einem erneuten Durchlauf eines Prozedurzyklus ohne zusätzliche Maßnahmen weiterhin zu erwarten ist. Die Prozedur wird im Allgemeinen erst am Ende des aktuellen Prozedurzyklus abgebrochen.

DML-Aufträge bis zum Ende des aktuellen Prozedurzyklus werden unterdrückt, wenn offensichtlich keine Transaktion mehr offen ist. Sonstige Prozeduraufträge (z.B. MOVE, SHOW, WAIT) werden bis zum Ende des Prozedurzyklus noch ausgeführt.

Bei der DML FPASCAN kann auch der Status-Code 162 auftreten, der ebenfalls zum Abbruch der Prozedur führt.

Eine ausführliche Beschreibung der Status-Codes finden Sie in den Handbüchern „[Anwendungen programmieren](#)“ und „[Meldungen](#)“.

## 8.11 Vordefinierte Variablen

Einige Variablen der UDS-Online-Utility sind vordefiniert und können ohne Deklaration in Anweisungen und Bedingungen genutzt werden.

Variable	Bedeutung
RCODE	Anweisungscode und Statuscode (5 Byte abdruckbar) einer DML READY, RELOCATE, FPASCAN oder FINISH
STATUSCODE	Statuscode (3Byte abdruckbar) einer DML READY, RELOCATE, FPASCAN oder FINISH.
RELOC-DESTINATION	Blocknummer des aktuellen Zielpegels der Verlagerung
RELOC-DEST-LOCKS	Anzahl der bei der letzten normal durchgeführten RELOCATE-DML von anderen Transaktionen gesperrten Zielseiten
RELOC-FREED-PAGES	Anzahl der Datenbankseiten, aus denen mit der letzten normal durchgeführten RELOCATE-DML Sätze bzw. Tabellen verlagert wurden. Der Wert bleibt auch erhalten, wenn die Transaktion später zurückgesetzt wird.
RELOC-LOCKED-PAGES	Anzahl der bei der letzten normal durchgeführten RELOCATE-DML von anderen Transaktionen gesperrten Seiten; Summe aus RELOC-DEST-LOCKS und RELOC-ORIGIN-LOCKS
RELOC-ORIGIN	Blocknummer des aktuellen Quellpegels der Verlagerung
RELOC-ORIGIN-LOCKS	Anzahl der bei der letzten normal durchgeführten RELOCATE-DML von anderen Transaktionen gesperrten Quellseiten
REORG-PPP-CURRENT	Seitennummer für die nächste PPP-Reorganisation
REORG-PPP-LOCKED	Anzahl der bei der letzten normal durchgeführten REORGPPP-DML von anderen Transaktionen gesperrten Seiten.
REORG-PPP-PAGES	Anzahl der Seiten, auf denen PPPs bei der letzten normal durchgeführten REORGPPP-DML reorganisiert wurden

Tabelle 20: Vordefinierte Variablen der UDS-Online-Utility



## 8.12 Vordefinierte Standardprozeduren

Es wird für jede Aufgabe der UDS-Online-Utility eine Standardprozedur bereitgestellt.

Die Standardprozedur **\*STDRELOC** ermöglicht eine einfache Folge von Verlagerungstransaktionen, die automatisch beendet wird, wenn die erste READY UPDATE-Anweisung fehlschlägt oder Quell- und Zielpegel aufeinandertreffen. Sie entspricht somit

```
READY UPDATE;RELOCATE;  
EXIT COND=STATUSCODE EQ 010;FINISH
```

Die Standardprozedur **\*STDFPASCAN** führt eine FPASCAN-Transaktion durch, die aber nicht durchgeführt wird, wenn die READY UPDATE-Anweisung fehlschlägt. Sie entspricht somit

```
READY UPDATE;FPASCAN;FINISH
```

Die Standardprozedur **\*STDREPPP** ist gleichbedeutend mit:

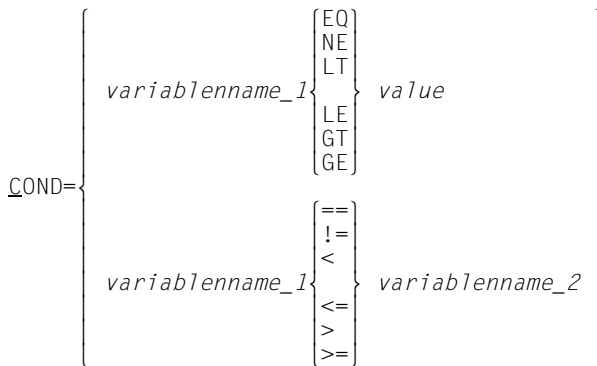
```
READY UPDATE; REORGPMP; EXIT COND = STATUSCODE EQ 010; FINISH;
```

## 8.13 Status-Codes

Eine Beschreibung der Status-Codes der UDS-Online-Utility finden Sie in den Handbüchern „[Anwendungen programmieren](#)“ und „[Meldungen](#)“.

## 8.14 Syntax von Bedingungen

Bedingungen in den Prozedur-Anweisungen können Sie folgendermaßen angeben:



Eine Bedingung wird auf Prozedurebene durch COND= oder C= eingeleitet. Die Bedingung selbst besteht aus drei durch Blank getrennte Strings:

- Der erste String ist der Name einer vordefinierten oder deklarierten Variablen.
- Der zweite String ist der Vergleichsoperator (EQ, NE, LT, LE, GT, GE, ==, !=, <, <=, >, >=).
- Der dritte String ist ein Wert oder der Name einer Variablen. Der Typ eines Wertes muss zum Typ der im ersten String benannten Variablen passen.

Bei Vergleichsoperator ==, !=, <, <=, > und >= wird der dritte String als Name einer Variablen interpretiert. Diese muss zum Typ der Variablen des ersten Strings passen.

Bei Bedingungen mit den vordefinierten Variablen RCODE und STATUSCODE ist zu beachten, dass es sich um Strings und nicht um Zahlen handelt. Insofern müssen auch führende Nullen explizit angegeben werden.

Die Bedingungen werden ausgewertet, wenn die entsprechende Prozedur-Anweisung ausgeführt wird.

## 8.15 Beispiele

Im Folgenden finden Sie einige Beispiele zur Verwendung der UDS-Online-Utility.

### **Beispiel 1: Verlagerung einer festen Anzahl von Seiten mit der Standardprozedur**

Sie möchten etwa 2000 Seiten nach vorne verlagern. Sie benutzen die Standardprozedur \*STDRELOC.

```
//SET-ONLINE-UTILITY-PARAMETERS -  
// DBH=*INDEPENDENT, -  
// CONFIGURATION-NAME=SESSION1  
//SET-RELOCATE-PARAMETERS -  
// SUBSCHEMA-NAME=SUB1, -  
// REALM-NAME=AREA1, -  
// PAGES-PER-DML=2  
//REPEAT-PROCEDURE -  
// PROCEDURE-NAME=*STDRELOC, -  
// CYCLE-LIMIT=1000  
//SHOW-VARIABLE RELOC-ORIGIN  
//SHOW-VARIABLE RELOC-DESTINATION
```

### **Beispiel 2: Verlagerung aller möglicher Seiten mit der Standardprozedur**

Sie möchten möglichst viel nach vorne verlagern. Sie benutzen die Standardprozedur \*STDRELOC. Der Realm hat 10000 Seiten.

```
//SET-ONLINE-UTILITY-PARAMETERS -  
// DBH=*INDEPENDENT, -  
// CONFIGURATION-NAME=SESSION2  
//SET-RELOCATE-PARAMETERS -  
// SUBSCHEMA-NAME=SUB2, -  
// REALM-NAME=AREA2, -  
// PAGES-PER-DML=1  
//REPEAT-PROCEDURE -  
// PROCEDURE-NAME=*STDRELOC -  
// CYCLE-LIMIT=10000  
//SHOW-VARIABLE RELOC-ORIGIN
```

### Beispiel 3: Vermeidung von Kollisionen

Sie möchten möglichst viel nach vorne verlagern. Irgendwann wird die Datenbank aber abgeschaltet. Dann soll sich die UDS-Online-Utility sofort beenden. Der Realm hat 10000 Seiten. Bei Hinweisen auf Kollisionen (hier: mehr als 5) soll sich die UDS-Online-Utility ebenfalls beenden. Normalerweise werden in einer Transaktion drei Seiten geleert.

```
//SET-ONLINE-UTILITY-PARAMETERS -
// DBH=*INDEPENDENT, -
// CONFIGURATION-NAME=SESSION3
//SET-RELOCATE-PARAMETERS -
// SUBSCHEMA-NAME=SUB3, -
// REALM-NAME=AREA3, -
// PAGES-PER-DML=3
//DECLARE-VARIABLE -
// VARIABLE-NAME=NRCLASHS, -
// TYPE=*INTEGER, -
// INITIAL-VALUE=0
//DECLARE-PROCEDURE -
// PROCEDURE-NAME=TA3, -
// CODE=*SYSDTA
/SEND-DATA -
/ RECORD='READY UPDATE;RELOCATE;'
/SEND-DATA -
/ RECORD='EXIT COND=STATUSCODE EQ 010'
/SEND-DATA -
/ RECORD='ADD NRCLASHS,1,COND=RELOC-FREED-PAGES LT 3;FINISH'
/SEND-DATA -
/ RECORD='EXIT COND=NRCLASHS GE 5'
/SEND-DATA RECORD=*EOF
//REPEAT-PROCEDURE -
// PROCEDURE-NAME=TA3, -
// CYCLE-LIMIT=10000
//SHOW-VARIABLE RELOC-ORIGIN
//SHOW-VARIABLE RELOC-DESTINATION
```

Alternativ können Sie die DMLs der Prozedur TA3 über SYSDTA auch direkt eingeben. Wesentlich ist dabei, dass die Prozedureingabe mit END abgeschlossen wird.

```
//...
//DECLARE-PROCEDURE -
// PROCEDURE-NAME=TA3, -
// CODE=*SYSDTA
READY UPDATE;RELOCATE;
EXIT COND=STATUSCODE EQ 010
ADD NRCLASHS,1,COND=RELOC-FREED-PAGES LT 3;FINISH

EXIT COND=NRCLASHS GE 5;END
```

```
//REPEAT-PROCEDURE -
// PROCEDURE-NAME=TA3, -
// CYCLE-LIMIT=10000
//...
```

#### **Beispiel 4: Vermeidung von Kollisionen orientiert an den gemeldeten Sperrern**

Sie möchten möglichst viel nach vorne verlagern. Irgendwann wird die Datenbank aber abgeschaltet. Dann soll sich die UDS-Online-Utility sofort beenden. Der Realm hat 10000 Seiten. Bei Hinweisen auf Kollisionen (hier: mehr als 10) soll sich die UDS-Online-Utility ebenfalls beenden. Es ist aber nicht ganz klar, wieviele Seiten in einer Transaktion normal verlagert werden. Die aufgetretenen Sperrkonflikte werden als Kriterium herangezogen.

```
//SET-ONLINE-UTILITY-PARAMETERS -
// DBH=*INDEPENDENT, -
// CONFIGURATION-NAME=SESSION4
//SET-RELOCATE-PARAMETERS -
// SUBSCHEMA-NAME=SUB4, -
// REALM-NAME=AREA4, -
// PAGES-PER-DML=2
//DECLARE-VARIABLE -
// VARIABLE-NAME=NRCLASHS, -
// TYPE=*INTEGER, -
// INITIAL-VALUE=0
//DECLARE-PROCEDURE -
// PROCEDURE-NAME=TA4, -
// CODE='READY UPDATE;RELOCATE;-
//EXIT COND=STATUSCODE EQ 010;-
//ADD NRCLASHS,1,COND=RELOC-LOCKED-PAGES GT 0;FINISH;-
//EXIT COND=NRCLASHS GE 10'
//REPEAT-PROCEDURE -
// PROCEDURE-NAME=TA4, -
// CYCLE-LIMIT=10000
//SHOW-VARIABLE RELOC-ORIGIN
//SHOW-VARIABLE RELOC-DESTINATION
```

## Beispiel 5: Vermeidung von Kollisionen mit Warten

Sie möchten möglichst viel nach vorne verlagern. Irgendwann wird die Datenbank aber abgeschaltet. Dann soll sich die UDS-Online-Utility sofort beenden. Der Realm hat 10000 Seiten. Bei Hinweisen auf Kollisionen soll die UDS-Online-Utility 10 Sekunden warten, weil zu erwarten ist, dass die parallel laufenden Anwendungen bald in anderen Bereichen aktiv werden. Die aufgetretenen Sperrkonflikte werden als Kriterium herangezogen. In der Variablen FREED wird nach Prozedurablauf ausgegeben, wieviele Seiten insgesamt freigegeben wurden. Dabei wird die Variable FREED erst nach einem erfolgreichen FINISH versorgt, damit evtl. asynchrone Abbrüche von Verlagerungstransaktionen den Wert nicht verfälschen.

```
//SET-ONLINE-UTILITY-PARAMETERS -
// DBH=*INDEPENDENT, -
// CONFIGURATION-NAME=SESSION5
//SET-RELOCATE-PARAMETERS -
// SUBSCHEMA-NAME=SUB5, -
// REALM-NAME=AREA5, -
// PAGES-PER-DML=2
//DECLARE-VARIABLE -
// VARIABLE-NAME=FREED, -
// TYPE=*INTEGER, -
// INITIAL-VALUE=0
//DECLARE-PROCEDURE -
// PROCEDURE-NAME=TA5, -
// CODE = 'READY UPDATE;RELOCATE;-
// EXIT COND=STATUSCODE EQ 010;-
// ADD FREED,RELOC-FREED-PAGES;-
// FINISH;-
// WAIT 10,COND=RELOC-LOCKED-PAGES GT 0'
//REPEAT-PROCEDURE -
// PROCEDURE-NAME=TA5 -
// CYCLE-LIMIT=10000
//SHOW-VARIABLE RELOC-ORIGIN
//SHOW-VARIABLE FREED
```

## Beispiel 6: Information über Statuscode

Sie möchten möglichst viel nach vorne verlagern. Für den Fall, dass eine Sondersituation entsteht, sollen die Statuscodes differenziert erkennbar sein.

```
//SET-ONLINE-UTILITY-PARAMETERS -
// DBH=*INDEPENDENT, -
// CONFIGURATION-NAME=SESSION6
//SET-RELOCATE-PARAMETERS -
// SUBSCHEMA-NAME=SUB6, -
// REALM-NAME=AREA6, -
// PAGES-PER-DML=1
//DECLARE-VARIABLE -
// VARIABLE-NAME=READY-CODE, -
// TYPE=*STRING,
// INITIAL-VALUE=*STD
//DECLARE-VARIABLE -
// VARIABLE-NAME=RELOC-CODE, -
// TYPE=*STRING,
// INITIAL-VALUE=*STD
//DECLARE-PROCEDURE -
// PROCEDURE-NAME=TA6, -
// CODE='READY UPDATE;
//MOVE READY-CODE,RCODE,COND=STATUSCODE NE 000;
//RELOCATE;-
//MOVE RELOC-CODE,RCODE,COND=STATUSCODE NE 000;
//EXIT COND=STATUSCODE EQ 010;-
//FINISH'
//REPEAT-PROCEDURE -
// PROCEDURE-NAME=TA6, -
// CYCLE-LIMIT=10000
//SHOW-VARIABLE READY-CODE
//SHOW-VARIABLE RELOC-CODE
//SHOW-VARIABLE RELOC-ORIGIN
//SHOW-VARIABLE RELOC-DESTINATION
```

## Beispiel 7: Verkleinerung von Realms

Sie können einen Realm dadurch verkleinern, dass Sie zunächst in einem RELOCATE-Lauf online Sätze und kleine Tabellen im Realm nach vorne verlagern (siehe „[Beispiel 2: Verlagerung aller möglicher Seiten mit der Standardprozedur](#)“ auf Seite 307).

Im Realm hinten liegende DBTT-Abschnitte, CALC-Bereiche bzw. Set-Occurrences, die mehrere Seiten umfassen, können Sie dann offline mit dem Dienstprogramm BREORG reorganisieren.

Abschließend können Sie - ebenfalls mit BREORG - die hinten im Realm freigewordenen Seiten freigeben und somit den Realm verkleinern.

## Beispiel 8: Reorganisieren von PPPs

Sie können die Parameter festlegen, die benötigt werden, um eine REORGPPP-DML mit einer SET-REORGANIZE-PPP-PARAMETERS-Anweisung durchführen zu können. Die PPP-Reorganisation findet als eine Abfolge von Reorganisationsaktionen statt, die in einem oder mehreren aufeinanderfolgenden Online-Utility-Läufen durchgeführt werden.

Wenn die REORGPPP-DML normal endet, stellen die vordefinierten Variablen REORG-PPP-CURRENT, REORG-PPP-PAGES und REORG-PPP-LOCKED Werte zur Verfügung, die zum Steuern des Ablaufs der PPP-Reorganisation verwendet werden können.

```
//SET-ONLINE-UTILITY-PARAMETERS CONFIGURATION-NAME=DBDMLUTI
//SET-REORGANIZE-PPP-PARAMETERS SUBSCHEMA-NAME = SUB, -
//                                REALM-NAME= AREA-3 ,-
//                                INITIALIZE=ANY, PAGES-PER-DML=1000
//DECLARE-VARIABLE VARIABLE-NAME=READY-CODE, -
//                                TYPE=*STRING, INITIAL-VALUE=*STD
//DECLARE-VARIABLE VARIABLE-NAME=REOPPP-CODE, -
//                                TYPE=*STRING, INITIAL-VALUE=*STD
//DECLARE-PROCEDURE PROCEDURE-NAME=TA1, -
// CODE='READY UPDATE; -
//MOVE READY-CODE,RCODE,COND=STATUSCODE NE 000; REORGPPP;-
//MOVE REOPPP-CODE,RCODE,COND=STATUSCODE NE 000;-
//EXIT COND=STATUSCODE EQ 010; FINISH'
//REPEAT-PROCEDURE PROCEDURE-NAME=TA1, CYCLE-LIMIT=1
//SHOW-VARIABLE READY-CODE
//SHOW-VARIABLE REOPPP-CODE
//SHOW-VARIABLE REORG-PPP-CURRENT
//SHOW-VARIABLE REORG-PPP-PAGES
//SHOW-VARIABLE REORG-PPP-LOCKED
```



---

## 9 Datenbank reorganisieren mit BREORG

Reorganisieren ist eine wichtige Arbeit zur Wartung der Datenbank. Wichtig deshalb, weil sie Ihnen unter anderem hilft, sowohl Speicherplatz als auch Zeit zu sparen.

Speicherplatz sparen können Sie durch Verkleinern von Realms, Verkleinern der Database Key Translation Table (DBTT) einer Satzart und damit Verringern der maximal zulässigen Anzahl von Sätzen und Neuaufbauen der Set-Tabellen mit einem höheren Füllgrad.

Zeit einsparen können Sie auch bei den Zugriffen auf die Datenbank durch Abbauen von Überlaufseiten bei Hashbereichen, Aktualisieren von physischen Adressverweisen, die als Probable Position Pointer (PPP) angelegt sind, und Neuaufbauen der Set-Tabellen mit einem geänderten Füllgrad.

Reorganisieren kann aber auch notwendig sein, wenn ein Realm zu klein geworden ist oder von einer Satzart mehr Sätze gespeichert werden sollen, als ursprünglich geplant war.

Regelmäßig sollten Sie dann reorganisieren, wenn Sie in Ihrer Datenbank Sort-Key-Tabellen, Adresslisten, Ketten oder Listen verwenden in Verbindung mit LOCATION MODE IS CALC bei Satzarten, die häufig geändert werden. Änderungen können in solchen Fällen dazu führen, dass die Probable Position Pointer (PPP) in den Tabellen nicht mehr stimmen. Dadurch verschlechtert sich die Zugriffszeit auf solche Sätze bei Verwendung der eingerichteten Tabellen (siehe Handbuch „[Datenbankbetrieb](#)“).

## 9.1 Funktionen

Das Dienstprogramm BREORG bietet folgende Funktionen:

- Puffergröße festlegen
- Realm-Größe ändern
- Satzmengengerüst ändern
- CALC-Bereiche reorganisieren
- Tabellen und Set-Konstrukte reorganisieren
- Probable Position Pointer (PPP) reorganisieren

Die Funktionen von BREORG können Sie nicht nur auf die Benutzerrealms der Datenbank anwenden, sondern auch auf die PRIVACY-AND-IQF-Datenbank im DBDIR und die Compilerdatenbank im DBCOM.

BREORG erweitert bei Bedarf automatisch die Realms der bearbeiteten Datenbank. Näheres hierzu siehe Handbuch „[Datenbankbetrieb](#)“, Automatische Realm-Erweiterung).

BREORG berücksichtigt beim Start ggf. eine zugewiesene UDS/SQL-Pubset-Deklaration (siehe Handbuch „[Datenbankbetrieb](#)“, Pubset-Deklarations-Jobvariable). Eine fehlerhafte Zuweisung führt zum Programmabbruch.

## 9.2 Systemumgebung

BREORG benötigt zum Ablaufen Informationen über die Bereiche, Sets, Satzarten und Tabellen, die zu reorganisieren sind. Diese Informationen bezieht er aus der Schema Information Area (SIA) des zugehörigen Schemas.

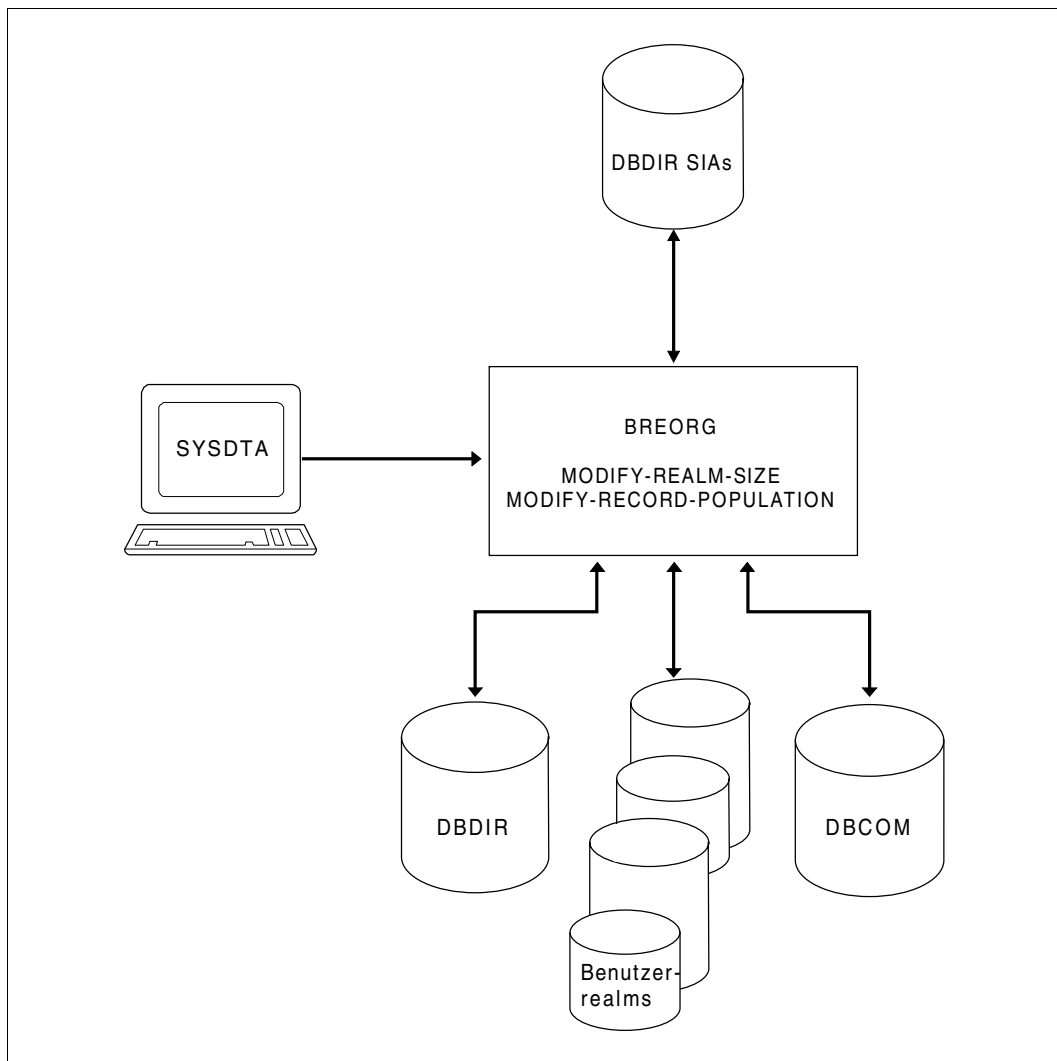


Bild 17: Systemumgebung bei den Funktionen **MODIFY-REALM-SIZE** und **MODIFY-RECORD-POPULATION**

Für die REORGANIZE-Funktionen benötigt BREORG zusätzlich noch Subschema-Informationen (nur beim Neuaufbau von mehrstufigen LIST-Sets), die er der Subschema Information Area (SSIA) entnimmt.

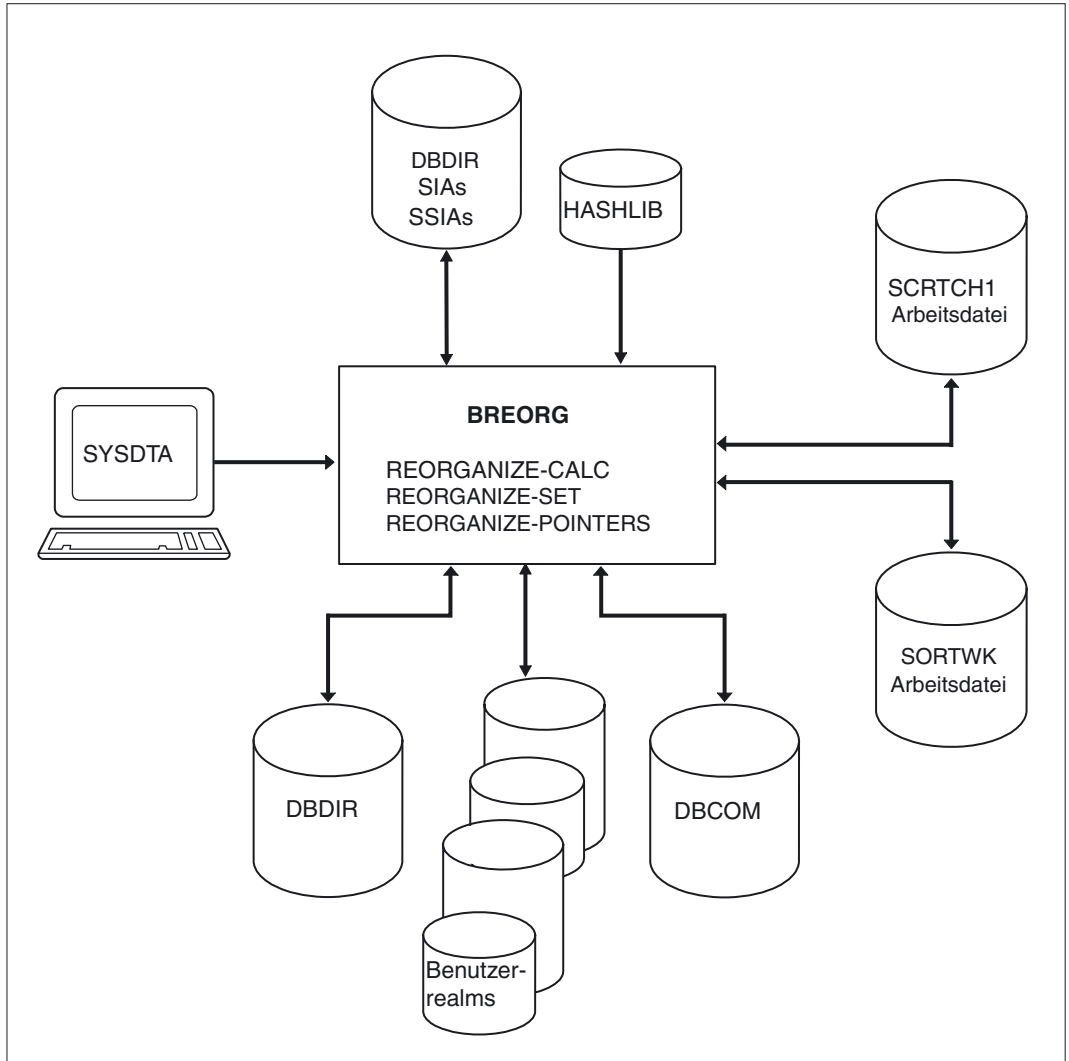


Bild 18: Systemumgebung für die Funktionen REORGANIZE-CALC, REORGANIZE-SET und REORGANIZE-POINTERS

## Arbeitsdateien

BREORG benötigt für die REORGANIZE-Anweisungen auf Platte verschiedene Arbeitsdateien, die er automatisch unter der entsprechenden Benutzerkennung auf gemeinschaftlicher Platte anlegt und bei normalem Lauf wieder löscht.

### Arbeitsdateien für die REORGANIZE-CALC- und REORGANIZE-SET-Anweisungen

Bei den REORGANIZE-CALC- und REORGANIZE-SET-Anweisungen handelt es sich um zwei Dateien, die standardmäßig die Dateikettungsnamen SCRTCH1 und SORTWK haben:

#### SCRTCH1

benötigt BREORG nur zum Reorganisieren direkter und indirekter Hashbereiche bzw. mehrstufiger Tabellen;  
enthält eine Zwischenversion der zu verarbeitenden CALC-Einträge bzw. Tabellenzeilen

#### SORTWK

benötigt der von BREORG benutzte SORT für die Sortierung interner Auswertungssätze (siehe auch Handbuch „[SORT \(BS2000\)](#)“).

Wollen Sie die beiden Arbeitsdateien explizit einrichten, so müssen diese die folgenden Eigenschaften besitzen:

#### *Arbeitsdatei-1*

Dateikettungsname SCRTCH1

Zugriffsmethode=PAM

Das Mengengerüst der zwischenzuspeichernden Daten ergibt sich

- bei der Reorganisation von indirekten Hashbereichen durch

$(12 + \textit{schlüssel länge}) * \textit{anzahl-einträge}$  Bytes

- bei der Reorganisation von direkten Hashbereichen durch

$8 * \textit{anzahl-einträge}$  Bytes

- bei Reorganisation mehrstufiger Tabellen durch

$12 * \textit{anzahl-einträge}$  Bytes

*schlüssellänge*

Länge des CALC-Keys

*anzahl-einträge*

Anzahl der CALC-Index-Einträge bzw. belegten Tabellenzeilen

*Arbeitsdatei-2*

Die Arbeitsdatei-2 wird vom SORT benötigt, wenn der virtuelle Speicher für die Vorsortierung nicht ausreicht. Die Primärzuweisung sollte sich am Mengengerüst der zu sortierenden Daten orientieren unter Berücksichtigung des von SORT empfohlenen Sicherheitsfaktors (siehe die Beschreibung der Arbeitsdateien im Handbuch „[SORT \(BS2000\)](#)“). Es sollte immer eine angemessene Sekundärzuweisung erfolgen, für den Fall, dass der Speicherplatz erweitert werden muss.

Dateikettungsname SORTWK

Zugriffsmethode=PAM

Das Mengengerüst der zu sortierenden Daten ergibt sich

- bei der Reorganisation von indirekten Hashbereichen durch  
 $(12 + \textit{schlüssellänge}) * \textit{anzahl-einträge}$  Bytes
- bei der Reorganisation von direkten Hashbereichen durch  
 $(\textit{satzlänge} + \textit{schlüssellänge} + 7) * \textit{anzahl-einträge}$  Bytes
- bei Reorganisation mehrstufiger Tabellen durch  
 $12 * \textit{anzahl-einträge}$  Bytes

*schlüssellänge*

Länge des CALC-Keys

*anzahl-einträge*

Anzahl der CALC-Index-Einträge bzw. belegten Tabellenzeilen

*satzlänge*

Länge der Sätze bzw. TabellenzeilenSekundärzuweisung, falls der Speicherplatz erweitert werden muss; *sekundär* sollte ungleich Null und nicht kleiner als 120 sein.

Richten Sie die beiden Arbeitsdateien nicht selbst ein, so richtet BREORG sie mit folgenden Namen und Größen ein:

UTI . *tsn* . SCRTCH1      (360,360) bei REORGANIZE-SET und REORGANIZE-CALC  
 UTI . *tsn* . SORTWK      (120,120)

*tsn*      ist die Prozessfolgennummer des aktuellen Prozesses.

### Arbeitsdateien für die REORGANIZE-POINTERS-Anweisung

Bei der REORGANIZE-POINTERS-Anweisung verwendet BREORG Arbeitsdateien für die Satzarten und eine weitere Arbeitsdatei zum Sortieren.

Die Arbeitsdateien für die Satzarten können Sie auch selbst über die Dateinamen, die Arbeitsdatei für das Sortieren über den Dateikettungsamen anlegen.

#### *Arbeitsdateien für die Satzarten*

- Dateinamen UTI.BREORG.*dbname*.*xxx*.*yyyyy*

<i>dbname</i>	Name der Datenbank
<i>xxx</i>	Realm-Nummer des angegebenen Realms
<i>yyyyy</i>	Nummer der Satzart, deren Probable Position Pointer (PPP) im Realm aktualisiert werden; yyyyy=0 wird gegebenenfalls für Probable Position Pointer in SYSTEM-Sets genutzt.

Zugriffsmethode=SAM

Das Mengengerüst der zwischenzuspeichernden Daten ergibt sich aus der Formel

$$\text{anzahl-ppps} * 11 \text{ Bytes}$$

Richten Sie die Dateien nicht selbst ein, so orientiert sich BREORG an der Größe der DBTT der Satzart *yyyyy* sowie der Größe des Realms *xxx*.

Die Mindestgröße geht von 5000 zwischenzuspeichernden Objekten aus.

- Dateinamen UTI.BREORG.*dbname*.*xxx*.00001

*dbname* Name der Datenbank

*xxx* Realm-Nummer des angegebenen Realms

Zugriffsmethode=SAM

Im User-Schema gibt es keine Satzart mit der Satzartnummer 1. In der Arbeitsdatei mit der Satzartnummer 1 werden alle aktualisierten Probable Position Pointer (PPP) sortiert nach ihrer Lage im

Realm abgelegt. Die Größe dieser Arbeitsdatei orientiert sich daher an der Summe der Größen der benötigten Einzeldateien UTI.BREORG.*dbname*.*xxx*.*yyyyy* (*yyyyy*=0 bzw. *yyyyy*>1).

Nach der PPP-Aktualisierung werden die Arbeitsdateien gelöscht.

#### *Arbeitsdatei zum Sortieren*

Diese Arbeitsdatei wird vom SORT benötigt, wenn der virtuelle Speicher für die Vorsortierung nicht ausreicht. Die Primärzuweisung sollte sich am Mengengerüst der zu sortierenden Daten orientieren unter Berücksichtigung des von SORT empfohlenen Sicherheitsfaktors (siehe die Beschreibung der Arbeitsdateien im Handbuch „[SORT \(BS2000\)](#)“). Es sollte immer eine angemessene Sekundärzuweisung erfolgen für den Fall, dass der Speicherplatz erweitert werden muss.

Dateikettungsname SRT1WK

Zugriffsmethode=PAM

Die benötigte Dateigröße entspricht dem Maximum der Größe aller Dateien UTI.BREORG.*dbname*.*xxx*.00001.

Wurde die Arbeitsdatei für das Sortieren explizit eingerichtet, muss sie bei Bedarf auch explizit gelöscht werden.



## 9.3 Datenbanksicherung

Haben Sie ALOG-Dateien eingerichtet und AFIM-Logging eingeschaltet, so schreibt BREORG bei allen Funktionen After-Images.

Fehlen zu Beginn von BREORG trotz eingeschaltetem AFIM-Logging ALOG-Dateien, beendet sich BREORG mit einer Fehlermeldung, bevor mit dem Bearbeiten begonnen wurde.

Bei einem Fehler auf der ALOG-Datei während des Ablaufs von BREORG wird das weitere Logging unterdrückt und BREORG nach Ausführung der aktuellen Anweisung beendet. Dabei entsteht im allgemeinen Fall eine Logging-Lücke.

### Konsolenmeldung über AFIM-Logging

BREORG wertet den Auftragsschalter 29 aus. Ist dieser Auftragsschalter gesetzt, so wird beim zwangsweisen Abschalten des AFIM-Loggings die Konsolenmeldung LOGGING STOPPED FOR DATABASE *dbname* ausgegeben.

## 9.4 Anweisungen für BREORG

### Regeln der Anweisungen

Falsch angegebene Anweisungen können Sie im Dialog über SDF korrigieren.

Erfolgt die Eingabe nicht über die SDF-Maske, so ist nicht zu jedem Zeitpunkt jede Anweisung erlaubt und wird deswegen abgewiesen (z.B. OPEN-DATABASE).

Erst nach Angabe der Anweisung END werden die folgenden Anweisungen in der von Ihnen vorgegebenen Reihenfolge durchgeführt:

1. MODIFY-REALM-SIZE
2. MODIFY-RECORD-POPULATION
3. REORGANIZE-CALC
4. REORGANIZE-POINTERS
5. REORGANIZE-SET

Im Batch-Betrieb werden bei fehlerhaften Anweisungen alle Anweisungen bis zur ersten fehlerhaften durchgeführt.

Anweisung	Bedeutung
ALLOCATE-BUFFERPOOL	Puffergröße festlegen (in Mbyte)
END	Eingabe der Anweisungen beenden
MODIFY-REALM-SIZE	Realm-Größe ändern
MODIFY-RECORD-POPULATION	Satzmengengerüst ändern
OPEN-DATABASE	Datenbank eröffnen
REORGANIZE-CALC	CALC-Bereiche reorganisieren
REORGANIZE-POINTERS	Alle Probable Position Pointer (PPP) eines Realms reorganisieren
REORGANIZE-SET	Tabellen und Set-Konstrukte reorganisieren
SPECIFY-SCHEMA	Schema angeben
SPECIFY-SUBSCHEMA	Subschema angeben
UNDO	Anweisung rückgängig machen

Tabelle 21: Anweisungen für BREORG

Die Anweisungen von BREORG sind im Folgenden in alphabetischer Reihenfolge beschrieben.

## Puffergröße festlegen (ALLOCATE-BUFFERPOOL)

Mit der Anweisung ALLOCATE-BUFFERPOOL legen Sie die Größe des verwendeten Buffer Pools in Mbyte fest.

Diese Anweisung muss die erste Anweisung sein, sofern für die Pufferinitialisierung nicht die Standardwerte genommen werden sollen.

Die Anweisung ALLOCATE-BUFFERPOOL wird anschließend nicht mehr in der SDF-Maske angeboten.

Die Anweisung kann durch die Anweisung UNDO nicht zurückgenommen werden.

ALLOCATE-BUFFERPOOL
BUFFER-SIZE = * <u>STD</u> / <integer 1..2000>

### **BUFFER-SIZE = \*STD**

Die Standard-Größe des Buffer Pools wird anlagenabhängig mit 1 oder 2 Mbyte festgelegt.

### **BUFFER-SIZE = <integer 1..2000>**

Die Größe des Buffer Pools muss in den angegebenen Grenzen liegen. Der Maximalwert ist abhängig von der Version des Betriebssystems und der Anlage.

## Eingabe der Anweisungen beenden (END)

Mit der Anweisung END beenden Sie die Eingabe der Anweisungen. Die Ausführung wird gestartet.

Die Anweisung END kann durch die Anweisung UNDO nicht zurückgenommen werden.

END

Diese Anweisung hat keine Operanden.

## Realm-Größe ändern (MODIFY-REALM-SIZE)

Mit der Anweisung MODIFY-REALM-SIZE ändern Sie die Größe eines Realms der Datenbank.

```
MODIFY-REALM-SIZE
```

```
REALM-NAME = <realmname>  
,REALM-SIZE = <integer 1..16777216> / *RELATIVE(...) / *MINIMUM  
  *RELATIVE(...)  
    | DIFFERENCE = <integer -16777216..16777216>
```

### **REALM-NAME = <realmname>**

Name des Realm, der geändert werden soll.

### **REALM-SIZE = <integer 1..16777216>**

Die neue Größe entspricht dem angegebenen Wert in Datenbankseiten.

### **REALM-SIZE = \*RELATIVE (...)**

Die neue Größe ergibt sich aus der alten Größe und der angegebenen Differenz (positiver oder negativer Wert möglich). Die Größe, die bei MINIMUM erzielt wird, kann jedoch nicht unterschritten werden.

### **DIFFERENCE = <integer -16777216..16777216>**

Differenz zur alten Realm-Größe in Datenbankseiten.

### **REALM-SIZE = \*MINIMUM**

Der Realm wird um die leeren Seiten am Ende reduziert.

## Vergrößern eines Realm

Die physische Erweiterung der Datei *realmname* fordert BREORG beim DVS des BS2000 an. Die hinzugekommenen Seiten werden durch die Freiplatzverwaltung des Realms erfasst. Reicht die vorhandene Freiplatzverwaltung des Realms (Free Place Administration, FPA) für die neue Realm-Größe nicht aus, werden je nach Bedarf neue Freiplatzverwaltungs-Tabellen (FPA-Extents) angelegt.

Bei der Erweiterung des DBDIR oder des DBCOM werden die neuen leeren Seiten immer formatiert. Ob bei der Erweiterung von Benutzerrealms die neuen leeren Seiten formatiert werden, hängt vom Zustand der Probable Position Pointer (PPP) ab:

- Sind in allen Benutzerrealms mit der Anweisung REORGANIZE-POINTERS die Probable Position Pointer (PPP) aktualisiert worden, wird auf die Formatierung der neuen leeren Seiten verzichtet.
- Kann ein PPP-Verweis in die neuen Seiten nicht ausgeschlossen werden, werden die neuen leeren Seiten formatiert.



Wenn Sie private Platten verwenden, müssen Sie darauf achten, dass auf der Platte ausreichend Platz zum Erweitern des Realm vorhanden ist.

Wenn die Platte, auf der der Realm endet, für eine Erweiterung nicht ausreicht, müssen Sie eine weitere Platte für diese Datei vor dem BREORG-Lauf zuweisen.

## Verkleinern eines Realm

BREORG verkleinert den Realm auf folgende Weise: Zunächst wird aus der Anweisung MODIFY-REALM-SIZE und den im Realm vorliegenden Daten die erzielbare Realm-Größe bestimmt. Nicht mehr benötigte FPA-Teile werden freigegeben. Weiterhin benötigte FPA-Teile werden nach Möglichkeit in der Regel an den Realm-Anfang verlagert. Das physische Verkleinern fordert dann BREORG über das DVS des BS2000 an, d.h. ein Kommando MODIFY-FILE-ATTRIBUTES zum Verkleinern der Realm-Datei ist nicht erforderlich.

Eine Reorganisation belegter Seiten des Realms ist mit dieser Anweisung nicht verbunden, es werden nur freie Seiten von Ende des Realms her entfernt und die Freiplatzverwaltung angepasst.



Der Realm DBDIR darf nur geändert werden unter dem PRIVACY-AND-IQF-SCHEMA.

Der Name DBDIR wird intern umgesetzt in den Realm-Namen DATABASE-DIRECTORY, DBCOM in den Realm-Namen DATABASE-COMPILER-REALM.

## Ablaufmeldungen

Nach dem Ausführen der Anweisung MODIFY-REALM-SIZE wird das Ergebnis für die neue Freiplatzverwaltung (FPA) eines Realm ausgegeben.

```
***** RESULTS OF FPA-REORGANIZATION OF AREA area name
NEW FPA FIRST PAGE      : area ref - seiten nr
NEW FPA LAST PAGE      : area ref - seiten nr
NEW NR OF EXTENTS      : number extents
NEW FPA SIZE            : number pages
NEW NR OF PAGES        : number pages

NR OF DATABASE ACCESSES : number physical io
```

*area name*

Name des erweiterten Realm bzw. reduzierten Realm

*area ref - seiten nr*

bei FPA FIRST PAGE: kleinster Act-Key unter allen Act-Keys von FPA-Seiten (nicht unbedingt identisch mit dem FPA-Beginn)

bei FPA LAST PAGE: größter Act-Key unter allen Act-Keys von FPA-Seiten (nicht unbedingt identisch mit dem FPA-Ende)

*number extents*

Anzahl der FPA-Extents

*number pages*

bei NEW FPA SIZE: Seitenanzahl des neuen FPA-Bereiches

bei NEW NR OF PAGES: neue Seitenanzahl des Realm

*number physical io*

Anzahl der Ein- und Ausgaben

## Satzmengengerüst ändern (MODIFY-RECORD-POPULATION)

Mit der Anweisung MODIFY-RECORD-POPULATION ändern Sie das maximale Mengengerüst (DBTT) einer Satzart.

```
MODIFY-RECORD-POPULATION
```

```
RECORD-NAME = <recordname>
```

```
,RECORD-POPULATION = <integer 1..2147483647> / *RELATIVE(...) / *MINIMUM
```

```
  *RELATIVE(...)
```

```
    | DIFFERENCE = <integer -2147483647..2147483647>
```

### **RECORD-NAME = <recordname>**

Name der Satzart, deren Mengengerüst geändert werden soll.

### **RECORD-POPULATION = <integer 1..2147483647>**

Die Anzahl der DBTT-Einträge für die Satzart entspricht dem angegebenen Wert.

### **RECORD-POPULATION = \*RELATIVE (...)**

Die neue Größe der DBTT ergibt sich aus der alten Größe und der angegebenen Differenz (positiver oder negativer Wert möglich). Die Größe, die bei MINIMUM erzielt wird, kann jedoch nicht unterschritten werden.

### **DIFFERENCE = <integer -2147483647..2147483647>**

Die Differenz zur alten DBTT-Größe in Anzahl von DBTT-Einträgen.

### **RECORD-POPULATION = \*MINIMUM**

Die DBTT der Satzart wird auf den kleinstmöglichen Wert reduziert.



Angaben, die bei einer 2-Kbyte-Datenbank unzulässig sind, können nicht über SDF abgefangen werden, wenn sie in den oben angegebenen Grenzen liegen.

Bei 2-Kbyte-Datenbanken erfolgen zusätzliche Überprüfungen, sodass für diese Datenbanken die alten Grenzen gelten.

Die DBTT belegt die für sie reservierten Seiten immer vollständig. So kann es vorkommen, dass sich nach dem BREORG-Lauf bei einer Vergrößerung mehr Sätze speichern lassen, als in der Anweisung verlangt ist. Für den anderen Fall, dass Sie mit der Anweisung weniger DBTT-Einträge freigeben, als eine DBTT-Seite der angegebenen Satzart enthält, so ändert sich die maximal zulässige Anzahl von Sätzen dieser Satzart nicht.





Für die DBTT werden immer ganze Seiten belegt. Die Anzahl der gewünschten Einträge wird auf volle Seiten aufgerundet.

Sollte dabei jedoch die maximale RSQ überschritten werden, wird abgerundet auf volle DBTT-Seiten.

Der SSIA-RECORD darf nur modifiziert werden unter dem PRIVACY-AND-IQF-SCHEMA.

Bei der Reorganisation einer Datenbank können unabhängig von der Aktivierung der Online-DBTT-Erweiterung DBTT-Extents entstehen. Eine Vergrößerung der DBTT durch BREORG wird im DBCOM und DBDIR durch Vergrößerung der schon vorhandenen DBTT realisiert. Im Benutzerrealm wird eine Vergrößerung der DBTT durch BREORG über DBTT-Extents realisiert, wenn die Ziel-DBTT insgesamt größer als 128 PAM-Seiten ist. Ist die Ziel-DBTT kleiner oder gleich 128 PAM-Seiten, wird sie durch BREORG immer als DBTT-Basis, also am Stück realisiert. Über das Ergebnis von Vergrößerungen oder Verkleinerungen von DBTTs werden Sie mit entsprechenden Meldungen informiert.



Auch der letzte DBTT-Extent ist immer vollständig vorhanden. Die Anzahl der gewünschten Einträge wird nicht nur auf volle Seiten aufgerundet, sondern auf volle DBTT-Extents.

## Ablaufmeldungen

Nach dem Ausführen der Anweisung MODIFY-RECORD-POPULATION wird das Ergebnis der DBTT-Reorganisation der Satzart ausgegeben.

```
***** BEGIN OF DBTT-SIZE-MODIFICATION AT hh:mm:ss
***** RESULTS OF DBTT-REORGANIZATION OF RECORD record name
      NEW DBTT FIRST PAGE      : area ref - seiten nr
      NEW DBTT LAST PAGE      : area ref - seiten nr
      NEW NR OF EXTENTS       : number extents
      NEW DBTT SIZE           : number pages
      NEW NR OF DBTT ENTRIES  : number entries
***** END OF DBTT-SIZE-MODIFICATION AT hh:mm:ss
```

*record name*

Name der geänderten Satzart

*area ref - seiten nr*

Act-Key der ersten bzw. der letzten DBTT-Seite

*number extents*

neue Anzahl der DBTT-Extents

*number pages*

neue Anzahl der DBTT-Seiten

*number entries*

neue Anzahl der Sätze

## Datenbank eröffnen (OPEN-DATABASE)

Mit der Anweisung OPEN-DATABASE legen Sie die Datenbank fest, die mit den nachfolgenden Anweisungen von BREORG bearbeitet werden soll.

Die Anweisung OPEN-DATABASE wird nicht angeboten, wenn die Datenbank über LINK=DATABASE zugewiesen ist.

OPEN-DATABASE

DATABASE-NAME = <dbname>

,SCHEMA-NAME = \*STD / <schemaname>

,USER-IDENTIFICATION = \*OWN / <userid>

### **DATABASE-NAME = <dbname>**

Name der Datenbank. Sie können nur eine Datenbank bearbeiten, die in Ihrer eigenen Kennung katalogisiert ist. Eine Datenbank aus einer fremden Kennung kann nur von der TSOS-Kennung des Systemverwalters bearbeitet werden.

### **SCHEMA-NAME = \*STD**

Es wird für die Bearbeitung der Datenbank der Name des Benutzerschemas genommen, das für die Datenbank definiert wurde.

### **SCHEMA-NAME = <schemaname>**

Mit *schemaname* geben Sie das Datenbankschema an, für dessen Objekte die BREORG-Anweisungen auszuführen sind.

Mögliche Werte:

PRIVACY-AND-IQF-SCHEMA

COMPILER-SCHEMA

Name des Benutzerschemas

### **USER-IDENTIFICATION = \*OWN**

Die Datenbank liegt in der eigenen Kennung.

### **USER-IDENTIFICATION = <userid>**

Die Angabe einer fremden Kennung ist nur unter der TSOS-Kennung erlaubt.

Zum Bearbeiten einer Datenbank benötigt BREORG Informationen über Realms, Satzarten und Setbeziehungen in der Datenbank. Der Name des Schemas ermöglicht BREORG den Zugriff auf die SIA, die diese Informationen enthält und die auch entsprechend abgeändert wird.

## CALC-Bereiche reorganisieren (REORGANIZE-CALC)

Mit der Anweisung REORGANIZE-CALC reorganisieren Sie CALC-Bereiche, die zu einer Satzart gehören. Das sind:

- durch LOCATION MODE CALC angelegte Bereiche
- SEARCH KEY USING CALC auf Satzartebene
- SEARCH KEY USING CALC in singulären Sets, in denen die Satzart Member ist

REORGANIZE-CALC

```

RECORD-NAME = <recordname>
,CALC-RECORD = NONE / list-poss(6): *WITHIN-POPULATION(...)
  *WITHIN-POPULATION(...)
    | REALM = *ALL / <realmname>
    | ,POPULATION = *UNCHANGED / <integer 1..2147483647>
,CALC-SEARCHKEY = NONE / list-poss(30): *KEY-POPULATION(...)
  *KEY-POPULATION(...)
    | KEY-REF = *ALL / <integer 1..65535>
    | ,POPULATION = *STD / *UNCHANGED / <integer 1..2147483647>

```

### RECORD-NAME = <recordname>

Name der Satzart, deren CALC-Bereiche reorganisiert werden sollen.

### CALC-RECORD = NONE

LOCATION MODE CALC-Bereiche werden nicht reorganisiert.

### CALC-RECORD = list-poss(6): \*WITHIN-POPULATION (...)

Die in dem angegebenen Realm bzw. in den angegebenen Realms liegenden LOCATION MODE CALC-Bereiche werden reorganisiert.

#### REALM = \*ALL

Alle CALC-Bereiche werden reorganisiert.

#### REALM = <realmname>

Nur der CALC-Bereich, der im angegebenen Realm liegt, wird reorganisiert.

#### POPULATION = \*UNCHANGED

Es wird nur eine PPP-Aktualisierung durchgeführt.

Eine PPP-Aktualisierung ist sinnvoll für indirekte LOCATION MODE CALC-Bereiche.

**POPULATION = <integer 1..2147483647>**

Der betroffene CALC-Bereich wird neu aufgebaut. Die Anzahl der angegebenen Einträge wird umgerechnet in Anzahl von Seiten. Anschließend wird die Seitenzahl auf die nächste größere Primzahl (= Größe des Hashbereiches in Datenbankseiten) gerundet.

In indirekten CALC-Bereichen werden zusätzlich die Probable Position Pointer (PPP) aktualisiert.

**CALC-SEARCHKEY = NONE**

Die SEARCH KEY USING CALC-Bereiche, die zu der Satzart gehören, werden nicht reorganisiert.

**CALC-SEARCHKEY = list-poss(30): \*KEY-POPULATION (...)**

Die CALC-Bereiche der angegebenen KEYS werden reorganisiert.

**KEY-REF = \*ALL**

Alle CALC SEARCH KEY-Bereiche werden reorganisiert.

**KEY-REF = <integer 1..65535>**

Nur der CALC SEARCH KEY-Bereich mit der angegebenen KEY REF wird reorganisiert. Die KEY REF kann aus dem BPSIA-Protokoll entnommen werden.

**POPULATION = \*STD**

Der betroffene CALC-Bereich wird neu aufgebaut. Falls ein LOCATION MODE CALC-Bereich vorhanden ist, wird die neue Größe des SEARCH KEY USING CALC-Bereiches aus der Größe des LOCATION MODE CALC-Bereiches ermittelt bzw. bei verteilten CALC-Bereichen aus der Summe der Größen der LOCATION MODE CALC-Bereiche berechnet. Gibt es keinen LOCATION MODE CALC-Bereich, orientiert sich die Größe an der DBTT-Größe der Satzart.

**POPULATION = \*UNCHANGED**

Es wird nur eine PPP-Aktualisierung durchgeführt.

**POPULATION = <integer 1..2147483647>**

Der betroffene CALC-Bereich wird in der angegebenen Größe neu aufgebaut. Die errechnete Seitenzahl wird auf die nächste größere Primzahl gerundet. Zusätzlich werden Probable Position Pointer (PPP) aktualisiert.



Wird in einer Liste derselbe REALM oder dieselbe KEY REF mehrfach angegeben, gilt die letzte Angabe.

Wird in einer Liste \*ALL für REALM oder KEY REF angegeben, gilt die Angabe \*ALL.

### LOCATION MODE CALC-Bereiche reorganisieren

Sätze einer Satzart, die mit LOCATION MODE IS CALC definiert ist, werden in der Regel in der Datenbank in einem Hashbereich gespeichert. Ihre Adresse in diesem Hashbereich errechnet der Database Handler aus dem jeweiligen CALC-Key und der Größe des Bereichs.

Im „indirekten“ Hashbereich werden nur Tabellen-Einträge bestehend aus CALC-Key, RSQ und Probable Position Pointer (PPP) gespeichert. Solche indirekten Hashbereiche werden bei einer LOCATION MODE CALC-Angabe erzeugt, wenn die Satzart Member in einem mit MODE IS LIST spezifizierten Set ist oder in der SSL für die Satzart PLACEMENT OPTIMIZATION definiert wurde oder in der SSL für die Satzart COMPRESSION FOR ALL ITEMS spezifiziert wurde.

Zum Reorganisieren berechnet BREORG aus der POPULATION-Angabe die Größe des neuen Hashbereichs und reserviert die errechnete Anzahl von Seiten. Dann errechnet er für jeden Satz bzw. Tabellen-Eintrag die Adresse im neu zugewiesenen Hashbereich und überträgt ihn dorthin. Die Seiten des alten Hashbereiches sind nach dem Reorganisieren wieder für andere Zwecke verfügbar.

Dadurch lässt sich die Verteilung der Einträge im Hashbereich ändern und das Anlegen von Überlaufseiten vermeiden. Eine Übersicht über Anzahl und Füllgrad der Primärseiten und der Überlaufseiten des neu angelegten Hashbereichs sollten Sie sich nach dem Reorganisieren von BSTATUS ausgeben lassen!

Die Anzahl der Einträge in der DBTT der betroffenen Satzart ändert BREORG nicht, trägt jedoch bei direktem Hashbereich in Spalte 0 der DBTT, die die physische Adresse der Sätze enthält, die neue physische Adresse des jeweiligen Satzes ein.

**CALC SEARCH KEY-Bereiche reorganisieren:**

Ein CALC SEARCH KEY-Bereich enthält nicht die Datensätze selbst, sondern Tabellen-Einträge, die pro Eintrag den CALC-Key, die RSQ (Record Sequence Number) und den Probable Position Pointer (PPP) des jeweiligen Datensatzes enthalten.

Beim Reorganisieren von CALC SEARCH KEYS sind drei Fälle zu unterscheiden:

1. Sie haben POPULATION = STD angegeben:

BREORG bestimmt die neue Größe des CALC SEARCH KEY-Bereiches selbst. Ist für die Satzart LOCATION MODE IS CALC spezifiziert worden, so wird diese Größe - bei verteilter Satzart die Summe aller Bereiche - genommen, um die Population zu ermitteln. Im anderen Fall wird die Größe der DBTT (Anzahl Einträge) als POPULATION-Angabe genutzt.

2. Sie haben POPULATION = UNCHANGED angegeben:

BREORG aktualisiert die Probable Position Pointer (PPP) der Tabellen-Einträge. Die Verteilung der Tabellen-Einträge im Primärbereich und in den Überlaufseiten bleibt dieselbe.

Ein Anpassen der Probable Position Pointer kann z.B. notwendig werden, wenn sich die Positionen der Sätze als Member eines LIST-Sets während der Verarbeitung in der Datenbank verändert haben.

3. Sie haben bei POPULATION = INTEGER ... einen neuen Wert angegeben:

BREORG berechnet aus der POPULATION-Angabe die Größe des neuen Hashbereichs und reserviert eine entsprechende Folge zusammenliegender Seiten. Anschließend überträgt er die Tabellen-Einträge in die neu zugewiesenen Seiten und aktualisiert gleichzeitig ihre Probable Position Pointer (PPP). Da BREORG die Tabellen-Einträge jeweils neu anlegt, bewirkt das Übertragen eine Neuverteilung auf Primärbereich und Überlaufseiten.

Die ursprünglich reservierten CALC-Seiten gibt BREORG während des Reorganisierens wieder frei.

Da die Datensätze selbst beim Reorganisieren der CALC SEARCH KEY-Bereiche nicht umgespeichert werden, ändert sich auch die Information in der entsprechenden DBTT nicht.

### Größe eines neuen Hashbereichs bestimmen

Die Anzahl der CALC-Seiten, die BREORG auf Grund einer POPULATION-Angabe neu zuweist, können Sie mit folgenden Formeln berechnen:

- für einen indirekten Hashbereich

- bei 2 Kbyte

$$\frac{2018}{(\text{calc-key-länge} + 7)} = \text{einträge-pro-seite}^1$$

- bei 4 Kbyte

$$\frac{3970}{(\text{calc-key-länge} + 10)} = \text{einträge-pro-seite}^1$$

- bei 8 Kbyte

$$\frac{8066}{(\text{calc-key-länge} + 10)} = \text{einträge-pro-seite}^1$$

und

$$\frac{\text{ganzzahl} - 1}{\text{einträge-pro-seite}} + 1 = \text{seitenanzahl}^2$$

- für einen direkten Hashbereich

- bei 2 Kbyte

$$\frac{2018}{(\text{satzlänge} + \text{calc-key-länge} + 15)} = \text{einträge-pro-seite}^1$$

- bei 4 Kbyte

$$\frac{3970}{(\text{satzlänge} + \text{calc-key-länge} + 22)} = \text{einträge-pro-seite}^1$$

<sup>1</sup> Ergebnis ist abzurunden

<sup>2</sup> Ergebnis ist aufzurunden auf die nächsthöhere Primzahl, wenn sich keine Primzahl ergibt



- bei 8 Kbyte

$$\frac{8066}{(\text{satzlänge} + \text{calc-key-länge} + 22)} = \text{einträge-pro-seite}^1$$

und

$$\frac{\text{ganzzahl} - 1}{\text{einträge-pro-seite}} + 1 = \text{seitenanzahl}^2$$

*seitenanzahl*

Anzahl der Seiten des Hashbereichs

*calc-key-länge*

Länge des CALC-Key (siehe [Seite 153](#))

*ganzzahl*

neue Menge an Datensätzen laut POPULATION-Angabe

*satzlänge*

Länge der Satzart (Benutzerteil und Systemteil) (siehe [Seite 149](#) ff)

*einträge-pro-seite*

Anzahl der Einträge (Sätze oder CALC-Tabellen-Einträge) pro Seite



Wenn der Realm, der vom Reorganisieren betroffen ist, mit einer Sekundärzuweisung = 0 eingerichtet ist, z. B. weil Sie keine automatische Realm-Erweiterung wünschen, müssen Sie darauf achten, dass in diesem Realm, ausreichend zusammenhängende leere Seiten (mindestens *seitenanzahl* Seiten) für den neuen Hashbereich zur Verfügung stehen!

Da der alte Hashbereich wieder verwendet werden kann, darf er als freier Bereich betrachtet werden.

Wenn der Realm mit einer Sekundärzuweisung > 0 eingerichtet ist, dann erweitert BREORG den betroffenen Realm bei Bedarf automatisch.

<sup>1</sup> Ergebnis ist abzurunden

<sup>2</sup> Ergebnis ist aufzurunden auf die nächsthöhere Primzahl, wenn sich keine Primzahl ergibt

## Ablaufmeldungen

Nach dem Ausführen der Anweisung REORGANIZE-CALC wird das Ergebnis der CALC-Reorganisation der Satzart und der CALC SEARCH KEYS im Set, der Satzart oder der CALC SEARCH KEYS im Set ausgegeben.

```
***** RESULTS OF CALC-REORGANIZATION OF {RECORD recordname
                                           }
                                           {SEARCH-KEY IN SET setname}

NEW CALC BEGIN           : area ref - seiten nr
NEW NR OF PRIMARY BUCKETS : number pages
NEW NR OF OVERFLOW BUCKETS: number pages
NR OF DATABASE ACCESSES  : number physical io
```

*recordname*

Name der reorganisierten Satzart

*setname*

Name des reorganisierten Sets

*area ref - seiten nr*

Act-Key der ersten CALC-Seite

*number pages*

bei NEW NR OF PRIMARY BUCKETS: neue Anzahl der CALC BUCKETS

bei NEW NR OF OVERFLOW BUCKETS: neue Anzahl der Überlaufseiten

*number physical io*

Anzahl der Ein- und Ausgaben

## Alle Probable Position Pointer (PPP) in einem Realm reorganisieren (REORGANIZE-POINTERS)

Mit der Anweisung REORGANIZE-POINTERS aktualisieren Sie alle in einem Benutzerrealm enthaltenen Probable Position Pointer (PPP).

REORGANIZE-POINTERS
REALM-NAME = <realname>

### **REALM-NAME = <realname>**

Name des Realms, dessen Probable Position Pointer (PPP) aktualisiert werden sollen. In einem BREORG-Lauf kann die Anweisung hintereinander für verschiedene Realms angegeben werden.

REORGANIZE-POINTERS geht wesentlich schneller als die Reorganisation der Probable Position Pointer (PPP) über die anderen REORGANIZE-Funktionen. Werden alle Realms der Datenbank so behandelt, dann werden bei einer nachfolgenden Realm-Erweiterung – durch die Anweisung MODIFY-REALM-SIZE oder online durch den DBH – die neu hinzukommenden Seiten nicht formatiert. Dies bedeutet einen erheblichen Performancegewinn. Bei der Verkleinerung eines Realms ergibt sich kein Performancegewinn.

Bei der Anweisung REORGANIZE-POINTERS verwendet BREORG Arbeitsdateien für die Satzarten und eine weitere Arbeitsdatei für die Sortierung (siehe hierzu [„Arbeitsdateien für die REORGANIZE-POINTERS-Anweisung“ auf Seite 319](#)).

## Tabellen und Set-Konstrukte reorganisieren (REORGANIZE-SET)

Mit der Anweisung REORGANIZE-SET reorganisieren Sie Set-Konstrukte (LIST, CHAIN, POINTER ARRAY) und SEARCH KEY USING INDEX auf Setebene bzw. Satzartebene. SEARCH KEY USING INDEX auf Satzartebene wird über den zugehörigen impliziten Set reorganisiert.

Sie können reorganisieren:

- alle Set Occurrences
- einzelne Set Occurrences, die Sie über die RSQ des Owners vorgeben oder
- Bereiche von Set Occurrences, die Sie über eine Bereichsangabe der Owner-RSQs definieren.

Auf Setebene können Sie mit der Funktion REORGANIZE-SET folgende Tabellen reorganisieren:

- einstufige und mehrstufige Adresslisten
- einstufige und mehrstufige Listen
- mehrstufige Sort-Key-Tabellen
- Ketten
- mehrstufige SEARCH-Key-Tabellen (auch Duplikattabellen).

Auf Satzartebene können Sie mehrstufige SEARCH-Key-Tabellen (auch Duplikattabellen), reorganisieren, wobei hier der Name des impliziten Sets anzugeben ist.

BREORG aktualisiert in den genannten Tabellen alle Adressverweise, die als Probable Position Pointer angelegt wurden, und baut die Tabellen wahlweise mit einem geänderten Füllgrad neu auf. Bei Ketten aktualisiert er auch sämtliche Probable Position Pointer der Kette.

Wollen Sie mehrere Tabellen in einem BREORG-Lauf reorganisieren, so sollten Sie zuerst die Funktionen ausführen, die Datensätze verlagern.

Anschließend sollten Sie die Probable Position Pointer aktualisieren, die noch auf die alte Position der Datensätze hinweisen.

## REORGANIZE-SET

```

SET-NAME = <setname>
,OWNER-SELECTION = *ALL / list-poss(30): <integer 1..2147483647> / *RANGE(...)
    *RANGE(...)
        | FROM-RSQ = <integer 1..2147483647>
        | ,TO-RSQ = <integer 1..2147483647>
,KEY-SELECTION = *ALL / list-poss(30): <integer 1..32767>
,FILLING = *UNCHANGED / <integer 1..100>

```

**SET-NAME = <setname>**

Name des zu reorganisierenden Sets oder des impliziten Sets.

Der Name des impliziten Sets setzt sich zusammen aus SYS\_ und *recordname*.



Der Unterstrich ( \_ ) ist als Bindestrich (-) einzugeben!

**OWNER-SELECTION = \*ALL**

Alle Set Occurrences werden reorganisiert.

**OWNER-SELECTION = list-poss(30): <integer 1..2147483647> / \*RANGE(...)**

Owner und Set Occurrences werden reorganisiert.

**<integer 1..2147483647>**

Die Owner mit den angegebenen RSQs werden reorganisiert.

**\*RANGE (...)**

Alle Set Occurrences, deren Owner-RSQs innerhalb der angegebenen Bereiche liegen, werden reorganisiert.

**FROM-RSQ = <integer 1..2147483647>**

RSQ des ersten Owners, dessen SET bzw. TABLE OCCURRENCE reorganisiert wird bzw. dessen TABLE OCCURRENCES reorganisiert werden.

**TO-RSQ = <integer 1..2147483647>**

RSQ des letzten Owners, dessen SET bzw. TABLE OCCURRENCE reorganisiert wird bzw. dessen TABLE OCCURRENCES reorganisiert werden.

**KEY-SELECTION = \*ALL**

Alle SEARCH KEY USING INDEX und das Set-Konstrukt werden reorganisiert.

**KEY-SELECTION = list-poss(30): <integer 0..32767>**

Alle SEARCH KEY USING INDEX, deren KEY REF angegeben wurde, werden reorganisiert. Die KEY REF kann dem BPSIA-Protokoll entnommen werden.

Wenn die Set-Konstrukte CHAIN, LIST und POINTER ARRAY nicht mit SORTED INDEXED BY definiert sind, ist im BPSIA-Protokoll keine KEY REF eingetragen. In diesem Fall ist für die KEY REF der Wert 0 anzugeben (nur im genannten Fall erlaubt). Es wird dann der Set reorganisiert.

**FILLING = \*UNCHANGED**

In den Tabellen bzw. Set-Konstrukten werden die Probable Position Pointer (PPP) aktualisiert.

**FILLING = <integer 1..100>**

Die Tabellen werden mit dem angegebenen Füllgrad neu aufgebaut.

Die Funktion REORGANIZE-SET reorganisiert Tabellen (ASC-KEY bzw. DESC KEY, SEARCH KEY) und Ketten. Reorganisieren bedeutet hier, BREORG aktualisiert die Probable Position Pointer (PPP).

- in den Sort-Key-Tabelleneinträgen und SEARCH-Key-Tabelleneinträgen,
- in der SCD (Set-Connection-Data) von Datensätzen in Ketten (Vorwärts-Rückwärtsverketzung) und
- in der SCD von Datensätzen mit Ownerverweis (PHYSICALLY LINKED TO OWNER)
- in der SCD von Ownersätzen mit Tabellenverweis (WITH PHYSICAL LINK)

bzw. baut die Tabellen neu auf.

Haben Sie FILLING = *integer...* angegeben, so baut BREORG die angegebenen Tabellen neu auf und füllt die neuen Tabellenseiten bis zu dem angegebenen Prozentsatz mit den vorhandenen, aktualisierten Einträgen. Bei mehrstufigen Tabellen gilt der angegebene Prozentsatz nur für die Grundstufe (Stufe 0). Auf Stufe 1 wird die Tabelle zu 95% gefüllt, auf allen darüberliegenden Stufen wird jeweils ein Tabelleneintrag freigelassen. Zusätzlich aktualisiert er die Einträge in der DBTT der Ownersatzart (Spaltennummer >0).

Bei der ASC-Key-Tabelle bzw. DESC-Key-Tabelle eines Sets mit MODE IS LIST stehen in den Tabellenseiten die Datensätze selbst, d.h. beim Neuaufbau einer solchen Tabelle werden die Datensätze verschoben. In dem Fall aktualisiert BREORG zusätzlich die DBTT-Einträge in Spalte 0 der DBTT der betroffenen Satzart.

Folgender Übersicht können Sie entnehmen, welche Probable Position Pointer (PPP) und Tabellen sich mit der Funktion REORGANIZE-SET reorganisieren lassen (siehe Handbuch „Entwerfen und Definieren“):

DDL- und SSL-Anweisungen		Probable Position Pointer (PPP)		Tabelle			
		Erläuterungen	aktualisieren möglich	Art	neuaufbauen möglich		
MODE IS CHAIN	ORDER IS FIRST/NEXT/PRIOR SORTED	Ownersatz enthält PPP <sup>1</sup> des 1. Membersatzes der Kette	Ja	-	-		
		Vorwärtsverkettung der Membersätze mit RSQ und PPP <sup>1</sup>	Ja				
	ORDER IS LAST oder LINKED TO PRIOR	Ownersatz enthält PPP des letzten Membersatzes der Kette	Ja				
	LINKED TO PRIOR	Rückwärtsverkettung der Membersätze mit RSQ und PPP	Ja				
	ORDER IS SORTED INDEXED BY DEFINED KEYS... ASC/DESC KEY IS	jede Tabellenzeile enthält PPP des Membersatzes	Ja			mehrstufige Sort-Key-Tabelle	Ja
	ORDER IS SORTED INDEXED BY DATABASE-KEY	jede Tabellenzeile enthält PPP des Membersatzes	Ja			mehrstufige Sort-Key-Tabelle	Ja
MODE IS POINTER-ARRAY	ORDER IS FIRST/LAST/NEXT/PRIOR	jede Tabellenzeile enthält PPP des Membersatzes	Ja	einstufige Adressliste	Ja		
	ORDER IS SORTED INDEXED BY DEFINED KEYS... ASC/DESC KEY IS...	jede Tabellenzeile enthält PPP des Membersatzes	Ja	mehrstufige Adressliste	Ja		
	ORDER IS SORTED INDEXED BY DATABASE-KEY oder ORDER IS IMMATERIAL	jede Tabellenzeile enthält PPP des Membersatzes	Ja	mehrstufige Adressliste	Ja		

Tabelle 22: Übersicht über die Möglichkeit der Funktion REORGANIZE-SET

(Teil 1 von 2)

DDL- und SSL-Anweisungen		Probable Position Pointer (PPP)		Tabelle	
		Erläuterungen	aktualisieren möglich	Art	neue aufbauen möglich
MODE IS LIST	ORDER IS FIRST/LAST/NEXT/PRIOR	kein PPP enthalten	-	einstufige Liste	Ja
	ORDER IS SORTED INDEXED (DB-Key o. ASC/DESC-Key)	kein PPP enthalten	-	mehrstufige Liste	Ja
SEARCH KEY ..USING INDEX	TYPE IS REPEATED-KEY	jede Tabellenzeile enthält PPP des Membersatzes	Ja	mehrstufige SEARCH-Key-Tabelle	Ja
	TYPE IS DATABASE-KEY-LIST	kein PPP enthalten	-	Duplikattabelle	Ja
MEMBER IS PHYSICALLY LINKED TO OWNER		Membersatz enthält Zeiger zum Ownersatz (PPP)	Ja	-	-

Tabelle 22: Übersicht über die Möglichkeit der Funktion REORGANIZE-SET

(Teil 2 von 2)

<sup>1</sup> Diese PPPs kommen bei MODE IS CHAIN generell vor.



Ist eine Tabelle vollständig in einer Seite enthalten, so baut BREORG die Tabelle nicht neu auf, selbst wenn Sie FILLING... angegeben haben.

Beim Neuaufbau einer Tabelle mit mehreren Tabellenseiten auf Stufe 0 fügt BREORG in jede Tabellenseite mindestens zwei Einträge ein.

Bei einer Reorganisation einer Datenbank können unabhängig von der Aktivierung der Online-DBTT-Erweiterung DBTT-Extents entstehen. Eine Vergrößerung der DBTT durch BREORG wird im DBCOM und DBDIR durch Vergrößerung der schon vorhandenen DBTT realisiert. Im Benutzerrealm wird eine Vergrößerung der DBTT durch BREORG über DBTT-Extents realisiert, wenn die Ziel-DBTT insgesamt größer als 128 PAM-Seiten ist. Ist die Ziel-DBTT kleiner oder gleich 128 PAM-Seiten, wird sie durch BREORG immer als DBTT-Basis, also am Stück realisiert. Über das Ergebnis von Vergrößerungen oder Verkleinerungen von DBTTs werden Sie mit entsprechenden Meldungen informiert.



## Ablaufmeldungen

Nach dem Ausführen der Anweisung REORGANIZE-SET wird das Ergebnis der Reorganisation eines Sets bzw. einer Tabelle ausgegeben.

```
***** RESULTS OF SET-REORGANIZATION OF SET set name
      NR OF PROCESSED TABLES      : number table occurrences
      NR OF PPP UPDATES            : number of actualized ppps
      NR OF DATABASE ACCESSES     : number physical io
```

*set name*

Name des Sets

*number table occurrences*

Anzahl aller in den Set-Occurrences bearbeiteten Tabellen

*number of actualized ppps*

Anzahl der aktualisierten Probable Position Pointer (PPP)

*number physical io*

Anzahl der Ein- und Ausgaben

## Schema angeben (SPECIFY-SCHEMA)

Mit der Anweisung SPECIFY-SCHEMA legen Sie das Schema fest, für dessen Objekte die Datenbank bearbeitet werden soll.

Die Anweisung SPECIFY-SCHEMA wird nur dann angeboten, wenn die Datenbank über LINK=DATABASE zugewiesen ist. Ansonsten wird der Schemaname in der Anweisung OPEN-DATABASE angegeben.

Wird die Anweisung SPECIFY-SCHEMA nicht als erste Anweisung nach ALLOCATE-BUFFERPOOL angegeben, wird das Benutzerschema angenommen. Die Anweisung SPECIFY-SCHEMA wird anschließend nicht mehr in der SDF-Maske angeboten.

SPECIFY-SCHEMA
SCHEMA-NAME = * <u>STD</u> / <schemaname>

### **SCHEMA-NAME = \*STD**

Es wird für die Bearbeitung der Datenbank der Name des Benutzerschemas genommen, das für die Datenbank definiert wurde (Voreinstellung).

### **SCHEMA-NAME = <schemaname>**

Mit *schemaname* geben Sie das Datenbankschema an, für dessen Objekte die BREORG-Anweisungen auszuführen sind.

Mögliche Werte:

PRIVACY-AND-IQF-SCHEMA  
 COMPILER-SCHEMA  
 Name des Benutzerschemas

Zum Bearbeiten einer Datenbank benötigt BREORG Informationen über die Größe der Realms, Satzarten und Setbeziehungen in der Datenbank. Der Name des Schemas ermöglicht BREORG den Zugriff auf die SIA, die diese Informationen enthält und die auch entsprechend abgeändert wird.

## Subschema angeben (SPECIFY-SUBSCHEMA)

Mit der Anweisung SPECIFY-SUBSCHEMA geben Sie das Subschema an, das für den Neuaufbau von mehrstufigen LIST-Sets mit benutzerdefinierten Schlüsseln, d.h. SORTED INDEXED BY DEFINED KEYS, benötigt wird.

SPECIFY-SUBSCHEMA
SUBSCHEMA-NAME = <subschemaname>

### **SUBSCHEMA-NAME = <subschemaname>**

Name eines Subschemas, das in der Datenbank enthalten ist und die Schlüsselbeschreibung des LIST-Set enthält.

Für die Funktion REORGANIZE-SET benötigt BREORG beim Reorganisieren einer mehrstufigen Liste aus einer SSIA die Beschreibung der zu reorganisierenden Membersatzart und deren Schlüssel. Mit dem Namen des Subschemas, den Sie in der Anweisung SPECIFY-SUBSCHEMA angeben, kann BREORG auf die entsprechende SSIA zugreifen.



Die Anweisung kann mehrfach angegeben werden. Sie gilt dann jeweils für die nachfolgenden Anweisungen, bis die nächste korrekte Anweisung SPECIFY-SUBSCHEMA angegeben wird.

## Anweisung rückgängig machen (UNDO)

Mit der Anweisung UNDO nehmen Sie die letzte korrekte Anweisung zurück. Ausnahme sind die Anweisungen ALLOCATE-BUFFERPOOL, END und UNDO.

Mit einer weiteren Anweisung UNDO wird die vorletzte Anweisung nicht ausgeführt usw.

Die Anweisung UNDO kann nicht durch UNDO zurückgenommen werden.

UNDO

Diese Anweisung hat keine Operanden.

## 9.5 Kommandofolge zum Starten von BREORG

Die hier beschriebene Kommandofolge geht davon aus, dass UDS/SQL mit IMON installiert wurde (siehe Abschnitt „START-Kommandos der UDS/SQL-Programme“ in Kapitel 2 des Handbuchs „[Aufbauen und Umstrukturieren](#)“).

```
01 [/ADD-FILE-LINK LINK-NAME=DATABASE,
      FILE-NAME=[ :catid: ][ $userid. ] dbname.DBDIR]]
02 [/CREATE-FILE FILE-NAME=arbeitsdatei-1 [,SUPPORT=*PUBLIC-DISK
      (SPACE=*RELATIVE(PRIMARY-ALLOCATION=primär,
      SECONDARY-ALLOCATION=sekundär))]
      ,SUPPORT=*PRIVATE-DISK(VOLUME=priv-vsn,
      DEVICE-TYPE=gerät [,SPACE=...])]
      /ADD-FILE-LINK LINK-NAME=SCRTCH1, FILE-NAME=arbeitsdatei-1,
      ACCESS-METHOD=*UPAM]
03 [/CREATE-FILE FILE-NAME=arbeitsdatei-2 [,SUPPORT=*PUBLIC-DISK
      (SPACE=*RELATIVE(PRIMARY-ALLOCATION=primär,
      SECONDARY-ALLOCATION=sekundär))]
      ,SUPPORT=*PRIVATE-DISK(VOLUME=priv-vsn,
      DEVICE-TYPE=gerät [,SPACE=...])]
      /ADD-FILE-LINK LINK-NAME=SORTWK, FILE-NAME=arbeitsdatei-2,
      ACCESS-METHOD=*UPAM]
04 /SELECT-PRODUCT-VERSION PRODUCT-NAME=UDS-SQL, VERSION=version.SCOPE=*TASK
05 /START-UDS-BREORG
06 [//ALLOCATE-BUFFERPOOL BUFFER-SIZE = ...]
07 [//OPEN-DATABASE DATABASE-NAME = ...]
08 [//SPECIFY-SCHEMA SCHEMA-NAME = ...]
09 [//SPECIFY-SUBSCHEMA SUBSCHEMA-NAME = ...]
10... weitere breorg-anweisungen
11 //END
```

- 01, 07     Sie müssen eine der beiden Anweisungen angeben.
- 02, 03     Mit diesen CREATE-FILE-Kommandos können Sie wahlweise die Arbeitsdateien für die Anweisungen REORGANIZE-SET bzw. REORGANIZE-CALC einrichten (siehe [Seite 317](#)). Analog dazu können Sie wahlweise die Arbeitsdateien für die Anweisung REORGANIZE-POINTERS einrichten (siehe [Seite 319](#)).
- 04         Die Version des Dienstprogramms wird ausgewählt.  
Die Angabe der Version wird generell empfohlen, da mehrere UDS/SQL-Versionen parallel installiert sein können.

- 05 BREORG kann von jeder beliebigen Kennung aus aufgerufen werden. Das UDS/SQL-Dienstprogramm kann auch mit den Aliasnamen BREORG oder START-UDS-REORGANIZATION gestartet werden.
- 08 Die Anweisung SPECIFY-SCHEMA wird nur angeboten, wenn vorher ein Kommando ADD-FILE-LINK..., LINK-NAME=DATABASE gegeben wurde.
- 09 Nur erforderlich bei der Funktion REORGANIZE-SET für den Neuaufbau von mehrstufigen LIST-Sets mit benutzerdefiniertem Schlüssel (d.h. SORTED INDEXED BY DEFINED KEYS).

## 9.6 Beispiele

### *Beispiel 1*

Der Realm KLEIDUNG der Datenbank VERSAND soll um 12 Datenbankseiten verkleinert werden.

```

/SELECT-PRODUCT-VERSION PRODUCT-NAME=UDS-SQL,VERSION=02.8A00
/START-UDS-BREORG
***** START      BREORG      (UDS/SQL V2.8 0000 )    2015-06-28  11:26:31
//OPEN-DATABASE DATABASE-NAME=VERSAND,SCHEMA-NAME=ARTIKELVERSAND
//MODIFY-REALM-SIZE REALM-NAME=KLEIDUNG,      REALM-SIZE=*RELATIVE(DIFFERENCE=-12)
//END
***** BEGIN OF REALM-SIZE-MODIFICATION      AT 11:26:32
***** RESULTS OF FPA-REORGANIZATION OF AREA KLEIDUNG
      NEW FPA FIRST PAGE      : NOT CHANGED
      NEW FPA LAST PAGE      : NOT CHANGED
      NEW FPA SIZE            : NOT CHANGED
      NEW NR OF PAGES        :           42
***** END    OF REALM-SIZE-MODIFICATION      AT 11:26:32

***** DIAGNOSTIC SUMMARY OF BREORG

                NO WARNINGS
                NO ERRORS
                NO SYSTEM-ERRORS

***** END OF DIAGNOSTIC SUMMARY
***** NR OF DATABASE ACCESSES :           149
***** NORMAL END  BREORG      (UDS/SQL V2.8 0000 )    2015-06-28  11:26:32

```

*Beispiel 2*

Der Realm ARTIKELRLM wird um soviel vergrößert, dass die FPA-Basis nicht mehr ausreicht. Es entsteht genau 1 FPA-Extent.

```
/SELECT-PRODUCT-VERSION PRODUCT-NAME=UDS-SQL,    VERSION=02.8A00
/START-UDS-BREORG
***** START      BREORG      (UDS/SQL V2.8 0000 )    2015-06-28  11:26:32
//OPEN-DATABASE DATABASE-NAME=VERSAND,SCHEMA-NAME=ARTIKELVERSAND
//MODIFY-REALM-SIZE REALM-NAME=ARTIKELRLM, REALM-SIZE=*RELATIVE(DIFFERENCE=2000)
//END
***** BEGIN OF REALM-SIZE-MODIFICATION      AT 11:26:33
***** RESULTS OF FPA-REORGANIZATION OF AREA ARTIKELRLM
NEW FPA FIRST PAGE      : NOT CHANGED
NEW FPA LAST  PAGE      :              11-              78
NEW NR OF EXTENTS      :              1
NEW FPA SIZE            :              33
NEW NR OF PAGES        :              2062
***** END    OF REALM-SIZE-MODIFICATION      AT 11:26:33

***** DIAGNOSTIC SUMMARY OF BREORG

                NO WARNINGS
                NO ERRORS
                NO SYSTEM-ERRORS

***** END OF DIAGNOSTIC SUMMARY
***** NR OF DATABASE ACCESSES :              108
***** NORMAL END  BREORG      (UDS/SQL V2.8 0000 )    2015-06-28  11:26:33
```



*Beispiel 3*

Die CALC-Bereiche der Satzart ARTIKEL der Datenbank VERSAND sollen reorganisiert werden.  
Die SEARCH KEY USING CALC-Bereiche, die zur Satzart ARTIKEL gehören, werden nicht reorganisiert.

```

/SELECT-PRODUCT-VERSION PRODUCT-NAME=UDS-SQL,   VERSION=02.800
/START-UDS-BREORG
***** START      BREORG      (UDS/SQL V2.8 0000 )    2015-06-28  11:26:33
//OPEN-DATABASE DATABASE-NAME=VERSAND,SCHEMA-NAME=ARTIKELVERSAND
//REORGANIZE-CALC RECORD-NAME=ARTIKEL,CALC-RECORD=*WITHIN-POPULATION(POPULATION=200),-
//  CALC-SEARCHKEY=NONE
//END
***** BEGIN OF CALC-REORGANIZATION      AT 11:26:34
***** RESULTS OF CALC-REORGANIZATION OF RECORD ARTIKEL
      NEW CALC BEGIN      :      5-      5
      NEW NR OF PRIMARY BUCKETS :      2
      NEW NR OF OVERFLOW BUCKETS:      0
***** RESULTS OF CALC-REORGANIZATION OF RECORD ARTIKEL
      NEW CALC BEGIN      :      6-      7
      NEW NR OF PRIMARY BUCKETS :      2
      NEW NR OF OVERFLOW BUCKETS:      0
***** RESULTS OF CALC-REORGANIZATION OF RECORD ARTIKEL
      NEW CALC BEGIN      :      7-      5
      NEW NR OF PRIMARY BUCKETS :      2
      NEW NR OF OVERFLOW BUCKETS:      0
***** RESULTS OF CALC-REORGANIZATION OF RECORD ARTIKEL
      NEW CALC BEGIN      :      8-      9
      NEW NR OF PRIMARY BUCKETS :      2
      NEW NR OF OVERFLOW BUCKETS:      0
***** RESULTS OF CALC-REORGANIZATION OF RECORD ARTIKEL
      NEW CALC BEGIN      :      9-      5
      NEW NR OF PRIMARY BUCKETS :      2
      NEW NR OF OVERFLOW BUCKETS:      0
***** RESULTS OF CALC-REORGANIZATION OF RECORD ARTIKEL
      NEW CALC BEGIN      :     10-      4
      NEW NR OF PRIMARY BUCKETS :      2
      NEW NR OF OVERFLOW BUCKETS:      0
***** END    OF CALC-REORGANIZATION      AT 11:26:34

***** DIAGNOSTIC SUMMARY OF BREORG

                NO WARNINGS
                NO ERRORS
                NO SYSTEM-ERRORS

***** END OF DIAGNOSTIC SUMMARY

```

```
***** NR OF DATABASE ACCESSES :          149
***** NORMAL END   BREORG      (UDS/SQL V2.8 0000 )    2015-06-28  11:26:34
```

---

## 10 Wiederverwendung von Database Keys und Freiplatzsuche steuern mit BMODTT

Mit dem Dienstprogramm BMODTT können Sie das Wiederverwenden von Database Keys und die Freiplatzsuche steuern.

### Database Keys Wiederverwenden

Die DBTT-Einträge gelöschter Sätze unterscheiden sich inhaltlich von denjenigen, für welche noch nie ein Satz gespeichert war. Standardmäßig oder im Falle der BMODTT-Anweisung REUSE machen DBH und BINLOAD beim Speichern von Sätzen keinen Unterschied zwischen den beiden Arten von DBTT-Einträgen. Nur nach einer BMODTT-Anweisung KEEP werden die Database Keys gelöschter Sätze nicht wiederverwendet. Mit Hilfe der BMODTT-Anweisung REMOVE können Sie einmalig alle Database Keys gelöschter Sätze denjenigen noch niemals verwendeter Sätze gleichsetzen, sodass das Gedächtnis über die in der Vergangenheit gelöschten Sätze verloren geht; dabei bleibt die bisherige Option in Bezug auf ihre weitere Wiederverwendbarkeit durch die Dienstprogramme BMODTT oder BOUTLOAD (REUSE, KEEP) unverändert.

Die Information über gelöschte Sätze wird nicht von BOUTLOAD ausgegeben und geht auch bei Anwendung der BOUTLOAD-Anweisungen EXPORT-RECORD oder REMOVE-RECORD mit RECORD-NAME=\*ALL verloren, da in diesen Fällen die Datenbank neu formatiert wird. Auch bei einer Umstellung der Datenbank mit BPGSIZE wird die Information über gesperrte DBTT-Einträge nicht in die neue Datenbank übernommen.

Die Spalte 'NO REUSE' im BPSIA-Protokoll gibt Auskunft über die grundsätzliche Wiederverwendbarkeit von Database Keys.

### Freiplatzsuche

Mit den Anweisungen SET oder RESET kann bei der Freiplatzsuche gesteuert werden, ob der Aspekt des sparsamen Umgangs mit Speicherplatz im Vordergrund steht (SET) oder ob die per SSL gewünschten Lagebestimmungen 'ATTACHED' und 'PLACEMENT OPTIMIZATION' erfüllt werden sollen (RESET, Standardeinstellung).

## 10.1 Systemumgebung

Der Datenbankadministrator darf BMODTT nicht während einer laufenden ändernden Session aufrufen.

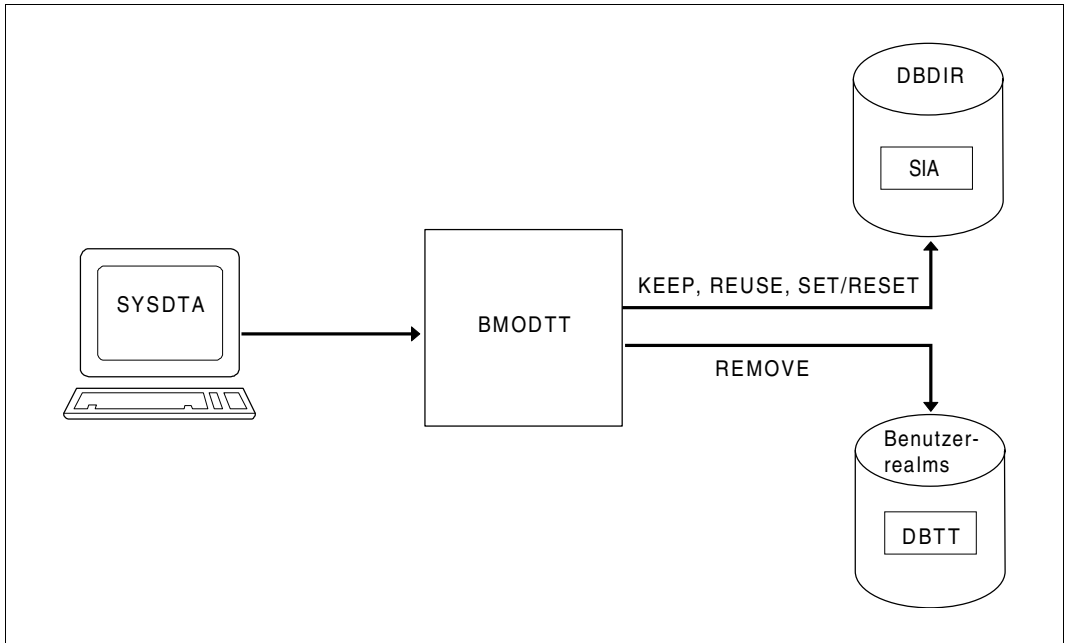


Bild 19: Systemumgebung von BMODTT

BPMODTT berücksichtigt beim Start ggf. eine zugewiesene UDS/SQL-Pubset-Deklaration (siehe Handbuch „[Datenbankbetrieb](#)“, Pubset-Deklarations-Jobvariable). Eine fehlerhafte Zuweisung führt zum Programmabbruch.

## 10.2 Anweisungen für BMODTT

Anweisung	Bedeutung
KEEP	frei gewordene Database Keys sperren
REMOVE	die gesperrten Database Keys einmalig zur Wiederverwendung freigeben
RESET	bei der Freiplatzsuche vom Ende des belegten Teils des Realms an suchen
REUSE	Database Keys grundsätzlich zur Wiederverwendung freigeben
SET	bei der Freiplatzsuche vom Anfang des Realm an suchen

Tabelle 23: Anweisungen für BMODTT

Im Folgenden sind die Anweisungen entsprechend dem Aufbau der Anweisungen zusammengefasst.

---

$\left. \begin{array}{l} \text{KEEP} \\ \text{REMOVE} \\ \text{REUSE} \end{array} \right\}$	$\left\{ \begin{array}{l} \text{DBKEY OF RECORD} \\ \text{OF RECORD} \end{array} \right\}$	$\left\{ \begin{array}{l} \text{satzname-1[,satzname-2]...} \\ \text{*ALL[ EXCEPT satzname-1[,satzname-2]...]} \end{array} \right\}$
---	--	--

---

**KEEP** Frei gewordene Database Keys werden gesperrt und können nicht wiederverwendet werden.

### REMOVE

Die gesperrten Database Keys einmalig zur Wiederverwendung freigeben. Dabei bleibt die Option in Bezug auf ihre Wiederverwendbarkeit erhalten. Für die angegebenen Satzarten wird der höchste verwendete Database Key neu ermittelt und in der entsprechenden SIA hinterlegt.

Desweiteren wird für die betroffenen Satzarten der Pegel, an welchem eine STORE-Anweisung oder BINILOAD Database Keys vergibt, vor den ersten freien Eintrag der DBTT gesetzt, bzw. auf den Eintrag mit der RSQ 1, falls dieser frei ist. Ein separates Setzen des Pegels ohne Freigabe der gesperrten Database Keys ist nicht möglich.

### REUSE

Frei gewordene Database Keys können grundsätzlich wieder verwendet werden. Dies ist die Standardeinstellung vor der Verwendung von BMODTT.

*satzname-1[,satzname-2]...*

Liste der betroffenen Satzarten

\*ALL Alle Satzarten der Datenbank sind betroffen.

\*ALL EXCEPT *satzname-1*[,*satzname-2*]...

Alle Satzarten der Datenbank sind betroffen, außer den nach EXCEPT aufgelisteten.

---

$\left. \begin{array}{l} \text{SET} \\ \text{RESET} \end{array} \right\}$	REUSE-FREE-SPACE OF REALM	$\left\{ \begin{array}{l} \textit{realmname-1}[, \textit{realmname-2}] \dots \\ *ALL[ EXCEPT \textit{realmname-1}[, \textit{realmname-2}] \dots] \end{array} \right\}$
---	---------------------------	--

---

**SET** Bei der Freiplatzsuche wird vorhandener Platz ab der ersten Seite des Realm ermittelt, die noch freien Platz enthält.

**RESET**

Bei der Freiplatzsuche wird vorhandener Platz ab der ersten freien Seite ermittelt, auf die bis zum Ende des Realms keine teilweise gefüllten Seiten mehr folgen, sondern nur noch freie oder voll belegte Seiten.

Dies ist die Standardeinstellung vor der Verwendung des Dienstprogramms BMODTT.

*realmname-1*[,*realmname-2*]...

Liste der betroffenen Realms

\*ALL Alle Realms der Datenbank sind betroffen

\*ALL EXCEPT *realmname-1*[,*realmname-2*]...

Alle Realms der Datenbank sind betroffen, außer den nach EXCEPT aufgelisteten.

## 10.3 Kommandofolge zum Starten von BMODTT

Die hier beschriebene Kommandofolge geht davon aus, dass UDS/SQL mit IMON installiert wurde (siehe Abschnitt „START-Kommandos der UDS/SQL-Programme“ in Kapitel 2 des Handbuchs „[Aufbauen und Umstrukturieren](#)“).

```
01 /ADD-FILE-LINK LINK-NAME=DATABASE,  
    FILE-NAME=[ :catid: ] [ $userid. ] dbname.DBDIR [ .copyname ]  
02 /SELECT-PRODUCT-VERSION PRODUCT-NAME=UDS-SQL, VERSION=version. SCOPE=*TASK  
03 /START-UDS-BMODTT  
04 bmodtt-anweisungen  
05 END
```

- 01 Hier ist die Angabe von *:catid:* zulässig (siehe Handbuch „[Datenbankbetrieb](#)“)
- 02 Die Version des Dienstprogramms wird ausgewählt.  
Die Angabe der Version wird generell empfohlen, da mehrere UDS/SQL-Versionen parallel installiert sein können.
- 03 BMODTT darf nur unter der Kennung des Datenbankadministrators aufgerufen werden. Das UDS/SQL-Dienstprogramm kann auch mit dem Aliasnamen BMODTT gestartet werden.
- 04 Bei jeder Umstrukturierung mit BCHANGE, BALTER wird die Standardeinstellung (REUSE) hinterlassen.

Das Dienstprogramm BMODTT ist wiederanlauffähig.





---

# Fachwörter

Dieses Fachwortverzeichnis enthält Definitionen wichtiger Begriffe, die in den Handbüchern zu UDS/SQL verwendet werden.

*Kursiv* gedruckte Fachwörter in den Definitionen verweisen auf entsprechende Definitionen für diese Fachwörter.

Ein „siehe“-Verweis für ein Fachwort verweist auf das in den UDS/SQL-Handbüchern hauptsächlich verwendete Fachwort.

## A

### **Act-Key**

act-key

(actual key) Aktuelle Adresse einer *Seite*, bestehend aus *Realmnummer* und *Seitennummer*.

### **Act-Key-0-Seite**

act-key-0 page

Erste *Seite* eines *Realm*. Sie enthält allgemeine Informationen über den *Realm*, z.B.

- Erstellungszeitpunkt des *Realm*,
- Zeitpunkt der letzten Änderung,
- *interne Versionsnummer* des *Realm*,
- Unterbrechungsinformationen des Systems (*Systembreak-Informationen*),
- ggf. Kenndaten für den *Warmstart*.

### **Act-Key-N-Seite**

act-key-N page

Kennseite eines *Realm* mit der höchsten *Seitennummer*.  
Kopie der *Act-Key-0-Seite*.

### **Administratortask**

administrator task

Task des *independent DBH*. Der *Datenbankadministrator* kann über diese Task den Ablauf des *independent DBH* steuern.

**Adresse, physische**

address, physical

siehe *Act-Key* oder *Probable Position Pointer (PPP)***Adressliste**

pointer array

Tabelle, die auf die *Membersätze* einer *Set-Occurrence* verweist. Dient dem *sequentiellen* und *direkten Zugriff* auf die *Membersätze*.**AFIM**

AFIM

siehe *After-Image***After-Image**

after-image

Geänderter Teil einer *Seite* **nach** einer Änderung des Seiteninhalts.After-Images schreibt der *DBH* sowohl in die *RLOG-Datei* als auch in die *ALOG-Datei*.**After-Image, ALOG-Datei**

after-image, ALOG file

Die After-Images werden in die *ALOG-Datei* geschrieben, wenn der *ALOG-Puffer* voll ist. Die After-Images in der *ALOG-Datei* werden zur *Langzeitsicherung*, d.h. für lange Zeit benötigt. Sie werden benutzt, um eine *Originaldatenbank* zu rekonstruieren oder eine *Schattendatenbank* zu aktualisieren.**After-Image, RLOG-Datei**

after-image, RLOG file

Die After-Images werden in die *RLOG-Datei* geschrieben, **bevor** die Änderungen auf der *Datenbank* festgeschrieben werden. Die After-Images in der *RLOG-Datei* werden nur zum *Warmstart* benötigt und deshalb zyklisch überschrieben.**ALOG-Datei**

ALOG file

Datei zur *Langzeitsicherung*, siehe *After-Image*.**ALOG-Folgenummer**

ALOG sequence number

Kennzeichnung im Dateinamen der *ALOG-Dateien* (000000001 - 999999999). Die erste *ALOG-Datei* einer *Datenbank* trägt immer die Folgenummer 000000001.

**Ankersatz**

anchor record

*Satz*, den UDS/SQL automatisch als *Ownersatz* für *SYSTEM-Sets* einrichtet. Er enthält keine mit der *Schema-DDL* definierten *Felder* und es kann auf ihn nicht zugegriffen werden.

**Anweisungscode**

statement code

Nummer, die im ersten Teil des Feldes *DATABASE-STATUS* hinterlegt wird und die darüber informiert, bei welcher *DML*-Anweisung ein Sonderzustand aufgetreten ist.

**Anwenderprogramm (AP)**

application program (AP)

Z.B. *COBOL-DML*-Programm, IQS.

**Anwendertask**

user task

Ausführung eines *Anwenderprogramms* bzw. *openUTM*-Teilprogramms, einschließlich der vom System dazugebundenen Teile.

**Anwendung**

application

Umsetzung einer Aufgabenstellung in ein *Anwenderprogramm* oder mehrere Anwenderprogramme, die mit UDS/SQL-*Datenbanken* arbeiten.

**Area**

area

siehe *Realm*

**Ascending-Key (ASC-Key)**

ascending key (ASC key)

*Primärschlüssel* eines *Set*. Der Ascending-Key legt die Reihenfolge der *Membersätze* in den *Set-Occurrences* nach aufsteigenden Schlüsselwerten fest.

**Auftrag**

request

Die Funktionen, die durch die *DAL*-Kommandos ADD DB, ADD RN, DROP DB, DROP RN, NEW RLOG und CHECKPOINT zunächst im *DBH* nur vorgemerkt sind, werden erst durch das *DAL*-Kommando PERFORM zur Durchführung angestoßen.

**automatische DBTT-Erweiterung**

automatic DBTT extension

Einige Dienstprogramme erweitern die Anzahl möglicher Sätze einer Satzart bei Engpässen automatisch; hierfür ist keine gesonderte Administration erforderlich.

Siehe auch *Online-DBTT-Erweiterung*.

**automatische Realm-Erweiterung**

automatic realm extension

Einige Dienstprogramme erweitern Realms bei Engpässen automatisch; hierfür ist keine gesonderte Administration erforderlich.

Siehe auch *Online-Realm-Erweiterung*.

## B

**Base Interface Block**

Base Interface Block

siehe *BIB*

**Before-Image**

before-image

Teil einer *Seite* vor einer Änderung des Seiteninhalts.

Before-Images schreibt der *DBH* in die *RLOG-Dateien*. Dort werden die Before-Images während des Datenbankbetriebs geschrieben, bevor die Änderungen auf der *Datenbank* festgeschrieben werden. Voraussetzung ist, dass *RLOG-Dateien* geführt werden.

**Benutzerdatenbank**

user database

Die *Realms* und Dateien der *Datenbank*, die der Anwender benötigt, um Daten in die Datenbank zu speichern und wiederzugewinnen.

Dies sind:

- das *Database Directory (DBDIR)*
- die *Benutzerrealms*
- die Modulbibliothek für *Hashroutinen (HASHLIB)*.

**Benutzerrealm**

user realm

Im Realm-Eintrag der *Schema-DDL* definierter *Realm*. Er enthält u.a. die Benutzersätze.

**Bezeichner**

identifizier

Name, den der Datenbankentwerfer für ein *Feld* vergibt, das UDS/SQL automatisch anlegt. Feldtyp und Feldlänge richtet UDS/SQL nach dem vorgegebenen Verwendungszweck des Feldes aus.

**BFIM**

BFIM

siehe *Before-Image*

**BIB**

BIB

(Base Interface Block) Standardschnittstelle zwischen UDS/SQL und jedem einzelnen Benutzer; enthält u.a. die *RECORD-AREA* (Benutzersätze wie im *Subschema* definiert).

**Buffer Pools**

buffer pools

siehe *System Buffer Pools* und *exklusiver Buffer Pool*

## C

**CALC-Key**

CALC key

*Schlüssel*, dessen Schlüsselwerte durch eine *Hashroutine* in eine relative *Seitennummer* umgerechnet werden.

**CALC-SEARCH-Key**

CALC SEARCH key

*Sekundärschlüssel*, der als *Zugriffspfad* für *direkten Zugriff* über *Hashverfahren* realisiert wird.

**CALC-Seite**

CALC page

*Seite* eines *Hashbereichs*.

**CALC-Tabelle**

## CALC table

Tabelle in einer direkten/indirekten *CALC-Seite*, deren Einträge auf die gespeicherten Sätze verweisen.

Sie enthält pro Zeile:

- den *CALC-Key*
- die *Satzfolgennummer*
- die Distanz zum zugehörigen *Seitenindex-Eintrag* (direkte *CALC-Seite*) bzw. den *Probable Position Pointer* (indirekte *CALC-Seite*)

**CALL-DML**

## CALL DML

*DML*, die von verschiedenen Programmiersprachen (Assembler, COBOL, FORTRAN, PASCAL, PL/1) über die *CALL*-Schnittstelle angesprochen wird.

**CHAIN**

## CHAIN

Speicherungsart für eine *Set-Occurrence*, bei der jeder *Satz* einen Zeiger auf seinen Nachfolger mitführt.

**Character Separated Values (CSV)**

## Character Separated Values

Ausgabeformat, bei dem die Werte durch ein vorgegebenes Zeichen getrennt sind

**CHECK-TABLE**

## CHECK-TABLE

Prüftabelle, die der *DDL*-Compiler bei der *Subschema-DDL*-Übersetzung erstellt und die vom COBOL-Compiler und von *CALL-DML* benutzt wird, um zu prüfen, ob die angegebenen *DML*-Anweisungen im *Anwenderprogramm* zulässig sind. Sie befindet sich im *COSSD* bzw. im *SSITAB-Modul*.

**Clone-Paar, Clone-Pubset, Clone-Session, Clone-Unit**

clone pair, clone pubset, clone session, clone unit

Eine Clone-Unit ist die Kopie einer (Original-)Unit (logische Platte im BS2000) zu einem bestimmten Zeitpunkt („Point-in-Time-Kopie“). Die Komponente TimeFinder/Clone erstellt diese Kopie wahlweise als komplette Kopie oder als „Snapshot“.

Nach der Aktivierung sind Unit und Clone-Unit voneinander getrennt, Anwendungen können auf beide zugreifen.

Unit und Clone-Unit bilden zusammen ein Clone-Paar. TimeFinder/Clone verwaltet es in einer sogenannten Clone-Session.

Wenn es zu allen Units eines Pubsets Clone-Units gibt, so bilden diese Clone-Units zusammen das Clone-Pubset.

Details zu diesem Thema finden Sie im Handbuch „[Einführung in die Systembetreuung](#)“.

**COBOL-DML**

COBOL DML

In den COBOL-Sprachumfang integrierte *DML*.

**COBOL-Laufzeitsystem**

COBOL runtime system

Laufzeitsystem. Mehrfachbenutzbare Routinen, die der COBOL-Compiler (COBOL2000 bzw. COBOL85) zur Ausführung komplexer Anweisungen auswählt.

**COBOL Subschema Directory (COSSD)**

COBOL Subschema Directory (COSSD)

liefert dem COBOL-Compiler die Subschema-Informationen für die Übersetzung der *DB-Anwenderprogramme*.

**Common Memory**

common memory

Von mehreren Tasks gemeinsam benutzbarer Speicherbereich. Er besteht bei UDS/SQL immer aus dem *Common Pool* und dem *Communication Pool* und je nach Anwendungsfall aus dem *SSITAB Pool* (siehe *SSITAB-Modul*), wenn die *CALL-DML* verwendet wird. Beim Einsatz von UDS-D, besteht er zusätzlich noch aus dem *Distribution Pool* und dem *Transfer Pool*.

**Common Pool**

common pool

Kommunikationsbereich des *independent DBH* für die Verständigung der *DBH-Module* untereinander. Er enthält u.a. einen Ein-/Ausgabe-Puffer für *Seiten (Buffer Pools)*.

**Communication Pool**

communication pool

Kommunikationsbereich des *independent DBH* für *Anwenderprogramme*. Er dient u.a. zur Aufnahme der Base Interface Blocks (*BIB*).

**Compilerdatenbank**

compiler database

Die *Realms* und Dateien der *Datenbank*, die die UDS/SQL-Compiler benötigen.

Dies sind:

- das *Database Directory (DBDIR)*
- der *Database Compiler Realm (DBCOM)*
- das *COBOL Subschema Directory (COSSD)*

**COMPILER-SCHEMA**

COMPILER-SCHEMA

UDS/SQL-internes *Schema* der *Compilerdatenbank*.

**COMPILER-SUBSCHEMA**

COMPILER-SUBSCHEMA

UDS/SQL-internes *Subschema* der *Compilerdatenbank*.

**Compound Key**

compound key

siehe *Schlüssel, zusammengesetzter*

**Connectionmodul**

connection module

siehe *Verbindungsmodul*

**Consistency Record**

consistency record

Verwaltungssatz mit Konsistenz-Zeitstempeln im *DBDIR*. Bei einer Änderung in einem *Realm* trägt der *DBH* im Consistency Record und im geänderten Realm Datum und Uhrzeit ein. Beim Anschließen von *Datenbanken* oder *Realms* an eine *Session* überprüft der *DBH* anhand dieser Zeitstempel, ob die *Realms* jeder *Datenbank* unter dem Konsistenzaspekt zueinander passen.

**COSSD**

COSSD

siehe *COBOL Subschema Directory*.



**CRA**

CRA

(Current Record of Area) *Satz*, der in der *Currency-Tabelle* als aktueller Satz eines bestimmten *Realm* (Area) verzeichnet ist.

**CRR**

CRR

(Current Record of Record) *Satz*, der in der *Currency-Tabelle* als aktueller Satz einer bestimmten *Satzart* (Record) verzeichnet ist.

**CRS**

CRS

(Current Record of Set) *Satz*, der in der *Currency-Tabelle* als aktueller Satz eines bestimmten *Set* verzeichnet ist.

**CRU**

CRU

(Current Record of Rununit) *Satz*, der in der *Currency-Tabelle* als aktueller Satz der *Verarbeitungskette* verzeichnet ist.

**CSV**

CSV

siehe Character Separated Values

**Currency-Tabelle**

currency table

Die Currency-Tabelle enthält

- die CURRENT-OF-AREA-Tabelle (Tabelle der *CRAs*),
- die CURRENT-OF-RECORD-Tabelle (Tabelle der *CRRs*),
- die CURRENT-OF-SET-Tabelle (Tabelle der *CRSs*).

**CURRENT-OF-AREA-Tabelle**

CURRENT OF AREA table

siehe *Currency-Tabelle*

**CURRENT-OF-RECORD-Tabelle**

CURRENT OF RECORD table

siehe *Currency-Tabelle*

**CURRENT-OF-SET-Tabelle**

CURRENT OF SET table

siehe *Currency-Tabelle*

## D

**DAL**

DAL

(Database Administrator Language) Datenbankadministratorsprache für Kommandos zum Überwachen und Steuern einer *Session*.

**Database Compiler Realm (DBCOM)**

database compiler realm (DBCOM)

Speichert Einzelheiten über die *Realms*, *Sätze* und *Sets*, die der Anwender in der *Schema-DDL* und der *Subschema-DDL* definiert hat.

**Database Directory (DBDIR)**

database directory (DBDIR)

Enthält u.a. die *SIA*, alle *SSIAs* und Informationen über die *Zugriffsberechtigungen*.

**Database Key**

database key

*Schlüssel*, dessen Schlüsselwerte einen *Satz* in der *Datenbank* eindeutig identifizieren. Er setzt sich aus einer *Satzartnummer* und einer *Satzfolgenummer* zusammen. Die Schlüsselwerte können vom Datenbankprogrammierer vergeben oder von UDS/SQL automatisch erzeugt werden.

**Database-Key-Feld**

database key item

Feld vom Typ DATABASE-KEY oder DATABASE-KEY-LONG, das für die Aufnahme von *Database-Key*-Werten definiert wird.

Felder vom Typ DATABASE-KEY und Felder vom Typ DATABASE-KEY-LONG unterscheiden sich hinsichtlich der Feldlänge (4 byte / 8 byte) und des Wertebereichs.

**DATABASE-KEY-Feld**

DATABASE-KEY item

siehe *Database-Key-Feld*

**DATABASE-KEY-LONG-Feld**

DATABASE-KEY-LONG item

siehe *Database-Key-Feld*

**DATABASE-STATUS**

DATABASE-STATUS

5 byte langes Feld zur Anzeige des Datenbankzustands. Der Datenbankzustand besteht aus dem *Anweisungscodex* und dem *Statuscode*.

**Datenbank (DB)**

database

Zusammengehörige Datenbestände, die mit Hilfe eines *Datenbanksystems* ausgewertet, bearbeitet und verwaltet werden.

Eine Datenbank wird durch den Datenbanknamen identifiziert.

Eine UDS/SQL-Datenbank besteht aus der *Benutzerdatenbank* und der *Compilerdatenbank*.

Zum Schutz vor Datenverlust kann parallel zur Datenbank (Originaldatenbank) eine *Schattendatenbank* betrieben werden.

**Datenbankadministrator**

database administrator

Person, die die *Datenbank* im laufenden Betrieb verwaltet und steuert. Der Datenbankadministrator bedient die Dienstprogramme und die Database Administrator Language (*DAL*).

**Datenbankkopie**

database copy

Kopie einer konsistenten *Datenbank*, die zu einem beliebigen Zeitpunkt erstellt wurde.

**Datenbank-Jobvariable**

database job variable

Jobvariable, in der UDS/SQL Informationen über den Zustand einer *Datenbank* hinterlegt.

**Datenbankseite**

database page

siehe *Seite*

**Datenbanksystem**

database system

Softwaresystem, das alle Aufgaben im Zusammenhang mit Verwaltung und Kontrolle großer Datenbestände unterstützt. Die im Datenbanksystem enthaltenen Verfahren führen zu einer stabilen, redundanzfreien und erweiterbaren Datenorganisation. Sie ermöglichen einer Vielzahl von Anwendern den parallelen Zugriff auf die *Datenbanken* und gewährleisten einen konsistenten Datenbestand.

**Datenbankzustand**

database status

siehe *DATABASE-STATUS*

**Datendeadlock**

data deadlock

siehe *Deadlock***Datengruppe**

group item

Benennbare Zusammenfassung von *Satzelementen*.**Datenschutz**

data protection (privacy)

Schutz vor unberechtigtem Zugriff auf Daten. Datenschutz wird in UDS/SQL verwirklicht durch das Schema/Subschema-Konzept und die Zugriffsrechtsprüfung. Die *Zugriffsrechte* werden mit dem Dienstprogramm BPRIVACY vergeben.

**Datensicherung**

data backup

Schutz vor Datenverlust bei Software- oder Hardware-Fehlern.

**DBCOM**DBCOMsiehe *Database Compiler Realm***DBDIR**DBDIRsiehe *Database Directory***DBH**DBH

(Database Handler) Programm (bzw. Programmgruppe), das den Zugriff auf die *Datenbank(en)* einer *Session* steuert und alle dabei notwendigen Verwaltungsarbeiten übernimmt.

**DBH, independent**DBH, independentsiehe *independent DBH***DBH, Ladeparameter**DBH load parameterssiehe *Ladeparameter (DBH)*.**DBH, linked-in**DBH, linked-insiehe *linked-in DBH*

**DBH-Ende**

DBH end

Beenden des *DBH* Programmlaufs. DBH-Ende kann entweder *Session-Ende* oder *Session-Abbruch* sein.

**DBH-Start**

DBH start

Starten des *DBH* Programmlaufs. DBH-Start kann entweder *Session-Beginn* oder *Session-Wiederanlauf* sein.

**DB-Key**

DB key

Siehe *Database Key*.

**DB-Konfiguration**

DB configuration

(database configuration) Die Menge aller *Datenbanken*, die einem *DBH* während einer *Session* momentan zugeschaltet ist. Die DB-Konfiguration kann sich im Laufe einer *Session* ändern, durch *DAL*-Kommandos oder durch die DBH-Fehlerbehandlung.

Eine DB-Konfiguration kann zu *Session-Beginn* auch leer sein. Mit *DAL*-Kommandos können Datenbanken nach *Session-Beginn* angeschlossen werden. Mit *DAL*-Kommandos können aber auch während einer *Session* Datenbanken ausgeschlossen werden.

**DB-Status-Datei**

DB status file

(database status file) Enthält Informationen über die letzten zurückgesetzten *Transaktionen*. Diese Informationen werden von UTM-S und bei verteilter Verarbeitung mit UDS-D/openUTM-D zum *Session-Wiederanlauf* benötigt.

**DBTT**

DBTT

(Database Key Translation Table) Tabelle, in der UDS/SQL mit Hilfe eines Database-Key-Wertes die *Seitenadresse (Act-Key)* des zugehörigen *Satzes* und der zugehörigen Tabellen findet.

Die DBTT des SSIA-RECORD besteht nur aus der DBTT-Basis. Bei allen anderen Satzarten besteht die DBTT jeweils aus einer Basistabelle (DBTT-Basis) und eventuell einer der mehreren Erweiterungstabellen (DBTT-Extents), welche durch eine Online-DBTT-Erweiterung oder durch BREORG entstehen.

**DBTT-Ankerseite**

DBTT anchor page

Im Realm der zugehörigen DBTT liegende Seite, in der DBTT-Basis und DBTT-Extents verwaltet werden. Möglicherweise sind mehrere untereinander verkettete DBTT-Ankerseiten zur Verwaltung der DBTT nötig.

**DBTT-Basis**

DBTT base

siehe *DBTT***DBTT-Extent**

DBTT extent

siehe *DBTT***DBTT-Seite**

DBTT page

*Seite*, die die *DBTT* oder einen Teil der DBTT einer *Satzart* enthält.

**DCAM**

DCAM

Teil des Datenkommunikationssystems TRANSDATA

**DCAM-Anwendung**

DCAM application

Kommunikationsanwendung, die die Kommunikationsmethode *DCAM* benutzt. Eine DCAM-Anwendung bietet Kommunikationsmöglichkeit zwischen

- einer DCAM-Anwendung und Datensichtstationen.
- DCAM-Anwendungen untereinander im selben oder in verschiedenen Verarbeitungsrechnern, sowie mit *entfernten Konfigurationen*.
- einer DCAM-Anwendung und einer *openUTM*-Anwendung.

**DDL**

DDL

(Data Description Language) Formale Sprache zur Beschreibung der logischen Datenstruktur.

**Deadlock**

deadlock

Gegenseitiges Blockieren von *Transaktionen*.

Ein Deadlock kann in folgenden Situationen auftreten:

- Datendeadlock: *Transaktionen* blockieren sich gegenseitig bei *konkurrierenden Zugriffen*
- Taskdeadlock: Eine *Transaktion*, die eine Sperre hält, kann diese nicht freigeben, da keine openUTM-Task frei ist. Diese Deadlock-Situation kann nur bei UDS/SQL-openUTM-Zusammenarbeit auftreten.

**Descending-Key (DESC-Key)**

descending key (DESC key)

*Primärschlüssel* eines *Set*. Der Descending-Key legt die Reihenfolge der *Membersätze* in den *Set-Occurrences* nach absteigenden Schlüsselwerten fest.

**direkter Hashbereich**

direct hash area

siehe *Hashbereich*

**direkter Zugriff**

direct access

Zugriff auf einen *Satz* über einen Feldinhalt. UDS/SQL unterstützt den direkten Zugriff über den *Database Key* sowie über *Hashverfahren* und *mehrstufige Tabellen*.

**Distribution Pool**

distribution pool

Kommunikationsbereich des *independent DBH* für die Verständigung von *UDSCT*, *Servertasks*, *Anwendertasks* und *Mastertask* untereinander bezüglich UDS-D-spezifischer Daten. Im Distribution Pool liegen die *Verteiltabelle* und UDS-D-spezifische Systemtabellen.

**DML**

DML

(Data Manipulation Language) Sprachmittel für den Zugriff auf eine UDS/SQL-*Datenbank*.

**Dummy-Teiltransaktion**

dummy subtransaction

Ist eine primäre *Teiltransaktion*, die UDS-D erzeugt, wenn die erste *READY*-Anweisung einer *Transaktion* eine *entfernte Datenbank* anspricht.

Die Dummy-Teiltransaktion dient dazu, die Transaktion in der *lokalen Konfiguration* bekannt zu machen, um im Fehlerfall ein Wiederherstellen der *Datenbank* zu ermöglichen.

**Duplikat-Kopf**

duplicates header

Enthält allgemeine Informationen über eine *Duplikat-Tabelle* bzw. eine *Seite* einer Duplikat-Tabelle:

- die Verkettung zur nächsten und zur vorhergehenden *Überlaufseite*
- die Anzahl freier Bytes in der Seite der Duplikat-Tabelle

**Duplikat-Tabelle**

duplicates table

Spezielle *SEARCH-Key-Tabelle*, in der ein mehrfach auftretender Schlüsselwert nur einmal gespeichert wird.

Die Duplikat-Tabelle enthält pro Schlüsselwert

- einen Tabellenindex-Eintrag mit dem Schlüsselwert und dem Verweis auf die zugehörige Tabellenzeile
- eine Tabellenzeile (DB-Key-Liste), die auf mehrere Seiten aufgeteilt sein kann, mit den *Satzfolgennummern* der *Sätze*, die diesen Schlüsselwert enthalten

**Duplikat-Tabelle, Grundstufe**

duplicates table, main level

Main Level bzw. Level 0; enthält einen Tabellenindex-Eintrag und den Beginn der zugehörigen Tabellenzeile (DB-Key-Liste).

**dynamischer Set**

dynamic set

*Set*, der zeitlich begrenzt durch die Dauer der *Transaktion*, *Membersätze* von Suchfragen aufnehmen kann.

## E

**entfernte Datenbank**

remote database

*Datenbank* einer *entfernten Konfiguration*.



**entfernte Konfiguration**

remote configuration

*DB-Konfigurationen*, die dem *Anwenderprogramm* nicht über */SET-FILE-LINK LINK-NAME=DATABASE,FILE-NAME=konfigurationsname* zugeordnet werden, sondern erst bei Ablauf des *Anwenderprogramms* über die *Verteiltabelle*. Mit entfernten Konfigurationen verkehrt das *Verbindungsmodul* des *Anwenderprogramms* über die *DCAM-Anwendungen*. Entfernte Konfigurationen liegen auf dem *lokalen* oder auf einem *entfernten Verarbeitungsrechner*.

**entfernter Verarbeitungsrechner**

remote host

Verarbeitungsrechner, der nicht lokal ist.

**entferntes Anwenderprogramm**

remote application program

*Anwenderprogramm*, das bezüglich einer bestimmten *Konfiguration* nicht lokal ist.**ESTIMATE-REPORT**

ESTIMATE-REPORT

Protokollausgabe nach dem BGSIA-Lauf. Dient dazu, die Größe der *Benutzer-realms* zu schätzen.**Event-Name**

event name

Name einer Ereigniskennung.

**exklusiver Buffer Pool**

exclusive buffer pool

Puffer, der zusätzlich zu den *System Buffer Pools* ausschließlich für die Pufferung von *Seiten* der angegebenen *Datenbank* verwendet wird.

## F

**Feld**

item

Kleinste benennbare Dateneinheit innerhalb einer *Satzart*. Das Feld ist definiert durch *Feldtyp* und *Feldlänge*.**Folgenummer**

sequence number

siehe *ALOG-Folgenummer*

**FPA**

FPA

siehe *Freiplatzverwaltung*.**FPA-Basis**

FPA base

siehe *Freiplatzverwaltung*.**FPA-Extent**

FPA extent

siehe *Freiplatzverwaltung*.**FPA-Seite**

FPA page

*Seite der Freiplatzverwaltung*.**Freiplatzverwaltung (FPA)**

Free Place Administration (FPA)

Freier Platz wird sowohl auf Realm-Ebene (*FPA-Seiten*), als auch auf Seiten- und Tabellenebene verwaltet. Die Freiplatzverwaltung der Seiten erfolgt in einer Basistabelle (FPA-Basis) und eventuell in einer oder mehreren Erweiterungstabellen (FPA-Extent), welche durch eine Online-Realmerweiterung oder durch BREORG entstehen.

**Fremdschlüssel**

foreign key

*Satzelement*, dessen Werte mit den *Primärschlüssel*werten einer anderen Tabelle (UDS/SQL-*Satzart*) übereinstimmen. Fremdschlüssel im Sinne von UDS/SQL werden im BPSQLSIA-Protokoll in der *Membersatzart* einer *Set-Beziehung* als "REFERENCES owner-satzart" qualifiziert.

**Funktionscode (FC)**

function code

Verschlüsselung einer *DML*-Anweisung. Wird beim *DAL*-Kommando DISPLAY und bei UDSMON ausgegeben.

## H

**Hashbereich**

hash area

Speicherbereich, in dem UDS/SQL Daten speichert oder wiedergewinnt aufgrund der Umrechnung von Schlüsselwerten in relative *Seitennummern*. Ein Hashbereich kann sowohl die Adressen von *Sätzen* als auch die Sätze selbst enthalten.

In einem *direkten Hashbereich* sind die Sätze selbst gespeichert, während in einem *indirekten Hashbereich* die Adressen der andernorts gespeicherten Sätze enthalten sind.

**HASHLIB**

HASHLIB

Modulbibliothek zur Aufnahme der *Hashroutinen* einer *Datenbank*.

**Hashroutine**

hash routine

Modul, das ein *Hashverfahren* ausführt.

**Hashverfahren**

hashing

Methode, mit der ein Schlüsselwert in eine *Seitenadresse* umgerechnet wird.

## I

**Identifizierung**

authorization

Erkennung der Benutzergruppe.

**impliziter Set**

implicit set

*SYSTEM-Set*, den UDS/SQL bildet, wenn ein *SEARCH-Key* auf Satzartebene definiert wird.

**independent DBH**

independent DBH

Selbständiges Programmsystem, das den simultanen Zugriff mehrerer Anwender auf eine *Datenbank (Mono-DB-Betrieb)* oder auf mehrere Datenbanken gleichzeitig (*Multi-DB-Betrieb*) ermöglicht. Der independent DBH ist als Taskfamilie konzipiert:

- eine *Mastertask (UDSSQL)*
- eine oder mehrere *Servertasks (UDSSUB)*
- eine *Administratortask (UDSADM)*

**INDEX-Search-Key**

INDEX search key

*Sekundärschlüssel*. Er wird als *Zugriffspfad* für *direkten Zugriff* über eine *mehrstufige Tabelle* realisiert.

**Indexseite**

index page

*Seite*, in der die höchsten (niedrigsten) Schlüsselwerte der nächstniedrigen Stufe einer indizierten Tabelle gespeichert werden.

**Indexstufe**

index level

Hierarchiestufe einer *Indexseite*.

**indirekter Hashbereich**

indirect hash area

siehe *Hashbereich*

**Inkonsistenz**

inconsistency

Widerspruch zwischen gespeicherten Informationen.

**Integrität**

integrity

Fehlerfreiheit und Vollständigkeit der gespeicherten Informationen

- Objekt-Integrität (Entity Integrity)
- *referentielle Integrität* (Referential Integrity)
- Benutzer-Integrität (User Integrity)

**interne Versionsnummer**

internal version number

Jeder *Realm* der *Datenbank*, inklusive *DBDIR* und *DBC*OM, besitzt eine interne Versionsnummer, die die Dienstprogramme (z. B. BREORG, BALTER) bei Veränderungen des Realms um eins erhöhen. Diese interne Versionsnummer steht in der *Act-Key-0-Seite* des Realms und zusätzlich im PHYS VERSION RECORD im DBDIR.

**Item**

item

siehe *Feld*

## K

**Katalogkennung**

catalog identifier

Bezeichnung der gemeinschaftlichen Platte (Public Volume Set), in der die BS2000-/UDS/SQL-Dateien gespeichert sind. Die Katalogkennung ist Bestandteil des Datenbank-/Datei-Namens und in Doppelpunkte eingeschlossen: „:catid:“.

**KDBS**

KDBS

(Compatible Database Interface) Kompatible Datenbankschnittstelle. KDBS ermöglicht, Programme auf Anwendungen von *Datenbanksystemen* verschiedener Hersteller zu übertragen.

**Kennwort für die UDS/SQL-Dateien**

password for UDS/SQL files

Wort, mit dem die von UDS/SQL eingerichteten Dateien geschützt sind (Standardwert: C'UDS '). Außerdem kann der *Datenbankadministrator* Kennwörter festlegen mit PP CATPASS oder durch MODIFY-FILE-ATTRIBUTES.

**Kette**

chain

siehe *CHAIN***Kommunikationspartner**

communication partners

Tasks bzw. Datensichtstationen

**Komprimierung**

compression

Nur belegte *Felder* eines *Satzes* werden gespeichert (siehe *SSL-Klausel* COMPRESSION).

**Konfiguration**

configuration

siehe *DB-Konfiguration*

**Konfigurationskennung**

configuration user ID

Kennung, in der der *Datenbankadministrator* den *DBH* startet.

**Konfigurationsname**

configuration name

Frei wählbarer Name der *Datenbankkonfiguration* einer *Session*. Aus dem Konfigurationsnamen bildet der *DBH*

- den Namen der *Session-Log-File*,
- den Namen der *DB-Status-Datei* und ihrer Sicherungskopie,
- den Namen der *RLOG-Dateien*,
- den Namen der Temporären *Realms*,
- den Namen der Session-Jobvariablen
- die *Event-Namen* des *PI-Eventing*,
- den Namen der *DCAM-Anwendung* für die Administration,
- die Namen für die *Common Pools*,
- die Namen der *Dump-Dateien*.

**konfigurationsübergreifend**

interconfiguration

Mindestens eine *entfernte Konfiguration* betreffend.

**konfigurationsübergreifende Konsistenz**

interconfiguration consistency

Eine *verteilte Transaktion*, die in mindestens einer *entfernten Konfiguration* geändert hat, wird so beendet, dass die Änderungen entweder auf den *Datenbanken* aller beteiligten *DB-Konfigurationen* durchgeführt werden oder auf keiner *Datenbank*.

Die konfigurationsübergreifende Konsistenz wird sichergestellt durch das *Zwei-Phasen-Ende-Protokoll*.

**konfigurationsübergreifender Deadlock**

interconfiguration deadlock

Zustand wechselseitiger Blockierungen von *verteilten Transaktionen* bei *konkurrierenden Zugriffen*.

**konkurrierender Zugriff**

contending access

Gleichzeitiger Zugriff auf eine *Seite* aus verschiedenen *Transaktionen*.**Konsistenz**

consistency

Widerspruchsfreiheit der gespeicherten Informationen.

**Konsistenz, logische**

consistency, logical

Widerspruchsfreiheit der gespeicherten Daten untereinander und in Bezug auf die Realität.

**Konsistenz, physische**

consistency, physical

Widerspruchsfreiheit der gespeicherten Daten in Bezug auf physisch richtige Speicherung sowie vollständige und richtige *Zugriffspfade* und Beschreibungsinformationen.**Konsistenz, Speicherkonsistenz**

consistency, storage

siehe *physische Konsistenz***Konsistenzfehler**

consistency error

Eine Verletzung der *physischen Konsistenz* der gespeicherten Daten.**Konsistenzpunkt**

consistency point

(Zeit-)Punkt, an dem die *Datenbank* konsistent ist, d.h. alle ändernden *Transaktionen* sind beendet und ihre Änderungen wurden im Datenbestand durchgeführt.**Konsistenzpunkt, festgeschriebener**

checkpoint

Konsistenzpunkt, bei dem die *ALOG-Datei* gewechselt wurde und auf den jederzeit mit Hilfe des Dienstprogramms *BMEND* nachgefahren werden kann**Kopie**

copy

siehe *Datenbankkopie*

**Kopie aktualisieren**

database copy update

*Datenbankkopie* durch Einspielen der *After-Images* auf einen festgeschriebenen *Konsistenzpunkt* vorsetzen.

## L

**Ladeparameter (DBH)**

load parameters (DBH)

Parameter, die der *DBH* beim Starten der *Session* anfordert. Die Parameter definieren die wesentlichen Merkmale einer *Session*.

**Linked-in-Control-System**

linked-in control system

Komponente von UDS/SQL bei *linked-in DBH*, die Steuerungsaufgaben übernimmt (entspricht dem *Subcontrol-System* bei *independent DBH*).

**linked-in DBH**

linked-in DBH

Modul, das in das jeweilige *DB-Anwenderprogramm* eingebunden oder nachgeladen wird und die Zugriffe auf eine *Datenbank (Mono-DB-Betrieb)* oder auf mehrere Datenbanken gleichzeitig (*Multi-DB-Betrieb*) steuert.

**Liste**

list

Tabelle, die die *Membersätze* einer *Set-Occurrence* enthält. Dient zum *sequentiellen* und *direkten Zugriff* auf die *Membersätze*.

Bei einer verteilbaren Liste können die Datenseiten, die die *Membersätze* enthalten (*Stufe-0-Seiten*), über mehrere *Realms* verteilt sein. Die Seiten, die die übergeordneten Tabellenstufen der verteilbaren Liste enthalten, liegen alle in einem *Realm* (*Tabellenrealm* einer verteilbaren Liste).

**Logging**

logging

Protokollierung über alle Änderungen in der *Datenbank*.

**logische Verbindung**

logical connection

Zuordnung zweier *Kommunikationspartner*, die es ihnen ermöglicht, Daten auszutauschen.

*DCAM-Anwendungen* kommunizieren über logische Verbindungen.



**lokale Datenbank**

local database

*Datenbank einer lokalen Konfiguration.*

**lokale Konfiguration**

local configuration

Die *Konfiguration*, die dem *Anwenderprogramm* vor seinem Aufruf mit `/SET-FILE-LINK LINK-NAME=DATABASE,FILE-NAME=konfigurationsname` zugewiesen wurde.

Mit der lokalen Konfiguration verkehrt das Anwenderprogramm über den *Communication Pool*. Die lokale Konfiguration liegt immer im Verarbeitungsrechner des Anwenderprogramms.

**lokale Transaktion**

local transaction

*Transaktion*, die nur auf die *lokale Konfiguration* zugreift.

**lokale Verteiltabelle**

local distribution table

Für einen *DBH* ist die *Verteiltabelle* lokal, die in seinem *Distribution Pool* liegt.

**lokaler Verarbeitungsrechner**

local host

Verarbeitungsrechner, in dem das *Anwenderprogramm* liegt.

**lokales Anwenderprogramm**

local application program

Ein *Anwenderprogramm* ist bezüglich einer *Konfiguration* lokal, wenn es über `/SET-FILE-LINK LINK-NAME=DATABASE,FILE-NAME=konfigurationsname` an sie angeschlossen wurde.

**M****Mainreference**

main reference

Die *Mainreference* dient im *DBH* der Verwaltung der zur Bearbeitung der Aufträge einer Transaktion erforderlichen Ressourcen, einschließlich solcher für die Übertragung der Aufträge vom Anwenderprogramm zum *DBH* und zurück.

**Mainrefnummer**

mainref number

Nummer, die bei *READY* der *Transaktion* zugewiesen wird. Diese Nummer ist nur zu einem Zeitpunkt eindeutig, nach Ende der Transaktion wird sie wieder einer anderen Transaktion zugewiesen.

**Maske**

pattern

Bei der Definition von *Feldern* eine symbolische Darstellung aller möglichen Felddinhalte.

**Maskenzeichenkette**

pattern string

Zeichenfolge, die eine *Maske* definiert.

**Mastertask (MT)**

master task

Task des *independent DBH*, in der das Modul *UDSSQL* abläuft. Steuert das Einleiten und Beenden einer *Session* und kommuniziert direkt oder über die *Administratortask* mit dem *Datenbankadministrator*.

**mehrstufige Tabelle**

multi-level table

*SEARCH-KEY-Tabelle*, die für jeden *Satz* der zugehörigen *Satzart* bzw. für jeden *Membersatz* der zugehörigen *Set-Occurrence* eine Zeile enthält, die aus dem Schlüsselwert des Satzes und aus dem Zeiger zum Satz besteht. Wird auch als Indextabelle bezeichnet.

**Member**

member

siehe *Membersatz* bzw. *Membersatzart*

**Member, AUTOMATIC**

member, AUTOMATIC

Ein *Satz* wird beim Speichern eingehängt.

**Member, MANDATORY**

member, MANDATORY

Ein *Satz* kann nicht ausgehängt werden.

**Member, MANUAL**

member, MANUAL

Der *Satz* wird beim Speichern nicht automatisch eingehängt.

**Member, OPTIONAL**

member, OPTIONAL

Der *Satz* kann ausgehängt werden.**Membersatz**

member record

Untergeordneter *Satz* in einer *Set-Occurrence*.**Membersatzart**

member record type

Untergeordnete *Satzart* in einem *Set*.**Mono-DB-Betrieb**

mono-DB operation

Der *DBH* arbeitet mit nur einer *Datenbank* einer *Konfiguration*.**Mono-DB-Konfiguration**

mono-DB configuration

Nur eine *Datenbank* ist an einer *Session* beteiligt.**Multi-DB-Betrieb**

multi-DB operation

Der *DBH* arbeitet mit mehreren *Datenbanken* einer *Konfiguration*.**Multi-DB-Konfiguration**

multi-DB configuration

Mehrere *Datenbanken* sind an einer *Session* beteiligt.**Multi-DB-Programm**

multi-DB program

*Anwenderprogramm*, das auf mehrere *Datenbanken* zugreift. Die *Datenbanken* können zu einer *Mono-* oder *Multi-DB-Konfiguration* oder zu mehreren *Mono-* oder *Multi-DB-Konfigurationen* gehören.

**Multithreading-Verfahren**

multithreading

Verfahren, durch das der *DBH* die Zentraleinheit (CPU) so intensiv wie möglich nutzen kann.

Im Multithreading-Verfahren bearbeitet der *DBH* parallel mehrere Aufträge unter Verwendung sogenannter *Threads*. In jedem *Thread* sind Informationen über den gegenwärtigen Zustand eines bestimmten Auftrags hinterlegt. Muss ein Auftrag auf den Abschluss eines Eingabe/Ausgabe-Vorgangs warten, nutzt der *DBH* die CPU für die Verarbeitung eines anderen Auftrags.

## N

**Netz**

network

Alle über TRANSDATA gekoppelten Rechner.

**netzweit eindeutig**

unique throughout the network

In allen zu einem *Netz* gehörenden Rechnern eindeutig.

## O

**offene Transaktion**

open transaction

Eine nicht mit FINISH oder mit FINISH WITH CANCEL bzw. COMMIT oder ROLLBACK abgeschlossene *Transaktion*.

**OLTP**

OLTP

(Online Transaction Processing) Bei einer OLTP-Anwendung greift eine sehr große Anzahl von Benutzern auf die gleichen Programme und Daten zu. Dies geschieht in der Regel unter der Steuerung eines Transaktionsmonitors (TP-Monitor)

**Online-DBTT-Erweiterung**

online DBTT extension

Erweiterung der Anzahl der möglichen Sätze einer Satzart im laufenden Datenbankbetrieb. Für die Administration der Online-Erweiterbarkeit von DBTTs stehen die DAL-Kommandos ACT DBTT-INCR, DEACT DBTT-INCR, DISPLAY DBTT-INCR und EXTEND DBTT zur Verfügung.  
Siehe auch *automatische DBTT-Erweiterung*.

**Online-Realm-Erweiterung**

online realm extension

Erweiterung von *Benutzerrealms* und *DBDIR* im laufenden Datenbankbetrieb. Für die Administration der Online-Erweiterbarkeit von Realms stehen die DAL-Kommandos ACT INCR, DEACT INCR, DISPLAY INCR, EXTEND REALM und REACT INCR zur Verfügung.  
Siehe auch *automatische Realm-Erweiterung*.

**Online-Sicherung**

online backup

Wenn AFIM-Logging eingeschaltet ist, kann eine Sicherung der *Datenbank* im laufenden Betrieb erstellt werden. Die Online-Sicherungsfähigkeit einer Datenbank wird mit dem Dienstprogramm BMEND festgelegt.

**Operatortask (OT)**

operator task

siehe *Mastertask***openUTM**

openUTM

(universal transaction monitor) Universeller Transaktionsmonitor. Er ermöglicht die einfache Erstellung und den Betrieb von Transaktionsanwendungen.

**Originaldatenbank**

original database

Der Begriff Originaldatenbank bezieht sich lediglich auf die Namensgebung der Datenbankdateien (*dbname.dbdatei*), nicht auf den inhaltlichen Stand der Datenbank (siehe auch *Schattendatenbank*).

**Owner**

owner

siehe *Ownersatz* bzw. *Ownersatzart***Ownersatz**

owner record

Übergeordneter *Satz* in einer *Set-Occurrence*.**Ownersatzart**

owner record type

Übergeordnete *Satzart* in einem *Set*.

## P

**PETA**

## PETA

(Preliminary End of Transaction) Anweisung bei UDS-D und openUTM-D, die ein vorläufiges Transaktionsende herbeiführt.

Die PETA-Anweisung gehört zur ersten Phase des *Zwei-Phasen-Ende-Protokolls*, das eine *verteilte Transaktion* beendet.

Die Anweisung PETA speichert ausfallsicher in der *RLOG-Datei* des lokalen *DBH*:

- alle geänderten *Seiten*
- die Rücksetz- und Sperrinformationen
- die Namen aller beteiligten *Konfigurationen*

Diese Informationen werden bei einem eventuellen *Warmstart* benötigt.

**POINTER-ARRAY**

## pointer array

siehe *Adressliste*

**PPP**

## PPP

siehe *Probable Position Pointer (PPP)*.

**Prepared to Commit (PTC)**

## prepared to commit (PTC)

Teil des *Zwei-Phasen-Ende-Protokolls*:

Zustand einer *Teiltransaktion* nach Durchführen der *PETA*-Anweisung und vor Erhalt der Nachricht, ob die gesamte *Transaktion* mit FINISH oder mit FINISH WITH CANCEL beendet wird.

**primäre Teiltransaktion (PTT)**

## primary subtransaction

*Teiltransaktion*, die in der *lokalen Konfiguration* abläuft.

Die erste *READY*-Anweisung einer *Transaktion* auf eine *lokale Datenbank* eröffnet die primäre Teiltransaktion.

Falls die erste *READY*-Anweisung eine *entfernte Datenbank* anspricht, erzeugt UDS-D eine sogenannte *Dummy-Teiltransaktion* als primäre Teiltransaktion.

**Primärschlüssel (DDL)**

primary key (DDL)

Der mittels "LOCATION MODE IS CALC" definierte *Schlüssel* einer *Satzart* bzw. der mittels "ORDER IS SORTED [ INDEXED]" definierte ordnungsbestimmende *Schlüssel* einer Set-Occurrence. Dient außerdem zum *Direktzugriff* auf einen *Satz* oder eine Menge von Sätzen mit gleichen Schlüsselwerten oder innerhalb eines Suchintervalls.

**Primärschlüssel (SQL)**

primary key (SQL)

Im weiteren Sinne (SQL) ein *Satzelement*, das einen Datensatz eindeutig identifiziert.

In UDS-SQL der im BPSQLSIA-Protokoll als "PRIMARY KEY" ausgegebene Database Key eines Ownersatzes (siehe auch *Fremdschlüssel*).

Ein einen Datensatz eindeutig identifizierendes *Satzelement* ist im BPSQLSIA-Protokoll als "UNIQUE" ausgewiesen, wenn es sich nicht um den obigen "PRIMARY KEY" handelt.

**PRIVACY-AND-IQF-Schema**

PRIVACY-AND-IQF SCHEMA

UDS/SQL-internes *Schema* für den Zugriffsschutz.

**PRIVACY-AND-IQF-Subschema**

PRIVACY-AND-IQF SUBSCHEMA

UDS/SQL-internes *Subschema* für den Zugriffsschutz.

**Probable Position Pointer (PPP)**

probable position pointer (PPP)

Wahrscheinliche Adresse einer *Seite*, bestehend aus *Realmnummer* und *Seitennummer*. Bei einer Lageänderung von Daten aktualisiert UDS/SQL die zugehörigen Probable Position Pointer (PPP) nicht in jedem Fall.

**Prüfsätze**

check records

Informationselemente zum Prüfen der Datenbank. Sie haben eine variable Länge von 20 bis 271 byte.

**Pubset-Deklaration**

pubset declaration

siehe *UDS/SQL-Pubset-Deklaration*

**Pubset-Deklarations-Jobvariable**

pubset declaration job variable

Jobvariable, in der eine *UDS/SQL-Pubset-Deklaration* vereinbart wird.

**P1-Eventing**

P1 eventing

Verständigung der Tasks untereinander.

## Q

**Quellprogramm**

source program

In einer Programmiersprache formuliertes, noch nicht in die Maschinensprache übersetztes Programm.

## R

**READY**

READY

Beginn einer *Transaktion* oder *Verarbeitungskette* bei *COBOL-DML*-Programmen.**READYC**

READYC

Beginn einer *Transaktion* oder *Verarbeitungskette* bei *CALL-DML*-Programmen.**Realm**

realm

Benennbare physische Untereinheit der *Datenbank*. Der Realm entspricht einer Datei. Außer den *Benutzerrealms* für die Daten gibt es die Realms *DBDIR* und *DBCOM*, die UDS/SQL selbst beansprucht.**Realm-Konfiguration**

realm configuration

Die *Realms* einer *Datenbank*, die an einer *Session* beteiligt sind.**Realm-Kopie**

realm copy

siehe *Datenbankkopie***Realm-Nummer**

realm reference number

*Realms* einer *Datenbank* werden, bei 1 beginnend, aufsteigend und lückenlos nummeriert. Die Realm-Nummer (Area-Reference) ist Bestandteil der *Seitenadresse*.



**RECORD AREA**

RECORD AREA

siehe *Satzbereich***REC-REF**

REC-REF

(Record Reference)

siehe *Satzartnummer***referentielle Integrität**

referential integrity

*Integrität* der Beziehungen zwischen Tabellen (UDS/SQL-*Satzarten*).**Rekonfiguration**

reconfiguration

Neugruppierung von *Datenbanken* in einer *DB-Konfiguration* nach einem *Session-Abbruch*. Voraussetzung für eine Rekonfiguration ist, dass die *SLF* gelöscht oder inhaltlich entwertet wird.**Returncode**

return code

Interner Code eines aufgerufenen Programms an das aufrufende Programm. Returncode ≠ 0: Fehler aufgetreten.

**RLOG-Datei**

RLOG file

Datei zur Ablaufsicherung. In die RLOG-Datei schreibt der *DBH* während der *Session* sowohl Daten vor ihrer Änderung (*Before-Images*) als auch Daten nach ihrer Änderung (*After-Images*). Mit Hilfe der *RLOG-Datei* kann der *DBH* Änderungen nicht abgeschlossener *Transaktionen* zurücksetzen. Es gibt eine RLOG-Datei pro *Konfiguration*. Die RLOG-Datei besteht aus zwei physischen Dateien.**Rollback**

rollback

Rückgängigmachen aller Änderungen einer *Transaktion*.**RSQ**

RSQ

siehe *Satzfolgennummer*.**RUNUNIT-ID**

RUNUNIT-ID

siehe *Transaktionskennung*

## S

**Satz**

record

Einzelne Ausprägung einer *Satzart*. Ein Satz besteht aus je einem Feldinhalt aller am Aufbau der Satzart beteiligten *Felder* und ist die kleinste Dateneinheit, die UDS/SQL über einen eindeutigen Identifizierer, den *Database Key*, verwaltet.

**Satzadresse**

record address

Adresse der *Seite*, in der sich der *Satz* befindet. Siehe *Seitenadresse*.

**Satzart**

record type

Benennbare Zusammenfassung von *Satzelementen*.

**Satzart, lineare**

record type, linear

*Satzart*, die weder *Owner* noch *Member* eines *Set* ist (entspricht Satzarten einer konventionellen Datei).

**Satzartnummer**

record reference number

*Satzarten* werden, bei 1 beginnend, aufsteigend und lückenlos numeriert. Die Satzartnummer ist Bestandteil des *Database Key*.

**Satzbereich**

record area

Vom Benutzer adressierbarer Bereich der *USER-WORK-AREA (UWA)*. Der Satzbereich enthält die *Satzarten* und die implizit definierten Felder (IMPLICITLY-DEFINED-DATA-NAMES) der Datenbank wie z.B. die AREA-ID-Felder der WITHIN-Klauseln des Schemas. Die Länge des Satzbereichs ist wesentlich durch die in ihm definierten Satzarten bestimmt.

**Satzelement**

record element

*Feld*, *Vektor* oder *Datengruppe*.

**Satzfolgenummer**

record sequence number

Der Datenbankprogrammierer kann die Satzfolgenummer vergeben oder UDS/SQL numeriert die *Sätze* einer *Satzart* selbst, bei 1 beginnend, aufsteigend und lückenlos in der Reihenfolge wie die Sätze gespeichert werden. Die Satzfolgenummer ist Bestandteil des *Database Key*.

**Satzhierarchie**

record hierarchy

Owner-/Memberbeziehung zwischen *Satzarten*:  
*Ownersatzart* ist übergeordnet  
*Membersatzart* ist untergeordnet.

**Satz-SEARCH-Key-Tabelle**

record SEARCH KEY table

*SEARCH-Key-Tabelle* für die Auswahl eines *Satzes* aus einer *Satzart*.

**SCD**

SCD

(Set Connection Data) Verknüpfungsinformation für die *Sätze* einer *Set-Occurrence*.

**Schattendatenbank**

backup database

Sicherung sämtlicher Dateien einer *Datenbank* jeweils unter „*dbname.dbdatei.copypname*“.

Die Schattendatenbank kann zu einem beliebigen Zeitpunkt erstellt werden und ist parallel zur Originaldatenbank im Benutzungsmodus RETRIEVAL ablauf-fähig.

Außerdem können die bereits abgeschlossenen *ALOG-Dateien* auf ihr parallel zur UDS/SQL-*Session* mit BMEND nachgefahren werden.

**Schema**

schema

Formalisierte Beschreibung der in der *Datenbank* zugelassenen Datenstruk-turen. Ein UDS/SQL-Schema wird mit der *Schema-DDL* beschrieben.

**Schema-DDL**

Schema DDL

Formale Sprache zur Beschreibung eines *Schemas*.

**Schlüssel**

key

*Feld, das der Datenbankprogrammierer für Direktzugriff auf Sätze benutzt und für das UDS/SQL entsprechend den Angaben im Schema einen optimierten Zugriffspfad anlegt.*

**Schlüssel, zusammengesetzter**

key, compound

*Schlüssel, der aus mehreren Schlüsselfeldern besteht.*

**Schlüsselfeld**

key item

*Feld, das durch Angaben im Schema zum Schlüssel erklärt wird.*

**Schlüsselnummer**

key reference number

*Schlüssel werden, bei 1 beginnend, aufsteigend und lückenlos numeriert.*

**Schnittstelle**

interface

In der Software: Speicherbereich, den mehrere Programme zum Austausch von Daten untereinander verwenden.

**SEARCH-Key**

SEARCH KEY

*Sekundärschlüssel. Zugriffspfade über Sekundärschlüssel realisiert UDS/SQL über Hashverfahren und mehrstufige Tabellen.*

**SEARCH-Key-Tabelle**

SEARCH KEY table

*Mehrstufige Tabelle, die UDS/SQL als Zugriffspfad über einen Sekundärschlüssel benutzt.*

**Seite**

page

Physische Untereinheit von *Realms*. Seiten identifiziert UDS/SQL über eindeutige Schlüssel (*Act-Key*). Die Länge einer Seite kann wahlweise 2048 byte, 4000 byte oder 8096 byte betragen. Innerhalb derselben Datenbank müssen alle Seiten gleich lang sein. Seiten der Länge 4000 byte oder 8096 byte sind in einen *Seitencontainer* eingebettet.

**Seitenadresse**

page address

Bei der Seitenadresse unterscheidet man die aktuelle Adresse einer *Seite*, den *Act-Key*, und die wahrscheinliche Adresse einer Seite, den *Probable Position Pointer (PPP)*.

**Seitencontainer**

page container

Seiten der Länge 4000 byte oder 8096 byte sind jeweils in einen sogenannten Seitencontainer eingebettet. Der Seitencontainer besteht aus einem 64 byte langen Header, der vor der Seite liegt, und einem 32 byte langen Trailer im Anschluss an die Seite.

**Seitenindex-Eintrag**

page index entry

Verweist auf die Position eines *Satzes* innerhalb einer *Seite*.

**Seitenkopf**

page header (page info)

Die ersten 20 byte einer *Seite* (mit Ausnahme der *FPA-Basis-Seiten* und *DBTT-Seiten* der Länge 2048 byte). Sie enthalten

- den *Act-Key* der *Seite* selbst
- die Anzahl der *Seitenindex-Einträge*
- die Länge und Position der in dieser Seite noch freien Bytes
- den Seitentyp (*ACT-Key-0-Seite*, *FPA-Seite*, *DBTT-Seite*, *DBTT-Ankerseite*, allgemeine Datenseite oder *CALC-Seite*)

**Seitennummer**

page number

In jedem *Realm* sind die *Seiten*, bei 0 beginnend, aufsteigend und lückenlos numeriert. Die Seitennummer ist Bestandteil der *Seitenadresse*.

Seitennummer = PAM-Seitennummer-1 bei Datenbanken mit einer Seitenlänge von 2048 byte

Seitennummer = (PAM-Seitennummer-1) / 2 bei Datenbanken mit einer Seitenlänge von 4000 byte

Seitennummer = (PAM-Seitennummer-1) / 4 bei Datenbanken mit einer Seitenlänge von 8096 byte.

**sekundäre Teiltransaktionen**

secondary subtransactions

*Teiltransaktionen*, die *entfernte Konfigurationen* ansprechen.

**Sekundärschlüssel**

secondary key

Jeder *Schlüssel*, der nicht *Primärschlüssel* ist; dient zum *Direktzugriff* auf einen *Satz* oder eine Menge von Sätzen mit gleichen Schlüsselwerten oder innerhalb eines Suchintervalls.

**sequentieller Zugriff**

sequential access

Zugriff auf einen *Satz* aufgrund seiner Position innerhalb einer vorgegebenen Satzreihenfolge.

**Servertask (ST)**

server task

Task des *independent DBH*, in der das Modul *UDSSUB* abläuft. Die Servertask bearbeitet die Anforderungen der *DB-Anwenderprogramme*.

**Session**

session

Zeitraum zwischen dem Starten und dem normalen Beenden des *DBH* (*independent/linked-in DBH*), in dem mit den *Datenbanken* der *Konfiguration* gearbeitet werden kann. Im allgemeinen Fall besteht eine Session aus einer Folge von *Session-Abschnitten* und *Session-Unterbrechungen*.

**Session-Abbruch**

session abort

Liegt vor, wenn der *DBH* nach erfolgreichem *Session-Beginn* abnormal beendet wird.

Ursachen für einen Session-Abbruch können sein: Stromausfall, Rechnerausfall, BS2000-Störung, DBH-Fehler, %TERM.

**Session-Abschnitt**

session section

Beginnt mit dem Starten eines *DBH* entweder bei *Session-Beginn* oder bei *Session-Wiederanlauf* und endet mit dem normalen *Session-Ende* oder mit *Session-Abbruch*.

**Session-Abschnittsnummer**

session section number

Nummer, die einen Session-Abschnitt eindeutig identifiziert.

**Session-Beginn**

session start

Liegt vor, wenn ein *DBH* unter einem *Konfigurationsnamen* gestartet wird, für den noch keine *Session-Log-File (SLF)* mit gültigem Inhalt existiert.

**Session-Ende**

session end

Wird erreicht durch

- *DAL* bei *independent DBH*,
- *TERM* in *DML-Anwenderprogrammen* bei *linked-in DBH*,
- die *DBH-Fehlerbehandlung*.

Während einer *Session-Unterbrechung* kann das *Session-Ende* auch erreicht werden, indem der Anwender die *SLF* inhaltlich entwertet. Bei inkonsistenten *Datenbanken* kann die *Konsistenz* auch ohne *SLF* mit *Warmstart* wiederhergestellt werden.

**Session-Jobvariable**

session job variable

Jobvariable, in der *UDS/SQL* Informationen über eine *Session* hinterlegt.**Session-Log-File (SLF)**

Session Log File (SLF)

Datei, die einer *Session* fest zugeordnet ist und die der *DBH* bei einem eventuellen *Session-Wiederanlauf* benötigt. Sie enthält Informationen über die aktuelle *DB-Konfiguration*, die Menge der aktuellen Dateikennwörter und über die aktuellen Werte der *DBH-Ladeparameter*.

**Session-Unterbrechung**

session interrupt

Zeitraum zwischen einem *Session-Abbruch* und dem zugehörigen *Session-Wiederanlauf*.

**Session-Wiederanlauf**

session restart

Start des *DBH* nach einer abgebrochenen *Session* unter gleichem *Konfigurationsnamen* und in der gleichen *Konfigurationskennung*. Mit Hilfe der *SLF* werden die *DBH-Ladeparameter* und die aktuellen Datei-Kennwörter wiederhergestellt, die bei *Session-Abbruch* vorlagen und die *Datenbanken* der damaligen *Konfiguration* werden ggf. mit *Warmstart* angeschlossen.

**Set**

set

Benennbare Beziehung zwischen zwei *Satzarten*.**Set, dynamischer**

set, dynamic

siehe *dynamischer Set*

**Set, impliziter**

set, implicit

siehe *impliziter Set***Set, singulärer**

set, singular

siehe *SYSTEM-Set***Set, Standard-**

set, standard

siehe *Standard-Set***Setnummer**

set reference number

*Sets* werden, bei 1 beginnend, aufsteigend und lückenlos numeriert.**Set-Occurrence**

set occurrence

Einzelne Ausprägung eines *Set*. Eine Set-Occurrence besteht aus genau einem *Ownersatz* und beliebig vielen ihm untergeordneten *Membersätzen*.**Set-SEARCH-Key-Tabelle**

set SEARCH KEY table

*SEARCH-Key-Tabelle* für die Auswahl eines *Membersatzes* aus einer *Set-Occurrence*.**Shared User Buffer Pool**

Shared User buffer pool

Gemeinsamer Puffer mehrerer Datenbanken, der zusätzlich zu den *System Buffer Pools* ausschließlich für die Pufferung von *Seiten* der ihm zugewiesenen *Datenbanken* verwendet wird.**SF-Pubset**

SF pubset

siehe *Single-Feature-Pubset***SIA**

SIA

(Schema Information Area) Sie enthält die vollständige Datenbankbeschreibung. Der *DBH* lädt die SIA zum Arbeiten generell in den Hauptspeicher.



**SIB**

SIB

(SQL Interface Block) Schnittstelle zwischen UDS/SQL und SQL-Anwenderprogramm(en); enthält die SQL-Anweisung mit eventuell vorhandenen Parametern und das Anweisungsergebnis.

**Single-Feature-Pubset**

single feature pubset

Ein Single-Feature-Pubset (SF-Pubset) besteht aus einer oder mehreren homogenen Platten, die in den wesentlichen Eigenschaften (Plattenformat, Allokierungseinheit) übereinstimmen müssen.

**SLF**

SLF

siehe *Session-Log-File (SLF)*.

**SM-Pubset**

SM pubset

siehe *System-Managed-Pubset*

**Snap-Paar, Snap-Pubset, Snap-Session, Snap-Unit**

snap pair, snap pubset, snap session, snap unit

Eine Snap-Unit ist die Kopie einer (Original-)Unit (logische Platte im BS2000) zu einem bestimmten Zeitpunkt („Point-in-Time-Kopie“). Die Komponente TimeFinder/Snap erstellt diese Kopie als „Snapshot“ nach der „Copy-On-First-Write-Strategie“: Nur wenn Daten geändert werden, werden zuvor die jeweiligen Original-Daten in einen zentralen Speicherbereich (Save-Pool) des Symmetrix-Systems geschrieben. Die Snap-Unit enthält die Verweise (Track-Pointer) auf die Original-Daten. Bei unveränderten Daten zielen die Verweise auf die Unit, bei veränderten auf den Save-Pool.

Nach der Aktivierung sind Unit und Snap-Unit voneinander getrennt, Anwendungen können auf beide zugreifen.

Unit und Snap-Unit bilden zusammen ein Snap-Paar. TimeFinder/Snap verwaltet es in einer sogenannten Snap-Session.

Wenn es zu allen Units eines Pubsets Snap-Units gibt, so bilden diese Snap-Units zusammen das Snap-Pubset.

Details zu diesem Thema finden Sie im Handbuch „[Einführung in die Systembetreuung](#)“.

**Sort-Key-Tabelle**

sort key table

Zusätzlicher *Direktzugriffspfad* mittels des *Primärschlüssels* auf Setebene auf die *Membersätze* einer *Set-Occurrence* bei "MODE IS CHAIN" und "ORDER IS SORTED INDEXED".

**spanned record**

spanned record

*Satz*, der länger ist als eine *Seite*. Spanned records gibt es **nur UDS/SQL-intern**.

Benutzersatzarten dürfen generell nicht länger sein als

- 2020 byte bei 2048 byte Seitenlänge
- 3968 byte bei 4000 byte Seitenlänge
- 8064 byte bei 8096 byte Seitenlänge

**SQL**

SQL

(Structured Query Language) SQL ist eine relationale Datenbanksprache, die von der ISO (International Organization for Standardization) standardisiert worden ist.

**SQL-DML**

SQL-DML

Data Manipulation Language von *SQL*, für die Abfrage und Änderung von Daten.

**SQL-Transaktion**

SQL transaction

Zusammengehörige Folge von *SQL*-Anweisungen, die UDS/SQL entweder ganz oder gar nicht bearbeitet, um die *Datenbank(en)* von einem konsistenten Zustand in einen anderen konsistenten Zustand zu überführen.

**SQL-Vorgang**

SQL conversation

siehe *Vorgang*

**SSIA**

SSIA

(Subschema Information Area) enthält alle Subschema-abhängigen Informationen, die der *Database Handler* benötigt, um für den Anwender auf die *Datenbank* innerhalb der Möglichkeiten des aufgerufenen *Subschemas* zuzugreifen. Der *DBH* lädt die SSIA, sobald sie bei einem *READY* angesprochen wird, in den Hauptspeicher.

**SSIA-RECORD**

SSIA-RECORD

UDS/SQL-interne *Satzart*, die im *Database Directory (DBDIR)* liegt. *Sätze* dieser Satzart sind u.a. die Schema Information Area (*SIA*) und die Subschema Information Areas (*SSIA*).

**SSITAB-Modul**

SSITAB module

Vom Dienstprogramm BCALLSI erzeugtes Modul. Es stellt die Subschema-Informationen für *CALL-DML*-Programme bereit.

**SSL**

SSL

(Storage Structure Language) Formale Sprache zur Beschreibung der Speicherstruktur.

**Standard-Set**

standard set

*Set*, der kein *dynamischer* oder *impliziter Set* oder *SYSTEM-Set* ist.

**Statuscode**

status code

Nummer, die im zweiten Teil des Feldes *DATABASE-STATUS* hinterlegt wird, und die darüber informiert, welcher Sonderzustand aufgetreten ist.

**String**

string

Eine Reihe aufeinanderfolgender alphanumerischer Zeichen.

**Subcontrol-System**

subcontrol system

Komponente des *independent DBH*, die Steuerungsaufgaben übernimmt.

**Subschema**

subschema

Für eine bestimmte *Anwendung* erforderlicher Teil eines *Schemas*, der für eine Anwendung in begrenztem Umfang neu strukturiert werden kann. Das Subschema wird mit der *Subschema-DDL* beschrieben.

**Subschema-DDL**

Subschema DDL

Formale Sprache zur Beschreibung eines *Subschemas*.

**Subschemamodul**

subschema module

Modul, das beim Übersetzen eines *COBOL-DML*-Programms aus der Übersetzung des *Subschemas* entsteht. Es muss in das *Anwenderprogramm* eingebunden werden und enthält die *UWA* sowie die *RECORD AREA*, die gleichzeitig Teil des Base Interface Block (*BIB*) ist. Der Name des Subschemamoduls sind die ersten acht Zeichen des Subschemanamens.

**Subschemasatz**

subschema record

*Satz laut Subschema-DDL.***SUB-SCHEMA SECTION**

SUB-SCHEMA SECTION

Bei einem COBOL-Programm mit *DML*-Anweisungen: Abschnitt in der DATA DIVISION zur Angabe des Schemanamens und des Subschemanamens.

**Subtask (ST)**

subtask

*siehe Servertask.***System Buffer Pools**

system buffer pools

Ein-/Ausgabe-Puffer für Datenbankseiten (siehe *Seite*). Sie liegen im *Common Pool (independent DBH)* bzw. *DBH-Arbeitsbereich (linked-in DBH)*. Ihre Größe bestimmen die *DBH-Ladeparameter* 2KB-BUFFER-SIZE, 4KB-BUFFER-SIZE bzw. 8KB-BUFFER-SIZE.

**Systembereich**

system area

*Realm*, der nur von UDS/SQL benötigt wird. Zu den Systembereichen einer Datenbank zählt man:

- das *Database Directory (DBDIR)*,
- den *Database Compiler Realm (DBCOM)*,
- das *COBOL Subschema Directory (COSSD)*

**Systembreak-Informationen**

system break information

Kennzeichen, ob die *Datenbank* konsistente oder inkonsistente Information enthält.

**System-Managed-Pubset**

system managed pubset

Ein System-Managed-Pubset besteht aus einem oder mehreren Volume-Sets, die wie bei einem *SF-Pubset* eine Zusammenfassung von mehreren homogenen Platten sind; die Homogenität bezieht sich auch hier auf bestimmte physikalische Eigenschaften wie z.B. Plattenformat und Allokierungseinheit.

**SYSTEM-Record**

SYSTEM record

*siehe Ankersatz*

**SYSTEM-Set**

SYSTEM set

*Set*, dessen *Ownersatzart* die symbolische *Satzart* SYSTEM ist.

**T****Tabelle, mehrstufige**

table, multi-level

siehe *mehrstufige Tabelle*

**Tabelle (SQL)**

table (SQL)

Eine Tabelle im *SQL*-Sinn entspricht einer UDS/SQL-*Satzart*.

**Tabellenkopf**

table header

Enthält allgemeine Informationen über eine Tabelle bzw. eine *Tabellenseite*:

- die Angabe über den Tabellentyp und die Stufennummer der Tabellenseite,
- die Anzahl der reservierten und der aktuellen Einträge in dieser Tabellenseite,
- die Verkettung mit weiteren Tabellenseiten der gleichen Stufe,
- den Verweis auf die zugehörige Tabellenseite der nächsthöheren Stufe und
- den Verweis auf die Seite mit der letzten Tabelle der Grundstufe (nur bei der Tabelle der höchsten Stufe).

**Tabellenseite**

table page

*Seite*, die eine Tabelle oder einen Tabellenteil enthält. Handelt es sich um eine *Tabelle*, die sich nicht über mehrere Seiten erstreckt, oder um die höchste Stufe einer mehrstufigen *Tabelle*, so ist mit „Tabellenseite“ nur das entsprechende Objekt gemeint, nicht die ganze *Seite*.

**TANGRAM**

TANGRAM

(Task and Group Affinity Management) Subsystem des BS2000; dieses Subsystem plant für Taskgruppen, die bei Multitask-Anwendungen auf größere gemeinsame Datenmengen zugreifen, die Zuordnung zu den Prozessoren.

**Task Attribut TP**

task attribute TP

Im BS2000 gibt es 4 Task Attribute: SYS, TP, DIALOG und BATCH.

Den Task Attributen sind jeweils spezielle, für das Task-Scheduling wichtige Ablaufparameter zugeordnet.

TP zeichnet sich gegenüber den anderen Task Attributen durch eine, speziell auf die Bedürfnisse des Teilhaberbetriebs optimierte Hauptspeicher-Verwaltung aus.

**Taskdeadlock**

task deadlock

siehe *Deadlock*

**Taskkommunikation**

task communication

Verständigung der *DBH*-Module untereinander. Siehe auch *Common Pool*.

**Taskpriorität**

task priority

Im BS2000 kann die Priorität für eine Task festgelegt werden. Diese Priorität wird bei der Initiierung und Aktivierung der Task berücksichtigt.

Es gibt variable und feste Prioritäten. Variable Prioritäten passen sich an, feste verändern sich nicht.

(UDS/SQL-Servertasks sollen mit einer festen Priorität gestartet werden, um eine gleichbleibende Performance zu erreichen).

**TCUA**

TCUA

(Transaction Currency Area) enthält die Currency-Informationen.

**Teiltransaktion**

subtransaction

In einer verteilten *Transaktion* bilden alle *Verarbeitungsketten*, die *Datenbanken einer Konfiguration* ansprechen, eine Teiltransaktion.

**Transaktion (TA)**

transaction

Zusammengehörige Folge von *DML*-Anweisungen, die UDS/SQL entweder ganz oder gar nicht bearbeitet, um die *Datenbank(en)* von einem konsistenten Zustand in einen anderen konsistenten Zustand zu überführen.

Bei UDS-D:

Gesamtheit aller zu einem Zeitpunkt gestarteten *Teiltransaktionen*.

**Transaktion normal beenden**

transaction, committing a

Eine *Transaktion* mit FINISH beenden, d.h. alle Änderungen festschreiben, die auf den *Datenbanken* gemacht wurden.

**Transaktion zurücksetzen**

transaction, rolling back a

Eine *Transaktion* mit FINISH WITH CANCEL beenden, d.h. alle Änderungen rückgängig machen, die auf den *Datenbanken* gemacht wurden.

**Transaktionskennung**

transaction identification (TA-ID)

Vergibt der *DBH* zur Kennzeichnung einer *Transaktion*; kann mit dem *DAL*-Kommando DISPLAY erfragt werden.

**Transfer Pool**

transfer pool

UDS-D-spezifischer Speicherbereich, in dem der *UDSCT* die *BIBs* von *entfernten Anwenderprogrammen* empfängt.

**UDSADM**

UDSADM

Modul des *independent DBH*. Das Modul läuft in der *Administratortask* ab.

**UDSHASH**

UDSHASH

Vom Dienstprogramm BGSIA erzeugtes Modul mit den Namen aller *Hashroutinen*, die in der *Schema-DDL* definiert wurden.

**UDSNET**

UDSNET

Verteilkomponente in der *Anwendertask*.

## U

**UDSSQL**

UDSSQL

Startmodul des *independent DBH*. Das Modul läuft in der *Mastertask* ab.

**UDSSUB**

UDSSUB

Startmodul des *independent DBH*. Das Modul läuft in der *Servertask* ab.

**UDS-D-Task UDSCT**

UDS-D task UDSCT

Task, die UDS/SQL für jede *Konfiguration* startet, damit sie an der verteilten Verarbeitung mit UDS-D teilnehmen kann.

**UDS/SQL / openUTM-D-Konsistenz**

UDS/SQL / openUTM-D consistency

Eine *Transaktion*, die sowohl *openUTM*-Daten als auch UDS/SQL-Datenbanken geändert hat, wird so beendet, dass entweder die openUTM-Daten und die UDS/SQL-Datenbanken geändert werden, oder keines von beiden.

**UDS/SQL-Pubset-Deklaration**

UDS/SQL pubset declaration

Vereinbarung in einer *Pubset-Deklarations-Jobvariable* zur Einschränkung der UDS/SQL-Pubset-Umgebung. Dadurch wird die Gefahr durch die Mehrdeutigkeit von Dateinamen verringert bzw. vermieden.

**Überlaufseite**

overflow page

*Seite* bei *Hashbereichen* und *Duplikat-Tabellen*, die diejenigen Daten aufnimmt, die nicht mehr in die Primärseite passen. Ihr Aufbau entspricht den Seiten des Hashbereichs bzw. der Duplikat-Tabelle.

**Umstrukturierung**

restructuring

Änderung von *Schema-DDL* oder *SSL* bei *Datenbanken*, in denen bereits Daten gespeichert sind.

**USER-WORK-AREA (UWA)**

USER-WORK-AREA (UWA)

Übergabebereich zur Kommunikation zwischen *Anwenderprogramm* und *DBH*.

**UTM**

UTM

siehe openUTM.

**UWA**

UWA

siehe *USER-WORK-AREA (UWA)*.



## V

**Vektor**

vector

*Feld* mit Wiederholungsfaktor. Der Wiederholungsfaktor muss größer als 1 sein. Er gibt an, wieviel Duplikate des Feldes zu dem Vektor zusammengefasst werden.

**Verarbeitungskette**

processing chain

Folge von *DML*-Anweisungen an eine *Datenbank* innerhalb einer *Transaktion*.

**Verbindungsmodul**

connection module

Modul, das in jedes *UDS/SQL-Anwenderprogramm* eingebunden werden muss und die Verbindung zum *DBH* herstellt.

**Versionsnummer, interne**

version number, internal

siehe *interne Versionsnummer*

**Verteiltabelle**

distribution table

Tabelle, die *UDS-D* anhand der zugewiesenen Eingabedatei im *Distribution Pool* aufbaut. Mit Hilfe der Verteiltabelle entscheidet die Verteilkomponente in der *Anwendertask*, ob eine *Verarbeitungskette* lokal oder entfernt bearbeitet werden soll.

In der Verteiltabelle ist zugeordnet:

*Subschema* - *Datenbank*

*Datenbank* - *Konfiguration*

*Konfiguration* - *Verarbeitungsrechner*.

**verteilte Datenbanken**

distributed database

Ein logisch zusammengehörender Datenbestand, der auf mehrere *UDS/SQL*-Konfigurationen verteilt ist.

**verteilte Transaktion**

distributed transaction

*Transaktion*, die auf mindestens eine *entfernte Konfiguration* zugreift.

Eine Transaktion kann verteilt sein über:

- UDS-D,
- openUTM-D,
- UDS-D und openUTM-D.

**Vorgang**

conversation

In einer *Anwendung* mit *SQL* werden *SQL*-spezifische Verwaltungsdaten über Transaktionsgrenzen hinweg aufbewahrt. Eine solche Verwaltungseinheit wird als Vorgang bezeichnet.

## W

**Warmstart (einer DB)**

warm start

Ein Warmstart wird von UDS/SQL durchgeführt, wenn eine inkonsistente *Datenbank* an eine *Session* angeschlossen wird. Ein Warmstart umfasst das Nachfahren der Änderungen abgeschlossener *Transaktionen*, die noch nicht auf der Datenbank festgeschrieben waren, den *Rollback* aller auf der Datenbank offenen Transaktionen und das Konsistentmachen der Datenbank. Für einen Warmstart wird die zugehörige *RLOG-Datei* benötigt und die *DB-Status-Datei*.

**Wiederanlauf (von BMEND)**

restart of BMEND

Fortsetzung eines abgebrochenen BMEND-Laufs.

**Wiederanlauf (einer Session)**

restart of a session

siehe *Session-Wiederanlauf*

**Wiederholungsgruppe**

repeating group

*Datengruppe* mit Wiederholungsfaktor. Der Wiederholungsfaktor muss größer als 1 sein. Er gibt an, wieviele Duplikate der Datengruppe zu der Wiederholungsgruppe zusammengefasst werden.

## Z

**Zeitquittung**

time acknowledgment

Nachrichten, die die *UDS-D-Task* zum entfernten *Anwenderprogramm* sendet, um mitzuteilen, dass noch eine *DML*-Anweisung bearbeitet wird.

**Zugriff, direkter**

access, direct

siehe *direkter Zugriff*

**Zugriff, konkurrierender**

access, contending

siehe *konkurrierender Zugriff*

**Zugriff, sequentieller**

access, sequential

siehe *sequentieller Zugriff*

**Zugriffsart**

access type

Art und Weise des Zugriffs, zum Beispiel Lesen, Ändern usw.

**Zugriffsberechtigte**

authorized users

Festgelegte Benutzergruppen und deren Benutzer, die auf die *Datenbank* zugreifen dürfen.

**Zugriffsberechtigung**

access authorization

Recht einer definierten Benutzergruppe in definierter Weise auf die *Datenbank* zuzugreifen. Die Zugriffsberechtigung wird im laufenden Datenbankbetrieb mit dem Dienstprogramm *ONLINE-PRIVACY* bzw. im Offline-Modus mit dem Dienstprogramm *BPRIVACY* festgelegt.

**Zugriffspfad**

access path

Hilfsmittel, um eine bestimmte, durch eine Suchfrage qualifizierte Untermenge aller *Sätze* auffinden zu können, ohne die ganze *Datenbank* sequentiell absuchen zu müssen.

**Zugriffsrechte**

access rights

Zugriffsrechte werden durch das Dienstprogramm BPRIVACY festgelegt. Sie regeln den Zugriff auf die *Datenbank*.

**Zustand PTC**

PTC state

siehe *Prepared to Commit*

**Zwei-Phasen-Ende-Protokoll**

two-phase commit protocol

Verfahren, um eine *verteilte Transaktion*, die in mindestens einer *entfernten Konfiguration* geändert hat, so zu beenden, dass die *konfigurationsübergreifende Konsistenz* bzw. die UDS/SQL-/openUTM-D-Konsistenz gesichert ist.

Das Zwei-Phasen-Ende-Protokoll wird gesteuert:

- von der Verteilkomponente in der *Anwendertask*, wenn die *Transaktion* über UDS-D verteilt ist.
- von openUTM-D, wenn die *Transaktion* über openUTM-D bzw. über openUTM-D und über UDS-D verteilt ist.

---

# Abkürzungen

ACS	Alias Catalog Service
Act-Key	Actual-Key
AFIM	After-Image
AP	Anwenderprogramm, Application Program
ASC	Ascending
BIB	Base Interface Block
BFIM	Before-Image
COBOL	Common Business Oriented Language
CODASYL	Conference on Data System Languages
CRA	Current Record of Area
CRR	Current Record of Record
CRS	Current Record of Set
CRU	Current Record of Rununit
COSSD	COBOL Subschema Directory
DAL	Database Administration Language
DB	Datenbank
DBCOM	Database Compiler Realm
DBDIR	Database Directory
DBH	Database Handler
DB-Key	Database Key
DBTT	Database Key Translation Table
DDL	Data Description Language
DESC	Descending
DML	Data Manipulation Language
DRV	Dual Recording by Volume
DSA	Database System Access
DSSM	Dynamische Verwaltung von Subsystemen

FC	Function Code
FPA	Free Place Administration
GS	Global Store (Globalspeicher)
HSMS	Hierarchisches Speicher Management System
ID	Identification (Kennung)
IMON	Installation Monitor
IQL	Interactive Query Language
IQS	Interactive Query System
KDBS	Kompatible Datenbank-Schnittstelle
KDCS	Kompatible Datenkommunikations-Schnittstelle
LM	Lock Manager
LMS	Library Maintenance System
MPVS	Multiple Public Volume Set
MR-NR	Mainref-Number
MT	Mastertask
OLTP	Online Transaction Processing
openUTM	Universeller Transaktionsmonitor
OT	Operatortask
PETA	Preliminary End of Transaction
PPP	Probable Position Pointer
PTC	Prepared to Commit
PTT	Primäre Teiltransaktion
PVS	Public Volume Set
REC-REF	Record-Reference
RSQ	Record-Sequence-Number (Satzfolgenummer)
SC	Subcontrol
SCD	Set Connection Data
SCI	Software Configuration Inventory
SECOLTP	Secure Online Transaction Processing
SECOS	Security Control System
SET-REF	Set-Reference
SIA	Schema Information Area
SIB	SQL Interface Block

SLF	Session-Log-File
SQL	Structured Query Language
SSD	Solid State Disk
SSIA	Subschema Information Area
SSITAB	Subschema Information Table
SSL	Storage Structure Language
ST	Servertask
STT	Sekundäre Teiltransaktion
TA	Transaction
TA-ID	Transaction-Identification
TANGRAM	Task and Group Affinity Management
TCUA	Transaction Currency Area
UDS/SQL	Universelles Datenbanksystem/Structured Query Language
UWA	User Work Area





---

# Literatur

Die Handbücher finden Sie im Internet unter <http://manuals.ts.fujitsu.com>. Handbücher, die mit einer Bestellnummer angezeigt werden, können Sie auch in gedruckter Form bestellen.

**UDS/SQL (BS2000)**  
**Anwendungen programmieren**  
Benutzerhandbuch

**UDS/SQL (BS2000)**  
**Aufbauen und Umstrukturieren**  
Benutzerhandbuch

**UDS/SQL (BS2000)**  
**Datenbankbetrieb**  
Benutzerhandbuch

**UDS/SQL (BS2000)**  
**Entwerfen und Definieren**  
Benutzerhandbuch

**UDS/SQL (BS2000)**  
**Meldungen**  
Benutzerhandbuch

**UDS/SQL (BS2000)**  
**Taschenbuch**

**UDS (BS2000)**  
**Dialogsystem IQS**  
Benutzerhandbuch

**UDS-KDBS (BS2000)**  
**Kompatible Datenbankschnittstelle**  
Benutzerhandbuch

**SQL für UDS/SQL**  
Sprachbeschreibung

**BS2000 OSD/BC**

**Kommandos**

Benutzerhandbuch

**BS2000 OSD/BC**

**Einführung in die Systembetreuung**

Benutzerhandbuch

**BS2000 OSD/BC**

**Makroaufrufe an den Ablaufteil**

Benutzerhandbuch

**BS2000 OSD/BC**

**Einführung in das DVS**

Benutzerhandbuch

**SDF (BS2000)**

**Dialogschnittstelle SDF**

Benutzerhandbuch

**SORT (BS2000)**

Benutzerhandbuch

**SPACEOPT (BS2000)**

**Optimierung und Reorganisation von Platten**

Benutzerhandbuch

**LMS (BS2000)**

**SDF-Format**

Benutzerhandbuch

**DSSM/SSCM**

**Verwaltung von Subsystemen in BS2000**

Benutzerhandbuch

**ARCHIVE (BS2000)**

Benutzerhandbuch

**DRV (BS2000)**

**Dual Recording by Volume**

Benutzerhandbuch

**HSMS / HSMS-SV** (BS2000)  
**Hierarchisches Speicher Management System**  
**Band 1: Funktionen, Verwaltung und Installation**  
Benutzerhandbuch

**SECOS** (BS2000)  
**Security Control System**  
Benutzerhandbuch

**openNet Server** (BS2000)  
**BCAM**  
Referenzhandbuch

**DCAM** (BS2000)  
**Programmschnittstellen**  
Beschreibung

**DCAM** (BS2000)  
**Makroaufrufe**  
Benutzerhandbuch

**OMNIS/OMNIS-MENU** (BS2000)  
**Funktionen und Kommandos**  
Benutzerhandbuch

**OMNIS/OMNIS-MENU** (BS2000)  
**Administration und Programmierung**  
Benutzerhandbuch

**openUTM**  
**Konzepte und Funktionen**  
Benutzerhandbuch

**openUTM**  
**Anwendungen programmieren mit KDCS für COBOL, C und C++**  
Benutzerhandbuch

**openUTM**  
**Anwendungen generieren**  
Benutzerhandbuch

**openUTM**  
**Anwendungen administrieren**  
Benutzerhandbuch

**openUTM**

**Einsatz von openUTM-Anwendungen unter BS2000**

Benutzerhandbuch

**openUTM**

**Meldungen, Test und Diagnose (BS2000)**

Benutzerhandbuch

**COBOL2000 (BS2000)**

**COBOL-Compiler**

Sprachbeschreibung

**COBOL2000 (BS2000)**

**COBOL-Compiler**

Benutzerhandbuch

**COBOL85 (BS2000)**

**COBOL-Compiler**

Beschreibung

**COBOL85 (BS2000)**

**COBOL-Compiler**

Benutzerhandbuch

**CRTE (BS2000)**

**Common Runtime Environment**

Benutzerhandbuch

**DRIVE/WINDOWS (BS2000)**

Programmiersystem

Benutzerhandbuch

**DRIVE/WINDOWS (BS2000)**

Programmiersprache

Sprachbeschreibung

**DRIVE/WINDOWS (BS2000)**

Lexikon der DRIVE-Anweisungen

Referenzhandbuch

**DRIVE/WINDOWS (BS2000/SINIX)**

Lexikon der DRIVE-SQL-Anweisungen für UDS

Referenzhandbuch

**DAB** (BS2000)  
**Disk Access Buffer**  
Benutzerhandbuch

**XHCS** (BS2000)  
8-bit-Code- und Unicode-Unterstützung im BS2000  
Benutzerhandbuch

**Unicode im BS2000**  
Übersichtshandbuch

**BS2000 OSD/BC**  
**Softbooks Deutsch**  
CD-ROM

**openSM2** (BS2000)  
**Software Monitor**  
Benutzerhandbuch

**SNMP Management** (BS2000)  
Benutzerhandbuch



---

# Stichwörter

## A

- abbauen Überlaufseiten 334, 335
- Ablaufmeldungen BCHECK 105
- abschalten Realm 43
- Act-Key 361
- Act-Key-0-Seite 361
  - prüfen 70
- Act-Key-N-Seite 361
- ACTKEY-Format
  - siehe Tabellenkopf
- ADD-Anweisung
  - UDS-Online-Utility 295
- ADD-REALM-Anweisung BMEND 36
- Administratortask 361
- Adresse
  - Hashbereich 154, 163
  - Tabelle 161
- Adresse, physisch 362
- Adressliste 362
- Adressverweise reorganisieren 334, 335
- AFIM 362
- AFIM-Logging 321
- AFIMs einspielen 60
- After-Image 362
  - ALOG-Datei 362
  - RLOG-Datei 362
- After-Image-Datei 321
- aktualisieren
  - Jobvariable 68
  - Kopie 384
- ALIAS-Feld, Länge 178
- ALOG-Datei 362
  - After-Image 362
- ALOG-Folgenummer 362
- alog-seq-nr 24
- Alphanumerischer Datentyp (SQL) 184
- ändern
  - Realm-Größe 325
  - Satzmengengerüst 328
- angeben
  - auszugebenden Realm 242
  - Schema 346
  - Subschema 347
- Ankersatz 158, 363
  - Database Key 158
- Anweisung rückgängig machen 59, 348
- Anweisungen
  - BCHECK 82
  - BMEND 35
  - BMODTT 357
  - BPRECORD 239
  - BPSIA 139
  - BPSQLSIA 189
  - BREORG 322
  - BSTATUS 210
- Anweisungscode 363
- Anweisungsfolge BPRECORD 239
- Anwenderprogramm 363
  - entfernt 377
  - lokal 385
- Anwendertask 363
- Anwendung 363
- Anzahl
  - DBTT-Einträge 152
  - DBTT-Extents 152, 223
  - gespeicherte Sätze 217, 223, 230
  - Primärseiten 154
  - Primärseiten, Hashbereich 163, 225
  - pro Realm, Satzart 148
  - pro Satzart, Schlüssel 169

- Anzahl (Forts.)
    - pro Set, Schlüssel 177
    - pro Subschema, Realm 165
    - Schlüsselfeld 180
    - Überlaufseiten, Hashbereich 226
  - Anzahl der Indexstufen
    - Set-Tabellen 217
    - Tabellen 217, 220
  - appl 24
  - Arbeitsdateien
    - BCHECK 79
    - BREORG 317
    - BSTATUS 208
  - Area 363
  - AREA INFORMATION 146, 167
  - Ascending-Key 363
  - auflisten
    - DBTT-Einträge 251
    - FPA-Einträge 250
  - Auftrag 363
  - Auftragsschalter
    - BCHECK 136
    - BREORG 321
  - ausdrucken
    - CALC-Key-Statistik 224
    - CALC-Seite 253
    - Datenseiten 258
    - Hashbereich 224, 256
    - Kennseite 246
    - Owner-Statistik 218
    - Realm-Statistik 212
    - Satzanzahl-Statistik 228
    - Satzart-Statistik 221
    - Schema-Information-Area 139, 142
    - Set-Statistik 215
    - Subschema-Information-Area 142
    - Tabelle 260
  - Ausgabe
    - BPSQLSIA 193
    - im CSV-Format 139, 140, 212, 215, 218, 221, 224, 228, 246, 250, 251, 253, 258
  - ausgeben Logging-Informationen 44
  - ausschalten Logging 41, 58
  - auswählen Subschemas 191
  - auszugebenden Realm angeben 242
  - AUTOMATIC Member 386
  - automatische DBTT-Erweiterung 364
  - automatische Realm-Erweiterung 364
- ## B
- Base Interface Block 364
  - BCHECK 69
    - Ablaufmeldungen 105
    - Anweisungen 82
    - Arbeitsdateien 79
    - Auftragsschalter 136
    - Beispiele 97
    - CHECK-Anweisung 84
    - Fehlermeldungen 104
    - KEY-Anweisung 94
    - Kommandofolge 96
    - Konsistenzkriterien 85
    - Meldungen 103
    - Meldungen zu Konsistenzfehlern 110
    - REALM-Anweisung 88
    - RECORD-Anweisung 89
    - SCHEMA-Anweisung 87
    - Schlüsselwertprüfung 89, 90
    - SCRTCH1-Datei 79
    - SET-Anweisung 92
    - SORTCORE-Anweisung 83
    - SORTWK-Datei 79
    - Systemumgebung 74
    - TYPE-Anweisung 85
    - Warnungen 104
  - Bedingungen
    - UDS-Online-Utility 306
  - beenden Eingabe 40, 189, 324
  - Before-Image 364
  - Beispiele
    - BCHECK 97
    - BREORG 351
    - UDS-Online-Utility 307
  - belegte DBTT-Einträge 223
  - Belegung Datenbankseiten 213
  - Benutzerdatenbank 364
  - Benutzerrealm 364
    - prüfen 69



- Beschreibung
  - DBTT-Einträge 251
  - Tabelle 261
- bezeichnen
  - Schema 87, 241
  - Subschema 211
- Bezeichner 365
- BFIM 365
- BIB (Base Interface Block) 365
- bilden Prüfsätze 70
- BMEND
  - ADD-REALM-Anweisung 36
  - ALLOCATE-BUFFER-POOL-Anweisung 37
  - Anweisungen 35
  - DISABLE-ONLINE-COPY-Anweisung 38
  - ENABLE-ONLINE-COPY-Anweisung 39
  - END-Anweisung 40
  - Funktionen 32
  - Jobvariable 64
  - KILL-LOG-Anweisung 41
  - Kommandofolge 63
  - OPEN-DATABASE-Anweisung 42
  - Regeln 33
  - REMOVE-REALM-Anweisung 43
  - SHOW-LOG-INFORMATION-Anweisung 44
  - START-LOG-Anweisung 50
  - STOP-LOG-Anweisung 58
  - UNDO-Anweisung 59
  - UPDATE-DATABASE-Anweisung 60
- BMEND, Wiederanlauf 410
- BMODTT 355
  - Anweisungen 357
  - KEEP-Anweisung 357
  - Kommandofolge 359
  - REMOVE-Anweisung 357
  - RESET-Anweisung 358
  - REUSE-Anweisung 357
  - SET-Anweisung 358
  - Systemumgebung 356
- BNR-Format
  - siehe Tabellenkopf
- BPRECORD 233
  - Anweisungen 239
  - Anweisungsfolge 239
- DISPLAY CALC-Anweisung 253
- DISPLAY DATA-Anweisung 258
- DISPLAY DBTT-Anweisung 251
- DISPLAY FPA-Anweisung 250
- DISPLAY PAGE ZERO-Anweisung 246
- Kommandofolge 262
- PRINT-Anweisung 243
- REALM-Anweisung 242
- SCHEMA-Anweisung 241
- Systemumgebung 236
- BPSIA 137
  - Anweisungen 139
  - DISPLAY-SCHEMA-Anweisung 139
  - DISPLAY-SUBSCHEMA-Anweisung 140, 230, 261
  - Kommandofolge 141
  - Protokoll 137
  - Systemumgebung 138
- BPSQLSIA
  - Anweisungen 189
  - Ausgabe 193
  - Dienstprogramm 181
  - END-Anweisung 189
  - Kommandofolge 192
  - OPEN-DATABASE-Anweisung 190
  - PRINT-RELATIONAL-SCHEMAINFO-Anweisung 191
  - Systemumgebung 182
  - Umsetzregeln 194
- BREAK-Anweisung
  - UDS-Online-Utility 296
- BREORG 314
  - ALLOCATE-BUFFERPOOL-Anweisung 323
  - Anweisungen 322
  - Arbeitsdateien 317
  - Auftragsschalter 321
  - Beispiele 351
  - END-Anweisung 324
  - Funktionen 314
  - Kommandofolge 349
  - MODIFY-REALM-SIZE-Anweisung 325
  - MODIFY-RECORD-POPULATION-Anweisung 328
  - OPEN-DATABASE-Anweisung 331

### BREORG (Forts.)

- REORGANIZE-CALC-Anweisung 332
- REORGANIZE-POINTERS-Anweisung 339
- REORGANIZE-SET-Anweisung 340
- SPECIFY-SCHEMA-Anweisung 346
- SPECIFY-SUBSCHEMA-Anweisung 347
- Systemumgebung 315
- UNDO-Anweisung 348

### BSTATUS 204

- Anweisungen 210
- Arbeitsdateien 208
- DISPLAY CALC-Anweisung 224
- DISPLAY REALM-Anweisung 212
- DISPLAY RECORD-Anweisung 221
- DISPLAY RECORDNUMBER-Anweisung 228
- DISPLAY TABLE FOR OWNER-Anweisung 218
- DISPLAY TABLE FOR SET-Anweisung 215
- Kommandofolge 231
- SUBSCHEMA-Anweisung 211
- Systemumgebung 208

### Buffer Pools

- siehe System Buffer Pools

## C

- c-string 24
- CALC INFORMATION 153
- CALC KEY INFORMATION 170
- CALC-Bereiche reorganisieren 332, 353
- CALC-Key 153, 162, 170, 175, 178, 224, 225, 238, 365
  - Länge 153, 162, 170, 178
  - Statistik 224
  - Statistik ausdrucken 224
  - Tabelle 162, 257
- CALC-Key-Statistik 206
- CALC-SEARCH-Key 253, 365
- CALC-SEARCH-KEY-INFORMATION 162
- CALC-Seite 256, 365
  - ausdrucken 253
  - prüfen 70
- CALC-Tabelle 366
- CALC-Tabellenzeile 70

### CALL-DML 366

- catid 24
- CHAIN 366
- Character Separated Values (CSV) 366
- CHECK-Anweisung BCHECK 84
- Check-Table 366
- Clone 367
- COBOL Subschema Directory 367
- COBOL-DML 367
- COBOL-Laufzeitsystem 367
- CODASYL-Subschema 181
- Common Memory 367
- Common Pool 367
- Communication Pool 368
- COMPILER-SCHEMA 368
- COMPILER-SUBSCHEMA 368
- Compilerdatenbank 368
  - prüfen 69
- Compound Key 368
- Connectionmodul 368
- Consistency Record 368
- copyname 24
- COSSD 367, 368
- CRA 369
- CRR 369
- CRS 369
- CRU 369
- CSV 369
- CSV-Format
  - Ausgabe 139, 140, 212, 215, 218, 221, 224, 228, 246, 250, 251, 253, 258
  - BPRECORD 246, 250, 251, 253, 258
  - BPSIA 139, 140
  - BSTATUS 212, 215, 218, 221, 224, 228
- Currency-Tabelle 167, 369
  - allgemein 165
  - Länge 165
- CURRENT-OF-AREA-Tabelle 369
- CURRENT-OF-RECORD-Tabelle 369
- CURRENT-OF-SET-Tabelle 369

## D

- DAL 370
- dal-cmd 24

- Database Compiler Realm 370
- Database Directory 370
- Database Key
  - Ankersatz 158
  - Feld 169
- DATABASE-KEY-Feld 370
- Database-Key-Feld 370
- DATABASE-KEY-LONG-Feld 370
- DATABASE-STATUS 370
- date 24
- Datenbank 371
  - entfernt 376
  - eröffnen 42, 190, 331
  - lokal 385
  - Name 164
  - Original prüfen 72
  - verteilt 409
- Datenbank-Jobvariable 371
- Datenbankadministrator 371
- Datenbankbetrieb SHARED-RETRIEVAL 74
- Datenbankdateien, Prüfläufe 75
- Datenbankkopie 371
- Datenbankseite 146, 371
  - Belegung 213
  - Länge 144
- Datenbanksystem 371
- Datenbankzustand 371
- Datendeadlock 372
- Datengruppe 372
- Datensatz 172, 244
- Datenschutz 372
- Datenseiten ausdrucken 258
- Datenseiten verlagern 266
- Datensicherung 372
- Datentypen 24
  - SQL 184
  - Zusätze 27
- DB-Key 168, 373
- DB-Konfiguration 373
- DB-Status-Datei 373
- DBCOM 372
- DBDIR 372
- DBH 372
  - independent 372, 380
  - Ladeparameter 372, 384
  - linked-in 372, 384
- DBH-Ende 373
- DBH-Start 373
- dbname 24
- DBTT 373
  - Füllgrad 223
  - Größe 223
  - Spaltennummer 161, 217, 219
- DBTT INFORMATION 151
- DBTT-Ankerseite 374
- DBTT-Ankertabelle 152
- DBTT-Basis 374
- DBTT-Einträge
  - Anzahl 152
  - auflisten 251
  - belegte 223
  - Beschreibung 251
  - Länge 152
- DBTT-Erweiterung
  - automatisch 364
  - online 388
- DBTT-Extent 374
- DBTT-Extents
  - Anzahl 152, 223
- DBTT-Seite 374
- DCAM 374
- DCAM-Anwendung 374
- DDL 374
- Deadlock 375
  - konfigurationsübergreifend 382
- DECLARE-PROCEDURE-Anweisung
  - UDS-Online-Utility 275
- DECLARE-VARIABLE-Anweisung
  - UDS-Online-Utility 276
- DELETE-PROCEDURE-Anweisung
  - UDS-Online-Utility 277
- DELETE-VARIABLE-Anweisung
  - UDS-Online-Utility 277
- Descending-Key 375
- device 24
- Dienstprogramm BPSQLSIA 181

- direkt
    - Hashbereich [375](#)
    - Zugriff [375](#), [411](#)
  - DISABLE-ONLINE-COPY-Anweisung
    - BMEND [38](#)
  - DISPLAY CALC-Anweisung
    - BPRECORD [253](#)
    - BSTATUS [224](#)
  - DISPLAY DATA-Anweisung
    - BPRECORD [258](#)
  - DISPLAY DBTT-Anweisung
    - BPRECORD [251](#)
  - DISPLAY FPA-Anweisung
    - BPRECORD [250](#)
  - DISPLAY PAGE ZERO-Anweisung
    - BPRECORD [246](#)
  - DISPLAY REALM-Anweisung
    - BSTATUS [212](#)
  - DISPLAY RECORD-Anweisung
    - BSTATUS [221](#)
  - DISPLAY RECORDNUMBER-Anweisung
    - BSTATUS [228](#)
  - DISPLAY TABLE FOR OWNER-Anweisung
    - BSTATUS [218](#)
  - DISPLAY TABLE FOR SET-Anweisung
    - BSTATUS [215](#)
  - DISPLAY-Anweisung
    - Ausgabe im CSV-Format [139](#), [140](#), [212](#), [215](#), [218](#), [221](#), [224](#), [228](#), [246](#), [250](#), [251](#), [253](#), [258](#)
  - DISPLAY-SCHEMA-Anweisung
    - BPSIA [139](#)
  - DISPLAY-SUBSCHEMA-Anweisung
    - BPSIA [140](#)
  - Distanzen zu den Sätzen prüfen [70](#)
  - Distribution Pool [375](#)
  - DML [375](#)
    - FPASCAN [269](#)
    - PREFRLM [269](#)
    - RELOCATE,RELOCATE [266](#)
    - REORGPPP [270](#)
  - DML FPASCAN
    - Aktuell gültige Parameter ausgeben [291](#)
  - DML RELOCATE
    - Aktuell gültige Parameter ausgeben [292](#)
  - DML REORGPPP
    - Aktuell gültige Parameter ausgeben [292](#)
  - Dummy-Teiltransaktion [375](#)
  - Duplikat-Kopf [376](#)
  - Duplikat-Tabelle [160](#), [261](#), [376](#)
    - Grundstufe [376](#)
  - Duplikate [153](#), [160](#)
  - dynamischer Set [156](#), [158](#), [376](#), [399](#)
- ## E
- Eigenschaften eines RELOCATE-DML festlegen [283](#)
  - Eingabe beenden [40](#), [189](#), [324](#)
  - einschalten Logging [50](#)
  - einspielen AFIMs [60](#)
  - Eintrag Tabelle [260](#)
  - Einträge auflisten, DBTT [251](#)
  - einzelne Realms prüfen [69](#)
  - einzelne Satzarten prüfen [69](#)
  - einzelne Sets prüfen [69](#)
  - ENABLE-ONLINE-COPY-Anweisung
    - BMEND [39](#)
  - END-Anweisung
    - BMEND [40](#)
    - BPSIA [140](#), [261](#)
    - BPSQLSIA [189](#)
    - BREORG [324](#)
    - BSTATUS [230](#)
    - UDS-Online-Utility [277](#), [296](#)
  - entfernt
    - Anwenderprogramm [377](#)
    - Datenbank [376](#)
    - Konfiguration [377](#)
    - Verarbeitungsrechner [377](#)
  - eröffnen Datenbank [42](#), [190](#), [331](#)
  - Erstellungszeitpunkt Realm [248](#)
  - Erweiterbarkeit, Satzart [223](#)
  - ESTIMATE-REPORT [377](#)
  - Event-Name [377](#)
  - EXIT-Anweisung
    - UDS-Online-Utility [296](#)
  - exklusiver Buffer Pool [377](#)

EXTENSIBILITY 223

## F

Fachwörter 361

Fehlerbehandlung

UDS-Online-Utility 303

Fehlermeldungen BCHECK 104

Feld 377

Database Key 169

Länge 180

variables 173

Feldtyp 170, 178, 180

festlegen

Puffergröße 37, 323

Sortierpuffer 83

Umfang der Ausgaben 243

FINISH-Anweisung

UDS-Online-Utility 297

First Scan

Freiplatzsuche 147, 269, 280, 297

Folgenummer 377

Format, Subschema 170

FPA 378

FPA-Basis 247, 378

FPA-Einträge auflisten 250

FPA-Extent 146, 247, 326, 378

FPA-Pegel in einem Realm festlegen 280

FPASCAN 269

FPASCAN-Anweisung

UDS-Online-Utility 297

Free Place Administration 144, 146, 251

freie Seiten 213

freier Platz, Hashbereich 226

Freiplatzsuche

First Scan 147, 269, 280, 297

Second Scan 147, 269, 280, 297

Suchmodus 269

Freiplatzverwaltung 146, 378

Fremdschlüssel 378

Füllgrad

DBTT 223

Hashbereich 226

Seiten 213

Set-Tabellen 217, 220

Tabelle 217, 220

Überlaufseiten 226

Funktionen

UDS-Online-Utility 264

Funktionen BREORG 314

Funktionscode 378

## G

gespeicherte Sätze, Anzahl 217, 223, 230

globale Konsistenzprüfung

Kriterien 85

Größe

DBTT 223

Realm 214

Größenberechnung

Hashbereich 336

SORTWK-Datei 209

Grundstufe Duplikat-Tabelle 376

## H

Hashbereich 153, 225, 379

Adresse 154, 163

Anzahl Primärseiten 163, 225

Anzahl Überlaufseiten 226

ausdrucken 224, 256

direkt 375

freier Platz 226

Füllgrad 226

Größenberechnung 336

indirekt 380

Zugriff 226

HASHLIB 379

Hashroutine 379

Hashverfahren 379

host 25

## I

Identifizierung 379

impliziter Set 150, 156, 379, 400

independent DBH 372, 380

INDEX-Search-Key 380

Indexseite 380

Indexstufe 380

indirekter Hashbereich 380

Inkonsistenz [380](#)  
Inkrementalprüfung [72, 74, 77](#)  
integer [25](#)  
Integrität [380](#)  
    referentielle [393](#)  
intern  
    Ergebnisse nutzen, summarische  
        Prüfung [81](#)  
    Realm Version [248](#)  
    Versionsnummer [381, 409](#)  
item [381](#)  
Item-String [172](#)  
    Länge [173](#)

## J

Jobvariable  
    aktualisieren [68](#)  
    BMEND [64](#)  
    Struktur [65](#)

## K

Katalogkennung [381](#)  
KDBS [381](#)  
KEEP-Anweisung  
    BMODTT [357](#)  
Kennseite ausdrucken [246](#)  
Kennwort [381](#)  
Kette [381](#)  
KEY INFORMATION [159, 179](#)  
KEY ITEM LIST [174](#)  
KEY-Anweisung BCHECK [94](#)  
KILL-LOG-Anweisung BMEND [41](#)  
Kohärenzprüfung [73, 74](#)  
Kommandofolge  
    BCHECK [96](#)  
    BMEND [63](#)  
    BMODTT [359](#)  
    BPRECORD [262](#)  
    BPSIA [141](#)  
    BPSQLSIA [192](#)  
    BREORG [349](#)  
    BSTATUS [231](#)  
Kommunikationspartner [381](#)  
Komprimierung [149, 382](#)

Konfiguration [382](#)  
    entfernt [377](#)  
    lokal [385](#)  
Konfigurationskennung [382](#)  
Konfigurationsname [382](#)  
konfigurationsübergreifend [382](#)  
    Deadlock [382](#)  
    Konsistenz [382](#)  
konkurrierender Zugriff [383, 411](#)  
Konsistenz [383](#)  
    der Datenbank prüfen [69](#)  
    konfigurationsübergreifend [382](#)  
    logisch [383](#)  
    physisch [383](#)  
    physische, prüfen [69](#)  
    Speicherkonsistenz [383](#)  
Konsistenzfehler [383](#)  
Konsistenzkriterien  
    BCHECK [85](#)  
    wählen [85](#)  
Konsistenzpunkt [383](#)  
    festgeschriebener [383](#)  
Kopf, Tabelle [260](#)  
Kopie [383](#)  
    aktualisieren [384](#)  
Kriterien  
    globale Konsistenzprüfung [85](#)  
kset [25](#)

## L

Ladeparameter DBH [372, 384](#)  
Länge  
    ALIAS-Feld [178](#)  
    CALC-Key [153, 162, 170, 178](#)  
    Currency-Tabelle [165](#)  
    Currency-Tabellen, allgemein [165](#)  
    Datenbankseiten [144](#)  
    DBTT-Einträge [152](#)  
    Feld [180](#)  
    Item-String [173](#)  
    Satzart [144, 149](#)  
    Schlüssel [144, 153, 162, 175](#)  
    Schlüssel Feld [160, 180](#)  
    Set-Connection-Data [158](#)

Länge (Forts.)  
   SSIA 165  
 lineare Satzart 394  
 linked-in DBH 372, 384  
 Linked-in-Control-System 384  
 Liste 384  
   verteilbar, siehe verteilbare Liste  
 LOCATION MODE 149, 177  
 Logging 384  
   ausschalten 41, 58  
   einschalten 50  
 Logging-Informationen ausgeben 44  
 logisch  
   Konsistenz 383  
   Verbindung 384  
 lokal  
   Anwenderprogramm 385  
   Datenbank 385  
   Konfiguration 385  
   Transaktion 385  
   Verarbeitungsrechner 385  
   Verteiltabelle 385

**M**

Mainreference 385  
 Mainrefnummer 386  
 MANDATORY Member 386  
 MANUAL Member 386  
 markieren Online-Sicherungsfähigkeit 39  
 Maske 386  
 Maskenzeichenkette 386  
 Mastertask 386  
 mehrstufige Tabelle 386, 405  
 Meldungen BCHECK 103  
 Meldungen zu Konsistenzfehlern, BCHECK 110  
 Member 386  
   AUTOMATIC 386  
   MANDATORY 386  
   MANUAL 386  
   OPTIONAL 387  
 Membersatz 387  
 Membersatzart 158, 387  
 Membersätze, zusätzliche Verkettung 158  
 Metasprache 20

Metasyntax, SDF-Anweisungen 22  
 Mitgliedschaft Set 156  
 MODIFY-REALM-SIZE-Anweisung  
   BREORG 325  
 MODIFY-RECORD-POPULATION  
   BREORG 328  
 Mono-DB-Betrieb 387  
 Mono-DB-Konfiguration 387  
 MOVE-Anweisung  
   UDS-Online-Utility 298  
 Multi-DB-Betrieb 387  
 Multi-DB-Konfiguration 387  
 Multi-DB-Programm 387  
 Multithreading-Verfahren 387

**N**

Name  
   Datenbank 164  
   Realm 213, 242  
   Satzart 149, 256  
   Satzarten 230  
   Schema 141, 142, 241  
   Set 156, 256  
   Subschema 141, 164, 165  
 name 25  
 nationale Felder 258  
 Nationaler Datentyp (SQL) 184  
 Netz 388  
 netzweit eindeutig 388  
 normal beenden Transaktion 407  
 Numerischer Datentyp (SQL) 185  
 Nummer  
   Realm 143, 145, 146, 152, 161, 165, 230,  
   238  
   Satzart 143, 148, 149, 152, 168, 256  
   Schlüssel 143, 160, 162, 166, 175, 180, 256  
   Set 143, 156, 160, 162, 179

**O**

offene Transaktion 388  
 Offline-Prüflauf 74  
 Online-DBTT-Erweiterung 388  
 Online-Prüflauf 74  
 Online-Realm-Erweiterung 388

Online-Sicherung [389](#)  
Online-Sicherungsfähigkeit  
  markieren [39](#)  
  zurücknehmen [38](#)  
ONLINE-UTILITY-Parameter festlegen [281](#)  
OPEN-DATABASE-Anweisung  
  BMEND [42](#)  
  BPSQLSIA [190](#)  
  BREORG [331](#)  
openUTM [389](#)  
Operatortask (OT) [389](#)  
OPTIONAL Member [387](#)  
Original prüfen, Datenbank [72](#)  
Originaldatenbank [389](#)  
Originaldatenbank prüfen [72](#)  
Originalrealms, Totalprüfung [75](#)  
Owner [389](#)  
Owner-Statistik [205](#)  
  ausdrucken [218](#)  
Ownersatz [389](#)  
Ownersatzart [158](#), [389](#)

## P

P1-Eventing [392](#)  
PAGE INDEX [245](#), [256](#)  
PAGE INFO [245](#), [256](#)  
parallele Prüfläufe [74](#)  
PETA [390](#)  
physisch  
  Adresse [362](#)  
  Konsistenz [383](#)  
physische Konsistenz prüfen [69](#)  
physische Strukturen prüfen [69](#)  
POINTER-ARRAY [390](#)  
PPP (Probable Position Pointer) [154](#), [161](#), [162](#),  
  [313](#), [314](#), [335](#), [339](#), [390](#), [391](#)  
PPP-Reorganisation  
  Ende [272](#)  
  Fortschrittsdaten [270](#)  
  gesperrte Seiten [272](#)  
  Initialisierung [271](#)  
  Prozedur [270](#)

Preferred-Realm  
  ausgeben [292](#)  
  festlegen [282](#)  
  verteilbare Liste [157](#), [263](#), [264](#), [269](#), [274](#),  
    [282](#), [294](#), [298](#)  
PREFRLM [269](#)  
PREFRLM-Anweisung  
  UDS-Online-Utility [298](#)  
Prepared to Commit (PTC) [390](#)  
primäre Teiltransaktion [390](#)  
Primärschlüssel (DDL) [363](#), [375](#), [391](#), [398](#)  
Primärschlüssel (SQL) [391](#)  
Primärseiten, Anzahl [154](#)  
PRINT-Anweisung BPRECORDER [243](#)  
PRINT-RELATIONAL-SCHEMAINFO-Anweisung  
  BPSQLSIA [191](#)  
PRIVACY-AND-IQF-Schema [391](#)  
PRIVACY-AND-IQF-Subschema [391](#)  
Probable Position Pointer  
  reorganisieren [270](#)  
Probable Position Pointer (PPP) [154](#), [161](#), [162](#),  
  [313](#), [314](#), [335](#), [339](#), [390](#), [391](#)  
  reorganisieren [339](#)  
Probable Position Pointers  
  aktualisieren [264](#)  
Protokoll BPSIA [137](#)  
Prozedur ausführen [278](#)  
Prozedur ausgeben  
  UDS-Online-Utility [291](#)  
Prozedur definieren [275](#)  
Prozedur löschen [277](#)  
Prozedur-Anweisungen  
  UDS-Online-Utility [294](#)  
prüfen  
  Act-Key-0-Seite [70](#)  
  Benutzerrealms [69](#)  
  CALC-Seiten [70](#)  
  Compilerdatenbank [69](#)  
  Distanzen zu den Sätzen [70](#)  
  Konsistenz der Datenbank [69](#)  
  Konsistenz, physische [69](#)  
  Originaldatenbank [72](#)  
  physische Strukturen [69](#)  
  PRIVACY-AND-IQF-Datenbank [69](#)



## prüfen (Forts.)

- Realms, einzelne 69
- Satzarten, einzelne 69
- Satzfolgennummern Sortierfolge 70
- Schattendatenbank 72
- SEARCH-Keys, einzelne 69
- Seitenkopf 70
- Seitenstrukturen der Realms 70
- Sets, einzelne 69
- Sortierfolge der Schlüssel 70
- Systemdaten 69
- Tabellenseiten 70

## Prüfläufe

- Datenbankdateien 75
- Offline 74
- Online 74
- parallele 74

## Prüfmodus 70

- einstellen 70
- wählen 84

## Prüfsatz 391

## Prüfsätze bilden 70

## Prüfung

- inkremental 72
- Kohärenz 73, 74
- Sortierung 70
- summarische 70, 71
- total 72

## Prüfverfahren 71

## Pubset-Deklaration 32, 74, 138, 182, 207, 235, 314, 356, 391

## Pubset-Deklarations-Jobvariable 391

## Puffergröße festlegen 37, 323

**Q**

## Quellpegel 267

## Quellprogramm 392

**R**

## Readme-Datei 17

## READY 392

## READY UPDATE-Anweisung

- UDS-Online-Utility 299

## READYC 392

## Realm 392

- abschalten 43
- Anzahl pro Subschema 165
- Erstellungszeitpunkt 248
- Größe 214
- Name 213, 242
- Nummer 143, 145, 146, 152, 161, 165, 230, 238
- Statistik 204, 212
- Zeitpunkt der letzten Änderung 248
- zuschalten 36

## REALM-Anweisung

- BCHECK 88
- BPRECORD 242

## Realm-Erweiterung

- automatisch 364
- online 388

## Realm-Größe ändern 325

## Realm-Konfiguration 392

## Realm-Kopie 392

## Realm-Name 146

## Realm-Nummer 145, 154, 392

## Realm-Statistik 213

- ausdrucken 212

## Realm-Version, interne 248

## realmname 25, 36, 43, 44, 60, 88, 212, 225, 228, 238, 242, 325, 332, 358

## realmref 25, 238

## Realms

- einzelne, prüfen 69
- zu prüfende, spezifizieren 88

## REC-REF (Record Reference) 393

## RECORD AREA 169, 177, 393

## RECORD INFORMATION 149, 168

## RECORD WITHIN LIST 148

## RECORD-Anweisung BCHECK 89

## recordname 25, 328, 332, 341

## recordref 25

## REFERENCE NUMBERS 145, 166

## referentielle Integrität 393

## Regeln BMEND 33

## Rekonfiguration 393

- relational
    - Schemainformation 181
    - Zugriff 181
  - RELOCATE-Anweisung
    - UDS-Online-Utility 299
  - REMARK-Anweisung
    - UDS-Online-Utility 300
  - REMOVE-Anweisung BMODTT 357
  - REMOVE-REALM-Anweisung
    - BMEND 43
  - reorganisieren 313
    - Adressverweise 334, 335
    - CALC-Bereiche 332, 353
    - Probable Position Pointer 270
    - Probable Position Pointer (PPP) 339
    - Set-Konstrukte 340
    - Set-Tabellen 220
    - Tabellen 217, 220, 340, 342
  - REORGANIZE-CALC-Anweisung
    - BREORG 332
  - REORGANIZE-POINTERS-Anweisung
    - BREORG 339
  - REORGANIZE-SET-Anweisung
    - BREORG 340
  - REORGPPP 264, 270
  - REORGPPP DML
    - Eigenschaften definieren 288
  - REORGPPP-Anweisung
    - UDS-Online-Utility 300
  - REPEAT-PROCEDURE-Anweisung
    - UDS-Online-Utility 278
  - RESET-Anweisung BMODTT 358
  - Returncode 393
  - REUSE-Anweisung BMODTT 357
  - RLOG-Datei 393
    - After-Image 362
  - Rollback 393
  - RSQ 393
  - rsq-Auswahl 240
  - rückgängig machen, Anweisung 59, 348
  - RUNUNIT-ID 393
- S**
- S-Variable
    - SHOW-VARIABLE-Anweisung 293, 301
  - Satz 394
  - Satz-SEARCH-Key-Tabelle 395
  - Satzadresse 394
  - Satzanzahl-Statistik 206, 228
    - ausdrucken 228
  - Satzart 145, 394
    - Anzahl pro Realm 148
    - Erweiterbarkeit 223
    - Länge 144, 149
    - linear 394
    - Name 149, 152, 256
    - Nummer 143, 148, 149, 152, 168, 256
    - Schemaformat 170, 177, 180
    - Subschemaformat 170, 177, 180
  - Satzart-Statistik 205, 221
    - ausdrucken 221
  - Satzarten
    - einzelne, prüfen 69
    - Name 230
    - spezifizieren, zu prüfende 89
  - Satzartnummer 394
  - Satzbereich 394
  - Satzelement 394
  - Satzfolgennummer 152, 223, 240, 251, 395
  - Satzfolgennummern Sortierfolge
    - prüfen 70
  - Satzhierarchie 395
  - Satzmengengerüst ändern 328
  - SCD 395
  - Schattendatenbank 395
    - prüfen 72
    - Totalprüfung 76
  - Schema 395
    - angeben 346
    - bezeichnen 87, 241
    - Name 141, 142, 164, 165, 241
  - SCHEMA-Anweisung
    - BCHECK 87
    - BPRECORD 241
  - Schema-DDL 395

- Schema-Information-Area 137, 142
  - ausdrucken 139, 142
- Schemaformat 170, 172
  - Satzart 170, 177, 180
- Schemainformation, relationale 181
- schemaname 25
- Schlüssel 396
  - Anzahl pro Satzart 169
  - Anzahl pro Set 177
  - Länge 144, 153, 162, 175
  - Nummer 143, 160, 162, 166, 175, 180, 256
  - Subschema 179
  - zusammengesetzt 396
- Schlüsselfeld 396
  - Anzahl 180
  - Länge 160, 180
  - Typ 180
- Schlüsselnummer 396
- Schlüsselwertprüfung
  - BCHECK 89, 90
- Schlüsselwort 20
- Schnittstelle 396
- SCRATCH1-Datei
  - BCHECK 79
  - BREORG 317
  - BSTATUS 208
- SDF-Anweisungen
  - UDS-Online-Utility 274
- SDF-Anweisungen, Metasyntax 22
- SEARCH-Key 396
- SEARCH-Key-Tabelle 396
- SEARCH-Keys
  - einzelne, prüfen 69
  - spezifizieren, zu prüfende 94
- Second Scan
  - Freiplatzsuche 147, 269, 280, 297
- Seite 396
- Seiten
  - freie 213
  - Füllgrad 213
- Seitenadresse 397
- Seitenauswahl 240
- Seitencontainer 397
- Seitenindex-Eintrag 243, 244, 245, 256, 397
- Seitenkopf 244, 245, 256, 397
  - prüfen 70
- Seitennummer 146, 152, 158, 240, 261, 397
- Seitenstrukturen der Realms prüfen 70
- sekundär
  - Teiltransaktion 397
- Sekundärschlüssel 398
- sequenzieller Zugriff 398, 411
- Servertask 398
- Session 398
- Session-Abbruch 398
- Session-Abschnitt 398
- Session-Abschnittsnummer 398
- Session-Beginn 398
- Session-Ende 399
- Session-Jobvariable 399
- Session-Log-File (SLF) 399
- Session-Unterbrechung 399
- Session-Wiederanlauf 399, 410
- Set 399
  - dynamisch 156, 158, 376, 399
  - implizit 150, 156, 379, 400
  - Mitgliedschaft 156
  - Name 156, 256
  - Nummer 143, 156, 160, 162, 179
  - singulär 156, 400
  - Sortierfolge 156
  - Standard 400, 403
  - Statistik 205, 215, 217
  - Typ 156
- SET INFORMATION 155, 176
- SET OCCURRENCE SELECTION 177
- SET-Anweisung
  - BCHECK 92
  - BMODTT 358
- Set-Connection-Data 149, 158, 161, 180, 243, 244, 245
  - Länge 158
- SET-FPA-SCAN-PARAMETERS-Anweisung
  - UDS-Online-Utility 280
- Set-Konstrukte reorganisieren 340
- Set-Modus 156
- Set-Occurrence 400
- Set-Occurrence-Selection 156

- SET-ONLINE-UTILITY-PARAMETERS-Anweisung
  - UDS-Online-Utility [281](#)
- SET-PREF-REALM-PARAMETERS-Anweisung
  - UDS-Online-Utility [282](#)
- SET-RELOCATE-PARAMETERS-Anweisung
  - UDS-Online-Utility [283](#)
- SET-REORGANIZE-PPP-PARAMETERS-Anweisung
  - UDS-Online-Utility [288](#)
- Set-SEARCH-Key-Tabelle [400](#)
- Set-Statistik ausdrucken [215](#)
- Set-Tabellen
  - Anzahl der Indexstufen [217](#)
  - Füllgrad [217](#), [220](#)
  - reorganisieren [220](#)
- setname [26](#)
- Setnummer [400](#)
- Sets
  - einzelne, prüfen [69](#)
  - zu prüfende, spezifizieren [92](#)
- SF-Pubset [400](#)
- Shared User Buffer Pool [400](#)
- SHARED-RETRIEVAL
  - Datenbankbetrieb [74](#)
- SHOW-Anweisung
  - UDS-Online-Utility [301](#)
- SHOW-FPA-SCAN-PARAMETERS-Anweisung
  - UDS-Online-Utility [291](#)
- SHOW-LOG-INFORMATION-Anweisung
  - BMEND [44](#)
- SHOW-PREF-REALM-PARAMETERS-Anweisung
  - UDS-Online-Utility [292](#)
- SHOW-PROCEDURE-Anweisung
  - UDS-Online-Utility [291](#)
- SHOW-RELOCATE-PARAMETERS-Anweisung
  - UDS-Online-Utility [292](#)
- SHOW-REORGANIZE-PPP-PARAMETERS-Anweisung
  - UDS-Online-Utility [292](#)
- SHOW-VARIABLE-Anweisung
  - S-Variable [293](#), [301](#)
  - UDS-Online-Utility [293](#)
- SIA [400](#)
- SIA VALIDATION DATE [165](#)
- SIA-PRINT-REPORT [142](#)
- SIA-Protokoll [142](#), [145](#)
- SIB (SQL Interface Block) [401](#)
- Sicherung [321](#)
- Single-Feature-Pubset [401](#)
- singulärer Set [156](#), [400](#)
- SM-Pubset [401](#)
- Snap [401](#)
- Sort-Key-Tabelle [401](#)
- SORTCORE-Anweisung BCHECK [83](#)
- Sortierfolge der Schlüssel prüfen [70](#)
- Sortierfolge Set [156](#)
- Sortierpuffer festlegen [83](#)
- Sortierungsprüfung [70](#), [71](#)
- Sortierverfahren [71](#)
- SORTWK-Datei
  - BCHECK [79](#)
  - BSTATUS [209](#)
  - Größenberechnung [209](#)
- Spaltennummer DBTT [161](#), [217](#), [219](#)
- spanned records [402](#)
- SPECIFY-SCHEMA-Anweisung
  - BREORG [346](#)
- SPECIFY-SUBSCHEMA-Anweisung
  - BREORG [347](#)
- Speicherkonsistenz
  - Konsistenz [383](#)
- spezifizieren
  - zu prüfende Realms [88](#)
  - zu prüfende Satzarten [89](#)
  - zu prüfende SEARCH-Keys [94](#)
  - zu prüfende Sets [92](#)
- SQL [402](#)
- SQL-Datentypen [184](#)
- SQL-DML [402](#)
- SQL-Schnittstelle [181](#)
- SQL-Transaktion [402](#)
- SQL-Vorgang [402](#)
- SQL-Zugriffe pro Basistabelle, zugelassene [196](#)
- SSIA [402](#)
  - Länge [165](#)
- SSIA PRINT REPORT [164](#)

- SSIA-RECORD 402
  - SSITAB-Modul 403
  - SSL 403
  - Standard-Set 400, 403
  - Standardprozeduren
    - UDS-Online-Utility 305
  - START-LOG-Anweisung BMEND 50
  - Starten
    - UDS-Online-Utilitx 273
  - Statistik
    - CALC-Key 224
    - Owner 205
    - Realm 204, 212
    - Satzanzahl 206, 228
    - Satzart 221
    - Set 205, 215, 217
  - Status-Codes
    - UDS-Online-Utility 305
  - Statuscode 403
  - STOP-LOG-Anweisung BMEND 58
  - String 173, 403
  - structured-name (Datentyp) 26
  - Struktur Jobvariable 65
  - Strukturierter Datentyp (SQL) 186
  - SUB-SCHEMA SECTION 404
  - Subcontrol-System 403
  - Subschema 403
    - angeben 347
    - bezeichnen 211
    - Format 170
    - Name 141
    - Schlüssel 179
  - Subschema Information Area 164
  - SUBSCHEMA-Anweisung 211
    - BSTATUS 211
  - Subschema-DDL 403
  - Subschema-Information-Area 137
    - ausdrucken 142
  - SubschemafORMAT Satzart 170, 177, 180
  - Subschemamodul 403
  - subschemaname 26
  - Subschemasatz 404
  - Subschemata auswählen 191
  - Subtask, siehe Servertask
  - Suchmodus
    - für Freiplatzsuche 269
    - summarische Prüfung 70, 71
      - interne Ergebnisse nutzen 81
    - Syntaxbeschreibung 22
    - System Buffer Pools 404
    - System-Managed-Pubset 404
    - SYSTEM-Record 404
    - SYSTEM-Set 405
    - Systembereich 404
    - Systembreak-Informationen 248, 404
    - Systemdaten prüfen 69
    - Systemumgebung
      - BCHECK 74
      - BMODTT 356
      - BPRECORD 236
      - BPSIA 138
      - BPSQLSIA 182
      - BREORG 315
      - BSTATUS 208
- ## T
- Tabelle
    - Adresse 161
    - ausdrucken 260
    - Beschreibung 261
    - CALC-Key 162, 257
    - Eintrag 260
    - Füllgrad 217, 220
    - Kopf 260
  - Tabelle (SQL) 405
  - Tabelle, mehrstufig 386, 405
  - Tabellen
    - Anzahl der Indexstufen 217, 220
    - reorganisieren 217, 220, 340, 342
  - Tabellenkopf 405
    - ACTKEY-Format 261
    - BNR-Format 261
  - Tabellenrealm
    - verteilbare Liste 156, 384
  - Tabellenseite 405
  - Tabellenseiten prüfen 70
  - Tabellenteil
    - verteilbare Liste 156

- TANGRAM 405
- Task Attribut TP 406
- Task Currency Area 164, 167, 168, 177
- Taskdeadlock 406
- Taskkommunikation 406
- Taskpriorität 406
- TCUA 406
- Teiltransaktion 406
  - primär 390
  - sekundär 397
- time 26
- Totalprüfung 72, 74
  - Originalrealms 75
  - Schattendatenbank 76
- Transaction Currency Area 143
- Transaktion 406
  - lokal 385
  - normal beenden 407
  - offen 388
  - verteilt 410
  - zurücksetzen 407
- Transaktionskennung 407
- Transfer Pool 407
- Typ
  - Schlüsselfeld 180
  - Set 156
- TYPE-Anweisung BCHECK 85
- U**
- Überlaufseite 408
- Überlaufseiten
  - abbauen 334, 335
  - Anzahl 226
  - Füllgrad 226
- UDS-D-Task UDSCT 408
- UDS-Online-Utility
  - Bedingungen 306
  - Beispiele 307
  - DECLARE-PROCEDURE-Anweisung 275
  - DECLARE-VARIABLE-Anweisung 276
  - DELETE-PROCEDURE-Anweisung 277
  - DELETE-VARIABLE-Anweisung 277
  - Fehlerbehandlung 303
  - Funktionen 264
  - Prozedur ausgeben 291
  - Prozedur-Anweisungen 294
  - REPEAT-PROCEDURE-Anweisung 278
  - SDF-Anweisungen 274
  - SET\_FPA-SCAN-PARAMETERS 280
  - SET-ONLINE-UTILITY-PARAMETERS 281
  - SET-PREF-REALM-PARAMETERS 282
  - SET-RELOCATE-PARAMETERS 283
  - SET-REORGANIZE-PPP-PARAMETERS 288
  - SHOW-FPA-SCAN-PARAMETERS 291
  - SHOW-PREF-REALM-PARAMETERS 292
  - SHOW-PROCEDURE 291
  - SHOW-RELOCATE-PARAMETERS 292
  - SHOW-REORGANIZE-PPP-PARAMETERS 292
  - SHOW-VARIABLE 293
  - Standardprozeduren 305
  - Starten 273
  - Status-Codes 305
  - Vordefinierte Variable 304
  - Wert einer Variablen ausgeben 293
- UDS-Online-Utility beenden 277
- UDS-Online-Utility
  - END-Anweisung 277
- UDS/openUTM-D-Konsistenz 408
- UDS/SQL-Pubset-Deklaration 32, 74, 138, 182, 207, 235, 314, 356, 408
- UDSCT
  - UDS-D-Task 408
- UDSHASH 407
- UDSNET 407
- UDSSQL 407
- UDSSUB 407
- Umfang der Ausgaben festlegen 243
- Umsetzregeln BPSQLSIA 194
- Umstrukturierung 408
- UNDO-Anweisung
  - BMEND 59
  - BREORG 348
- Unicode 258
- UPDATE-DATABASE-Anweisung BMEND 60
- USER-WORK-AREA (UWA) 168, 408
- userid 26

UTM siehe openUTM

UWA [408](#)

## V

Variable [20](#)

Wert ausgeben [293](#)

Variable definieren [276](#)

Variable löschen [277](#)

variables Feld [173](#)

Vektor [409](#)

Verarbeitungskette [409](#)

Verarbeitungsrechner

entfernt [377](#)

lokal [385](#)

Verbindung

logisch [384](#)

Verbindungsmodul [409](#)

Verlagern

Datenseiten [266](#)

Versionsnummer, intern [381](#), [409](#)

verteilbare Liste [91](#), [93](#), [95](#), [156](#), [157](#), [263](#), [264](#),  
[266](#), [269](#), [274](#), [282](#), [286](#), [287](#), [294](#), [298](#), [384](#)

Preferred-Realm [157](#), [263](#), [264](#), [269](#), [274](#),  
[282](#), [294](#), [298](#)

Tabellenrealm [156](#), [384](#)

Tabellenteil [156](#)

verteilt

Datenbank [409](#)

Transaktion [410](#)

Verteiltabelle [409](#)

lokal [385](#)

volume [26](#)

Vordefinierte Variable

UDS-Online-Utility [304](#)

Vorgang [410](#)

## W

wählen

Konsistenzkriterien [85](#)

Prüfmodus [84](#)

Wahlwort [20](#)

WAIT-Anweisung

UDS-Online-Utility [302](#)

Warmstart [410](#)

Warnungen BCHECK [104](#)

Wiederanlauf

BMEND [410](#)

Session [410](#)

Wiederholungsgruppe [410](#)

## X

x-string [26](#)

## Z

Zählverfahren [71](#)

Zeitpunkt der letzten Änderung, Realm [248](#)

Zeitquittung [411](#)

Zielpegel [267](#)

zu prüfende Realms spezifizieren [88](#)

zu prüfende Satzarten spezifizieren [89](#)

zu prüfende SEARCH-Keys spezifizieren [94](#)

zu prüfende Sets spezifizieren [92](#)

zugelassene SQL-Zugriffe pro Basistabelle [196](#)

Zugriff

direkt [375](#), [411](#)

Hashbereich [226](#)

konkurrierend [383](#), [411](#)

relationaler [181](#)

sequenziell [398](#), [411](#)

Zugriffsart [411](#)

Zugriffsberechtigter [411](#)

Zugriffsberechtigung [411](#)

Zugriffspfad [411](#)

Zugriffsrecht [412](#)

zurücknehmen Online-Sicherungsfähigkeit [38](#)

zurücksetzen

Transaktion [407](#)

zusammengesetzter Schlüssel [396](#)

Zusätze, Datentypen [27](#)

zusätzliche Verkettung, Membersätze [158](#)

zuschalten Realm [36](#)

Zustand PTC [412](#)

Zwei-Phasen-Ende-Protokoll [412](#)

