

Deutsch



FUJITSU Software BS2000

# SESAM/SQL-Server V9.0

Sprachbeschreibung Teil 2

Benutzerhandbuch

Ausgabe Oktober 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>7</b>
<b>1.1</b>	<b>Zielsetzung und Zielgruppen des Handbuchs</b> . . . . .	<b>7</b>
<b>1.2</b>	<b>Konzept des Handbuchs</b> . . . . .	<b>8</b>
<b>1.3</b>	<b>Darstellungsmittel</b> . . . . .	<b>9</b>
<b>2</b>	<b>Grundlagen für das Arbeiten mit Utility-Anweisungen</b> . . . . .	<b>11</b>
<b>2.1</b>	<b>Utility-Funktionen von SESAM/SQL</b> . . . . .	<b>12</b>
<b>2.2</b>	<b>Space-Zustände und zulässige Utility-Anweisungen</b> . . . . .	<b>13</b>
<b>2.3</b>	<b>Utility-Anweisungen bei partitionierten Tabellen</b> . . . . .	<b>22</b>
<b>2.4</b>	<b>Pragma CHECK</b> . . . . .	<b>24</b>
<b>2.5</b>	<b>Datendarstellung in den Ein- und Ausgabedateien bei LOAD und UNLOAD</b> . . . . .	<b>26</b>
2.5.1	Lesbare Darstellung der Daten in Ein- und Ausgabedatei . . . . .	27
2.5.1.1	Lesbare Darstellung der Eingabedaten für LOAD . . . . .	27
2.5.1.2	Lesbare Darstellung der Ausgabedaten bei UNLOAD . . . . .	29
2.5.1.3	Lesbare Darstellung von Zeitwerten . . . . .	32
2.5.2	Standarddarstellung der Daten in der Ein- und Ausgabedatei . . . . .	33
<b>2.6</b>	<b>Bandsicherungen mit HSMS und ARCHIVE</b> . . . . .	<b>36</b>
2.6.1	HSMS-Lauf bei COPY und RECOVER steuern . . . . .	36
2.6.1.1	HSMS-Archiv für SESAM/SQL einrichten . . . . .	37
2.6.1.2	HSMS-Parameterdatei erstellen . . . . .	43
2.6.1.3	Löschen von HSMS-Aufträgen über die HSMS-Benutzeroberfläche . . . . .	51
2.6.1.4	Änderung der Benutzerkennung, des Pubsets oder des Systems . . . . .	51
2.6.2	ARCHIVE-Lauf bei COPY und RECOVER steuern . . . . .	56
2.6.2.1	ARCHIVE-Parameterdatei erstellen . . . . .	56
<b>2.7</b>	<b>Bearbeitung von Eingabe-, Ausgabe- und Fehlerdateien</b> . . . . .	<b>65</b>
<b>2.8</b>	<b>Benutzerkennung und Linknamen</b> . . . . .	<b>67</b>

<b>3</b>	<b>Utility-Anweisungen</b> . . . . .	<b>69</b>
<b>3.1</b>	<b>Überblick über die Utility-Anweisungen (alphabetisch)</b> . . . . .	<b>70</b>
<b>3.2</b>	<b>Alphabetischer Nachschlageteil</b> . . . . .	<b>71</b>
	ALTER CATALOG - Eigenschaften einer Datenbank ändern . . . . .	73
	ALTER DATA FOR TABLE - Mischen von Spaltenwerten, Anonymisieren von Daten . . . . .	74
	ALTER MEDIA DESCRIPTION - Medientabelle ändern . . . . .	76
	ALTER PARTITIONING FOR TABLE - Partitionierung einer Basistabelle ändern . . . . .	80
	CHECK CONSTRAINTS - Integritätsbedingungen prüfen . . . . .	91
	CHECK FORMAL - Formalprüfung von Tabellen und Indizes . . . . .	94
	COPY - SESAM-Sicherungsbestand erstellen . . . . .	97
	CREATE CATALOG - Datenbank (Catalog-Space) anlegen . . . . .	105
	CREATE MEDIA DESCRIPTION - Dateieigenschaften der datenbank-spezifischen Dateien festlegen . . . . .	115
	CREATE REPLICATION - Replikat erzeugen . . . . .	119
	DROP MEDIA DESCRIPTION - Medientabellen-Einträge für eine datenbank-spezifische Dateiart löschen . . . . .	127
	EXPORT TABLE - Tabelle aus einer Datenbank in eine Exportdatei exportieren . . . . .	129
	IMPORT TABLE - Tabelle aus einer Export-Datei in eine Datenbank importieren . . . . .	133
	LOAD - Anwenderdaten in eine Basistabelle laden . . . . .	140
	MIGRATE - Datenbanken und Tabellen umstellen . . . . .	166
	MODIFY - Metadaten zu SESAM-Sicherungsbeständen pflegen . . . . .	171
	RECOVER - Zurücksetzen und Reparieren, Indizes neu aufbauen . . . . .	178
	Anwender-Spaces und Datenbank reparieren oder zurücksetzen . . . . .	178
	Indizes neu aufbauen . . . . .	217
	REFRESH REPLICATION - Replikat aktualisieren . . . . .	219
	REFRESH SPACE - Replikat erweitern . . . . .	223
	REORG - Spaces und Basistabellen reorganisieren . . . . .	227
	REORG [CATALOG_]SPACE - Catalog-Space und Anwender-Spaces reorganisieren . . . . .	228
	REORG ONLINE TABLE - Basistabelle reorganisieren . . . . .	233
	UNLOAD - Anwenderdaten aus einer Tabelle ausgeben . . . . .	234

<b>4</b>	<b>Anhang</b> . . . . .	<b>257</b>
<b>4.1</b>	<b>Syntaxübersicht</b> . . . . .	<b>257</b>
<b>4.2</b>	<b>SQL-Schlüsselwörter</b> . . . . .	<b>263</b>
	<b>Literatur</b> . . . . .	<b>273</b>
	<b>Stichwörter</b> . . . . .	<b>275</b>



---

# 1 Einleitung

Das Datenbanksystem SESAM/SQL-Server erfüllt durch seine Funktionen und seine Architekturmerkmale alle Anforderungen, die heute an einen leistungsfähigen Datenbankserver gestellt werden. Diese Eigenschaft drückt sich auch im Produktnamen SESAM/SQL-Server aus.

SESAM/SQL-Server gibt es als Standard Edition mit Singletask-Betrieb und als Enterprise Edition, die den Multitask-Betrieb beinhaltet.

Der Einfachheit halber ist im Folgenden von SESAM/SQL die Rede, wenn das Datenbanksystem SESAM/SQL-Server gemeint ist.

Folgende einleitenden Beschreibungen befinden sich zentral im „[Basishandbuch](#)“:

- Kurzbeschreibung des Produkts
- Konzept der SESAM/SQL-Server-Dokumentation
- Beispieldatenbank
- Readme-Datei
- Änderungen gegenüber den Vorgänger-Handbüchern

## 1.1 Zielsetzung und Zielgruppen des Handbuchs

Das vorliegende Handbuch richtet sich an alle SQL-Benutzer, die Utility-Anweisungen zum Aufbauen und Warten einer SESAM/SQL-Datenbank einsetzen.

Das „[Basishandbuch](#)“ wird als bekannt vorausgesetzt, insbesondere die SESAM/SQL-Objekte und -Konzepte sowie das Utility-Konzept, auf denen die Utility-Anweisungen basieren.

Wenn Sie Utility-Anweisungen in ein Programm einbetten, müssen Sie mit der Programmiersprache COBOL vertraut sein.

### 1.2 Konzept des Handbuchs

Das vorliegende Handbuch „SQL-Sprachbeschreibung Teil 2: Utilities“ enthält eine vollständige Beschreibung der Utility-Anweisungen und der durch sie realisierten Dienstprogramm-funktionen.

Das Kapitel „Grundlagen für das Arbeiten mit Utility-Anweisungen“ enthält Informationen, die im Zusammenhang mit der Ausführung von Utility-Anweisungen benötigt werden.

Das Kapitel „Utility-Anweisungen“ enthält einen alphabetischen Nachschlageteil über alle Utility-Anweisungen.

Der Anhang „Syntaxübersicht“ ist ein alphabetischer Nachschlageteil über alle im Handbuch verwendeten Syntaxelemente. Einige der benötigten Syntaxelemente sind im Handbuch „[SQL-Sprachbeschreibung Teil 1: SQL-Anweisungen](#)“ beschrieben.

Der Anhang „SQL-Schlüsselwörter“ enthält eine Auflistung aller von SESAM/SQL in SQL- sowie Utility-Anweisungen verwendeten Schlüsselwörter.

## 1.3 Darstellungsmittel

In diesem Handbuch verwenden wir folgende Darstellungsmittel:

<p>_____</p> <p>_____</p> <p><b>GROSS</b></p> <p><u>unterstrichen</u></p> <p><b>fett</b></p> <p><i>kursiv</i></p> <p>Schreibmaschienschrift</p> <p>::=</p> <p>[ ]</p> <p>{ }</p> <p>{ } (mehrzeilig)</p> <p>{ } (einzeilig)</p> <p>,...</p> <p>...</p> <p></p> <p></p>	<p>Syntaxdefinitionen; Fortsetzungszeilen innerhalb von Syntaxdefinitionen sind eingerückt.</p> <p>SQL-Schlüsselwörter</p> <p>Voreinstellungen</p> <p>Hervorhebung im Fließtext</p> <p>Variablen in Syntaxdefinitionen und im Fließtext</p> <p>Programmtext in Syntaxdefinitionen und in Beispielen</p> <p>Definitionszeichen</p> <p>Die Angabe auf der rechten Seite von ::= definiert die Syntax für das Element auf der linken Seite.</p> <p>Optionale Angaben</p> <p>Die eckigen Klammern sind Metazeichen, die in einer SQL-Anweisung nicht angegeben werden.</p> <p>Alternative Angaben in Syntaxdefinitionen. Jede Zeile enthält eine Alternative.</p> <p>Die geschweiften Klammern sind Metazeichen, die in einer SQL-Anweisung nicht angegeben werden.</p> <p>Zusammenfassung von Klauseln in Syntaxdefinitionen, die gemeinsam wiederholt werden können.</p> <p>Die geschweiften Klammern sind Metazeichen, die in einer SQL-Anweisung nicht angegeben werden.</p> <p>In Syntaxdefinitionen bedeutet diese Schreibweise, dass die vorausgehende Angabe beliebig oft, durch Komma getrennt, wiederholt werden kann. Wird keine Wiederholung angegeben, fällt auch das Komma weg.</p> <p>In Syntaxdefinitionen bedeuten die Punkte, dass die vorausgehende Angabe beliebig oft wiederholt werden kann. In Beispielen bedeuten die Punkte, dass die restlichen Teile für das Beispiel ohne Bedeutung sind.</p> <p>Die Punkte sind Metazeichen, die in einer SQL-Anweisung nicht angegeben werden.</p> <p>Hinweise auf besonders wichtige Informationen</p> <p>Warnhinweise</p>
--	---



---

## 2 Grundlagen für das Arbeiten mit Utility-Anweisungen

Das vorliegende Kapitel enthält eine Zusammenstellung der Informationen, die Sie im Zusammenhang mit der Ausführung von Utility-Anweisungen benötigen:

- [Utility-Funktionen von SESAM/SQL](#)
- [Space-Zustände und zulässige Utility-Anweisungen](#)
- [Pragma CHECK](#)
- [Datendarstellung in den Ein- und Ausgabedateien bei LOAD und UNLOAD](#)
- [Bandsicherungen mit HSMS und ARCHIVE](#)
- [Bearbeitung von Eingabe-, Ausgabe- und Fehlerdateien](#)
- [Benutzererkennung und Linknamen](#)

Eine einführende Beschreibung des Utility-Konzepts finden Sie im „[Basishandbuch](#)“.

## 2.1 Utility-Funktionen von SESAM/SQL

SESAM/SQL stellt Utility-Funktionen über Utility-Anweisungen bereit.

Die folgende Tabelle zeigt, welche Utility-Funktionen SESAM/SQL anbietet und durch welche Utility-Anweisungen sie realisiert werden.

Utility-Funktion	Utility-Anweisung
Datenbank (Catalog-Space) anlegen	CREATE CATALOG
Eigenschaften der Datenbank ändern	ALTER CATALOG
SESAM-Sicherungsbestand erstellen	COPY
Datenbank, Catalog-Space, Anwender-Space(s) reparieren	RECOVER [USING]
Replikat erzeugen	CREATE REPLICATION
Replikat aktualisieren	REFRESH REPLICATION
Replikat erweitern	REFRESH SPACE
Datenbank, Catalog-Space, Anwender-Space(s) zurücksetzen	RECOVER TO
Indizes neu aufbauen	RECOVER INDEX
Anwenderdaten in Basistabelle laden	LOAD
Anwenderdaten aus Basistabelle ausgeben	UNLOAD
Mischen von Spaltenwerten, Anonymisieren von Daten	ALTER DATA FOR TABLE
Tabelle aus Datenbank in Export-Datei exportieren	EXPORT TABLE
Tabelle aus Export-Datei in Datenbank importieren	IMPORT TABLE
Catalog-Space, Anwender-Spaces oder Basistabellen reorganisieren	REORG
Partitionierung einer Basistabelle ändern	ALTER PARTITIONING FOR TABLE
Integritätsbedingungen prüfen	CHECK CONSTRAINTS
Tabellen und Indizes formal prüfen	CHECK FORMAL
Medientabelle bearbeiten	ALTER MEDIA DESCRIPTION CREATE MEDIA DESCRIPTION DROP MEDIA DESCRIPTION
Informationen (Metadaten) über die SESAM-Sicherungsbestände pflegen	MODIFY
Datenbank und Tabellen umstellen	MIGRATE

Tabelle 1: Utility-Funktionen von SESAM/SQL

Die einzelnen Utility-Anweisungen sind im [Kapitel „Utility-Anweisungen“ auf Seite 69](#) ausführlich beschrieben.

## 2.2 Space-Zustände und zulässige Utility-Anweisungen

Außer bei den Utility-Anweisungen MODIFY sowie ALTER/CREATE/DROP MEDIA DESCRIPTION beginnt SESAM/SQL die Ausführung einer Utility-Anweisung zunächst exklusiv auf einem oder mehreren Spaces (Catalog-Space, Anwender-Spaces). Je nach Utility-Anweisung ist während der Ausführung der Anweisung lesender oder ändernder Zugriff anderer SESAM/SQL-Benutzer auf einen von der Utility-Anweisung betroffenen Space möglich, eingeschränkt möglich oder überhaupt nicht möglich (siehe „[Basishandbuch](#)“). Die während der Ausführung einer Utility-Anweisung bestehenden Sperren auf Anwender-Spaces, Tabellen und Indizes bleiben in bestimmten Fällen auch nach Ausführung der Utility-Anweisung erhalten und führen zu verschiedenen Zuständen eines Anwender-Space.

Bei SESAM/SQL kann ein Anwender-Space folgende Space-Zustände annehmen:

- „space o.k.“ besagt, dass der Anwender-Space uneingeschränkt bearbeitet werden kann und keine gesperrten Tabellen und Indizes enthält.
- „check pending“ besagt, dass auf dem Anwender-Space mindestens eine Basistabelle liegt, die sich im Zustand „check pending“ befindet. Eine Basistabelle befindet sich im Zustand „check pending“, wenn mindestens eine Integritätsbedingung für diese Basistabelle verletzt ist oder wenn nach IMPORT TABLE, LOAD oder RECOVER (falls nicht der aktuelle Stand wiederhergestellt werden konnte) die Integritätsbedingungen für diese Tabelle noch nicht geprüft sind.  
Basistabellen im Zustand „check pending“ können nur mit dem Pragma CHECK OFF (siehe [Seite 24](#)) gelesen werden.  
Sie können mit DML-Anweisungen geändert werden.
- „copy pending“ besagt, dass von dem Anwender-Space mit der Utility-Anweisung COPY ein SESAM-Sicherungsbestand erstellt werden muss, bevor wieder ändernd auf ihn zugegriffen werden kann.
- „load running“ besagt, dass in eine Tabelle auf dem Anwender-Space mit MIGRATE, LOAD OFFLINE oder IMPORT TABLE Datensätze zugeladen werden. Der Space ist zum Ändern gesperrt. Vom Zuladen nicht betroffene Tabellen können gelesen werden.  
Wenn das Zuladen aufgrund eines Fehlers abgebrochen wird, bleibt der Zustand erhalten und der Anwender-Space muss mit RECOVER wiederhergestellt werden.
- „recover pending“ besagt, dass SESAM/SQL bei der Reparatur des Anwender-Space mit der Utility-Anweisung RECOVER die DA-LOG-Dateien nicht bis zum aktuellen Stand nachfahren konnte.
- „space defect“ besagt, dass der Anwender-Space nur durch Maßnahmen des Media Recovery wiederhergestellt werden kann.

„reorg pending“ besagt, dass die maximale Größe des Space (64 GByte) erreicht ist. Der Space kann nicht mehr erweitert werden.

Folgende SQL-Anweisungen können in diesem Zustand noch ausgeführt werden:

- DELETE zum Löschen einzelner Sätze
- Anweisungen zur Abfrage von Daten
- DROP TABLE DEFERRED
- DROP INDEX DEFERRED
- REORG SPACE

Der Reorganisation des Anwender-Space (REORG SPACE) bringt bestenfalls kurzfristige Entlastung.

Langfristig sollte der Space durch Löschen nicht benötigter Sätze, Verlagern von Tabellen und Indizes oder Partitionieren von Tabellen entlastet werden.

Der Zustand „index pending“ ist seit SESAM/SQL V3.1 entfallen. Defekte Indizes werden bei COPY und CREATE REPLICATION ignoriert (siehe [Abschnitt „Alphabetischer Nachschlageteil“ auf Seite 71](#)).

Die folgenden Space-Zustände können gleichzeitig auftreten. Sie müssen in der angegebenen Reihenfolge bearbeitet werden:

1. „load running“
2. „check pending“
3. „copy pending“

Beim Catalog-Space sind aus Anwendersicht nur die Zustände „space o.k.“ und „space defect“ von Bedeutung. Beim Zustand „space defect“ des Catalog-Space sind Maßnahmen des Media Recovery erforderlich (siehe [„Basishandbuch“](#)).

[Tabelle 2](#) zeigt, in welchen Zuständen sich ein Anwender-Space befinden kann, welche Utility-Anweisungen in den einzelnen Zuständen erlaubt sind und in welchen Zustand der Anwender-Space durch eine erlaubte Utility-Anweisung überführt wird.

Der Zustand „space defect“ kann immer auftreten und ist deshalb in [Tabelle 2](#) als resultierender Zustand nicht berücksichtigt.

Wenn der resultierende Space-Zustand davon abhängt, ob für den Anwender-Space Logging eingeschaltet ist oder nicht, steht in [Tabelle 2](#) an entsprechender Stelle der Hinweis „mit Logging“ bzw. „ohne Logging“.

Zustand des Anwender-Space	erlaubte Utility-Anweisung	mögliche neue Zustände des Anwender-Space
„space o.k.“	ALTER DATA FOR TABLE	– „copy pending“ (mit Logging)
	ALTER PARTITIONING FOR TABLE	– „space o.k.“ – „copy pending“ wenn Datensätze verschoben oder gelöscht wurden (mit Logging).
	CHECK CONSTRAINTS	– „space o.k.“ – „check pending“
	CHECK FORMAL, COPY, EXPORT TABLE, UNLOAD, REORG	– „space o.k.“
	LOAD OFFLINE	– „check pending“ – „copy pending“ (mit Logging) – „load running“, während Datensätze zugeladen werden oder nachdem LOAD OFFLINE aufgrund eines Fehlers abgebrochen wurde. – „space o.k.“ (ohne Logging)
	LOAD ONLINE	– „check pending“ – „space o.k.“
	IMPORT TABLE	– „copy pending“, wenn Datensätze importiert werden (mit Logging) – „load running“, während Datensätze importiert werden oder nachdem IMPORT TABLE aufgrund eines Fehlers abgebrochen wurde. – „space o.k.“, wenn keine Datensätze importiert werden oder wenn Datensätze bei ausgeschaltetem Logging importiert werden.
	MIGRATE	– „copy pending“ (mit Logging) – „load running“, während Datensätze migriert werden oder nachdem MIGRATE aufgrund eines Fehlers abgebrochen wurde. – „space o.k.“ (ohne Logging)

Tabelle 2: Space-Zustände und zulässige Utility-Anweisungen

(Teil 1 von 7)

Zustand des Anwender-Space	erlaubte Utility-Anweisung	mögliche neue Zustände des Anwender-Space
„space o.k.“ (Fortsetzung)	RECOVER	<ul style="list-style-type: none"> <li>– „recover pending“</li> <li>– „check pending“</li> <li>– „copy pending“, nach RECOVER USING, wenn CREATE INDEX oder DROP INDEX für eine partitionierte Tabelle auf diesem Space nachgefahren werden muss, aber nicht alle Partitionen in der RECOVER-Anweisung enthalten sind.</li> <li>oder</li> <li>nach RECOVER TO, wenn Anpassung des Anwender-Space (mit Logging) an Definitionen im Catalog-Space erforderlich war bzw. nach RECOVER ADJUST.</li> <li>– „space o.k.“</li> </ul>
	RECOVER INDEX	<ul style="list-style-type: none"> <li>– „copy pending“ (mit Logging)</li> <li>– „space o.k.“ (ohne Logging)</li> </ul>
„check pending“	ALTER DATA FOR TABLE (Die Tabelle selber darf nicht im Zustand „check pending“ sein.)	– „check pending“
	ALTER PARTITIONING FOR TABLE (Die Tabelle selber darf nicht im Zustand „check pending“ sein.)	– „check pending“
	CHECK CONSTRAINTS ON TABLE CHECK CONSTRAINTS ON SPACE	– „space o.k.“
	CHECK CONSTRAINTS (für einzelne Integritätsbedingungen), UNLOAD	– „check pending“
	EXPORT TABLE (Die Tabelle selber darf nicht im Zustand „check pending“ sein.)	– „check pending“

Tabelle 2: Space-Zustände und zulässige Utility-Anweisungen

(Teil 2 von 7)

Zustand des Anwender-Space	erlaubte Utility-Anweisung	mögliche neue Zustände des Anwender-Space
„check pending“ (Fortsetzung)	IMPORT TABLE	<ul style="list-style-type: none"> <li>– „check pending“</li> <li>– „copy pending“, wenn Datensätze importiert werden (mit Logging)</li> <li>– „load running“, während Datensätze importiert werden oder nachdem IMPORT TABLE aufgrund eines Fehlers abgebrochen wurde.</li> </ul>
	LOAD OFFLINE	<ul style="list-style-type: none"> <li>– „check pending“</li> <li>– „copy pending“ (mit Logging)</li> <li>– „load running“, während Datensätze zugeladen werden oder nachdem LOAD OFFLINE aufgrund eines Fehlers abgebrochen wurde.</li> <li>– „space o.k.“ (ohne Logging)</li> </ul>
	LOAD ONLINE	<ul style="list-style-type: none"> <li>– „check pending“</li> <li>– „space o.k.“</li> </ul>
	MIGRATE	<ul style="list-style-type: none"> <li>– „check pending“</li> <li>– „copy pending“ (mit Logging)</li> <li>– „load running“, während Datensätze migriert werden oder nachdem MIGRATE aufgrund eines Fehlers abgebrochen wurde.</li> </ul>
	RECOVER	<ul style="list-style-type: none"> <li>– „copy pending“, nach RECOVER USING, wenn CREATE INDEX oder DROP INDEX für eine partitionierte Tabelle auf diesem Space nachgefahren werden muss, aber nicht alle Partitionen in der RECOVER-Anweisung enthalten sind.</li> <li>oder</li> <li>nach RECOVER TO, wenn Anpassung des Anwender-Space (mit Logging) an Definitionen im Catalog-Space erforderlich war bzw. nach RECOVER ADJUST.</li> <li>– „check pending“</li> <li>– „space o.k.“</li> <li>– „recover pending“</li> </ul>
	RECOVER INDEX	<ul style="list-style-type: none"> <li>– „check pending“</li> <li>– „copy pending“ (mit Logging)</li> </ul>

Tabelle 2: Space-Zustände und zulässige Utility-Anweisungen

(Teil 3 von 7)

Zustand des Anwender-Space	erlaubte Utility-Anweisung	mögliche neue Zustände des Anwender-Space
„copy pending“	ALTER DATA FOR TABLE	<ul style="list-style-type: none"> <li>– „copy pending“</li> <li>– „load running“, während Spaltenwerte vertauscht werden oder nachdem ALTER DATA FOR TABLE aufgrund eines Fehlers abgebrochen wurde.</li> </ul>
	ALTER PARTITIONING FOR TABLE	<ul style="list-style-type: none"> <li>– „copy pending“</li> <li>– „load running“, während Sätze verschoben werden oder nachdem ALTER PARTITIONING FOR TABLE aufgrund eines Fehlers abgebrochen wurde.</li> </ul>
	CHECK CONSTRAINTS, CHECK FORMAL, UNLOAD, EXPORT TABLE	<ul style="list-style-type: none"> <li>– „copy pending“</li> </ul>
	COPY	<ul style="list-style-type: none"> <li>– „space o.k.“</li> </ul>
	IMPORT TABLE	<ul style="list-style-type: none"> <li>– „copy pending“</li> <li>– „load running“, während Datensätze importiert werden oder nachdem IMPORT TABLE aufgrund eines Fehlers abgebrochen wurde.</li> </ul>
	LOAD OFFLINE	<ul style="list-style-type: none"> <li>– „check pending“</li> <li>– „copy pending“</li> <li>– „load running“, während Datensätze zugeladen werden oder nachdem LOAD OFFLINE aufgrund eines Fehlers abgebrochen wurde.</li> </ul>
	MIGRATE	<ul style="list-style-type: none"> <li>– „copy pending“</li> <li>– „load running“, während Datensätze migriert werden oder nachdem MIGRATE aufgrund eines Fehlers abgebrochen wurde.</li> </ul>

Tabelle 2: Space-Zustände und zulässige Utility-Anweisungen

(Teil 4 von 7)

Zustand des Anwender-Space	erlaubte Utility-Anweisung	mögliche neue Zustände des Anwender-Space
„copy pending“ (Fortsetzung)	RECOVER	<ul style="list-style-type: none"> <li>– „copy pending“, nach RECOVER USING, wenn CREATE INDEX oder DROP INDEX für eine partitionierte Tabelle auf diesem Space nachgefahren werden muss, aber nicht alle Partitionen in der RECOVER-Anweisung enthalten sind.</li> <li>oder</li> <li>nach RECOVER TO, wenn Anpassung des Anwender-Space (mit Logging) an Definitionen im Catalog-Space erforderlich war bzw. nach RECOVER ADJUST.</li> <li>– „check pending“</li> <li>– „space o.k.“</li> <li>– „recover pending“</li> </ul>
„space defect“	RECOVER	<ul style="list-style-type: none"> <li>– „copy pending“, nach RECOVER USING, wenn CREATE INDEX oder DROP INDEX für eine partitionierte Tabelle auf diesem Space nachgefahren werden muss, aber nicht alle Partitionen in der RECOVER-Anweisung enthalten sind.</li> <li>oder</li> <li>nach RECOVER TO, wenn Anpassung des Anwender-Space (mit Logging) an Definitionen im Catalog-Space erforderlich war bzw. nach RECOVER ADJUST.</li> <li>– „check pending“</li> <li>– „space o.k.“</li> <li>– „recover pending“</li> </ul>

Tabelle 2: Space-Zustände und zulässige Utility-Anweisungen

(Teil 5 von 7)

Zustand des Anwender-Space	erlaubte Utility-Anweisung	mögliche neue Zustände des Anwender-Space
„recover pending“	RECOVER	<ul style="list-style-type: none"> <li>– „copy pending“ , nach RECOVER USING, wenn CREATE INDEX oder DROP INDEX für eine partitionierte Tabelle auf diesem Space nachgefahren werden muss, aber nicht alle Partitionen in der RECOVER-Anweisung enthalten sind.</li> <li>oder</li> <li>nach RECOVER TO, wenn Anpassung des Anwender-Space (mit Logging) an Definitionen im Catalog-Space erforderlich war bzw. nach RECOVER ADJUST.</li> <li>– „check pending“</li> <li>– „space o.k.“</li> </ul>
	UNLOAD OFFLINE	– „recover pending“
„load running“	RECOVER	<ul style="list-style-type: none"> <li>– „copy pending“ nach RECOVER USING, wenn CREATE INDEX oder DROP INDEX für eine partitionierte Tabelle auf diesem Space nachgefahren werden muss, aber nicht alle Partitionen in der RECOVER-Anweisung enthalten sind.</li> <li>oder</li> <li>nach RECOVER TO, wenn Anpassung des Anwender-Space (mit Logging) an Definitionen im Catalog-Space erforderlich war bzw. nach RECOVER ADJUST.</li> <li>– „check pending“</li> <li>– „space o.k.“</li> </ul>
	UNLOAD (Bearbeitete Tabelle selbst nicht „load running“) EXPORT (Bearbeitete Tabelle selbst nicht „load running“)	– „load running“

Tabelle 2: Space-Zustände und zulässige Utility-Anweisungen

(Teil 6 von 7)

Zustand des Anwender-Space	erlaubte Utility-Anweisung	mögliche neue Zustände des Anwender-Space
„reorg pending“	REORG SPACE	<ul style="list-style-type: none"> <li>– „space o.k.“</li> <li>– „reorg pending“</li> </ul> Der Reorganisation des Anwender-Space bringt bestenfalls kurzfristige Entlastung.
	UNLOAD EXPORT TABLE CHECK FORMAL CHECK CONSTRAINTS	<ul style="list-style-type: none"> <li>– „reorg pending“</li> </ul>
	RECOVER	<ul style="list-style-type: none"> <li>– „space o.k.“</li> <li>– „reorg pending“</li> <li>– „check pending“</li> <li>– „copy pending“</li> <li>– „recover pending“</li> </ul> Die Recovery des Anwender-Space bringt bestenfalls kurzfristige Entlastung.

Tabelle 2: Space-Zustände und zulässige Utility-Anweisungen

(Teil 7 von 7)

## 2.3 Utility-Anweisungen bei partitionierten Tabellen

Viele Utility-Anweisungen bearbeiten die Anwender-Spaces, auf denen die Daten einer Datenbank gespeichert sind. Die Anwender-Spaces müssen verfügbar sein um sie bearbeiten zu können, d.h. die Utility-Anweisung muss darauf zugreifen können.

Die Daten einer partitionierten Tabelle sind in den sogenannten Partitionen auf mehrere Anwender-Spaces verteilt. Es kann vorkommen, dass eine oder mehrere Partitionen bzw. Anwender-Spaces der partitionierten Tabelle nicht verfügbar sind. Dabei wird zusätzlich zwischen logischer und physikalischer Verfügbarkeit unterschieden. Grundlegende Informationen zu partitionierten Tabellen und der Verfügbarkeit von Partitionen finden Sie im „[Basishandbuch](#)“.

Die folgende Tabelle zeigt für die einzelnen Utility-Anweisungen die Voraussetzungen an die Verfügbarkeit der Partitionen und wie die Reaktion im Fehlerfall ist, d.h. wenn mindestens eine Partition die Verfügbarkeitsvoraussetzung nicht erfüllt.

Anweisung	Voraussetzung an die Verfügbarkeit der Partitionen	Reaktion im Fehlerfall/ ggf. besondere Maßnahmen
ALTER DATA FOR TABLE	alle Partitionen der Tabelle logisch und physikalisch	Abweisung mit SQLSTATE
ALTER PARTITIONING FOR TABLE	betroffene Partitionen logisch und physikalisch	Abweisung mit SQLSTATE
CHECK CONSTRAINTS ON SPACE	alle Partitionen der Tabellen, die auf dem Space eine Partition haben, logisch und physikalisch	Abweisung mit SQLSTATE
CHECK CONSTRAINTS ON TABLE	alle Partitionen der Tabelle logisch und physikalisch	Abweisung mit SQLSTATE
CHECK FORMAL SPACE	alle Partitionen des Space physikalisch	Abweisung mit SQLSTATE
CHECK FORMAL TABLE	alle Partitionen der Tabelle physikalisch	Meldung in Fehlerdatei und SQL-Warnung
EXPORT TABLE ... ALL DATA	alle Partitionen physikalisch	Abweisung mit SQLSTATE
EXPORT TABLE ... WHERE	betroffene Partitionen logisch und physikalisch	Abweisung mit SQLSTATE
IMPORT TABLE	alle Partitionen physikalisch	Abweisung mit SQLSTATE

Tabelle 3: Verfügbarkeit von Partitionen bei Utility-Anweisungen

(Teil 1 von 2)

Anweisung	Voraussetzung an die Verfügbarkeit der Partitionen	Reaktion im Fehlerfall/ ggf. besondere Maßnahmen
LOAD OFFLINE mit Zählfeld	alle Partitionen physikalisch	Abweisung mit SQLSTATE
LOAD OFFLINE ohne Zählfeld	betroffene Partitionen physikalisch	Meldung in Fehlerdatei
... mit GENERATE INDEX	alle Partitionen physikalisch	SQL-Warnung, Indizes sind „defekt“ gesetzt. RECOVER INDEX ON TABLE notwendig
... mit CONSTRAINT CHECK	alle Partitionen logisch und physikalisch	Tabelle im Zustand „check pending“. CHECK CONSTRAINTS notwendig
LOAD ONLINE mit Zählfeld	alle Partitionen logisch und physikalisch	Abweisung mit SQLSTATE
LOAD ONLINE ohne Zählfeld	betroffene Partitionen logisch und physikalisch	Abweisung mit SQLSTATE
LOAD ONLINE mit CHECK CONSTRAINTS	alle Partitionen logisch und physikalisch	Abweisung mit SQLSTATE
MIGRATE CALL DML TABLE	alle Partitionen physikalisch	Abweisung mit SQLSTATE
RECOVER ... [USING] ... TO und NO INDEX nicht angegeben	alle Partitionen physikalisch	Abweisung mit SQLSTATE
RECOVER ... [USING] ... TO und Integritätsbedingung für die Tabelle definiert	alle Partitionen physikalisch	Abweisung mit SQLSTATE
RECOVER INDEX ON TABLE	alle Partitionen physikalisch	Abweisung mit SQLSTATE
UNLOAD OFFLINE	alle Partitionen physikalisch	Meldung in Fehlerdatei
UNLOAD ONLINE	betroffene Partitionen logisch und physikalisch	Meldung mit SQLSTATE
Sonstige UTILITY-Anweisungen	keine	---

Tabelle 3: Verfügbarkeit von Partitionen bei Utility-Anweisungen

(Teil 2 von 2)

## 2.4 Pragma CHECK

Syntaktisch gesehen ist ein Pragma ein gewöhnlicher SQL-Kommentar, den SESAM/SQL als Mitteilung an sich selbst interpretiert.

Das Pragma CHECK legt fest, ob mit der SQL- oder Utility-Anweisung, in der dieses Pragma angegeben ist, Tabellen im Zustand „check pending“ (siehe [Seite 13](#)) bearbeitet werden können. Die Wirkung des Pragmas CHECK ist jeweils auf die Anweisung beschränkt, in der dieses Pragma angegeben ist. Kommt das Pragma CHECK innerhalb derselben Anweisung mehrfach vor, so gilt das zuletzt angegebene Pragma.

Innerhalb einer Anweisung ist das Pragma CHECK nur wirksam, wenn vor dem Pragma in der betreffenden Anweisung ausschließlich Trenner (inklusive Pragmas und SQL-Kommentare) stehen. Trenner sind im Handbuch „[SQL-Sprachbeschreibung Teil 1: SQL-Anweisungen](#)“ beschrieben.

Das Pragma CHECK ist in Verbindung mit folgenden SQL-Anweisungen wirksam (siehe Handbuch „[SQL-Sprachbeschreibung Teil 1: SQL-Anweisungen](#)“).

- CALL und in Routinen
- DECLARE CURSOR
- SELECT-Anweisung
- SELECT-Anweisung (dynamisch übersetzbar)
- DELETE ... WHERE CURRENT OF
- DELETE ... WHERE CURRENT OF (dynamisch übersetzbar)
- DELETE ... WHERE *suchbedingung*
- DELETE ... WHERE *suchbedingung* (dynamisch übersetzbar)
- INSERT
- MERGE
- UPDATE ... WHERE CURRENT OF
- UPDATE ... WHERE CURRENT OF (dynamisch übersetzbar)
- UPDATE ... WHERE *suchbedingung*
- UPDATE ... WHERE *suchbedingung* (dynamisch übersetzbar)

Das Pragma CHECK ist in Verbindung mit der folgenden Utility-Anweisung wirksam:

- UNLOAD ONLINE (siehe [Seite 234](#))

Bei allen anderen Anweisungen bleibt die Angabe des Pragmas CHECK ohne Wirkung.

---

```
-- %PRAGMA CHECK { ON }  
                  { OFF }
```

---

### ON

Bei der Ausführung der SQL- oder Utility-Anweisung berücksichtigt SESAM/SQL Integritätsbedingungen. Da SESAM/SQL Integritätsbedingungen (z.B. Eindeutigkeitsbedingung) zur Optimierung von Anweisungen verwendet, weist SESAM/SQL eine Anweisung bei Angabe von CHECK ON mit SQLSTATE ab, falls sich eine in der Anweisung direkt oder indirekt (über Views oder Referenzbedingungen) angesprochene Basistabelle im Zustand „check pending“ befindet.

### OFF

Bei der Ausführung der SQL- oder Utility-Anweisung berücksichtigt SESAM/SQL Integritätsbedingungen nicht. Damit ist die Bearbeitung von Basistabellen möglich, die sich im Zustand „check pending“ befinden, also von Tabellen, deren Integritätsbedingungen möglicherweise nicht erfüllt sind.

Die Ausführung einer ändernden SQL-Anweisung (INSERT, UPDATE, MERGE, DELETE) mit Angabe von CHECK OFF wird von SESAM/SQL mit SQLSTATE abgewiesen, falls sich die zu bearbeitende Tabelle nicht im Zustand „check pending“ befindet.

## 2.5 Datendarstellung in den Ein- und Ausgabedateien bei LOAD und UNLOAD

Die Utility-Anweisung LOAD lädt Daten aus einer Eingabedatei in die Datenbank; die Utility-Anweisung UNLOAD schreibt Daten aus der Datenbank in eine Ausgabedatei.

Es gibt verschiedene Möglichkeiten, die Daten in einer Ein- bzw. Ausgabedatei darzustellen:

- Sehr komfortable Möglichkeiten sind die lesbare Darstellung der Daten im Delimiterformat oder im CSV-Format. Dazu geben Sie bei der LOAD- bzw. UNLOAD-Anweisung die Klausel `DELIMITER_FORMAT` oder `CSV_FORMAT` an. Die Werte stehen in ihrer signifikanten Länge im Ein- bzw. Ausgabesatz. Das DELIMITER-Zeichen zeigt das Ende eines Wertes an.

Zur Darstellung der Werte wird der codierte Zeichensatz der Ausgabedatei (CCSN) verwendet. Im Delimiterformat können Sie auch Ein- bzw. Ausgabedateien mit dem CCSN UTFE bearbeiten.

Dieser Handbuchabschnitt verwendet für Dateien mit EBCDIC- oder UTFE-Zeichensatz den Begriff „(mit einem Editor) lesbare Darstellung“. UTFE-Dateien können z.B. in BS2000 mit EDT ab V17.0 oder in Windows mit dem openFT-Editor ab V10.0 gelesen werden. Für Dateien mit EBCDIC-Zeichensatz spricht man traditionell auch von „abdruckbarer Darstellung“.

- Eine weitere Möglichkeit, die Daten in lesbarer Form darzustellen, besteht darin, im selbst definierten Ausgabeformat der LOAD- bzw. UNLOAD-Anweisung den Datentyp `CHARACTER` anzugeben.
- Ferner gibt es die Alternative, die Daten im Format der korrespondierenden Tabellenspalte oder in einem mit dem Datentyp der korrespondierenden Tabellenspalte verträglichen Format darzustellen. Dazu beschreiben Sie in der LOAD- bzw. UNLOAD-Anweisung die einzelnen Datenformate explizit mit dem Datentyp. Die Darstellung der Daten im Format der Tabellenspalte wird Standarddarstellung genannt (siehe [Abschnitt „Standarddarstellung der Daten in der Ein- und Ausgabedatei“ auf Seite 33](#)).

## 2.5.1 Lesbare Darstellung der Daten in Ein- und Ausgabedatei

In vielen Fällen ist es komfortabel, die Eingabedaten für LOAD und die Ausgabedaten von UNLOAD in einer mit einem Editor lesbaren, nicht sedezimal verschlüsselten Form in der Ein- bzw. Ausgabedatei vorzufinden.

Zusätzlich zu der in SQL angebotenen Datenkonvertierung auf Basis der Verträglichkeit von Datentypen (siehe Handbuch „[SQL-Sprachbeschreibung Teil 1: SQL-Anweisungen](#)“) bietet SESAM/SQL bei den Utility-Anweisungen LOAD und UNLOAD die Konvertierung von Daten einer Datenbank in Zeichenketten.

### 2.5.1.1 Lesbare Darstellung der Eingabedaten für LOAD

SESAM/SQL konvertiert alphanumerische Zeichenketten der Eingabedatei gemäß dem CCSN der Eingabedatei soweit möglich in den Datentyp der korrespondierenden Tabellenspalte. Dieser Datentyp darf ein beliebiger SQL-Datentyp sein.

Die Eingabewerte müssen - bis auf die unten beschriebenen Abweichungen - den üblichen Konventionen zur Darstellung von Werten des betreffenden Datentyps genügen, um korrekt konvertiert zu werden (siehe Handbuch „[SQL-Sprachbeschreibung Teil 1: SQL-Anweisungen](#)“).

#### Darstellung im Delimiter-Format

##### *Beispiel*

```
100;Siemens-AG;Otto-Hahn-Ring 6;81739;...
101;Login GmbH;Rosenheimer Str. 34;81667;...
```

In der LOAD-Anweisung geben Sie die DELIMITER\_FORMAT-Klausel mit dem DELIMITER-Zeichen ' ; ' für eine EBCDIC-Datei bzw. N ' ; ' für eine UTFE-Datei an.

##### *Regeln für Eingabedateien im Delimiter-Format*

- Die Werte müssen nur in der signifikanten Länge in der Eingabedatei stehen.
- Die Werte werden durch DELIMITER-Zeichen voneinander getrennt.
- Das DELIMITER-Zeichen darf nicht Bestandteil eines Wertes der Eingabedatei sein, da der Wert sonst nicht wie gewünscht interpretiert werden kann.
- Die maximale Länge und Größe eines Wertes orientiert sich am Zieldatentyp. Wenn der Maximalwert überschritten wird, dann ist eine Konvertierung nicht möglich.
- Ein Wert darf länger als die definierte Länge der betreffenden Spalte vom Datentyp (NATIONAL) CHARACTER (VARYING) sein; beim LOAD wird der Rest abgeschnitten und eine Warnung ausgegeben.
- Die Eingabedatei darf auch CCSN=UTFE haben.

## Darstellung im CSV-Format

### Beispiel

```
100,"Siemens-AG",Otto-Hahn-Ring 6,81739,...
101,"Login,Logout GmbH",Rosenheimer!,Aiblinger Str.,81667,...
```

In der LOAD-Anweisung geben Sie die CSV\_FORMAT-Klausel mit dem DELIMITER-Zeichen ',' (Komma) für eine EBCDIC-Datei bzw. ';' für eine UTFE-Datei an. Zusätzlich geben Sie das QUOTE-Zeichen '"' und das ESCAPE-Zeichen '\' an.

### Regeln für Eingabedateien im CSV-Format

Grundlegende Informationen zum Aufbau von CSV-Dateien finden Sie im Handbuch „[SQL-Sprachbeschreibung Teil 1: SQL-Anweisungen](#)“.

- Die Werte müssen nur in der signifikanten Länge in der Eingabedatei stehen.
- Die Werte werden durch DELIMITER-Zeichen voneinander getrennt.
- Ein Satz einer SAM-Datei kann mehrere Eingabezeilen, getrennt durch NEWLINE-Zeichen, enthalten.
- Wenn am Ende des SAM-Satzes ein ESCAPE-Zeichen steht, dann gilt damit die Eingabezeile als noch nicht beendet und wird mit dem folgenden SAM-Satz fortgesetzt. Somit kann sich eine Eingabezeile über mehrere SAM-Sätze erstrecken. (Voraussetzung: in der LOAD-Anweisung ist das ESCAPE-Zeichen angegeben.)
- Spaltenwerte können das DELIMITER-Zeichen oder das NEWLINE-Zeichen enthalten. Die Spaltenwerte müssen dann in QUOTE-Zeichen eingeschlossen werden. (Voraussetzung: in der LOAD-Anweisung ist das QUOTE-Zeichen angegeben.) Auch die Escape-Folge *escape delimiter* in einem Spaltenwert wird als ein DELIMITER-Zeichen interpretiert. (Voraussetzung: in der LOAD-Anweisung ist das ESCAPE-Zeichen angegeben.)
- Die Eingabedatei darf auch CCSN=UTFE haben.

## Darstellung im selbst definierten Ausgabeformat

### Beispiel

```
100Siemens-AG   Otto-Hahn-Ring 6       81739...
101Login GmbH  Rosenheimer Str. 34   81667...
```

Im Ladeformat der LOAD-Anweisung geben Sie für formatiert dargestellte Eingabewerte den Datentyp CHARACTER an.

### 2.5.1.2 Lesbare Darstellung der Ausgabedaten bei UNLOAD

Bei Ausgabe mit dem Datentyp CHARACTER konvertiert SESAM/SQL die Ausgabewerte einer Tabellenspalte beliebigen SQL-Datentyps soweit möglich in alphanumerische Zeichenketten der Ausgabedatei gemäß deren CCSN. National-Werte, die nicht gemäß dem CCSN der Datenbank konvertiert werden können, führen zu einem Eintrag in der Fehlerdatei. Alternativ kann die Ausgabe mit dem Datentyp NCHAR erfolgen.

SESAM/SQL legt die Ausgabewerte - bis auf die auf [Seite 31](#) beschriebenen Abweichungen - gemäß den SQL-Konventionen zur Darstellung von Werten des betreffenden Datentyps in der Ausgabedatei ab.



Die Ausgabedatei erhält immer den CCSN der Datenbank, auch wenn alle Werte z.B. im Datentyp NCHAR ausgegeben werden. Eine solche Ausgabedatei mit NCHAR-Daten können Sie mit EDT ab V17.0 oder dem openFT-Editor ab V10.0 lesen und bearbeiten.

#### Darstellung im Delimiter-Format

##### *Beispiel*

```
100;Siemens-AG;Otto-Hahn-Ring 6;81739;...
101;Login GmbH;Rosenheimer Str. 34;81667;...
```

In der UNLOAD-Anweisung geben Sie die DELIMITER\_FORMAT-Klausel mit dem DELIMITER-Zeichen ';' für eine EBCDIC-Datei bzw. N' für eine UTFE-Datei an.

##### *Regeln für Ausgabedateien im Delimiter-Format*

- Die Werte werden nur in der signifikanten Länge in die Ausgabedatei geschrieben.

##### *Ausnahmen*

- Werte deren Spalten mit dem Datentyp (NATIONAL) CHARACTER VARYING definiert sind, werden in der abgespeicherten Länge ohne Längensfeld ausgegeben. Führende oder nachfolgende Leerzeichen bleiben erhalten.
- Werte deren Spalten mit dem Datentyp (NATIONAL) CHARACTER definiert sind, werden mit führenden aber ohne nachfolgende Leerzeichen ausgegeben. Besteht ein Wert nur aus Leerzeichen, wird genau ein Leerzeichen in die Ausgabedatei ausgegeben.
- Die Ausgabedatei kann auch mit CCSN=UTFE angelegt werden.

## Darstellung im CSV-Format

### Beispiel

```
knr,firma,strasse,plz,...  
100,Siemens-AG,Otto-Hahn-Ring 6,81739,...  
101;"Login,Logout GmbH";Rosenheimer Str. 34;81667;...
```

In der UNLOAD-Anweisung geben Sie die CSV\_FORMAT-Klausel mit dem DELIMITER-Zeichen ',' für eine EBCDIC-Datei bzw. N',' für eine UTFE-Datei an. Zusätzlich geben Sie das QUOTE-Zeichen '"' und das ESCAPE-Zeichen \"!\" an. Die Überschriftszeile mit den Spaltennamen wird durch die Angabe von WITH HEADER ausgegeben.

### Regeln für Ausgabedateien im CSV-Format

Grundlegende Informationen zum Aufbau von CSV-Dateien finden Sie im Handbuch „[SQL-Sprachbeschreibung Teil 1: SQL-Anweisungen](#)“.

- Die Werte werden nur in der signifikanten Länge in die Ausgabedatei geschrieben.
- Die Werte werden durch DELIMITER-Zeichen voneinander getrennt.
- Wenn Spaltenwerte das DELIMITER-Zeichen oder das NEWLINE-Zeichen enthalten, dann werden die Spaltenwerte in QUOTE-Zeichen eingeschlossen, wenn ein QUOTE-Zeichen angegeben ist.  
Wenn kein QUOTE-Zeichen, aber ein ESCAPE-Zeichen angegeben ist, dann wird dem DELIMITER-Zeichen ein ESCAPE-Zeichen vorangestellt.  
Ein NEWLINE-Zeichen kann dann nicht ausgegeben werden.
- Die Ausgabedatei darf auch CCSN=UTFE haben.

## Darstellung im selbst definierten Ausgabeformat

### Beispiel

```
100Siemens-AG   Otto-Hahn-Ring 6       81739...  
101Login GmbH  Rosenheimer Str. 34    81667...
```

In der UNLOAD-Anweisung geben Sie für formatiert darzustellende Ausgabewerte im entsprechenden Ausgabeformat den Datentyp CHARACTER an.

Die Längenangabe für CHARACTER ist so zu wählen, dass sich jeder mögliche Wert aus dem Wertebereich der zugeordneten Tabellenspalte durch die CHARACTER-Zeichenkette darstellen lässt. Beispielsweise beansprucht die formatierte Ausgabe von INTEGER-Werten bis zu 11 Zeichen.

**Abweichungen gegenüber der Darstellung der Werte an der SQL-Schnittstelle**

Es gelten folgende Abweichungen gegenüber den in der „[SQL-Sprachbeschreibung Teil 1: SQL-Anweisungen](#)“ dargestellten Regeln zur Darstellung der Werte an der SQL-Schnittstelle (z.B. bei INSERT):

- Zeichenketten, also alphanumerische (EBCDIC-)Zeichenketten und Unicode-Zeichenketten, werden ohne Hochkommata dargestellt.
- Multiple Werte werden ohne spitze Klammern dargestellt.
- Für die Darstellung von Zeitwerten gelten die im folgenden [Abschnitt „Lesbare Darstellung von Zeitwerten“](#) beschriebenen Regeln.

### 2.5.1.3 Lesbare Darstellung von Zeitwerten

Zeitwerte, d.h. Werte für die Datentypen DATE, TIME(3) und TIMESTAMP(3) geben Sie in der Eingabedatei für LOAD abweichend von den Konventionen an der SQL-Schnittstelle an: Die Schlüsselwörter DATE, TIME und TIMESTAMP sind bei LOAD nicht Bestandteil eines Zeitwerts. Entsprechendes gilt bei UNLOAD für die Ausgabe von Zeitwerten.

#### Darstellung von Werten des Datentyps DATE

Werte des Datentyps DATE werden wie folgt dargestellt:

*jahr-monat-tag*

<i>jahr</i>	Vierstellige ganze Zahl zwischen 0001 und 9999, die das Jahr angibt.
<i>monat</i>	Zweistellige ganze Zahl zwischen 01 und 12, die den Monat angibt.
<i>tag</i>	Zweistellige ganze Zahl zwischen 01 und 31 (passend zu Monat und Jahr), die den Tag angibt.

*Beispiel*

2010-06-09 ist der Wert für das Datum 9. Juni 2010.

#### Darstellung von Werten des Datentyps TIME(3)

Werte des Datentyps TIME(3) werden wie folgt dargestellt:

*stunde:minute:sek.millisek*

<i>stunde</i>	Zweistellige ganze Zahl zwischen 00 und 23, die die Stunde angibt.
<i>minute</i>	Zweistellige ganze Zahl zwischen 00 und 59, die die Minute angibt.
<i>sek</i>	Zweistellige ganze Zahl zwischen 00 und 61, die die Sekunde angibt.
<i>millisek</i>	Dreistellige ganze Zahl zwischen 000 und 999, die die tausendstel Sekunde angibt.

*Beispiel*

19:30:25.000 ist der Wert für die Uhrzeit 19 Uhr, 30 Minuten, 25 Sekunden.

#### Darstellung von Werten des Datentyps TIMESTAMP(3)

Werte des Datentyps TIMESTAMP(3) werden wie folgt dargestellt:

*jahr-monat-tag stunde:minute:sek.millisek*

Es gelten die bei DATE und TIME(3) gegebenen Erläuterungen.

## 2.5.2 Standarddarstellung der Daten in der Ein- und Ausgabedatei

Dieser Abschnitt ist für Sie nur relevant, wenn Sie nicht mit lesbaren Ein- bzw. Ausgabedaten arbeiten wollen. In diesem Fall beschreiben Sie die einzelnen Datenformate explizit mit einem Datentyp, der mit dem Datentyp der korrespondierenden Tabellenspalte verträglich ist oder Sie verwenden implizit den Datentyp der korrespondierenden Tabellenspalte.

Bei der Standarddarstellung ist zu beachten, dass alle Werte in voller Länge gemäß der Definition des Datentyps dargestellt werden. Das gilt auch für Werte, die nicht signifikant sind. So erzeugt etwa der UNLOAD für Spalten, die in der Tabelle den NULL-Wert haben, in der Ausgabedatei Werte entsprechend den Regeln, die bei WHEN NULL THEN beschrieben sind (siehe [Seite 243](#)). Die einzige Ausnahme bilden Werte vom Datentyp (NATIONAL) CHARACTER VARYING; diese liegen immer in signifikanter Länge vor (maßgeblich ist das Längenfeld); dementsprechend bildet z.B. der UNLOAD solche NULL-Werte, die als (NATIONAL) CHARACTER VARYING darzustellen sind, beim Fehlen der Klausel WHEN NULL THEN lediglich auf ein Längenfeld mit dem Inhalt 0 ab.

Die Besonderheit von NULL-Werten ist bei der Standarddarstellung der Daten speziell dann zu beachten, wenn die Ausgabedatei eines UNLOAD als Eingabedatei eines nachfolgenden LOAD dienen soll. Da beim UNLOAD auch NULL-Werte in bestimmter Länge dargestellt werden, sollte beim nachfolgenden LOAD verhindert werden, dass diese Werte als signifikant interpretiert werden. Zu diesem Zweck müssen beim LOAD entsprechende WHEN ... THEN NULL Bedingungen angegeben werden; es sei denn, die Tabelle, in die zugeladen werden soll, ist eine CALL-DML-Tabelle und der betreffende Wert ist der nicht signifikante Attributwert der korrespondierenden Spalte.

[Tabelle 4](#) zeigt, welche Datentypen es bei den Ladeformaten (LOAD) bzw. Ausgabeformaten (UNLOAD) gibt und wie SESAM/SQL die zugehörigen Werte in der Eingabedatei erwartet bzw. in welcher Darstellung SESAM/SQL diese Werte bei UNLOAD in die Ausgabedatei schreibt.

In den Formatbeschreibungen bei LOAD und UNLOAD geben Sie den Datentyp in gleicher Weise an wie an der SQL-Schnittstelle (siehe Handbuch „[SQL-Sprachbeschreibung Teil 1: SQL-Anweisungen](#)“) mit folgenden Ausnahmen:

- Der Datentyp FLOAT darf nicht angegeben werden. Anstatt FLOAT verwenden Sie REAL oder DOUBLE PRECISION.
- Die Angabe *dimension* (Anzahl der Spaltenelemente bei multiplen Spalten) ist nicht erlaubt.

Datentyp	Länge des Werts in der Datei	Standarddarstellung in der Ein- und Ausgabedatei
CHARACTER	max. 256 Byte (256 Zeichen)	beliebige alphanumerische (EBCDIC-)Zeichenkette
CHARACTER VARYING	max. 32000 Byte (32000 Zeichen) davor: 2 Byte (als Längenfeld)	Längenfeld (enthält in Binärdarstellung die Länge der anschließenden Zeichenkette); beliebige alphanumerische (EBCDIC-)Zeichenkette
NATIONAL CHARACTER	max. 256 Byte (128 Code Units)	beliebige Unicode-Zeichenkette
NATIONAL CHARACTER VARYING	max. 32000 Byte (16000 Code Units) davor: 2 Byte (als Längenfeld)	Längenfeld (enthält in Binärdarstellung die Länge der anschließenden Zeichenkette <b>in Bytes</b> ); beliebige Unicode-Zeichenkette
NUMERIC	max. 31 Byte (Anzahl Stellen)	entpackte Darstellung, Darstellung des Vorzeichens im ersten Halbbyte des letzten Byte. Beispiel Darstellung von 364 mit 3 Stellen: X'F3 F6 F4' Darstellung von -364 mit 3 Stellen: X'F3 F6 D4' Darstellung von 364 mit 4 Stellen: X'F0 F3 F6 F4'
DECIMAL	max. 16 Byte ([Anzahl Stellen / 2])  [ . . . ] bedeutet „nächstgrößere ganze Zahl“	gepackte Darstellung, Darstellung des Vorzeichens im zweiten Halbbyte des letzten Byte. Beispiel Darstellung von 364 mit 3 Stellen: X'36 4F' Darstellung von -364 mit 3 Stellen: X'36 4D' Darstellung von 364 mit 4 Stellen: X'00 36 4F'
INTEGER	4 Byte	Binärdarstellung, Darstellung des Vorzeichens im ersten Bit (Darstellung negativer Zahlen als Zweierkomplement). Beispiel Darstellung von 364 als X'00 00 01 6C' Darstellung von -364 als X'FF FF FE 94'
SMALLINT	2 Byte	Binärdarstellung, Darstellung des Vorzeichens im ersten Bit (Darstellung negativer Zahlen als Zweierkomplement). Beispiel Darstellung von 364 als X'01 6C' Darstellung von -364 als X'FE 94'

Tabelle 4: Daten in der Eingabedatei von LOAD bzw. in der Ausgabedatei von UNLOAD

(Teil 1 von 2)

Datentyp	Länge des Werts in der Datei	Standarddarstellung in der Ein- und Ausgabedatei
REAL	4 Byte	Darstellung gemäß BS2000-Konventionen, d.h.: Binärdarstellung des Exponenten im ersten Byte mit Darstellung des Vorzeichens des Exponenten im ersten Bit (Darstellung negativer Zahlen als Zweierkomplement).  Binärdarstellung der Mantisse in den Bytes 2 bis 4. Darstellung des Vorzeichens der Mantisse im ersten Bit von Byte 2 (Darstellung negativer Zahlen als Zweierkomplement).
DOUBLE PRECISION	8 Byte	Darstellung gemäß BS2000-Konventionen, d.h.: Binärdarstellung des Exponenten im ersten Byte mit Darstellung des Vorzeichens des Exponenten im ersten Bit (Darstellung negativer Zahlen als Zweierkomplement).  Binärdarstellung der Mantisse in den Bytes 2 bis 8. Darstellung des Vorzeichens der Mantisse im ersten Bit von Byte 2 (Darstellung negativer Zahlen als Zweierkomplement).
DATE	6 Byte	Jeweils zwei Bytes werden zusammengefasst für die Darstellung von Jahr, Monat, Tag und sind als SMALLINT zu interpretieren. Beispiel DATE'1994-06-08' wird dargestellt als X'07CA 0006 0008'
TIME(3)	8 Byte	Jeweils zwei Bytes werden zusammengefasst für die Darstellung von Stunde, Minute, Sekunde, Millisekunde und sind als SMALLINT zu interpretieren. Beispiel TIME'13:57:19.210' wird dargestellt als X'000D 0039 0013 00D2'
TIMESTAMP(3)	14 Byte	Die ersten 6 Byte bedeuten DATE, die restlichen 8 Byte bedeuten TIME(3).

Tabelle 4: Daten in der Eingabedatei von LOAD bzw. in der Ausgabedatei von UNLOAD

(Teil 2 von 2)

## 2.6 Bandsicherungen mit HSMS und ARCHIVE

SESAM-Sicherungsbestände auf Magnetbandkassette können Sie mit der Utility-Anweisung COPY ... USING DIRECTORY über die Softwareprodukte HSMS oder ARCHIVE erzeugen.

Wenn eine HSMS-Parameterdatei eingerichtet ist, dann wird der für COPY ... USING DIRECTORY angegebener Name als HSMS-Archiv betrachtet. Andernfalls erfolgt die Sicherung mit ARCHIVE.

Beim Einspielen der Sicherung über RECOVER oder CREATE REPLICATION wird das richtige Verfahren durch die Metadaten bestimmt, die von COPY in der CAT-REC-Datei bzw. im Catalog-Space hinterlegt worden sind.

Grundlegende Informationen zur Erstellung von SESAM-Sicherungsbeständen finden Sie im „[Basishandbuch](#)“.

### 2.6.1 HSMS-Lauf bei COPY und RECOVER steuern

Um HSMS für SESAM/SQL nutzen zu können, müssen Sie folgende Schritte durchführen:

- Einrichten eines HSMS-Archivs  
Über die Attribute des HSMS-Archivs werden die Eigenschaften des Archivs eingestellt.
- Anlegen einer SESAM-spezifischen HSMS-Parameterdatei  
Mit Hilfe der HSMS-Parameterdatei werden Parameter übermittelt, die über die HSMS-Archiv-Attribute nicht angesprochen werden können.  
Sollen nur Standardeinstellungen verwendet werden, muss eine leere HSMS-Parameterdatei angelegt werden, damit HSMS als Sicherungsverfahren ausgewählt wird.

HSMS unterstützt das System der Miteigentümerschaft des Softwareprodukte SECOS, mit dessen Hilfe Sie Zugriffsrechte auf fremde Benutzerkennungen verwalten können (siehe das SECOS-Handbuch „[Security Control System - Access Control](#)“).

### 2.6.1.1 HSMS-Archiv für SESAM/SQL einrichten

Das HSMS-Archiv muss für die Nutzung mit SESAM/SQL folgenden Anforderungen genügen:

- Das HSMS-Archiv soll ausschließlich als Backup-Archiv dienen.
- Die Ausführung der HSMS-Anweisung BACKUP-FILES und RESTORE-FILES darf nicht auf bestimmte Bandverarbeitungszeiten beschränkt sein.
- Die Directory-Datei des HSMS-Archivs darf nicht mit dem Konvertierungsprogramm DIRCONV bearbeitet werden, sonst könnten die CATIDs im HSMS-Archiv verändert werden.  
Eine Ausnahme bildet der Umzug von einem SF-Pubset auf ein SM-Pubset. Hier ist der Einsatz von DIRCONV notwendig (siehe [Abschnitt „Änderung der Benutzerkennung, des Pubsets oder des Systems“ auf Seite 51](#)).

Werden diese Bedingungen nicht beachtet, können unter Umständen Sicherungen im HSMS-Archiv nicht wieder gefunden werden.

Es ist empfehlenswert, ein privates HSMS-Archiv für die Nutzung durch SESAM/SQL einzurichten. Auf diese Weise können die Eigenschaften des HSMS-Archivs bedarfsgerecht und unabhängig von anderen Anforderungen auf den Einsatz von SESAM/SQL zugeschnitten werden.

Die Systemeinstellungen von HSMS dürfen den Anforderungen von SESAM/SQL nicht entgegenstehen. Bandzugriffe, die sich auf das für SESAM-Sicherungen eingerichtete HSMS-Archiv beziehen, müssen jederzeit zugelassen werden. Dazu muss der HSMS-Verwalter mit der HSMS-Anweisung MODIFY-TAPE-CONTROL folgende Einstellungen vornehmen:

```
MODIFY-TAPE-CONTROL ARCHIVE-NAME = <hsms_archive_name>  
    ,WRITE-CONTROL = *PROCESS-REQUESTS, READ-CONTROL = *PROCESS-REQUESTS
```

Der Name *hsms\_archive\_name* hat eine Länge von maximal 22 Zeichen. Ist die Bandverarbeitung auf einen bestimmten Zeitraum beschränkt und wird außerhalb dieses Zeitraumes die Sicherung eines Spaces veranlasst, bleibt der Space bis zum Abschluss der Sicherung gesperrt. Von der Sperrung betroffen sind DML-, DDL- und Utility-Zugriffe.

### HSMS-Archiv anlegen mit CREATE-ARCHIVE

Mit der HSMS-Anweisung CREATE-ARCHIVE wird ein HSMS-Archiv angelegt. Das HSMS-Archiv muss in der DBH-Kennung angelegt werden.

Im Folgenden werden alle Attribute von CREATE-ARCHIVE aufgeführt, die für ein SESAM/SQL geeignetes Backup-Archiv eine Bedeutung haben.

Es wird darauf hingewiesen, welche Parameter angegeben werden müssen und wo fest einzustellende Parameterwerte einzutragen sind. Eine vollständige Beschreibung der Operationen finden Sie im Handbuch „[HSMS \(BS2000\)](#)“.

#### CREATE-ARCHIVE

```

ARCHIVE-NAME = <filename 1..22 without cat-id>
,ENVIRONMENT = *STD
,OWNER-FIELD = *NONE / <c-string 1..54>
,ALLOWED-USAGE = *BACKUP(...)
    *BACKUP(...)
        | SAVE-FILE-STRUCTURE = *SEVERAL-SVID
,USER-ACCESS = *ALL-USERS(...)
    *ALL-USERS(...)
        | ACCESS = *READ / *WRITE
,DIRECTORY-NAME = <filename 1..54>(...)
    <filename 1..54>(...)
        | NEW-DIRECTORY = *NO / *YES
,RETENTION-PERIOD = <integer 0..32767 days>
,COMPRESS-FILES = *NO / *YES
,S2-DEVICE-TYPE = *STD / <c-string 1..8>
,LOCATION = *ANY / <alphanum-name 1..8>
,OPERATION-CONTROL = *PARAMETERS(...)
    *PARAMETERS(...)
        | PARALLEL-RUNS = 1 / <integer 1..16>
        | ,WRITE-CHECKPOINTS = *YES / *NO
        | ,OPERATOR-INTERACTION = *NOT-ALLOWED / *ALLOWED
,TAPE-CONTROL = *PARAMETERS(...)
    *PARAMETERS(...)
        | NEW-STD-SAVE-FILE = *IN-PERIODS(...)
            | *IN-PERIODS(...)
                | CONTINUATION-PERIOD = <integer 1..31 days>
        | ,BLOCKING-FACTOR = *STD / <integer 1..15 2Kbyte> / *MAX
        | ,STREAM-MODE = *NO / *YES
        | ,UNLOAD-TAPE = *NO / *YES

```

## Operandenbeschreibung

### **ARCHIVE-NAME = <filename 1..22 without cat-id>**

Name des HSMS-Archivs. Er wird bei COPY ... USING DIRECTORY angegeben. Ohne Benutzerkennung ist die Länge des Namens auf 12 Zeichen beschränkt.

Voreingestellt für die Benutzerkennung ist die Benutzerkennung, unter der das HSMS-Archiv eingerichtet worden ist. Diese Kennung muss die DBH-Kennung sein. Eine fremde Benutzerkennung kann nur ein HSMS-Verwalter angeben.

### **ENVIRONMENT = \*STD**

Bezeichnet die HSMS-Umgebung, in der das Archiv definiert werden soll.

Das Attribut muss auf \*STD eingestellt werden. \*STD entspricht der Standardumgebung, in der der Benutzer im Benutzerkatalog eingetragen ist.

### **OWNER-FIELD = \*NONE / <c-string 1 .. 54>**

Hier kann der Anwender einen Kommentar zu dem einzurichtenden HSMS-Archiv eingeben.

Der Kommentar wird bei der HSMS-Anweisung SHOW-ARCHIVE-ATTRIBUTES mit INFORMATION=\*FULL ausgegeben.

### **ALLOWED-USAGE = \*BACKUP (...)**

Das HSMS-Archiv dient ausschließlich als Backup-Archiv für BS2000-Dateien.

### **SAVE-FILE-STRUCTURE = \*SEVERAL-SVID**

HSMS legt Sicherungsdateien in SEVERAL-SVID-Struktur an.

Eine Sicherungsdatei in SEVERAL-SVID-Struktur kann mehrere Sicherungsversionen aufnehmen. Die aktuelle Sicherungsdatei wird solange fortgeschrieben, bis ein Wechsel der Sicherungsdatei veranlasst wird.

Die Sicherungsversions-Identifikation (SVID) kennzeichnet eine Sicherungsversion und wird von HSMS intern festgelegt.

Die Sicherungsdateistruktur muss mit SEVERAL-SVID eingestellt werden.

### **USER-ACCESS = \*ALL-USERS (...)**

Erlaubt das Einspielen von Sicherungen in andere Benutzerkennungen, die nicht Archiveigentümer sind.

### **ACCESS = \*READ**

Erlaubt den lesenden Zugriff auf Sicherungen, die in einer anderen Benutzerkennung liegen als das HSMS-Archiv. Die Benutzerkennung muss in der Umgebung sein, die mit dem Operanden ENVIRONMENT für das HSMS-Archiv vereinbart worden ist.

### **ACCESS = \*WRITE**

Erlaubt den lesenden und schreibenden Zugriff auf Sicherungen, die in einer anderen Benutzerkennung liegen als das HSMS-Archiv. Die Benutzerkennung muss in der Umgebung sein, die mit dem Operanden ENVIRONMENT für das HSMS-Archiv vereinbart worden ist.

**DIRECTORY-NAME = <filename 1..54> (...)**

Hier werden Informationen über die im HSMS-Archiv verwalteten Sicherungsbestände gespeichert. Der Name wird gegebenenfalls durch die Benutzerkennung und die Standard-Katalogkennung des Aufrufers ergänzt. Eine fremde Benutzerkennung darf nur ein HSMS-Verwalter angeben.

**NEW-DIRECTORY = \*YES**

HSMS legt das Archivverzeichnis neu an. Die Datei für das neue Archivverzeichnis muss leer sein. Sie darf aber katalogisiert sein.

**NEW-DIRECTORY = \*NO**

Ein bestehendes Verzeichnis wird unter HSMS-Verwaltung genommen. Das Archivverzeichnis muss zuvor mit der HSMS-Funktion IMPORT-FILES bereitgestellt worden sein.

Dieser Fall tritt ein, wenn eine Sicherung auf einem anderen System eingespielt oder ein Archivverzeichnis in das HSMS-Archiv übernommen werden soll.

**RETENTION-PERIOD = <0..32767 days>**

Anzahl der Tage, die für die physikalische Schutzfrist der Sicherungsdateien standardmäßig gelten soll.

**COMPRESS-FILES = \*NO / \*YES**

Gibt an, ob die Daten vor dem Schreiben auf den Ausgabeträger komprimiert werden sollen. Für die Nutzung mit SESAM/SQL wird \*YES empfohlen.

**S2-DEVICE-TYPE = \*STD / <c-string 1..8>**

Legt den Gerätetyp fest, der für Sicherungen in dieses HSMS-Archiv auf der Speicher-ebene S2 benutzt wird.

**LOCATION = \*ANY / <alphanum-name 1..8>**

Lagerort des Datenträgers für die Sicherung. Der Lagerort muss MAREN bekannt sein. Ist MAREN nicht im Einsatz, muss \*ANY angegeben werden. In diesem Fall wird kein Lagerort verwendet, um die Datenträger auszuwählen.

**OPERATION-CONTROL = \*PARAMETERS(...)**

Parameter, die für den Ablauf der Verarbeitung relevant sind.

**PARALLEL-RUNS = 1 / <integer 1..16>**

Anzahl der gleichzeitig ablaufenden Sicherungstasks.

Während der Sicherung muss für jede Task ein Bandgerät zur Verfügung stehen.

**WRITE-CHECKPOINTS = \*YES / \*NO**

Legt fest, ob während der Verarbeitung Wiederaufsetzpunkte in die ARCHIVE-Checkpointdatei geschrieben werden sollen. Diese ermöglichen bei einem Abbruch einen späteren Wiederanlauf.

Diese Funktion kann bei der Sicherung von Datenbankdateien, die auf Additional-Mirror-Units oder Clone-Units liegen, genutzt werden. In diesem Fall sollte \*YES (Voreinstellung) angegeben werden. Sonst geben Sie \*NO an.

**OPERATOR-INTERACTION = \*NOT-ALLOWED / \*ALLOWED**

Legt fest, ob Meldungen, die eine Antwort des Operators erfordern, am Bedienplatz ausgegeben werden sollen.

Bei \*NOT-ALLOWED führt HSMS eine Standardbehandlung durch.

**TAPE-CONTROL = \*PARAMETERS(...)**

Parameter, die für das Schreiben auf Magnetbandkassette relevant sind.

**NEW-STD-SAVE-FILE = \*IN-PERIODS (...)****CONTINUATION-PERIOD = <integer 1..31 days>**

Fortsetzungsperiode, nach der HSMS automatisch eine neue Sicherungsdatei anlegt. Geben Sie in der HSMS-Parameterdatei für den Parameter

SAVE-FILES = \*STD an, wird die Standard-Sicherungsdatei von HSMS nach Ablauf dieser Periode gewechselt.

**BLOCKING-FACTOR = \*STD / <integer 1..15 2Kbyte> / \*MAX**

Blockungsfaktor, mit dem die Sicherungsdatei auf Magnetbandkassette geschrieben wird. Es wird die Anzahl der 2-Kbyte-Blöcke (PAM-Seiten) angegeben, die mit einer Ein-/Ausgabe auf Magnetbandkassette geschrieben werden. Der Wert ist abhängig vom Gerätetyp.

\*MAX wählt den Blockungsfaktor, der in der aktuellen BS2000-Version maximal möglich ist. Derzeit ist das der Wert 128 (Blockgröße 256 Kbyte). Sicherungsdateien, die mit dem maximalen Wert einer BS2000-Version geschrieben werden, können in einer niedrigeren BS2000-Version, die diesen Blockungsfaktor nicht unterstützt, nicht gelesen werden.

Für die Nutzung mit SESAM/SQL wird \*MAX empfohlen.

**STREAM-MODE = \*NO / \*YES**

Gibt an, ob die Magnetbandkassette im Streaming-Modus beschrieben werden soll, falls dieser vom betreffenden Bandgerät unterstützt wird.

Für die Nutzung mit SESAM/SQL wird \*YES empfohlen.

**UNLOAD-TAPE = \*NO / \*YES**

Gibt an, ob eine Magnetbandkassette nach dem Ende der Verarbeitung entladen werden soll. Die Einstellung wirkt sich nur auf die Anweisung RESTORE-FILES aus, da der Parameter UNLOAD-TAPE in der Anweisung BACKUP-FILES von SESAM/SQL versorgt wird.

Für die Nutzung mit SESAM/SQL wird \*YES empfohlen.

### **Musterprozedur zum Einrichten eines HSMS-Archivs**

Mit der Musterprozedur SESAM.P.CREATE.HSMS-ARCHIVE können Sie ein für SESAM/SQL geeignetes HSMS-Archiv einrichten. Die Prozedur wird in der Bibliothek SIPANY.SESAM-SQL.090.TOOLS bereitgestellt.

Sie können diejenigen Eigenschaften des Archivs über Parameter festlegen, die im Rahmen der SESAM/SQL-spezifischen Vorgaben wählbar sind. Dabei müssen Sie die Einschränkungen berücksichtigen, die im [Abschnitt „HSMS-Archiv für SESAM/SQL einrichten“ auf Seite 37](#) beschrieben werden.

### **Attribute eines HSMS-Archivs ändern mit MODIFY-ARCHIVE-ATTRIBUTES**

Mit der Anweisung MODIFY-ARCHIVE-ATTRIBUTES werden die Attribute eines HSMS-Archivs geändert. Dabei gelten folgende Einschränkungen:

- Der Archivtyp kann nicht geändert werden.
- Soll ein Archivverzeichnis für eine andere HSMS-Grundfunktion genutzt werden, muss die alte Archivdefinition gelöscht und das Archivverzeichnis in ein neues HSMS-Archiv übernommen werden.

Die Attribute des HSMS-Archivs dürfen nur im Rahmen der für die einzelnen Operanden aufgeführten Möglichkeiten geändert werden, wenn das HSMS-Archiv weiterhin für SESAM/SQL nutzbar sein soll.

### **HSMS-Archiv-Verzeichnis ändern mit MODIFY-ARCHIVE**

Mit der Anweisung MODIFY-ARCHIVE wird ein HSMS-Archiv-Verzeichnis geändert. MODIFY-ARCHIVE bietet folgende Möglichkeiten:

- Ändern der Zusatzinformationen einer Sicherungsversion
- Löschen von Sicherungsdateien
- Verwalten des Datenträger-Pools

Es können nur ganze Sicherungsdateien gelöscht werden. Dabei werden alle Sicherungsversionen in einer Sicherungsdatei mit dieser zusammen gelöscht.

### 2.6.1.2 HSMS-Parameterdatei erstellen

Die HSMS-Parameterdatei ist eine SESAM-spezifische SAM-Datei mit variablem Satzformat zur Parametrisierung des HSMS-Aufrufs. Sie muss auf der BS2000-Benutzerkennung des DBH angelegt werden. Auch das HSMS-Archiv selber muss auf der Kennung des DBH liegen.

Die Parameterdatei darf ein Schreibkennwort haben. Ein Lesekennwort ist jedoch nicht erlaubt, weil die Parameterdatei von der Service-Task gelesen werden muss.

Die HSMS-Parameterdatei wird bei jeder COPY-, RECOVER- bzw. CREATE REPLICATION-Anweisung geöffnet, gelesen und vor dem HSMS-Aufruf wieder geschlossen. D.h., Sie können die Datei zwischen zwei COPY- bzw. RECOVER-Anweisungen verändern.

Der Name der HSMS-Parameterdatei muss sich entweder aus dem Pfadnamen des HSMS-Archivs oder dem Namen der DBH-Sitzung ableiten.

*hsms\_archive\_name*.HSMS-PAR

oder

SESAM*kn*.HSMS-PAR

Dabei bedeuten:

*hsms\_archive\_name*

Name des HSMS-Archivs, den Sie in der Klausel USING DIRECTORY in der COPY-Anweisung angeben.

Das Format entspricht dem von Dateinamen ohne Katalogkennung, die Länge ist aber auf 22 Zeichen (ohne Benutzerkennung auf 12 Zeichen) beschränkt.

*k* Konfigurationsname

*n* DBH-Name

Zunächst sucht SESAM/SQL die Datei *hsms\_archive\_name*.HSMS-PAR, danach sessionübergreifend die Datei SESAM*kn*.HSMS-PAR. Ist keine der beiden Dateien vorhanden, erfolgt die Bandsicherung mit ARCHIVE.

Sollen nur Standardeinstellungen verwendet werden, muss eine leere HSMS-Parameterdatei angelegt werden, damit HSMS als Sicherungsverfahren ausgewählt wird.

## Übersicht der zulässigen HSMS-Parameter

In der HSMS-Parameterdatei können Sie Parameter angeben, die das Verhalten von BACKUP-FILES (für COPY-Anweisungen) und RESTORE-FILES (für RECOVER oder CREATE REPLICATION-Anweisungen) steuern.

Folgende HSMS-Parameter dürfen Sie in der HSMS-Parameterdatei angeben:

Parameter	Bedeutung
SAVE-FILE	Unterteilung der Sicherungsdateien festlegen
SAVE-CATREC-COPY	CAT-REC-Kopie sichern
SAVE-ACL	ACL-Einträge sichern
SAVE-DIRECTORY	Archivverzeichnis sichern
HSMS-ARCHIVE-USERID	aktuelle Benutzerkennung des HSMS-Archivs
REQUEST-NAME	Name zum Ansprechen des Auftrags
PARALLEL-RUNS	Anzahl der gleichzeitig ablaufenden Sicherungstasks
REORGANIZE-SPACE	Datenträger reorganisieren
MAIL	Report mit E-Mail versenden

Tabelle 5: Zulässige HSMS-Parameter

Für BACKUP-FILES-Anweisungen werden nur die Angaben SAVE-FILE, SAVE-CATREC-COPY, SAVE-ACL, SAVE-DIRECTORY, PARALLEL-RUNS, REQUEST-NAME und MAIL ausgewertet.

Für RESTORE-FILES-Anweisungen werden nur die Angaben HSMS-ARCHIVE-USERID, PARALLEL-RUNS, REORGANIZE-SPACE, REQUEST-NAME und MAIL ausgewertet.

Allen Parametern, die nicht in der Datei angegeben werden, wird der Standardwert zugewiesen. Ist die gesamte Parameterdatei leer, gelten die Standardwerte.

Für den Parameter PARALLEL-RUNS gilt die Voreinstellung der HSMS-Archivdefinition. Der Parameter HSMS-ARCHIV-USER-ID hat keinen Standardwert. Er wird nur angegeben, wenn er sich gegenüber dem Zeitpunkt der Sicherungserstellung geändert hat.

## Beschreibung der HSMS-Parameter

Die HSMS-Parameterdatei kann Einträge für einen HSMS-Lauf bei COPY, RECOVER oder CREATE REPLICATION enthalten. Die Einträge werden von SESAM/SQL ausgewertet, aber ihre Konsistenz nicht geprüft.

Folgende Parameter und Parameterwerte kann die HSMS-Parameterdatei enthalten:

---

```
SAVE-FILE = STD / NEW / CONTINUE
SAVE-CATREC-COPY = YES / NO
SAVE-ACL = YES / NO
SAVE-DIRECTORY = YES / NO
HSMS-ARCHIVE-USERID = <name 1..8>
REQUEST-NAME = <name 1..8>
PARALLEL-RUNS = <integer 1..16>
REORGANIZE-SPACE = YES / NO
MAIL = YES / NO
```

---

### SAVE-FILE

Mit dem Parameter SAVE-FILE der Parameterdatei wird der Operand SAVE-FILE der BACKUP-FILES-Anweisung gesteuert (siehe Handbuch „[HSMS \(BS2000\)](#)“).

HSMS führt nach jedem Auftrag einen Spulvorgang durch, der nicht von der Anwendung unterdrückt werden kann. Mit dem Parameter SAVE-FILE können Sie eine Einstellung zugunsten der Bandausnutzung oder der Performance wählen.

Bei der Standard-Einstellung entstehen Wartezeiten durch Spulvorgänge. Diese Spulvorgänge treten auf, wenn eine Sicherungsdatei mit der CAT-REC-Kopie fortgeschrieben wird, nachdem die Spaces gesichert worden sind. Auf diese Weise wird die Kapazität der Bänder gut ausgenutzt.

Um Wartezeiten zu vermeiden, können Sie für jeden an HSMS erteilten Sicherungsauftrag eine neue Sicherungsdatei angelegen. Bei einer Sicherung einschließlich der CAT-REC-Kopie werden mindestens zwei Bänder beschrieben. Die Kapazität der Bänder wird in diesem Fall schlecht ausgenutzt.

**SAVE-FILE = STD**

Die SESAM-Sicherungen werden in die Standard-Sicherungsdatei geschrieben. Die Standard-Sicherungsdatei wird fortgeschrieben, bis durch HSMS ein periodischer Wechsel initiiert wird.

BACKUP-FILES wird von SESAM/SQL für eine Sicherung einschließlich der CAT-REC-Kopie mit folgenden Einstellungen des Operanden SAVE-FILE aufgerufen:

<i>Operand SAVE-FILE</i>	<i>für die Sicherung der</i>
SAVE-FILE = *STD	Spaces
SAVE-FILE = *STD	CAT-REC-Kopie

**SAVE-FILE = NEW**

Für einen SESAM-Sicherungsauftrag werden zwei Sicherungsdateien genutzt.

In eine Sicherungsdatei werden Spaces geschrieben und in eine weitere die CAT-REC-Kopie.

BACKUP-FILES wird von SESAM/SQL für eine Sicherung einschließlich der CAT-REC-Kopie mit folgenden Einstellungen des Operanden SAVE-FILE aufgerufen:

<i>Operand SAVE-FILE</i>	<i>für die Sicherung der</i>
SAVE-FILE = *NEW	Spaces
SAVE-FILE = *NEW	CAT-REC-Kopie

**SAVE-FILE = CONTINUE**

Jeweils ein SESAM-Sicherungsauftrag, bestehend aus den Spaces und der CAT-REC-Kopie wird in einer Sicherungsdatei gespeichert.

BACKUP-FILES wird von SESAM/SQL für eine Sicherung einschließlich der CAT-REC-Kopie mit folgenden Einstellungen des Operanden SAVE-FILE aufgerufen:

<i>Operand SAVE-FILE</i>	<i>für die Sicherung der</i>
SAVE-FILE = *NEW	Spaces
SAVE-FILE = *CONTINUE (SAVE-FILE-ID = *LATEST)	CAT-REC-Kopie

**SAVE-CATREC-COPY**

Mit diesem Parameter wird die Sicherung der CAT-REC-Kopie gesteuert.

**SAVE-CATREC-COPY = YES**

Die CAT-REC-Kopie wird mitgesichert. Eine CAT-REC-Kopie entsteht nur bei COPY CATALOG oder COPY CATALOG\_SPACE.

**SAVE-CATREC-COPY = NO**

Die CAT-REC-Kopie wird nicht mitgesichert.

**SAVE-ACL**

Mit diesem Parameter wird das Sichern der ACL-Einträge (bis SECOS V3.0) gesteuert. Die ACL-Einträge müssen mitgesichert werden, wenn die Sicherung auf einem fremden System wieder eingespielt werden soll.

**SAVE-ACL = YES**

Die ACL-Einträge werden mitgesichert.

**SAVE-ACL = NO**

Die ACL-Einträge werden nicht mitgesichert.

**SAVE-DIRECTORY**

Mit diesem Parameter wird das Sichern des Archivverzeichnisses gesteuert. Das Archivverzeichnis muss mitgesichert werden, wenn die Sicherung auf einem fremden System eingespielt werden soll.

**SAVE-DIRECTORY = YES**

Das Archivverzeichnis wird mitgesichert.

**SAVE-DIRECTORY = NO**

Das Archivverzeichnis wird nicht mitgesichert.

**HSMS-ARCHIVE-USERID = <name 1..8>**

Mit diesem Parameter wird die aktuelle Benutzerkennung des HSMS-Archivs verwaltet. Er muss angegeben werden, wenn eine Sicherung eingespielt werden soll, bei der das HSMS-Archiv zum Zeitpunkt der Sicherung auf einer anderen Benutzerkennung lag.

Der Zeichenvorrat für die Benutzerkennung des HSMS-Archivs ist A ... Z, 0 ... 9, \$, #, @. Der Name kann frei gewählt werden, darf jedoch nicht mit einer Zahl beginnen.

**REQUEST-NAME = <name 1..8>**

Mit diesem Parameter wird der Auftragsname festgelegt, über den der Auftrag bei den HSMS-Anweisungen zur Auftragsverwaltung angesprochen werden kann.

Der Wert des Parameters REQUEST-NAME der HSMS-Parameterdatei wird in den Operanden REQUEST-NAME der BACKUP-FILES Anweisung übernommen. Wenn der Parameter REQUEST-NAME in der HSMS-Parameterdatei nicht angegeben ist, dann wird der REQUEST-NAME=\*STD gesetzt.

Wenn der Datenträger über MAREN zugewiesen wird, kann der Benutzer durch einen optionalen Rep den MAREN-Lagerort angeben, von dem der Datenträger kommen muss. Wenn dieser optionale HSMS-Rep installiert ist, wird REQUEST-NAME als MAREN-Lagerort angesehen und an MAREN übergeben.

Siehe auch Handbuch „[HSMS \(BS2000\)](#)“, Abschnitt „Datenträgerzuweisung über MAREN“.

PARALLEL-RUNS = <integer 1..16>

Mit diesem Parameter wird die Anzahl der gleichzeitig ablaufenden Sicherungstasks festgelegt. Für jede Task muss ein Bandgerät zur Verfügung stehen.

Der Wert des Parameters PARALLEL-RUNS in der HSMS-Parameterdatei wird in den Operanden PARALLEL-RUNS der BACKUP-FILES- bzw. RESTORE-FILES-Anweisung übernommen.

Ist der Parameter PARALLEL-RUNS in der HSMS-Parameterdatei nicht angegeben, gilt der entsprechende Wert aus der Definition des HSMS-Archivs.

REORGANIZE-SPACE = YES / NO

Dieser Parameter bestimmt, ob die Dateien vor dem Zurückschreiben gelöscht werden oder auf der Platte stehen bleiben.

Der Wert des Parameters REORGANIZE-SPACE in der HSMS-Parameterdatei wird in den Operanden REORGANIZE-SPACE der RESTORE-FILES-Anweisung übernommen.

Ist der Parameter REORGANIZE-SPACE in der HSMS-Parameterdatei nicht angegeben, wird der Operand REORGANIZE-SPACE in der RESTORE-FILES-Anweisung mit dem Wert \*NO versorgt.

MAIL = YES / NO

Dieser Parameter bestimmt, ob der Report einer BACKUP-FILES oder RESTORE-FILES-Anweisung in eine Datei geschrieben wird (MAIL=NO) oder ob er als Anhang per E-Mail versandt wird (MAIL=YES). Die Empfängeradresse der E-Mail muss im Benutzereintrag des Aufrufers eingetragen sein.

## Syntaxregeln

- In jeder Zeile der Parameterdatei darf nur ein Parameter in der Form *parameter=wert* angegeben werden.
- Die Schlüsselwörter (Parameterbezeichnungen) müssen in voller Länge angegeben werden.
- Eingestreute Leerzeichen sind erlaubt.
- Zeilen, die mit „\*“ in Spalte 1 beginnen, werden als Kommentar behandelt.
- Jeder Parameter darf maximal einmal angegeben werden.

## Allgemeine Regeln

- Die Parameterangaben werden jedesmal ausgewertet, wenn bei einer COPY-, RECOVER- oder CREATE REPLICATION-Anweisung Bandsicherungen betroffen sind.

In folgenden Fällen wird die COPY-, RECOVER- bzw. CREATE REPLICATION-Anweisung mit einer Fehlermeldung abgebrochen:

- Wenn die HSMS-Parameterdatei keine SAM-Datei ist.
- Wenn die HSMS-Parameterdatei wegen DMS-Fehler nicht geöffnet, geschlossen oder gelesen werden kann.
- Bei Parameterangaben, die nicht erkannt werden (z.B. falsch geschriebene Schlüsselwörter).  
Im Meldungstext werden die Zeilennummer, in der die falsche Parameterangabe steht, und der Dateiname der HSMS-Parameterdatei ausgegeben.
- Bei der Angabe falscher Werte.  
Im Meldungstext werden die Nummer der Zeile, in der der falsche Wert steht, sowie der Dateiname der HSMS-Parameterdatei ausgegeben.

### *Beispiel einer HSMS-Parameterdatei*

```
* HSMS PARAMETER FILE SESAM/SQL
* permitted specifications:
*
* SAVE-FILE           = STD / NEW / CONTINUE
* SAVE-CATREC-COPY    = YES / NO
* SAVE-ACL             = YES / NO
* SAVE-DIRECTORY      = YES / NO
* HSMS-ARCHIVE-USERID = <userid hsms archive>
* REQUEST-NAME        = <request name>
* PARALLEL-RUNS       = <integer 1 ... 16>
* REORGANIZE-SPACE    = YES / NO
* MAIL                = YES / NO
*
SAVE-FILE=STD
SAVE-CATREC-COPY=NO
SAVE-ACL=NO
SAVE-DIRECTORY=NO
MAIL=YES
*
```

## Nutzung der HSMS-Anweisungen BACKUP-FILES und RESTORE-FILES durch SESAM/SQL

Wesentlich für die Nutzung eines HSMS-Archivs für SESAM-Sicherungsbestände sind die HSMS-Anweisungen BACKUP-FILES und RESTORE-FILES (siehe Handbuch „[HSMS \(BS2000\)](#)“).

Die BACKUP-FILES-Anweisung steuert das Schreiben der Spaces und der CAT-REC-Kopie in das HSMS-Archiv bei der SESAM/SQL-Anweisung COPY.

Das Schreiben einer Sicherung in ein HSMS-Archiv erfolgt in maximal zwei Aufrufen von HSMS. Dabei sichert ein Aufruf den Catalog-Space und die Anwender-Spaces und der andere Aufruf sichert die CAT-REC-Kopie.

Die RESTORE-FILES-Anweisung steuert das Lesen des Catalog-Space und der Anwender-Spaces aus einem HSMS-Archiv.

Das Lesen einer Sicherungskopie aus einem HSMS-Archiv erfolgt in mehreren Aufrufen von HSMS. Die Anzahl der Aufrufe ist abhängig von der Zusammenstellung der einzuspielenden Spaces:

- Einspielen des Catalog-Space
- Einspielen der Anwender-Spaces  
(Bei Verteilung der Spaces auf unterschiedliche Pubsets sind es mehrere Aufrufe.)

Der Name der Protokolldatei der beiden Anweisungen lautet

<dbh\_kennung>.<hsms\_archive\_name>.SYSLSL. Die Protokolldatei wird fortgeschrieben.

HSMS legt die Protokolldatei nur an, wenn die von SESAM/SQL formulierten Anweisungen zu Aktionen von HSMS führen.

Die Fehlermeldungen sind in den Protokollen der Service-Task zu finden, wenn HSMS keine Protokolldatei angelegt hat, weil Anweisungen von HSMS abgewiesen wurden.

Eine Anweisung wird von HSMS beispielsweise dann abgewiesen, wenn ein nicht vorhandenes Archiv angegeben wird.

### 2.6.1.3 Löschen von HSMS-Aufträgen über die HSMS-Benutzeroberfläche

Solange ein HSMS-Auftrag läuft befindet sich die Service-Task im Zustand TPR, wenn

- ein SESAM-Sicherungsbestand mit HSMS erzeugt wird (COPY-Anweisung),
- ein SESAM-Sicherungsbestand mit HSMS eingespielt wird (RECOVER-Anweisung).

Ein solcher HSMS-Auftrag kann über die Benutzeroberfläche von HSMS mit der Anweisung DELETE-REQUESTS aus der Auftragsdatei gelöscht werden.

Dieses Löschen wirkt aber erst, wenn der HSMS-Auftrag beendet ist und die Service-Task sich wieder im Zustand TU befindet.

Aus SESAM/SQL-Sicht ist der Auftrag dann abgebrochen worden.

### 2.6.1.4 Änderung der Benutzerkennung, des Pubsets oder des Systems

Dieser Abschnitt beschreibt, wie Sie bei folgenden Spezialfällen von RECOVER und CREATE REPLICATION vorgehen müssen:

- Umzug eines HSMS-Archivs auf eine andere Benutzerkennung
- Einspielen von Sicherungen auf ein anderes Pubset oder eine andere Benutzerkennung
- Einspielen von Sicherungen auf einen anderen System

Eine Beschreibung der auftretenden Parameter finden Sie in dem [Abschnitt „HSMS-Archiv für SESAM/SQL einrichten“ auf Seite 37](#) und dem [Abschnitt „HSMS-Parameterdatei erstellen“ auf Seite 43](#).

Zum Einspielen einer SESAM-Sicherung in eine andere Kennung oder auf einen anderen System kann eine mitgesicherte CAT-REC-Kopie verwendet werden.

Dazu muss die CAT-REC-Kopie mit der HSMS-Anweisung RESTORE-FILES in der DB-Kennung bereitgestellt und auf den Namen der CAT-REC-Datei

„<catalog.name>.CAT-REC“ umbenannt werden. Anschließend kann die Sicherungskopie mit RECOVER aus dem HSMS-Archiv eingespielt werden.

In der eingespielten CAT-REC-Datei besteht schon ein CAT-LOG-Eintrag, der nach der Sicherung angelegt wurde. Sollen die hier protokollierten Daten nicht nachgefahren werden, verwenden Sie die Anweisung RECOVER CATALOG TO zum Einspielen der Sicherung.

Alternativ können Sie den Eintrag auch mit Hilfe des Utility-Monitors löschen (siehe Handbuch [„Utility-Monitor“](#)).

Mit der HSMS-Anweisung SHOW-ARCHIVE können die im HSMS-Archiv vorhandenen Sicherungsversionen und die gesicherten Dateien aufgelistet werden. Die gewünschte CAT-REC-Kopie wird über den Zeitpunkt ihrer Entstehung ausgewählt.

## Miteigentümerschaft

Bei der Arbeit mit HSMS in Verbindung mit SECOS können Sie die Zugriffsrechte auf Dateien sehr detailliert durch die Definition einer Miteigentümerschaft steuern. Es wird zwischen Lese-, Schreib- und Ausführungsrechten unterschieden. Voraussetzung für die Nutzung der Miteigentümerschaft ist der Einsatz von SECOS und HSMS (siehe Handbücher „[Security Control System - Access Control](#)“ und „[HSMS \(BS2000\)](#)“).

Wenn Sie SESAM-Sicherungen mit HSMS verwalten, haben Sie durch das Einrichten einer Miteigentümerschaft die Möglichkeit, die Datenbank in eine andere Kennung als den DBH zu legen.

Näheres zum Thema „Datenbankdateien und Jobvariablen auf fremden Benutzerkennungen“ finden Sie im gleichnamigen Abschnitt im „[Basishandbuch](#)“.

## Umzug eines HSMS-Archivs auf eine andere Benutzerkennung

Beim Erstellen einer Sicherung mit COPY wird der Name des HSMS-Archivs zusammen mit der Benutzerkennung in den RECOVERY\_UNITS gespeichert. Beim Einspielen der Sicherung mit RECOVER oder CREATE REPLICATION wird das HSMS-Archiv unter dieser Benutzerkennung gesucht.

Sind das HSMS-Archiv und damit auch der DBH in der Zwischenzeit auf eine andere Benutzerkennung umgezogen, muss diese Kennung in der HSMS-Parameterdatei angegeben werden.

## Einspielen von Sicherungen auf ein anderes Pubset oder eine andere Benutzerkennung

Die Sicherungskopien werden im HSMS-Archiv mit vollqualifiziertem Dateinamen, d.h. mit Katalogkennung und Benutzerkennung, abgelegt.

Bei REORGANIZE-SPACE=NO werden die Sicherungen in die Benutzerkennung eingespielt, die für die Datenbank im Datenbankkatalog eingetragen ist. Dabei wird das Pubset verwendet, auf dem die Datei der Datenbank zum Zeitpunkt des Einspielens liegt.

Bei REORGANIZE-SPACE=YES oder wenn die Datenbank beim Einspielen nicht vorhanden ist, wird die Datenbank in die Benutzerkennung eingespielt, die im Datenbankkatalog für die Datenbank eingetragen ist. Dabei wird das Pubset gewählt, das im HSMS-Archiv für diese Datei hinterlegt ist.

Zum Einspielen einer Sicherung in eine andere Benutzerkennung benötigt die DBH-Kennung das Miteigentümerrecht auf folgenden Kennungen:

- die Kennung, auf der die Dateien der Datenbank während der Sicherung lagen
- die Kennung, auf der die Sicherung eingespielt werden soll

Es kann sinnvoll sein, die Miteigentümerschaft teilqualifiziert zu vergeben. Auf diese Weise wird sichergestellt, dass für alle betroffenen Dateien eine Miteigentümerschaft definiert ist.

Bei CREATE REPLICATION verwendet HSMS zum Einspielen der Dateien den ursprünglichen Namen und hängt das Suffix „REPL“ an.

In einem zweiten Schritt werden die Dateien von SESAM/SQL auf den Namen des Replikats umbenannt. Die neue Benutzerkennung muss der DBH-Kennung für alle in diesem Prozess verwendeten Dateinamen eine Miteigentümerschaft einräumen.

Ist die Datenbank beim Wiedereinspielen nicht vorhanden und soll eine Sicherung auf ein anderes Pubset eingespielt werden als dasjenige, in dem die Sicherung erfolgte, müssen die Dateien vor dem Einspielen per CREATE FILE-Kommando angelegt werden.

Wenn die einzelnen Spaces einer Datenbank beim Wiedereinspielen auf mehrere Pubsets verteilt werden sollen und diese Pubsets nicht diejenigen sind, auf denen die Sicherung erfolgte, müssen die Dateien vor dem Einspielen per CREATE-FILE-Kommando angelegt werden.

### **Umzug eines HSMS-Archivs von einem SF-Pubset auf ein SM-Pubset**

Soll das HSMS-Archiv von einem SF-Pubset auf ein SM-Pubset verlagert werden, müssen vor dem Einspielen der Sicherung folgende Schritte durchgeführt werden:

- Das HSMS-Archivverzeichnis wird auf dem SM-Pubset bereitgestellt.
- Im HSMS-Archivverzeichnis werden die Katalogkennungen mit Hilfe der DIRCONV-Anweisung RENAME-CATID auf die CATID des SM-Pubset umbenannt.
- Das HSMS-Archiv wird auf dem SM-Pubset neu eingerichtet mit den gleichen Attributen und dem gerade modifizierten HSMS-Archivverzeichnis.

Die Sicherung kann danach eingespielt werden, wie auf [Seite 52](#) beschrieben.

SESAM/SQL erkennt, dass der DBH jetzt in einer SM-Pubset Umgebung abläuft und berücksichtigt dieses beim Einspielen der Sicherung.

### **Datenbankdateien auf einem SM-Pubset**

Mit einer HSMS-Anweisung BACKUP-FILES können nur Dateien, die auf demselben SM-Pubset liegen, gesichert werden. Deshalb können mit einer SESAM-Anweisung COPY also nur Datenbankdateien gesichert werden, die auf demselben SM-Pubset liegen.

## Einspielen von Sicherungen auf einem anderen System

Sie können eine Sicherung auf einem anderen System einspielen, wenn das Archivverzeichnis, die CAT-REC-Kopie und die ACL-Einträge mitgesichert worden sind. Der Name der Kennung muss gleich bleiben, wenn eine Sicherung auf einem anderen System eingespielt wird.

Das HSMS-Archiv auf dem Ziel-System muss in zwei Schritten zugänglich gemacht werden, damit auf eine Sicherung des Archivs zugegriffen werden kann:

1. Das HSMS-Archivverzeichnis wird mit der HSMS-Anweisung IMPORT-FILES auf dem Ziel-System eingespielt:

```
IMPORT-FILES
  FILE-NAMES=:<catid>:<bs2000_benutzerkennung>.<hsms_directory_name>,
  NEW-FILE-NAMES = *BY-RULE(
    NEW-CATALOG-ID = <new_cat_id>),
  SAVE-FILE = *BY-VOLUME(
    SAVE-FILE-ID = <sicherungsversion>,
    VOLUMES = <vsn 1>, ...)
```

In obiger Anweisung kann auch der Opernd `FILE-NAMES=*DIRECTORY` verwendet werden.

Ändert sich das Pubset, muss bei `FILE-NAMES` das ursprüngliche Pubset und bei `NEW-CATALOG-ID` das aktuelle Pubset angegeben werden. Als `<sicherungsversion>` muss die zu restaurierende Sicherungsversion im Format `S.yymmdd.hhmmss` angegeben werden. Unter `<vsn1>`, ... werden die Archivnummern der Datenträger angegeben, die das ursprüngliche HSMS-Archiv enthalten.

2. Das HSMS-Archiv wird auf dem Ziel-System eingerichtet mit den gleichen Attributen wie auf dem Ausgangs-System und mit dem gerade eingespielten Archivverzeichnis.

```
CREATE-ARCHIVE
  ARCHIVE-NAME = <hsms_archiv_name>,
  ALLOWED-USAGE = *BACKUP(
    SAVE-FILE-STRUCTURE = *SEVERAL-SVID),
  DIRECTORY-NAME = <hsms_directory_name>(
    NEW-DIRECTORY = *NO),
  TAPE-CONTROL = *PARAMETERS(
    NEW-STD-SAVE-FILE = *IN-PERIODS(
      CONTINUATION-PERIOD = <anzahl tage>))
```

Unter `<hsms_archiv_name>` ist der Name des ursprünglichen HSMS-Archivs anzugeben. `<hsms_directory_name>` bezeichnet das gerade restaurierte HSMS-Archivverzeichnis.

Aus dem so eingerichteten HSMS-Archiv wird die CAT-REC.COPY mit der HSMS-Anweisung RESTORE-FILES restauriert.

```
RESTORE-FILES
  FILE-NAMES =
    :<catid>:<bs2000_benutzerkennung>.<catalog_name>.CAT-REC.COPY,
  NEW-FILE.NAMES = *BY-RULE(
    NEW-CATALOG-ID = <new catid>),
  ARCHIVE-NAME = <hsms_archiv_name>,
  SELECT-SAVE-VERSIONS = *BY-ATTRIBUTES(
    SAVE-VERSION-DATE = <date> (TIME = <time>))
```

Die so restaurierte <catalog\_name>.CAT-REC.COPY ist auf den Namen <catalog\_name>.CAT-REC umzubenennen.

Als Nächstes muss die Umgebung des Ziel-Systems bezüglich des Pubsets bzw. der Privatplatten geprüft werden:

- Liegt auf dem Ziel-System die gleiche Umgebung wie auf dem Ausgangs-System vor, sind keine Änderungen notwendig.
- Ist das Pubset verschieden bzw. sind Privatplatten nicht vorhanden, dann müssen die Dateien vor dem Einspielen per CREATE-FILE-Kommando angelegt werden.

Nach dem Starten des DBH kann die Datenbank über den Utility-Monitor mit der Anweisung RECOVER CATALOG <catalog\_name> wiederhergestellt werden. Sollen die hier protokollierten Daten nicht nachgefahren werden, verwenden Sie die Anweisung RECOVER CATALOG TO zum Einspielen der Sicherung. Alternativ können Sie auch den CAT-LOG-Eintrag in der CAT-REC-Datei mit Hilfe des Utility-Monitors löschen.

## 2.6.2 ARCHIVE-Lauf bei COPY und RECOVER steuern

Wenn Sie eine Sicherung mit ARCHIVE vornehmen möchten, geben Sie bei COPY für USING DIRECTORY den Namen eines ARCHIVE-DIRECTORY an. Für das zu sichernde Archiv darf keine HSMS-Parameterdatei existieren.

Der ARCHIVE-Lauf wird über eine SESAM-spezifische ARCHIVE-Parameterdatei gesteuert. Wie Sie diese Parameterdatei erstellen können, wird im folgenden Abschnitt beschrieben.

### 2.6.2.1 ARCHIVE-Parameterdatei erstellen

Die ARCHIVE-Parameterdatei muss eine SAM-Datei sein, die auf der DBH-Kennung vorhanden ist. Die Datei darf ein Schreibkennwort aber kein Lesekennwort haben, weil sie von der Service-Task gelesen werden muss.

Die ARCHIVE-Parameterdatei wird bei jeder COPY- bzw. RECOVER-Anweisung geöffnet, gelesen und vor dem ARCHIVE-Aufruf wieder geschlossen. D.h., Sie können die Datei zwischen zwei COPY- bzw. RECOVER-Anweisungen verändern.

Der Name der ARCHIVE-Parameterdatei muss sich entweder aus dem Pfadnamen der ARCHIVE-Directory-Datei oder aus dem Namen der DBH-Sitzung ableiten:

*archive\_directory\_name*.ARC-PAR

oder

SESAM $k_n$ .ARC-PAR

*archive\_directory\_name*

Name der ARCHIVE-Directory-Datei, den Sie in der Klausel USING DIRECTORY in der COPY-Anweisung angeben.

Der ARCHIVE-Directory-Pfadname (d.h. :*catid:\$userid.dateiname*) darf maximal 46 Zeichen lang sein, damit der Pfadname der ARCHIVE-Parameterdatei die maximale Länge eines BS2000-Pfadnamens nicht überschreitet.

*k* Konfigurationsname

*n* DBH-Name

SESAM/SQL sucht zuerst die Datei *archive\_directory\_name*.ARC-PAR, danach die session-übergreifende Datei SESAM $k_n$ .ARC-PAR. Wenn keine der beiden Dateien vorhanden ist, wird die COPY- bzw. RECOVER-Anweisung mit Standardwerten für den ARCHIVE-Lauf ausgeführt (siehe [Seite 61](#)). Wenn die ARCHIVE-Parameterdatei leer ist, wird der ARCHIVE-Lauf mit einer Fehlermeldung abgebrochen.

## Übersicht der zulässigen ARCHIVE-Parameter

In der ARCHIVE-Parameterdatei können Sie Parameter für die ARCHIVE-Anweisungen PARAM, SAVE und RESTORE angeben (siehe Handbuch „[ARCHIVE \(BS2000\)](#)“).

Mit den Parametern der PARAM-Anweisung werden Parameterwerte für den gesamten ARCHIVE-Lauf gesetzt, der durch die Utility-Anweisungen COPY bzw. RECOVER angestoßen wird.

Mit den Parametern der SAVE-Anweisung werden Parameterwerte für den Sicherungslauf gesetzt, der durch die Utility-Anweisung COPY angestoßen wird.

Mit den Parametern der RESTORE-Anweisung werden Parameterwerte für den Rekonstruktionslauf gesetzt, der durch die Utility-Anweisungen RECOVER und CREATE REPLICATION angestoßen wird.

Außerdem darf die ARCHIVE-Parameterdatei das BS2000-Kommando SECURE-RESOURCE-ALLOCATION enthalten. Mit diesem Kommando können Sie Betriebsmittel für den ARCHIVE-Lauf reservieren, um zu verhindern, dass der ARCHIVE-Lauf wegen eines Betriebsmittelengpasses scheitert (siehe Handbuch „[Kommandos](#)“).

Folgende ARCHIVE-Parameter dürfen Sie in der ARCHIVE-Parameterdatei angeben:

Parameter	Bedeutung
<i>Für die PARAM-Anweisung:</i>	
OPERATOR	Meldungen an der Bedienstation ausgeben oder Standardbehandlung durchführen
STREAM	Streaming-Mode steuern
<i>Für die SAVE-Anweisung:</i>	
DIRSAVE	ARCHIVE-Directory-Datei sichern
DEVICE	Gerätetyp der Datenträger angeben
SAVE-ACL	ACL-Einträge sichern
DRIVES	Anzahl Parallelläufe angeben
COMPRESS	Sicherung durch die Software komprimieren
RETPD	Schutzfrist in Tagen angeben
<i>Für die RESTORE-Anweisung:</i>	
SPACE	Datenträger reorganisieren oder Sicherungsversion auf den Plattenplatz schreiben, auf dem die Originalversion liegt
DRIVES	Anzahl Parallelläufe angeben
ENVIRONMENT-ATTRIBUTES	Angabe, ob die Attribute für den Zugriffsschutz von der Zieldatei oder von der Sicherungsdatei eingespielt werden sollen

Tabelle 6: Zulässige ARCHIVE-Parameter

## Beschreibung der ARCHIVE-Parameter

Die ARCHIVE-Parameterdatei hat den gleichen Aufbau wie die systemweite Parameterdatei von ARCHIVE (siehe Handbuch „[ARCHIVE \(BS2000\)](#)“), allerdings erweitert um den zusätzlichen Eintrag des BS2000-Kommandos SECURE-RESOURCE-ALLOCATION.

Die ARCHIVE-Parameterdatei kann die nachfolgend aufgeführten Einträge für einen ARCHIVE-Lauf bei COPY, RECOVER und CREATE REPLICATION enthalten. Die Einträge werden von SESAM/SQL ausgewertet, aber ihre Konsistenz nicht geprüft. Daher ist darauf zu achten, dass die Angaben der ARCHIVE-Operanden - z.B. zum Gerätetyp - nicht im Widerspruch zu den entsprechenden Angaben des SECURE-RESOURCE-ALLOCATION-Kommandos stehen dürfen.

Eine vollständige Beschreibung der Einträge der ARCHIVE-Parameterdatei finden Sie im Handbuch „[ARCHIVE \(BS2000\)](#)“ bei den entsprechenden Anweisungen.

Hier sei nur kurz ihre Bedeutung wiederholt und ggf. auf Besonderheiten bei COPY, RECOVER und CREATE REPLICATION hingewiesen.

SECURE-RESOURCE-ALLOCATION ...

OPERATOR = NO / YES

STREAM = NO / YES

DIRSAVE = NO / YES

DEVICE = TAPE-C4 / *device-type*

SAVE-ACL = NO / YES

DRIVES = 1 / *integer*

COMPRESS = NO / YES

RETPD = 0 / *days*

SPACE = REORG / KEEP

ENVIRONMENT-ATTRIBUTES = FROM-TARGET / FROM-ORIGIN

## SECURE-RESOURCE-ALLOCATION ...

Mit diesem BS2000-Kommando können Sie Betriebsmittel für den ARCHIVE-Lauf im Voraus reservieren. Die Betriebsmittel werden grundsätzlich für jede COPY-, RECOVER- bzw. CREATE REPLICATION-Anweisung reserviert und nach der Ausführung wieder freigegeben. Es ist also nicht möglich, die Betriebsmittel vor dem ersten Zugriff einer Reihe von COPY-, RECOVER- bzw. CREATE REPLICATION-Anweisungen zu reservieren und erst nach dem letzten Zugriff wieder freizugeben.

Um mögliche Widersprüche zu vermeiden, sollten Sie hier lediglich die Operanden DEVICE und WAIT verwenden, denn die übrigen Operanden werden bei COPY, RECOVER und CREATE REPLICATION in der Regel nicht benötigt.

Mit DEVICE=\*PARAMETERS(TYPE=...,NUMBER=...) geben Sie an, wieviele Geräte eines Typs zu reservieren sind. Die Angabe muss natürlich zu den entsprechenden ARCHIVE-Parametern passen.

Mit WAIT bestimmen Sie, wie lange zur Erfüllung des Auftrags gewartet werden darf.

Die vollständige Beschreibung des Kommandos SECURE-RESOURCE-ALLOCATION entnehmen Sie bitte dem Handbuch „[Kommandos](#)“.

### **OPERATOR = NO / YES**

Der Parameter legt fest, ob für Archive-Meldungen mit Operator-Antwort ARCHIVE ohne Meldungsausgabe am Bedienplatz eine Standardbehandlung durchführen soll oder ob ARCHIVE diese Meldungen ausgibt und auf Operatorantwort warten soll.

Wenn Sie diesen Parameter nicht angeben, wird der SESAM/SQL-Standardwert YES zugewiesen.

### **STREAM = NO / YES**

Der Parameter legt fest, ob Ein-/Ausgaben auf Bandgeräte im Streaming-Modus erfolgen.

Wenn Sie diesen Parameter nicht angeben, wird der SESAM/SQL-Standardwert YES zugewiesen.

### **DIRSAVE = NO / YES**

Der Parameter legt fest, ob die aktuelle Directory-Datei mitgesichert werden soll.

Wenn Sie diesen Parameter nicht angeben, gilt der BS2000-Standardwert (derzeit NO, siehe BS2000-Standardwerte, [Seite 61](#)).

### **DEVICE = TAPE-C4 / *device-type***

Der Parameter legt den Gerätetyp (MBK oder Magnetband) für die Archivnummern fest. Die Angabe bei DEVICE für die SAVE-Anweisung muss mit der Angabe in dem SECURE-RESOURCE-ALLOCATION-Kommando übereinstimmen, damit die reservierten Geräte für ARCHIVE zugreifbar sind.

Die Angabe wird nicht von SESAM/SQL geprüft.

Wenn Sie diesen Parameter nicht angeben, gilt der BS2000-Standardwert (derzeit TAPE-C4, siehe BS2000-Standardwerte, [Seite 61](#)).

### **SAVE-ACL = NO / YES**

Der Parameter legt fest, ob die ACL-Einträge der gesicherten Dateien mitgesichert werden.

Wenn Sie diesen Parameter nicht angeben, gilt der BS2000-Standardwert (derzeit YES, siehe BS2000-Standardwerte, [Seite 61](#)).

**DRIVES = 1 / integer**

Der Parameter gibt die Zahl der Parallelläufe (maximal 16) an. Mehrere Geräte werden parallel bedient. Der Wert für *integer* muss kleiner oder gleich der Anzahl der zur Verfügung stehenden Geräte sein.

Bei der Angabe DRIVES=1 führt SESAM/SQL keine Paketierung der Dateien durch. Bei Angaben DRIVES>1 wird pro Space ein Paket gebildet. Diese Pakete werden dann von ARCHIVE auf die Geräte verteilt.

Die Angabe bei DRIVES sollte mit der Angabe bei NUMBER im Kommando SECURE-RESOURCE-ALLOCATION übereinstimmen, damit alle benötigten Geräte reserviert sind.

Außerdem sollte die Angabe bei DRIVES bei einer RECOVER-Anweisung mit der Angabe der zugehörigen COPY-Anweisung übereinstimmen, damit ARCHIVE beim RESTORE die gleiche Verteilung der Pakete auf die Geräte vornehmen kann. Wenn die Angaben nicht übereinstimmen, wird von ARCHIVE beim RESTORE keine Parallelität ausgenutzt, d.h. alle Spaces werden seriell nacheinander eingespielt.

Wenn Sie diesen Parameter nicht angeben, gilt der Standardwert der ARCHIVE-Anweisungen RESTORE und SAVE (derzeit 1).

**COMPRESS = NO / YES**

Der Parameter legt fest, ob die Dateien mit Komprimierung durch die Software in die Sicherungsdatei geschrieben werden sollen.

Wenn Sie diesen Parameter nicht angeben, gilt der BS2000-Standardwert (derzeit NO, siehe BS2000-Standardwerte, [Seite 61](#)).

**RETPD = 0 / days**

Der Parameter legt die Schutzfrist in Tagen für die Sicherungsversionen fest. Für *days* können Sie einen Wert zwischen 0 und 32767 angeben.

Wenn Sie diesen Parameter nicht angeben, gilt der BS2000-Standardwert (derzeit 0, siehe BS2000-Standardwerte, [Seite 61](#)).

**SPACE = REORG / KEEP**

Der Parameter legt fest, ob die aus der Sicherung zu ersetzenden Dateien mit gleichem Namen vor dem Zurückschreiben gelöscht werden sollen.

Wenn Sie diesen Parameter nicht angeben, gilt der BS2000-Standardwert (derzeit REORG, siehe BS2000-Standardwerte, [Seite 61](#)).

**ENVIRONMENT-ATTRIBUTES = FROM-TARGET / FROM-ORIGIN**

Mit diesem Parameter legen Sie fest, ob die Attribute für den Zugriffsschutz von der Sicherungsversion oder der Originalversion übernommen werden sollen.

Wenn Sie diesen Parameter nicht angeben, gilt der Standardwert der ARCHIVE-Anweisung RESTORE (derzeit FROM-TARGET).

## Syntaxregeln

- In jeder Zeile darf nur ein Parameter in der Form *parameter=wert* angegeben werden.  
*Ausnahme:* Kommando SECURE-RESOURCE-ALLOCATION.
- Das Kommando SECURE-RESOURCE-ALLOCATION muss vollständig als BS2000-Kommando angegeben werden.  
Wenn es über mehrere Zeilen angegeben wird, muss „-“ als Fortsetzungszeichen bis zur Spalte 72 angegeben werden.  
Die Parameterangabe des SECURE-RESOURCE-ALLOCATION-Kommandos darf die Länge von 1736 Zeichen nicht überschreiten.
- Die Schlüsselwörter müssen in voller Länge angegeben werden.
- Zeilen, die mit „\*“ in Spalte 1 beginnen, werden als Kommentar behandelt.
- Jeder Parameter und das SECURE-RESOURCE-ALLOCATION-Kommando darf jeweils nur einmal angegeben werden.

## Allgemeine Regeln

- Die BS2000-Standardwerte sind in der systemweiten Parameterdatei SYSPAR.ARCHIVE.*version* voreingestellt. Der Systemverwalter kann diese Datei modifizieren und andere Werte als Standardwerte voreinstellen, die somit bei allen ARCHIVE-Läufen gültig sind, wenn Sie in der ARCHIVE-Parameterdatei keine anderen Werte explizit angegeben haben.
- Die Parameterangaben werden jedesmal ausgewertet, wenn bei einer COPY-, RECOVER- oder CREATE REPLICATION-Anweisung Bandsicherungen betroffen sind.
- Die Parameter OPERATOR, STREAM und DRIVES, sowie das Kommando SECURE-RESOURCE-ALLOCATION werden sowohl bei COPY als auch bei RECOVER und CREATE REPLICATION ausgewertet.
- Die Parameter DIRSAVE, DEVICE, SAVE-ACL, COMPRESS und RETPD werden nur bei COPY ausgewertet.
- Die Parameter SPACE und ENVIRONMENT-ATTRIBUTES werden nur bei RECOVER und CRATE REPLICATION ausgewertet.
- Parametern, die in der ARCHIVE-Parameterdatei angegeben sind, wird der angegebene Wert zugewiesen.  
Parametern, die nicht in der ARCHIVE-Parameterdatei angegeben sind, werden die bei der Parameterbeschreibung angegebenen Werte zugewiesen.

- Wenn keine ARCHIVE-Parameterdatei vorhanden ist, wird den Parametern OPERATOR und STREAM der SESAM/SQL-Standardwert YES zugewiesen. Allen anderen Parametern wird von SESAM/SQL kein Wert zugewiesen, d.h., sie werden mit ARCHIVE-Standardwerten ausgeführt (siehe Parameterbeschreibung). Es wird kein SECURE-RESOURCE-ALLOCATION-Kommando ausgeführt.
- In der RESTORE-Anweisung wird dem Parameter REPLACE grundsätzlich der Wert YES zugewiesen.
- Zusätzlich zu den Parametern, die Sie angeben können, werden von SESAM/SQL den folgenden Parametern der PARAM-Anweisung immer die folgenden Werte zugewiesen:

```
CATID=YES  
OLS=YES  
UNLOAD=YES
```

Siehe Handbuch „[ARCHIVE \(BS2000\)](#)“, PARAM-Anweisung.

- Nach Beendigung des ARCHIVE-Laufs wird die Reservierung der Betriebsmittel mit einem weiteren SECURE-RESOURCE-ALLOCATION-Kommando ohne Parameter von SESAM/SQL implizit freigegeben.
- Die ARCHIVE-Anweisungen werden in einer Datei *archive\_directory\_name.SYSLST* protokolliert. Werden mehrere RECOVER SPACE-Anweisungen für dieselbe Datenbank parallel bearbeitet, dann werden die weiteren Anweisungen in einer Datei *archive\_directory\_name.SYSLST.tsn* protokolliert.

In folgenden Fällen wird die COPY-, RECOVER- bzw. CREATE REPLICATION -Anweisung mit einer Fehlermeldung abgebrochen:

- Wenn die ARCHIVE-Parameterdatei leer ist.
- Wenn die ARCHIVE-Parameterdatei keine SAM-Datei ist.
- Wenn die ARCHIVE-Parameterdatei wegen DMS-Fehler nicht geöffnet, geschlossen oder gelesen werden kann.
- Bei Parameterangaben, die nicht erkannt werden (z.B. falsch geschriebene Schlüsselwörter); im Meldungstext werden die Zeilennummer, in der die falsche Parameterangabe steht, und der Dateiname der ARCHIVE-Parameterdatei ausgegeben.
- Bei der Angabe falscher Werte; im Meldungstext werden die Zeilennummer, in der die falsche Parameterangabe steht, und der Dateiname der ARCHIVE-Parameterdatei ausgegeben.

*Ausnahmen*

- Die Angabe des Gerätetyps DEVICE= wird von SESAM/SQL nicht geprüft; die Prüfung wird ARCHIVE überlassen.
- Die Parameterangaben des SECURE-RESOURCE-ALLOCATION-Kommandos werden von BS2000 geprüft.

In folgenden Fällen wird die COPY-, RECOVER- bzw. CREATE REPLICATION-Anweisung mit einer Fehlermeldung abgebrochen und die entsprechende Fehlermeldung von ARCHIVE bzw. von BS2000 in die Datei *archive\_directory\_name*.SYSLST ausgegeben:

- beim abgebrochenen bzw. fehlerhaften ARCHIVE-Lauf (beinhaltet falsche Angabe eines Gerätetyps bei DEVICE= (siehe [Seite 59](#)))
- wenn BS2000 Fehler im SECURE-RESOURCE-ALLOCATION-Kommando feststellt werden (beinhaltet Syntaxfehler in den Parameterangaben des SECURE-RESOURCE-ALLOCATION-Kommandos (siehe [Seite 58](#))).

*Beispiel einer ARCHIVE-Parameterdatei*

```
SECURE-RESOURCE-ALLOCATION DEVICE=(TYPE=TAPE-C4,NUMBER=1),  
WAIT=(TIME=TASK-STD)  
*   ***reserve resources***  
*  
OPERATOR=YES  
*   ***output messages to console***  
*  
STREAM=YES  
*   ***perform input and output in streaming mode***  
*  
DIRSAVE=YES  
*   ***store directory file***  
*  
DEVICE=TAPE-C4  
*   ***specify device type***  
*  
SAVE-ACL=YES  
*   ***store ACL entries***  
*  
DRIVES=1  
*   ***no parallel runs, no data packaging***  
*  
COMPRESS=YES  
*   ***write data to backup file in compressed form***  
*  
RETPD=365  
*   ***retention period for backup version is 365 days***  
*  
SPACE=REORG  
*   ***reorganize disk***  
*  
ENVIRONMENT-ATTRIBUTES=FROM-TARGET  
*   ***take access protection attributes from backup version***  
*
```

## 2.7 Bearbeitung von Eingabe-, Ausgabe- und Fehlerdateien

Bei einigen Utility-Anweisungen können bzw. müssen Sie Dateinamen für Eingabe-, Ausgabe- oder Fehlerdateien angeben (siehe [Kapitel „Utility-Anweisungen“ auf Seite 69](#)). Die Dateinamen können Sie mit Katalogkennung und BS2000-Benutzerkennung angeben. Der Dateiname muss unabhängig von der Katalogkennung eindeutig sein, d.h. zwei Dateinamen dürfen sich nicht nur durch die Katalogkennung unterscheiden.

Bei Eingabedateien ist zu beachten:

- Falls der Dateiname ohne BS2000-Benutzerkennung angegeben ist, erwartet SESAM/SQL die Datei in der BS2000-Benutzerkennung, in der der DBH abläuft.
- Falls die angegebene BS2000-Benutzerkennung nicht die DBH-Kennung ist, muss die BS2000-Benutzerkennung des DBH die erforderlichen Zugriffsrechte besitzen.

Bei Ausgabe- und Fehlerdateien ist zu beachten:

- Falls der Dateiname ohne BS2000-Benutzerkennung angegeben ist, legt SESAM/SQL die Datei unter der BS2000-Benutzerkennung an, in der der DBH abläuft, sofern die Datei dort nicht schon vorhanden ist. Ist keine Katalogkennung angegeben, so verwendet SESAM/SQL das Standard-Pubset der BS2000-Benutzerkennung, in der der DBH abläuft.
- Falls die angegebene BS2000-Benutzerkennung nicht die DBH-Kennung ist, muss der Datenbankverwalter Vorbereitungen für das Anlegen der Datei in dieser BS2000-Benutzerkennung treffen, siehe „Basishandbuch“, Abschnitt „Datenbankdateien und Jobvariablen auf fremden Benutzerkennungen“. Außerdem muss die BS2000-Benutzerkennung des DBH die erforderlichen Zugriffsrechte besitzen.
- Wenn keine Fehlerdatei angegeben ist, dann wird ein Standard-Dateiname verwendet und die Datei auf der DB-Kennung oder, falls das nicht möglich ist, auf der DBH-Kennung angelegt.
- Liegt die Datenbank in einer eigenen DB-Kennung, so versucht SESAM/SQL die Fehlerdatei ebenfalls in der DB-Kennung anzulegen. Falls dies nicht möglich ist, wird die Fehlerdatei in der DBH-Kennung angelegt.

Stellt SESAM/SQL beim Anlegen einer Datei fest, dass eine Datei gleichen Namens unter einer anderen Katalogkennung bereits vorhanden ist, wird die Ausführung der Utility-Anweisung mit SQLSTATE abgebrochen. Analog wird verfahren, wenn SESAM/SQL beim Öffnen einer Datei feststellt, dass es mehrere Dateien dieses Namens unter verschiedenen Katalogkennungen gibt. Mit der CATID-Liste kann dieses Problem umgangen werden, siehe „[Basishandbuch](#)“.

Den Versuch, bei der Ausführung einer Utility-Anweisung eine Datei in einer anderen BS2000-Benutzerkennung (DB-Kennung) als der DBH-Kennung anzulegen, weist SESAM/SQL bei nicht ausreichenden Zugriffsrechten mit SQLSTATE zurück.

### Fehlerdatei lesen

Eine von SESAM/SQL bei Ausführung der Utility-Anweisungen LOAD, UNLOAD oder CHECK FORMAL erstellte Fehlerdatei können Sie mit dem Dateibearbeiter EDT (siehe Handbuch „EDT (BS2000)“) lesen oder per SHOW-FILE-Kommando ausgeben.

Die Namen der Fehlerdateien für die Utility-Anweisungen LOAD, UNLOAD und CHECK FORMAL lauten:

LOAD	<i>catalog.space</i> .EXC.L
UNLOAD OFFLINE	<i>catalog.space</i> .EXC.U
UNLOAD ONLINE	<i>file_name</i> .EXC.U
CHECK FORMAL	<i>catalog.space</i> .EXC.C

Dabei bezeichnen *catalog* und *space* die Namen der Datenbank bzw. des Anwender-Space, die von der Ausführung der Utility-Anweisung betroffen sind.

Für eine partitionierte Tabelle wird bei LOAD und UNLOAD immer der Spacename der ersten Partition verwendet. Die Fehlerdatei enthält die Fehlerinformationen für alle Partitionen der Tabelle.

Bei CHECK FORMAL wird für jede fehlerhafte Partition eine Fehlerdatei mit dem Spacenaamen der Partition gebildet.

## 2.8 Benutzerkennung und Linknamen

### BS2000-Benutzerkennung für die SESAM/SQL-Datenbank

Alle Dateien einer SESAM/SQL-Datenbank müssen unter derselben BS2000-Benutzerkennung katalogisiert sein. Möglich sind

- die **DBH-Kennung**, die Benutzerkennung in der der DBH abläuft oder
- die **DB-Kennung**, eine Benutzerkennung in der die Datenbank angelegt ist.

Diese BS2000-Benutzerkennung trägt der DBH ins SQL-Datenbankverzeichnis ein (siehe Handbuch „[Datenbankbetrieb](#)“).

Die Utility-Anweisungen laufen unter der Steuerung des DBH in der DBH-Kennung ab.

Liegt eine Datenbank in einer DB-Kennung, versuchen DBH und Utilities immer zuerst, auf Dateien in dieser Kennung zuzugreifen.

In diesem Fall muss der Datenbankverwalter Vorbereitungen für das Anlegen von Dateien durch DBH und Utilities (z.B. SESAM-Sicherungsbestände und Logging-Dateien) in dieser BS2000-Benutzerkennung treffen, siehe „Basishandbuch“, Abschnitt „Datenbankdateien und Jobvariablen auf fremden Benutzerkennungen“.

Arbeiten Sie mit SESAM-Sicherungsbeständen auf Magnetbandkassette, gelten für die beiden Softwareprodukte HSMS und ARCHIVE unterschiedliche Bedingungen:

- Mit HSMS können Sie SESAM-Sicherungskopien auch von SESAM/SQL-Datenbanken erstellen und einspielen, die nicht in der DBH-Kennung liegen.
- Mit ARCHIVE können Sie nur SESAM-Sicherungskopien von SESAM/SQL-Datenbanken erstellen und einspielen, die in der DBH-Kennung liegen.

### Verwendung von Linknamen

Für die Ausführung von Utility-Anweisungen ist die Zuweisung von Sicherungs- und Arbeitsdateien mit BS2000-Linknamen nicht möglich.



---

## 3 Utility-Anweisungen

Dieses Kapitel beschreibt die Utility-Anweisungen als

- Überblick über die Utility-Anweisungen
- alphabetischen Nachschlageteil

### 3.1 Überblick über die Utility-Anweisungen (alphabetisch)

Die folgende Tabelle gibt einen Überblick über die Utility-Anweisungen.

<b>UTILITY-Anweisung</b>	<b>Utility-Funktion</b>	<b>Seite</b>
ALTER CATALOG	Eigenschaften der Datenbank ändern	<a href="#">74</a>
ALTER DATA FOR TABLE	Mischen von Spaltenwerten, Anonymisieren von Daten	<a href="#">80</a>
ALTER MEDIA DESCRIPTION	Medientabelle ändern	<a href="#">76</a>
ALTER PARTITIONING FOR TABLE	Partitionierung einer Basistabelle ändern	<a href="#">80</a>
CHECK CONSTRAINTS	Integritätsbedingungen prüfen	<a href="#">91</a>
CHECK FORMAL	Basistabellen und Indizes formal prüfen	<a href="#">94</a>
COPY	SESAM-Sicherungsbestand erstellen	<a href="#">97</a>
CREATE CATALOG	Datenbank (Catalog-Space) anlegen	<a href="#">105</a>
CREATE MEDIA DESCRIPTION	ersten Mediensatz für eine datenbank-spezifische Dateiart in die Medientabelle aufnehmen	<a href="#">115</a>
CREATE REPLICATION	Replikat erzeugen	<a href="#">119</a>
DROP MEDIA DESCRIPTION	alle Mediensätze für eine datenbank-spezifische Dateiart aus der Medientabelle löschen	<a href="#">127</a>
EXPORT TABLE	Tabelle aus einer Datenbank in eine Export-Datei exportieren	<a href="#">129</a>
IMPORT TABLE	Tabelle aus einer Export-Datei in eine Datenbank importieren	<a href="#">133</a>
LOAD	Anwenderdaten in eine Basistabelle laden	<a href="#">140</a>
MIGRATE	Datenbanken und Tabellen migrieren	<a href="#">166</a>
MODIFY	Informationen (Metadaten) über die SESAM-Sicherungsbestände pflegen	<a href="#">171</a>
RECOVER	Zurücksetzen und Reparieren, Indizes neu aufbauen	<a href="#">178</a>
REFRESH REPLICATION	Replikat aktualisieren	<a href="#">219</a>
REFRESH SPACE	Replikat erweitern	<a href="#">223</a>
REORG	Catalog-Space, Anwender-Spaces und Basistabellen reorganisieren	<a href="#">227</a>
UNLOAD	Anwenderdaten aus einer Basistabelle ausgeben	<a href="#">234</a>

Tabelle 7: Überblick über die Utility-Anweisungen

## 3.2 Alphabetischer Nachschlageteil

In diesem Abschnitt sind die Utility-Anweisungen in einem einheitlichen Stil beschrieben. Die Anweisungen sind alphabetisch angeordnet, und pro Anweisung gibt es einen Eintrag, der den Namen der Anweisung als Kopfzeile hat.

Die Beschreibung der Utility-Anweisungen baut auf den Kenntnissen auf, die im „[Basis-handbuch](#)“ und im einführenden Teil des Handbuchs „[SQL-Sprachbeschreibung Teil 1: SQL-Anweisungen](#)“ vermittelt werden.

Im [Abschnitt „Syntaxübersicht“ auf Seite 257](#) sind alle für die Darstellung der Utility-Anweisungen benötigten Syntaxelemente in alphabetischer Reihenfolge aufgelistet. Es sind auch solche Syntaxelemente aufgeführt, die im Handbuch „[SQL-Sprachbeschreibung Teil 1: SQL-Anweisungen](#)“ erklärt sind und auf die deshalb im vorliegenden Handbuch nicht näher eingegangen wird.

### Aufbau eines Eintrags

Jeder Eintrag besteht aus mehreren Abschnitten.

Bei einem Eintrag können Abschnitte fehlen, wenn sie für die entsprechende Utility-Anweisung keine Bedeutung haben. Außerdem können nach der Erklärung der Syntax weitere Abschnitte vorhanden sein, die Besonderheiten der jeweiligen Utility-Anweisung beschreiben. Die wichtigsten Abschnitte sind hier zusammengestellt.

### Anweisungsname - Kurzbeschreibung

Nach der Überschrift wird die Wirkungsweise der Anweisung beschrieben.

Dieser Abschnitt beschreibt auch die Voraussetzungen, die für die erfolgreiche Ausführung der Anweisung erfüllt sein müssen. Insbesondere sind die notwendigen Zugriffsrechte zusammengestellt.

---

ANWEISUNGSNAME KLAUSEL *parameter* . . .

---

*parameter*

Erklärungen zu dem Parameter

Die Klauseln und Parameter sind in der Reihenfolge beschrieben, in der sie in der Syntaxdefinition vorkommen.

## Beispiele

Dieser Abschnitt zeigt ein oder mehrere Beispiele für die Verwendung der Anweisung. Die meisten Beispiele beziehen sich auf die Beispieldatenbank AUFTRAGKUNDEN, die im „[Basishandbuch](#)“ beschrieben ist.



Handbuchbeispiele werden durch nebenstehendes Symbol gekennzeichnet, wenn sie als Bestandteil einer Anweisungsdatei oder eines ESQL-COBOL-Programms in der Beispieldatenbank enthalten sind.

## Siehe auch

Verwandte Anweisungen

## ALTER CATALOG - Eigenschaften einer Datenbank ändern

ALTER CATALOG ändert die Eigenschaften einer Datenbank.

Der codierte Zeichensatz (Coded Character Set, CCS; synonym: Code-Tabelle) für die Datenbank (siehe Klausel CODE-TABLE auf [Seite 108](#)) kann geändert werden.

Der aktuelle Berechtigungsschlüssel muss das Sonder-Privileg UTILITY besitzen.

```
ALTER CATALOG catalog
```

```
    ALTER CODE_TABLE ccs_name
```

*catalog*

Name der SESAM/SQL-Datenbank, deren Eigenschaften geändert werden.

```
ALTER_CODE_TABLE ccs_name
```

Maximal 8 Zeichen langer Name für einen im BS2000 (Systemkomponente XHCS) definierten codierten Zeichensatz (Coded Character Set, CCS).

*ccs\_name* muss als alphanumerisches Literal angegeben werden, bestehend aus Grossbuchstaben, Ziffern, @, # oder \$. Das erste Zeichen darf keine Ziffer sein. Für *ccs\_name* dürfen die CCS-Namen von EBCDIC-Zeichensätzen oder selbst definierten Zeichensätzen auf Basis von EBCDIC-Zeichensätzen angegeben werden. Für *ccs\_name* darf auch der Wert `_NONE_` angegeben werden. Dann wird für die Datenbank kein codierter Zeichensatz verwendet. Dies entspricht dem Verhalten von SESAM/SQL vor V5.0.

Der *ccs\_name* (siehe auch: CREATE CATALOG) wird in den Metadaten der Datenbank eingetragen und verwendet, z.B. bei der Utility-Anweisung LOAD zur Prüfung gegen den CCSN der Eingabedatei.

Einführende Informationen zur Anwendung codierter Zeichensätze in SESAM/SQL finden Sie im Abschnitt „Unicode-Konzept in SESAM/SQL“ im „[Basishandbuch](#)“

### Beispiel

Das Beispiel ändert den codierten Zeichensatz der Datenbank AUFTRAGKUNDEN:



```
ALTER CATALOG auftragkunden ALTER CODE_TABLE 'EDF041'
```

### Siehe auch

CREATE CATALOG

## ALTER DATA FOR TABLE - Mischen von Spaltenwerten, Anonymisieren von Daten

ALTER DATA FOR TABLE mischt Spaltenwerte einer Basistabelle derart, dass keine Rückschlüsse auf den ursprünglichen Inhalt möglich sind. Die ursprünglichen Werte einer Spalte und ihre Häufigkeitsverteilung bleiben dabei erhalten. Die Durchmischung ist von Spalte zu Spalte und von Fall zu Fall unterschiedlich.

Wichtige (z.B. personenbezogene) Benutzerdaten werden dadurch anonymisiert. Einführende Informationen zur Anonymisierung von Daten in SESAM/SQL finden Sie im „[Basishandbuch](#)“

Der aktuelle Berechtigungsschlüssel muss das Sonder-Privileg UTILITY besitzen oder Eigentümer des Schemas sein, zu dem die Basistabelle gehört.

```
ALTER DATA FOR TABLE tabelle
    SHUFFLE VALUES FOR COLUMN { spalte
                                (spalte, . . . ) } , . . .
```

### *tabelle*

Name einer Basistabelle.

Sie darf keine Nur-CALL-DML-Tabelle sein und darf sich nicht im Zustand „check pending“ befinden.

```
{ spalte
  (spalte, . . . ) }
```

Namen der Spalten beliebigen Datentyps, deren Werte gemischt werden sollen. Spalten, deren Namen in Klammern angegeben sind, werden als Einheit betrachtet und zusammenhängend gemischt. Die logische Beziehung zwischen diesen Spalten bleibt erhalten.

Wenn multiple Spalten angegeben werden, dann werden alle Ausprägungen der multiplen Spalte als Ganzes gemischt. Die Definitionslänge aller definierten Ausprägungen darf nicht größer sein als 512 Bytes.

Die Spalten des Primärschlüssels dürfen nicht angegeben werden.

CHECK-, Eindeutigkeits- und Referenzbedingungen dürfen nur für einzelne zu mischende Spalten definiert sein. Sie dürfen nicht für mehrere Spalten definiert sein, wenn mindestens eine Spalte davon gemischt werden soll.

Die betroffenen Anwender-Spaces erhalten den Zustand „copy pending“, wenn sie der logischen Datensicherung unterliegen.

Wenn Indizes für die Tabelle definiert sind, dann werden sie neu aufgebaut. Die Index-Spaces erhalten den Zustand „copy pending“, wenn sie der logischen Datensicherung unterliegen.

## Beispiel

Das Beispiel mischt die personenbezogenen Daten der Tabelle Kontakt. Die Spalten *vorname*, *nachname*, und *anrede* werden als Einheit betrachtet und zusammenhängend gemischt.



```
ALTER DATA FOR TABLE kontakt.anonym
SHUFFLE VALUES FOR COLUMN konr,knr,(vorname,nachname,anrede),
kotelefon,funktion,abteilung,koinfo
```

## ALTER MEDIA DESCRIPTION - Medientabelle ändern

Mit ALTER MEDIA DESCRIPTION können Sie für eine bestimmte datenbank-spezifische oder space-spezifische Dateart die Beschreibung der Dateieigenschaften in der Medientabelle ändern. Außerdem können Sie neue Einträge (Mediensätze) in die Medientabelle aufnehmen oder Einträge aus der Medientabelle löschen. Die Medientabelle ist im „[Basis-handbuch](#)“ beschrieben.

SESAM/SQL kennt folgende datenbank-spezifische Datearten:

- CAT-REC-Datei
- CAT-LOG-Dateien
- DA-LOG-Dateien
- PBI-Dateien  
(Mediensätze gelten auch für die Arbeitsdatei bei Online-Sicherung mit HSMS)

SESAM/SQL kennt folgende space-spezifische Dateart:

- DDL-TA-LOG-Dateien

Die mit ALTER MEDIA DESCRIPTION vorgenommenen Änderungen betreffen nur neu anzulegende Dateien der angegebenen Dateart. Auf bereits bestehende Dateien hat ALTER MEDIA DESCRIPTION keine Auswirkungen.

Der aktuelle Berechtigungsschlüssel muss das Sonder-Privileg UTILITY besitzen.

```
ALTER MEDIA DESCRIPTION FOR {
    DALOG
    CATLOG
    CATREC
    PBI
    DDLTA
} AT CATALOG catalog
```

```
{
    PRIMARY zahl
    SECONDARY zahl
    [NO] SHARE
    [NO] DEVICE REQUEST
    ADD STOGROUP stogroup
    DROP STOGROUP stogroup
} , . . .
```

FOR {  
 DALOG  
 CATLOG  
 CATREC  
 PBI  
 DDLTA }

Gibt die Dateart an, für die Änderungen in der Medientabelle vorgenommen werden sollen.

DALOG Es werden Änderungen für die Dateart DA-LOG-Datei vorgenommen.

CATLOG Es werden Änderungen für die Dateart CAT-LOG-Datei vorgenommen.

CATREC Es werden Änderungen für die Dateart CAT-REC-Datei vorgenommen.

PBI Es werden Änderungen für die Dateart PBI-Datei vorgenommen (gilt auch für die Arbeitsdatei bei Online-Sicherung mit HSMS).

DDLTA Es werden Änderungen für die Dateart DDL-TA-LOG-Datei vorgenommen.

AT CATALOG *catalog*

Name der Datenbank, zu der die bearbeitete Medientabelle gehört.

Jede der folgenden Space-Klauseln PRIMARY, SECONDARY, [NO] SHARE dürfen Sie nur einmal angeben.

PRIMARY *zahl*

Neue Primärzuweisung für Dateien der angegebenen Dateart in 2K-Einheiten (BS2000-Halfpage).

*zahl* muss eine Ganzzahl von 1 bis 33 554 430 sein.

Ist die Angabe für *zahl* zu klein, korrigiert SESAM/SQL den Wert automatisch auf die erforderliche Mindestgröße.

PRIMARY *zahl* nicht angeben:

Primärzuweisung für neu anzulegende Dateien der angegebenen Dateart bleibt unverändert.

SECONDARY *zahl*

Neue Sekundärzuweisung für Dateien der angegebenen Dateart in 2K-Einheiten (BS2000-Halfpage), falls als Dateart CATREC oder PBI angegeben ist.

*zahl* muss eine Ganzzahl von 0 bis 32767 sein.

Ist die Angabe für *zahl* zu klein, korrigiert SESAM/SQL den Wert automatisch auf die erforderliche Mindestgröße.

Falls als Dateart DALOG oder CATLOG angegeben ist, darf die Space-Klausel SECONDARY nicht angegeben werden.

SECONDARY *zahl* nicht angeben:

Sekundärzuweisung für neu anzulegende Dateien der angegebenen Dateart bleibt unverändert.

**[NO] SHARE**

SHARE legt fest, dass die Dateien der angegebenen Dateart gemeinsam benutzbar sind.

NO SHARE legt fest, dass die Dateien der angegebenen Dateart nicht gemeinsam benutzbar sind.

**[NO] DEVICE REQUEST**

Mit DEVICE REQUEST legen Sie fest, dass bei abgearbeiteter Medientabelle weitere Geräte für die angegebene Dateart am Bedienplatz angefordert werden.

Wenn SESAM/SQL dann bei abgearbeiteter Medientabelle weitere Geräte anfordert, sind am Bedienplatz folgende Eingaben möglich:<sup>1</sup>

**PUBLIC *catid***

Die Datei wird auf dem angegebenen Pubset angelegt.

*catid* BS2000-Katalogkennung.  
Die BS2000-Katalogkennung muss angegeben werden.  
*catid* muss als alphanumerisches Literal angegeben werden.  
Wurde eine CATID-Liste definiert, muss die Katalogkennung in der Liste angegeben worden sein.

**IGNORE**

Die Datei wird nicht angelegt.

**PRIVATE *catid, gerätetyp, vsn, vsn...***

Die Datei wird auf Privatplatten mit den angegebenen Datenträgerkennzeichen und dem angegebenen Gerätetyp angelegt und mit der angegebenen Katalogkennung katalogisiert. Es können maximal 6 Datenträgerkennzeichen angegeben werden.

*catid* BS2000-Katalogkennung.  
Die BS2000-Katalogkennung muss angegeben werden.  
*catid* muss als alphanumerisches Literal angegeben werden.  
Wurde eine CATID-Liste definiert, muss die Katalogkennung in der Liste angegeben worden sein.

*gerätetyp* Gerätetyp der Privatplatte.  
*gerätetyp* muss als alphanumerisches Literal angegeben werden.

*vsn* Datenträgerkennzeichen.  
*vsn* muss als alphanumerisches Literal angegeben werden.

Mit NO DEVICE REQUEST legen Sie fest, dass bei abgearbeiteter Medientabelle keine weiteren Geräte am Bedienplatz angefordert werden.

**[NO] DEVICE REQUEST nicht angeben:**

Die aktuelle Einstellung für die angegebene Dateart bleibt erhalten.

---

<sup>1</sup> Die Angabe TAPE *gerätetyp* kann in SESAM/SQL ab V7.0 nicht mehr angegeben werden

Die Klausel [NO] DEVICE REQUEST dürfen Sie nur einmal angeben.

#### ADD STOGROUP *stogroup*

Für die angegebene Dateart wird ein neuer Eintrag (Mediensatz) in die Medientabelle aufgenommen. *stogroup* bezeichnet den Namen der Storage Group, die zusätzlich für Dateien der angegebenen Dateart bereitgestellt wird. Der einfache Name der Storage Group kann durch den Datenbanknamen *catalog* qualifiziert werden.

Für die angegebene Dateart darf noch kein Mediensatz vorhanden sein, der einen Magnetbandgerätetyp oder einen Magnetbandkassetten-Gerätetyp bezeichnet (Klausel TAPE, siehe [Abschnitt „CREATE MEDIA DESCRIPTION - Dateieigenschaften der datenbank-spezifischen Dateien festlegen“ auf Seite 115](#)).

Außerdem ist zu beachten, dass die Anzahl der Mediensätze pro Dateart in der Medientabelle begrenzt ist: Für DA-LOG-, CAT-LOG- und DDL-TA-LOG-Dateien können jeweils 10, für PBI-Dateien 8 und für CAT-REC-Dateien 2 Sätze eingetragen werden.

Die Klausel ADD STOGROUP dürfen Sie höchstens einmal angeben. Innerhalb derselben ALTER MEDIA DESCRIPTION-Anweisung dürfen Sie nicht zugleich die Klauseln ADD STOGROUP und DROP STOGROUP angeben.

#### DROP STOGROUP *stogroup*

Für die angegebene Dateart wird ein Eintrag (Mediensatz) in der Medientabelle gelöscht. Dieser Mediensatz wird über den Namen *stogroup* der zugehörigen Storage Group identifiziert. Der einfache Name der Storage Group kann durch den Datenbanknamen *catalog* qualifiziert werden.

Der zu löschende Mediensatz darf nicht der Einzige für die angegebene Dateart sein. Andernfalls wird ALTER MEDIA DESCRIPTION abgewiesen.

Die Klausel DROP STOGROUP dürfen Sie höchstens einmal angeben. Innerhalb derselben ALTER MEDIA DESCRIPTION-Anweisung dürfen Sie nicht zugleich die Klauseln ADD STOGROUP und DROP STOGROUP angeben.

### Beispiel

Das Beispiel ändert in der Medientabelle der Datenbank AUFTRAGKUNDEN die Beschreibung der Dateieigenschaften für DA-LOG-Dateien: Die Primärzuweisung für DA-LOG-Dateien wird erhöht von dem Standardwert 768 2K-Einheiten auf 1500 2K-Einheiten.



```
ALTER MEDIA DESCRIPTION FOR DALOG AT CATALOG auftragkunden  
PRIMARY 1500, SHARE
```

### Siehe auch

CREATE MEDIA DESCRIPTION, DROP MEDIA DESCRIPTION

## **ALTER PARTITIONING FOR TABLE - Partitionierung einer Basistabelle ändern**

Mit ALTER PARTITIONING FOR TABLE können Sie

- einer Basistabelle eine Partition hinzufügen (ADD)
- die Partitions Grenzen einer Partition ändern (ALTER)
- eine Partition löschen (DROP)

Zur Ausführung von ALTER PARTITIONING FOR TABLE muss der aktuelle Berechtigungsschlüssel entweder das Sonder-Privileg UTILITY besitzen oder Eigentümer des Schemas sein, zu dem die Basistabelle gehört.

Die verschiedenen Formen dieser Anweisung sind im Folgenden in eigenen Abschnitten beschrieben.

**ALTER PARTITIONING FOR TABLE ... ADD PARTITION**

Diese Anweisung fügt einer Basistabelle eine Partition hinzu.

Eine nicht-partitionierte Basistabelle mit Primärschlüssel wird dadurch in eine partitionierte Basistabelle umgewandelt.

ALTER PARTITIONING FOR TABLE *tabelle*

ADD PARTITION ON SPACE *space*

SET UPPER LIMIT OF  $\left\{ \begin{array}{l} \text{ADDED} \\ \text{PRIOR} \end{array} \right\}$  PARTITION TO VALUE  $\left\{ \begin{array}{l} < \\ <= \end{array} \right\}$  (*spaltenwert, ...*)

$\left[ \left\{ \begin{array}{l} \text{ONLY WHEN EMPTY} \\ \text{PRESERVE ROWS} \end{array} \right\} \right] \left[ \left\{ \begin{array}{l} \text{GENERATE INDEX} \\ \text{NO INDEX} \end{array} \right\} \right]$

*tabelle*

Name einer partitionierten Basistabelle oder einer nicht-partitionierten Basistabelle mit Primärschlüssel. Die Tabelle darf keine Nur-CALL-DML-Tabelle und keine CALL-DML-Tabelle mit Kennwortschutz sein und sie darf sich nicht im Zustand „check pending“ befinden.

ADD PARTITION ON SPACE *space*

Name eines existierenden Anwender-Space.

Es darf sich keine Partition der Tabelle *tabelle* darauf befinden.

Der Eigentümer des Space muss auch der Eigentümer des Schemas sein, zu dem die Basistabelle gehört.

Die neue Partition wird auf dem angegebenen Anwender-Space angelegt.

Die maximal möglich Anzahl von Partitionen einer Basistabelle (16) darf nicht überschritten werden.

SESM/SQL nummeriert die Partitionen der Tabelle automatisch neu gemäß den Obergrenzen der Partitionen aufsteigend von 1 bis 16.

SET UPPER LIMIT OF  $\left\{ \begin{array}{l} \text{ADDED} \\ \text{PRIOR} \end{array} \right\}$  PARTITION TO VALUE  $\left\{ \begin{array}{l} < \\ <= \end{array} \right\}$  (*spaltenwert,...*)

Bei einer **partitionierten** Tabelle muss die angegebene Obergrenze verschieden sein von den bereits existierenden Obergrenzen der Partitionen. Es wird die Partition mit der nächst höheren als der angegebenen Obergrenze geteilt.

Bei Angabe von ADDED ist die neue Partition die niedrigere der beiden Partitionen und erhält die angegebene Obergrenze.

Bei Angabe von PRIOR ist die neue Partition die höhere der beiden Partitionen. Die zu teilende Partition erhält die angegebene Obergrenze.

Eine **nicht-partitionierte** Basistabelle wird in eine partitionierte Tabelle mit zwei Partitionen umgewandelt.

Bei Angabe von ADDED ist die neue Partition die niedrigere der beiden Partitionen und erhält die angegebene Obergrenze. Die Obergrenze der höheren Partition ergibt sich aus dem höchsten Primärschlüsselwert.

Bei Angabe von PRIOR ist die neue Partition die höhere der beiden Partitionen. Die Obergrenze der höheren Partition ergibt sich aus dem höchsten Primärschlüsselwert. Die niedrigere Partition erhält die angegebene Obergrenze.

Durch den vorangestellten Vergleichsoperator wird die Obergrenze mit eingeschlossen oder ausgeschlossen:

- <= Sätze mit Primärschlüsselwert gleich *spaltenwert*,... gehören zu **dieser** Partition
- < Sätze mit Primärschlüsselwert gleich *spaltenwert*,... gehören zur **nächsten** Partition

Für den Vergleich gelten die lexikografischen Regeln, siehe Handbuch „[SQL-Sprachbeschreibung Teil 1: SQL-Anweisungen](#)“.

#### ONLY WHEN EMPTY

Die neue Partition wird nur dann hinzugefügt, wenn sie leer bleibt, d.h. wenn keine Sätze zu verschieben sind.

Wenn in einer CALL DML-Tabelle Indizes definiert sind, dann erhalten die betroffenen Anwender-Spaces den Zustand „copy pending“, wenn sie der logischen Datensicherung unterliegen.

#### PRESERVE ROWS

Die betroffenen Sätze bleiben erhalten.

Sie werden entsprechend den neuen Obergrenzen verschoben.

Wenn Sätze von einer Partition in eine andere übertragen werden, dann werden die Satznummern gemäß dem Präfix ROW\_ID\_PREFIX (siehe View SYS.PARTITIONS im SYS\_INFO\_SCHEMA, Handbuch [SQL-Sprachbeschreibung Teil 1: SQL-Anweisungen](#)) angepasst.

Die betroffenen Anwender-Spaces erhalten den Zustand „copy pending“, wenn sie der logischen Datensicherung unterliegen.

### GENERATE INDEX

Wenn Sekundärindizes definiert sind und durch die Angabe PRESERVE ROWS Sätze verschoben wurden, dann werden die Sekundärindizes der Tabelle neu aufgebaut.

Dies gilt auch für Sekundärindizes, die bereits vor der Anweisung ALTER PARTITIONING als defekt gekennzeichnet waren.

Die betroffenen Index-Spaces erhalten den Zustand „copy pending“ wenn sie der logischen Datensicherung unterliegen.

Wenn ONLY WHEN EMPTY angegeben wurde oder durch die Angabe PRESERVE ROWS keine Sätze verschoben wurden, dann werden die Sekundärindizes nur dann neu aufgebaut, wenn es zuvor schon als defekt gekennzeichnete Sekundärindizes gab. Sonst werden die Sekundärindizes nicht neu aufgebaut.

### NO INDEX

Wenn Sekundärindizes definiert sind und durch die Angabe PRESERVE ROWS Sätze verschoben wurden, dann werden die Sekundärindizes der Tabelle als defekt gekennzeichnet und nicht neu aufgebaut.

Wenn ONLY WHEN EMPTY angegeben wurde oder durch die Angabe PRESERVE ROWS keine Sätze verschoben wurden, dann werden die Sekundärindizes nicht neu aufgebaut. Sie behalten ihren ursprünglichen Zustand.

**ALTER PARTITIONING FOR TABLE ... ALTER PARTITION**

Diese Anweisung ändert die Partitionsgrenze einer Partition.

---

ALTER PARTITIONING FOR TABLE *tabelle*

ALTER PARTITION ON SPACE *space*

SET UPPER LIMIT TO VALUE  $\left\{ \begin{array}{l} < \\ <= \end{array} \right\}$  (*spaltenwert, ...*)

$\left[ \begin{array}{l} \text{ONLY WHEN EMPTY} \\ \text{PRESERVE ROWS} \end{array} \right] \left[ \begin{array}{l} \text{GENERATE INDEX} \\ \text{NO INDEX} \end{array} \right]$

---

*tabelle*

Name einer partitionierten Basistabelle. Sie darf sich nicht im Zustand „check pending“ befinden.

ALTER PARTITION ON SPACE *space*

Name des Anwender-Space mit der Partition, deren Partitionsgrenze geändert werden soll.

SET UPPER LIMIT TO VALUE  $\left\{ \begin{array}{l} < \\ <= \end{array} \right\}$  (*spaltenwert, ...*)

Die angegebene Obergrenze muss kleiner sein als die Obergrenze der nächst höheren Partition. Sie muss auch größer sein als die Obergrenze der nächst niedrigeren Partition. Die Obergrenze der höchsten Partition kann nicht geändert werden.

Durch den vorangestellten Vergleichsoperator wird die Obergrenze mit eingeschlossen oder ausgeschlossen:

- <= Sätze mit Primärschlüsselwert gleich *spaltenwert, ...* gehören zu **dieser** Partition
- < Sätze mit Primärschlüsselwert gleich *spaltenwert, ...* gehören zur **nächsten** Partition

Für den Vergleich gelten die lexikografischen Regeln, siehe Handbuch „[SQL-Sprachbeschreibung Teil 1: SQL-Anweisungen](#)“.

ONLY WHEN EMPTY

Die Änderung der Obergrenze der Partition darf zu keiner Verschiebung von Sätzen führen.

### PRESERVE ROWS

Die betroffenen Sätze bleiben erhalten.  
Sie werden entsprechend den neuen Obergrenzen verschoben.

Wenn Sätze von einer Partition in eine andere übertragen werden, dann werden die Satznummern gemäß dem Präfix ROW\_ID\_PREFIX (siehe View SYS.PARTITIONS im SYS\_INFO\_SCHEMA, Handbuch [SQL-Sprachbeschreibung Teil 1: SQL-Anweisungen](#)) angepasst.

Die betroffenen Anwender-Spaces erhalten den Zustand „copy pending“, wenn sie der logischen Datensicherung unterliegen.

### GENERATE INDEX

Wenn Indizes definiert sind und durch die Angabe PRESERVE ROWS Sätze verschoben wurden, dann werden die Indizes der Tabelle neu aufgebaut. Dies gilt auch für Indizes, die bereits vor der Anweisung ALTER PARTITIONING als defekt gekennzeichnet waren.

Die betroffenen Index-Spaces erhalten den Zustand „copy pending“ wenn sie der logischen Datensicherung unterliegen.

Wenn ONLY WHEN EMPTY angegeben wurde oder durch die Angabe PRESERVE ROWS keine Sätze verschoben wurden, dann werden die Indizes nur dann neu aufgebaut, wenn es zuvor schon als defekt gekennzeichnete Indizes gab. Sonst werden die Indizes nicht neu aufgebaut.

### NO INDEX

Wenn Indizes definiert sind und durch die Angabe PRESERVE ROWS Sätze verschoben wurden, dann werden die Sekundärindizes der Tabelle als defekt gekennzeichnet und nicht neu aufgebaut.

Wenn ONLY WHEN EMPTY angegeben wurde oder durch die Angabe PRESERVE ROWS keine Sätze verschoben wurden, dann werden die Indizes nicht neu aufgebaut. Sie behalten ihren ursprünglichen Zustand.

**ALTER PARTITIONING FOR TABLE ... DROP PARTITION**

Diese Anweisung löscht eine Partition.

Eine partitionierte Basistabelle wird dabei in eine nicht-partitionierte Basistabelle umgewandelt, wenn sie nach dem Löschen nur noch eine Partition besitzt.

---

ALTER PARTITIONING FOR TABLE *tabelle*

DROP PARTITION ON SPACE *space*

DROP UPPER LIMIT OF { DROPPED }  
PRIOR } PARTITION

{ DELETE ROWS [DEFERRED]  
ONLY WHEN EMPTY  
PRESERVE ROWS } ] [ { GENERATE INDEX }  
NO INDEX } ]

---

*tabelle*

Name einer partitionierten Basistabelle. Sie darf sich nicht im Zustand „check pending“ befinden.

DROP PARTITION ON SPACE *space*

Name des Anwender-Space mit der zu löschenden Partition.

DROPPED

Die Obergrenze der zu löschenden Partition wird gelöscht. Die Klausel darf nicht für die höchste Partition angegeben werden.

PRIOR

Die Obergrenze der zu löschenden Partition wird als Obergrenze der nächstniedrigeren Partition verwendet. Die Klausel darf nicht für die niedrigste Partition angegeben werden.

ONLY WHEN EMPTY

Die zu löschende Partition darf keine Sätze enthalten.

### PRESERVE ROWS

Die Sätze der zu löschenden Partition bleiben erhalten.

Bei Angabe von PRIOR werden sie in die nächstniedrigere Partition verschoben. Bei Angabe von DROPPED werden sie in die nächsthöhere Partition verschoben.

Wenn Sätze von einer Partition in eine andere übertragen werden, dann werden die Satznummern gemäß dem Präfix ROW\_ID\_PREFIX (siehe View SYS.PARTITIONS im SYS\_INFO\_SCHEMA, Handbuch [SQL-Sprachbeschreibung Teil 1: SQL-Anweisungen](#)) angepasst.

Die betroffenen Anwender-Spaces erhalten den Zustand „copy pending“, wenn sie der logischen Datensicherung unterliegen.

Die maximal mögliche Anzahl von Sätzen in einer Partition (ca. 268 Mio.) darf nur dann überschritten werden, wenn die Tabelle nach dem Löschen nur noch eine Partition besitzt und in der Folge von SESAM/SQL in eine nicht-partitionierte Basistabelle umgewandelt wird.

### DELETE ROWS [DEFERRED]

Die Sätze der zu löschenden Partition werden gelöscht.

Die Tabelle darf aber nicht durch eine Referenzbedingung einer anderen Basistabelle referenziert sein.

Die Klausel DEFERRED bewirkt ein schnelles Löschen der Sätze.

Es wird nur der zusammenhängende Teil der Partition gelöscht. Eventuell vorhandene Auslagerungen bleiben erhalten. Bei der nächsten Reorganisation des Anwender-Spaces mit der Utility-Anweisung REORG SPACE werden alle existierenden Tabellen und Indizes auf dem Anwender-Space neu aufgebaut. Damit verschwinden auch die nicht gelöschten Auslagerungen.

Informationen zur Speicherstruktur von Basistabellen finden Sie im [„Basishandbuch“](#).

Die betroffenen Anwender-Spaces erhalten den Zustand „copy pending“, wenn sie der logischen Datensicherung unterliegen.

### GENERATE INDEX

Wenn Indizes definiert sind und durch die Angabe PRESERVE ROWS Sätze verschoben wurden oder DELETE ROWS angegeben ist, dann werden die Indizes der Tabelle neu aufgebaut. Dies gilt auch für Indizes, die bereits vor der Anweisung ALTER PARTITIONING als defekt gekennzeichnet waren.

Die betroffenen Index-Spaces erhalten den Zustand „copy pending“ wenn sie der logischen Datensicherung unterliegen.

Wenn ONLY WHEN EMPTY angegeben wurde oder durch die Angabe PRESERVE ROWS keine Sätze verschoben wurden, dann werden die Indizes nur dann neu aufgebaut, wenn es zuvor schon als defekt gekennzeichnete Indizes gab. Sonst werden die Indizes nicht neu aufgebaut.

Wenn durch die Anweisung ALTER PARTITIONING aus einer partitionierten Tabelle eine nicht-partitionierte Tabelle entsteht, dann werden die Sekundärindizes der Tabelle neu aufgebaut.

Die betroffenen Index-Spaces erhalten den Zustand „copy pending“ wenn sie der logischen Datensicherung unterliegen.

### NO INDEX

Wenn Indizes definiert sind und durch die Angabe PRESERVE ROWS Sätze verschoben wurden oder DELETE ROWS angegeben ist, dann werden die Sekundärindizes der Tabelle als defekt gekennzeichnet und nicht neu aufgebaut.

Wenn ONLY WHEN EMPTY angegeben wurde und durch die Angabe PRESERVE ROWS keine Sätze verschoben wurden, dann werden die Indizes nicht neu aufgebaut. Sie behalten ihren ursprünglichen Zustand.

Wenn durch die Anweisung ALTER PARTITIONING aus einer partitionierten Tabelle eine nicht-partitionierte Tabelle entsteht, dann werden die Sekundärindizes der Tabelle als defekt gekennzeichnet und nicht neu aufgebaut.

## Beispiele

1. Das Beispiel fügt der nicht-partitionierten Tabelle KUNDE eine Partition hinzu.

```
ALTER PARTITIONING FOR TABLE kunde
  ADD PARTITION ON SPACE spaceneu
  SET UPPER LIMIT OF ADDED PARTITION TO VALUE <= (999)
  ONLY WHEN EMPTY
```

Die Tabelle KUNDE wird dadurch zu einer partitionierten Tabelle mit 2 Partitionen. Die niedrigere Partition erhält die Obergrenze 999. Die Obergrenze der höheren Partition ergibt sich aus dem höchsten Primärschlüsselwert.

2. Das Beispiel ändert die Obergrenze der oben eingerichteten Partition.

```
ALTER PARTITIONING FOR TABLE kunde
  ALTER PARTITION ON SPACE spaceneu
  SET UPPER LIMIT TO VALUE <= (500)
  PRESERVE ROWS GENERATE INDEX
```

Die Datensätze werden verschoben. Indizes werden ggf. neu aufgebaut.

3. Das Beispiel löscht eine Partition der partitionierten Tabelle KUNDE.

```
ALTER PARTITIONING FOR TABLE kunde
  DROP PARTITION ON SPACE spaceneu
  DROP UPPER LIMIT OF PRIOR PARTITION
  DELETE ROWS DEFERRED GENERATE INDEX
```

Die Tabelle KUNDE aus Beispiel 1 wird dadurch wieder zu einer nicht-partitionierten Tabelle. Die Partition wird gelöscht. Auslagerungen bleiben zunächst erhalten. Indizes werden neu aufgebaut.

4. Die Partitionen der Basistabelle UMSATZ seien nach den Monaten des Jahres organisiert. Der Primärschlüssel sei eine Spalte *umsatzdatum* vom Datentyp DATE.

```
CREATE TABLE umsatz ...
USING PARTITION BY RANGE PARTITION 1 VALUE <= DATE'2012-01-31' ON SPACE sp1
USING PARTITION BY RANGE PARTITION 2 VALUE <= DATE'2012-02-29' ON SPACE sp2
...
USING PARTITION BY RANGE PARTITION 11 VALUE <= DATE'2012-11-30'
                                                ON SPACE sp11
USING PARTITION BY RANGE PARTITION 12 VALUE <= () ON SPACE sp12
```

Zu Beginn des Jahres 2013 sollen nun die Daten vom Januar 2012 gelöscht werden. Der freiwerdende Space soll für die Daten vom Januar 2013 genutzt werden. Diese Aufgabe kann komfortabel mit der Utility-Anweisung ALTER PARTITIONING FOR TABLE ausgeführt werden.

Im ersten Schritt wird mit folgender Utility-Anweisung die Partition 1 aus den Metadaten der Tabelle *umsatz* entfernt. Die Datensätze auf dem Space *sp1* werden gelöscht.

```
ALTER PARTITIONING FOR TABLE umsatz
    DROP PARTITION ON SPACE sp1 DELETE ROWS
```

Im zweiten Schritt wird eine neue (unbegrenzte) Partition eingerichtet. Die vorangehende Partition erhält als neue Obergrenze den 31.12.2012. Eventuell bereits angefallenen Umsätze für das Jahr 2013 werden in die neue Partition übertragen.

```
ALTER PARTITIONING FOR TABLE umsatz
    ADD PARTITION ON SPACE sp1
    SET UPPER LIMIT OF PRIOR PARTITION TO VALUE <= DATE'2012-12-31'
    PRESERVE ROWS
```

SESAM/SQL passt die Partitionsnummern automatisch an, so dass die geänderte Tabelle *umsatz* folgende Partitionen enthält:

```
PARTITION 1 VALUE <= DATE'2012-02-29' ON SPACE sp2
PARTITION 2 VALUE <= DATE'2012-03-31' ON SPACE sp3
...
PARTITION 11 VALUE <= DATE'2012-12-31' ON SPACE sp12
PARTITION 12 VALUE <= () ON SPACE sp1
```

## CHECK CONSTRAINTS - Integritätsbedingungen prüfen

Mit CHECK CONSTRAINTS können Sie prüfen, ob Integritätsbedingungen erfüllt sind. Eine Prüfung der Integritätsbedingungen mit CHECK CONSTRAINTS für Anwender-Spaces und Basistabellen ist immer notwendig, wenn für die angegebenen Tabellen Integritätsbedingungen definiert sind und sich nach Ausführung der Utility-Anweisung LOAD mit Angabe NO CONSTRAINT CHECK der Datenbestand geändert hat:

SESAM/SQL versetzt die von der Utility-Anweisung betroffenen Tabellen bzw. den bearbeiteten Anwender-Space in den Zustand „check pending“ (siehe [Seite 13](#)).

Damit die gesperrten Tabellen wieder mit SQL-Anweisungen bearbeitet werden können, müssen Sie CHECK CONSTRAINTS ausführen. Bei erfolgreicher Prüfung macht SESAM/SQL die Tabellen wieder allgemein verfügbar. Andernfalls verbleiben die Tabellen im Zustand „check pending“.

Wenn sich keine Tabelle auf einem Anwender-Space mehr im Zustand „check pending“ befindet, dann befindet sich auch der Anwender-Space nicht länger im Zustand „check pending“.

Wie Sie mit Tabellen verfahren, die nach der Ausführung von CHECK CONSTRAINTS im Zustand „check pending“ verbleiben, ist im „[Basishandbuch](#)“ beschrieben.

Der aktuelle Berechtigungsschlüssel muss das Sonder-Privileg UTILITY besitzen oder Eigentümer der zu prüfenden Tabellen bzw. Anwender-Spaces sein.

Wird CHECK CONSTRAINTS auf partitionierte Tabellen (ON TABLE) bzw. Partitionen auf Spaces (ON SPACE) angewandt, dann müssen alle betroffenen Partitionen verfügbar sein, siehe [Tabelle 3 auf Seite 22](#).

---

```
CHECK CONSTRAINTS { integritätsbedingungsname , ... } [ { SCOPE ALL } ]
                   { ON TABLE tabelle , ... }
                   { ON SPACE space , ... }
```

---

*integritätsbedingungsname* ,...

Name einer Tabellenbedingung oder Spaltenbedingung, die geprüft werden soll bzw.

Namen von Tabellen- oder Spaltenbedingungen, die geprüft werden sollen.

Jeder *integritätsbedingungsname* darf nur einmal angegeben werden.

**ON TABLE** *tabelle* ,...

Name einer Basistabelle bzw. Namen mehrerer Basistabellen. Jeder Tabellename darf nur einmal angegeben werden.

Für einige (siehe [SCOPE PENDING](#)) oder alle angegebenen Basistabellen (siehe [SCOPE ALL](#)) werden alle für sie definierten Tabellen- und Spaltenbedingungen geprüft. Bei erfolgreicher Prüfung versetzt SESAM/SQL die Tabellen, die sich im Zustand „check pending“ befinden, in den Zustand „space o.k.“. Stellt SESAM/SQL die Verletzung einer Integritätsbedingung fest, so versetzt SESAM/SQL alle geprüften Basistabellen in den Zustand „check pending“ bzw. belässt sie im Zustand „check pending“.

**ON SPACE** *space* ,...

Name eines Anwender-Space bzw. Namen mehrerer Anwender-Spaces. Maximal 1000 Spacenames dürfen angegeben werden. Jeder Spacename darf nur einmal angegeben werden.

Für einige (siehe [SCOPE PENDING](#)) oder alle Basistabellen (siehe [SCOPE ALL](#)) der angegebenen Anwender-Spaces werden alle für sie definierten Tabellen- und Spaltenbedingungen geprüft. Bei erfolgreicher Prüfung versetzt SESAM/SQL Tabellen, die sich im Zustand „check pending“ befinden, in den Zustand „space o.k.“. Stellt SESAM/SQL die Verletzung einer Integritätsbedingung fest, so versetzt SESAM/SQL alle Basistabellen der geprüften Anwender-Spaces in den Zustand „check pending“ bzw. belässt sie im Zustand „check pending“.

**SCOPE ALL**

Die Integritätsbedingungen aller angegebenen Tabellen werden geprüft.

Folgendes ist zu beachten:

Bei Verletzung einer Integritätsbedingung für eine Tabelle erhalten alle zu prüfenden Tabellen den Zustand „check pending“, auch wenn bei diesen Tabellen keine Verletzung von Integritätsbedingungen festgestellt wurden.

SCOPE ALL ist nur wirksam in Verbindung mit den Klauseln ON TABLE und ON SPACE.

**SCOPE PENDING**

Von den angegebenen Tabellen bzw. von den Tabellen der angegebenen Anwender-Spaces werden nur diejenigen geprüft, die sich im Zustand „check pending“ befinden. SCOPE PENDING ist nur wirksam in Verbindung mit den Klauseln ON TABLE und ON SPACE.

## Beispiel

Das Beispiel prüft alle für die Basistabelle AUFTRAG definierten Integritätsbedingungen, wenn sich die Tabelle im Zustand „check pending“ befindet. Die Tabelle AUFTRAG ist definiert im Schema AUFTRAGSVER der Datenbank AUFTRAGKUNDEN.



```
CHECK CONSTRAINTS ON TABLE auftragkunden.auftragsver.auftrag
```

Stellt SESAM/SQL die Verletzung einer für AUFTRAG definierten Integritätsbedingung fest, so verbleibt AUFTRAG im Zustand „check pending“. Andernfalls versetzt SESAM/SQL die Tabelle AUFTRAG in den Zustand „space o.k.“.

## Siehe auch

CHECK FORMAL

## CHECK FORMAL - Formalprüfung von Tabellen und Indizes

Mit CHECK FORMAL können Sie die formale Korrektheit prüfen. Die Formalprüfung umfasst im Wesentlichen die Prüfung SESAM/SQL-interner Blöcke und Tabellen.

Wenn eine Inkonsistenz entdeckt wird, legt SESAM/SQL eine Fehlerdatei an (siehe [Seite 95](#)). Eine durch CHECK FORMAL entdeckte Inkonsistenz wird von SESAM/SQL als SQLSTATE gemeldet. Der Space bzw. Index erhält den Zustand „defekt“. Sofern möglich, wird die Prüfung durch CHECK FORMAL anschließend fortgesetzt.

CHECK FORMAL wird auch auf Replikate und auf SESAM-Sicherungsbestände zugelassen, die im SQL-Datenbankverzeichnis eingetragen sind. Ein CHECK FORMAL auf Fremdkopien ist möglich, wenn die Fremdkopie als Datenbank im SQL-Datenbankverzeichnis eingetragen ist. Damit sie ihre Eigenschaft als Fremdkopie nicht verliert, muss sie mit ACCESS=READ eingetragen werden.

Die Datenbank, deren Spaces, Tabellen oder Indizes geprüft werden sollen, muss im SQL-Datenbankverzeichnis mit ACCESS=ADMIN eingetragen werden, damit Spaces (bei defekten Tabellen) und defekte Indizes als „defekt“ gekennzeichnet werden. Ist die Datenbank mit ACCESS=READ, als Sicherungsbestand, als Replikat oder als Fremdkopie eingetragen, so werden Spaces (bei defekten Tabellen) und defekte Indizes nicht als „defekt“ markiert. Hinweise auf den Fehler erhalten Sie lediglich im SQLSTATE und in der Fehlerdatei.

Der aktuelle Berechtigungsschlüssel muss eine der folgende Eigenschaften besitzen:

- Er muss das Sonder-Privileg UTILITY besitzen
- Er muss Eigentümer des zu prüfenden Space sein
- Er muss Eigentümer des Space sein, auf dem die zu prüfende Basistabelle bzw. der zu prüfende Index liegt.

---

```
CHECK FORMAL { TABLE tabelle
                INDEX index
                SPACE space } [NO ACTION]
```

---

### TABLE *tabelle*

Name der Basistabelle, die geprüft werden soll.

Bei partitionierten Tabellen werden alle Partitionen der Tabelle geprüft. Ist eine Partition nicht verfügbar (siehe [Tabelle 3 auf Seite 22](#)), so wird dies in der Fehlerdatei protokolliert (siehe [Seite 95](#)). Die Prüfung wird mit der nächsten Partition fortgesetzt.

Wird bei der Prüfung ein Fehler festgestellt, so wird der Anwender-Space, auf dem die Tabelle oder die Partition liegt, als „defekt“ gekennzeichnet.

**INDEX** *index*

Name des Index, der geprüft werden soll.

Wird bei der Prüfung ein Fehler festgestellt, so wird der Index als „defekt“ gekennzeichnet.

**SPACE** *space*

Name des Space, dessen Basistabellen, Partitionen und Indizes geprüft werden sollen. Beim Catalog-Space muss der Spacename CATALOG als *spezialname* angegeben werden.

Der Space wird als „defekt“ gekennzeichnet, wenn bei der Prüfung einer Basistabelle oder Partition des Space ein Fehler festgestellt wird. Wird bei der Prüfung eines Index des Space ein Fehler festgestellt, so wird nur der Index als „defekt“ gekennzeichnet.

**NO ACTION**

Bei der Angabe dieser Klausel entfällt die Defektkennzeichnung von Spaces bei defekten Tabellen. Hinweise auf den Fehler erhalten Sie lediglich im SQLSTATE und in der Fehlerdatei. Defekte Indizes werden jedoch weiterhin als „defekt“ gekennzeichnet.

Mit defekten Indizes kann weiter gearbeitet werden, sie werden nicht mehr verwendet. Eine weitere Bearbeitung von defekten Tabellen kann schwere Fehler verursachen, die auch zum Absturz des DBH führen können. Deshalb sollte unbedingt ein RECOVER vorgenommen werden.

**Fehlerdatei bei CHECK FORMAL**

SESAM/SQL legt beim CHECK FORMAL bei Bedarf eine neue Datei an oder schreibt die bestehende fort. Der Dateiname lautet:

*catalog.space*. EXC.C

*catalog* Name der Datenbank.

*space* Name des Anwender-Space, auf dem die Prüfung durchgeführt wird.

Bei CHECK FORMAL TABLE für eine partitionierte Tabelle wird für jede fehlerhafte Partition eine Fehlerdatei mit dem Spacenamens der Partition gebildet.

EXC.C Standard-Endung des Namens der Fehlerdatei für CHECK-Anweisungen

Wenn *catalog* in einer DB-Kennung liegt, so wird die Fehlerdatei in der DB-Kennung angelegt, sofern die Vorbereitungen dafür getroffen worden sind, siehe Abschnitt „Datenbankdateien und Jobvariablen auf fremden Benutzerkennungen“ im „[Basishandbuch](#)“. Kann die Fehlerdatei nicht in der DB-Kennung angelegt werden, so wird sie in der DBH-Kennung angelegt.

**Beispiel**

Das Beispiel führt die Formalprüfung für die Basistabelle AUFTRAG durch. Die Tabelle AUFTRAG gehört zum Schema AUFTRAGSVER der Datenbank AUFTRAGKUNDEN.



```
CHECK FORMAL TABLE auftragkunden.auftragsver.auftrag
```

**Siehe auch**

CHECK CONSTRAINTS

## COPY - SESAM-Sicherungsbestand erstellen

Mit COPY können Sie SESAM-Sicherungsbestände der gesamten SESAM/SQL-Datenbank, des Catalog-Space oder eines oder mehrerer Anwender-Spaces erstellen. Die SESAM-Sicherungsbestände können Sie auf Platte oder Magnetbandkassette anlegen.

Mit COPY können Sie SESAM-Sicherungsbestände online oder offline erstellen. Ergänzende Informationen zur Online- und Offline-Erstellung von SESAM-Sicherungsbeständen finden Sie im „[Basishandbuch](#)“.

SESAM-Sicherungsbestände auf Magnetbandkassette können bei COPY wahlweise mit Hilfe der Softwareprodukte HSMS oder ARCHIVE erstellt werden (siehe [Abschnitt „Band-sicherungen mit HSMS und ARCHIVE“ auf Seite 36](#)).

Eine Sicherung mit HSMS kann auch von der Additional-Mirror-Unit eines Multi-Mirror-Paares der Local-Symmetrix oder von der Additional-Mirror-Unit der Target-Unit eines Remote-Copy-Paares einer Remote-Symmetrix durchgeführt werden (siehe „[Basishandbuch](#)“). Soweit in der folgenden Beschreibung von Additional-Mirror-Units (BCVs) gesprochen wird, sind Additional-Mirror-Units oder Clone-Units gemeint.

Als Aufsetzpunkt für Replikate wird bei jedem COPY CATALOG und COPY CATALOG\_SPACE eine CAT-REC-Datei mit dem festen Namen *catalog.CAT-REC.COPY* auf den gleichen Medien angelegt wie die CAT-REC-Datei. Es werden keine Metadaten über diese zusätzliche Kopie hinterlegt.

Falls das Logging eingeschaltet ist, ist mit COPY CATALOG und COPY CATALOG\_SPACE stets ein Wechsel der CAT-LOG-Datei und der DA-LOG-Datei verbunden. Für eine Datenbank, die ohne Logging betrieben wird, legt SESAM/SQL beim ersten COPY CATALOG bzw. COPY CATALOG\_SPACE die CAT-REC-Datei und die CAT-REC-Kopie gemäß den Angaben des CAT-REC-Medienatzes in der Medientabelle an. Falls kein CAT-REC-Medienatz existiert, verwendet SESAM/SQL Standardwerte (siehe „[Basishandbuch](#)“).

Liegen auf einem Space Tabellen und Indizes, kann dieser Space auch dann gesichert werden, wenn einer oder mehrere der Indizes defekt sind. SESAM/SQL gibt bei der Sicherung eines solchen Space aber die Warnung aus, dass Indizes defekt sind.

Eine Sicherung, auf der sich defekte Indizes befinden, kann für ein RECOVER oder CREATE REPLICATION genutzt werden. Die defekten Indizes müssen anschließend gesondert neu aufgebaut werden.

Der aktuelle Berechtigungsschlüssel muss das Sonder-Privileg UTILITY besitzen oder Eigentümer des zu kopierenden Anwender-Space sein.

Die Umstellung von Winter- auf Sommerzeit bzw. von Sommer- auf Winterzeit erfolgt bei SESAM/SQL ohne Unterbrechung des laufenden Betriebs. Während der kritischen Phase der Zeitumstellung von der Sommer- auf die Winterzeit (letzte Stunde der Sommerzeit und erste Stunde der Winterzeit) sind jedoch keine SESAM-Sicherungsbestände erlaubt. SESAM-Sicherungsbestände während dieser kritischen Phase der Zeitumstellung werden von SESAM/SQL abgewiesen.

## Datenbanken auf einer DB-Kennung ungleich der DBH-Kennung

Liegt eine Datenbank auf einer DB-Kennung ungleich der DBH-Kennung, dann versucht SESAM/SQL bei COPY auf Platte die Sicherungsdateien in der DB-Kennung anzulegen. Dazu müssen Vorbereitungen getroffen worden sein, siehe Abschnitt „Datenbankdateien und Jobvariablen auf fremden Benutzerkennungen“ im „[Basishandbuch](#)“.

Wenn nicht alle Dateien in der DB-Kennung angelegt werden können, wird die Anweisung abgewiesen.

Können die Sicherungsdateien nicht in der DB-Kennung angelegt werden, dann werden sie in der DBH-Kennung angelegt.

Wenn der Sicherungsbestand bei COPY mit dem Softwareprodukt HSMS erstellt werden soll, dann muss auf der DB-Kennung die DBH-Kennung für alle Datenbankdateien als Mit-eigentümer definiert werden.

ARCHIVE kann nur verwendet werden, wenn die Datenbank in der DBH-Kennung liegt.

Die CAT-REC-Datei und die CAT-REC-Kopie werden in der DB-Kennung angelegt, sofern die Vorbereitungen dafür getroffen worden sind, siehe Abschnitt „Datenbankdateien und Jobvariablen auf fremden Benutzerkennungen“ im „[Basishandbuch](#)“. Die CAT-REC-Datei muss immer in der DB-Kennung angelegt werden können. Kann die CAT-REC-Kopie nicht in der DB-Kennung angelegt werden, dann wird sie in der DBH-Kennung angelegt.

### *COPY-Anweisung*

---

```

COPY { CATALOG catalog
      CATALOG_SPACE catalog
      SPACE space , . . . } [ { ONLINE
                             OFFLINE } ]
      [ USING { STOGROUP stogroup
              DIRECTORY { hsms_archiv_name
                          archive_directory_datei } [ { BY_ADD_MIRROR_UNIT
                                                         BY_SRDF_TARGET } ] } ]
      [[NO] CHECK FORMAL] [LOG] [EXCEPT NO LOG INDEX SPACE]
  
```

---

### CATALOG *catalog*

Name der Datenbank, von der ein SESAM-Sicherungsbestand erstellt werden soll.

Die Sicherung der Datenbank umfasst die Sicherung des zugehörigen Catalog-Space und aller Anwender-Spaces der Datenbank.

Mit EXCEPT NO LOG INDEX SPACE werden keine Spaces gesichert, auf denen nur Indizes liegen und die nicht in der logischen Datensicherung sind. Ist diese Klausel eingeschaltet, kann die Sicherung nicht als Sicherungsbestand zum Retrieval im SQL-Datenbankverzeichnis eingetragen werden.

**CATALOG\_SPACE** *catalog*

Name der Datenbank, deren Catalog-Space gesichert werden soll.

**SPACE** *space ,...*

Name des zu sichernden Anwender-Space bzw. Namen der zu sichernden Anwender-Spaces. Maximal dürfen Sie 999 Spacennamen angeben. Jeden Spacennamen dürfen Sie nur einmal angeben. Als Spacennamen dürfen Sie nicht CATALOG angeben. Die Spaces werden absteigend nach ihrer Space-Größe sortiert und in dieser Reihenfolge gesichert.

**ONLINE**

SESAM-Sicherungsbestand wird online erstellt.

Während der Online-Sicherung können SQL-Anweisungen zum Lesen und Ändern von Daten sowie CALL-DML-Anweisungen auf den von der Sicherung betroffenen Anwender-Spaces ausgeführt werden. Bei der Angabe von ONLINE wird die PBI-Datei (bei Plattenkopie und ARCHIVE) bzw. die Arbeitsdatei (bei HSMS) immer in der Kennung angelegt, in der der DBH gestartet wurde. Es wird das Medium verwendet, das in der Medienbeschreibung für PBI definiert wurde. Bei Sicherungen von einer Spiegelplatte (BY\_ADD\_MIRROR\_UNIT, BY\_SRDF\_TARGET) wird keine PBI- bzw. Arbeitsdatei angelegt.

Soll die Online-Sicherung in ein HSMS-Archiv erfolgen, so dürfen die zu sichernden Datenbank-Dateien nicht auf Privatplatte liegen.

**OFFLINE**

SESAM-Sicherungsbestand wird offline erstellt.

Während der Offline-Sicherung sperrt SESAM/SQL die von der Sicherung betroffenen Anwender-Spaces gegen ändernde Zugriffe anderer Benutzer.

**USING STOGROUP** *stogroup*

Der SESAM-Sicherungsbestand wird als Plattenkopie auf die Storage Group *stogroup* gelegt. Die SESAM-Sicherungsbestände werden entweder in der DB-Kennung angelegt oder in der DBH-Kennung.

Wenn sich die Datenbank auf einer DB-Kennung befindet, dann werden die Sicherungsdateien ebenfalls in der DB-Kennung angelegt, sofern in der DB-Kennung die DBH-Kennung als Miteigentümer definiert ist oder die Dateien vorher mit dem Kommando CREATE-FILE in der DB-Kennung angelegt wurden.

Wurden die Dateien vorher mit dem Kommando CREATE-FILE in der DB-Kennung angelegt, dann wird USING STOGROUP ignoriert.

USING DIRECTORY  $\left\{ \begin{array}{l} \text{hsms\_archiv\_name} \\ \text{archive\_directory\_datei} \end{array} \right\}$

SESAM/SQL erstellt mit Hilfe der Softwareprodukte HSMS oder ARCHIVE einen SESAM-Sicherungsbestand auf Magnetbandkassette.

Die Sicherung erfolgt durch Aufruf von HSMS bzw. ARCHIVE als Unterprogramm.

*hsms\_archiv\_name* bzw. *archive\_directory\_datei* muss als alphanumerisches Literal angegeben werden.

Wird eine HSMS-Parameterdatei gefunden, erfolgt die Sicherung mit HSMS.

*hsms\_archiv\_name* bezeichnet das zu nutzende HSMS-Archiv. Die Länge des Namens ist auf 22 Zeichen (ohne Benutzerkennung auf 12 Zeichen) beschränkt.

Der Name der HSMS-Parameterdatei ergibt sich aus dem Namen des HSMS-Archivs oder einer Kombination aus Konfigurationsnamen und DBH-Namen (Konnektionsmodul-Parameter CNF bzw. NAM), hier *k* und *n* genannt:

*hsms\_archiv\_name*.HSMS-PAR oder SESAM*kn*.HSMS-PAR

Zunächst sucht SESAM/SQL die Datei *hsms\_archiv\_name*.HSMS-PAR, danach sessionübergreifend die Datei SESAM*kn*.HSMS-PAR.

Die HSMS-Parameterdatei und das HSMS-Archiv müssen in der BS2000-Benutzerkennung des DBH angelegt worden sein. Die HSMS-Parameterdatei darf leer sein. Sie hat dann lediglich die Aufgabe, HSMS als Sicherungsverfahren auszuwählen.

SESAM/SQL legt eine Protokolldatei unter dem Namen *hsms\_archiv\_name*.SYSLST an. Bei weiteren COPY-Anweisungen wird die Datei fortgeschrieben.

Wenn sich die Datenbank auf einer DB-Kennung befindet und die Sicherung mit HSMS erfolgen soll, dann muss in der DB-Kennung die DBH-Kennung als Miteigentümer definiert sein. Es genügt nicht, wenn die Datenbankdateien als mehrbenutzbar angelegt sind.

Wird keine HSMS-Parameterdatei gefunden, erfolgt die Sicherung mit ARCHIVE.

*archive\_directory\_datei* bezeichnet ein ARCHIVE-Directory. Für ARCHIVE kann ebenfalls eine Parameterdatei angegeben werden. Ihr Name wird analog zu dem Verfahren bei HSMS gebildet:

*archive\_directory\_datei*.ARC-PAR oder SESAM*kn*.ARC-PAR

Das ARCHIVE-Directory und die ARCHIVE-Parameterdatei müssen in der BS2000-Benutzerkennung liegen, in der der DBH abläuft.

ARCHIVE legt eine Protokolldatei unter dem Namen *archive\_directory\_datei*.SYSLST an, die bei weiteren COPY-Anweisungen fortgeschrieben wird.

USING DIRECTORY darf bei einer Sicherung mit ARCHIVE nur im Zusammenhang mit Datenbanken angegeben werden, die in der BS2000-Benutzerkennung liegen, in der der DBH abläuft.

**BY\_ADD\_MIRROR\_UNIT**

Mit der Angabe von **BY\_ADD\_MIRROR\_UNIT** können Datenbank-Dateien, die auf der Additional-Mirror-Unit eines Multi-Mirror-Paares der Local-Symmetrix liegen, in ein HSMS-Archiv gesichert werden. Dabei braucht sich der Anwender weder um das Abspalten noch um die anschließende Synchronisation der Additional-Mirror-Unit zu kümmern.

Die Angabe dieser Klausel ist nur erlaubt, wenn die Sicherung mit HSMS erfolgt und die Datenbankdateien auf derselben Additional-Mirror-Unit liegen. Die Benutzerkennung, auf der der DBH läuft, muss das Privileg **HSMS-ADMINISTRATION** besitzen. Nähere Informationen zur Sicherung mit HSMS von Additional-Mirror-Units finden Sie im „[Basishandbuch](#)“.

**BY\_SRDF\_TARGET**

Mit der Angabe von **BY\_SRDF\_TARGET** können Datenbank-Dateien, die auf der Additional-Mirror-Unit einer Target-Unit eines Remote-Copy-Paares in einer Remote-Symmetrix liegen, in ein HSMS-Archiv gesichert werden. Dabei braucht sich der Anwender weder um das Abspalten noch um die anschließende Synchronisation der Additional-Mirror-Unit, die der Target-Unit zugewiesen ist, zu kümmern.

Die Angabe dieser Klausel ist nur erlaubt, wenn die Sicherung mit HSMS erfolgt und die Datenbankdateien auf derselben Additional-Mirror-Unit in der Remote-Symmetrix liegen. Das HSMS-Archiv muss auf dem System liegen, an das auch die Source-Unit angeschlossen ist. Die Benutzerkennung, auf der der DBH läuft, muss das Privileg **HSMS-ADMINISTRATION** besitzen.

Nähere Informationen zur Sicherung mit HSMS von Additional-Mirror-Units finden Sie im „[Basishandbuch](#)“.

**USING-Klausel nicht angeben:**

SESAM/SQL legt einen SESAM-Sicherungsbestand (Plattenkopie) auf der Default Storage Group **D0STOGROUP** an. Falls die Default Storage Group nicht verfügbar ist, wird die Utility-Anweisung **COPY** abgewiesen.

### [NO] CHECK FORMAL

Ist CHECK FORMAL angegeben, so wird bei der Erstellung eines SESAM-Sicherungsbestandes auf Platte (online oder offline) die Prüfung auf formale Korrektheit durchgeführt, indem SESAM/SQL automatisch die Utility-Anweisung CHECK FORMAL für den SESAM-Sicherungsbestand ausführt.

Bei der Online-Erstellung eines SESAM-Sicherungsbestandes mit HSMS von einer Additional-Mirror-Unit führt SESAM/SQL die Formalprüfung auf der abgetrennten Additional-Mirror-Unit durch, während die Dateien in ein HSMS-Archiv gesichert werden.

Wenn die Source- oder Normal-Unit zu einem shared Pubset gehört, dann muss der SESAM/SQL-DBH auf dem Server ablaufen, der für das Shared Pubset die Master-Seite darstellt.

Bei der Offline-Erstellung eines SESAM-Sicherungsbestandes auf Magnetbandkassette führt SESAM/SQL die Formalprüfung auf dem Original bereits vor der Kopieerstellung durch.

Wird bei der Formalprüfung in den zuvor genannten Fällen ein Fehler gefunden, so war die Sicherung nicht erfolgreich; der Originalspace wird aber nicht als „defekt“ gekennzeichnet.

Die Angabe von CHECK FORMAL zusammen mit den Klauseln ONLINE und USING DIRECTORY ist nur zulässig, wenn entweder die Klausel BY\_ADD\_MIRROR\_UNIT oder BY\_SRDF\_TARGET angegeben ist.

Sonst ist bei der Online-Erstellung eines SESAM-Sicherungsbestandes auf Magnetbandkassette die Formalprüfung nicht möglich. In diesem Fall sollte die Utility-Anweisung CHECK FORMAL vor der COPY-Anweisung eingegeben werden.

Ist [NO] CHECK FORMAL nicht angegeben, so gelten folgende Voreinstellungen: Falls gleichzeitig die Klauseln ONLINE und USING DIRECTORY angegeben sind, gilt NO CHECK FORMAL. In allen anderen Fällen gilt CHECK FORMAL.

### LOG

Nach COPY CATALOG ... LOG ist das Logging für alle Anwender-Spaces sowie den Catalog-Space der Datenbank eingeschaltet.

LOG schaltet nach Erstellung des SESAM-Sicherungsbestandes für einen Anwender-Space das zuvor mit der SQL-Anweisung ALTER SPACE für diesen Anwender-Space ausgeschaltete Logging wieder ein. Im erstellten SESAM-Sicherungsbestand ist das Logging noch ausgeschaltet.

Mit COPY SPACE ... LOG lässt sich darüber hinaus auch das Logging für einen Anwender-Space einschalten, für den mit CREATE SPACE NO LOG zunächst kein Logging vereinbart war.

LOG nicht angegeben:

Die aktuelle Vereinbarung für das Logging bleibt unverändert.

**EXCEPT NO LOG INDEX SPACE**

Nur möglich bei COPY CATALOG.

Wenn Sie diese Klausel angeben, werden von der Sicherung die Spaces ausgenommen, für die die beiden folgenden Bedingungen gleichzeitig gelten:

- Die Spaces sind reine Index-Spaces und
- die logische Datensicherung für diese Spaces ist ausgeschaltet.

Eine Sicherung, die mit COPY CATALOG ... EXCEPT NO LOG INDEX SPACE erstellt wurde, kann nicht als Sicherungsbestand zum Retrieval im SQL-Datenbankverzeichnis eingetragen werden.

Aus einer mit COPY CATALOG ... EXCEPT NO LOG INDEX SPACE erstellten Sicherung können Sie ein Teilreplikat erzeugen. Dieses Teilreplikat kann zur Recovery genutzt und auch mit REFRESH REPLICATION aktualisiert werden. Für ein Retrieval kann dieses Teilreplikat jedoch nicht eingesetzt werden.

**BS2000-Dateinamen für SESAM-Sicherungsbestände und CAT-REC-Kopie**

Catalog-Space:                 :*catid:bs2000\_benutzerkennung.catalog.CATALOG.nnnnnn*

Anwender-Spaces:           :*catid:bs2000\_benutzerkennung.catalog.space.nnnnnn*

CAT-REC-Kopie               :*catid:bs2000\_benutzerkennung.catalog.CAT-REC.COPY*

*nnnnnn* ist eine 6-stellige Ganzzahl, die die SESAM-Sicherungsbestände fortlaufend durchnummeriert. Bei einer Sicherung mit ARCHIVE oder HSMS wird diese Nummer nicht sichtbar.

**SESAM-Sicherungsbestände verwalten**

Informationen über SESAM-Sicherungsbestände von Anwender-Spaces enthält die Catalog-Tabelle RECOVERY\_UNITS (siehe „[Basishandbuch](#)“). Mit Hilfe des Utility-Monitors (siehe Handbuch „[Utility-Monitor](#)“) oder eines ESQL-Programms können Sie auf diese Informationen zugreifen. Einträge zu nicht mehr benötigten SESAM-Sicherungsbeständen können Sie mit der Utility-Anweisung MODIFY aus den RECOVERY\_UNITS löschen.

Informationen über SESAM-Sicherungsbestände von Datenbank und Catalog-Space sind in der CAT-REC-Datei vermerkt. Auch auf die Informationen in der CAT-REC-Datei können Sie mit dem Utility-Monitor zugreifen (Offline-Update). Die aktuelle CAT-REC-Datei können Sie auch mit der Utility-Anweisung MODIFY pflegen (Online-Update, siehe [Seite 177](#)).

Die Sicherungsversionen im HSMS-Archiv und im ARCHIVE-Directory müssen gesondert gepflegt werden.

## SESAM-Sicherungsbestände lesen

Auf einen SESAM-Sicherungsbestand einer gesamten SESAM/SQL-Datenbank kann der Anwender unter einem separaten DBH lesend zugreifen. Dies entlastet den laufenden Betrieb mit der Originaldatenbank. Das Lesen eines SESAM-Sicherungsbestandes ist von mehreren DBHs gleichzeitig möglich.

Der Anwender kann den SESAM-Sicherungsbestand mit der Administrationsanweisung `ADD-SQL-DB-CATALOG-ENTRY` in das SQL-Datenbankverzeichnis einfügen. Dazu muss er bei dem Operanden `COPY-NUMBER` die Nummer des SESAM-Sicherungsbestandes des Catalog-Space angeben.

Erlaubt sind nur lesende DML-Zugriffe, d.h. auch keine Utility-Anweisungen. Der SESAM-Sicherungsbestand, auf den lesend zugegriffen wird, steht unverändert für spätere Recovery-Maßnahmen zur Verfügung.

## Beispiele

1. Das Beispiel erstellt einen SESAM-Sicherungsbestand der Datenbank `AUFTRAGKUNDEN`, d.h., des Catalog-Space und der Anwender-Spaces. Von der Sicherung ausgenommen ist der Space `INDEXSPACE`, da auf ihm ausschließlich Indizes liegen und er nicht in der logischen Datensicherung ist.



```
COPY CATALOG auftragkunden USING STOGROUP stogroup3  
EXCEPT NO LOG INDEX SPACE
```

2. Das Beispiel erstellt einen SESAM-Sicherungsbestand des Catalog-Space der Datenbank `AUFTRAGKUNDEN`.



```
COPY CATALOG_SPACE auftragkunden USING STOGROUP stogroup3  
NO CHECK FORMAL
```

3. Das Beispiel erstellt einen SESAM-Sicherungsbestand des Anwender-Space `TABLESPACE` der Datenbank `AUFTRAGKUNDEN`.



```
COPY SPACE tablespace USING STOGROUP stogroup3
```

In allen Fällen legt SESAM/SQL die SESAM-Sicherungsbestände (Plattenkopien) auf der Storage Group `STOGROUP3` an. Die einzelnen SESAM-Sicherungsbestände werden offline erstellt. Auf formale Korrektheit werden die SESAM-Sicherungsbestände im 1. und im 3. Beispiel geprüft.

## Siehe auch

`RECOVER`, `CREATE REPLICATION`

## CREATE CATALOG - Datenbank (Catalog-Space) anlegen

CREATE CATALOG legt eine Datenbank an und erzeugt die zugehörige BS2000-Datei für den Catalog-Space.

Der Catalog-Space kann auf Pubsets mit „großen Dateien“ bis zu 4 TByte groß werden. Sonst kann er bis zu 64 GByte groß werden.

Der codierte Zeichensatz der Datenbank (Coded Character Set, CCS; synonym: Code-Tabelle) wird im Datenbankbetrieb verwendet.

CREATE CATALOG legt den Berechtigungsschlüssel für einen Benutzer mit umfassenden Befugnissen fest. Dieser Benutzer heißt universeller Benutzer.

Nach erfolgreicher Ausführung von CREATE CATALOG ist der Catalog-Space und damit die durch ihn repräsentierte Datenbank in das SQL-Datenbankverzeichnis der laufenden DBH-Session eingetragen.

Vor der Ausführung von CREATE CATALOG muss im SQL-Datenbankverzeichnis noch Platz für mindestens einen Eintrag vorhanden sein (siehe Handbuch „[Datenbankbetrieb](#)“). Außerdem darf im SQL-Datenbankverzeichnis der bei CREATE CATALOG angegebene Datenbankname noch nicht eingetragen sein.

Die Anweisung CREATE CATALOG darf nur auf bestimmten System-Benutzerkennungen eingegeben werden. Diese Benutzerkennungen legt der SESAM/SQL-Systemverwalter mit der DBH-Option ADMINISTRATOR bzw. mit der Administrationsanweisung MODIFY-ADMINISTRATION fest.

Soll eine Datenbank in einer DB-Kennung angelegt werden, muss der Datenbankverwalter Vorbereitungen für das Anlegen der Datenbankdateien treffen, siehe Abschnitt „Datenbankdateien und Jobvariablen auf fremden Benutzerkennungen“ im „[Basishandbuch](#)“.

Folgende Dateien sind betroffen:

- der Catalog-Space und bei Logging die CAT-REC-Datei; diese beiden Dateien müssen in derselben BS2000-Benutzerkennung vorhanden sein.
- die CAT-LOG-Dateien und DA-LOG-Dateien, wenn sie in dieser DB-Kennung stehen sollen. Diese Dateien werden sonst in der DBH-Kennung eingerichtet.
- die Anwender-Spaces zu dieser Datenbank spätestens vor der jeweiligen SQL-Anweisung CREATE SPACE.

---

```

CREATE CATALOG catalog [PASSWORD kennwort]
  [ON USER_ID bs2000_benutzerkennung]
  [CODE_TABLE ccs_name]
  [CATALOG_SPACE space_parameter_liste]
  [STOGROUP stogroup stogroup_attributes]
  [MEDIA STOGROUP stogroup stogroup_attributes]
  USER berechtigungsschlüssel [, systembenutzer_kennung]

```

$$space\_parameter\_liste ::= \left. \begin{array}{l} \text{PRIMARY } zahl \\ \text{SECONDARY } zahl \\ \text{PCTFREE } zahl \\ \text{[NO] SHARE} \\ \text{[NO] DESTROY} \\ \text{NO LOG} \end{array} \right\} \dots$$

$$stogroup\_attributes ::= \left\{ \begin{array}{l} \text{VOLUMES}(\text{volumename } , \dots) \text{ ON } gerätetyp \\ \text{PUBLIC} \end{array} \right\} \text{ [ON } catid \text{]}$$

$$systembenutzer\_kennung ::= \left\{ \begin{array}{l} utm\_benutzer \\ bs2000\_benutzer \end{array} \right\}$$

$$utm\_benutzer ::= \left( \left\{ \begin{array}{l} rechnername \\ * \end{array} \right\}, \left\{ \begin{array}{l} utm\_anwendungsname \\ * \end{array} \right\}, \left\{ \begin{array}{l} utm\_benutzerkennung \\ * \end{array} \right\} \right)$$

$$bs2000\_benutzer ::= \left( \left\{ \begin{array}{l} rechnername \\ * \end{array} \right\}, \text{[*]}, \left\{ \begin{array}{l} bs2000\_benutzerkennung \\ * \end{array} \right\} \right)$$


---

*catalog*

Name der neu zu erstellenden SESAM/SQL-Datenbank. Ihr Catalog-Space wird mit CREATE CATALOG eingerichtet.

Für *catalog* darf noch kein Eintrag im SQL-Datenbankverzeichnis vorhanden sein. Außerdem muss im SQL-Datenbankverzeichnis noch Platz für mindestens einen Eintrag sein. SESAM/SQL trägt *catalog* als logischen und physikalischen Namen der neuen Datenbank in das SQL-Datenbankverzeichnis ein (siehe Handbuch „[Datenbankbetrieb](#)“).

SESAM/SQL verwendet eine bereits vorhandene BS2000-Datei für den Catalog-Space nur dann, wenn sie leer ist.

Die Initialisierung des Catalog-Space mit CREATE CATALOG ist jedoch auch in diesem Fall erforderlich.

PASSWORD *kennwort*

BS2000-Kennwort für die Datei des Catalog-Space. *kennwort* muss als alphanumerisches Literal angegeben werden.

Für die Angabe von *kennwort* gibt es folgende Möglichkeiten:

- 'C'*zeichenkette*"  
*zeichenkette* enthält vier abdruckbare Zeichen.
- 'X'*hex-zeichenkette*"  
*hex-zeichenkette* enthält acht sedezimale Zeichen.
- 'n'  
*n* bezeichnet eine ganze Zahl von - 2147483648 bis + 2147483647.

*kennwort* gilt für die gesamte Datenbank, d.h. außer für den Catalog-Space auch für alle Anwender-Spaces, die später mit CREATE SPACE (siehe Handbuch „[SQL-Sprachbeschreibung Teil 1: SQL-Anweisungen](#)“) angelegt werden, für alle automatisch erzeugten Anwender-Spaces, für die CAT-REC-Datei, die CAT-LOG-Dateien, die DA-LOG-Dateien und die PBI-Datei sowie für SESAM-Sicherungsbestände.



Wenn die Datenbank auf einer DB-Kennung liegen soll, gilt:

- bei Miteigentümerschaft vergibt SESAM/SQL das Kennwort für die Dateien
- werden die Dateien vom Datenbankverwalter selbst eingerichtet, muss auch das Kennwort beim Einrichten vergeben werden

Siehe Abschnitt „Datenbankdateien und Jobvariablen auf fremden Benutzerkennungen“ im „[Basishandbuch](#)“.

PASSWORD *kennwort* nicht angeben:

Der Catalog-Space wird ohne Kennwort angelegt.

**ON USER\_ID** *bs2000\_benutzerkennung*

Name der BS2000-Benutzerkennung (DB-Kennung), in der die Datenbank angelegt werden soll. Für Catalog-Space und CAT-REC-Datei bei Logging müssen Vorbereitungen getroffen worden sein, siehe [Seite 105](#).

**ON USER\_ID** *bs2000\_benutzerkennung* nicht angeben:  
Der Catalog-Space wird in der DBH-Kennung angelegt.

**CODE\_TABLE** *ccs\_name*

Maximal 8 Zeichen langer Name für einen im BS2000 (Systemkomponente XHCS) definierten codierten Zeichensatz (Coded Character Set, CCS).

*ccs\_name* muss als alphanumerisches Literal angegeben werden, bestehend aus Grossbuchstaben, Ziffern, @, # oder \$. Das erste Zeichen darf keine Ziffer sein. Für *ccs\_name* dürfen die CCS-Namen von EBCDIC-Zeichensätzen oder selbst definierten Zeichensätzen auf Basis von EBCDIC-Zeichensätzen angegeben werden. Für *ccs\_name* darf auch der Wert `_NONE_` angegeben werden. Dann wird für die Datenbank kein codierter Zeichensatz verwendet. Dies entspricht dem Verhalten von SESAM/SQL vor V5.0.

Der *ccs\_name* (siehe auch: ALTER CATALOG) wird in den Metadaten der Datenbank eingetragen und verwendet, z.B. bei der Utility-Anweisung LOAD zur Prüfung gegen den CCSN der Eingabedatei.

Einführende Informationen zur Anwendung codierter Zeichensätze in SESAM/SQL finden Sie im Abschnitt „Unicode-Konzept in SESAM/SQL“ im „[Basishandbuch](#)“.

**CODE\_TABLE** *ccs\_name* nicht angeben:

In SESAM/SQL wird der CCSN `_NONE_` in den Metadaten der Datenbank eingetragen, siehe oben.

**CATALOG\_SPACE** *space\_parameter\_liste*

Jede der folgenden Space-Klauseln PRIMARY, SECONDARY, PCTFREE, [NO] SHARE, [NO] DESTROY, NO LOG dürfen Sie nur einmal angeben.

**PRIMARY** *zahl*

Primärzuweisung der BS2000-Datei für den Catalog-Space in 2K-Einheiten (BS2000-Halpages). *zahl* muss eine Ganzzahl von 1 bis 33 554 430 sein.

In jedem Fall legt SESAM/SQL den Catalog-Space mindestens in der Größe an, die für die Catalog-Tabellen zur Aufnahme der Metadaten der Datenbank benötigt wird. Eine zu kleine Primärzuweisung wird somit von SESAM/SQL ignoriert.

**PRIMARY** *zahl* nicht angeben:

SESAM/SQL legt den Catalog-Space mindestens in der Größe an, die für die Catalog-Tabellen zur Aufnahme der Metadaten der Datenbank benötigt wird.

**SECONDARY** *zahl*

Sekundärzuweisung der BS2000-Datei für den Catalog-Space in 2K-Einheiten (BS2000-Halfpages). *zahl* muss eine Ganzzahl von 0 bis 32767 sein.

Falls für *zahl* ein Wert kleiner als 24 angegeben wird, verwendet SESAM/SQL automatisch SECONDARY 24.

SECONDARY *zahl* nicht angegeben:

Es gilt SECONDARY 24.

**PCTFREE** *zahl*

Freiplatzreservierung der BS2000-Datei für den Catalog-Space in Prozent.

*zahl* muss eine Ganzzahl zwischen 0 und 70 sein.

SESAM/SQL hinterlegt die Angabe zur Freiplatzreservierung in den Metadaten der Datenbank. Die Angabe zur Freiplatzreservierung wird ausgewertet bei der Reorganisation des Catalog-Space mit der Utility-Anweisung REORG.

PCTFREE *zahl* nicht angegeben:

Es gilt PCTFREE 20.

**[NO] SHARE**

SHARE gibt an, dass die Datei für den Catalog-Space gemeinsam benutzbar ist.

NO SHARE gibt an, dass die Datei für den Catalog-Space nicht gemeinsam benutzbar ist. Bei einer bereits bestehenden Datei bleibt die vorhandene Einstellung erhalten.

**[NO] DESTROY**

DESTROY gibt an, dass beim Löschen der Datei für den Catalog-Space der Speicherplatz mit binär Null überschrieben wird.

NO DESTROY gibt an, dass beim Löschen der Datei für den Catalog-Space nur der Speicherplatz freigegeben wird.

Bei einer bereits bestehenden Datei bleibt die vorhandene Einstellung erhalten.

[NO] DESTROY nicht angegeben:

Bei einer bereits bestehenden Datei bleibt die vorhandene Einstellung erhalten.

**NO LOG**

Die gesamte SESAM/SQL-Datenbank wird ohne Logging betrieben.

Dies gilt für das CAT-Logging auf dem Catalog-Space ebenso wie für das DA-Logging auf allen mit der SQL-Anweisung CREATE SPACE neu zu erstellenden Anwender-Spaces. Es gilt auch auf allen automatisch erzeugten Anwender-Spaces mit den Namen *D0berechtigungsschlüssel*. *berechtigungsschlüssel* steht hier für den Berechtigungsschlüssel mit der entsprechenden CREATE SPACE-Berechtigung.

SESAM/SQL legt keine CAT-LOG-Datei an. Die CAT-REC-Datei erzeugt SESAM/SQL beim Erstellen des ersten SESAM-Sicherungsbestandes für die Datenbank bzw. den Catalog-Space mit der Utility-Anweisung COPY.

SESAM/SQL verwendet dabei die Angaben des CAT-REC-Mediensatzes in der Medientabelle (siehe Klausel MEDIA STOGROUP *stogroup*).

NO LOG nicht angeben:

Die gesamte SESAM/SQL-Datenbank wird mit Logging betrieben.

Für den Catalog-Space erfolgt CAT-Logging. Für mit CREATE SPACE neu einzurichtende Anwender-Spaces und automatisch erzeugte Anwender-Spaces gilt die Voreinstellung: mit DA-Logging.

Wenn die Datenbank auf der DBH-Kennung liegt, richtet SESAM/SQL die CAT-REC-Datei und die erste CAT-LOG-Datei gemäß den Einträgen in der Medientabelle ein (siehe Klausel MEDIA STOGROUP *stogroup*).

Wenn die Datenbank auf einer DB-Kennung liegt, müssen für die CAT-REC-Datei Vorbereitungen getroffen worden sein, siehe [Seite 105](#).

Das Logging kann nachträglich nicht für die gesamte Datenbank ausgeschaltet werden. Es kann aber für einzelne Anwender-Spaces ausgeschaltet werden.

CATALOG\_SPACE *space\_parameter\_liste* nicht angeben:

SESAM/SQL verwendet für die einzelnen Space-Klauseln die oben beschriebenen Standardwerte.

### STOGROUP *stogroup*

Name für eine Storage Group. STOGROUP *stogroup* definiert die Storage Group für den Catalog-Space. Der Name darf nicht D0STOGROUP sein. Der einfache Name der Storage Group kann durch den Datenbanknamen *catalog* qualifiziert werden. Ein in *stogroup* enthaltener Datenbankname muss mit dem Datenbanknamen *catalog* übereinstimmen. Den Namen der Storage Group trägt SESAM/SQL in die Metadaten der Datenbank ein.

Bei STOGROUP und MEDIA STOGROUP müssen unterschiedliche Namen für *stogroup* verwendet werden.

Der bei USER angegebene Berechtigungsschlüssel, d.h. der universelle Benutzer, wird Eigentümer der Storage Group und erhält das Sonder-Privileg USAGE für diese Storage Group.

Neben *stogroup* trägt SESAM/SQL eine weitere Storage Group, die sogenannte Default Storage Group D0STOGROUP, in die Metadaten der Datenbank ein. Gemäß Voreinstellung umfasst die Default Storage Group gemeinschaftlich benutzbare Platten (Public-Platten) auf dem Default-Pubset der Kennung, in der der DBH abläuft. Mit ALTER STOGROUP kann der Datenbankverwalter diese Einstellung jederzeit ändern.

Eigentümer der Default Storage Group ist der bei USER angegebene Berechtigungsschlüssel.

PUBLIC hat die Berechtigung zur Nutzung der Default Storage Group. Die Klausel STOGROUP ist für den Catalog-Space wirkungslos, wenn der Datenbankverwalter den Catalog-Space bereits mit dem Kommando CREATE-FILE eingerichtet hat. Die Storage Group wird aber in die Metadaten der Datenbank eingetragen.

STOGROUP *stogroup* nicht angeben:

SESAM/SQL legt den Catalog-Space auf der Default Storage Group an.

Mindestens eine der Klauseln STOGROUP und MEDIA STOGROUP sollte angegeben werden. Andernfalls legt SESAM/SQL sowohl den Catalog-Space als auch die CAT-REC- sowie die erste CAT-LOG-Datei auf der Default Storage Group an. Dies wäre ungünstig im Hinblick auf Maßnahmen des Media-Recovery.

MEDIA STOGROUP *stogroup*

Name für eine Storage Group. MEDIA STOGROUP *stogroup* definiert die Storage Group für die CAT-REC-Datei. Sofern Logging für die Datenbank vereinbart ist, gilt diese Storage Group auch für die erste CAT-LOG-Datei. Der Name darf nicht D0STOGROUP sein. Der einfache Name der Storage Group kann durch den Datenbanknamen *catalog* qualifiziert werden. Ein in *stogroup* enthaltener Datenbankname muss mit dem Datenbanknamen *catalog* übereinstimmen. Aufgrund der Angaben bei MEDIA STOGROUP trägt SESAM/SQL in die Medientabelle den jeweils ersten Mediensatz für die Dateiarart CAT-REC und ggf. auch für die Dateiarart CAT-LOG ein. SESAM/SQL legt die CAT-REC-Datei und ggf. die erste CAT-LOG-Datei entsprechend diesen Angaben an. Für die Dateieigenschaften (Primärzuweisung, Sekundärzuweisung etc.) gelten Standardwerte (siehe „[Basishandbuch](#)“).

Bei STOGROUP und MEDIA STOGROUP müssen unterschiedliche Namen für *stogroup* verwendet werden.

Der bei USER angegebene Berechtigungsschlüssel, d.h. der universelle Benutzer, wird Eigentümer der Storage Group und erhält das Sonder-Privileg USAGE für diese Storage Group.

MEDIA STOGROUP *stogroup* nicht angeben:

SESAM/SQL legt die CAT-REC-Datei und ggf. die erste CAT-LOG-Datei auf der Default Storage Group D0STOGROUP an. In diesem Fall enthalten die entsprechenden Einträge in der Medientabelle die Angaben D0STOGROUP sowie Standardwerte für die Eigenschaften von CAT-LOG und CAT-REC-Datei (siehe „[Basishandbuch](#)“).

Die Berechtigung zur Nutzung der Default Storage Group hat PUBLIC.

Mindestens eine der Klauseln STOGROUP und MEDIA STOGROUP sollte angegeben werden, da SESAM/SQL andernfalls sowohl den Catalog-Space als auch die CAT-REC- sowie die erste CAT-LOG-Datei auf der Default Storage Group anlegt. Es ist ungünstig im Hinblick auf Recovery-Maßnahmen, wenn alle Dateien auf derselben Storage Group liegen.

*stogroup\_attributes ::=*

Jedes der folgenden Attribute dürfen Sie nur einmal angeben.

**VOLUMES** (*volumename* ,...)

Datenträgerkennzeichen von Privatplatten. *volumename* muss als alphanumerisches Literal angegeben werden. Jedes Datenträgerkennzeichen darf nur einmal angegeben werden. Es dürfen maximal 100 Platten angegeben werden.

**ON** *gerätetyp*

Gerätetyp der Privatplatten. *gerätetyp* muss als alphanumerisches Literal angegeben werden. Die Platten einer Storage Group müssen vom gleichen Gerätetyp sein.

Die VOLUMES-Angabe wird nicht genutzt, wenn die dazugehörigen Dateien schon vor dem CREATE CATALOG vorhanden sind. Die Angaben werden aber in den Metadaten der Datenbank abgelegt und wirken sich bei weiteren Anweisungen (REORG, COPY) aus.

**PUBLIC**

Die Storage Group umfasst alle gemeinschaftlichen Platten (Pubset), deren Kennzeichen die bei **ON** *catid* angegebene *catid* ist.

**ON** *catid*

Katalogkennung des Pubsets, in dem die Dateien katalogisiert werden sollen. *catid* muss als alphanumerisches Literal angegeben werden.

Bei Privatplatten (d.h. bei Angabe von VOLUMES ... in der STOGROUP-Klausel), sind die Dateien in diesem Pubset katalogisiert, aber auf Privatplatte angelegt. Bei der Angabe von PUBLIC werden die Dateien auch auf diesem Pubset angelegt.

**ON** *catid* nicht angeben:

Es wird das Standard-Pubset verwendet, das der BS2000-Benutzerkennung zugewiesen ist, in der der DBH abläuft.

**USER** *berechtigungsschlüssel* [, *systembenutzer\_kennung*]

Definition des universellen Benutzers.

*berechtigungsschlüssel*

gibt den Berechtigungsschlüssel für den universellen Benutzer an.

*systembenutzer\_kennung*

gibt die System-Benutzerkennung (BS2000-Benutzer, UTM-Benutzer) für den universellen Benutzer an. Fehlt diese Angabe, so verwendet SESAM/SQL die System-Benutzerkennung des Benutzers, der CREATE CATALOG eingegeben hat.

Solange der universelle Benutzer nicht mit den SQL-Anweisungen CREATE USER und CREATE SYSTEM\_USER weitere SQL-Benutzer eingerichtet und diesen SQL-Benutzern mit der SQL-Anweisung GRANT entsprechende Privilegien erteilt hat, ist nur der universelle Benutzer berechtigt, für die neue Datenbank SQL- und Utility-Anweisungen abzusetzen (siehe „[Basishandbuch](#)“ und Handbuch „[SQL-Sprachbeschreibung Teil 1: SQL-Anweisungen](#)“).

*utm\_benutzer ::=*

Angabe des UTM-Benutzers.

*rechnername*

Symbolischer Rechnername. *rechnername* muss als alphanumerisches Literal angegeben werden.

Falls DCAM auf dem Rechner nicht verfügbar ist, ist dem Rechner der symbolische Name 'HOMEPROC' fest zugeordnet.

Bei UTM-D: Angabe des lokalen Rechners, auf dem der SESAM/SQL-Datenbankanschluss generiert wurde.

\* Alle Rechner

*utm\_anwendungsname*

Name der UTM-Anwendung. *utm\_anwendungsname* muss als alphanumerisches Literal angegeben werden.

Bei UTM-D: Name der lokalen UTM-Anwendung

\* Alle UTM-Anwendungen

*utm\_benutzerkennung*

Bei lokalen UTM-Systembenutzern geben Sie die UTM-Benutzerkennung an, die mit KDCSIGN definiert wurde. Bei UTM-D geben Sie den Namen der lokalen UTM-Session (LSES) an. *utm\_benutzerkennung* muss als alphanumerisches Literal angegeben werden.

\* Alle UTM-Benutzerkennungen

*bs2000\_benutzer ::=*

Angabe des BS2000-Benutzers.

*rechnername*

Symbolischer Rechnername. *rechnername* muss als alphanumerisches Literal angegeben werden. Falls DCAM auf dem Rechner nicht verfügbar ist, ist dem Rechner der symbolische Name 'HOMEPROC' fest zugeordnet.

\* Alle Rechner

*bs2000\_benutzerkennung*

BS2000-Benutzerkennung. *bs2000\_benutzerkennung* muss als alphanumerisches Literal angegeben werden.

\* Alle BS2000-Benutzerkennungen

## BS2000-Dateinamen für den Catalog-Space und die CAT-REC-Datei

Die BS2000-Datei für den Catalog-Space und bei Logging die CAT-REC-Datei wird entweder unter der DBH-Kennung oder unter einer DB-Kennung angelegt.

Catalog-Space:            :*catid:bs2000\_benutzerkennung.catalog.CATALOG*

CAT-REC-Datei:         :*catid:bs2000\_benutzerkennung.catalog.CAT-REC*

## Beispiele

1. Das Beispiel legt den Catalog-Space der Datenbank AUFTRAGKUNDEN an.

```
CREATE CATALOG auftragkunden PASSWORD 'C''PW#1''
CODE_TABLE 'EDF041' CATALOG_SPACE SHARE NO DESTROY
MEDIA STOGROUP auftragkunden.stogroup1 PUBLIC
USER utiuniv
```

Für den Catalog-Space bzw. die Datenbank wird das Kennwort 'PW#1' vergeben. Als codierter Zeichensatz der Datenbank wird EDF041 gewählt. Als BS2000-Datei soll der Catalog-Space mehrfach benutzbar sein und beim Löschen nicht mit binär Null überschrieben werden. Ansonsten gelten die von SESAM/SQL vergebenen Standardwerte. SESAM/SQL legt den Catalog-Space auf der Default Storage Group D0STOGROUP in der Kennung an, in der der DBH abläuft. Die CAT-REC-Datei und die erste CAT-LOG-Datei legt SESAM/SQL auf der Storage Group STOGROUP1 an.

Als Berechtigungsschlüssel für den universellen Benutzer wird UTIUNIV vereinbart. Als System-Benutzerkennung für den universellen Benutzer wird die System-Benutzerkennung des Benutzers übernommen, der CREATE CATALOG eingegeben hat.

2. Das Beispiel legt den Catalog-Space der Datenbank AUFTRAGKUNDEN an.



```
CREATE CATALOG auftragkunden
CATALOG_SPACE SHARE DESTROY
STOGROUP stogroup1 PUBLIC
MEDIA STOGROUP stogroup2 PUBLIC
USER utiuniv, (*,*,*)
```

Als BS2000-Datei soll der Catalog-Space mehrfach benutzbar sein und beim Löschen mit binär Null überschrieben werden. Ansonsten gelten die von SESAM/SQL vergebenen Standardwerte.

SESAM/SQL legt den Catalog-Space auf der Storage Group STOGROUP1 in der Kennung an, in der der DBH abläuft. Die CAT-REC-Datei und die erste CAT-LOG-Datei legt SESAM/SQL auf der Storage Group STOGROUP2 an.

Als Berechtigungsschlüssel für den universellen Benutzer wird UTIUNIV vereinbart. Als System-Benutzerkennung für den universellen Benutzer werden alle Rechner, alle UTM-Benutzerkennungen und alle BS2000-Benutzerkennungen zugelassen.

## Siehe auch

ALTER CATALOG

## CREATE MEDIA DESCRIPTION - Dateieigenschaften der datenbank-spezifischen Dateien festlegen

Mit CREATE MEDIA DESCRIPTION können Sie die Dateieigenschaften für die Dateien einer bestimmten datenbank-spezifischen oder space-spezifischen Dateiart festlegen.

CREATE MEDIA DESCRIPTION nimmt den ersten Eintrag (Mediensatz) für die betreffende datenbank-spezifische Dateiart in die Medientabelle auf. Die Medientabelle ist im „[Basis](#)handbuch“ beschrieben.

SESAM/SQL kennt folgende datenbank-spezifischen Dateiarten:

- CAT-REC-Datei
- CAT-LOG-Dateien
- DA-LOG-Dateien
- PBI-Dateien  
(Mediensätze gelten auch für die Arbeitsdatei bei Online-Sicherung mit HSMS)

SESAM/SQL kennt folgende space-spezifische Dateiart:

- DDL-TA-LOG-Dateien

Der aktuelle Berechtigungsschlüssel muss das Sonder-Privileg UTILITY besitzen.

---

```

CREATE MEDIA DESCRIPTION FOR {
  DALOG
  CATLOG
  CATREC
  PBI
  DDLTA
} AT CATALOG catalog

[ {
  PRIMARY zahl
  SECONDARY zahl
  [NO] SHARE
} ] ... [[NO] DEVICE REQUEST] [STOGROUP stogroup] 1

```

---

<sup>1</sup> TAPE *gerätetyp* kann in SESAM/SQL ab V7.0 nicht mehr angegeben werden

FOR {  
 DALOG  
 CATLOG  
 CATREC  
 PBI  
 DDLTA }

Gibt die Dateiart an, für die der erste Mediensatz aufgenommen werden soll. Falls für die betreffende Dateiart bereits ein Mediensatz in der Medientabelle existiert, wird CREATE MEDIA DESCRIPTION abgewiesen. Dabei ist zu beachten, dass bereits bei Ausführung von CREATE CATALOG der erste Mediensatz für die CAT-REC-Datei aufgenommen wurde. Sofern Logging für die Datenbank vereinbart ist, wurde auch der erste Mediensatz für CAT-LOG-Dateien aufgenommen.

- DALOG Es wird der erste Mediensatz für die Dateiart DA-LOG-Datei aufgenommen.
- CATLOG Es wird der erste Mediensatz für die Dateiart CAT-LOG-Datei aufgenommen.
- CATREC Es wird der erste Mediensatz für die Dateiart CAT-REC-Datei aufgenommen.
- PBI Es wird der erste Mediensatz für die Dateiart PBI-Datei aufgenommen (gilt auch für die Arbeitsdatei bei Online-Sicherung mit HSMS).
- DDLTA Es wird der erste Mediensatz für die Dateiart DDL-TA-LOG-Datei aufgenommen.

AT CATALOG *catalog*

Name der Datenbank, zu der die bearbeitete Medientabelle gehört.

Jede der folgenden Space-Klauseln PRIMARY, SECONDARY, [NO] SHARE dürfen Sie nur einmal angeben:

PRIMARY *zahl*

Primärzuweisung für Dateien der angegebenen Dateiart in 2 K-Einheiten (BS2000-Halfpages)

*zahl* muss eine Ganzzahl von 1 bis 33 554 430 sein.

Eine zu kleine Primärzuweisung wird von SESAM/SQL automatisch auf den Standardwert korrigiert.

PRIMARY *zahl* nicht angegeben: SESAM/SQL setzt folgende Standardwerte ein:

- DA-LOG-Dateien, CAT-LOG-Dateien: 768
- CAT-REC-Datei: 12
- PBI-Dateien: 576
- DDL-TA-LOG-Dateien: 1536

**SECONDARY** *zahl*

Sekundärzuweisung für Dateien der angegebenen Dateart in 2K-Einheiten (BS2000-Halfpages), falls als Dateart CATREC oder PBI angegeben ist. Falls als Dateart DALOG oder CATLOG angegeben ist, darf die Space-Klausel SECONDARY nicht angegeben werden.

*zahl* muss eine Ganzzahl von 0 bis 32767 sein.

Eine zu kleine Sekundärzuweisung wird von SESAM/SQL automatisch auf den Standardwert korrigiert.

SECONDARY *zahl* nicht angegeben: SESAM/SQL setzt folgende Standardwerte ein:

- DA-LOG-Datei, CAT-LOG-Datei: 0
- CAT-REC-Datei: 12
- PBI-Dateien: 72
- DDL-TA-LOG-Dateien: 384

**[NO] SHARE**

SHARE legt fest, dass die Dateien der angegebenen Dateart gemeinsam benutzbar sind.

NO SHARE legt fest, dass die Dateien der angegebenen Dateart nicht gemeinsam benutzbar sind.

**[NO] DEVICE REQUEST**

DEVICE REQUEST legt fest, dass SESAM/SQL bei abgearbeiteter Medientabelle weitere Geräte für die angegebene Dateart am Bedienplatz anfordert.

Wenn SESAM/SQL dann bei abgearbeiteter Medientabelle weitere Geräte anfordert, sind folgende Eingaben möglich: <sup>1</sup>

**PUBLIC** *catid*

Die Datei wird auf dem angegebenen Pubset angelegt.

*catid* BS2000-Katalogkennung.

Die BS2000-Katalogkennung muss angegeben werden.

*catid* muss als alphanumerisches Literal angegeben werden.

Wurde eine CATID-Liste definiert, muss die Katalogkennung in der Liste angegeben worden sein.

**IGNORE**

Die Datei wird nicht angelegt.

---

<sup>1</sup> TAPE *gerätetyp* kann in SESAM/SQL ab V7.0 nicht mehr angegeben werden

PRIVATE *catid,gerätetyp, vsn, vsn...*

Die Datei wird auf Privatplatten mit den angegebenen Datenträgerkennzeichen und dem angegebenen Gerätetyp angelegt und mit der angegebenen Katalogkennung katalogisiert.

Es können maximal 6 Datenträgerkennzeichen angegeben werden.

*catid* BS2000-Katalogkennung.

Die BS2000-Katalogkennung muss angegeben werden.

*catid* muss als alphanumerisches Literal angegeben werden. Wurde eine CATID-Liste definiert, muss die Katalogkennung in der Liste angegeben worden sein.

*gerätetyp* Gerätetyp der Privatplatte.

*gerätetyp* muss als alphanumerisches Literal angegeben werden.

*vsn* Datenträgerkennzeichen.

*vsn* muss als alphanumerisches Literal angegeben werden.

NO DEVICE REQUEST legt fest, dass SESAM/SQL bei abgearbeiteter Medientabelle keine weiteren Geräte für die angegebene Dateart am Bedienplatz anfordert.

STOGROUP *stogroup*

Name für die Storage Group, auf deren Platten die Dateien der angegeben Dateart angelegt werden sollen. Der einfache Name der Storage Group kann durch den Datenbanknamen *catalog* qualifiziert werden.

STOGROUP *stogroup* nicht angegeben:

SESAM/SQL legt die Dateien der angegebenen Dateart auf der Default Storage Group D0STOGROUP an.

## Beispiel

Das Beispiel nimmt den ersten Mediensatz für DA-LOG-Dateien in die Medientabelle der Datenbank AUFTRAGKUNDEN auf.



```
CREATE MEDIA DESCRIPTION FOR DALOG AT CATALOG auftragkunden
SHARE NO DEVICE REQUEST STOGROUP stogroup1
```

Diese Anweisung legt Speichermedium und Dateieigenschaften für die DA-LOG-Dateien der Datenbank AUFTRAGKUNDEN fest. Demnach wird SESAM/SQL die DA-LOG-Dateien der Datenbank AUFTRAGKUNDEN auf der Storage Group STOGROUP1 mit dem Standardwert für die Primärzuweisung von 768 2K-Einheiten und der Eigenschaft „gemeinsam benutzbar“ anlegen. Bei abgearbeiteter Medientabelle wird SESAM/SQL keine weiteren Geräte für DA-LOG-Dateien am Bedienplatz anfordern.

## Siehe auch

ALTER MEDIA DESCRIPTION, DROP MEDIA DESCRIPTION

## CREATE REPLICATION - Replikat erzeugen

Mit CREATE REPLICATION können Sie

- aus einem SESAM-Sicherungsbestand, der mit COPY CATALOG erzeugt wurde, ein Replikat generieren. Dabei kann es sich um eine Magnetbandkassetten- oder Plattensicherung handeln.
- aus einer Fremdkopie, die mit beliebigen Mitteln aus einer konsistenten Datenbank erzeugt wurde, ein Replikat generieren.

Dieses Replikat ist dann ein Abbild des Originals zum Zeitpunkt der Erstellung des SESAM-Sicherungsbestandes oder der Fremdkopie. Ebenso können Sie auch ein Teilreplikat erzeugen, wenn Sie einen Teil der Anwender-Spaces auswählen mit FOR SPACE.

Ein Replikat kann nur aus SESAM-Sicherungsbeständen oder aus Fremdkopien erzeugt werden, die in SESAM/SQL V9.0 entstanden sind.

Erläuterungen zum Generieren einer Fremdkopie finden Sie im „[Basishandbuch](#)“. Ein Space darf sich beim Erzeugen der Fremdkopie nicht im Zustand „check pending“ oder „recover pending“ befinden, wenn er später in das Replikat aufgenommen werden soll.

Die Dateinamen des Replikats sind genauso aufgebaut wie die einer Datenbank mit Ausnahme der CAT-REC-Datei. Deren Name ist *replikat.CAT-REC.REPL*.

Replikate und Teilreplikate können zum Lesen verwendet werden. Wenn ein Replikat eine partitionierte Tabelle enthält, dann müssen alle Partitionen der Tabelle im Replikat enthalten sein. Ein Teilreplikat muss immer in sich abgeschlossen sein, d.h. das Teilreplikat muss alle Indizes zu allen Tabellen beinhalten, die zum Teilreplikat gehören.

Replikate und Teilreplikate können mit der Utility-Anweisung REFRESH REPLICATION aktualisiert werden. Das ist die einzige Form von Änderungen, die für ein Replikat zulässig ist. Das Original, aus dem das Replikat entstanden ist, kann mit der Utility-Anweisung RECOVER USING/TO REPLICATION auf Basis des Replikats repariert oder zurückgesetzt werden. Mit denselben Anweisungen kann auch ein neues Original erzeugt werden. Ein späteres Aktualisieren mit REFRESH REPLICATION ist nur möglich, wenn für alle Spaces des Replikats Logging eingeschaltet ist. Ein Replikat ohne Logging kann aber erzeugt werden.

Replikate können auch aus Sicherungen mit defekten Indizes erzeugt werden. Die Nutzung eines Replikats mit defekten Indizes ist auf zwei Weisen eingeschränkt:

- Auf dem Replikat ist kein RECOVER INDEX möglich.
- Nach einem RECOVER INDEX im Original müssen betroffene Spaces mit REFRESH REPLICATION FOR SPACE zuerst aus dem Replikat entfernt werden. Mit REFRESH SPACE können sie anschließend wieder dem Replikat hinzugefügt werden (siehe Abschnitt „Replikat einer Datenbank“ im „[Basishandbuch](#)“).

Sie können ein Replikat nur auf Platte anlegen.

Wenn das Softwareprodukt „Jobvariablen“ in BS2000 installiert ist, stellt SESAM/SQL nach dem CREATE REPLICATION zwei Einträge in einer Jobvariablen zur Verfügung. Sie zeigen, ab welcher CAT-LOG- und DA-LOG-Datei die Logging-Dateien beim nächsten REFRESH REPLICATION bereitgestellt werden müssen.

Die Jobvariable heißt *SESAM.replikat.NEXT-REPL-LOG*.

Liegt das Replikat in einer DB-Kennung, so wird versucht, die Jobvariable ebenfalls in der Kennung des Replikats anzulegen. Dazu müssen Vorbereitungen getroffen worden sein, siehe Abschnitt „Datenbankdateien und Jobvariablen auf fremden Benutzerkennungen“ im „[Basishandbuch](#)“. Kann die Jobvariable nicht in der Kennung des Replikats angelegt werden, so wird sie in der DBH-Kennung angelegt.

Die Anweisung CREATE REPLICATION darf nur auf bestimmten System-Benutzerkennungen eingegeben werden. Diese Benutzerkennungen legt der SESAM/SQL-Systemverwalter mit der DBH-Option ADMINISTRATOR bzw. mit der Administrationsanweisung MODIFY-ADMINISTRATION fest.

Für die Kopie, aus der das (Teil)replikat erzeugt wird, werden Leserechte benötigt.

Nach Ausführung des CREATE REPLICATION ist das Replikat im DBH-Datenbankverzeichnis eingetragen und zugreifbar.

Soll ein Replikat aus einem mit HSMS erzeugten Sicherungsbestand erstellt werden, müssen in folgenden Fällen Miteigentümerschaftsrechte definiert werden (siehe Abschnitt „Datenbankdateien und Jobvariablen auf fremden Benutzerkennungen“ im „[Basishandbuch](#)“):

- das Replikat soll in einer Benutzerkennung ungleich der DBH-Kennung liegen
- die Original-Datenbank liegt in einer DB-Kennung

Es kann sinnvoll sein, die Miteigentümerschaft teilqualifiziert zu vergeben. Auf diese Weise wird sichergestellt, dass für alle betroffenen Dateien eine Miteigentümerschaft definiert ist.

---

```

CREATE REPLICATION replik [PASSWORD kennwort]
  [ON USER_ID bs2000_benutzerkennung]
  [USING MEDIA stogroup_attributes]

FROM [FOREIGN] CATREC catrec_datei
  [PASSWORD kennwort]
  [ON USER_ID bs2000_benutzerkennung]
[FOR SPACE space,...]
[ { RENAME }
  { COPY } ]

stogroup_attributes ::= { VOLUMES (volumename,...) ON gerätetyp } [ON catid]

```

---

*replik*

Name, unter dem das Replikat angelegt werden soll.

Ist unter dem Namen *replik* schon ein nicht-leerer Catalog-Space angelegt, muss es sich um einen alten Catalog-Space eines Replikats handeln. Andernfalls wird CREATE REPLICATION abgebrochen.

Für den angegebenen Namen darf noch kein Eintrag im SQL-Datenbankverzeichnis existieren. Es muss noch mindestens ein Eintrag frei sein.

PASSWORD *kennwort*

BS2000-Kennwort für das Replikat. *kennwort* muss als alphanumerisches Literal angegeben werden.

Für die Angabe von *kennwort* gibt es folgende Möglichkeiten:

- '*C*'*zeichenkette*"  
*zeichenkette* enthält vier abdruckbare Zeichen.
- '*X*'*hex-zeichenkette*"  
*hex-zeichenkette* enthält acht sedezimale Zeichen.
- '*n*'  
*n* bezeichnet eine ganze Zahl von - 2147483648 bis + 2147483647.

*kennwort* gilt für das Replikat und für die CAT-REC-Datei des Replikats.



Wenn das Replikat in einer DB-Kennung liegen soll, gilt:

- bei Miteigentümerschaft vergibt SESAM/SQL das Kennwort für die Dateien
- werden die Dateien vom Datenbankverwalter selbst eingerichtet, muss auch das Kennwort beim Einrichten vergeben werden

Siehe Abschnitt „Datenbankdateien und Jobvariablen auf fremden Benutzerkennungen“ im „[Basishandbuch](#)“.

Ist RENAME angegeben, erhält das Replikat das Kennwort nach dem Umbenennen.

PASSWORD *kennwort* nicht angegeben:

Das Replikat wird ohne Kennwort angelegt. Das Kennwort einer Fremdkopie bleibt erhalten.

ON USER\_ID *bs2000\_benutzerkennung*

Name der BS2000-Benutzerkennung (DB-Kennung), in der das Replikat angelegt werden soll.

Wird das Replikat aus einer Plattenkopie erzeugt, muss der Datenbankverwalter Vorbereitungen für die Dateien des Replikats treffen, siehe Abschnitt „Datenbankdateien und Jobvariablen auf fremden Benutzerkennungen“ im „[Basishandbuch](#)“.

Wird das Replikat aus einem mit HSMS gesicherten Sicherungsbestand erzeugt, verwendet HSMS zum Einspielen der Dateien den Namen des Sicherungsbestands und hängt das Suffix „REPL“ an. In einem zweiten Schritt werden die Dateien von SESAM/SQL auf den Namen des Replikats umbenannt.

Auf der DB-Kennung muss der DBH-Kennung für alle in diesem Prozess verwendeten Dateinamen eine Miteigentümerschaft eingeräumt werden.

Es kann sinnvoll sein, sowohl für den Sicherungsbestand als auch für das Replikat die Miteigentümerschaft teilqualifiziert zu vergeben. Auf diese Weise wird sichergestellt, dass für alle betroffenen Dateien eine Miteigentümerschaft definiert ist.

Die Benutzerkennung wird ohne „\$“ angegeben. Die CAT-REC-Datei heißt hier *replikat.CAT-REC.REPL*.

Wird *bs2000\_benutzerkennung* zusammen mit RENAME angegeben, so muss *bs2000\_benutzerkennung* dieselbe Benutzerkennung sein, die unter FROM CATREC ... ON USER\_ID spezifiziert ist. Andernfalls wird mit SQLSTATE abgebrochen.

ON USER\_ID *bs2000\_benutzerkennung* nicht angegeben:

Das Replikat wird in der DBH-Kennung angelegt.

**USING MEDIA *stogroup\_attributes***

Hier werden die Medien identifiziert, auf denen die Dateien des Replikats angelegt werden sollen.

**USING MEDIA *stogroup\_attributes*** darf nicht zusammen mit **RENAME** angegeben werden. Andernfalls wird mit **SQLSTATE** abgebrochen.

***stogroup\_attributes ::=***

Jedes der folgenden Attribute dürfen Sie nur einmal angeben.

**VOLUMES (*volumename*, ...)**

Datenträgerkennzeichen von Privatplatten. *volumename* muss als alphanumerisches Literal angegeben werden. Jedes Datenträgerkennzeichen darf nur einmal angegeben werden. Es dürfen maximal 100 Platten angegeben werden.

**ON *gerätetyp***

Gerätetyp der Privatplatten. *gerätetyp* muss als alphanumerisches Literal angegeben werden. Alle Platten müssen vom gleichen Gerätetyp sein.

Die **VOLUMES**-Angabe wird nicht genutzt, wenn die dazugehörigen Dateien schon vor dem **CREATE REPLICATION** vorhanden sind.

**PUBLIC**

Die Storage Group umfasst alle gemeinschaftlichen Platten (Pubset), deren Kennzeichen die bei **ON *catid*** angegebene *catid* ist.

**ON *catid***

Katalogkennung des Pubset, in dem die Dateien angelegt werden sollen.

*catid* muss als alphanumerisches Literal angegeben werden.

Bei Privatplatten (d.h. bei Angabe von **VOLUMES**), sind die Dateien in diesem Pubset katalogisiert, aber auf Privatplatte angelegt. Bei der Angabe von **PUBLIC** werden die Dateien auch auf diesem Pubset angelegt.

**ON *catid*** nicht angeben:

Es wird das Standard-Pubset verwendet, das der BS2000-Benutzerkennung zugewiesen ist, in der der DBH abläuft.

**USING MEDIA *stogroup\_attributes*** nicht angeben und keine Dateien angelegt:

Es wird das Standard-Pubset verwendet, das der BS2000-Benutzerkennung zugewiesen ist, in der der DBH abläuft.

**FROM [FOREIGN] CATREC *catrec\_datei***

Name einer CAT-REC-Datei oder einer CAT-REC-Kopie ohne BS2000-Benutzerkennung. Diese Datei enthält die Metadaten, mit denen das Replikat erstellt werden soll. Auf diese Datei müssen Sie lesend zugreifen können.

**FOREIGN**

Wenn FOREIGN angegeben ist, wird eine Fremdkopie einer Datenbank über ihre CAT-REC-Datei angegeben. Der Name hat das Format *catalog.CAT-REC*. Die Dateinamen der Fremdkopie lauten dann:

*catalog.CATALOG*  
*catalog.space*  
...

FOREIGN nicht angeben:

Der Name der *catrec\_datei* hat das Format *catalog.CAT-REC.COPY*. Der letzte Sicherungssatz in der Datei muss eine Sicherung der gesamten Datenbank beschreiben.

**PASSWORD *kennwort***

BS2000-Kennwort der CAT-REC-Datei und der Sicherungen oder Fremdkopien, die eingespielt werden. *kennwort* muss als alphanumerisches Literal angegeben werden.

Für die Angabe von *kennwort* gibt es folgende Möglichkeiten:

- 'C'*zeichenkette*"  
*zeichenkette* enthält vier abdruckbare Zeichen.
- 'X'*hex-zeichenkette*"  
*hex-zeichenkette* enthält acht sedezimale Zeichen.
- 'n'  
*n* bezeichnet eine ganze Zahl von - 2147483648 bis + 2147483647.

**ON USER\_ID *bs2000\_benutzerkennung***

Kennung, auf der die CAT-REC-Datei und die Kopien liegen. Die Benutzerkennung wird ohne „\$“ angegeben.

**FOR SPACE** *space*,...

Namen der Anwender-Spaces, die zu diesem Teilreplikат gehören sollen. Sie müssen alle zu der Datenbank gehören, von der der SESAM-Sicherungsbestand stammt. Maximal 999 Spacenames können angegeben werden. Kein Space darf mehrfach angegeben werden.

Der Catalog-Space und die angegebenen Anwender-Spaces sind dann im Teilreplikат enthalten. Der Catalog-Space darf nicht als Anwender-Space angegeben werden.

Bei der Auswahl der Anwender-Spaces ist darauf zu achten, dass das Replikат in sich abgeschlossen bleibt. Zu einem Table-Space kann der zugehörige Index-Space in das Replikат aufgenommen werden. Der Index-Space muss aber nicht aufgenommen werden, wenn er z.B. nicht der logischen Datensicherung unterliegt. Ein Index-Space sollte nicht ohne den zugehörigen Table-Space in das Replikат genommen werden.

**FOR SPACE** *space*,... nicht angegeben:

Alle Anwender-Spaces und der Catalog-Space der Sicherung gehören zum Replikат, sofern die verwendete Sicherung eine komplette Catalog-Sicherung war. Aus einer Sicherung mit COPY CATALOG EXCEPT NO LOG INDEX SPACE entsteht ein Teilreplikат, in dem die NO LOG Indespaces nicht enthalten sind.

**RENAME**

Nur bei FOREIGN CATREC erlaubt. Andernfalls wird mit SQLSTATE abgebrochen. RENAME verlangt das exklusive Schreibrecht. Die Fremdkopie muss in derselben Benutzerkennung liegen wie das Replikат. Die Fremdkopie darf in keinem DBH geöffnet sein. Es dürfen keine Dateien mit dem Namen des Replikats existieren. Das gilt nicht, wenn Fremdkopie- und Replikatsname gleich sind. Da die Dateien der Fremdkopie umbenannt werden, stehen nach dem RENAME die Dateien nicht mehr zur Verfügung. Dies gilt auch für den Fehlerfall. Es ist keine Wiederholung des CREATE REPLICATION mit dieser Funktion möglich.

**COPY**

Die Dateien werden kopiert.

**BS2000-Dateinamen für das Replikат**

Die BS2000-Datei für den Catalog-Space und die CAT-REC-Datei des Replikats wird entweder unter der DBH-Kennung oder unter einer DB-Kennung angelegt.

Im ersten Fall legt der DBH die Dateien an.

Im zweiten Fall müssen Vorbereitungen für das Anlegen der Dateien getroffen worden sein, siehe Abschnitt „Datenbankdateien und Jobvariablen auf fremden Benutzerkennungen“ im „[Basishandbuch](#)“. Bei einem Replikат aus einer Sicherung mit HSMS ist zu beachten, dass für alle betroffenen Dateien auf der DB-Kennung die DBH-Kennung als Miteigentümer definiert wird.

Die CAT-REC-Datei des Replikats wird beim CREATE REPLICATION angelegt und enthält alle Sätze, die auch das jeweilige Original enthalten hat.

Replikat: `:catid:bs2000_benutzererkennung_replikat.CATALOG`  
 Spacename `:catid:bs2000_benutzererkennung_replikat.einf_spacename`  
 CAT-REC-Datei des Replikats: `:catid:bs2000_benutzererkennung_replikat.CAT-REC.REPL`

Hilfsdateien für ein Replikat aus einer Sicherung mit HSMS:  
`:catid:bs2000_benutzererkennung_catalog.CATALOG.REPL`  
`:catid:bs2000_benutzererkennung_catalog.einf_spacename.REPL`

## SESAM-Sicherungsbestand lesen

Auf einen SESAM-Sicherungsbestand einer gesamten SESAM/SQL-Datenbank kann der Anwender unter einem separaten DBH lesend zugreifen. Dies entlastet den laufenden Betrieb mit der Originaldatenbank. Das Lesen eines SESAM-Sicherungsbestandes ist von mehreren DBHs gleichzeitig möglich.

Der Anwender kann den SESAM-Sicherungsbestand mit der Administrationsanweisung ADD-SQL-DB-CATALOG-ENTRY in das SQL-Datenbankverzeichnis einfügen. Dazu muss er bei dem Operanden COPY-NUMBER die Nummer des SESAM-Sicherungsbestandes des Catalog-Space angeben.

Erlaubt sind nur lesende DML-Zugriffe, d.h. auch keine Utility-Anweisungen. Der SESAM-Sicherungsbestand, auf den lesend zugegriffen wird, steht unverändert für spätere Recovery-Maßnahmen zur Verfügung.

## Beispiel

Das Beispiel erzeugt aus dem letzten SESAM-Sicherungsbestand der Datenbank AUFTRAGKUNDEN ein Teilreplikat. In diesem Teilreplikat sind der Catalog-Space und der Anwender-Space TABLESPACE enthalten.



```
CREATE REPLICATION auftragrep
FROM CATREC 'AUFTRAGKUNDEN.CAT-REC.COPY'
FOR SPACE tablespace
```

Das Replikat basiert auf dem Stand der Datenbank AUFTRAGKUNDEN, der in der aktuellen CAT-REC-Kopie der Originaldatenbank festgehalten ist.

## Siehe auch

REFRESH REPLICATION

## DROP MEDIA DESCRIPTION - Medientabellen-Einträge für eine datenbank-spezifische Dateart löschen

Mit DROP MEDIA DESCRIPTION können Sie alle Einträge zu einer bestimmten datenbank-spezifischen oder space-spezifischen Dateart aus der Medientabelle löschen. Die Medientabelle ist im „[Basishandbuch](#)“ beschrieben. Auf die zugehörigen Dateien hat DROP MEDIA DESCRIPTION keine Auswirkung, d.h. bestehende Dateien werden nicht gelöscht.

SESAM/SQL kennt folgende datenbank-spezifische Datearten:

- CAT-REC-Datei
- CAT-LOG-Dateien
- DA-LOG-Dateien
- PBI-Dateien  
(Mediensätze gelten auch für die Arbeitsdatei bei Online-Sicherung mit HSMS)

SESAM/SQL kennt folgende space-spezifische Dateart:

- DDL-TA-LOG-Dateien

Der aktuelle Berechtigungsschlüssel muss das Sonder-Privileg UTILITY besitzen.

```
DROP MEDIA DESCRIPTION FOR {
    DALOG
    CATLOG
    CATREC
    PBI
    DDLTA
} AT CATALOG catalog
```

```
FOR {
    DALOG
    CATLOG
    CATREC
    PBI
    DDLTA
}
```

Gibt die Dateart an, für die alle Einträge (Mediensätze) aus der Medientabelle entfernt werden sollen.

- DALOG Es werden alle Mediensätze für die Dateart DA-LOG-Datei gelöscht.
- CATLOG Es werden alle Mediensätze für die Dateart CAT-LOG-Datei gelöscht.
- CAT-REC Es werden alle Mediensätze für die Dateart CAT-REC-Datei gelöscht.
- PBI Es werden alle Mediensätze für die Dateart PBI-Datei gelöscht (gilt auch für die Arbeitsdatei bei Online-Sicherung mit HSMS).

DDLTA     Es werden alle Mediensätze für die Dateiart DDL-TA-LOG-Datei gelöscht.  
AT CATALOG *catalog*  
Name der Datenbank, zu der die bearbeitete Medientabelle gehört.

### Beispiel

Das Beispiel löscht alle Mediensätze für die Dateiart PBI-Datei der Datenbank AUFTRAGKUNDEN.



```
DROP MEDIA DESCRIPTION FOR PBI AT CATALOG auftragkunden
```

### Siehe auch

CREATE MEDIA DESCRIPTION, ALTER MEDIA DESCRIPTION

## EXPORT TABLE - Tabelle aus einer Datenbank in eine Exportdatei exportieren

Mit EXPORT TABLE können Sie eine Basistabelle einer SESAM/SQL-Datenbank in eine BS2000-Datei exportieren. Diese BS2000-Datei wird Export-Datei genannt. Die Export-Datei wird mit Coded Character Set Name (CCSN) angelegt.

In der Export-Datei werden Metadaten der Tabelle abgelegt. Wahlweise können Sie auch Anwenderdaten mit exportieren. Aus dieser Export-Datei kann mit IMPORT TABLE eine neue Tabelle generiert werden.

Mit Hilfe einer Export-Datei kann eine Basistabelle auf einfache Weise kopiert werden. Im Vergleich zu dem Anweisungspaar LOAD/UNLOAD sind weniger Arbeitsschritte nötig, da die Struktur der Tabelle in den Metadaten der Export-Datei abgelegt ist und in der Ziel-tabelle nicht neu eingerichtet werden muss (siehe „[Basishandbuch](#)“).

Das Anweisungspaar EXPORT/IMPORT TABLE bietet folgende Möglichkeiten:

- eine Basistabelle innerhalb einer Datenbank zu kopieren
- eine Basistabelle von einer Datenbank in eine andere zu kopieren
- eine Basistabelle von einem Space auf einen anderen zu verlagern
- eine Basistabelle in eine partitionierte Tabelle zu verlagern
- die Anzahl und die Grenzen der Partitionen einer partitionierten Tabelle zu modifizieren
- eine Basistabelle zu restrukturieren
- die Satznummern einer Basistabelle neu zu vergeben
- eine Basistabelle zu reorganisieren
- eine einzelne Basistabelle zu archivieren

EXPORT/IMPORT TABLE ist für beliebig große Basistabellen möglich. Es gibt keine Einschränkungen für die Länge und die Anzahl der enthaltenen Objekte, wie zum Beispiel Spalten oder Integritätsbedingungen.

In der Export-Datei werden folgende Metadaten abgelegt:

- Die Metadaten der Basistabelle und aller dazugehörigen Spalten.  
Die Metadaten für eine Spalte beinhalten auch SQL- bzw. CALL-DML-Defaultwerte.
- Die Metadaten für alle Indizes auf dieser Basistabelle.
- Die Metadaten aller Integritätsbedingungen mit Ausnahme von Referenzbedingungen, die auf dieser Basistabelle definiert sind.  
Hierzu zählt auch eine Nicht-NULL-Bedingung auf einer Spalte, da sie wie eine Check-Bedingung behandelt wird.

Nicht mit abgelegt werden Viewdefinitionen und der Kennwortkatalog von CALL-DML-Tabellen mit Passwortschutz. Der Passwortschutz muss mit dem SESAM/SQL-Dienstprogramm SEPA neu festgelegt werden.

EXPORT TABLE kann auch auf einem Replikat oder Sicherungsbestand der aktuellen SESAM/SQL-Version durchgeführt werden.

Wird eine Export-Datei auf Magnetbandkassette gespeichert, kann sie sich über mehrere physikalische Bänder erstrecken.

Die Anzahl der Sätze, die aus der Tabelle *tabelle* in die Export-Datei kopiert wurden, wird im Kommunikationsbereich des ESQL-COBOL-Programms im Feld `SQLrowcount` zurückgegeben.

Bei Abbruch eines EXPORT-Laufs wird der Inhalt einer unvollständigen Export-Datei logisch gelöscht. Durch das logische Löschen wird verhindert, dass unvollständige Export-Dateien versehentlich für eine IMPORT-Anweisung verwendet werden.

Betroffene Spaces der Tabelle *tabelle* sind während der Durchführung von EXPORT TABLE exklusiv gegen Änderungen durch fremde Transaktionen gesperrt. Auf den betroffenen Spaces und auch auf der Tabelle kann aber parallel gelesen werden. Bei einer partitionierten Tabelle müssen, falls Anwenderdaten exportiert werden sollen, alle betroffenen Partitionen verfügbar sein (siehe [Tabelle 3 auf Seite 22](#)).

Der aktuelle Berechtigungsschlüssel muss das Sonder-Privileg UTILITY besitzen oder Eigentümer des Schemas sein, zu dem die Tabelle gehört.

Wird eine Suchbedingung definiert, muss der aktuelle Berechtigungsschlüssel das SELECT-Privileg auf die entsprechende Tabelle und alle weiteren, in der Suchbedingung angesprochenen Tabellen haben. Er muss auch das EXECUTE-Privileg für alle in der Suchbedingung angesprochenen User Defined Functions (UDF, siehe Handbuch „[SQL-Sprachbeschreibung Teil 1: SQL-Anweisungen](#)“) haben.

---

```
EXPORT TABLE tabelle INTO FILE datei [PASSWORD kennwort] [ { ALL DATA
  WHERE suchbedingung } ]
  NO DATA ]
```

---

#### *tabelle*

Name der Tabelle, die exportiert werden soll. Die Tabelle muss existieren und eine Basistabelle sein. Die Tabelle kann auch partitioniert sein.

Betroffene Spaces dürfen nicht im Zustand „recover pending“ sein. Die Tabelle selber darf sich nicht im Zustand „check pending“ befinden, falls Anwenderdaten exportiert werden.

**INTO FILE** *datei*

Name der Export-Datei. *datei* muss als alphanumerisches Literal angegeben werden. Ist die Datei noch nicht vorhanden, wird sie auf der DBH-Kennung angelegt, als SAM-Datei mit variabler Satzlänge und einer maximalen Blocklänge von 32768 Byte. Als CCSN der Export-Datei wird der CCSN der Datenbank (siehe Klausel CODE-TABLE auf [Seite 108](#)) eingetragen. Wenn für die Datenbank kein CCSN definiert ist, dann wird die Export-Datei ohne CCSN angelegt.

Die Datei darf vor Ausführung von EXPORT TABLE vorhanden sein. Dateiattribute werden ggf. von EXPORT TABLE korrigiert und vorhandene Daten werden überschrieben.

Soll die Export-Datei auf einer anderen Kennung als der DBH liegen, muss der Datenbankverwalter Vorbereitungen für die Export-Datei treffen, siehe Abschnitt „Datenbankdateien und Jobvariablen auf fremden Benutzerkennungen“ im „[Basishandbuch](#)“.

Die Export-Datei darf nicht von einem anderen Anwender geöffnet sein.

Bei Angabe der WHERE-Klausel darf die Export-Datei nicht auf Band liegen.



Wenn eine Tabelle exportiert wird, die Spalten mit dem den Datentypen NATIONAL CHARACTER (VARYING) enthält, dann kann die Export-Datei nicht mehr mit SESAM/SQL bis V4.0 importiert werden.

**PASSWORD** *kennwort*

Eine neu angelegte Export-Datei wird mit Passwort-Schutz angelegt. Eine bereits bestehende Datei muss mit dem Schreib-Passwort geöffnet werden.

Für die Angabe von *kennwort* gibt es folgende Möglichkeiten:

- 'C'*zeichenkette*"  
*zeichenkette* enthält vier abdruckbare Zeichen.
- 'X'*hex-zeichenkette*"  
*hex-zeichenkette* enthält acht sedezimale Zeichen.
- 'n'  
*n* bezeichnet eine ganze Zahl von - 2147483648 bis + 2147483647.

**ALL DATA**

Alle Anwenderdaten werden in die Export-Datei geschrieben.

Bei einer partitionierten Tabelle müssen alle Partitionen der Tabelle verfügbar sein (siehe [Tabelle 3 auf Seite 22](#)).

**WHERE** *suchbedingung*

Es werden nur die Anwenderdaten übernommen, die die Suchbedingung erfüllen.

Die Tabelle *tabelle* darf keine Nur-CALL-DML-Tabelle sein.

Bei einer partitionierten Tabelle müssen alle betroffenen Partitionen der Tabelle verfügbar sein (siehe [Tabelle 3 auf Seite 22](#)).

In der Suchbedingung dürfen keine Benutzervariablen und keine Fragezeichen als Platzhalter verwendet werden.

Es kann auf andere Tabellen als die Tabelle *tabelle* referenziert werden. Unter den referenzierten Tabellen dürfen jedoch keine Nur-CALL-DML-Tabellen sein.

Die Abarbeitung der Suchbedingung auf andere referenzierte Tabellen als *tabelle* erfolgt in dem Standard-Isolationslevel der Sitzung. Dieses gilt nicht, wenn Sie mit SET TRANSACTION oder dem Pragma ISOLATION LEVEL einen anderen Wert angegeben haben.

Bei WHERE *suchbedingung* sind weitere Pragmas wirksam (siehe Handbuch „[SQL-Sprachbeschreibung Teil 1: SQL-Anweisungen](#)“).

**NO DATA**

Es werden keine Anwenderdaten in die Export-Datei übernommen.

**Beispiel**

In dem Beispiel wird die Tabelle AUFTRAG mit allen Anwenderdaten aus dem Schema AUFTRAGSVER der Datenbank AUFTRAGKUNDEN in die Exportdatei DAT.EXPORT.AUFTRAG exportiert.



```
EXPORT TABLE auftragkunden.auftragsver.auftrag  
INTO FILE 'DAT.EXPORT.AUFTRAG' ALL DATA
```

**Siehe auch**

IMPORT TABLE

## IMPORT TABLE - Tabelle aus einer Export-Datei in eine Datenbank importieren

Mit IMPORT TABLE können Sie eine nicht-partitionierte oder partitionierte Basistabelle erzeugen, deren Struktur aus einer Export-Datei importiert wird. Der Coded Character Set Name (CCSN) der Export-Datei wird zur Prüfung gegen den codierten Zeichensatz der Datenbank verwendet.

Die Tabelle wird mit allen Spalten und dem Primärschlüssel (falls vorhanden) erzeugt. Wahlweise können Sie noch Anwenderdaten, Integritätsbedingungen oder Indizes aus der Export-Datei übernehmen.

Weitere Informationen zu der Export-Datei und zu den Anwendungsmöglichkeiten der Kombination EXPORT/ IMPORT TABLE finden Sie im [Abschnitt „EXPORT TABLE - Tabelle aus einer Datenbank in eine Exportdatei exportieren“ auf Seite 129](#).

Aus den Metadaten der Export-Datei wird in der angegebenen Datenbank eine Basistabelle wie folgt erzeugt:

- Der Tabellentyp wird von der exportierten Tabelle übernommen. Für eine partitionierte Tabelle kann nur der Tabellentyp SQL oder CALL-DML übernommen werden.
- Alle Spalten der exportierten Tabelle werden übernommen. Der Datentyp, die Voreinstellungen und die Reihenfolge der Spalten bleiben erhalten.
- Existiert ein Primärschlüssel, wird dieser übernommen. Für partitionierte Tabellen muss ein Primärschlüssel vorhanden sein. Auch bei Angabe von NO CONSTRAINT wird der Primärschlüssel importiert. Der Name des Primärschlüssels bleibt gleich. Existiert in dem aktuellen Schema bereits eine gleichnamige Integritätsbedingung, wird ein neuer Name generiert. In diesem Fall wird eine Warnung ausgegeben.

Nicht mit importiert wird der Kennwortkatalog einer CALL-DML-Tabelle mit Passwortschutz. Der Passwortschutz muss für eine nicht-partitionierte Tabelle mit dem SESAM/SQL Dienstprogramm SEPA neu festgelegt werden. Für eine partitionierte Tabelle kann mit SEPA kein Passwortschutz festgelegt werden.

Die Anzahl der Sätze, die aus der Export-Datei in die Tabelle *tabelle* kopiert wurden, wird im Kommunikationsbereich des ESQL-COBOL-Programms im Feld `SQLrowcount` zurückgegeben.

Die IMPORT TABLE-Anweisung wird in folgenden Schritten durchgeführt:

- Erzeugen der leeren Tabelle auf dem oder den entsprechenden Anwender-Spaces. Dabei müssen alle Spaces, auf denen die Tabelle angelegt werden soll, physikalisch verfügbar sein (siehe [Tabelle 3 auf Seite 22](#)).
- Eintragen der Anwenderdaten (optional)

- Erzeugen der Indizes und Integritätsbedingungen (optional)

Die betroffenen Spaces sind während der jeweiligen Schritte in unterschiedlicher Weise gesperrt:

- Exklusiv gesperrt werden zunächst der oder die Spaces, auf denen die Tabelle angelegt werden soll.
- Lesende Zugriffe fremder Transaktionen auf dem Space sind wieder möglich, nachdem die leere Tabelle erzeugt wurde. Davon ausgenommen ist die gerade angelegte Tabelle.
- Exklusiv gesperrt wird anschließend der Space, auf dem Indizes angelegt werden. Dieser Fall tritt ein, wenn WITH INDEX oder WITH CONSTRAINT angegeben wird.

Ist das Logging für den Space eingeschaltet, auf dem die Tabelle oder eine Partition davon erstellt wird, befindet er sich nach IMPORT TABLE im Zustand „copy pending“. Dieser Fall tritt nur ein, wenn Datensätze in die Tabelle übertragen werden.

Der aktuelle Berechtigungsschlüssel muss das Sonder-Privileg UTILITY besitzen oder Eigentümer des Schemas sein, zu dem die Tabelle gehört.

Wenn die Klausel USING nicht angegeben wird und noch kein Default-Space des Schema-Eigentümers besteht, muss der aktuelle Berechtigungsschlüssel das Nutzungsrecht für die Default-Storage Group D0STOGROUP besitzen.

Tritt während der Ausführung von IMPORT TABLE ein Fehler auf, sind zwei Fälle zu unterscheiden:

- Die logische Datensicherung für den Space ist eingeschaltet und der Space ist im Zustand „copy pending“:  
SESAM/SQL hat die Tabelle erzeugt und die Anwenderdaten eingetragen.  
Sie können jetzt:
  - den Space mit COPY sichern und anschließend Indizes sowie Integritätsbedingungen nachträglich erzeugen,
  - den Space mit COPY sichern, die Tabelle mit DROP TABLE löschen und IMPORT TABLE wiederholen,
  - den Space mit RECOVER TO zurücksetzen, die Tabelle mit DROP TABLE löschen und IMPORT TABLE wiederholen.
- Der Space ist nicht im Zustand „copy pending“:  
Sie sollten den Space mit RECOVER SPACE reparieren, die Tabelle mit DROP TABLE löschen und IMPORT TABLE wiederholen.

---

```

IMPORT TABLE tabelle [USING {SPACE space
PARTITION BY RANGE partition, ..., letzte_partition } ]

FROM FILE datei [PASSWORD kennwort] { [ALL DATA] [ {NEW ROW_IDS} ]
NO DATA }

[ {WITH INDEX [USING SPACE space] } ] [ {WITH CONSTRAINT } ]
[ NO INDEX ] [ NO CONSTRAINT ]

partition ::= PARTITION partnr VALUE { <
<= } (spaltenwert, ...) ON SPACE space

letzte_partition ::= PARTITION partnr [VALUE <=( )] ON SPACE space

partnr ::= vorzeichenlose_ganzzahl

spaltenwert ::= { alphanumerisches_literal
numerisches_literal
zeit_literal }

```

---

### *tabelle*

Die Basistabelle erhält den Namen *tabelle*.

Der Name muss nicht mit dem Namen der exportierten Tabelle übereinstimmen.

Die Tabelle *tabelle* darf noch nicht existieren. Es muss aber das Schema vorhanden sein, in das die Tabelle importiert werden soll.

#### USING SPACE *space*

Es wird eine nicht-partitionierte Tabelle angelegt.

Der Space *space* muss existieren und der Space-Eigentümer muss gleichzeitig der Schema-Eigentümer sein.

Der Space darf sich nicht im Zustand „recover pending“ befinden.

#### USING PARTITION BY RANGE *partition*, ..., *letzte\_partition*

Es wird eine partitionierte Tabelle angelegt.

Gibt an, dass eine partitionierte Tabelle mit mindestens 2 und höchstens 16 Partitionen erzeugt werden soll. Alle Partitionen einer Tabelle müssen auf unterschiedlichen Spaces liegen und alle Spaces müssen bereits in der Datenbank definiert sein. Die Tabelle muss einen Primärschlüssel besitzen. Dieser kann eine einzelne Spalte oder eine Kombination aus mehreren Spalten sein.

Für partitionierte Tabellen muss bei Auswahl von ALL DATA auch die Klausel NEW ROW\_IDS verwendet werden.

$$\textit{partition} ::= \text{PARTITION } \textit{partnr} \text{ VALUE } \left\{ \begin{array}{l} < \\ <= \end{array} \right\} (\textit{spaltenwert} [ , \dots ])$$

ON SPACE *space*

Legt die Eigenschaften einer Partition fest.

*partnr* ist eine positive ganze Zahl von 1 bis 16 und stellt die laufende Nummer der Partition dar. *partnr* muss für die einzelnen Partitionen aufsteigend vergeben werden. Werden weniger als 16 Partitionen definiert, darf die Nummernfolge Lücken enthalten und die erste Partition muss nicht mit 1 beginnen.

(*spaltenwert,...*) ist eine Folge von Spaltenwerten, die für die betreffende Partition die Obergrenze des Primärschlüsselintervalls definiert. *spaltenwert* muss ein Datenformat besitzen, das mit dem Datenformat der entsprechenden Spalte des Primärschlüssels vergleichbar ist. Zusätzlich gelten für *spaltenwert* die Regeln für die „Defaultwerte für Tabellenspalten“, siehe Handbuch „[SQL-Sprachbeschreibung Teil 1: SQL-Anweisungen](#)“.

Die untere Grenze der Partition ergibt sich implizit aus der Obergrenze der vorhergehenden Partition bzw. aus dem niedrigsten Primärschlüssel der Tabelle (bei der ersten Partition). Alle eingelesenen Sätze aus dem damit definierten Primärschlüsselintervall werden dieser Partition zugeordnet.

Durch den vorangestellten Vergleichsoperator wird die Obergrenze mit eingeschlossen oder ausgeschlossen:

- <= Sätze mit Primärschlüsselwert gleich *spaltenwert,...* gehören zu **dieser** Partition
- < Sätze mit Primärschlüsselwert gleich *spaltenwert,...* gehören zur **nächsten** Partition

Für den Vergleich gelten die lexikografischen Regeln, siehe Handbuch „[SQL-Sprachbeschreibung Teil 1: SQL-Anweisungen](#)“.

Besteht der Primärschlüssel aus einer Kombination mehrerer Spalten, dann dürfen Sie maximal soviele Werte angeben wie Spalten im Primärschlüssel vorhanden sind. Geben Sie weniger Elemente an als Spalten im Primärschlüssel definiert sind, dann ergänzt SESAM/SQL die fehlenden Werte, indem jeweils der höchste Wert gesetzt wird, der zum Datentyp der entsprechenden Spalte passt. Die angegebenen Obergrenzen müssen für die einzelnen Partitionen streng aufsteigend sein.

*space* gibt den Namen des Space an, in dem diese Partition abgespeichert wird. Der Space muss existieren und der Space-Eigentümer gleichzeitig der Schema-Eigentümer sein. Die Spaces der Partitionen einer partitionierten Tabelle müssen disjunkt sein, d.h. ein Space darf nicht für zwei Partitionen verwendet werden. Der Space darf sich nicht im Zustand „recover pending“ befinden.

*letzte\_partition* ::= PARTITION *partnr* [VALUE <=( )] ON SPACE *space*

Für die letzte Partition gelten die gleichen Bedingungen wie für *partition*.

Lediglich die Obergrenze darf nicht angegeben werden, da sie sich hier aus dem jeweils höchsten Primärschlüsselwert ergibt. Die VALUE-Klausel kann daher auch weggelassen werden.

USING nicht angegeben

Es wird eine nicht-partitionierte Tabelle auf dem Default-Space des Schema-Eigentümers angelegt. Ist noch kein Default-Space vorhanden, wird er implizit erzeugt.

FROM FILE *datei*

Als *datei* muss der Name der Export-Datei angegeben werden, die importiert werden soll. *datei* muss als alphanumerisches Literal angegeben werden.

Befindet sich die Datei in einer anderen BS2000-Kennung als der DBH, muss die DBH-Kennung leseberechtigt sein.

Der CCSN der Export-Datei muss mit dem CCSN der Datenbank (siehe Klausel CODE-TABLE auf [Seite 108](#)) übereinstimmen. Wenn für die Datenbank kein CCSN verwendet wird, dann wird auch der CCSN der Export-Datei ignoriert.



Eine Export-Datei, die eine Tabelle mit dem den Datentypen NATIONAL CHARACTER (VARYING) enthält, kann nicht mehr mit SESAM/SQL bis V4.0 importiert werden.

PASSWORD *kennwort*

Ist die Export-Datei kennwortgeschützt, muss das Lese-Kennwort angegeben werden.

Für die Angabe von *kennwort* gibt es folgende Möglichkeiten:

- 'C'*zeichenkette*"  
*zeichenkette* enthält vier abdruckbare Zeichen.
- 'X'*hex-zeichenkette*"  
*hex-zeichenkette* enthält acht sedezimale Zeichen.
- 'n'  
*n* bezeichnet eine ganze Zahl von - 2147483648 bis + 2147483647.

[ALL DATA]

Alle in der Export-Datei eingetragenen Anwenderdaten werden in die Tabelle geladen. Sind keine Anwenderdaten vorhanden, wird eine Warnung ausgegeben.

NEW ROW\_IDS

Hat die Tabelle einen Primärschlüssel, werden für die importierten Sätze neue Satznummern in linear aufsteigender Reihenfolge vergeben.

Hat die Tabelle keinen Primärschlüssel, werden die Satznummern der exportierten Tabelle beibehalten. Es wird eine Warnung ausgegeben.

**OLD ROW\_IDS**

Die Satznummern der exportierten Tabelle werden übernommen.

Soll eine partitionierte Tabelle erzeugt werden, dann ist die Angabe von OLD ROW\_IDS unzulässig.

War die Tabelle, aus der die Export-Datei erzeugt wurde, partitioniert, dann wird die Angabe von OLD ROW\_IDS ignoriert. Es wird ein WARNING ausgegeben und NEW ROW\_IDS wird verwendet.

**NO DATA**

Es wird eine leere Basistabelle erzeugt.

**WITH INDEX [USING SPACE *space*]**

Alle Indizes werden übernommen. Sind in der Export-Datei keine Metadaten zu Indizes vorhanden, wird eine Warnung ausgegeben.

Der Name des Index wird aus der exportierten Tabelle übernommen. Ist in dem aktuellen Schema bereits ein gleichnamiger Index vorhanden, wird ein neuer Name generiert und eine Warnung ausgegeben.

Die Statistikinformation des Index wird aktualisiert.

Für Indizes, die Eindeutigkeitsbedingungen zugeordnet sind, gilt:

- Es wird WITH INDEX und WITH CONSTRAINT angegeben und der Index ist in der exportierten Tabelle explizit erzeugt worden:  
In diesem Fall wird der Index auf dem gleichen Space angelegt wie die anderen Indizes und gilt weiterhin als explizit erzeugt.
- In allen anderen Fällen:  
Der Index wird auf dem gleichen Space angelegt wie die Tabelle und gilt als implizit erzeugt.

Geben Sie keinen Index-Space an, wird der Index bei einer nicht-partitionierten Tabelle auf dem gleichen Space wie die Tabelle angelegt. Bei einer partitionierten Tabelle wird er auf dem Space der ersten Partition angelegt.

**USING SPACE *space***

Wenn Sie einen Index-Space angeben, dann wird der Index auf diesem Space angelegt.

Der Space-Eigentümer muss der Eigentümer des Schemas sein.

Der Index-Space darf sich nicht im Zustand „recover pending“ befinden.

**NO INDEX**

Indizes, die sich nicht auf Eindeutigkeitsbedingungen beziehen, werden nicht übernommen.

**WITH CONSTRAINT**

Alle Eindeutigkeits- und Check-Bedingungen werden übernommen. Enthält die Export-Datei hierfür keine Metadaten, wird eine Warnung ausgegeben.

Der Name der Integritätsbedingung wird aus der exportierten Tabelle übernommen.

Ist in dem aktuellen Schema bereits eine gleichnamige Integritätsbedingung vorhanden, wird ein neuer Name generiert und eine Warnung ausgegeben. Enthält der Text einer Check-Bedingung qualifizierte Spaltennamen, müssen die dort angegebenen Schema- und Tabellennamen mit den entsprechenden Namen der aktuellen Tabelle identisch sein. Andernfalls wird die Check-Bedingung nicht übernommen und eine Warnung ausgegeben.

### NO CONSTRAINT

Eindeutigkeitsbedingungen und Check-Bedingungen werden nicht übernommen.

## Beispiele

Im ersten Beispiel wird die Tabelle AUFTRAG unter dem Namen AUFTRAG\_NEU aus der Exportdatei DAT.EXPORT.AUFTRAG in das Schema EXPORT\_SCHEMA der Datenbank AUFTRAGKUNDEN importiert und auf dem Space TABLESPACE gespeichert. Dabei werden alle Anwenderdaten geladen, die in der Exportdatei enthalten sind. Die Satznummern der exportierten Tabelle werden übernommen. Die Indizes auf der Tabelle AUFTRAG werden übernommen und auf dem Space INDEXSPACE gespeichert. Eindeutigkeitsbedingungen und Check-Bedingungen werden nicht auf die neue Tabelle übertragen.



```
IMPORT TABLE auftragkunden.export_schema.auftrag_neu
  USING SPACE tablespace
  FROM FILE 'DAT.EXPORT.AUFTRAG' ALL DATA OLD ROW_IDS
  WITH INDEX USING SPACE indexspace
  NO CONSTRAINT
```

Im zweiten Beispiel wird die Tabelle AUFTRAG aus der Exportdatei DAT.EXIMPT.AUFTRAG importiert. AUFTRAG wird dabei als partitionierte Tabelle auf den Spaces TABLESPACE, TABLESP002 und TABLESP003 gespeichert. Es werden alle Anwenderdaten geladen, die in der Exportdatei enthalten sind. Die Satznummern der exportierten Tabelle werden neu vergeben. Die Indizes der Tabelle AUFTRAG werden übernommen und auf dem Space INDEXSPACE gespeichert. Bedingungen werden nicht auf die neue Tabelle übertragen.



```
IMPORT TABLE auftrag
  USING PARTITION BY RANGE
    PARTITION 01 VALUE <= (299) ON SPACE tablespace,
    PARTITION 02 VALUE <= (399) ON SPACE tablesp002,
    PARTITION 03           ON SPACE tablesp003
  FROM FILE 'DAT.EXIMPT.AUFTRAG' ALL DATA NEW ROW_IDS
  WITH INDEX USING SPACE indexspace
  NO CONSTRAINT
```

## Siehe auch

EXPORT TABLE

## LOAD - Anwenderdaten in eine Basistabelle laden

Mit LOAD können Sie Anwenderdaten aus einer BS2000-Datei vom Typ SAM in eine Basistabelle einer SESAM/SQL-Datenbank laden. Sie können angeben, in welche Spalten der Tabelle geladen werden soll. Die Anzahl eingefügter Sätze wird im Kommunikationsbereich des ESQLE-COBOL-Programms im Feld `SQLrowcount` zurückgegeben. Sie können auch die ersten Sätze (z.B. Kommentare oder bereits eingelesene Sätze) vom Einlesen ausnehmen.

SESAM/SQL bietet mehrere Ladeformate an, nämlich `TRANSFER_FORMAT`, `UNLOAD_FORMAT`, `DELIMITER_FORMAT`, `CSV_FORMAT` und die Standarddarstellung; außerdem ist es möglich, die Darstellung der einzelnen Spalten in der Eingabedatei selbst zu definieren (siehe auch [„LOAD-Formate“ auf Seite 161](#)). Falls weder ein Ladeformat angegeben noch die Darstellung der einzelnen Spalten beschrieben ist, so müssen alle Werte der Eingabedatei in der Standarddarstellung vorliegen, die dem Datentyp der jeweils zugeordneten Spalte in der Basistabelle entspricht.

Der Coded Character Set Name (CCSN) der Eingabedatei wird zur Prüfung gegen den codierten Zeichensatz der Datenbank verwendet.

Im `DELIMITER_FORMAT`, im `CSV_FORMAT` und im selbst definierten Format werden zu ladende Zeichenketten nötigenfalls in den codierten Zeichensatz der Datenbank konvertiert.

Zusammen mit der Utility-Anweisung UNLOAD bietet LOAD die Möglichkeit, Anwenderdaten aus einer Basistabelle in eine andere zu übertragen. Als Eingabedatei für LOAD dient hierbei eine zuvor mit UNLOAD erstellte Ausgabedatei. Für diesen Zweck können die vier namentlich genannten Formate genutzt werden (die Angabe `UNLOAD_FORMAT` bei LOAD entspricht der Angabe `LOAD_FORMAT` bei UNLOAD). Bei der Standarddarstellung und bei selbst definierten Formaten ist die Behandlung von NULL-Werten genau zu beachten (siehe Klausel `WHEN ... THEN NULL`).

Sollen Anwenderdaten zusammen mit der Tabellenstruktur einer Basistabelle kopiert werden, bietet sich die Verwendung des Anweisungspaares `EXPORT/IMPORT TABLE` an. Mit diesem Anweisungspaar ist es auch möglich, nur die Metadaten einer Basistabelle zu kopieren. Nähere Informationen zu den Anwendungsmöglichkeiten von `EXPORT/IMPORT TABLE` finden Sie im [Abschnitt „EXPORT TABLE - Tabelle aus einer Datenbank in eine Exportdatei exportieren“ auf Seite 129](#) und im [Abschnitt „IMPORT TABLE - Tabelle aus einer Export-Datei in eine Datenbank importieren“ auf Seite 133](#).

Bei der Ausführung der LOAD-Anweisung können Sie zwischen den Betriebsarten `OFFLINE` und `ONLINE` wählen. Die wesentlichen Unterschiede betreffen die Performance, das Sperrverhalten sowie die Fehlerbehandlung nach Abbruch einer LOAD-Anweisung. Detaillierte Informationen zu `OFFLINE/ONLINE` finden Sie auf [Seite 143](#).

Bei der Ausführung der LOAD-Anweisung prüft SESAM/SQL in jedem Fall die Primärschlüsselbedingung. Falls Integritätsbedingungen definiert sind, müssen Sie im Rahmen des Ladevorgangs oder danach in jedem Fall die Einhaltung der Integritätsbedingungen prüfen.

Solange die Integritätsbedingungen nicht geprüft sind, befindet sich die Basistabelle, in die geladen wurde, im Zustand „check pending“. Sie ist dann für andere Benutzer gesperrt (siehe [Seite 13](#)). Andere Tabellen, die auf demselben Anwender-Space liegen, können nur gelesen werden.

Außerdem müssen bei LOAD OFFLINE die Indizes der Tabelle neu aufgebaut werden. Im Allgemeinen ist es nicht sinnvoll, die Integritätsbedingungen zu prüfen, bevor die Indizes neu aufgebaut sind, da die Prüfung der Integritätsbedingungen intakte Indizes voraussetzt. Bei LOAD ONLINE werden Indizes immer automatisch angepasst.

Der aktuelle Berechtigungsschlüssel muss das Sonder-Privileg UTILITY besitzen oder Eigentümer des Schemas sein, in dem die Tabelle liegt, in die geladen werden soll.

Die Möglichkeiten der Darstellung der Daten im Eingabesatz sind im [Abschnitt „Datendarstellung in den Ein- und Ausgabedateien bei LOAD und UNLOAD“ auf Seite 26](#) beschrieben.

---

```
LOAD [ { OFFLINE } ] [SORTED] FILE datei [PASSWORD kennwort] [(ladedarstellung,...)]
```

```
[SKIP FIRST zahl RECORDS]
```

```
INTO TABLE tabelle [(ladespalte,...)] [ladeparameter]...
```

```
{
TRANSFER_FORMAT
UNLOAD_FORMAT
DELIMITER_FORMAT TERMINATED BY delimiter
CSV_FORMAT DELIMITER delimiter [QUOTE quote] [ESCAPE escape]
}
```

```
[USING FILE datei [PASSWORD kennwort]]
```

```
[COUNTING_FIELD spalte]
```

```
ladedarstellung ::= POSITION ( { zahl } ) [ datentyp ] [ WHEN nullbedingung THEN NULL ]
```

```
nullbedingung ::= {
spalte vergleichs_op literal
POSITION ( { zahl } ) vergleichs_op { alphanumerisches_literal }
{ national_literal }
}
```

```
vergleichs_op ::= {
=
<
>
<=
>=
<>
}
```

```
ladespalte ::= {
spalte
spalte(posnr)
spalte(min..max)
}
```

```
ladeparameter ::= {
[NO] OVERWRITE
{
NO INDEX
GENERATE INDEX
}
[NO] CONSTRAINT CHECK
}
```

---

{ OFFLINE }  
{ ONLINE }

Mit OFFLINE oder ONLINE legen Sie die Betriebsart des LOAD fest. Dabei bestehen folgende Unterschiede:

- Bei LOAD OFFLINE ist während des Zuladens der Anwender-Space exklusiv gegen Ändern gesperrt. Die Tabelle selbst ist exklusiv gegen Lesen gesperrt; sonstige Tabellen und Indizes auf dem Anwender-Space können von fremden Transaktionen per DML gelesen werden. Ist GENERATE INDEX angegeben, dann sind während der Index-Generierung (nach dem Zuladen der Primärdaten) die Index-Spaces exklusiv gesperrt. Die Verarbeitung findet überwiegend in einer Service-Task statt.

Bei LOAD ONLINE wird der Anwender-Space nicht exklusiv gesperrt. Es werden lediglich die gleichen Sperren angefordert wie im Rahmen einer DML-Änderungsanweisung. In Deadlock-Situationen wird der LOAD ONLINE als Utility gegenüber DML-Anweisungen bevorzugt, so dass er nur bei Konkurrenz mit anderen Utilities abgebrochen (aber nicht zurückgesetzt) werden kann. Falls die Tabelle in den Zustand „check pending“ gelangt, dann wird der Tabellen-Space zum Festschreiben dieses Zustandes temporär exklusiv gesperrt. Die Verarbeitung findet vollständig in einer DBH-Task statt.

- Bei der Ausführung der LOAD-Anweisung wird bei LOAD OFFLINE das Logging (LD-Sicherung) für den betroffenen Anwender-Space unterbrochen. Somit können Änderungen bei einer Reparatur nur bis zum Beginn des Ladevorgangs nachgefahren werden. Bei Tabellen, die auf einem Anwender-Space mit Logging liegen, muss daher nach Ausführung der letzten LOAD-Anweisung und anschließender erfolgreicher Constraint-Prüfung und ggf. Indexaufbau mit der COPY-Anweisung ein SESAM-Sicherungsbestand des betroffenen Anwender-Space erstellen werden. Andernfalls verbleibt er im Zustand „copy pending“ und kann nur mit Utility-Anweisungen bearbeitet werden (siehe [Seite 18](#)).

Bei LOAD ONLINE wird das Logging nicht unterbrochen. Die zugeladenen Sätze für einen Anwender-Space, der mit Logging betrieben wird, werden auf dem DA-LOG protokolliert. Der betroffene Anwender-Space erhält also nicht den Zustand „copy pending“. Insbesondere können Replikat-Spaces nach LOAD ONLINE mit der Anweisung REFRESH REPLICATION aktualisiert werden.

Die Ausführung von LOAD ONLINE wird abgewiesen, wenn sich der Anwender-Space oder ein betroffener Index-Space im Zustand „copy pending“ befindet.

- Beim Zuladen relativ großer Datenmengen (insbesondere beim Zuladen in eine leere Tabelle) ist LOAD OFFLINE schneller als LOAD ONLINE. Relativ groß bedeutet, dass die Anzahl der Sätze in der Eingabedatei verglichen mit der Anzahl der bereits in der Basistabelle existierenden Sätze groß ist.

Beim Laden relativ kleiner Datenmengen ist LOAD ONLINE schneller als LOAD OFFLINE, insbesondere, wenn Indizes aktualisiert und Integritätsbedingungen geprüft werden sollen. Relativ klein bedeutet, dass die Anzahl zuzuladender Sätze im Verhältnis zu den bereits in der Basistabelle existierenden Sätzen gering ist.

Der Grund für diese Performance-Unterschiede liegt u.a. in der Behandlung der Indizes (siehe INDEX-Klausel auf [Seite 152](#)) und der Integritätsbedingungen (siehe CHECK-Klausel auf [Seite 153](#)), siehe Handbuch „[Performance](#)“, Kapitel „Performance-relevante Aspekte von Utility-Funktionen“.

- Wird LOAD OFFLINE mit Fehler abgebrochen, nachdem bereits ein Teil der Sätze zugeladen wurde, dann ist der Tabellen-Space im Zustand „load running“ und die zugehörigen Indizes sind defekt. Unbeteiligte Objekte auf dem Tabellen-Space können noch gelesen, aber nicht geändert werden. Zur Reparatur des Spaces ist ein RECOVER SPACE und ggf. anschließend RECOVER INDEX erforderlich.

Wird LOAD ONLINE mit einem Fehler oder wegen eines Deadlocks abgebrochen, nachdem bereits ein Teil der Sätze zugeladen wurde, dann bleiben diese Sätze erhalten und die Tabelle ist nach wie vor formal korrekt, möglicherweise aber im Zustand „check pending“ sofern die Integritätsbedingungen nicht geprüft werden konnten.

- Bei LOAD ONLINE dürfen Eingabedatei und Fehlerdatei nicht auf Band liegen.

[SORTED] FILE *datei* [PASSWORD *kennwort*]

Name der Eingabedatei. *datei* muss als alphanumerisches Literal angegeben werden. Bei LOAD ONLINE darf *datei* keine Banddatei sein.

Die Eingabedatei muss eine SAM-Datei mit variabler Satzlänge sein und darf nicht leer sein. Sie darf nicht mehrere Sätze mit demselben Primärschlüsselwert enthalten; andernfalls ist undefiniert, welcher dieser Sätze zugeladen wird. Liegt die Eingabedatei nicht in der Kennung des DBH, so muss die DBH-Kennung eine Leseberechtigung für diese Datei haben. Andernfalls kann der DBH nicht auf die Eingabedatei zugreifen.

Der Coded Character Set Name (CCSN) der Eingabedatei wird geprüft. Er darf für alle LOAD-Formate ein EBCDIC-Zeichensatz, ein selbst definierter Zeichensatz auf EBCDIC-Basis bzw. \*NONE sein. Für das DELIMITER\_FORMAT darf der Zeichensatz der Eingabedatei auch UTFE sein (siehe [Seite 155](#)).

Bei SAM-Dateien ist die Satzlänge beschränkt. Ein Satz der SAM-Datei wird genau einem Satz der Basistabelle zugeordnet (Ausnahme: CSV-Format). Insbesondere beim Laden in Spalten vom Datentyp (NATIONAL) CHARACTER VARYING kann es somit erforderlich sein, die zu ladenden Daten nach Spalten getrennt auf mehrere Eingabedateien zu verteilen. Das Laden erfordert dann mehrmalige Ausführung der Utility-Anweisung LOAD.

Das Format einer CSV-Datei ist im Handbuch „[SQL-Sprachbeschreibung Teil 1: SQL-Anweisungen](#)“ ausführlich beschrieben.

SORTED angeben:

SESAM/SQL setzt voraus, dass die Eingabedatei nach Primärschlüsselwerten sortiert ist. Bei nicht sortierter Eingabedatei erfolgt für jeden falsch einsortierten Satz ein Eintrag in die Fehlerdatei.

Die Angabe SORTED bleibt ohne Wirkung, wenn in eine Tabelle geladen werden soll, die ohne Primärschlüssel definiert ist, oder wenn ONLINE angegeben ist.

SORTED nicht angeben:

Eine mögliche Sortierung der Eingabedatei nach Primärschlüsselwerten ist für die Ausführung von LOAD OFFLINE ohne Bedeutung, da SESAM/SQL die Sortierung durchführt.

Bei LOAD ONLINE wird eine Sortierung der Eingabedatei nach Primärschlüsselwerten nicht explizit ausgenutzt. Gerade bei großen Ladedateien kann durch eine Sortierung ein besseres Pufferverhalten in SESAM/SQL und dadurch eine deutlich bessere Performance erzielt werden.

Bei Tabellen, die ohne Primärschlüssel definiert sind, ist die Reihenfolge nicht definiert, in der die Sätze in die Tabelle eingefügt werden.

PASSWORD *kennwort*

BS2000-Kennwort, falls die Eingabedatei kennwortgeschützt ist. *kennwort* muss als alphanumerisches Literal angegeben werden.

Für die Angabe von *kennwort* gibt es folgende Möglichkeiten:

- 'C'*zeichenkette*"  
*zeichenkette* enthält vier abdruckbare Zeichen.
- 'X'*hex-zeichenkette*"  
*hex-zeichenkette* enthält acht sedezimale Zeichen.
- 'n'  
*n* bezeichnet eine ganze Zahl von - 2147483648 bis + 2147483647.

(*ladedarstellung* ,...)

Liste der Ladedarstellungen.



Falls weder (*ladedarstellung*, ... ) noch TRANSFER\_FORMAT, UNLOAD\_FORMAT oder DELIMITER\_FORMAT angegeben ist, so müssen alle Werte der Eingabedatei in der Standarddarstellung vorliegen, die dem Datentyp der jeweils zugeordneten Spalte der Basistabelle *tabelle* entspricht.

Die Ladedarstellungen beschreiben die Formate der Eingabewerte in der Eingabedatei und die Positionen der Eingabewerte im Eingabesatz. Es gelten die folgenden Regeln:

- Entweder muss für jede aus Eingabewerten gleichen Typs gebildete Spalte der Eingabedatei eine Ladedarstellung angegeben werden, oder es darf überhaupt keine Ladedarstellung angegeben werden.

- Ist nur die Liste der Ladedarstellungen angegeben, nicht jedoch die Liste der Ladespalten (*ladespalte* ,...), so ordnet SESAM/SQL die angegebenen Ladedarstellungen bzw. Positionen der Reihe nach den Spalten der Tabelle *tabelle* zu.
- Ist sowohl die Liste der Ladedarstellungen als auch die Liste der Ladespalten angegeben, so ordnet SESAM/SQL jedem Element aus der Liste der Ladespalten eine Ladedarstellung zu. Somit müssen die Liste der Ladedarstellungen und die Liste der Ladespalten gleich viele Elemente enthalten.
- Für eine multiple Spalte ist stets nur die Angabe der Ladedarstellung für ein einzelnes Spaltenelement erforderlich. Dies gilt insbesondere auch dann, wenn die Spalte als Ganzes, d.h. über den Spaltennamen angesprochen wird oder wenn über einen Teilbereich mehrere Spaltenelemente gleichzeitig angesprochen werden. Entsprechendes gilt für die WHEN...THEN NULL-Klausel.
- Die Liste der Ladedarstellungen darf nicht angegeben werden, wenn eines der Formate TRANSFER\_FORMAT, UNLOAD\_FORMAT, CSV\_FORMAT oder DELIMITER\_FORMAT angegeben ist.

Die Datendarstellung bei LOAD ist im [Abschnitt „Datendarstellung in den Ein- und Ausgabedateien bei LOAD und UNLOAD“ auf Seite 26](#) beschrieben.

*ladedarstellung* ::=

POSITION (*zahl*)

Positive ganze Zahl, die die Position des Eingabewerts im Eingabesatz angibt. Positionsangaben, die zu sich überlappenden Eingabefeldern führen, sind zulässig. SESAM/SQL prüft jedoch nicht auf sich überlappende Eingabefelder ab.

Da beim Laden Sätze variabler Länge erlaubt sind, ist darauf zu achten, dass eine in *ladedarstellung* angegebene Position nicht außerhalb des Satzes liegt.

Ist nur die Liste der Ladedarstellungen angegeben, nicht jedoch die Liste der Ladespalten (*ladespalte* , ...), so ordnet SESAM/SQL die angegebenen Ladedarstellungen bzw. Positionen der Reihe nach den Spalten der Tabelle *tabelle* zu.

POSITION (\*)

POSITION (\*) muss angegeben werden, wenn eine genaue Positionsangabe nicht möglich oder nicht erwünscht ist. Die Anfangspositionen errechnen sich dann jeweils aus voranstehenden Werten der Eingabedatei.

*datentyp*

Datentyp des Eingabewertes. Die Datentypen entsprechen den Datentypen an der SQL-Schnittstelle (siehe Handbuch „[SQL-Sprachbeschreibung Teil 1: SQL-Anweisungen](#)“).

Für *datentyp* dürfen Sie nicht FLOAT angeben. Anstatt FLOAT müssen Sie REAL oder DOUBLE PRECISION verwenden.

Die Angabe von *dimension* in *datentyp* ist nicht zulässig.

Fehlen die Angaben zu *länge* bzw. *stellen* [,*bruchteil*], so setzt SESAM/SQL für *länge* bzw. *stellen* jeweils den Wert 1 und für *bruchteil* den Wert 0. Beim Datentyp (NATIONAL) CHARACTER VARYING ignoriert SESAM/SQL die Angabe zu *länge*.

*datentyp* muss mit dem Datentyp der zugeordneten Ladespalte verträglich sein.

Wenn Sie die Eingabewerte für eine bestimmte Tabellenspalte in lesbarer Form angeben wollen, müssen Sie den Datentyp CHARACTER mit ausreichender Längenangabe verwenden (siehe [Abschnitt „Lesbare Darstellung der Daten in Ein- und Ausgabedatei“ auf Seite 27](#)).

Wenn die Eingabewerte und die zugehörige Spalte der Tabelle unterschiedliche Datentypen (CHARACTER (VARYING) oder NATIONAL CHARACTER (VARYING)) haben, dann gelten wegen der Datenkonvertierung folgende Randbedingungen:

- Bei einer Konvertierung von CHARACTER (VARYING) nach NATIONAL CHARACTER (VARYING) wird dem Eingabewert ein EBCDIC-Code gemäß dem CCSN der Eingabedatei zugeordnet. Der Eingabedatei darf **nicht** CCSN=\*NONE zugeordnet sein.
- Bei einer Konvertierung von NATIONAL CHARACTER (VARYING) nach CHARACTER (VARYING) wird dem Spaltenwert ein EBCDIC-Code gemäß dem CCSN der Datenbank zugeordnet. Der Datenbank darf **nicht** CODE\_TABLE \_NONE\_ zugeordnet sein, siehe Klausel CODE\_TABLE auf [Seite 108](#).
- Wenn Eingabewert und zugehörige Spalte der Tabelle den Datentyp CHARACTER (VARYING) haben und für die Datenbank ein codierter Zeichensatz verwendet wird, dann müssen der CCSN der Eingabedatei und der CCSN der Datenbank gleich sein.
- Bei einer Konvertierung von (NATIONAL) CHARACTER (VARYING) in einen numerischen Datentyp oder in einen Zeitdatentyp darf der Eingabedatei der CCSN=\*NONE zugeordnet sein. Auch der Datenbank darf CODE\_TABLE \_NONE\_ zugeordnet sein.

*datentyp* nicht angegeben:

Die Daten müssen gemäß dem Datentyp der zugehörigen *ladespalte* dargestellt sein, siehe [Tabelle 4 auf Seite 34](#). Wenn für eine *ladespalte* vom Datentyp (NATIONAL) CHARACTER VARYING in der zugeordneten Ladedarstellung kein Datentyp oder der Datentyp (NATIONAL) CHARACTER VARYING angegeben ist, so erwartet SESAM/SQL ein 2 Byte langes Längenfeld unmittelbar vor dem einzulesenden Wert.



Das Längenfeld enthält bei NATIONAL CHARACTER VARYING die Länge in Bytes, nicht in Zeichen.

WHEN *nullbedingung* THEN NULL

Angabe einer Nullbedingung.

Wenn die Nullbedingung erfüllt ist, erhält die Tabelle in der betreffenden Spalte an Stelle eines Spaltenwerts den NULL-Wert. Für Spalten, die als NOT NULL definiert sind, dürfen Sie keine WHEN...THEN NULL-Klausel angeben.

*nullbedingung* ::=

*spalte*

Name der Spalte, deren Werte mit dem Literal *literal* verglichen werden.

*spalte* muss mit einer *ladespalte* (siehe [Seite 150](#)) übereinstimmen. Für *spalte* dürfen Sie keine multiple Spalte angeben.

*vergleichs\_op* ::=  $\left. \begin{array}{l} = \\ < \\ > \\ <= \\ >= \\ <> \end{array} \right\}$

Es gelten die für den Vergleich von zwei Werten üblichen Regeln (siehe Handbuch „[SQL-Sprachbeschreibung Teil 1: SQL-Anweisungen](#)“).

*literal*

Literal.

Der Datentyp von *literal* muss mit dem Datentyp von *ladedarstellung* verträglich sein.

POSITION (*zahl*)

Positive ganze Zahl, die die Position des ersten Vergleichsoperanden in der Eingabedatei angibt.

POSITION (\*)

Als erster Vergleichsoperand der Nullbedingung dient der durch die zugehörige Ladedarstellung festgelegte Wert der Eingabedatei.

Falls sich die Ladedarstellung auf mehrere Ausprägungen einer multiplen Spalte bezieht, müssen Sie zur Bestimmung des ersten Vergleichsoperanden der Nullbedingung POSITION (\*) angeben. SESAM/SQL vergleicht dann nacheinander die einzelnen Eingabewerte für die multiple Spalte mit dem nachfolgenden Literal und vergibt ggf. für die zugehörige Ausprägung den NULL-Wert.

$\left\{ \begin{array}{l} \textit{alphanumerisches\_literal} \\ \textit{national\_literal} \end{array} \right\}$

Alphanumerisches Literal (z.B. '0' oder X'F0') oder

National-Literal (z.B. N'0' oder NX'0030' oder U&'0' oder U&'0030').

Die Vergleichslänge wird durch die OCTET\_LENGTH des Literals bestimmt (siehe Handbuch „[SQL-Sprachbeschreibung Teil 1: SQL-Anweisungen](#)“).

### SKIP FIRST *zahl* RECORDS

*zahl* steht für eine positive ganze Zahl.

Die ersten *zahl* Sätze der Eingabedatei werden nicht gelesen.

Dies können z.B. Kommentare, die Überschriftszeile (siehe UNLOAD, [Seite 248](#)) oder bereits eingelesene Sätze (abgebrochener LOAD, siehe [Seite 158](#)) sein.

Bei *zahl*=0 (Standardwert) werden alle Sätze der Eingabedatei gelesen. *zahl* muss kleiner oder gleich der Anzahl der Sätze in der Eingabedatei sein.

Bei Angabe der Klausel TRANSFER\_FORMAT werden alle Beschreibungssätze (am Dateianfang) gelesen und interpretiert. Danach kommen ggf. die zu überlesenden Sätze, dann die Datensätze. D.h. *zahl* muss den Wert 0 haben oder mindestens gleich der Anzahl der Beschreibungssätze sein.

Überlesene Sätze werden mitgezählt. Die Nummer eines Satzes in der Fehlerdatei bezieht sich also stets auf die vollständige Eingabedatei.

#### *Hinweise zur Anwendung dieser Klausel im Fehlerfall*

Wenn ein LOAD ONLINE mit Fehler (ungleich einem Consistency-Check) abgebrochen wird, dann erhalten Sie in der LOAD-Fehlerdatei (siehe [Seite 158](#)) eine Meldung über die Anzahl der bearbeiteten Sätze bei Abbruch des Ladevorganges:

*n* INPUT RECORDS PROCESSED.

Diese Meldung dient zur Fehlerfindung und auch dazu, den abgebrochenen Ladevorgang ohne Änderung der Eingabedatei mit SKIP FIRST *n* RECORDS zu Ende zu führen. *n* bezieht sich dabei stets auf die vollständige Eingabedatei, beinhaltet also auch die überlesenen Sätze.

Auch wenn ein LOAD OFFLINE mit Fehler abgebrochen wird, erhalten Sie die Meldung *n* INPUT RECORDS PROCESSED in der LOAD-Fehlerdatei.

- Für *n*=0 wurde kein Satz zugeladen, der LOAD muss komplett wiederholt werden. Falls die Tabelle im Zustand „load running“ ist, muss der Space zuvor mit RECOVER SPACE wieder hergestellt werden.
- Bei *n*>0 wurden alle Primärdaten vollständig zugeladen. Eventuell müssen Sie noch die Indizes neu aufbauen (RECOVER INDEX ON TABLE) oder die Integritätsbedingungen prüfen (CHECK CONSTRAINTS ON TABLE).

### INTO TABLE *tabelle*

Basistabelle, in die geladen werden soll.

Bei LOAD ONLINE darf die angegebene Tabelle keine Nur-CALL-DML-Tabelle sein.

Bei einer partitionierten Tabelle müssen alle betroffenen Partitionen zur Verfügung stehen, siehe [Tabelle 3 auf Seite 22](#).

(*ladespalte*,... )

Liste der Spalten, in die geladen werden soll.

Die Liste der Ladespalten gibt an, in welche Spalten der Tabelle *tabelle* geladen werden soll bzw. in welche Ausprägungen einer multiplen Spalte der Tabelle *tabelle* geladen werden soll.

Es gelten die folgenden Regeln:

- Die Liste der Ladespalten darf nicht mehrfach dieselbe *ladespalte* enthalten.
- Die Liste der Ladespalten darf nicht angegeben werden, wenn TRANSFER\_FORMAT angegeben ist.
- Falls die Eingabedatei nicht im Transferformat vorliegt (Klausel TRANSFER\_FORMAT nicht angegeben) gilt:  
Ist weder die Liste der Ladespalten noch die Liste der Ladedarstellungen angegeben, so müssen für alle Spalten der Tabelle *tabelle* Werte in der Eingabedatei vorhanden sein. LOAD erwartet die Eingabewerte in der Reihenfolge und mit den Formaten gemäß den Spaltendefinitionen für die Tabelle *tabelle*.
- Ist nur die Liste der Ladespalten angegeben, nicht jedoch die Liste der Ladedarstellungen, so müssen die Eingabewerte im Format vorliegen, das für die jeweils zugehörige *ladespalte* definiert ist, siehe [Tabelle 4 auf Seite 34](#) (Ausnahme: DELIMITER\_FORMAT).
- Ist keine *ladespalte* angegeben und enthält die Tabelle *tabelle* eine multiple Spalte, so erwartet LOAD die maximale Anzahl Ausprägungen.
- Ist sowohl die Liste der Ladedarstellungen als auch die Liste der Ladespalten angegeben, so ordnet SESAM/SQL jedem Element aus der Liste der Ladespalten eine Ladedarstellung zu. Somit müssen die Liste der Ladedarstellungen und die Liste der Ladespalten gleich viele Elemente enthalten.
- Wird eine multiple Spalte nur über den Spaltennamen angesprochen und nicht über eine einzelne Ausprägung bzw. einen Ausprägungsbereich, so erwartet LOAD die maximale Anzahl Ausprägungen.  
Eine multiple Spalte nur über den Spaltennamen anzusprechen, empfiehlt sich vor allem dann, wenn die Anzahl der belegten Ausprägungen in den einzelnen Eingabesätzen verschieden ist. Nicht belegte Ausprägungen in der Eingabedatei sind durch die Angabe „WHEN *nullbedingung* THEN NULL“ wie NULL-Werte zu kennzeichnen.
- Ist die Tabelle *tabelle* mit einfachem oder zusammengesetztem Primärschlüssel definiert, so darf entweder keine *ladespalte* angegeben sein oder die Liste der Ladespalten muss den Primärschlüssel enthalten.

*ladespalte ::=*

*spalte*

Name der Spalte, in die geladen werden soll.

Wenn *spalte* eine multiple Spalte ist, gilt im Folgenden:

$sp_{anz}$  bezeichnet die Anzahl der bereits vorhandenen Ausprägungen für die Spalte *spalte*.

$sp_{max}$  bezeichnet die maximal mögliche Anzahl an Ausprägungen für die Spalte *spalte*.

*spalte (posnr)*

*posnr* ist eine positive ganze Zahl.

Wenn *posnr* größer ist als  $sp_{anz} + 1$ , so wird das  $(sp_{anz} + 1)$ -te Element der multiplen Spalte *spalte* geladen.

Andernfalls wird das *posnr*-te Spaltenelement überschrieben, falls OVERWRITE angegeben ist. Gilt NO OVERWRITE, so erfolgt ein Eintrag in die Fehlerdatei (siehe Klausel USING FILE auf [Seite 158](#)).

Ist *spalte* keine multiple Spalte oder *posnr* größer als  $sp_{max}$ , so erfolgt eine Fehlermeldung.

*spalte (min .. max)*

Der Wert ist das Aggregat aus den Spaltenelementen *min* bis *max* der multiplen Spalte *spalte*.

*min* nicht größer ist als  $sp_{anz} + 1$ : Es wird geladen in die Spaltenelemente *min* bis *max*.

*min* größer ist als  $sp_{anz} + 1$ : Es wird geladen in die Spaltenelemente  $sp_{anz} + 1$  bis  $sp_{anz} + 1 + max - min$ .

Ist *spalte* keine multiple Spalte, *min* größer als *max* oder *max* größer als  $sp_{max}$ , so erfolgt eine Fehlermeldung.

*posnr* bzw. *min .. max* nicht angegeben:

Ist *spalte* eine multiple Spalte, so ist die Angabe *spalte* gleichbedeutend mit der Angabe *spalte*(1 ..  $sp_{max}$ ).

*spalte*, *spalte(posnr)* bzw. *spalte (min .. max)* darf nicht mehrfach mit sich überlappenden Ausprägungen als *ladespalte* angegeben werden. Werden bei LOAD ONLINE mehrere Teilbereiche einer multiplen Spalte angegeben, dann darf der daraus gebildete Ausprägungsbereich darüber hinaus auch keine Lücken enthalten.

*ladeparameter ::=*

Jeden der im Folgenden angegebenen Ladeparameter dürfen Sie höchstens einmal angeben:

**[NO] OVERWRITE**

OVERWRITE legt fest, dass Spaltenwerte in Sätzen der Tabelle *tabelle* überschrieben werden sollen durch Werte aus Eingabesätzen, die denselben Primärschlüsselwert enthalten.

OVERWRITE bewirkt, dass NULL-Werte im Eingabesatz die zugeordneten Spaltenwerte löschen.

Die Angabe OVERWRITE ist nicht erlaubt, wenn die Tabelle, in die geladen wird, ohne Primärschlüssel definiert ist.

Bei Verwendung eines Zählfeldes (siehe Klausel COUNTING\_FIELD auf [Seite 160](#)) wird OVERWRITE ignoriert.

Bei NO OVERWRITE verfährt SESAM/SQL wie folgt:

Wo in der Tabelle *tabelle* noch keine Spaltenwerte vorhanden sind, trägt SESAM/SQL die zugehörigen Eingabewerte ein. Wo bereits Spaltenwerte vorhanden sind, lehnt SESAM/SQL die zugehörigen Eingabewerte ab und schreibt die abgelehnten Eingabewerte in die Fehlerdatei (siehe Klausel USING FILE auf [Seite 158](#)).

Die Anzahl zugeladener Sätze in die Tabelle *tabelle* wird im Kommunikationsbereich des ESQL-COBOL-Programms im Feld SQLrowcount zurückgegeben.

Dabei werden nur die eingefügten Sätze gezählt, geänderte Sätze werden nicht berücksichtigt.



Mit [NO] OVERWRITE wird nicht festgelegt, ob der LOAD Sätze ändern darf. Sätze mit identischem Primärschlüsselwert werden grundsätzlich geändert. Die Klausel gibt nur an, ob auch signifikante Spaltenwerte überschrieben werden dürfen.

{ NO INDEX  
GENERATE INDEX }

Bei LOAD OFFLINE wird mit der INDEX-Klausel festgelegt, wie die in der Tabelle *tabelle* definierten Indizes behandelt werden sollen. Die Angabe dieser Klausel hat bei Tabellen ohne Index keine Auswirkung.

- Wird NO INDEX angegeben, so werden die betroffenen Indizes nicht neu generiert und sind somit nach dem LOAD ungültig.  
Die Angabe NO INDEX in Verbindung mit der Angabe CONSTRAINT CHECK ist im Allgemeinen nicht sinnvoll, da die Prüfung der Integritätsbedingungen intakte Indizes voraussetzt.

- Wird GENERATE INDEX angegeben, so werden alle Indizes nach dem Laden neu generiert. Bei partitionierten Tabellen müssen dazu alle Partitionen verfügbar sein, siehe [Tabelle 3 auf Seite 22](#). Nicht verfügbare Partitionen werden nicht in der Fehlerdatei protokolliert.

Bei LOAD ONLINE darf die INDEX-Klausel nicht angegeben werden. Die durch das Zuladen betroffenen Indizes der Tabelle werden grundsätzlich angepasst. Ungültige Indizes werden ignoriert und bleiben nach dem LOAD ungültig.

#### [NO] CONSTRAINT CHECK

Mit der CHECK-Klausel wird festgelegt, wie die Integritätsbedingungen, die sich auf die Tabelle *tabelle* beziehen, behandelt werden sollen.

Wird CONSTRAINT CHECK angegeben, so werden alle Integritätsbedingungen geprüft. Bei einer partitionierten Tabellen müssen dazu alle Partitionen verfügbar sein, siehe [Tabelle 3 auf Seite 22](#). Falls eine Partition nicht verfügbar ist, dann bleibt die Tabelle im Zustand „check pending“. Nicht verfügbare Partitionen werden nicht in der Fehlerdatei protokolliert.

War die Tabelle vor dem LOAD im Zustand „check pending“ (siehe [Seite 13](#)) und werden keine Integritätsbedingungs-Verletzungen festgestellt, so wird dieser Zustand wieder aufgehoben.

Wird NO CONSTRAINT CHECK angegeben oder werden bei der Prüfung der Integritätsbedingungen Fehler festgestellt, so bleibt bzw. wird die Tabelle gesperrt, sofern Integritätsbedingungen definiert sind. Sie ist danach im Zustand „check pending“. Ein weiterer LOAD bzw. Reparaturmaßnahmen sind jedoch möglich.

Die Prüfung von Integritätsbedingungen erfolgt bei OFFLINE, nachdem alle Sätze in die Tabelle eingetragen wurden. Die Prüfung erfolgt bei ONLINE in vielen, aber nicht allen Fällen beim Einfügen eines jeden Satzes in die Tabelle. Der Ablauf bei ONLINE ist in diesen Fällen wesentlich schneller als der Ablauf bei OFFLINE.

Die Angabe einer CHECK-Klausel ist bei Tabellen ohne Integritätsbedingungen wirkungslos. Voreinstellung bei LOAD OFFLINE ist NO CONSTRAINT CHECK; Voreinstellung bei LOAD ONLINE ist CONSTRAINT CHECK.

### TRANSFER\_FORMAT

Die Daten werden aus der mit der UNLOAD-Anweisung erstellten Transferdatei geladen.



Wenn die Transferdatei Daten der Datentypen NATIONAL CHARACTER (VARYING) enthält, dann kann sie nicht mehr mit SESAM/SQL bis V4.0 geladen werden.

Wenn die Datei zuvor mit UNLOAD ... TRANSFER\_FORMAT aus einer Basistabelle mit Primärschlüssel erstellt wurde, dann sind die Sätze nach den Werten dieses Primärschlüssels sortiert. Stimmt der Primärschlüssel der Tabelle *tabelle* mit dem Primärschlüssel der Tabelle überein, aus der die Transferdatei erstellt wurde, dann lässt sich der Zuladevorgang bei LOAD OFFLINE durch Angabe von SORTED beschleunigen.

Bei der Angabe von TRANSFER\_FORMAT ... dürfen keine *ladedarstellung* und keine *ladespalte* angegeben werden. Bei der Angabe von TRANSFER\_FORMAT ... dürfen Spalten, in die Daten geladen werden sollen, *nicht* das Attributformat OLDEST besitzen. Dieses Attributformat darf ausschließlich in Nur-Call-DML-Tabellen vorkommen.

### UNLOAD\_FORMAT

UNLOAD\_FORMAT fordert, dass jedem Wert der Eingabedatei ein Indikator vom Datentyp SMALLINT vorangestellt ist (sogenanntes UNLOAD\_FORMAT). Dieser Indikator zeigt an, ob der Eingabewert als NULL-Wert zu interpretieren ist. LOAD interpretiert einen Eingabewert als NULL-Wert, wenn der Indikator einen negativen Wert hat.

Eine Eingabedatei im UNLOAD\_FORMAT wird mit der Utility-Anweisung UNLOAD, Klausel LOAD\_FORMAT erstellt. Die Sätze einer Datei im UNLOAD\_FORMAT sind nach den Primärschlüsselwerten der Basistabelle sortiert, aus der mit UNLOAD ... LOAD\_FORMAT entladen wurde. Stimmt der Primärschlüssel der Tabelle *tabelle* mit dem Primärschlüssel der Tabelle, aus der die Eingabedatei im UNLOAD\_FORMAT erstellt wurde, überein, dann lässt sich der Zuladevorgang bei LOAD OFFLINE durch Angabe von SORTED beschleunigen.

Die Tabelle, in die geladen wird, muss die gleichen Spalten im identischen Format enthalten, wie die Tabelle, aus der zuvor mit UNLOAD ... LOAD\_FORMAT in die Eingabedatei für LOAD geschrieben wurde. Bei der Angabe von UNLOAD\_FORMAT ... darf keine *ladedarstellung* angegeben werden.

**DELIMITER\_FORMAT TERMINATED BY *delimiter***

SESAM/SQL erwartet die Eingabedatei im sogenannten Delimiter-Format. Im Delimiter-Format sind alle Werte im codierten Zeichensatz der Eingabedatei dargestellt und mit einem DELIMITER-Zeichen abgeschlossen. Genaueres zur Darstellung der Werte im Delimiter-Format siehe [Abschnitt „Lesbare Darstellung der Daten in Ein- und Ausgabedatei“ auf Seite 27](#).

Bei der Angabe von DELIMITER\_FORMAT ... darf keine *ladedarstellung* angegeben werden. Bei der Angabe von DELIMITER\_FORMAT ... dürfen keine Spalten, in die Daten geladen werden sollen, das „Attributformat OLDEST“ besitzen, das ausschließlich in Nur-Call-DML-Tabellen vorkommen kann.

Der CCSN der Eingabedatei wird geprüft:

- Wenn der Eingabedatei ein EBCDIC-Zeichensatz oder ein selbst definierter Zeichensatz auf EBCDIC-Basis bzw. kein Zeichensatz (CCSN=\*NONE) zugeordnet ist, dann werden alle Eingabewerte als Werte vom Datentyp CHARACTER interpretiert und nötigenfalls in das Format der Tabellenspalte konvertiert.
- Wenn der Eingabedatei der CCSN=UTFE zugeordnet ist, dann werden alle Eingabewerte gemäß UTFE interpretiert und in das Format der Tabellenspalte konvertiert.

Behandlung von NULL-Werten:

- Der NULL-Wert wird in der Eingabedatei nicht dargestellt. Das den fehlenden Wert abschließende (nicht-entwertete) DELIMITER-Zeichen wird jedoch dargestellt. Zwei aufeinanderfolgende DELIMITER-Zeichen innerhalb und am Ende eines Datensatzes werden also wie folgt interpretiert: das erste DELIMITER-Zeichen schließt den vorausgehenden Wert ab, das zweite DELIMITER-Zeichen schließt einen nicht dargestellten Nullwert ab.
- Wenn der letzte Wert eines Datensatzes kein NULL-Wert ist, kann das abschließende DELIMITER-Zeichen weggelassen werden.

***delimiter***

Definition des DELIMITER-Zeichens, das in der Eingabedatei jeden Wert abschließt. Als DELIMITER-Zeichen darf kein Wert gewählt werden, der Bestandteil der Eingabewerte ist.

Für die Angabe von *delimiter* gibt es folgende Möglichkeiten:

- der Eingabedatei ist ein EBCDIC-Zeichensatz oder einen selbst definierter Zeichensatz auf EBCDIC-Basis bzw. kein Zeichensatz (CCSN=\*NONE) zugeordnet:

alphanumerisches Literal der Länge 1.

Beispiele: TERMINATED BY ';' oder TERMINATED BY X'5E'.

Aus Kompatibilitätsgründen kann auch die sedezimale Darstellung angegeben werden: 'X"*xx*' (x: eine Stelle der sedezimalen Darstellung).

- die Eingabedatei hat CCSN=UTFE:

National-Literal der Länge 1, das auch in UTFE eine sedezimale Darstellung in einem Byte hat (Wertebereich NX '0000' ... NX '009F').

Beispiele: TERMINATED BY N';' oder TERMINATED BY NX'003B'  
oder auch: TERMINATED BY U&';' oder TERMINATED BY U&'003B'

Soll als DELIMITER-Zeichen ein einfaches Hochkomma (') definiert werden, so müssen Sie vier einfache Hochkommata angeben.

Beispiele: TERMINATED BY ""' oder ... N""' oder ...U&""'.

CSV\_FORMAT DELIMITER *delimiter* [QUOTE *quote*] [ESCAPE *escape*]

SESAM/SQL erwartet die Eingabedatei im sogenannten CSV-Format. Im CSV-Format sind alle Werte im codierten Zeichensatz der Eingabedatei dargestellt und mit einem nicht-entwerteten DELIMITER-Zeichen oder mit einem Zeilenende-Zeichen abgeschlossen. Genaueres zur Darstellung der Werte im CSV-Format siehe [Abschnitt „Lesbare Darstellung der Daten in Ein- und Ausgabedatei“ auf Seite 27](#).

Bei der Angabe von CSV\_FORMAT darf keine *ladedarstellung* angegeben werden. Es dürfen auch keine Spalten, in die Daten geladen werden sollen, das „Attributformat OLDEST“ besitzen, das ausschließlich in Nur-Call-DML-Tabellen vorkommen kann.

Der CCSN der Eingabedatei wird geprüft:

- Wenn der Eingabedatei ein EBCDIC-Zeichensatz oder ein selbst definierter Zeichensatz auf EBCDIC-Basis bzw. kein Zeichensatz (CCSN=\*NONE) zugeordnet ist, dann werden alle Eingabewerte als Werte vom Datentyp CHARACTER interpretiert und nötigenfalls in das Format der Tabellenspalte konvertiert.
- Wenn der Eingabedatei der CCSN=UTFE zugeordnet ist, dann werden alle Eingabewerte gemäß UTFE interpretiert und in das Format der Tabellenspalte konvertiert.

Behandlung von NULL-Werten:

- Der NULL-Wert wird in der Eingabedatei nicht dargestellt. Das den fehlenden Wert abschließende (nicht-entwertete) DELIMITER-Zeichen wird jedoch dargestellt. Zwei aufeinanderfolgende DELIMITER-Zeichen innerhalb und am Ende eines Datensatzes werden also wie folgt interpretiert: das erste DELIMITER-Zeichen schließt den vorausgehenden Wert ab, das zweite DELIMITER-Zeichen schließt einen nicht dargestellten Nullwert ab.
- Nach dem letzten signifikanten Wert einer Zeile der Eingabedatei braucht kein DELIMITER-Zeichen mehr angegeben zu werden. Nicht angegebene Werte gelten als NULL-Werte.



*delimiter*, *quote* und *escape* müssen unterschiedliche Zeichen sein.

*delimiter*

Definition des Trennzeichens (DELIMITER-Zeichen), das in der Eingabedatei jeden Wert abschließt. Ein DELIMITER-Zeichen kann auch Bestandteil eines Wertes sein, siehe die nachfolgenden Beschreibungen zu *quote* und *escape*.

Für die Angabe von *delimiter* gibt es folgende Möglichkeiten:

- der Eingabedatei ist ein EBCDIC-Zeichensatz oder einen selbst definierter Zeichensatz auf EBCDIC-Basis bzw. kein Zeichensatz (CCSN=\*NONE) zugeordnet:

alphanumerisches Literal der Länge 1.

Beispiele: DELIMITER ';' oder DELIMITER X'5E'.

Aus Kompatibilitätsgründen kann auch die sedezimale Darstellung angegeben werden: 'X"xx"' (x: eine Stelle der sedezimalen Darstellung).

- die Eingabedatei hat CCSN=UTFE:

National-Literal der Länge 1, das auch in UTFE eine sedezimale Darstellung in einem Byte hat (Wertebereich NX'0000' . . . NX'009F').

Beispiele: DELIMITER N';' oder DELIMITER NX'003B'

oder auch: DELIMITER U&';' oder DELIMITER U&'003B'

Soll als DELIMITER-Zeichen ein einfaches Hochkomma (') definiert werden, so müssen Sie vier einfache Hochkommata angeben.

Beispiele: DELIMITER "" oder ... N"" oder ...U&"".

*quote*

Definition des QUOTE-Zeichens, in das die Werte der Eingabedatei eingeschlossen sein können. QUOTE-Zeichen gehören nicht zum Wert selbst. Ein QUOTE-Zeichen innerhalb eines Wertes muss verdoppelt werden. Wenn ein Wert in QUOTE-Zeichen eingeschlossen ist, dann kann er auch NEWLINE-Zeichen (die nicht als Zeilenwechsel interpretiert werden) oder DELIMITER-Zeichen enthalten. Ein Wert, der lediglich aus einem öffnenden und schließenden QUOTE-Zeichen besteht, wird als Wert der Länge 0 interpretiert.

Für die Angabe von *quote* gelten die obigen Ausführungen zu *delimiter* entsprechend.

*escape*

Definition des Entwertungszeichens (ESCAPE-Zeichen), mit dem ESCAPE-Folgen, bestehend aus zwei Zeichen, in der Eingabedatei beginnen.

Mit ESCAPE-Folgen können DELIMITER-Zeichen, QUOTE-Zeichen und ESCAPE-Zeichen als Teil eines Spaltenwerts geschrieben und NEWLINE-Zeichen als Trenner zwischen zwei Eingabezeilen ignoriert werden.

Für die Angabe von *escape* gelten die obigen Ausführungen zu *delimiter* entsprechend.

USING FILE *datei* [PASSWORD *kennwort*]

Name der Fehlerdatei. *datei* muss als alphanumerisches Literal angegeben werden. Bei LOAD ONLINE darf *datei* keine Banddatei sein.

In die Fehlerdatei werden folgende Fehlerinformationen des Ladevorganges protokolliert:

- Informationen über fehlerhafte Sätze der Eingabedatei. Diese Information umfasst die Satznummer des fehlerhaften Satzes in der Eingabedatei, eventuell den Hinweis in welche Spalte fehlerhaft geladen wurde, sowie einen satzspezifischen SQLSTATE.
- nicht verfügbare Partitionen einer partitionierten Tabelle, die davon betroffenen Satznummern in der Eingabedatei sowie den entsprechenden SQLSTATE.
- Meldung über die Anzahl der bearbeiteten Sätze bei Abbruch eines laufenden Ladevorganges durch einen Fehler (ungleich einem Consistency-Check):

*n* INPUT RECORDS PROCESSED

*n* Nummer des letzten gelesenen Satzes der Eingabedatei, der, falls er nicht fehlerhaft war, zu einer Änderung der Tabelle geführt hat.

Überlesene Sätze werden mitgezählt. Eine Satznummer in der Fehlerdatei bezieht sich also stets auf die vollständige Eingabedatei.

Bei einer Eingabedatei im TRANSFER\_FORMAT (Transferdatei) zählt SESAM/SQL Header- und Beschreibungssätze bei der Ermittlung der Satznummer mit. SESAM/SQL meldet auch bei einer Transferdatei die Satznummer der SAM-Datei als Satznummer.

Somit kann mit dem Dateibearbeiter EDT in der Fehlerdatei direkt auf den Satz positioniert werden, der Informationen zum fehlerhaften Satz der Eingabedatei enthält (siehe Handbuch „[EDT \(BS2000\)](#)“).

Die Fehlerdatei wird von SESAM/SQL nur dann angelegt bzw. verwendet, wenn während des Ladevorganges Fehler auftreten. Eine bereits vorhandene Fehlerdatei wird von LOAD fortgeschrieben.

Die Einträge aus verschiedenen LOAD-Anweisungen sind durch zwei Überschriftenzeilen, die den Namen der Anweisung, Datum, Uhrzeit und den Namen der Eingabedatei protokollieren, getrennt.

Soll die Fehlerdatei nicht in der DBH-Kennung liegen, müssen Vorbereitungen dafür getroffen worden sein, siehe Abschnitt „Datenbankdateien und Jobvariablen auf fremden Benutzerkennungen“ im „[Basishandbuch](#)“.

**PASSWORD** *kennwort*

BS2000-Kennwort für die Fehlerdatei. *kennwort* muss als alphanumerisches Literal angegeben werden.

Für die Angabe von *kennwort* gibt es folgende Möglichkeiten:

- 'C'*zeichenkette*"  
*zeichenkette* enthält vier abdruckbare Zeichen.
- 'X'*hex-zeichenkette*"  
*hex-zeichenkette* enthält acht sedezimale Zeichen.
- 'n'  
*n* bezeichnet eine ganze Zahl von - 2147483648 bis + 2147483647.

**USING FILE** *datei* nicht angegeben:

Bei Bedarf legt LOAD die Fehlerdatei in der BS2000-Benutzerkennung des DBH unter folgendem Standarddateinamen an:

*catalog.space.EXC.L*

*catalog* Name der Datenbank.

*space* Name des Anwender-Space, auf dem die Tabelle liegt, in die geladen werden soll.

Bei einer partitionierten Tabelle wird der Spacename der ersten Partition verwendet. Die Fehlerdatei enthält die Fehlerinformationen für alle Partitionen der Tabelle.

**EXC.L** Standard-Endung des Namens der Fehlerdatei für LOAD-Anweisungen

Wenn *catalog* in einer DB-Kennung liegt, so wird die Fehlerdatei in der DB-Kennung angelegt, sofern die Vorbereitungen dafür getroffen worden sind, siehe Abschnitt „Datenbankdateien und Jobvariablen auf fremden Benutzerkennungen“ im „[Basishandbuch](#)“. Kann die Fehlerdatei nicht in der DB-Kennung angelegt werden, so wird sie in der DBH-Kennung angelegt.

**COUNTING\_FIELD** *spalte*

Wenn für die Tabelle, in die geladen werden soll, ein einfacher oder zusammengesetzter Primärschlüssel definiert ist, können Sie für das Laden eine Spalte als sogenanntes Zählfeld vereinbaren. Die Spalte für das Zählfeld muss vom Datentyp NUMERIC, DECIMAL, SMALLINT oder INTEGER sein. Dem Zählfeld dürfen keine Werte in der Eingabedatei zugeordnet sein. Der Wert für das Zählfeld wird von SESAM/SQL vergeben. Wenn Sie die Klausel COUNTING\_FIELD angeben, ignoriert SESAM/SQL die Angabe OVERWRITE.

Die Zählung beginnt, indem der höchste bisher in der Tabelle vergebene Wert für das Zählfeld in der letzten Stelle bzw. in der letzten Dezimalstelle um 1 erhöht wird.

Über ein Zählfeld gewährleistet SESAM/SQL die Eindeutigkeit der Primärschlüsselwerte.

Für die Verwendung des Zählfelds gelten die folgenden Regeln:

- Es dürfen nicht mehrere Schlüsselspalten als Zählfeld vereinbart werden.
- Sind allen Schlüsselspalten Werte in der Eingabedatei zugeordnet, dann dürfen Sie kein Zählfeld vereinbaren.
- Sind keiner Schlüsselspalte Werte in der Eingabedatei zugeordnet, dann müssen Sie ein Zählfeld vereinbaren.
- Sind einigen, aber nicht allen Schlüsselspalten Werte in der Eingabedatei zugeordnet, dann können Sie ein Zählfeld vereinbaren.
- Für Schlüsselspalten, die nicht als Zählfeld vereinbart und denen keine Werte in der Eingabedatei zugeordnet sind, muss ein Defaultwert definiert sein. Dieser Defaultwert wird dann zur Bildung des Primärschlüsselwerts herangezogen.
- Bei LOAD ONLINE darf das Zählfeld weder in einer Referenzbedingung noch in einer Check-Bedingung der Tabelle enthalten sein.
- Bei einer partitionierten Tabelle müssen alle Partitionen zur Verfügung stehen (siehe [Tabelle 3 auf Seite 22](#)).

Wenn die Klausel SORTED bei LOAD OFFLINE nicht angegeben wird, dann darf die partitionierte Tabelle keine Spalte vom Typ CHARACTER VARYING mit einer Länge von mehr als 253 Zeichen und keine Spalte vom Typ NATIONAL CHARACTER VARYING mit einer Länge von mehr als 126 Zeichen enthalten.

COUNTING FIELD *spalte* nicht angeben:  
Es wird kein Zählfeld verwendet.

## LOAD-Formate

Die folgende Tabelle enthält eine Übersicht über die LOAD-Formate.

In den Beispielen für LOAD-Anweisungen sind nicht alle möglichen Klauseln berücksichtigt (z.B. OVERWRITE, GENERATE INDEX, CONSTRAINT CHECK, USING FILE .., COUNTING\_FIELD ..).

Format	Eingabedatei erstellt durch	Nullwerte	Anweisung (Beispiel)	Geladene Spalten
TRANSFER_FORMAT	UNLOAD-Anweisung mit TRANSFER_FORMAT	Interne Darstellung (Y/N für alle Spalten an jedem Satz-Ende)	LOAD FILE <i>datei</i> INTO TABLE <i>tabelle</i> TRANSFER_FORMAT	Alle in der Transferdatei definierten Spalten
UNLOAD_FORMAT	UNLOAD-Anweisung mit LOAD_FORMAT	Interne Darstellung (negativer Indikator bei entsprechenden Spalten)	LOAD FILE <i>datei</i> INTO TABLE <i>tabelle</i> UNLOAD_FORMAT	Alle Spalten laut Reihenfolge von <i>tabelle</i> ; Spaltendarstellung <sup>1</sup> gemäß Definition in <i>tabelle</i> ; mit Indikator bei jeder Spalte
			LOAD FILE <i>datei</i> INTO TABLE <i>tabelle</i> ( <i>spalte, spalte, ...</i> ) UNLOAD_FORMAT	Nur die angegebenen Spalten in dieser Reihenfolge; Spaltendarstellung wie oben
DELIMITER_FORMAT	UNLOAD-Anweisung mit DELIMITER_FORMAT bzw. Editor	Nur DELIMITER-Zeichen	LOAD FILE <i>datei</i> INTO TABLE <i>tabelle</i> DELIMITER_FORMAT TERMINATED BY <i>delimiter</i>	Alle Spalten laut Reihenfolge von <i>tabelle</i> ; Spaltendarstellung in <i>datei</i> lesbar und konvertierbar in Definition von <i>tabelle</i>
			LOAD FILE <i>datei</i> INTO TABLE <i>tabelle</i> ( <i>spalte, spalte, ...</i> ) DELIMITER_FORMAT TERMINATED BY <i>delimiter</i>	Nur die angegebenen Spalten in dieser Reihenfolge; Spaltendarstellung wie oben

Tabelle 8: LOAD-Ladeformate

(Teil 1 von 2)

Format	Eingabedatei erstellt durch	Nullwerte	Anweisung (Beispiel)	Geladene Spalten
CSV_FORMAT	UNLOAD-Anweisung mit CSV_FORMAT bzw. Editor oder Kalkulationsprogramm	Nur DELIMITER-Zeichen (darf am Zeilende entfallen)	LOAD FILE <i>datei</i> INTO TABLE <i>tabelle</i> CSV_FORMAT DELIMITER <i>delimiter</i> QUOTE <i>quote</i> ESCAPE <i>escape</i>	Alle Spalten laut Reihenfolge von <i>tabelle</i> ; Spaltendarstellung in <i>datei</i> lesbar und konvertierbar in Definition von <i>tabelle</i>
			LOAD FILE <i>datei</i> INTO TABLE <i>tabelle</i> ( <i>spalte, spalte, ...</i> ) CSV_FORMAT DELIMITER <i>delimiter</i>	Nur die angegebenen Spalten in dieser Reihenfolge; Spaltendarstellung wie oben; keine QUOTE- und ESCAPE-Zeichen
Standard-Format	UNLOAD-Anweisung mit Standardformat	NULL-Werte nicht erkennbar!	LOAD FILE <i>datei</i> INTO TABLE <i>tabelle</i>	Alle Spalten laut Reihenfolge von <i>tabelle</i> ; Spaltendarstellung gleich in <i>datei</i> und <i>tabelle</i>
			LOAD FILE <i>datei</i> INTO TABLE <i>tabelle</i> ( <i>spalte, spalte, ...</i> )	Nur die angegebenen Spalten in dieser Reihenfolge; Spaltendarstellung wie oben
Selbst definiertes Format	UNLOAD-Anweisung mit Standardformat oder selbst definiertem Format bzw. Editor	Falls <i>nullbedingung</i> zutrifft	LOAD FILE <i>datei</i> ( <i>ladedarstellung, ladedarstellung</i> WHEN <i>nullbedingung</i> THEN NULL,...) INTO TABLE <i>tabelle</i>	Alle Spalten laut Reihenfolge von <i>tabelle</i> ; Spaltendarstellung laut <i>ladedarstellung</i> und konvertierbar in Definition von <i>tabelle</i>
			LOAD FILE <i>datei</i> ( <i>ladedarstellung, ladedarstellung</i> WHEN <i>nullbedingung</i> THEN NULL,...) INTO TABLE <i>tabelle</i> ( <i>spalte, spalte, ...</i> )	Nur die angegebenen Spalten in dieser Reihenfolge; Spaltendarstellung wie oben

Tabelle 8: LOAD-Ladeformate

(Teil 2 von 2)

<sup>1</sup> Darstellung des Wertes der Spalte in der Ladedatei

## Beispiele

Die Beispiele laden Daten aus einer Eingabedatei in die Tabelle KUNDE. Die Tabelle KUNDE gehört zum Schema AUFTRAGSVER der Datenbank AUFTRAGKUNDEN.

1. Die Daten stehen in der Eingabedatei LOAD.D.KUNDE. Die Werte sind durch das DELIMITER-Zeichen „;“ voneinander getrennt.



```
LOAD OFFLINE SORTED FILE 'LOAD.D.KUNDE'
INTO TABLE auftragkunden.auftragsver.kunde
OVERWRITE GENERATE INDEX CONSTRAINT CHECK
DELIMITER_FORMAT TERMINATED BY ';' ;'
```

Es wird aus der nach Primärschlüsselwerten sortierten Eingabedatei LOAD.D.KUNDE im OFFLINE-Modus geladen. Spaltenwerte in Sätzen der Tabelle KUNDE werden überschrieben durch Werte in Eingabesätzen, die denselben Primärschlüsselwert enthalten. Für die Tabelle KUNDE definierte Indizes werden neu erstellt. Die für die Tabelle KUNDE definierten Integritätsbedingungen werden geprüft. Die Anzahl zugeladener Sätze wird im Feld SQLrowcount der Diagnostic Area zurückgemeldet.

Nach dem Laden befindet sich die Tabelle KUNDE im Zustand „copy pending“ und, falls Integritätsbedingungen verletzt sind, außerdem im Zustand „check pending“.

Die Eingabedatei LOAD.D.KUNDE hat folgenden Inhalt:

```
100;Siemens AG;Otto-Hahn-Ring 6;81739;Muenchen;D;089/636-8;Elektro;
101;Login GmbH;Rosenheimer Str. 34;81667;Muenchen;D;089/4488870;PC Netzwerke;
102;JIKO GmbH;Posener Str. 12;30659;Hannover;D;0551/123874;Import-Export;
103;Plenzer Trading;Paul-Heyse-Str. 12;80336;Muenchen;D;089/923764;Fruechtehandel;
104;Jonas Fischladen;Hirschgartenstr. 12;12587;Berlin;D;016/5739921;Einzelhandel;
105;Pudelshop Anke;Am Muehlentor 26;41179;Moenchengladbach;D;040/873562;Dienstleistung;
106;Foreign Ltd.;26 West York St.;New York, NY;USA;001703/2386532;Handelsvertretung;
107;Externa & Co KG;Bernner Weg 78;3000;Bern 33;CH;;Anwaltskanzlei;
```

- Die Daten stehen lesbar und formatiert in der Eingabedatei LOAD.F.KUNDE. In der Ladedarstellung der LOAD-Anweisung wird den lesbar dargestellten Eingabewerten der Datentyp CHARACTER zugewiesen.

```
LOAD ONLINE SORTED FILE 'LOAD.F.KUNDE'  
  (POSITION(1) INTEGER,  
   POSITION(5) CHARACTER(40),  
   POSITION(45) CHARACTER(40),  
   POSITION(85) CHARACTER(5) WHEN PLZ='00000' THEN NULL,  
   POSITION(90) CHARACTER(40),  
   POSITION(130) CHARACTER(3),  
   POSITION(133) CHARACTER(25),  
   POSITION(158) CHARACTER(50))  
INTO TABLE auftragkunden.auftragsver.kunde  
(knr, firma, strasse, plz, ort, land, ktelefon, kinfo)  
OVERWRITE CONSTRAINT CHECK
```

Es wird aus der nach Primärschlüsselwerten sortierten Eingabedatei LOAD.F.KUNDE im ONLINE-Modus geladen. Für die Position 85 der Eingabedatei, die der Spalte PLZ der Tabelle KUNDE zugeordnet ist, ist eine WHEN...THEN NULL-Klausel definiert: Wenn auf den Positionen 85 bis 89 der Eingabedatei der Wert 00000 steht, wird in der Tabelle KUNDE an entsprechender Stelle der NULL-Wert eingetragen.

Spaltenwerte in Sätzen der Tabelle KUNDE werden überschrieben durch Werte in Eingabesätzen, die denselben Primärschlüsselwert enthalten. Die für die Tabelle KUNDE definierten Integritätsbedingungen werden geprüft.

Vorteil von LOAD ONLINE:

- die Indizes werden aktualisiert
- das Logging wird nicht unterbrochen
- DML-Anwendungen können parallel dazu auf die Basistabelle zugreifen

Wenn Integritätsbedingungen verletzt sind, befindet sich die Tabelle KUNDE nach dem Laden im Zustand „check pending“.

Die Eingabedatei LOAD.F.KUNDE hat folgenden Inhalt:

____Siemens AG		Otto-Hahn-Ring 6	
81739Muenchen	D	089/636-8	Elektro
____Login GmbH		Rosenheimer Str.34	
81667Muenchen	D	089/4488870	PC Netzwerke
____JIKO GmbH		Posener Str. 12	
30659Hannover	D	0551/123874	Import-Export
____Plenzer Trading		Paul-Heyse-Str. 12	
80336Muenchen	D	089/923764	Fruechtehandel
____Jonas Fischladen		Hirschgartenstr. 12	
12587Berlin	D	016/5739921	Einzelhandel
____Pudelshop Anke		Am Muehlentor 26	
41179Moenchengladbach	D	040/873562	Dienstleistung
____Foreign Ltd.		26 West York St.	
00000New York, NY	USA001703/2386532		Handelsvertretung
____Externa & Co KG		Berner Weg 78	
03000Bern 33	CH		Anwaltskanzlei

Die Primärschlüssel sind vom Typ INTEGER und müssen in der Eingabedatei binär eingetragen sein. Im obigen Beispiel wird dies durch die Zeichenfolge „\_\_\_\_“ dargestellt.

### Siehe auch

UNLOAD, EXPORT/IMPORT TABLE

## MIGRATE - Datenbanken und Tabellen umstellen

MIGRATE ermöglicht die Migration, d.h.

- die Umstellung einer SESAM/SQL-Datenbank der Version 1.1 oder einer früheren Version in eine Basistabelle der aktuellen SESAM/SQL-Version (im Folgenden SESAM/SQL-Server-Tabelle genannt)
- die Umstellung einer CALL-DML/SQL-Tabelle in eine SQL-Tabelle
- die Umstellung einer Nur-CALL-DML-Tabelle in eine CALL-DML/SQL-Tabelle.

Die Ausführung der Utility-Anweisung MIGRATE unterbricht das Logging für den betroffenen Anwender-Space. Somit können bei einer Reparatur des Anwender-Space Änderungen nur bis zum Stand unmittelbar vor der Migration nachgefahren werden.

Sofern für den von MIGRATE betroffenen Anwender-Space das Logging eingeschaltet ist, muss der Datenbankverwalter nach Ausführung von MIGRATE einen SESAM-Sicherungsbestand dieses Anwender-Space mit der Utility-Anweisung COPY erstellen. Andernfalls verbleibt der Anwender-Space im Zustand „copy pending“ und kann dann nur mit Utility-Anweisungen bearbeitet werden (siehe [Abschnitt „Space-Zustände und zulässige Utility-Anweisungen“ auf Seite 13](#)).

### Umstellung einer Datenbank der SESAM/SQL V1.1 oder einer früheren Version in eine SESAM/SQL-Server-Tabelle

Ausgangspunkt für die Umstellung einer Datenbank der SESAM/SQL V1.1 oder einer früheren Version (im Folgenden V1-Datenbank genannt) in eine SESAM/SQL-Server-Tabelle bildet eine zuvor mit SESASB erstellte Sicherungsdatei (DB-SIB-Datei) der umzustellenden Datenbank. Ist diese Datenbank mit Kennwortkatalog eingerichtet, wird für die Umstellung außerdem eine zuvor mit dem Dienstprogramm SEPA erstellte Sicherungsdatei des Kennwortkatalogs (PK-SIB-Datei) benötigt.

Die Spaltennamen der umzustellenden V1-Datenbank müssen den Regeln für Spaltennamen einer SESAM/SQL-Server-Tabelle genügen. Andernfalls müssen vor der Umstellung die Spaltennamen mit SESASB entsprechend geändert werden.

Der aktuelle Berechtigungsschlüssel muss das Sonder-Privileg UTILITY besitzen oder eine der im Folgenden genannten Bedingungen erfüllen:

- Falls die Klausel USING SPACE nicht angegeben ist und der Default-Space *D0berechtigungsschlüssel* des Eigentümers des Schemas, dem die SESAM/SQL-Server-Tabelle zugeordnet werden soll, nicht existiert:  
Der aktuelle Berechtigungsschlüssel muss Eigentümer des Schemas sein, dem die SESAM/SQL-Server-Tabelle zugeordnet werden soll. Außerdem muss der aktuelle Berechtigungsschlüssel das Sonder-Privileg USAGE für die Default Storage Group D0STOGROUP besitzen.
- Falls die Klausel USING SPACE nicht angegeben ist, aber der Default-Space *D0berechtigungsschlüssel* des Eigentümers des Schemas, dem die SESAM/SQL-Server-Tabelle zugeordnet werden soll, existiert:  
Der aktuelle Berechtigungsschlüssel muss Eigentümer des Schemas sein, dem die SESAM/SQL-Server-Tabelle zugeordnet werden soll.
- Falls die Klausel USING SPACE angegeben ist:  
Der aktuelle Berechtigungsschlüssel muss Eigentümer des Schemas sein, dem die SESAM/SQL-Server-Tabelle zugeordnet werden soll. Außerdem muss der Eigentümer dieses Schemas mit dem Eigentümer des in der Klausel USING SPACE angegebenen Anwender-Space identisch sein.

---

```
MIGRATE vIdbsib [PASSWORD_CATALOG vIpksib] [PASSWORD kenwort]
```

```
[ { WITH INDEX } ] TO [CALL DML] TABLE tabelle
```

```
[USING SPACE space]
```

---

*vIdbsib*

Name der DB-SIB-Datei für die umzustellende Datenbank der Version V1.1 oder einer früheren Version. *vIsib* muss als alphanumerisches Literal angegeben werden.

PASSWORD\_CATALOG *vIpksib*

Name der PK-SIB-Datei für den Kennwortkatalog der umzustellenden V1-Datenbank. *vIpksib* muss als alphanumerisches Literal angegeben werden.

PASSWORD\_CATALOG *vIpksib* muss angegeben werden, wenn eine Datenbank mit Kennwortschutz in eine CALL-DML-Tabelle umgestellt werden soll (Angabe: TO CALL DML TABLE, siehe [Seite 169](#)).

In den folgenden Fällen darf `PASSWORD_CATALOG v1pksib` nicht angegeben werden:

- Die Datenbank soll in eine SQL-Tabelle umgestellt werden (Angabe: `TO TABLE`, siehe [Seite 169](#)).
- Die umzustellende Datenbank ist ohne Kennwortkatalog definiert.
- Der Kennwortkatalog ist zerstört.

#### `PASSWORD` *kennwort*

BS2000-Kennwort für die DB-SIB-Datei und, sofern vorhanden, für die PK-SIB-Datei der umzustellenden Datenbank. *kennwort* muss als alphanumerisches Literal angegeben werden.

Für die Angabe von *kennwort* gibt es folgende Möglichkeiten:

- `'C'zeichenkette'`  
*zeichenkette* enthält vier abdruckbare Zeichen.
- `'X'hex-zeichenkette'`  
*hex-zeichenkette* enthält acht sedezimale Zeichen.
- `'n'`  
*n* bezeichnet eine ganze Zahl von - 2147483648 bis + 2147483647.

#### `WITH INDEX` angegeben:

Es werden alle für die umzustellende Datenbank definierten Indizes aufgebaut. Die Namen für diese Indizes werden von SESAM/SQL vergeben. SESAM/SQL legt die Indizes auf den Anwender-Space, der die Basistabelle aufnimmt, in die umgestellt wird.

Wenn nach der Umstellung ein Fehler bei der Indexdefinition bzw. beim Aufbauen des Index auftritt, gibt SESAM/SQL eine Warnung aus. Die migrierten Daten bleiben erhalten und der betroffene Anwender-Space befindet sich in demselben Zustand, als wäre `WITH INDEX` nicht angegeben. Ein Anwender-Space, für den Logging eingeschaltet ist, befindet sich demzufolge im Zustand „copy pending“. In diesem Fall müssen Sie mit der Utility-Anweisung `COPY` einen SESAM-Sicherungsbestand des Anwender-Space erstellen, bevor Sie die Indizes definieren können.

#### `NO INDEX` angegeben:

Die Indizes der umzustellenden Datenbank werden nicht übernommen.

TO [CALL DML] TABLE *tabelle*

*tabelle*

Name der SESAM/SQL-Server-Tabelle, in die umgestellt werden soll.

Datenbank und Schema, in denen die Tabelle *tabelle* liegen soll, müssen vorhanden sein. *tabelle* muss von allen Tabellennamen und Viewnamen dieses Schemas verschieden sein, d.h. die Tabelle *tabelle* darf noch nicht existieren. *tabelle* kann mit einem Schemanamen qualifiziert werden.

CALL DML angegeben:

Es wird in eine CALL-DML-Tabelle umgestellt.

Für jedes Attribut der umzustellenden Datenbank muss ein Defaultwertzeichen für die Darstellung des nicht-signifikanten Werts definiert sein.

CALL DML nicht angegeben:

Es wird in eine SQL-Tabelle umgestellt.

Die umzustellende Datenbank darf keine „alten Attributformate“ enthalten, d.h. Attributformate der SESAM-Version 13.1 oder einer älteren Version.

USING SPACE *space*

Name des Anwender-Space, auf dem die Tabelle *tabelle* liegen soll. Der Anwender-Space muss bereits für die zugehörige Datenbank definiert sein.

Der einfache Spacename kann mit dem Namen der Datenbank qualifiziert werden. Dieser Datenbankname muss mit dem Datenbanknamen der Tabelle übereinstimmen.

USING SPACE *space* nicht angegeben:

Die Tabelle *tabelle* wird im Default-Space *D0berechtigungsschlüssel* des aktuellen Berechtigungsschlüssels gespeichert. Existiert dieser Default-Space noch nicht, so wird er auf der Default Storage Group *D0STOGROUP* erzeugt.

## Behandlung des Compound Key bei der Umstellung

Bei der Umstellung einer V1-Datenbank mit Compound Key in eine SESAM/SQL-Server-Tabelle übernimmt SESAM/SQL den Namen des Compound Key als Namen für die entsprechende Primärschlüsselbedingung dieser SESAM/SQL-Server-Tabelle.

Um Namenskonflikte bei der Umstellung von zwei oder mehreren V1-Datenbanken mit gleichem Compound Key-Namen in SESAM/SQL-Server-Tabellen desselben Schemas zu vermeiden, vergibt SESAM/SQL ab der Zweiten dieser SESAM/SQL-Server-Tabellen einen neuen eindeutigen Namen für eine Primärschlüsselbedingung, falls der abgeleitete Name schon existiert. Diese Namen kann der Datenbankverwalter über das Informationsschema ermitteln (siehe Handbuch „[SQL-Sprachbeschreibung Teil 1: SQL-Anweisungen](#)“).

Bei zwei V1-Datenbanken mit demselben Compound Key-Namen kann dieser Name nur dann unverändert für die zugehörigen Primärschlüsselbedingungen übernommen werden, wenn die V1-Datenbanken in Tabellen umgestellt werden, die in verschiedenen Schemata liegen.

### Umstellung einer CALL-DML/SQL-Tabelle in eine SQL-Tabelle

Bei der Umstellung einer CALL-DML/SQL-Tabelle in eine SQL-Tabelle entfernt SESAM/SQL alle nicht-signifikanten Werte und löscht den Kennwortkatalog. Somit ist der Zugriff auf diese Tabelle mit CALL-DML-Anweisungen nicht mehr möglich.

Der aktuelle Berechtigungsschlüssel muss das Sonder-Privileg UTILITY besitzen oder Eigentümer des Schemas sein, dem die umzustellende Tabelle zugeordnet ist.

Bei der Umstellung einer partitionierten CALL-DML/SQL-Tabelle in eine SQL-Tabelle müssen alle Partitionen verfügbar sein (siehe [Tabelle 3 auf Seite 22](#)).

---

```
MIGRATE [CALL DML] TABLE tabelle
```

---

*tabelle*

Name der CALL-DML/SQL-Tabelle, die in eine SQL-Tabelle umgestellt werden soll.

### Umstellung einer Nur-CALL-DML-Tabelle in eine CALL-DML/SQL-Tabelle

Falls in einer Tabelle alle alten Attributformate, d.h. Attributformate der SESAM-Version 13.1 oder einer älteren Version, mit ALTER COLUMN geändert worden sind, bleibt die Tabelle trotzdem weiterhin eine Nur-CALL-DML-Tabelle. Es kann weiterhin nur mit CALL-DML zugegriffen werden.

Um auch mit SQL zugreifen zu können, muss ein MIGRATE durchgeführt werden. MIGRATE stellt eine Nur-CALL-DML-Tabelle in eine CALL-DML/SQL-Tabelle um.

---

```
MIGRATE CALL DML ONLY TABLE tabelle
```

---

*tabelle*

Name der Nur-CALL-DML-Tabelle, die in eine CALL-DML/SQL-Tabelle umgestellt werden soll. Die Tabelle darf keine „alten Attributformate“ mehr enthalten.

## MODIFY - Metadaten zu SESAM-Sicherungsbeständen pflegen

Mit MODIFY können Sie folgende Elemente löschen:

- Sätze aus den Catalog-Tabellen RECOVERY\_UNITS und DA\_LOGS
- Recovery-Unit- und CAT-LOG-Sätze aus der aktuellen CAT-REC-Datei

Nach den folgenden Funktionsbeschreibungen werden die verschiedenen Formen der Utility-Anweisung MODIFY dargestellt.

### Sätze aus den Catalog-Tabellen RECOVERY\_UNITS und DA\_LOGS löschen

Die Catalog-Tabelle RECOVERY\_UNITS enthält Sätze mit Informationen über SESAM-Sicherungsbestände von Anwender-Spaces.

Einen neuen Satz in die Catalog-Tabelle RECOVERY\_UNITS nimmt SESAM/SQL auf für jeden SESAM-Sicherungsbestand, der mit der Utility-Anweisung COPY erstellt wurde. Jeder Satz enthält u.a. den Namen des Anwender-Space, den Zeitstempel des SESAM-Sicherungsbestandes und Informationen über die DA-LOG-Datei, die an die Sicherung anschließt.

Die Catalog-Tabelle DA\_LOGS enthält Sätze mit Informationen über vorhandene DA-LOG-Dateien.

Ein Satz in DA\_LOGS enthält u.a. die Nummer der zugehörigen DA-LOG-Datei, Angaben über den Zeitpunkt, zu dem diese Datei angelegt wurde, sowie Informationen über alle Anwender-Spaces, für die auf der DA-LOG-Datei Änderungen protokolliert sind.

Mit MODIFY können Sie die Catalog-Tabellen RECOVERY\_UNITS und DA\_LOGS pflegen, indem Sie nicht mehr benötigte Sätze löschen. Dazu bietet MODIFY zwei Möglichkeiten:

- In RECOVERY\_UNITS können Sätze gelöscht werden, die sich auf SESAM-Sicherungsbestände eines bestimmten Anwender-Space beziehen.
- In RECOVERY\_UNITS und DA\_LOGS können Sätze gelöscht werden, unabhängig von ihrer Zuordnung zu bestimmten Anwender-Spaces.

Wenn Sie mit MODIFY Sätze aus den Catalog-Tabellen RECOVERY\_UNITS bzw. DA\_LOGS löschen, bleiben die zugehörigen SESAM-Sicherungsdateien der Anwender-Spaces bzw. die DA-LOG-Dateien erhalten.

Aus Datenschutzgründen sollten Sie nicht mehr benötigte SESAM-Sicherungsbestände und DA-LOG-Dateien, die auf Platte angelegt sind, im BS2000 mit dem Kommando DELETE-FILE ... OPTION=\*DESTROY-ALL löschen. SESAM-Sicherungsbestände und DA-LOG-Dateien auf Magnetbandkassette sind Bestandteil des zugeordneten HSMS-Archivs oder ARCHIVE-Directory und müssen deshalb mit HSMS- oder ARCHIVE-Mitteln verwaltet werden (siehe Handbücher „[HSMS \(BS2000\)](#)“ und „[ARCHIVE \(BS2000\)](#)“).

Mindestens die letzten beiden Sätze eines jeden Spaces bleiben in der Tabelle RECOVERY\_UNITS erhalten. Einer dieser Sätze bezieht sich auf den zuletzt erstellten SESAM-Sicherungsbestand des Space. Auf Basis des Satzes, der den letzten SESAM-Sicherungsbestand beschreibt, kann ein RECOVER durchgeführt werden. Die dafür benötigten Sätze in der Tabelle DA\_LOGS bleiben ebenfalls erhalten.

Fremdkopien stellen in diesem Zusammenhang einen Sonderfall dar. Bei der Arbeit mit ihnen entstehen keine neuen Einträge in der Tabelle RECOVERY\_UNITS. Um aus der Catalog-Tabelle DA\_LOGS Sätze löschen zu können, kann die Klausel UNRESTRICTED angegeben werden. Es werden dann Sätze aus den DA\_LOGS gelöscht, ohne Rücksicht auf einen letzten RECOVERY\_UNITS Satz.



### **ACHTUNG!**

Mit UNRESTRICTED werden in den RECOVERY\_UNITS und DA\_LOGS alle Sätze mit dem angegebenen Alter gelöscht, bis auf den jeweils letzten. Dieser bleibt aus technischen Gründen erhalten.

Es ist kein RECOVER mehr möglich, wenn die dafür benötigten Sätze in den Catalog-Tabellen RECOVERY\_UNITS und DA\_LOGS gelöscht wurden. Der älteste für eine Fremdkopie eines Space benötigte Satz in DA\_LOGS ist der Satz, der beim Schließen des Space nach dem letzten Update auf ihm entstanden ist. Entsteht die Fremdkopie während oder nach einer Session, in der der Space nicht geöffnet wurde oder in der nur gelesen wurde, wird der Aufsetzpunkt im Space nicht aktualisiert.

Die Catalog-Tabellen RECOVERY\_UNITS und DA\_LOGS sind im „[Basishandbuch](#)“ näher beschrieben.

### **Sätze aus der CAT-REC-Datei löschen**

Die CAT-REC-Datei (Catalog-Recovery-Datei) enthält Recovery-Unit-Sätze für den Catalog-Space und die dazu gehörenden CAT-LOG-Sätze.

Einen neuen Recovery-Unit-Satz nimmt SESAM/SQL in die CAT-REC-Datei auf für jeden SESAM-Sicherungsbestand der gesamten Datenbank oder des Catalog-Space, der mit der Utility-Anweisung COPY erstellt wurde.

Jeder Recovery-Unit-Satz enthält u.a. den Namen der Sicherung des Catalog-Space und den Zeitstempel der SESAM-Sicherung.

Recovery-Unit-Sätze der CAT-REC-Datei entsprechen den Sätzen der Catalog-Tabelle RECOVERY\_UNITS, beziehen sich aber auf den Catalog-Space.

Einen neuen CAT-LOG-Satz nimmt SESAM/SQL in die CAT-REC-Datei auf für jeden Wechsel der CAT-LOG-Datei. In der CAT-LOG-Datei sind alle Änderungen protokolliert, die sich auf den Catalog-Space beziehen.

Jeder CAT-LOG-Satz enthält u.a. den Namen der CAT-LOG-Datei, das Erstellungsdatum und die Versionsnummer des zugehörigen SESAM-Sicherungsbestandes.

CAT-LOG-Sätze der CAT-REC-Datei entsprechen den Sätzen der Catalog-Tabelle DA\_LOGS, beziehen sich aber auf den Catalog-Space.

Mit MODIFY können Sie die aktuelle CAT-REC-Datei der Datenbank pflegen, indem Sie nicht mehr benötigte Sätze löschen (Online-update der CAT-REC-Datei).

Wenn Sie mit MODIFY Sätze aus der CAT-REC-Datei löschen, bleiben die zugehörigen SESAM-Sicherungsdateien des Catalog und des Catalog-Space sowie die CAT-LOG-Dateien erhalten.

Aus Datenschutzgründen sollten Sie nicht mehr benötigte SESAM-Sicherungsbestände und CAT-LOG-Dateien, die auf Platte angelegt sind, im BS2000 mit dem Kommando DELETE-FILE ... OPTION=\*DESTROY-ALL löschen. SESAM-Sicherungsbestände auf Magnetbandkassette sind Bestandteil des zugeordneten HSMS-Archivs oder ARCHIVE-Directory und müssen deshalb mit HSMS oder ARCHIVE verwaltet werden (siehe Handbücher „[HSMS \(BS2000\)](#)“ und „[ARCHIVE \(BS2000\)](#)“).

Der jüngste Recovery-Unit-Satz und die dazu gehörenden CAT-LOG-Sätze bleiben in der CAT-REC-Datei erhalten. Auf Basis dieses Satzes, der den letzten SESAM-Sicherungsbestand beschreibt, kann ein RECOVER durchgeführt werden. Die dafür benötigten CAT-LOG-Sätze bleiben ebenfalls erhalten.

Fremdkopien stellen in diesem Zusammenhang einen Sonderfall dar. Bei der Arbeit mit ihnen entstehen keine Recovery-Unit-Sätze in der CAT-REC-Datei. CAT-LOG-Sätze können nicht ohne entsprechenden Recovery-Unit-Satz gelöscht werden.

Die CAT-REC-Datei ist im „[Basishandbuch](#)“ näher beschrieben.

### In RECOVERY\_UNITS Sätze für einen bestimmten Anwender-Space löschen

Der aktuelle Berechtigungsschlüssel muss das Sonder-Privileg UTILITY besitzen oder Eigentümer des angegebenen Anwender-Space sein.

---

```
MODIFY RECOVERY SPACE space DELETE { ALL
                                         AGE n
                                         DATE 'jahr-monat-tag' }
```

---

*space*

Name des Anwender-Space, für den Sätze aus RECOVERY\_UNITS gelöscht werden sollen.

DELETE ALL

Es werden alle Sätze zu *space* aus RECOVERY\_UNITS gelöscht. Mindestens die letzten beiden Sätze bleiben aber erhalten. Einer der verbleibenden Sätze bezieht sich auf den zuletzt erstellten SESAM-Sicherungsbestand des Anwender-Space *space*.

DELETE AGE *n*

*n* steht für eine positive ganze Zahl.

Es werden alle Sätze zu *space* aus RECOVERY\_UNITS gelöscht, deren Zeitstempel ein Datum angibt, das mehr als *n* Tage zurückliegt. Mindestens die letzten beiden Sätze bleiben aber erhalten. Einer der verbleibenden Sätze bezieht sich auf den zuletzt erstellten SESAM-Sicherungsbestand des Anwender-Space *space*.

DELETE DATE '*jahr-monat-tag*'

DATE '*jahr-monat-tag*' bezeichnet ein Datum.

Es werden alle Sätze zu *space* aus RECOVERY\_UNITS gelöscht, deren Zeitstempel ein Datum angibt, das älter ist als '*jahr-monat-tag*'. Mindestens die letzten beiden Sätze bleiben aber erhalten. Einer der verbleibenden Sätze bezieht sich auf den zuletzt erstellten SESAM-Sicherungsbestand des Anwender-Space *space*.

## In RECOVERY\_UNITS und DA\_LOGS Sätze unabhängig von ihrer Zuordnung zu bestimmten Anwender-Spaces löschen

Der aktuelle Berechtigungsschlüssel muss das Sonder-Privileg UTILITY besitzen.

---

```
MODIFY RECOVERY DELETE [UNRESTRICTED] { ALL
                                         AGE n
                                         DATE 'jahr-monat-tag' } AT CATALOG catalog
```

---

### DELETE ALL

Für jeden Anwender-Space bleiben in den RECOVERY\_UNITS mindestens die letzten beiden Sätze erhalten. Einer der verbleibenden Sätze bezieht sich auf den zuletzt erstellten SESAM-Sicherungsbestand des Anwender-Space. Die Sätze in DA\_LOGS zu den an diese SESAM-Sicherungsbestände anschließenden DA-LOG-Dateien bleiben ebenfalls erhalten.

Alle anderen Sätze in RECOVERY\_UNITS und DA\_LOGS werden gelöscht. ALL darf nicht angegeben werden, wenn UNRESTRICTED ausgewählt wurde.

### DELETE AGE *n*

*n* steht für eine positive ganze Zahl.

Für jeden Anwender-Space bleiben in den RECOVERY\_UNITS mindestens die letzten beiden Sätze erhalten. Einer der verbleibenden Sätze bezieht sich auf den zuletzt erstellten SESAM-Sicherungsbestand des Anwender-Space. Die Sätze in DA\_LOGS zu den an diese SESAM-Sicherungsbestände anschließenden DA-LOG-Dateien bleiben ebenfalls erhalten.

Alle anderen Sätze in RECOVERY\_UNITS und DA\_LOGS, deren Zeitstempel ein Datum angibt, das mehr als *n* Tage zurückliegt, werden gelöscht.

### DELETE DATE '*jahr-monat-tag*'

DATE '*jahr-monat-tag*' bezeichnet ein Datum.

Für jeden Anwender-Space bleiben in den RECOVERY\_UNITS mindestens die letzten beiden Sätze erhalten. Einer der verbleibenden Sätze bezieht sich auf den zuletzt erstellten SESAM-Sicherungsbestand des Anwender-Space. Die Sätze in DA\_LOGS zu den an diesen SESAM-Sicherungsbestand anschließenden DA-LOG-Dateien bleiben ebenfalls erhalten.

Alle anderen Sätze in RECOVERY\_UNITS und DA\_LOGS, deren Zeitstempel ein Datum angibt, das älter ist als '*jahr-monat-tag*', werden gelöscht.

**UNRESTRICTED**

Mit UNRESTRICTED werden in den RECOVERY\_UNITS und DA\_LOGS alle Sätze mit dem angegebenen Alter gelöscht, bis auf den jeweils letzten. Dieser bleibt aus technischen Gründen erhalten.

**ACHTUNG!**

Es ist kein RECOVER mehr möglich, wenn mit UNRESTRICTED dafür nötige Einträge in der Catalog-Tabelle DA\_LOGS gelöscht wurden.

**AT CATALOG** *catalog*

Name der Datenbank, deren Catalog-Tabellen RECOVERY\_UNITS und DA\_LOGS mit MODIFY bearbeitet werden sollen.

**Beispiele (Sätze aus den Catalog-Tabellen löschen)**

1. Mit Ausnahme der verbleibenden Sätze für den zuletzt erstellten SESAM-Sicherungsbestand des Anwender-Space TABLESPACE löscht das Beispiel aus der Catalog-Tabelle RECOVERY\_UNITS der Datenbank AUFTRAGKUNDEN alle Sätze zu SESAM-Sicherungsbeständen von TABLESPACE.



```
MODIFY RECOVERY SPACE tablespace DELETE ALL
```

2. Mit Ausnahme der verbleibenden Sätze für den zuletzt erstellten SESAM-Sicherungsbestand des Anwender-Space BLOBSPACE löscht das Beispiel aus der Catalog-Tabelle RECOVERY\_UNITS der Datenbank AUFTRAGKUNDEN alle Sätze zu SESAM-Sicherungsbeständen von BLOBSPACE, deren Erstellungsdatum mehr als 15 Tage zurückliegt. Für den zuletzt erstellten SESAM-Sicherungsbestand von BLOBSPACE bleiben die zugehörigen Sätze in RECOVERY\_UNITS auch dann erhalten, wenn dieser SESAM-Sicherungsbestand älter als 15 Tage ist.

```
MODIFY RECOVERY SPACE blobspace DELETE AGE 15
```

3. Das Beispiel belässt für alle Anwender-Spaces der Datenbank AUFTRAGKUNDEN jeweils die verbleibenden Sätze in RECOVERY\_UNITS, die sich auf den zuletzt erstellten SESAM-Sicherungsbestand des betreffenden Anwender-Space beziehen. Die Sätze in DA\_LOGS zu den an diese SESAM-Sicherungsbestände anschließenden DA\_LOG-Dateien bleiben ebenfalls erhalten. Alle anderen Sätze in den zur Datenbank AUFTRAGKUNDEN gehörenden Catalog-Tabellen RECOVERY\_UNITS und DA\_LOGS werden gelöscht.

```
MODIFY RECOVERY DELETE ALL AT CATALOG auftragkunden
```

## Recovery-Unit- und CAT-LOG-Sätze aus der CAT-REC-Datei löschen

Der aktuelle Berechtigungsschlüssel muss das Sonder-Privileg UTILITY besitzen.

---

```
MODIFY RECOVERY CATALOG_SPACE catalog DELETE { ALL
                                                AGE n
                                                DATE 'jahr-monat-tag' }
```

---

### *catalog*

Logischer Name der Datenbank, deren aktuelle CAT-REC-Datei mit MODIFY bearbeitet werden soll (Online-update der CAT-REC-Datei). Die Datenbank muss im Datenbankkatalog eingetragen sein. Die Datenbank darf bereits eröffnet sein.

### DELETE ALL

Es werden alle Recovery-Unit- und CAT-LOG-Sätze zu *catalog* aus der CAT-REC-Datei gelöscht. Der jüngste Recovery-Unit-Satz und die dazu gehörenden CAT-LOG-Sätze bleiben aber erhalten, damit noch eine Recovery mit SESAM/SQL möglich ist.

### DELETE AGE *n*

*n* steht für eine positive ganze Zahl.

Es werden alle Recovery-Unit- und CAT-LOG-Sätze zu *catalog* aus der CAT-REC-Datei gelöscht, deren Zeitstempel ein Datum angibt, das mehr als *n* Tage zurückliegt. Der jüngste Recovery-Unit-Satz und die dazu gehörenden CAT-LOG-Sätze bleiben aber erhalten, damit noch eine Recovery mit SESAM/SQL möglich ist.

### DELETE DATE '*jahr-monat-tag*'

DATE '*jahr-monat-tag*' bezeichnet ein Datum.

Es werden alle Recovery-Unit- und CAT-LOG-Sätze zu *catalog* aus der CAT-REC-Datei gelöscht, deren Zeitstempel ein Datum angibt, das älter ist als '*jahr-monat-tag*'. Der jüngste Recovery-Unit-Satz und die dazu gehörenden CAT-LOG-Sätze bleiben aber erhalten, damit noch eine Recovery mit SESAM/SQL möglich ist.

## Beispiel (Sätze aus der CAT-REC-Datei löschen)

Das Beispiel belässt für die Datenbank AUFTRAGKUNDEN jeweils den verbleibenden Recovery-Unit-Satz in der CAT-REC-Datei, der sich auf den zuletzt erstellten SESAM-Sicherungsbestand des Catalog-Space bezieht. Die dazu gehörenden CAT-LOG-Sätze in der CAT-REC-Datei bleiben ebenfalls erhalten. Alle anderen Recovery-Unit- und CAT-LOG-Sätze in der CAT-REC-Datei, die zur Datenbank AUFTRAGKUNDEN gehört, werden gelöscht.

```
MODIFY RECOVERY CATALOG_SPACE auftragkunden DELETE ALL
```

## RECOVER - Zurücksetzen und Reparieren, Indizes neu aufbauen

Mit RECOVER können Sie

- Teile einer SESAM/SQL-Datenbank oder auch die gesamte Datenbank reparieren bzw. zurücksetzen,
- Indizes neu aufbauen.

Im Folgenden sind die verschiedenen Formen der RECOVER-Anweisung in eigenen Abschnitten beschrieben.

## Anwender-Spaces und Datenbank reparieren oder zurücksetzen

Mit RECOVER können Sie folgende Einheiten reparieren, zurücksetzen oder neu aufbauen:

- einen einzelnen Anwender-Space
- ein Space-Set
- eine Space-Liste
- den Catalog-Space der Datenbank
- die gesamte Datenbank

Sie verwenden dafür:

- SESAM-Sicherungsbestände, die mit der Utility-Anweisung COPY erstellt wurden
- Fremdkopien, die mit beliebigen BS2000-Mitteln (z.B. TimeFinder) erstellt wurden
- Replikate, die mit der Utility Anweisung CREATE REPLICATION erzeugt wurden

Bei der gleichzeitigen Aktualisierung verschiedener Anwender-Spaces kann ein höherer Parallelitätsgrad erreicht werden, wenn Sie mehrere, parallel ablaufende RECOVER-Anweisungen eingeben. Weitere Informationen dazu entnehmen Sie bitte dem „[Basishandbuch](#)“, Abschnitt „Ausführung von Utility-Anweisungen durch SESAM/SQL“.

Als Space-Set wird eine Einheit aus mehreren Anwender-Spaces mit gemeinsamem Zeitstempel bezeichnet. Spaces mit gleichem Zeitstempel entstehen bei der Sicherung mit einer COPY CATALOG-Anweisung oder bei der Sicherung mehrerer Spaces mit einer COPY-Anweisung. Die Reparatur eines Space-Set als Einheit ist nur möglich, wenn außerdem das Logging zwischenzeitlich für keinen Anwender-Space dieses Space-Sets unterbrochen wurde.

Eine Space-Liste bezeichnet mehrere Anwender-Spaces aus der Sicherung einer gesamten Datenbank oder eines Space-Set. Die Spaces einer Space-Liste müssen alle den gleichen Zeitstempel besitzen. Die Reparatur der Spaces einer Space-Liste ist nur möglich, wenn außerdem das Logging zwischenzeitlich für keinen Anwender-Space dieser Space-Liste unterbrochen wurde.

Wenn Sie für RECOVER eine Fremdkopie verwenden, dann müssen Sie die Dateien der Fremdkopie zuvor auf die entsprechenden Spaces kopiert haben. Die Klausel FOREIGN COPY bewirkt, dass nur die Loggingdateien auf Basis der in den Spaces vorhandenen Aufsetzinformationen nachgefahren werden. SESAM-Sicherungsbestände bleiben dabei unberücksichtigt. Zum Einspielen der Fremdkopie müssen der betroffene Space oder die Datenbank geschlossen sein.

Space-Liste bei Fremdkopie:

Beim RECOVER mit Fremdkopie kann eine Space-Liste angegeben werden. Bei Fremdkopien verschiedener Spaces müssen die Spaces vom selben Sicherungszeitpunkt stammen. Die Administrationsanweisung PREPARE-FOREIGN-COPY sorgt beim Erstellen der Fremdkopien dafür, dass die Spaces denselben Sicherungszeitstempel besitzen.

Ein Replikat kann auf einer beliebigen Kennung liegen. Welche das ist, wird über eine Klausel in der Anweisung angegeben.

Bei SESAM-Sicherungen auf Platte wird die jeweils benötigte Sicherungsdatei immer zuerst in der DB-Kennung gesucht. Sollte sie nicht in der DB-Kennung vorhanden sein, wird sie aus der DBH-Kennung eingespielt. Sind SESAM-Sicherungsbestände in der DB- und der DBH-Kennung vorhanden, werden diejenigen aus der DB-Kennung verwendet.

Bei SESAM-Sicherungen auf Magnetbandkassette mit HSMS oder ARCHIVE sind alle Informationen über den Ort der Sicherung in der Catalog-Tabelle RECOVERY\_UNITS und in dem zugehörigen HSMS-Archiv bzw. ARCHIVE-Directory abgelegt.

Wenn Sie für RECOVER eine mit ARCHIVE erstellte SESAM-Online-Sicherung verwenden, dann wird die mitgesicherte PBI-Datei nachgefahren.

Mit Hilfe der Maske INF.5.2 des Utility-Monitors können Sie sich die benötigten Logging-Dateien entweder auf den Bildschirm, in eine Datei oder in Jobvariablen ausgeben lassen (siehe Handbuch „[Utility-Monitor](#)“).

Um den Catalog-Space einer Datenbank bzw. die gesamte Datenbank mit RECOVER reparieren zu können, muss die Catalog-Recovery-Datei (CAT-REC-Datei) vorhanden sein. Es ist ratsam, die CAT-REC-Datei doppelt zu führen (siehe „[Basishandbuch](#)“). Falls die CAT-REC-Datei dennoch einmal defekt sein sollte, wenden Sie sich bitte an den Service.

Für partitionierte Tabellen müssen beim Zurücksetzen mit Aufbau der Indizes (RECOVER ... TO und NO INDEX nicht angegeben) und wenn Integritätsbedingungen für die Tabelle definiert sind alle Partitionen verfügbar sein (siehe [Tabelle 3 auf Seite 22](#)).

Die folgende Tabelle gibt einen detaillierten Überblick über die möglichen RECOVER-Anweisungen und ihre Syntaxkombinationen (x bezeichnet die zulässigen Kombinationen).

Anweisung	USING					TO					USING TO [ANY] <i>zeitstempel_literal</i>					RESTART	ADJUST	weitere mögliche Klauseln
	COPY_FILE	COPY_NUMBER	<i>zeitstempel_literal</i>	FOREIGN COPY	REPLICATION	COPY_FILE	COPY_NUMBER	[ANY] <i>zeitstempel_literal</i>	FOREIGN COPY	REPLICATION	COPY_FILE	COPY_NUMBER	<i>zeitstempel_literal</i>	FOREIGN COPY	REPLICATION			
RECOVER CATALOG	x	x	x	x	x	x	x	x <sup>1</sup>	x	x	x <sup>1</sup>	x <sup>1</sup>	x <sup>1</sup>	x <sup>1</sup>	x	-	-	PASSWORD GENERATE INDEX <sup>2</sup> SCOPE PENDING NO INDEX
	x	x	x	x	x	x	x	x <sup>1</sup>	x	x	x <sup>1</sup>	x <sup>1</sup>	x <sup>1</sup>	x <sup>1</sup>	x	-	-	
	x	x	x	x	x	x	x	x <sup>1</sup>	x	x	x <sup>1</sup>	x <sup>1</sup>	x <sup>1</sup>	x <sup>1</sup>	x	-	-	
	x	x	x	-	-	x	x	x	-	-	x	x	x	-	-	-	-	
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
RECOVER CATALOG_SPACE	x	x	x	x	x	x	x	x	x	x	-	-	-	-	-	-	-	PASSWORD GENERATE INDEX <sup>2</sup> SCOPE PENDING NO INDEX
	x	x	x	x	x	x	x	x	x	x	-	-	-	-	-	-	-	
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
RECOVER SPACE <i>space</i>	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	PASSWORD GENERATE INDEX <sup>2</sup> SCOPE PENDING NO INDEX
	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
	x	x	x	-	x	x	x	x	-	x	x	x	x	-	x	-	-	
	-	-	-	-	x	x	x	x	x	x	x	x	x	x	x	-	x	
RECOVER SPACE <i>space, space, ...</i>	-	-	x	x	x	-	-	x	x	x	-	-	-	-	-	-	-	PASSWORD GENERATE INDEX <sup>2</sup> SCOPE PENDING NO INDEX
	-	-	x	x	x	-	-	x	x	x	-	-	-	-	-	-	-	
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
	-	-	x	-	x	-	-	-	-	-	-	-	-	-	-	-	-	
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
RECOVER SPACESET	-	-	x	-	-	-	-	x	-	-	-	-	-	-	-	-	-	PASSWORD GENERATE INDEX <sup>2</sup> SCOPE PENDING NO INDEX
	-	-	x	-	-	-	-	x	-	-	-	-	-	-	-	-	-	
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
	-	-	x	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
	-	-	-	-	-	-	-	x	-	-	-	-	-	-	-	-	-	

Tabelle 9: Überblick über mögliche RECOVER-Anweisungen und ihre Syntax-Kombinationen

<sup>1</sup> Bei Angabe von *zeitstempel\_literal* ist auch ANY zulässig

<sup>2</sup> GENERATE INDEX ON NO LOG INDEX SPACE

## RECOVER SPACE

Zur Ausführung von RECOVER SPACE muss der aktuelle Berechtigungsschlüssel entweder das Sonder-Privileg UTILITY besitzen oder Eigentümer des von RECOVER betroffenen Anwender-Space sein.

In diesem Abschnitt werden folgende RECOVER-Anweisungen beschrieben:

- RECOVER eines Space mit SESAM-Sicherungsbeständen bzw. Fremdkopien
- RECOVER eines Space mit Replikaten

Bei der Reparatur eines Anwender-Space mit RECOVER *space* USING erhält der Space, wenn er der logischen Datensicherung unterliegt, den Zustand „copy pending“, wenn CREATE INDEX oder DROP INDEX für eine partitionierte Tabelle auf diesem Space nachgefahren werden muss.

### ***RECOVER SPACE mit SESAM-Sicherungsbeständen bzw. Fremdkopien***

---

```
RECOVER SPACE space [ { [USING rec_unit] [TO rec_unit]
                        RESTART
                        ADJUST
                      } ]
```

```
[PASSWORD kennwort] [SCOPE PENDING] [NO INDEX]
```

```
rec_unit ::= { COPY_FILE datei
                COPY_NUMBER zahl
                zeitstempel_literal
                FOREIGN COPY
              }
```

```
zeitstempel_literal ::= TIMESTAMP 'jahr-monat-tag stunde:minute:sekunde'
```

---

**RECOVER SPACE** *space*

Name des Anwender-Space, der repariert oder zurückgesetzt werden soll.

Als Name für den Anwender-Space darf nicht „CATALOG“ angegeben werden.

Bei RECOVER SPACE und RECOVER SPACE ... USING müssen für den Anwender-Space *space* folgende Bedingungen erfüllt sein:

- Bei SESAM-Sicherungen muss der benötigte SESAM-Sicherungsbestand von *space* vorhanden und in der Catalog-Tabelle RECOVERY\_UNITS vermerkt sein. Fremdkopien haben keine Einträge in der Catalog-Tabelle RECOVERY\_UNITS. Die Aufsetzinformation für das Nachfahren der Loggingdateien wird im Space selber angelegt.
- Für den Anwender-Space *space* darf seit dem Erstellen des benötigten Sicherungsbestandes das Logging nicht unterbrochen worden sein (z.B. durch LOAD OFFLINE, IMPORT TABLE, MIGRATE oder ALTER SPACE ... NO LOG).
- Alle im Anschluss an den benötigten Sicherungsbestand von *space* erstellten DA-LOG-Dateien müssen vorhanden sein.

**RECOVER SPACE** *space* ohne USING, TO, RESTART und ADJUST

SESAM/SQL repariert den Anwender-Space *space* auf der Basis des zuletzt erstellten SESAM-Sicherungsbestandes dieses Anwender-Space.

**USING** *rec\_unit*

SESAM/SQL repariert den Anwender-Space *space* unter Verwendung des durch *rec\_unit* identifizierten SESAM-Sicherungsbestandes oder der Fremdkopie.

USING darf keine Marke spezifizieren.

Soll eine Fremdkopie verwendet werden, muss der Space beim Erzeugen dieser Fremdkopie im Logging gewesen sein. Der DA-LOG-Eintrag, der in der Fremdkopie als Aufsatzpunkt vermerkt ist, muss in den Metadaten der Datenbank noch eingetragen sein.

Spaces, die diese Bedingungen nicht erfüllen, bleiben bei einem RECOVER SPACE ... USING FOREIGN COPY unverändert. Auf sie kann später nur zugegriffen werden, wenn sie seit Erzeugen der Fremdkopie nicht geändert worden sind. Andernfalls wird bei einem Zugriffsversuch die Fehlermeldung ausgegeben, dass der Space inkonsistent ist. War der Space nicht im Logging, als die Fremdkopie erzeugt wurde, ist nur RECOVER SPACE ... TO FOREIGN COPY möglich.

**TO** *rec\_unit*

SESAM/SQL setzt den Anwender-Space *space* auf den durch *rec\_unit* spezifizierten SESAM-Sicherungsbestand oder die Fremdkopie zurück. Wird bei TO eine Marke angegeben, dann werden die Logging-Dateien auf Basis der letzten davor liegenden SESAM-Sicherung bis zur Marke nachgefahren.

**ACHTUNG!**

Ein RECOVER SPACE ... TO kann nicht rückgängig gemacht werden. Ein neuerer Stand als der durch *rec\_unit* spezifizierte kann nicht mehr erreicht werden.

Es können Inkonsistenzen zwischen dem eingespielten SESAM-Sicherungsbestand bzw. der Fremdkopie für einen Anwender-Space und den zugehörigen Metadaten im Catalog-Space auftreten. In diesem Fall passt SESAM/SQL die Daten im zurückgesetzten Anwender-Space an die Metadaten im Catalog-Space an. Wie SESAM/SQL diese Anpassung im Einzelnen vornimmt, ist im „[Basishandbuch](#)“ beschrieben.

**ACHTUNG!**

Die Daten einer Tabelle können bei RECOVER SPACE ... TO verloren gehen. Zur Anpassung einer Tabelle oder eines Index an die zugehörigen Metadaten im Catalog-Space werden die Tabelle oder der Index gelöscht und anschließend neu aufgebaut.

Ist der Space zum Zeitpunkt eines RECOVER SPACE ... TO im Logging und wurden Indizes neu aufgebaut, kommt er in den Zustand „copy pending“.

**USING** *rec\_unit* **TO** *zeitstempel\_literal* <sup>1</sup>

Dies ist ein Rücksetzen auf den bei TO spezifizierten Stand.

Ist sowohl USING als auch TO angegeben, so muss USING eine Sicherung spezifizieren und bei TO ist nur die Angabe des Zeitstempels einer Sicherung oder einer Marke erlaubt. Auf Basis der bei USING spezifizierten Sicherung wird, durch Nachfahren der Logging-Dateien, der bei TO spezifizierte Stand wiederhergestellt. Die bei USING spezifizierte Sicherung muss älter sein als der bei TO angegebene Zeitstempel.

**ACHTUNG!**

Es gelten die Hinweise und Warnungen, die bei RECOVER SPACE ... TO beschrieben sind, in analoger Weise.

<sup>1</sup> **TIMESTAMP** 'jahr-monat-tag stunde:minute:sekunde' ist im Handbuch „[SQL-Sprachbeschreibung Teil 1: SQL-Anweisungen](#)“ unter „Zeitstempel“ näher erklärt.

**RESTART**

SESAM/SQL setzt eine wegen fehlender DA-LOG-Datei(en) unterbrochene Reparatur des Anwender-Space *space* an der Stelle fort, wo die Reparatur unterbrochen wurde.

Wurde die Ausführung der Utility-Anweisung RECOVER nicht wegen fehlender DA-LOG-Datei(en) unterbrochen, sondern aus einem anderen Grund, wird RECOVER SPACE ... RESTART eventuell abgewiesen. In diesem Fall müssen Sie die Reparatur des Anwender-Space mit RECOVER neu beginnen.

**ADJUST**

SESAM/SQL passt nach einer wegen fehlender DA-LOG-Datei(en) unterbrochenen Reparatur den Anwender-Space *space* im momentanen, durch den bisherigen Reparaturverlauf erreichten Zustand an die aktuellen Metadaten des Catalog-Space an. Wie SESAM/SQL diese Anpassung im Einzelnen vornimmt, ist im „[Basis](#)handbuch“ beschrieben.

Wurde die Ausführung der Utility-Anweisung RECOVER nicht wegen fehlender DA-LOG-Datei(en) unterbrochen, sondern aus einem anderen Grund, wird RECOVER SPACE ... ADJUST eventuell abgewiesen. In diesem Fall müssen Sie die Reparatur des Anwender-Space mit RECOVER neu beginnen.

*rec\_unit ::=*

Sicherungsbestand oder Marke.

Der durch *rec\_unit* spezifizierte SESAM-Sicherungsbestand des Anwender-Space oder die durch *rec\_unit* spezifizierte Marke muss in der Catalog-Tabelle RECOVERY\_UNITS vermerkt sein.

Fremdkopien haben keine Einträge in der Catalog-Tabelle RECOVERY\_UNITS. Die Aufsetzinformation für das Nachfahren der Loggingdateien wird im Space selber angelegt.

*COPY\_FILE datei*

Dateiname eines mit der Utility-Anweisung COPY erstellten SESAM-Sicherungsbestandes. *datei* muss als alphanumerisches Literal angegeben werden.

Bei RECOVER SPACE ... TO COPY\_FILE kann auf die Datei *datei* auch dann zurückgesetzt werden, wenn *datei* nicht (mehr) in der Catalog-Tabelle RECOVERY\_UNITS eingetragen ist.

Bei RECOVER SPACE ... TO COPY\_FILE können Sie für *datei* auch '\*DUMMY' angeben. In diesem Fall setzt SESAM/SQL auf den leeren Anwender-Space zurück und sorgt für die Konsistenz zwischen dem zurückgesetzten Anwender-Space und den Metadaten im Catalog-Space. Wie SESAM/SQL diese Konsistenz herstellt, ist im „[Basishandbuch](#)“ beschrieben.



#### **ACHTUNG!**

Bei RECOVER SPACE ... TO COPY\_FILE '\*DUMMY' sind im zurückgesetzten Anwender-Space alle Tabellen und Indizes initialisiert, die Tabellen enthalten jedoch keine Anwenderdaten. Indizes enthalten Daten nur dann, wenn sie zu Tabellen gehören, die in anderen Anwender-Spaces liegen.

Die Angabe '\*DUMMY' ist z.B. sinnvoll, wenn zu einem defekten Anwender-Space kein SESAM-Sicherungsbestand existiert oder wenn Inhalte der Tabellen eines Space gelöscht werden sollen.

*COPY\_NUMBER zahl*

Versionsnummer eines mit der Utility-Anweisung COPY erstellten SESAM-Sicherungsbestandes. *zahl* bezeichnet eine maximal 6-stellige ganze Zahl.

Der durch *zahl* identifizierte SESAM-Sicherungsbestand muss in der Catalog-Tabelle RECOVERY\_UNITS vermerkt sein.

*zeitstempel\_literal*

Zeitstempel, der den Zeitpunkt eines SESAM-Sicherungsbestandes oder einer bei LOAD OFFLINE, MIGRATE oder IMPORT gesetzten Marke bezeichnet. Die Kopie bzw. Marke muss in der Tabelle RECOVERY\_UNITS bekannt sein.

Näheres zum Zeitpunkt einer Kopieerstellung finden Sie bei der Utility-Anweisung „[COPY - SESAM-Sicherungsbestand erstellen](#)“ auf Seite 97.

Nähere Informationen zum Gebrauch von Marken finden Sie im „[Basishandbuch](#)“. Bei RECOVER SPACE ... USING kann keine Marke angegeben werden.

### FOREIGN COPY

Die Klausel bewirkt, dass nur die Logginginformationen nachgefahren und keine SESAM-Sicherungsbestände eingespielt werden. Die Dateien der Fremdkopie müssen vor dem Start der RECOVER-Anweisung auf die entsprechenden Spaces kopiert werden. Zum Einspielen der Fremdkopie muss der Space geschlossen sein. Die Erzeugung einer Fremdkopie wird im „[Basishandbuch](#)“ beschrieben.

### PASSWORD *kennwort*

BS2000-Kennwort. *kennwort* muss als alphanumerisches Literal angegeben werden.

Für die Angabe von *kennwort* gibt es folgende Möglichkeiten:

- 'C'*zeichenkette*"  
*zeichenkette* enthält vier abdruckbare Zeichen.
- 'X'*hex-zeichenkette*"  
*hex-zeichenkette* enthält acht sedezimale Zeichen.
- '*n*'  
*n* bezeichnet eine ganze Zahl von - 2147483648 bis + 2147483647.

Mit PASSWORD *kennwort* geben Sie das Kennwort für alle SESAM-Sicherungsbestände an, die SESAM/SQL während der Ausführung der RECOVER-Anweisung benötigt.

Die Angabe des BS2000-Kennworts für einen SESAM-Sicherungsbestand ist immer dann erforderlich, wenn nach dem Erstellen des SESAM-Sicherungsbestandes das BS2000-Kennwort der Datenbank geändert wurde, d.h., wenn die BS2000-Kennwörter für Datenbank und SESAM-Sicherungsbestand nicht mehr übereinstimmen.

### SCOPE PENDING

Bei Angabe dieser Klausel wird der Space repariert oder zurückgesetzt, wenn er nicht für die Update-Verarbeitung eröffnet werden kann, weil er z.B. zuvor wegen eines Fehlers defekt gesetzt worden ist, eine Inkonsistenz zwischen Catalog und Space festgestellt wird oder ein DMS-Fehler beim Öffnen der Datei gemeldet wird. Auch ein Space, der eine Tabelle enthält, die im Rahmen einer früheren Fehlerbehandlung als defekt markiert wurde, wird repariert. Es erfolgt keine Prüfung auf formale Korrektheit. Ein Space, der nur Indizes enthält, wird nicht repariert, wenn er einen oder mehrere defekte Indizes enthält.

SCOPE PENDING ist bei Fremdkopien nicht erlaubt.

### NO INDEX

Mit dieser Klausel werden Indizes als defekt gekennzeichnet und nicht neu aufgebaut, wenn sie durch das Zurücksetzen eines Space ungültig geworden sind.

Indizes werden ungültig, wenn die Basistabelle und ein zugehöriger Index auf unterschiedlichen Anwender-Spaces liegen oder wenn die Tabelle partitioniert ist und nicht alle betroffenen Anwender-Spaces gleichzeitig zurückgesetzt werden.

NO INDEX ist nur bei RECOVER SPACE ... TO, RECOVER SPACE USING ... TO und bei RECOVER SPACE ... ADJUST erlaubt.

**RECOVER SPACE mit Replikaten**


---

```
RECOVER SPACE space { USING replikat_beschreibung [TO zeitstempel_literal] }
                    { TO replikat_beschreibung }
```

```
[SCOPE PENDING] [NO INDEX]
```

```
replikat_beschreibung ::=
```

```
  REPLICATION replikat
  [PASSWORD kennwort]
  [ON USER_ID bs2000_benutzerkennung]
  { COPY }
  { RENAME }
```

**RECOVER SPACE *space***

Name des Anwender-Space, der repariert oder zurückgesetzt werden soll. Als Name darf nicht „CATALOG“ angegeben werden.

Voraussetzung für RECOVER SPACE *space* USING:

- Das Replikat muss vorhanden sein.
- Alle im Anschluss an das Replikat erstellten und in der Catalog-Tabelle DA\_LOGS vermerkten DA-LOG-Dateien müssen vorhanden sein.
- Für den Anwender-Space *space* darf das Logging nicht unterbrochen worden sein.

USING *replikat\_beschreibung*

SESAM/SQL repariert den Anwender-Space *space* aus dem angegebenen Replikat.

USING *replikat\_beschreibung* TO *zeitstempel\_literal* <sup>1</sup>

Auf Basis des bei USING spezifizierten Replikats wird durch Nachfahren der DA-LOG-Dateien der bei TO spezifizierte Stand wiederhergestellt. Das bei USING spezifizierte Replikat muss älter sein als der bei TO angegebene Zeitstempel.

---

<sup>1</sup> TIMESTAMP 'jahr-monat-tag stunde:minute:sekunde' ist im Handbuch „SQL-Sprachbeschreibung Teil 1: SQL-Anweisungen“ unter „Zeitstempel“ näher erklärt.

*zeitstempel\_literal*

Zeitstempel, der den Zeitpunkt eines SESAM-Sicherungsbestandes oder einer bei LOAD OFFLINE, MIGRATE oder IMPORT gesetzten Marke bezeichnet. Die Kopie bzw. Marke muss in der Tabelle RECOVERY\_UNITS bekannt sein.

Näheres zum Zeitpunkt einer Kopieerstellung finden Sie bei der Utility-Anweisung „[COPY - SESAM-Sicherungsbestand erstellen](#)“ auf Seite 97.

Nähere Informationen zum Gebrauch von Marken finden Sie im „[Basishandbuch](#)“.

**ACHTUNG!**

Ein RECOVER SPACE USING ... TO kann nicht rückgängig gemacht werden. Ein neuerer Stand als der durch *replikat\_beschreibung* spezifizierte kann nicht mehr erreicht werden.

Es können Inkonsistenzen zwischen dem eingespielten Replikat-Space und den zugehörigen Metadaten im Catalog-Space auftreten. In diesem Fall passt SESAM/SQL die Daten im zurückgesetzten Anwender-Space an die Metadaten im Catalog-Space an. Wie SESAM/SQL diese Anpassung im Einzelnen vornimmt, ist im „[Basishandbuch](#)“ beschrieben.

**ACHTUNG!**

Die Daten einer Tabelle können bei RECOVER SPACE USING ... TO verloren gehen. Zur Anpassung einer Tabelle oder eines Index an die zugehörigen Metadaten im Catalog-Space werden die Tabelle oder der Index gelöscht und anschließend neu aufgebaut.

RECOVER SPACE USING ... TO mit einem Replikat führt immer dazu, dass der angegebene Space und ggf. abhängige Index-Spaces in den Zustand „copy pending“ versetzt werden (siehe auch „[Basishandbuch](#)“, Abschnitt „Replikat einer Datenbank“).

**TO** *replikat\_beschreibung*

SESAM/SQL setzt den Anwender-Space *space* auf den Stand des angegebenen Replikats zurück.

**ACHTUNG!**

Es gelten die Hinweise und Warnungen, die bei RECOVER SPACE USING ... TO beschrieben sind, in analoger Weise.

*replikat\_beschreibung* ::= =

Replikat oder Teilreplikat, das für das Reparieren, Zurücksetzen oder Erzeugen eines eigenständigen Originals verwendet werden soll.

REPLICATION *replikat*

*replikat* ist der physikalische Name des Replikats.

PASSWORD *kennwort*

BS2000-Kennwort des Replikats. *kennwort* muss als alphanumerisches Literal angegeben werden.

Für die Angabe von *kennwort* gibt es folgende Möglichkeiten:

- 'C'*zeichenkette*"  
*zeichenkette* enthält vier abdruckbare Zeichen.
- 'X'*hex-zeichenkette*"  
*hex-zeichenkette* enthält acht sedezimale Zeichen.
- 'n'  
*n* bezeichnet eine ganze Zahl von - 2147483648 bis + 2147483647.

ON USER\_ID *bs2000\_benutzerkennung*

Kennung des Replikats. Die Benutzerkennung wird ohne „\$“ angegeben.

COPY

Der angegebene Space des Replikats wird auf den Space des Catalogs kopiert. Muss der Space des Replikats dabei in eine Kennung ungleich der DBH-Kennung kopiert werden, müssen dort alle erforderlichen Dateien bereits angelegt sein. Alternativ kann auch in der DB-Kennung die DBH-Kennung als Miteigentümer für die Dateien des Catalogs definiert werden.

Die Angaben zur Storage Group für den Anwender-Space werden ignoriert.

Das Replikat kann weiterhin aktualisiert und für Wiedergewinnungsanwendungen genutzt werden. Das Replikat selbst wird bei diesem RECOVER nicht aktualisiert.



Wird COPY aus einem Grund abgewiesen, der auch ein REFRESH REPLICATION verhindert, so ist der Replikat-Anwender-Space weiterhin verwendbar.

Wenn die Ursache behoben ist (z.B. Bereitstellen fehlender Logging-Dateien), kann die Utility-Anweisung wiederholt werden. Kann die Ursache nicht behoben werden (z.B. der Space ist im Original zurückgesetzt worden), dann muss der Anwender-Space mit einem SESAM-Sicherungsbestand repariert oder mit RECOVER SPACE ... TO REPLICATION zurückgesetzt werden.

## RENAME

Der betroffene Space des angegebenen Replikats wird auf den Space des Catalogs umbenannt. Er darf im Original nicht mehr vorhanden sein, weil die Umbenennung sonst nicht möglich ist. Nach der Umbenennung fehlt er im Replikat und kann mit REFRESH SPACE wieder hergestellt werden.

RENAME kann nur genutzt werden, wenn Replikat und Catalog in derselben BS2000-Kennung liegen. Ist diese Kennung ungleich der DBH-Kennung, so muss die DBH-Kennung als Miteigentümer für alle Dateien des Replikats und Catalogs in der DB-Kennung definiert sein. Das Replikat darf in keinem SQL-Datenbankverzeichnis eingetragen sein.



Wird RENAME aus einem Grund abgewiesen, der auch ein REFRESH REPLICATION verhindert, so ist der Replikat-Anwender-Space nicht mehr vorhanden. Der betroffene Anwender-Space muss dann mit SESAM-Sicherungsbeständen repariert werden.

## SCOPE PENDING

Bei Angabe dieser Klausel wird der Space repariert oder zurückgesetzt, wenn er nicht für die Update-Verarbeitung eröffnet werden kann, weil er z.B. zuvor wegen eines Fehlers defekt gesetzt worden ist, eine Inkonsistenz zwischen Catalog und Space festgestellt wird oder ein DMS-Fehler beim Öffnen der Datei gemeldet wird. Auch ein Space, der eine Tabelle enthält, die im Rahmen einer früheren Fehlerbehandlung als defekt markiert wurde, wird repariert. Es erfolgt keine Prüfung auf formale Korrektheit. Ein Space, der nur Indizes enthält, wird nicht repariert, wenn er einen oder mehrere defekte Indizes enthält.

## NO INDEX

Mit dieser Klausel werden Indizes als defekt gekennzeichnet und nicht neu aufgebaut, wenn sie durch das Zurücksetzen eines Space ungültig geworden sind. Indizes werden ungültig, wenn die Basistabelle und ein zugehöriger Index auf unterschiedlichen Anwender-Spaces liegen oder wenn die Tabelle partitioniert ist und nicht alle betroffenen Anwender-Spaces gleichzeitig zurückgesetzt werden.

Die Angabe von NO INDEX ist nur bei RECOVER SPACE ... TO und RECOVER SPACE USING ...TO erlaubt.

Weitere Informationen zum Konzept und der Anwendung finden Sie im „[Basishandbuch](#)“.

## RECOVER eines Space-Set und einer Space-Liste

Zur Ausführung von RECOVER SPACESET oder RECOVER SPACE mit einer Space-Liste muss der aktuelle Berechtigungsschlüssel entweder das Sonder-Privileg UTILITY besitzen oder Eigentümer aller durch den zugehörigen Zeitstempel identifizierten Anwender-Spaces sein.

In diesem Abschnitt werden folgende RECOVER-Anweisungen beschrieben:

- RECOVER eines Space-Set
- RECOVER einer Space-Liste mit SESAM-Sicherungsbeständen bzw. Fremdkopien
- RECOVER einer Space-Liste mit Replikaten

Bei der Reparatur von Anwender-Spaces mit RECOVER Space-Set / Space-Liste USING erhält ein betroffener Space, der der logischen Datensicherung unterliegt, den Zustand „copy pending“, wenn CREATE INDEX oder DROP INDEX für eine partitionierte Tabelle auf diesem Space nachgefahren werden muss, aber nicht alle Partitionen der partitionierten Tabelle in der RECOVER-Anweisung enthalten sind.

## Space-Set und Space-Liste

Ein Space-Set besteht aus mehreren Anwender-Spaces. Ein Space-Set entsteht durch eine COPY-Anweisung mehrerer Anwender-Spaces und wird durch einen Zeitstempel identifiziert. Die gemeinsame Sicherung stellt einen konsistenten Sicherungsbestand dieser Spaces dar. Bei der Recovery eines Space-Set berücksichtigt SESAM/SQL die Konsistenz zwischen den Tabellen und ihren Indizes.

Zu einer gemeinsamen SESAM-Sicherung gehören die Spaces, die mit COPY CATALOG oder COPY *space-liste* gesichert worden sind.

Als gemeinsam gesicherte Spaces einer Fremdkopie gelten die Spaces, die zusammen nach einem PREPARE-FOREIGN-COPY gesichert worden sind.

Die Spaces eines Replikats sind ebenfalls zueinander konsistent und bilden insgesamt auch ein Space-Set.

Bei SESAM-Sicherungen kann ein Space-Set über den Zeitstempel der gemeinsamen Sicherung spezifiziert werden. Eine Untermenge kann aber auch als Space-Liste angegeben werden. Bei Fremdkopien und Replikaten kann ein Space-Set nur über die Aufzählung der Spaces als Space-Liste angegeben werden.

In einer Space-Liste dürfen nur Spaces eines Space-Set aufgeführt werden.

***RECOVER eines Space-Set***


---

```
RECOVER SPACESET AT CATALOG catalog { USING zeitstempel_literal }
                                     { TO zeitstempel_literal }
```

```
[PASSWORD kennwort] [SCOPE PENDING] [NO INDEX]
```

```
zeitstempel_literal ::= TIMESTAMP 'jahr-monat-tag stunde:minute:sekunde'
```

---

RECOVER SPACESET AT CATALOG *catalog*

Name der Datenbank, zu der der Space-Set gehört, der repariert bzw. zurückgesetzt werden soll.

USING *zeitstempel\_literal*<sup>1</sup>

SESAM/SQL repariert alle Anwender-Spaces, von denen jeweils ein zu dem durch *zeitstempel\_literal* identifizierten Zeitpunkt erstellter SESAM-Sicherungsbestand vorliegt. Diese SESAM-Sicherungsbestände, die alle mit derselben COPY-Anweisung erstellt wurden, verwendet SESAM/SQL bei der Reparatur.

Alle benötigten DA-LOG-Dateien müssen vorhanden sein.

Anwender-Spaces werden von der Reparatur ausgenommen, wenn sie nicht in der logischen Datensicherung sind oder die logische Datensicherung seit dem Erstellen des SESAM-Sicherungsbestandes unterbrochen worden ist.

In diesem Fall wird eine Warnung ausgegeben.

---

<sup>1</sup> TIMESTAMP '*jahr-monat-tag stunde:minute:sekunde*' ist im Handbuch „[SQL-Sprachbeschreibung Teil 1: SQL-Anweisungen](#)“ unter „Zeitstempel“ näher erklärt.

*TO zeitstempel\_literal*

SESAM/SQL setzt alle Anwender-Spaces zurück, von denen ein zu dem durch *zeitstempel\_literal* identifizierten Zeitpunkt erstellter SESAM-Sicherungsbestand vorliegt. Auf diesen SESAM-Sicherungsbestand setzt SESAM/SQL jeweils zurück. Alle durch *zeitstempel\_literal* identifizierten SESAM-Sicherungsbestände wurden mit derselben COPY-Anweisung erstellt.

**ACHTUNG!**

Ein RECOVER SPACESET ... TO kann nicht rückgängig gemacht werden. Ein neuerer Stand als der durch *zeitstempel\_literal* identifizierte kann nicht mehr erreicht werden.

Beim Zurücksetzen können Inkonsistenzen zwischen den eingespielten SESAM-Sicherungsbeständen und den zugehörigen Metadaten im Catalog-Space auftreten. In diesem Fall passt SESAM/SQL die Daten des zurückgesetzten Anwender-Space an die aktuellen Metadaten an. Wie SESAM/SQL diese Anpassung im Einzelnen vornimmt, ist im „[Basishandbuch](#)“ beschrieben.

**ACHTUNG!**

Die Daten einer Tabelle können bei RECOVER SPACESET ... TO verloren gehen. Zur Anpassung einer Tabelle oder eines Index an die zugehörigen Metadaten im Catalog-Space werden die Tabelle oder der Index gelöscht und anschließend neu aufgebaut.

Ist der Space zum Zeitpunkt eines RECOVER SPACESET ... TO im Logging und wurden Indizes neu aufgebaut, kommt er in den Zustand „copy pending“.

Gelingt das Zurücksetzen bei einem Teil der Spaces eines Space-Sets nicht, werden diese defekt gesetzt. Der Zustand der Spaces nach einem RECOVER kann in dem View SYS\_SPACE\_PROPERTIES des SYS\_INFO\_SCHEMA abgefragt werden. Die defekt gesetzten Spaces können in einer Space-Liste zusammengefasst und nach Beseitigung der Abbruch-Ursache separat zurückgesetzt werden.

*zeitstempel\_literal*

Zeitstempel, der den Zeitpunkt eines SESAM-Sicherungsbestandes bezeichnet. Mindestens eine der so identifizierten Kopien muss in der Tabelle RECOVERY\_UNITS bekannt sein.

Näheres zum Zeitpunkt einer Kopieerstellung finden Sie bei der Utility-Anweisung „[COPY - SESAM-Sicherungsbestand erstellen](#)“ auf Seite 97.

**PASSWORD** *kennwort*

BS2000-Kennwort. *kennwort* muss als alphanumerisches Literal angegeben werden.

Für die Angabe von *kennwort* gibt es folgende Möglichkeiten:

- 'C'*zeichenkette*"  
*zeichenkette* enthält vier abdruckbare Zeichen.
- 'X'*hex-zeichenkette*"  
*hex-zeichenkette* enthält acht sedezimale Zeichen.
- 'n'  
*n* bezeichnet eine ganze Zahl von - 2147483648 bis + 2147483647.

Mit **PASSWORD** *kennwort* geben Sie das Kennwort für alle SESAM-Sicherungsbestände an, die SESAM/SQL während der Ausführung der RECOVER-Anweisung benötigt.

Die Angabe des BS2000-Kennworts für einen SESAM-Sicherungsbestand ist immer dann erforderlich, wenn nach dem Erstellen des SESAM-Sicherungsbestandes das BS2000-Kennwort der Datenbank geändert wurde, d.h., wenn die BS2000-Kennwörter für Datenbank und SESAM-Sicherungsbestand nicht mehr übereinstimmen.

**SCOPE PENDING**

Bei Angabe dieser Klausel werden Spaces repariert, wenn sie nicht für die Update-Verarbeitung eröffnet werden können, weil sie z.B. zuvor wegen eines Fehlers defekt gesetzt worden sind, eine Inkonsistenz zwischen Catalog und Space festgestellt wird oder ein DMS-Fehler beim Öffnen der Datei gemeldet wird. Auch Spaces, die eine Tabelle enthalten, die im Rahmen einer früheren Fehlerbehandlung als defekt markiert wurde, werden repariert. Es erfolgt keine Prüfung auf formale Korrektheit. Ein Space, der nur Indizes enthält, wird nicht repariert, wenn er einen oder mehrere defekte Indizes enthält.

SCOPE PENDING ist nur bei RECOVER SPACESET ... USING erlaubt.

**NO INDEX**

Mit dieser Klausel werden Indizes als defekt gekennzeichnet und nicht neu aufgebaut, wenn sie durch das Zurücksetzen eines Space ungültig geworden sind.

Indizes werden ungültig, wenn die Basistabelle und ein zugehöriger Index auf unterschiedlichen Anwender-Spaces liegen oder wenn die Tabelle partitioniert ist und nicht alle betroffenen Anwender-Spaces gleichzeitig zurückgesetzt werden.

NO INDEX ist nur bei RECOVER SPACESET ... TO erlaubt.

**RECOVER einer Space-Liste mit SESAM-Sicherungsbeständen bzw. Fremdkopien**


---

```
RECOVER SPACE space, space ... { USING { zeitstempel_literal }
                                { FOREIGN COPY }
                                TO   { zeitstempel_literal }
                                { FOREIGN COPY } }
```

```
[PASSWORD kennwort] [SCOPE PENDING] [NO INDEX]
```

```
zeitstempel_literal ::= TIMESTAMP 'jahr-monat-tag stunde:minute:sekunde'
```

---

**RECOVER SPACE space , space ...**

Namen der Anwender-Spaces, die repariert oder zurückgesetzt werden sollen. Die angegebenen Anwender-Spaces müssen den gleichen Sicherungszeitpunkt haben. Maximal 999 Spacenames können angegeben werden. Als Name für einen Anwender-Space darf nicht „CATALOG“ angegeben werden.

**USING zeitstempel\_literal<sup>1</sup>**

SESAM/SQL repariert die Anwender-Spaces *space, space ...*, von denen jeweils ein SESAM-Sicherungsbestand vorliegt, der zu dem durch *zeitstempel\_literal* identifizierten Zeitpunkt erstellt wurde. Diese SESAM-Sicherungsbestände wurden alle mit derselben COPY-Anweisung erstellt.

Alle benötigten DA-LOG-Dateien müssen vorhanden sein.

Anwender-Spaces werden von der Reparatur ausgenommen, wenn sie nicht in der logischen Datensicherung sind oder die logische Datensicherung seit dem Erstellen des SESAM-Sicherungsbestandes unterbrochen worden ist.

In diesem Fall wird eine Warnung ausgegeben.

**USING FOREIGN COPY**

Die Klausel bewirkt, dass nur die Logginginformationen nachgefahren und keine SESAM-Sicherungsbestände eingespielt werden. Die Dateien der Fremdkopie müssen vor dem Start der RECOVER-Anweisung auf die entsprechenden Spaces kopiert werden. Zum Einspielen der Fremdkopie muss der Space geschlossen sein. Die Erzeugung einer Fremdkopie wird im „[Basishandbuch](#)“ beschrieben. Die Fremdkopien müssen gemeinsam nach der Administrationsanweisung PREPARE-FOREIGN-COPY entstanden sein.

---

<sup>1</sup> TIMESTAMP 'jahr-monat-tag stunde:minute:sekunde' ist im Handbuch „[SQL-Sprachbeschreibung Teil 1: SQL-Anweisungen](#)“ unter „Zeitstempel“ näher erklärt.

**TO** *zeitstempel\_literal*

SESAM/SQL setzt die Anwender-Spaces *space, space ...* auf die durch *zeitstempel\_literal* identifizierten SESAM-Sicherungsbestände zurück. Alle durch *zeitstempel\_literal* identifizierten SESAM-Sicherungsbestände wurden mit derselben COPY-Anweisung erstellt.

Die benötigten SESAM-Sicherungsbestände von *space, space ...* müssen vorhanden und in der Catalog-Tabelle RECOVERY\_UNITS vermerkt sein.

**ACHTUNG!**

Ein RECOVER SPACE ... TO kann nicht rückgängig gemacht werden. Ein neuerer Stand als der durch *zeitstempel\_literal* identifizierte kann nicht mehr erreicht werden.

Es können Inkonsistenzen zwischen einem eingespielten SESAM-Sicherungsbestand für einen Anwender-Space und den zugehörigen Metadaten im Catalog-Space auftreten. In diesem Fall passt SESAM/SQL die Daten im zurückgesetzten Anwender-Space an die Metadaten im Catalog-Space an. Wie SESAM/SQL diese Anpassung im Einzelnen vornimmt, ist im „[Basishandbuch](#)“ beschrieben.

**ACHTUNG!**

Die Daten einer Tabelle können bei RECOVER SPACE ... TO verloren gehen. Zur Anpassung einer Tabelle oder eines Index an die zugehörigen Metadaten im Catalog-Space werden die Tabelle oder der Index gelöscht und anschließend neu aufgebaut.

Ist der Space zum Zeitpunkt eines RECOVER SPACE ... TO im Logging und wurden Indizes neu aufgebaut, kommt er in den Zustand „copy pending“.

Gelingt das Zurücksetzen bei einem Teil der Spaces einer Space-Liste nicht, werden diese defekt gesetzt. Der Zustand der Spaces nach einem RECOVER kann in dem View SYS\_SPACE\_PROPERTIES des SYS\_INFO\_SCHEMA abgefragt werden. Die defekt gesetzten Spaces können erneut in einer Space-Liste zusammengefasst und nach Beseitigung der Abbruch-Ursache separat zurückgesetzt werden.

**TO FOREIGN COPY**

Die Dateien der Fremdkopie müssen vor dem Start der RECOVER-Anweisung auf die entsprechenden Spaces kopiert werden. Zum Einspielen der Fremdkopie muss der Space geschlossen sein. Die Erzeugung einer Fremdkopie wird im „[Basishandbuch](#)“ beschrieben.

**ACHTUNG!**

Es gelten die Hinweise und Warnungen, die zuvor bei RECOVER SPACE ... TO beschrieben sind, in analoger Weise.

*zeitstempel\_literal*

Zeitstempel, der den Zeitpunkt eines SESAM-Sicherungsbestandes bezeichnet. Jede der so identifizierten Kopien muss in der Tabelle RECOVERY\_UNITS bekannt sein.

Näheres zum Zeitpunkt einer Kopieerstellung finden Sie bei der Utility-Anweisung „[COPY - SESAM-Sicherungsbestand erstellen](#)“ auf Seite 97.

PASSWORD *kennwort*

BS2000-Kennwort. *kennwort* muss als alphanumerisches Literal angegeben werden.

Für die Angabe von *kennwort* gibt es folgende Möglichkeiten:

- 'C'*zeichenkette*"  
*zeichenkette* enthält vier abdruckbare Zeichen.
- 'X'*hex-zeichenkette*"  
*hex-zeichenkette* enthält acht sedezimale Zeichen.
- 'n'  
*n* bezeichnet eine ganze Zahl von - 2147483648 bis + 2147483647.

Mit PASSWORD *kennwort* geben Sie das Kennwort für alle SESAM-Sicherungsbestände an, die SESAM/SQL während der Ausführung der RECOVER-Anweisung benötigt.

Die Angabe des BS2000-Kennworts für einen SESAM-Sicherungsbestand ist immer dann erforderlich, wenn nach dem Erstellen des SESAM-Sicherungsbestandes das BS2000-Kennwort der Datenbank geändert wurde, d.h., wenn die BS2000-Kennwörter für Datenbank und SESAM-Sicherungsbestand nicht mehr übereinstimmen.

## SCOPE PENDING

Bei Angabe dieser Klausel werden Spaces repariert, wenn sie nicht für die Update-Verarbeitung eröffnet werden können, weil sie z.B. zuvor wegen eines Fehlers defekt gesetzt worden sind, eine Inkonsistenz zwischen Catalog und Space festgestellt wird oder ein DMS-Fehler beim Öffnen der Datei gemeldet wird. Auch Spaces, die eine Tabelle enthalten, die im Rahmen einer früheren Fehlerbehandlung als defekt markiert wurde, werden repariert. Es erfolgt keine Prüfung auf formale Korrektheit. Ein Space, der nur Indizes enthält, wird nicht repariert, wenn er einen oder mehrere defekte Indizes enthält.

SCOPE PENDING ist bei Fremdkopien und bei RECOVER SPACE ... TO nicht erlaubt.

## NO INDEX

Mit dieser Klausel werden Indizes als defekt gekennzeichnet und nicht neu aufgebaut, wenn sie durch das Zurücksetzen eines Space ungültig geworden sind. Indizes werden ungültig, wenn die Basistabelle und ein zugehöriger Index auf unterschiedlichen Anwender-Spaces liegen oder wenn die Tabelle partitioniert ist und nicht alle betroffenen Anwender-Spaces gleichzeitig zurückgesetzt werden.

NO INDEX ist nur bei RECOVER SPACE ... TO erlaubt.

**RECOVER einer Space-Liste mit Replikaten**


---

```
RECOVER SPACE space, space ... { USING replikart_beschreibung
                                  TO replikart_beschreibung }
```

```
[SCOPE PENDING] [NO INDEX]
```

```
replikart_beschreibung ::=
```

```
  REPLICATION replikart
  [PASSWORD kennwort]
  [ON USER_ID bs2000_benutzerkennung]
  { COPY
    RENAME }
```

---

```
RECOVER SPACE space, space ...
```

Namen der Anwender-Spaces, die repariert oder zurückgesetzt werden sollen.

Alle angegebenen Spaces müssen im Replikart aktuell sein

(siehe Handbuch „[Utility-Monitor](#)“, Maske COP.4.6).

Maximal 999 Spacenames können angegeben werden. Als Name für einen Anwender-Space darf nicht „CATALOG“ angegeben werden.

Voraussetzung für RECOVER SPACE *space* USING:

- Das Replikart muss vorhanden sein.
- Alle im Anschluss an das Replikart erstellten und in der Catalog-Tabelle DA\_LOGS vermerkten DA-LOG-Dateien müssen vorhanden sein.
- Die Spaces müssen im Logging sein, das Logging darf nicht unterbrochen sein.

USING *replikart\_beschreibung*

SESAM/SQL repariert die Anwender-Spaces *space*, *space* ... mit dem angegebenen Replikart.

TO *replikart\_beschreibung*

SESAM/SQL setzt die die Anwender-Spaces *space*, *space* ... auf den Stand des angegebenen Replikarts zurück.

**ACHTUNG!**

Ein RECOVER SPACE ... TO kann nicht rückgängig gemacht werden. Ein neuerer Stand als der durch *replikart\_beschreibung* spezifizierte kann nicht mehr erreicht werden.

Es können Inkonsistenzen zwischen dem eingespielten Replikat-Space und den zugehörigen Metadaten im Catalog-Space auftreten. In diesem Fall passt SESAM/SQL die Daten im zurückgesetzten Anwender-Space an die Metadaten im Catalog-Space an. Wie SESAM/SQL diese Anpassung im Einzelnen vornimmt, ist im „[Basishandbuch](#)“ beschrieben.



### ACHTUNG!

Die Daten einer Tabelle können bei RECOVER SPACE ... TO verloren gehen. Zur Anpassung einer Tabelle oder eines Index an die zugehörigen Metadaten im Catalog-Space werden die Tabelle oder der Index gelöscht und anschließend neu aufgebaut.

RECOVER SPACE ... TO mit einem Replikat führt immer dazu, dass diese Spaces und ggf. abhängige Indexspaces in den Zustand „copy pending“ versetzt werden (siehe auch „[Basishandbuch](#)“, Abschnitt „Replikat einer Datenbank“).

*replikat\_beschreibung* ::= =

Replikat oder Teilreplikat, das für das Reparieren, Zurücksetzen oder Erzeugen eines eigenständigen Originals verwendet werden soll.

REPLICATION *replikat*

*replikat* ist der physikalische Name des Replikats.

PASSWORD *kennwort*

BS2000-Kennwort des Replikats. *kennwort* muss als alphanumerisches Literal angegeben werden.

Für die Angabe von *kennwort* gibt es folgende Möglichkeiten:

- 'C'*zeichenkette*"  
*zeichenkette* enthält vier abdruckbare Zeichen.
- 'X'*hex-zeichenkette*"  
*hex-zeichenkette* enthält acht sedezimale Zeichen.
- 'n'  
*n* bezeichnet eine ganze Zahl von - 2147483648 bis + 2147483647.

ON USER\_ID *bs2000\_benutzerkennung*

Kennung des Replikats. Die Benutzerkennung wird ohne „\$“ angegeben.

## COPY

Die angegebenen Spaces des Replikats werden auf die Spaces des Catalogs kopiert. Müssen die Spaces des Replikats dabei in eine Kennung ungleich der DBH-Kennung kopiert werden, müssen dort alle erforderlichen Dateien bereits angelegt sein. Alternativ kann auch in der DB-Kennung die DBH-Kennung als Miteigentümer für die Dateien des Catalogs definiert werden.

Die Angaben zur Storage Group für den Anwender-Space werden ignoriert. Das Replikat kann weiterhin aktualisiert und für Wiedergewinnungsanwendungen genutzt werden. Das Replikat selbst wird bei diesem RECOVER nicht aktualisiert.



Wird COPY aus einem Grund abgewiesen, der auch ein REFRESH REPLICATION verhindert, so sind die Anwender-Spaces des Replikats weiterhin verwendbar.

Wenn die Ursache behoben ist (z.B. Bereitstellen fehlender Logging-Dateien), kann die Utility-Anweisung wiederholt werden. Kann die Ursache nicht behoben werden (z.B. ein Space ist im Original zurückgesetzt worden), müssen die Spaces mit den SESAM-Sicherungsbeständen repariert oder mit RECOVER SPACE ... TO REPLICATION zurückgesetzt werden.

## RENAME

Die betroffenen Spaces des angegebenen Replikats werden auf die Spaces des Catalogs umbenannt. Diese Dateien dürfen im Original nicht mehr vorhanden sein, weil die Umbenennung sonst nicht möglich ist. Nach der Umbenennung fehlen die Spaces im Replikat. Sie können mit REFRESH SPACE wieder hergestellt werden. RENAME kann nur genutzt werden, wenn Replikat und Catalog in derselben BS2000-Kennung liegen. Ist diese Kennung ungleich der DBH-Kennung, so muss die DBH-Kennung als Miteigentümer für alle Dateien des Replikats und Catalogs in der DB-Kennung definiert sein. Das Replikat darf in keinem SQL-Datenbankverzeichnis eingetragen sein.



Wird RENAME aus einem Grund abgewiesen, der auch ein REFRESH REPLICATION verhindert, so sind die Spaces des Replikats nicht mehr vorhanden. Die Spaces müssen dann mit den SESAM-Sicherungsbeständen repariert werden.

## SCOPE PENDING

Bei Angabe dieser Klausel werden Spaces repariert, wenn sie nicht für die Update-Verarbeitung eröffnet werden können, weil sie z.B. zuvor wegen eines Fehlers defekt gesetzt worden sind, eine Inkonsistenz zwischen Catalog und Space festgestellt wird oder ein DMS-Fehler beim Öffnen der Datei gemeldet wird. Auch Spaces, die eine Tabelle enthalten, die im Rahmen einer früheren Fehlerbehandlung als defekt markiert wurde, werden repariert. Es erfolgt keine Prüfung auf formale Korrektheit. Ein Space, der nur Indizes enthält, wird nicht repariert, wenn er einen oder mehrere defekte Indizes enthält.

SCOPE PENDING ist nur bei RECOVER SPACE ... USING erlaubt.

**NO INDEX**

Mit dieser Klausel werden Indizes als defekt gekennzeichnet und nicht neu aufgebaut, wenn sie durch das Zurücksetzen eines Space ungültig geworden sind. Indizes werden ungültig, wenn die Basistabelle und ein zugehöriger Index auf unterschiedlichen Anwender-Spaces liegen oder wenn die Tabelle partitioniert ist und nicht alle betroffenen Anwender-Spaces gleichzeitig zurückgesetzt werden.

NO INDEX ist nur bei RECOVER SPACE ... TO erlaubt.

Weitere Informationen zum Konzept und der Anwendung finden Sie im [„Basishandbuch“](#).

## RECOVER CATALOG\_SPACE

Die Anweisung RECOVER CATALOG\_SPACE darf nur auf bestimmten System-Benutzerkennungen eingegeben werden. Diese Benutzerkennungen legt der SESAM/SQL-Systemverwalter mit der DBH-Option ADMINISTRATOR bzw. mit der Administrationsanweisung MODIFY-ADMINISTRATION fest.

In diesem Abschnitt werden folgende RECOVER-Anweisungen beschrieben:

- RECOVER eines Catalog-Space mit SESAM-Sicherungsbeständen bzw. Fremdkopien
- RECOVER eines Catalog-Space mit Replikaten

### ***RECOVER CATALOG\_SPACE mit SESAM-Sicherungsbeständen bzw. Fremdkopien***

---

```
RECOVER CATALOG_SPACE catalog [ { USING rec_unit }
                                { TO rec_unit } ]
```

```
[PASSWORD kennwort]
```

```
rec_unit ::= { COPY_FILE datei }
              { COPY_NUMBER zahl }
              { zeitstempel_literal }
              { FOREIGN COPY }
```

```
zeitstempel_literal ::= TIMESTAMP 'jahr-monat-tag stunde:minute:sekunde'
```

---

### RECOVER CATALOG\_SPACE *catalog*

Name der Datenbank, deren Catalog-Space repariert werden soll.

Bei RECOVER CATALOG\_SPACE *catalog* und RECOVER CATALOG\_SPACE *catalog* USING müssen für den Catalog-Space die folgenden Bedingungen erfüllt sein:

- Die CAT-REC-Datei muss vorhanden sein.
- Bei einer SESAM-Sicherung muss der benötigte SESAM-Sicherungsbestand des Catalog-Space vorhanden und in der CAT-REC-Datei eingetragen sein.
- Alle im Anschluss an den benötigten SESAM-Sicherungsbestand bzw. die Fremdkopie des Catalog-Space erstellten und in der CAT-REC-Datei eingetragenen CAT-LOG-Dateien müssen vorhanden sein.

- Soll eine Fremdkopie verwendet werden, muss der Aufsetzpunkt in der CAT-REC-Datei noch vorhanden sein. Es kann jeder Stand der CAT-REC-Datei verwendet werden, der bei oder nach dem Erzeugen der Fremdkopie des Catalog-Space entstanden ist. Die Reparatur erfolgt auf den Stand der CAT-REC-Datei.

#### RECOVER CATALOG\_SPACE *catalog* ohne USING oder TO

SESAM/SQL verwendet für die Reparatur des Catalog-Space den durch den letzten Eintrag in der CAT-REC-Datei identifizierten SESAM-Sicherungsbestand (d.h. den zuletzt erstellten SESAM-Sicherungsbestand) des Catalog-Space und die daran anschließenden CAT-LOG-Dateien.

#### USING *rec\_unit*

SESAM/SQL verwendet für die Reparatur den durch *rec\_unit* identifizierten SESAM-Sicherungsbestand bzw. die Fremdkopie des Catalog-Space und die daran anschließenden CAT-LOG-Dateien.

Der Catalog-Space kann nur mit der aktuellen CAT-REC-Datei repariert werden. Ein RECOVER CATALOG\_SPACE ... USING auf Basis einer älteren CAT-REC-Datei entspricht einem Zurücksetzen des Catalog-Space auf den Stand der verwendeten CAT-REC-Datei.

Soll eine Fremdkopie verwendet werden, muss der Catalog-Space beim Erzeugen dieser Fremdkopie im Logging gewesen sein. Der Aufsetzpunkt in der CAT-REC-Datei muss noch vorhanden sein. Seit dem Erzeugen der Fremdkopie darf das Logging nicht unterbrochen worden sein.

War der Catalog-Space nicht im Logging, als die Fremdkopie erzeugt wurde, ist nur RECOVER CATALOG\_SPACE ... TO FOREIGN COPY möglich.

#### TO *rec\_unit*

SESAM/SQL setzt auf die angegebene Sicherung des Catalog-Space zurück.



#### **ACHTUNG!**

Wird der Catalog-Space separat zurückgesetzt, kann in der Regel auf die Anwender-Spaces nicht mehr zugegriffen werden, weil ihr Änderungszeitstempel nicht mehr mit dem im Catalog-Space vermerkten übereinstimmt.

Die Anwender-Spaces bleiben bei RECOVER CATALOG\_SPACE ... TO unverändert und müssen einzeln repariert werden.



#### **ACHTUNG!**

Ein RECOVER TO kann nicht rückgängig gemacht werden. Alle auf den mit *rec\_unit* angesprochenen Datensatz folgenden Datensätze werden in der CAT-REC-Datei gelöscht. Ein neuerer Stand als der durch *rec\_unit* identifizierte kann nicht mehr erreicht werden. Sie können jedoch vor dem Zurücksetzen die CAT-REC-Datei, die CAT-LOG-Dateien und die DA-LOG-Dateien außerhalb von SESAM/SQL sichern.

*rec\_unit ::=*

Sicherungsbestand, der für Reparatur und Zurücksetzen verwendet werden soll. Neben SESAM-Sicherungsbeständen ist die Verwendung einer Fremdkopie möglich.

*COPY\_FILE datei*

Dateiname eines mit der Utility-Anweisung COPY erstellten SESAM-Sicherungsbestandes. Die Datei muss in der CAT-REC-Datei eingetragen sein.

*datei* muss als alphanumerisches Literal angegeben werden.

*COPY\_NUMBER zahl*

Versionsnummer eines mit der Utility-Anweisung COPY erstellten SESAM-Sicherungsbestandes. *zahl* bezeichnet eine maximal 6-stellige ganze Zahl.

Der durch *zahl* identifizierte SESAM-Sicherungsbestand muss in der CAT-REC-Datei eingetragen sein.

*zeitstempel\_literal*<sup>1</sup>

Zeitstempel, der den Zeitpunkt eines SESAM-Sicherungsbestandes bezeichnet. Die Kopie muss in der CAT-REC-Datei eingetragen sein.

Näheres zum Zeitpunkt einer Kopiererstellung finden Sie bei der Utility-Anweisung [„COPY - SESAM-Sicherungsbestand erstellen“ auf Seite 97](#).

FOREIGN COPY

Die Klausel bewirkt, dass kein SESAM-Sicherungsbestand eingespielt wird. Bei USING werden nur die Logginginformationen nachgefahren. Die Dateien der Fremdkopie müssen vor dem Start der RECOVER-Anweisung auf die entsprechenden Spaces kopiert werden. Zum Einspielen der Fremdkopie muss die Datenbank geschlossen sein.

Die Erzeugung einer Fremdkopie wird im [„Basishandbuch“](#) beschrieben.

PASSWORD *kennwort*

BS2000-Kennwort. *kennwort* muss als alphanumerisches Literal angegeben werden.

Für die Angabe von *kennwort* gibt es folgende Möglichkeiten:

- 'C'*zeichenkette*"  
*zeichenkette* enthält vier abdruckbare Zeichen.
- 'X'*hex-zeichenkette*"  
*hex-zeichenkette* enthält acht sedezimale Zeichen.
- 'n'  
*n* bezeichnet eine ganze Zahl von - 2147483648 bis + 2147483647.

Mit PASSWORD *kennwort* geben Sie das Kennwort für alle SESAM-Sicherungsbestände an, die SESAM/SQL während der Ausführung der RECOVER-Anweisung benötigt.

<sup>1</sup> TIMESTAMP 'jahr-monat-tag stunde:minute:sekunde' ist im Handbuch [„SQL-Sprachbeschreibung Teil 1: SQL-Anweisungen“](#) unter „Zeitstempel“ näher erklärt.

Die Angabe des BS2000-Kennworts für einen SESAM-Sicherungsbestand ist immer dann erforderlich, wenn nach dem Erstellen des SESAM-Sicherungsbestandes das BS2000-Kennwort der Datenbank geändert wurde, d.h., wenn die BS2000-Kennwörter für Datenbank und SESAM-Sicherungsbestand nicht mehr übereinstimmen.

### ***RECOVER CATALOG\_SPACE mit Replikaten***

---

```
RECOVER CATALOG_SPACE catalog { USING replikat_beschreibung
                                TO replikat_beschreibung }
```

*replikat\_beschreibung* ::=

```
REPLICATION replikat
[PASSWORD kennwort]
[ON USER_ID bs2000_benutzerkennung]
[WITH CATREC catrec_datei
 [PASSWORD kennwort] [ON USER_ID bs2000_benutzerkennung]]
{ COPY
  RENAME }
```

---

RECOVER CATALOG\_SPACE *catalog*

Name der Datenbank, deren Catalog-Space mit Hilfe des Replikats neu erzeugt, zurückgesetzt oder repariert werden soll. Für *catalog* muss ein Eintrag im SQL-Datenbankverzeichnis existieren.

Voraussetzung für RECOVER CATALOG\_SPACE *catalog* USING:

- Die CAT-REC-Datei muss vorhanden sein.
- Das Replikat muss vorhanden sein.
- Alle im Anschluss an das Replikat des Catalog-Space erstellten und in der CAT-REC-Datei eingetragenen CAT-LOG-Dateien müssen vorhanden sein.
- Der Catalog muss im Logging sein.

USING *replikat\_beschreibung*

SESAM/SQL bringt den Catalog-Space auf den Stand, der in der angegebenen CAT-REC-Datei beschrieben ist.

Die Klausel WITH CATREC muss angegeben werden.

TO *replikat\_beschreibung*

SESAM/SQL setzt den Catalog-Space auf den Stand des angegebenen Replikats zurück bzw. erzeugt die Datenbank mit diesem Stand neu.

Die Klausel WITH CATREC darf nicht angegeben werden.

**ACHTUNG!**

Wenn der Catalog-Space neu erzeugt oder zurückgesetzt wird, fehlen noch die Spaces bzw. die Spaces passen im Allgemeinen nicht mehr zum Original. Sie können mit RECOVER SPACE *space-liste* TO REPLICATION (wieder) hergestellt werden.

*replikat\_beschreibung* ::= =

Replikat oder Teilreplikat, das für das Reparieren oder Zurücksetzen verwendet werden soll.

REPLICATION *replikat*

*replikat* ist der physikalische Name des Replikats.

PASSWORD *kennwort*

BS2000-Kennwort des Replikats. *kennwort* muss als alphanumerisches Literal angegeben werden.

Für die Angabe von *kennwort* gibt es folgende Möglichkeiten:

- 'C'*zeichenkette*"  
*zeichenkette* enthält vier abdruckbare Zeichen.
- 'X'*hex-zeichenkette*"  
*hex-zeichenkette* enthält acht sedezimale Zeichen.
- 'n'  
*n* bezeichnet eine ganze Zahl von - 2147483648 bis + 2147483647.

ON USER\_ID *bs2000\_benutzerkennung*

Kennung des Replikats. Die Benutzerkennung wird ohne „\$“ angegeben.

WITH CATREC *catrec\_datei*

CAT-REC-Datei oder CAT-REC-Kopie des Originals. Dabei darf keine BS2000-Benutzerkennung angegeben werden. Die angegebene Datei muss von der Datenbank stammen, aus der das Replikat erzeugt wurde und darf nicht älter sein als das Replikat nach dem letzten REFRESH REPLICATION.

PASSWORD *kennwort*

BS2000-Kennwort der CAT-REC-Datei. *kennwort* muss als alphanumerisches Literal angegeben werden (siehe oben bei „PASSWORD *kennwort*“).

ON USER\_ID *bs2000\_benutzerkennung*

Kennung, auf der CAT-REC- und Logging-Dateien liegen. Die Benutzerkennung wird ohne „\$“ angegeben.

## COPY

Der Catalog-Space des angegebenen Replikats wird auf die Datei des Catalog-Space kopiert. Liegt der Catalog-Space nicht in der DBH-Kennung, muss die Datei bereits angelegt sein. Alternativ kann auch in der DB-Kennung die DBH-Kennung als Miteigentümer für die Dateien des Catalogs definiert werden.

Das Replikat darf im SQL-Datenbankverzeichnis nicht mit STATUS=ACTIVE eingetragen sein.

Das Replikat kann weiterhin aktualisiert und für Wiedergewinnungsanwendungen genutzt werden. Das Replikat selbst wird bei diesem RECOVER nicht aktualisiert. Die Angaben zur Storage Group für den Catalog-Space und die Anwender-Spaces werden ignoriert.



Wird COPY aus einem Grund abgewiesen, der auch ein REFRESH REPLICATION verhindert, so ist das Replikat weiterhin verwendbar. Wenn die Ursache behoben ist (z.B. Bereitstellen fehlender Logging-Dateien), kann die Utility-Anweisung wiederholt werden. Kann die Ursache nicht behoben werden (z.B. ein Space ist im Original zurückgesetzt worden), muss die Datenbank mit den SESAM-Sicherungsbeständen repariert werden oder kann mit RECOVER CATALOG\_SPACE ... TO REPLICATION zurückgesetzt werden.

## RENAME

Der Catalog-Space des angegebenen Replikats wird auf den Namen des Catalog-Space umbenannt. Der Catalog-Space des Originals darf nicht mehr vorhanden sein, weil sonst keine Umbenennung möglich ist.

Nach der Umbenennung ist der ursprüngliche Catalog-Space des Replikats nicht mehr vorhanden und das gesamte Replikat muss mit CREATE REPLICATION neu erstellt werden.

RENAME kann nur genutzt werden, wenn Replikat und Catalog in derselben BS2000-Kennung liegen. Ist diese Kennung ungleich der DBH-Kennung, so muss die DBH-Kennung als Miteigentümer für alle Dateien des Replikats und Catalogs in der DB-Kennung definiert sein. Das Replikat darf in keinem SQL-Datenbankverzeichnis eingetragen sein.



Wird RENAME aus einem Grund abgewiesen, der auch ein REFRESH REPLICATION verhindert, so ist der Catalog-Space des Replikats nicht mehr vorhanden und das Replikat damit defekt. Die Datenbank muss dann mit den SESAM-Sicherungsbeständen repariert und das Replikat neu erzeugt werden.

## RECOVER CATALOG

Die Anweisung RECOVER CATALOG darf nur auf bestimmten System-Benutzerkennungen eingegeben werden. Diese Benutzerkennungen legt der SESAM/SQL-Systemverwalter mit der DBH-Option ADMINISTRATOR bzw. mit der Administrationsanweisung MODIFY-ADMINISTRATION fest.

In diesem Abschnitt werden folgende RECOVER-Anweisungen beschrieben:

- RECOVER eines Catalog mit SESAM-Sicherungsbeständen bzw. Fremdkopien
- RECOVER eines Catalog mit Replikaten

### ***RECOVER CATALOG mit SESAM-Sicherungsbeständen bzw. Fremdkopien***

---

```
RECOVER CATALOG catalog [ { USING rec_unit
                           TO rec_unit
                           [USING rec_unit] TO [ANY] zeitstempel_literal } ]
```

```
[PASSWORD kennwort] [SCOPE PENDING] [GENERATE INDEX ON NO LOG INDEX SPACE]
```

```
rec_unit ::= { COPY_FILE datei
                COPY_NUMBER zahl
                zeitstempel_literal
                FOREIGN COPY }
```

```
zeitstempel_literal ::= TIMESTAMP 'jahr-monat-tag stunde:minute:sekunde'
```

---

### RECOVER CATALOG *catalog*

Name der Datenbank, die als Ganzes mit allen Anwender-Spaces repariert werden soll.

Für RECOVER CATALOG gelten folgende Voraussetzungen, wenn nicht mit RECOVER CATALOG *catalog* TO *rec\_unit* auf eine Datenbank-Gesamtsicherung zurückgesetzt wird:

- Die CAT-REC-Datei muss vorhanden sein.
- Bei einer SESAM-Sicherung muss der benötigte SESAM-Sicherungsbestand des Catalog-Space vorhanden und in der CAT-REC-Datei vermerkt sein.
- Alle im Anschluss an den benötigten SESAM-Sicherungsbestand bzw. die Fremdkopie des Catalog-Space erstellten und in der CAT-REC-Datei vermerkten CAT-LOG-Dateien müssen vorhanden sein.

- Bei einer SESAM-Sicherung müssen die benötigten und in der Catalog-Tabelle RECOVERY\_UNITS vermerkten SESAM-Sicherungsbestände aller Anwender-Spaces vorhanden sein.
- Alle im Anschluss an die benötigten SESAM-Sicherungsbestände bzw. die Fremdkopie der einzelnen Anwender-Spaces erstellten und in der Catalog-Tabelle DA\_LOGS vermerkten DA-LOG-Dateien müssen vorhanden sein.
- Soll eine Fremdkopie verwendet werden, muss der Aufsetzpunkt in der CAT-REC-Datei noch vorhanden sein. Es kann jeder Stand der CAT-REC-Datei verwendet werden, der bei oder nach dem Erzeugen der Fremdkopie des Catalog-Space entstanden ist. Die Reparatur erfolgt auf den Stand der CAT-REC-Datei.
- Für die Anwender-Spaces darf seit dem Erstellen des benötigten SESAM-Sicherungsbestandes bzw. der Fremdkopie das Logging nicht unterbrochen worden sein (z.B. durch LOAD OFFLINE, MIGRATE).

Spaces, die nicht der logischen Datensicherung unterliegen, sind nach einem RECOVER CATALOG unverändert. Sie müssen anschließend gesondert wiederhergestellt werden. Eine Ausnahme bilden Index-Spaces. Sie können mit GENERATE INDEX ON NO LOG INDEX SPACE wiederhergestellt werden. Die Indizes werden dabei neu aufgebaut.

RECOVER CATALOG *catalog* ohne USING oder TO

SESAM/SQL repariert zunächst den Catalog-Space der Datenbank *catalog* auf der Basis des zuletzt erstellten SESAM-Sicherungsbestandes des Catalog-Space und der CAT-LOG-Dateien, die an diesen SESAM-Sicherungsbestand anschließen. Nach erfolgter Reparatur des Catalog-Space repariert SESAM/SQL alle Anwender-Spaces, für die das Logging eingeschaltet ist, auf der Basis des jeweils zuletzt erstellten SESAM-Sicherungsbestandes. Alle Anwender-Spaces, für die kein Logging eingeschaltet ist, werden nicht behandelt und müssen anschließend einzeln repariert oder zurückgesetzt werden, falls sie defekt waren.

USING *rec\_unit*

SESAM/SQL repariert den Catalog-Space der Datenbank *catalog* auf der Basis des durch *rec\_unit* identifizierten SESAM-Sicherungsbestandes bzw. der Fremdkopie des Catalog-Space und der CAT-LOG-Dateien, die an diesen Sicherungsbestand anschließen.

Die darauf folgende Reparatur der Anwender-Spaces verläuft analog der Reparatur im Fall RECOVER CATALOG *catalog* ohne USING-Klausel. Bei einer Fremdkopie (*rec\_unit* = FOREIGN\_COPY) werden aber keine Anwender-Spaces eingespielt.

**TO *rec\_unit***

SESAM/SQL setzt den Catalog-Space auf den Stand der durch *rec\_unit* angegebenen Sicherung zurück. Anschließend werden die letzten im Catalog-Space verzeichneten Sicherungen der Anwender-Spaces eingespielt.

Wird eine Sicherung verwendet, bei der die Anwender-Spaces vor dem Catalog-Space gesichert wurden, dann werden gegebenenfalls Änderungen auf den Spaces nachgefahren. Wurden die Anwender-Spaces bei einer Fremdkopie zu einem späteren Zeitpunkt als der Catalog-Space gesichert, dann ist ihre Aufsetzinformation in der DA\_LOGS-Tabelle nicht vorhanden. Sie können nicht repariert werden.

**TO ANY *zeitstempel\_literal***<sup>1</sup>

SESAM/SQL setzt die gesamte Datenbank auf den Stand des angegebenen Zeitpunktes zurück. Transaktionen, die zu diesem Zeitpunkt offen waren, werden zurückgesetzt. Der angegebene Zeitpunkt ist frei wählbar. Er darf aber nicht vor dem Erzeugen des Catalogs (CREATE CATALOG) liegen. Er darf auch nicht in der kritischen Stunde der Umstellung von Sommer- auf Winterzeit liegen.

SESAM/SQL spielt den letzten, vor dem Zeitpunkt liegenden SESAM-Sicherungsbestand des Catalog-Space ein und fährt die CAT-LOG-Dateien, die an diesen SESAM-Sicherungsbestand anschließen, bis zum angegebenen Zeitpunkt nach. Anschließend werden die letzten im Catalog-Space verzeichneten Sicherungen der Anwender-Spaces eingespielt. Die Änderungen auf diesen Spaces werden mit Hilfe der DA-LOG-Dateien ebenfalls bis zum angegebenen Zeitpunkt nachgefahren.

Alle Anwender-Spaces, für die kein Logging eingeschaltet ist, werden nicht behandelt und müssen anschließend, falls sie defekt waren, einzeln repariert oder zurückgesetzt werden.

**USING *rec\_unit* TO [ANY] *zeitstempel\_literal***

SESAM/SQL setzt die gesamte Datenbank auf den Stand des angegebenen Zeitpunktes zurück. Transaktionen, die zu diesem Zeitpunkt offen waren, werden zurückgesetzt. Basis ist der durch *rec\_unit* identifizierte SESAM-Sicherungsbestand bzw. die Fremdkopie des Catalog-Space. Bei Angabe von ANY ist der angegebene Zeitpunkt frei wählbar, sonst muss *zeitstempel\_literal* den Zeitpunkt eines SESAM-Sicherungsbestandes des Catalog-Space bezeichnen. Der Zeitpunkt darf aber nicht vor dem Erzeugen des Catalogs (CREATE CATALOG) liegen. Er darf auch nicht in der kritischen Stunde der Umstellung von Sommer- auf Winterzeit liegen. Die bei USING *rec\_unit* angegebene Sicherung muss älter sein als der bei TO [ANY] *zeitstempel\_literal* angegebene Zeitpunkt.

SESAM/SQL spielt den angegebenen SESAM-Sicherungsbestand des Catalog-Space ein und fährt die CAT-LOG-Dateien, die an diesen SESAM-Sicherungsbestand anschließen, bis zum angegebenen Zeitpunkt nach.

<sup>1</sup> **TIMESTAMP** 'jahr-monat-tag stunde:minute:sekunde' ist im Handbuch „SQL-Sprachbeschreibung Teil 1: SQL-Anweisungen“ unter „Zeitstempel“ näher erklärt.

Die darauf folgende Reparatur der Anwender-Spaces verläuft analog der Reparatur im Fall RECOVER CATALOG TO ANY *zeitstempel\_literal* ohne USING-Klausel. Bei einer Fremdkopie (*rec\_unit* = FOREIGN\_COPY) werden aber keine Anwender-Spaces eingespielt.

**ACHTUNG!**

Ein RECOVER TO kann (in allen drei vorangehenden Fällen) nicht rückgängig gemacht werden.

Alle auf den mit *rec\_unit* angesprochenen Datensatz folgenden Datensätze werden in der CAT-REC-Datei gelöscht. Die zugehörigen CAT-LOG-Dateien werden überschrieben. Ein neuerer Stand als der durch *rec\_unit* identifizierte kann nicht mehr erreicht werden.

Sie können jedoch vor dem Zurücksetzen die CAT-REC-Datei, die CAT-LOG-Dateien und die DA-LOG-Dateien außerhalb von SESAM/SQL sichern.

*rec\_unit* ::=

Sicherungsbestand, der für die Reparatur und für das Zurücksetzen verwendet werden soll.

Neben SESAM-Sicherungsbeständen ist die Verwendung von Fremdkopien möglich.

*COPY\_FILE datei*

Dateiname eines mit der Utility-Anweisung COPY erstellten SESAM-Sicherungsbestandes des Catalog-Space. Die Datei muss in der CAT-REC-Datei eingetragen sein. *datei* muss als alphanumerisches Literal angegeben werden.

*COPY\_NUMBER zahl*

Versionsnummer eines mit der Utility-Anweisung COPY erstellten SESAM-Sicherungsbestandes des Catalog-Space. *zahl* bezeichnet eine maximal 6-stellige ganze Zahl. Der durch *zahl* identifizierte SESAM-Sicherungsbestand muss in der CAT-REC-Datei eingetragen sein.

*zeitstempel\_literal*

Zeitstempel, der den Zeitpunkt eines SESAM-Sicherungsbestandes des Catalog-Space bezeichnet. Die Kopie muss in der CAT-REC-Datei eingetragen sein.

Näheres zum Zeitpunkt einer Kopieerstellung finden Sie bei der Utility-Anweisung [„COPY - SESAM-Sicherungsbestand erstellen“ auf Seite 97](#).

**FOREIGN COPY**

Die Klausel bewirkt, dass keine SESAM-Sicherungsbestände eingespielt werden. Die Dateien der Fremdkopie müssen vor der RECOVER-Anweisung auf die entsprechenden Spaces kopiert worden sein. Zum Einspielen der Fremdkopie muss der Catalog geschlossen sein. Bei USING werden noch die Logginginformationen nachgefahren.

Die Erzeugung einer Fremdkopie wird im [„Basishandbuch“](#) beschrieben.

**PASSWORD** *kennwort*

BS2000-Kennwort. *kennwort* muss als alphanumerisches Literal angegeben werden.

Für die Angabe von *kennwort* gibt es folgende Möglichkeiten:

- 'C'*zeichenkette*"  
*zeichenkette* enthält vier abdruckbare Zeichen.
- 'X'*hex-zeichenkette*"  
*hex-zeichenkette* enthält acht sedezimale Zeichen.
- 'n'  
*n* bezeichnet eine ganze Zahl von - 2147483648 bis + 2147483647.

Mit **PASSWORD** *kennwort* geben Sie das Kennwort für alle SESAM-Sicherungsbestände an, die SESAM/SQL während der Ausführung der RECOVER-Anweisung benötigt.

Die Angabe des BS2000-Kennworts für einen SESAM-Sicherungsbestand ist immer dann erforderlich, wenn nach dem Erstellen des SESAM-Sicherungsbestandes das BS2000-Kennwort der Datenbank geändert wurde, d.h., wenn die BS2000-Kennwörter für Datenbank und SESAM-Sicherungsbestand nicht mehr übereinstimmen.

**SCOPE PENDING**

Bei Angabe dieser Klausel wird zunächst der Catalog-Space repariert. Die Spaces werden repariert, wenn sie nicht für die Update-Verarbeitung eröffnet werden können, weil sie z.B. zuvor wegen eines Fehlers defekt gesetzt worden sind, eine Inkonsistenz zwischen Catalog und Space festgestellt wird oder ein DMS-Fehler beim Öffnen der Datei gemeldet wird. Auch Spaces, die eine Tabelle enthalten, die im Rahmen einer früheren Fehlerbehandlung als defekt markiert wurde, werden repariert. Es erfolgt keine Prüfung auf formale Korrektheit. Ein Space, der nur Indizes enthält, wird nicht repariert, wenn er einen oder mehrere defekte Indizes enthält.

**ACHTUNG!**

Wird **SCOPE PENDING** bei **RECOVER CATALOG TO** angegeben, dann wird der Catalog-Space immer zurückgesetzt. Mit den Anwender-Spaces wird so verfahren wie bei **RECOVER SPACE TO** beschrieben.

**SCOPE PENDING** ist bei Fremdkopien nicht erlaubt.

**GENERATE INDEX ON NO LOG INDEX SPACE**

Es werden keine Sicherungen eingespielt von Spaces, auf denen ausschließlich Indizes liegen und die nicht in der logischen Datensicherung sind. Diese Spaces werden zurückgesetzt und die Indizes neu aufgebaut.

Ist gleichzeitig **SCOPE PENDING** angegeben, werden Indizes nur aufgebaut, wenn der Index-Space nicht geöffnet werden kann. Defekte Indizes auf dem Space werden toleriert und führen nicht zu einem Neuaufbau des Index-Space.

**RECOVER CATALOG mit Replikaten**


---

```
RECOVER CATALOG catalog { USING replikat_beschreibung }
                        { TO replikat_beschreibung }
```

```
[GENERATE INDEX ON NO LOG INDEX SPACE]
```

```
replikat_beschreibung ::=
```

```
  REPLICATION replikat
  [PASSWORD kennwort]
  [ON USER_ID bs2000_benutzerkennung]
  [WITH CATREC catrec_datei
   [PASSWORD kennwort] [ON USER_ID bs2000_benutzerkennung]]
  { COPY }
  { RENAME }
```

---

**RECOVER CATALOG *catalog***

Name der Datenbank, die mit Hilfe des (Teil-)Replikats ganz oder teilweise neu erzeugt, zurückgesetzt oder repariert werden soll. Für *catalog* muss ein Eintrag im SQL-Datenbankverzeichnis existieren.

Voraussetzung für RECOVER CATALOG *catalog* USING:

- Die CAT-REC-Datei muss vorhanden sein.
- Das Replikat muss vorhanden sein.
- Alle im Anschluss an das Replikat des Catalog-Space erstellten und in der CAT-REC-Datei vermerkten CAT-LOG-Dateien müssen vorhanden sein.
- Alle im Anschluss an ein Replikat erstellten und in der Catalog-Tabelle DA\_LOGS vermerkten DA-LOG-Dateien müssen vorhanden sein.
- Wird die Klausel USING genutzt, um verbleibende Logging-Dateien nachzufahren, gelten die Einschränkungen von REFRESH REPLICATION (siehe [Seite 219](#)).

USING *replikat\_beschreibung*

SESAM/SQL bringt die Datenbank auf den Stand, der in der angegebenen CAT-REC-Datei beschrieben ist.

Die Klausel WITH CATREC muss angegeben werden.

**TO** *replikat\_beschreibung*

SESAM/SQL setzt die Datenbank auf den Stand des angegebenen Replikats zurück bzw. erzeugt die Datenbank mit diesem Stand neu.

Die Klausel WITH CATREC darf nicht angegeben werden.

**ACHTUNG!**

Ein RECOVER TO kann nicht rückgängig gemacht werden.

Alle auf den mit *rec\_unit* angesprochenen Datensatz folgenden Datensätze werden in der CAT-REC-Datei gelöscht. Die zugehörigen CAT-LOG-Dateien werden überschrieben. Ein neuerer Stand als der durch *rec\_unit* identifizierte kann nicht mehr erreicht werden.

Sie können jedoch vor dem Zurücksetzen die CAT-REC-Datei, die CAT-LOG-Dateien und die DA-LOG-Dateien außerhalb von SESAM/SQL sichern.

*replikat\_beschreibung* ::= =

Replikat oder Teilreplikat, das für das Reparieren, Zurücksetzen oder Erzeugen eines eigenständigen Originals verwendet werden soll. Bei einem Teilreplikat werden nur die Anwender-Spaces repariert, zurückgesetzt oder erzeugt, die im Teilreplikat aktuell sind (siehe Handbuch „[Utility-Monitor](#)“, Maske COP.4.6). Die anderen Spaces müssen mit RECOVER SPACE zurückgesetzt oder repariert werden.

REPLICATION *replikat*

*replikat* ist der physikalische Name des Replikats.

PASSWORD *kennwort*

BS2000-Kennwort des Replikats. *kennwort* muss als alphanumerisches Literal angegeben werden.

Für die Angabe von *kennwort* gibt es folgende Möglichkeiten:

- 'C'*zeichenkette*"  
*zeichenkette* enthält vier abdruckbare Zeichen.
- 'X'*hex-zeichenkette*"  
*hex-zeichenkette* enthält acht sedezimale Zeichen.
- 'n'  
*n* bezeichnet eine ganze Zahl von - 2147483648 bis + 2147483647.

ON USER\_ID *bs2000\_benutzerkennung*

Kennung des Replikats. Die Benutzerkennung wird ohne „\$“ angegeben.

WITH CATREC *catrec\_datei*

CAT-REC-Datei oder CAT-REC-Kopie des Originals. Dabei darf keine BS2000-Benutzerkennung angegeben werden. Die angegebene Datei muss von der Datenbank stammen, aus der das Replikat erzeugt wurde und darf nicht älter sein als das Replikat nach dem letzten REFRESH REPLICATION.

**PASSWORD** *kennwort*

BS2000-Kennwort der CAT-REC-Datei. *kennwort* muss als alphanumerisches Literal angegeben werden.

Für die Angabe von *kennwort* gibt es folgende Möglichkeiten:

- 'C'*zeichenkette*"  
*zeichenkette* enthält vier abdruckbare Zeichen.
- 'X'*hex-zeichenkette*"  
*hex-zeichenkette* enthält acht sedezimale Zeichen.
- 'n'  
*n* bezeichnet eine ganze Zahl von - 2147483648 bis + 2147483647.

**ON USER\_ID** *bs2000\_benutzerkennung*

Kennung, auf der CAT-REC- und Logging-Dateien liegen. Die Benutzerkennung wird ohne „\$“ angegeben.

**COPY**

Das angegebene Replikat wird auf die Spaces des (defekten) Catalog kopiert. Muss das Replikat dabei in eine Kennung ungleich der DBH-Kennung kopiert werden, müssen dort alle erforderlichen Dateien bereits angelegt sein. Alternativ kann auch in der DB-Kennung die DBH-Kennung als Miteigentümer für die Dateien des Catalogs definiert werden.

Die Angaben zur Storage Group für die Anwender-Spaces werden ignoriert.

Das Replikat darf im SQL-Datenbankverzeichnis nicht mit STATUS=ACTIVE eingetragen sein.

Das Replikat kann weiterhin aktualisiert und für Wiedergewinnungsanwendungen genutzt werden. Das Replikat selbst wird bei diesem RECOVER nicht aktualisiert.



Wird COPY aus einem Grund abgewiesen, der auch ein REFRESH REPLICATION verhindert, so ist das Replikat weiterhin verwendbar.

Ist die Ursache behoben (z.B. Bereitstellen fehlender Logging-Dateien), so kann die Utility-Anweisung wiederholt werden. Wenn die Ursache nicht behoben werden kann (z.B. ein Space ist im Original zurückgesetzt worden), muss die Datenbank mit den SESAM-Sicherungsbeständen repariert werden oder kann mit RECOVER CATALOG ... TO REPLICATION zurückgesetzt werden.

## RENAME

Das angegebene Replikat wird auf den Namen des Original-Catalogs umbenannt. Das Original darf nicht mehr vorhanden sein, weil sonst keine Umbenennung möglich ist. Nach der Umbenennung ist das ursprüngliche Replikat nicht mehr vorhanden und muss mit CREATE REPLICATION neu erstellt werden.

RENAME kann nur genutzt werden, wenn Replikat und Catalog in derselben BS2000-Kennung liegen. Ist diese Kennung ungleich der DBH-Kennung, so muss die DBH-Kennung als Miteigentümer für alle Dateien des Replikats und Catalogs in der DB-Kennung definiert sein. Das Replikat darf in keinem SQL-Datenbankverzeichnis eingetragen sein.



Wird RENAME aus einem Grund abgewiesen, der auch ein REFRESH REPLICATION verhindert, so ist das Replikat nicht mehr vorhanden. Die Datenbank muss dann mit den SESAM-Sicherungsbeständen repariert werden.

## GENERATE INDEX ON NO LOG INDEX SPACE

Spaces, auf denen ausschließlich Indizes liegen und die nicht in der logischen Datensicherung sind, dürfen nicht Bestandteil des Replikats sein. Die Index-Spaces werden zurückgesetzt und die Indizes neu aufgebaut.

Wurde eine neue eigenständige Datenbank generiert, kann auf die SESAM-Sicherungsbestände der ursprünglichen Datenbank nicht zugegriffen werden. Deshalb müssen in der neuen Datenbank die Storage Group-Angaben neu zugeordnet, eine Kopie der Datenbank erzeugt und die Metadaten der alten SESAM-Sicherungsbestände gelöscht werden.

## Indizes neu aufbauen

Mit RECOVER INDEX können Sie Indizes neu aufbauen und zwar:

- einen einzelnen Index
- alle Indizes, die zu einer bestimmten Basistabelle gehören
- alle Indizes, die zu Basistabellen eines bestimmten Anwender-Space gehören

Enthält ein Anwender-Space, für den Logging eingeschaltet ist, einen Index, der mit RECOVER INDEX neu aufgebaut werden soll, so unterbricht RECOVER INDEX das Logging für diesen Anwender-Space. Nach der Ausführung von RECOVER INDEX befindet sich der Anwender-Space im Zustand „copy pending“ (siehe [Seite 13](#)). Von dem Anwender-Space muss mit COPY ein SESAM-Sicherungsbestand erstellt werden, da der neu aufgebaute Index den Ausgangspunkt für das weitere Logging darstellt.

Zur Ausführung von RECOVER INDEX (ohne die Angabe ON TABLE bzw. ON SPACE) muss der aktuelle Berechtigungsschlüssel das Sonder-Privileg UTILITY besitzen oder Eigentümer des Schemas sein, dem der Index zugeordnet ist.

Zur Ausführung von RECOVER INDEX ON TABLE *tabelle* muss der aktuelle Berechtigungsschlüssel das Sonder-Privileg UTILITY besitzen oder Eigentümer des Schemas sein, dem die Tabelle *tabelle* zugeordnet ist.

Zur Ausführung von RECOVER INDEX ON SPACE *space* muss der aktuelle Berechtigungsschlüssel das Sonder-Privileg UTILITY besitzen oder Eigentümer des Anwender-Space *space* sein.

---


$$\text{RECOVER INDEX } \left\{ \begin{array}{l} \textit{index} \\ \text{ON TABLE } \textit{tabelle} \\ \text{ON SPACE } \textit{space} \end{array} \right\} \left[ \left\{ \begin{array}{l} \text{SCOPE ALL} \\ \text{SCOPE PENDING} \end{array} \right\} \right]$$


---

### RECOVER INDEX *index*

Name des Index, der berücksichtigt werden soll. Der Index muss bereits vorhanden sein.

### ON TABLE *tabelle*

Name der Basistabelle, für die alle auf ihr definierten Indizes berücksichtigt werden sollen.

Bei einer partitionierten Tabelle müssen alle Partitionen verfügbar sein (siehe [Tabelle 3 auf Seite 22](#)).

### ON SPACE *space*

Name des Anwender-Space, für den alle Indizes berücksichtigt werden sollen, die auf Tabellen dieses Anwender-Space definiert sind.

**SCOPE ALL**

SESAM/SQL baut alle berücksichtigten Indizes neu auf.

Wegen des für die Sortierung beim Aufbau der Indizes benötigten Platzbedarfs empfiehlt es sich, SCOPE ALL nur in begründeten Fällen anzugeben.

**SCOPE PENDING**

SESAM/SQL baut von den berücksichtigten Indizes nur diejenigen neu auf, die nicht verfügbar sind.

**Beispiele**

1. Das Beispiel repariert die Datenbank AUFTRAGKUNDEN.

```
RECOVER CATALOG auftragkunden
```

Die Datenbank wird repariert auf der Basis des zuletzt erstellten SESAM-Sicherungsbestandes des Catalog-Space sowie der zuletzt erstellten SESAM-Sicherungsbestände der Anwender-Spaces TABLESPACE und BLOBSPACE.

2. Das Beispiel repariert die Datenbank AUFTRAGKUNDEN.



```
RECOVER CATALOG auftragkunden USING COPY_NUMBER 2
```

Zunächst wird der Catalog-Space auf der Basis des SESAM-Sicherungsbestandes mit der Versionsnummer 2 repariert. Anschließend werden die Anwender-Spaces TABLESPACE und BLOBSPACE repariert auf der Basis des jeweils zuletzt erstellten SESAM-Sicherungsbestandes. Für die Reparatur der Anwender-Spaces werden die Metadaten des reparierten Catalog-Space verwendet. Der Space INDEXSPACE wird nicht repariert, da er nicht in der logischen Datensicherung ist.

3. Das Beispiel setzt die Datenbank AUFTRAGKUNDEN zurück.



```
RECOVER CATALOG auftragkunden TO COPY_NUMBER 2  
GENERATE INDEX ON NO LOG INDEX SPACE
```

Zunächst wird der Catalog-Space auf den SESAM-Sicherungsbestand mit der Versionsnummer 2 zurückgesetzt. Mit Hilfe der Metadaten des zurückgesetzten Catalog-Space werden die Anwender-Spaces TABLESPACE und BLOBSPACE repariert auf der Basis des jeweils zuletzt erstellten SESAM-Sicherungsbestandes.

Für den Space INDEXSPACE werden keine Sicherungen eingespielt, da auf ihm ausschließlich Indizes liegen und er nicht in der logischen Datensicherung ist. Der Space INDEXSPACE wird zurückgesetzt und die Indizes werden neu aufgebaut.

**Siehe auch**

COPY, CREATE/REFRESH REPLICATION, CREATE CATALOG, CREATE/DROP INDEX

## REFRESH REPLICATION - Replikat aktualisieren

Mit REFRESH REPLICATION können Sie Replikate aktualisieren:

- ein komplettes (Teil-)Replikat durch Nachfahren der Logging-Dateien
- eine Auswahl von Spaces eines (Teil-)Replikats durch Nachfahren der Logging-Dateien aktualisieren und das (Teil-)Replikat dabei entsprechend verkleinern

Wenn das Replikat als Ganzes nicht mehr mit REFRESH REPLICATION aktualisiert werden kann (z.B. durch Unterbrechung des Logging, siehe unten), dann muss REFRESH REPLICATION FOR SPACE verwendet werden.

Mit REFRESH REPLICATION FOR SPACE werden der Catalog-Space des Replikats und alle angegebenen Anwender-Spaces des Replikats aktualisiert. Anwender-Spaces des Replikats, die nicht angegeben sind, werden logisch aus dem Replikat entfernt (sogenannte „ehemalige Replikat-Spaces“). Das Replikat wird damit zu einem Teilreplikat des ursprünglichen Replikats (siehe auch „[Basishandbuch](#)“, Abschnitt „Replikat aktualisieren und erweitern“).

Informationen über die aktuellen und ehemaligen Replikat-Spaces erhalten Sie über den Utility-Monitor, Maske COP.4.6 (siehe Handbuch „[Utility-Monitor](#)“).

Für die Aktualisierung eines Replikats gelten folgende Regeln:

- Wenn ein Replikat eine partitionierte Tabelle enthält, dann müssen alle Partitionen der Tabelle im Replikat enthalten sein. Ein Teilreplikat muss immer in sich abgeschlossen sein, d.h. das Teilreplikat muss alle Indizes zu allen Tabellen beinhalten, die zum Teilreplikat gehören.
- Alle Replikat-Spaces müssen im Original mit Logging betrieben werden und das Logging darf nicht unterbrochen worden sein.

Gründe für eine Unterbrechung des Logging sind:

- Explizites Abschalten der logischen Datensicherung durch ALTER SPACE NO LOG
- Implizite Unterbrechung der logischen Datensicherung durch die Utility-Anweisungen LOAD OFFLINE, MIGRATE, IMPORT TABLE, RECOVER INDEX, RECOVER TO
- Wurden in der Original-Datenbank Anwender-Spaces gelöscht (DROP SPACE), so müssen die entsprechenden Replikat-Spaces aus dem Replikat entfernt werden. Dazu werden sie bei REFRESH REPLICATION FOR SPACE nicht angegeben. Dies gilt auch, wenn die Spaces wieder unter dem selben Namen im Original neu angelegt worden sind.

- Spaces der Original-Datenbank dürfen keinen älteren Stand haben, als die des Replikats (z.B. nach Rücksetzen eines Space). Die entsprechenden Replikat-Spaces müssen aus dem Replikat entfernt werden. Dazu werden sie bei REFRESH REPLICATION FOR SPACE nicht angegeben.
- Die CAT-LOG- und DA-LOG-Dateien des Originals müssen ab dem Stand des Replikats vorhanden sein. Der DBH muss darauf zugreifen können. Die Jobvariable *benutzerkennung.SESAM.replikat.NEXT-REPL-LOG* enthält Nummer und Unternummer der jeweils ersten bereitzustellenden Dateien.
- Es muss eine CAT-REC-Datei der Original-Datenbank vorhanden sein, die aktueller ist als die CAT-REC-Datei des Replikats. Diese kann durch COPY CATALOG, COPY CATALOG\_SPACE oder die Administrationsanweisung CHANGE-CATALOG erstellt werden.
- Während der Ausführung der Anweisung ist kein Zugriff auf das Replikat möglich. Es darf auch in keinem SQL-Datenbankverzeichnis eines anderen DBH eingetragen und dort geöffnet sein.

Die Anweisung REFRESH REPLICATION darf nur auf bestimmten System-Benutzerkennungen eingegeben werden. Diese Benutzerkennungen legt der SESAM/SQL-Systemverwalter mit der DBH-Option ADMINISTRATOR bzw. mit der Administrationsanweisung MODIFY-ADMINISTRATION fest.

Bei REFRESH REPLICATION und REFRESH REPLICATION FOR SPACE werden die folgenden Sicherungs-Einheiten des Replikats nacheinander aktualisiert:

- CAT-REC-Datei
- Catalog-Space
- Anwender-Spaces

Während einer Anweisung REFRESH REPLICATION oder REFRESH REPLICATION FOR SPACE gelten folgende Randbedingungen:

- Es werden das Replikat mit allen Spaces, die CAT-REC-Datei des Replikats und die CAT-REC-Kopie des Originals exklusiv gesperrt.
- Es ist kein Zugriff auf das Replikat möglich.
- Es werden CDML-Anwendungen mit dem Status 9U/9R abgewiesen.

Bricht die Anweisung wegen fehlender Logging-Dateien ab, so kann die Aktualisierung des Replikats nach Bereitstellung der Dateien durch einen erneuten Aufruf der Anweisung fertig gestellt werden. Bricht die Anweisung ab, weil der DBH beendet wird oder der DBH einen internen Wiederanlauf durchführt, so muss das Replikat in der Regel neu erzeugt werden.

---

```
REFRESH REPLICATION replikat [FOR SPACE space[, space ...]]
FROM CATREC catrec_datei
[PASSWORD kennwort]
[ON USER_ID bs2000_benutzerkennung]
```

---

REFRESH REPLICATION *replikat*

Name des Replikats, das aktualisiert werden soll.

FOR SPACE *space*

Name bzw. Namen der Anwender-Spaces des Replikats, die aktualisiert werden sollen.

Bei Angabe einer Liste darf jeder angegebene Name nur einmal vorhanden sein.

Maximal 999 Spacenames können angegeben werden. Der Catalog-Space darf nicht als Anwender-Space angegeben werden.

Anwender-Spaces, die im Replikat vorhanden sind und bei FOR SPACE nicht angegeben werden, werden logisch aus dem Replikat entfernt.

FROM CATREC *catrec\_datei*

Name der CAT-REC-Datei oder CAT-REC-Kopie, die den Stand beschreibt, bis zu dem das Replikat aktualisiert werden soll. Der Name darf keine BS2000-Benutzerkennung enthalten.

Die CAT-REC-Datei muss von derselben Original-Datenbank stammen, von der das Replikat erzeugt wurde. Sie kann beim CHANGE-CATLOG oder beim COPY erzeugt worden sein oder mit dem BS2000-Kommando COPY-FILE.

Wenn *catalog.CAT-REC[.COPY]* als Name der CAT-REC-Datei angegeben wird, dann müssen auch die Logging-Dateien unter den Namen *catalog.version.C.nnnn* und *catalog.version.D.nnnn* zu finden sein

Wenn *catalog.CAT-REC* (ohne *.COPY*) angegeben wird, dann darf die Original-Datenbank nicht geöffnet sein

**PASSWORD** *kennwort*

BS2000-Kennwort für die CAT-REC-Datei. *kennwort* muss als alphanumerisches Literal angegeben werden.

Für die Angabe von *kennwort* gibt es folgende Möglichkeiten:

- 'C'*zeichenkette*"  
*zeichenkette* enthält vier abdruckbare Zeichen.
- 'X'*hex-zeichenkette*"  
*hex-zeichenkette* enthält acht sedezimale Zeichen.
- 'n'  
*n* bezeichnet eine ganze Zahl von - 2147483648 bis + 2147483647.

*kennwort* gilt für die CAT-REC-Datei sowie für die dazu gehörenden CAT-LOG- und DA-LOG-Dateien der Original-Datenbank.

**ON USER\_ID** *bs2000\_benutzerkennung*

Name der BS2000-Benutzerkennung für die CAT-REC-Kopie, sofern sie nicht in der DBH-Kennung liegt. In dieser Kennung müssen auch die CAT-LOG- und DA-LOG-Dateien liegen. Die Dateien müssen shareable und mit Leserecht versehen sein. Die Benutzerkennung wird ohne „\$“ angegeben.

Befindet sich die CAT-REC-Kopie in der DBH-Kennung, so wird keine Benutzerkennung angegeben.

**Beispiele**

1. Das Beispiel aktualisiert das Replikat AUFTRAGREP mit dem aktuellen Stand der Datenbank AUFTRAGKUNDEN.



```
REFRESH REPLICATION auftragrep
FROM CATREC 'AUFTRAGKUNDEN.CAT-REC.COPY'
```

2. Das Beispiel aktualisiert den Anwender-Space TABLESPACE im Replikat AUFTRAGREP032 mit dem aktuellen Stand der Datenbank AUFTRAGKUNDEN. Der einzige weitere Replikat-Space BLOBSpace wird logisch aus dem Replikat entfernt, weil er nicht angegeben ist.



```
REFRESH REPLICATION auftragrep032 FOR SPACE tablespace
FROM CATREC 'AUFTRAGKUNDEN.CAT-REC.COPY'
```

**Siehe auch**

CREATE REPLICATION, REFRESH SPACE, RECOVER USING/TO REPLICATION

## REFRESH SPACE - Replikat erweitern

Mit REFRESH SPACE können Sie Replikat-Spaces aus einer Sicherungskopie neu erzeugen und durch Nachfahren der Logging-Dateien auf den aktuellen Stand des Replikats bringen.

Wenn Replikat-Spaces aus dem Replikat entfernt wurden (z.B. nach einer vorübergehenden Unterbrechung des Logging im Original) oder im Original neu erzeugt wurden, dann können Sie diese Spaces mit REFRESH SPACE wieder dem (Teil-)Replikat hinzufügen (siehe auch „[Basishandbuch](#)“, Abschnitt „Replikat aktualisieren und erweitern“). REFRESH SPACE prüft nicht, ob angegebene Spaces bereits im Replikat enthalten sind. Bereits vorhandene Spaces werden überschrieben. Wird ein Replikat-Space neu angelegt, so wird das Medium des Catalog-Spaces des Replikats verwendet. Informationen über die aktuellen und ehemaligen Replikat-Spaces erhalten Sie über den Utility-Monitor, Maske COP.4.6 (siehe Handbuch „[Utility-Monitor](#)“).

Für die Erweiterung eines Replikats gelten folgende Regeln:

- Alle Spaces müssen im Original mit Logging betrieben werden und das Logging darf nicht unterbrochen worden sein.
- Der DBH muss auf die verwendeten Sicherungskopien zugreifen können. Das Replikat muss so weit aktualisiert sein, dass die Sicherung dort bekannt ist.
- Werden mehrere Spaces in einer Anweisung angegeben, dann müssen alle Sicherungen den gleichem Zeitstempel besitzen.
- Alle seit dem Zeitpunkt der Sicherung bis zum aktuellen Stand des Replikats entstandenen DA-LOG-Dateien müssen bereitgestellt sein. Der DBH muss darauf zugreifen können. Welche Dateien benötigt werden, kann über die RECOVERY\_UNITS im INFORMATION\_SCHEMA festgestellt werden.
- Während der Anweisungsausführung ist kein Zugriff auf das Replikat möglich. Es darf auch in keinem SQL-Datenbankverzeichnis eines anderen DBH eingetragen und dort geöffnet sein.

Bei REFRESH SPACE wird ein Replikat um die angegebenen Anwender-Spaces erweitert. Der Aktualisierungsstand des Replikats ändert sich dadurch nicht.

Zur Ausführung von REFRESH SPACE muss der aktuelle Berechtigungsschlüssel das Sonder-Privileg UTILITY besitzen oder Eigentümer aller betroffenen Anwender-Spaces sein.

Während einer Anweisung REFRESH SPACE gelten folgende Randbedingungen:

- Es werden das Replikat mit allen Spaces und die CAT-REC-Datei des Replikats exklusiv gesperrt.
- Es ist kein Zugriff auf das Replikat möglich.
- Es werden CDML-Anwendungen mit dem Status 9U/9R abgewiesen.

Bricht eine Anweisung wegen fehlender Logging-Dateien ab, so kann die Erweiterung des Replikats nach Bereitstellung der Dateien durch einen erneuten Aufruf der Anweisung wiederholt werden. Bricht eine Anweisung ab, weil der DBH beendet wird oder der DBH einen internen Wiederanlauf durchführt, so muss das Replikat in der Regel neu erzeugt werden.

---

```
REFRESH SPACE space[, space ...]
```

```
    USING rec_unit
```

```
    [PASSWORD kennwort]
```

```
    [ON USER_ID bs2000_benutzerkennung]
```

```
rec_unit ::= { COPY_FILE datei
                COPY_NUMBER zahl
                zeitstempel_literal
                FOREIGN COPY }
```

```
zeitstempel_literal ::= TIMESTAMP 'jahr-monat-tag stunde:minute:sekunde'
```

---

```
REFRESH SPACE space[, space ...]
```

Name bzw. Namen der Anwender-Spaces, mit denen das Replikat erweitert werden soll. Maximal 999 Spacennamen können angegeben werden.

**USING** *rec\_unit*

Angabe des SESAM-Sicherungsbestandes oder der Fremdkopie, auf deren Basis die Replikate-Spaces neu erzeugt und auf den Stand des Replikats nachgefahren werden sollen.

**COPY\_FILE** *datei*

Dateiname eines mit der Utility-Anweisung COPY erstellten SESAM-Sicherungsbestandes. *datei* muss als alphanumerisches Literal angegeben werden.

Die Kopiedatei muss älter sein als der aktuelle Stand des Replikats.

Die Angabe von COPY\_FILE ist nur beim Erweitern um einen einzelnen Space erlaubt.

**COPY\_NUMBER** *zahl*

Versionsnummer eines mit der Utility-Anweisung COPY erstellten SESAM-Sicherungsbestandes. *zahl* bezeichnet eine maximal 6-stellige ganze Zahl. Die Kopienummer muss in der Tabelle RECOVERY\_UNITS des Replikats bekannt sein. Die Angabe von COPY\_NUMBER ist nur beim Erweitern um einen einzelnen Space erlaubt.

*zeitstempel\_literal*<sup>1</sup>

Zeitstempel, der den Zeitpunkt von SESAM-Sicherungskopien bezeichnet. Die Kopien müssen in der Tabelle RECOVERY\_UNITS des Replikats bekannt sein.

Die Angabe von *zeitstempel\_literal* ist auch beim Erweitern um mehrere Spaces in einer Space-Liste erlaubt. Die Kopien müssen dann den gleichen Zeitstempel besitzen.

Näheres zum Zeitpunkt einer Kopieerstellung finden Sie bei der Utility-Anweisung [„COPY - SESAM-Sicherungsbestand erstellen“ auf Seite 97](#).

**FOREIGN COPY**

Bei der Angabe FOREIGN COPY werden nur Logging-Informationen nachgefahren und keine SESAM-Sicherungsbestände eingespielt. Die Dateien der Fremdkopie muss der Anwender zuvor selber auf die entsprechenden Spaces des Replikats kopiert haben. Zum Einspielen der Fremdkopie muss der Space geschlossen sein.

Die Erzeugung einer Fremdkopie wird im [„Basishandbuch“](#) beschrieben.

Die Fremdkopie muss älter sein als der aktuelle Stand des Replikats.

Die Angabe von FOREIGN COPY ist auch beim Erweitern um mehrere Spaces in einer Space-Liste erlaubt. Die Fremdkopien müssen dann den gleichen Zeitstempel besitzen.

---

<sup>1</sup> `TIMESTAMP 'jahr-monat-tag stunde:minute:sekunde'` ist im Handbuch [„SQL-Sprachbeschreibung Teil 1: SQL-Anweisungen“](#) unter [„Zeitstempel“](#) näher erklärt.

**PASSWORD** *kennwort*

Kennwort , unter dem die SESAM-Sicherung angelegt wurde. *kennwort* muss als alphanumerisches Literal angegeben werden.

Für die Angabe von *kennwort* gibt es folgende Möglichkeiten:

- 'C'*zeichenkette*"  
*zeichenkette* enthält vier abdruckbare Zeichen.
- 'X'*hex-zeichenkette*"  
*hex-zeichenkette* enthält acht sedezimale Zeichen.
- 'n'  
*n* bezeichnet eine ganze Zahl von - 2147483648 bis + 2147483647.

**ON USER\_ID** *bs2000\_benutzerkennung*

Name der BS2000-Benutzerkennung unter der die Sicherung oder eine der benötigten Logging-Dateien liegt. Die Dateien werden zuerst in dieser Kennung, dann in der DBH-Kennung gesucht. Ist diese Klausel nicht angegeben, dann müssen alle Dateien in der DBH-Kennung vorhanden sein.

**Beispiel**

Das Beispiel erzeugt den Replikat-Space BLOBSPACE von AUFTRAGREP032 neu aus der SESAM-Sicherung mit der Versionsnummer 1.



```
REFRESH SPACE auftragrep032.blobspace  
USING COPY_NUMBER 1
```

**Siehe auch**

CREATE/REFRESH REPLICATION, RECOVER USING/TO REPLICATION

## REORG - Spaces und Basistabellen reorganisieren

Mit REORG können Sie den Catalog-Space, einzelne Anwender-Spaces oder auch einzelne Basistabellen auf Anwender-Spaces reorganisieren.

Reorganisation bedeutet, dass SESAM/SQL logisch zusammengehörige Speicherbereiche auch physikalisch zusammenhängend abspeichert. Bei der Reorganisation berücksichtigt SESAM/SQL auch die Freiplatzreservierung, die zuvor in den Anweisungen CREATE CATALOG für den Catalog-Space oder in den SQL-Anweisungen CREATE SPACE bzw. ALTER SPACE für einen Anwender-Space mit der Klausel PCTFREE vereinbart wurde.

REORG STATISTICS (siehe Handbuch „[SQL-Sprachbeschreibung Teil 1: SQL-Anweisungen](#)“) wird durch REORG SPACE mit abgedeckt.

Mit REORG SPACE kann auch die Verlagerung eines Anwender-Space auf eine neue Storage Group veranlasst werden.

Mit REORG ONLINE TABLE kann eine einzelne (partitionierte) Basistabelle oder eine einzelne Partition einer partitionierten Tabelle reorganisiert werden. REORG ONLINE TABLE kann parallel zu anderen Anweisungen auf dieser Tabelle ausgeführt werden und fordert keine exklusiven Sperren an.

## REORG [CATALOG\_]SPACE - Catalog-Space und Anwender-Spaces reorganisieren

Datenbanken in einer DB-Kennung werden behandelt wie Datenbanken in der DBH-Kennung. Voraussetzung ist, dass die DBH-Kennung in der DB-Kennung als Miteigentümer der Datenbankdateien erklärt ist.

Um Datenbanken in einer DB-Kennung zu reorganisieren, ohne dass die DBH-Kennung Miteigentümer der Datenbank-Dateien ist, muss COPY angegeben werden.

SESAM/SQL führt die Reorganisation eines Anwender-Space in zwei Phasen durch:

- in der ersten Phase wird die Arbeitsdatei mit den reorganisierten Blöcken des Space aufgebaut. In dieser Phase kann der Anwender mit DML-Anweisungen lesend auf den Originalspace zugreifen.
- in einer zweiten Phase wird die Arbeitsdatei in den Space umbenannt (RENAME) oder kopiert (COPY). In dieser Phase kann auf den betreffenden Space nicht zugegriffen werden. Der Anwender kann die Sperrzeit kurz halten, wenn er verhindert, dass der Space kopiert wird. Das geschieht durch die explizite Angabe von RENAME oder indem keine Arbeitsdatei angegeben wird.

Während der Reorganisation eines Catalog-Space ist kein Zugriff auf den Catalog-Space möglich. Während der Reorganisation eines Anwender-Space kann jedoch während der ersten Phase der Reorganisation auf den Anwender-Space zugegriffen werden.

REORG SPACE ist auch für Replikate zugelassen.

- Es können nur Anwender-Spaces eines Replikates reorganisiert werden, nicht der Catalog-Space.
- Während der gesamten Reorganisation ist der Space exklusiv gesperrt.
- Da die Storage Group Definitionen in den Metadaten für das Original gelten und im Allgemeinen nicht für das Replikat, werden bei REORG [CATALOG\_]SPACE von Replikaten mit RENAME für die Anlage der Standard-Arbeitsdatei nicht die Metadaten ausgewertet, sondern die Standard-Arbeitsdatei wird auf dem Medium angelegt, auf dem sich der Replikat-Space befindet.

Bei der Reorganisation eines Anwender-Space können Sie eine Banddatei als Arbeitsdatei angeben, wenn der Space nur Tabellen und keine Indizes enthält. In diesem Fall wird die Reorganisation in drei Phasen durchgeführt:

- In der ersten Phase werden alle Tabellen des Space in die Arbeitsdatei exportiert. Auf den Originalspace kann in dieser Phase mit DML-Anweisungen lesend zugegriffen werden.
- In der zweiten Phase wird der Space logisch gelöscht. Auf den Space kann in dieser Phase nicht zugegriffen werden.
- In der dritten Phase werden alle Tabellen aus der Arbeitsdatei in den Space importiert. Auf den Space kann in dieser Phase nicht zugegriffen werden.

Der aktuelle Berechtigungsschlüssel muss das Sonder-Privileg UTILITY besitzen oder Eigentümer des zu reorganisierenden Anwender-Space sein.

---

```

REORG { SPACE space [ { OLD ROW_IDS } ]
      { CATALOG_SPACE catalog } } [USING FILE datei [PASSWORD kennwort]]

[ { COPY
  { RENAME } ] ] [MINIMIZE]

```

---

**CATALOG\_SPACE** *catalog*

Name der Datenbank, deren Catalog-Space reorganisiert werden soll.

**SPACE** *space*

Name des Anwender-Space, der reorganisiert werden soll.

**NEW ROW\_IDS**

Die Satznummern von Tabellen mit Primärschlüssel werden reorganisiert. Damit sind die Indizes der Tabellen des Space ungültig geworden. Die Indizes müssen mit RECOVER INDEX repariert werden. Wenn für den Space die logische Datensicherung eingeschaltet ist, dann ist der Space anschließend im Zustand „copy pending“.

Bei Tabellen ohne Primärschlüssel wird NEW ROW\_IDS ignoriert. Die Satznummern bleiben in jedem Fall erhalten.

**OLD ROW\_IDS**

Die Satznummern der Tabellen bleiben erhalten.

USING FILE *datei* [PASSWORD *kennwort*]

Name der Arbeitsdatei. *datei* muss als alphanumerisches Literal angegeben werden.

Ist bei USING FILE eine noch nicht existierende Arbeitsdatei angegeben, so legt SESAM/SQL die Datei unter dem angegebenen Namen auf den Medien an, die für die Standard-Arbeitsdatei gelten. Ist der Dateiname mit Catid angegeben, so wird die Datei auf dem durch die Catid bezeichneten Pubset angelegt.

Wenn die Datenbank in einer DB-Kennung liegt, muss bei RENAME die Arbeitsdatei ebenfalls auf der DB-Kennung angelegt sein.

Für die Festlegung der Größe von Primär- und Sekundärzuweisung der Arbeitsdatei sollten Sie sich an den entsprechenden Werten für den zu reorganisierenden Space (Catalog-Space bzw. Anwender-Space) orientieren. Wenn SESAM/SQL bei der Reorganisation gleichzeitig eine größere Freiplatzreservierung für den Space berücksichtigt, empfiehlt es sich, entsprechend größere Werte für Primär- und Sekundärzuweisung zu wählen. In jedem Fall sollten die Werte für Primär- und Sekundärzuweisung der Arbeitsdatei durch 12 teilbar sein.

Als Arbeitsdatei kann eine Banddatei verwendet werden, wenn auf dem angegebenen Space nur Tabellen liegen. Indizes können über eine Banddatei nicht reorganisiert werden. Partitionierte CALL-DML-Tabellen können über eine Banddatei nicht reorganisiert werden, wenn NEW ROW\_IDS angegeben ist. Wenn der Space in der logischen Datensicherung ist, dann ist er nach Ausführung von REORG im Zustand „copy pending“.

Sie können einen Space mit Tabellen und Indizes über das Anweisungspaar EXPORT/IMPORT TABLE so umstrukturieren, dass die Indizes anschließend auf einem separaten Space liegen.

Der Catalog-Space kann nicht über eine Banddatei reorganisiert werden, weil er Indizes enthält. Liegt die Arbeitsdatei auf Band, und ist RENAME angegeben, so wird die Anweisung abgewiesen.

Den Namen der Arbeitsdatei geben Sie als alphanumerisches Literal an:

```
'[:catid:]bs2000_benutzerkennung.einf_dateiname'
```

*catid*

BS2000-Katalogkennung.

*catid* muss als alphanumerisches Literal angegeben werden.

Wurde eine CATID-Liste definiert, muss die Katalogkennung in der Liste angegeben worden sein.

*catid* nicht angegeben:

Es wird die Default-Katalogkennung verwendet, die der BS2000-Benutzerkennung zugewiesen ist, in der der DBH abläuft.

*bs2000\_benutzerkennung*

BS2000-Benutzerkennung, in der die Arbeitsdatei angelegt wird.

*bs2000\_benutzerkennung* nicht angegeben:

Die Arbeitsdatei wird in der BS2000-Benutzerkennung angelegt, in der der DBH abläuft.

*einf\_dateiname*

Nicht-qualifizierter Dateiname gemäß BS2000-Konventionen

PASSWORD *kennwort*

BS2000-Kennwort für die Arbeitsdatei. *kennwort* muss als alphanumerisches Literal angegeben werden.

Für die Angabe von *kennwort* gibt es folgende Möglichkeiten:

- 'C'*zeichenkette*"  
*zeichenkette* enthält vier abdruckbare Zeichen.
- 'X'*hex-zeichenkette*"  
*hex-zeichenkette* enthält acht sedezimale Zeichen.
- 'n'  
*n* bezeichnet eine ganze Zahl von - 2147483648 bis + 2147483647.

USING FILE *datei* nicht angegeben:

SESAM/SQL verwendet eine Standard-Arbeitsdatei mit folgendem Namen:

- Name der Arbeitsdatei für den Catalog-Space:  
*:catid:dbh-kennung.catalog.CATALOG.REORG*
- Name der Arbeitsdatei für Anwender-Spaces:  
*:catid:dbh-kennung.catalog.space.REORG*

Bei COPY wird die Standard-Arbeitsdatei auf dem Default-Pubset der DBH-Kennung angelegt.

Bei RENAME wird die Standard-Arbeitsdatei auf dem in der Storage Group des Space angegebenen Pubset oder private Volume angelegt. Dadurch wird ein Space auf eine andere Storage Group verlagert, wenn diese zuvor mit ALTER SPACE für diesen Space neu zugewiesen worden ist.

Hat der Anwender bereits eine Datei, die den Namenskonventionen für die Standard-Arbeitsdatei genügt, auf einem dem DHB zugänglichen Pubset angelegt, wird diese Datei als Standard-Arbeitsdatei verwendet.

Wenn die Datenbank in einer DB-Kennung liegt, so wird die Arbeitsdatei in der DB-Kennung angelegt, sofern die Vorbereitungen dafür getroffen worden sind, siehe Abschnitt „Datenbankdateien und Jobvariablen auf fremden Benutzerkennungen“ im „[Basishandbuch](#)“. Kann die Arbeitsdatei nicht in der DB-Kennung angelegt werden, so wird die Anweisung abgewiesen.

## COPY

Bei COPY wird die reorganisierte Arbeitsdatei in die Space-Datei kopiert, d.h. der Space behält seine ursprüngliche Lage (außer bei Arbeitsdateien auf Band). Eine von SESAM/SQL angelegte Standard-Arbeitsdatei oder eine bei USING FILE angegebene, aber nicht vom Anwender angelegte Arbeitsdatei wird wieder gelöscht. Fehlt die Angabe USING FILE und ist COPY angegeben, dann wird die Standard-Arbeitsdatei in der DBH-Kennung angelegt und kopiert.

COPY ist der Standard-Wert bei Angabe von USING FILE.

## RENAME

Bei RENAME wird die existierende Space-Datei gelöscht und die reorganisierte Arbeitsdatei in die Space-Datei umbenannt, d.h. der Space liegt jetzt dort, wo die Arbeitsdatei lag. RENAME ist der Standard-Wert, wenn USING FILE nicht angegeben wird.

Wenn die Datenbank in einer DB-Kennung liegt, muss in der DB-Kennung die DBH-Kennung als Miteigentümer der Dateien erklärt sein. Wenn die Arbeitsdatei in diesem Fall vom Anwender nur mehrbenutzbar in der DB-Kennung angelegt worden ist, der DBH aber nicht Miteigentümer ist, scheitert die REORG-Anweisung erst beim Umbenennen.

## MINIMIZE

Bei der Angabe von MINIMIZE wird der Speicherplatz der Space-Datei, der nach der Reorganisation nicht mehr benötigt wird, freigegeben.

## Beispiel

Das Beispiel reorganisiert den Space TABLESPACE. Die Datei DAT.REORG.TABLE wird als Arbeitsdatei verwendet.



```
REORG SPACE tablespaces USING FILE 'DAT.REORG.TABLE'
```

## REORG ONLINE TABLE - Basistabelle reorganisieren

SESAM/SQL führt die Reorganisation einer Basistabelle durch Modifizieren und Kopieren der Blöcke innerhalb des Anwender-Space im laufenden Betrieb („online“) durch. Da hierfür keine exklusiven Transaktionssperren angefordert werden, können andere DML-Anwendungen lesend und ändernd auf die Basistabelle zugreifen.

Indizes einer Basistabelle sind von der Reorganisation nicht betroffen.

REORG ONLINE TABLE ist nicht für die Tabellen von Replikaten zugelassen.

Der aktuelle Berechtigungsschlüssel muss das Sonder-Privileg UTILITY besitzen oder Eigentümer des Schemas sein, in dem die zu reorganisierende Tabelle liegt.

---

```
REORG ONLINE TABLE tabelle [ON SPACE space]
```

---

TABLE *tabelle*

Name der Basistabelle, die reorganisiert werden soll.

Die Tabelle muss einen Primärschlüssel besitzen.

Bei partitionierten Tabellen kann auch eine einzelne Partition reorganisiert werden. Die Partition wird durch die Klausel ON SPACE *space* festgelegt.

ON SPACE *space*

Name des Space, auf dem die Basistabelle bzw. die Partition liegt.

Wenn diese Klausel bei einer partitionierten Tabelle nicht angegeben wird, dann werden nacheinander alle Partitionen der Basistabelle reorganisiert.

### Beispiel

Das Beispiel reorganisiert die Partition der Tabelle LEISTUNG, die auf dem Anwender-Space TABLESPACE liegt.



```
REORG ONLINE TABLE leistung ON SPACE tablespace
```

## UNLOAD - Anwenderdaten aus einer Tabelle ausgeben

Mit UNLOAD können Sie Anwenderdaten einer Tabelle einer SESAM/SQL-Datenbank in eine BS2000-Datei vom Typ SAM ausgeben.

Sie können alle Daten einer Tabelle (UNLOAD TABLE) oder auch Daten aus ausgewählten Spalten der Tabelle ausgeben (UNLOAD DATA).

Mit UNLOAD ONLINE können Sie Anwenderdaten von Basistabellen oder Views ausgeben, die auszugebenden Daten durch eine Suchbedingung einschränken und eine Sortierreihenfolge für die Ausgabedatei definieren.

Bei UNLOAD OFFLINE können Sie Anwenderdaten von Basistabellen ausgeben. Außerdem können Sie Daten im TRANSFER\_FORMAT ausgeben:

- aus einem Anwender-Space, für den die Reparatur nicht vollständig durchgeführt werden konnte (Klausel FROM SPACE, Zustand des Anwender-Space: „recover pending“)
- aus einem SESAM-Sicherungsbestand (Klausel FROM COPY\_FILE)

Die Anzahl entladener Sätze wird im Kommunikationsbereich des ESQL-COBOL-Programms im Feld `SQLrowcount` zurückgegeben.

SESAM bietet mehrere Ausgabeformate an, nämlich TRANSFER\_FORMAT, LOAD\_FORMAT, DELIMITER\_FORMAT, CSV\_FORMAT und die Standarddarstellung; außerdem ist es möglich, die Darstellung der einzelnen Spalten selbst zu definieren (siehe auch „UNLOAD-Formate“ auf Seite 253). Ist für bestimmte Spalten in letzterem Fall keine Darstellung angegeben, so werden die Werte in der Standarddarstellung ausgegeben, die dem Datentyp der jeweils zugeordneten Spalte in der Tabelle entspricht. Falls weder ein Ausgabeformat angegeben noch die Darstellung bestimmter Spalten beschrieben ist, dann werden alle Werte in der Standarddarstellung des Datentyps der jeweils zugeordneten Spalte ausgegeben.

Die Ausgabedatei wird mit Coded Character Set Name (CCSN) angelegt. Im DELIMITER\_FORMAT, im CSV\_FORMAT und im selbst definierten Format werden die auszugebenden Daten nötigenfalls in den codierten Zeichensatz der Ausgabedatei konvertiert.

Zusammen mit der Utility-Anweisung LOAD bietet UNLOAD zusätzlich die Möglichkeit, Anwenderdaten von einer Tabelle in eine andere zu übertragen. Die Ausgabedatei von UNLOAD dient hierbei als Eingabedatei für ein nachfolgendes LOAD. Zu diesen Zweck können die vier namentlich genannten Formate genutzt werden (die Angabe LOAD\_FORMAT bei UNLOAD entspricht der Angabe UNLOAD\_FORMAT bei LOAD). Vorsicht ist aber bei der Standarddarstellung und bei selbst definierten Ausgabeformaten geboten (insbesondere wenn ein Ausgabeformat nicht explizit festgelegt wurde), da dann die Behandlung von NULL-Werten genau zu beachten ist (siehe Klausel WHEN NULL THEN).

Sollen Anwenderdaten zusammen mit der Tabellenstruktur einer Basistabelle kopiert werden, bietet sich die Verwendung des Anweisungspaares EXPORT/IMPORT TABLE an. Mit diesem Anweisungspaar ist es auch möglich, nur die Metadaten einer Basistabelle zu kopieren. Nähere Informationen zu den Anwendungsmöglichkeiten von EXPORT/IMPORT TABLE finden Sie im [Abschnitt „EXPORT TABLE - Tabelle aus einer Datenbank in eine Exportdatei exportieren“](#) auf Seite 129 und im [Abschnitt „IMPORT TABLE - Tabelle aus einer Export-Datei in eine Datenbank importieren“](#) auf Seite 133.

Der aktuelle Berechtigungsschlüssel muss das Sonder-Privileg UTILITY besitzen oder Eigentümer des Schemas sein, in dem die Tabelle liegt, aus der Daten ausgegeben werden sollen.

Wenn eine Suchbedingung oder eine Sortierreihenfolge angegeben wird, dann muss der aktuelle Berechtigungsschlüssel das Tabellen-Privileg SELECT für die entsprechende Tabelle und alle weiteren in der Suchbedingung angesprochenen Tabellen haben. Er muss auch das EXECUTE-Privileg für alle in der Suchbedingung oder in der Sortierreihenfolge angesprochenen User Defined Functions (UDF, siehe Handbuch „[SQL-Sprachbeschreibung Teil 1: SQL-Anweisungen](#)“) haben.

Die Anweisung UNLOAD ... FROM COPY\_FILE darf zusätzlich nur auf bestimmten System-Benutzerkennungen eingegeben werden. Diese Benutzerkennungen legt der SESAM/SQL-Systemverwalter mit der DBH-Option ADMINISTRATOR bzw. mit der Administrationsanweisung MODIFY-ADMINISTRATION fest.

---

```
UNLOAD [ { OFFLINE } ] { TABLE }
      [ { ONLINE } ] { DATA } tabelle [(ausgabespalte ...)]
```

```
[FROM { SPACE space
      { COPY_FILE datei [PASSWORD kennwort] } } ]
```

```
INTO FILE datei [PASSWORD kennwort]
```

```
{ [(ausgabadarstellung ...)]
  TRANSFER_FORMAT
  LOAD_FORMAT
  DELIMITER_FORMAT TERMINATED BY delimiter [ { EBCDIC }
                                               { UTFE } ] }
{ CSV_FORMAT DELIMITER delimiter [QUOTE quote] [ESCAPE escape]
  [ { EBCDIC } ] [WITH HEADER] }
```

```
[WHERE suchbedingung]
```

```
[ORDER BY { { spalte
              { spalte(posnr)
              { spaltennummer
              { ausdruck
              } } } } [ { ASC } ] } , ... ]
```

```
ausgabespalte ::= { spalte
                   { spalte(posnr)
                   { spalte(min..max)
                   }
```

```
ausgabadarstellung ::= POSITION ( { zahl
                                   { * }
                                   ) [ datentyp ]
                       [WHEN NULL THEN nulldeskriptor]
```

```
nulldeskriptor ::= { literal
                     { POSITION(zahl) = alphanumerisches_litera
                     }
```

---

### OFFLINE

Bei UNLOAD OFFLINE sperrt SESAM/SQL den Anwender-Space gegen ändernde Zugriffe anderer Benutzer. Die Verarbeitung findet überwiegend in einer Service-Task statt.

Bei UNLOAD OFFLINE werden alle Sätze einer Basistabelle entladen.

In diesem Fall ist UNLOAD OFFLINE wesentlich schneller als UNLOAD ONLINE .

Bei UNLOAD OFFLINE können mit der Klausel FROM SPACE bzw. FROM COPY\_FILE auch Daten aus Basistabellen ausgegeben werden, wenn auf deren Anwender-Spaces im normalen SESAM/SQL-Betrieb nicht zugegriffen werden kann.

### ONLINE

Bei UNLOAD ONLINE wird der Anwender-Space nicht exklusiv gesperrt. Es werden lediglich die gleichen Sperren angefordert wie im Rahmen einer DML-Suchanweisung. Die Verarbeitung findet vollständig in einer DBH-Task statt.

Bei UNLOAD ONLINE können die auszugebenden Daten durch eine Suchbedingung eingeschränkt werden. Es kann auch eine Sortierreihenfolge für die Ausgabedatei festgelegt werden. Sowohl Inhalte von Basistabellen als auch von Views können ausgegeben werden.

Bei UNLOAD ONLINE dürfen Ausgabedatei und Fehlerdatei nicht auf Band liegen.

### TABLE

Die Daten aller Spalten werden in die Ausgabedatei geschrieben.

Über die Liste der Ausgabespalten (siehe *ausgabespalte* , ...) können Sie beliebigen Spalten eine Ausgabedarstellung (siehe *ausgabedarstellung* , ...) zuordnen. Auf diese Weise können Sie für die betreffenden Spalten eine von der Spaltendefinition in der Datenbank abweichende Ausgabedarstellung festlegen und außerdem die Darstellung der NULL-Werte in der Ausgabedatei definieren.

Die Ausgabereihenfolge der Spalten entspricht der Reihenfolge der Spalten gemäß der Definition der Basistabelle, aus der Daten ausgegeben werden sollen. Insbesondere gilt das auch für die Spalten, die in der Liste der Ausgabespalten nicht angegeben sind.

### DATA

Nur die Daten der Spalten, die in der Liste der Ausgabespalten (siehe *ausgabespalte*, ...) aufgeführt sind, werden in die Ausgabedatei geschrieben. Die Ausgabereihenfolge entspricht der Reihenfolge der Spalten in dieser Liste. UNLOAD DATA ist nur zulässig, wenn Sie eine Liste der Ausgabespalten angeben.

### *tabelle*

Name der Basistabelle oder des Views, aus dem Daten ausgegeben werden sollen.

Bei UNLOAD ONLINE darf die Tabelle *tabelle* keine Nur-CALL-DML-Tabelle sein.

Bei einer partitionierten Tabelle müssen alle betroffenen Partitionen zur Verfügung stehen, siehe [Tabelle 3 auf Seite 22](#).

Wenn *tabelle* einen View bezeichnet, dann muss der Parameter ONLINE angegeben werden; der Operand TRANSFER\_FORMAT darf in diesem Fall nicht angegeben werden.

*(ausgabespalte ,...)*

Liste der Spalten der Tabelle *tabelle*, aus denen Daten ausgegeben werden sollen. SESAM/SQL gibt die Daten in der angegebenen Reihenfolge der Ausgabespalten aus.

Es gelten folgende Regeln:

- Die Liste der Ausgabespalten darf nicht mehrfach dieselbe Ausgabespalte enthalten.
- *(ausgabespalte ,...)* muss angegeben werden, wenn UNLOAD DATA angegeben ist.
- Falls die Liste der Ausgabedarstellungen (*ausgabedarstellung ,...*) angegeben ist, muss die Anzahl der angegebenen Ausgabespalten übereinstimmen mit der Anzahl der angegebenen Ausgabedarstellungen.
- Falls UNLOAD TABLE angegeben ist, dürfen Sie die Liste der Ausgabespalten nur angeben, wenn gleichzeitig die Liste der Ausgabedarstellungen angegeben ist.
- Bei multiplen Spalten gibt SESAM/SQL stets die maximale Anzahl an Ausprägungen aus, wenn die Liste der Ausgabespalten nicht angegeben ist oder die multiple Spalte in der Liste der Ausgabespalten nur über den Spaltennamen angesprochen wird.  
Wenn nicht alle Ausprägungen der multiplen Spalte mit Werten versorgt sind, so behandelt SESAM/SQL die fehlenden Werte wie NULL-Werte.

*ausgabespalte ::=*

*spalte*

Name der Spalte, aus der Daten ausgegeben werden sollen.

Wenn *spalte* eine multiple Spalte ist, gilt im Folgenden:

$sp_{anz}$  bezeichnet die Anzahl der vorhandenen Ausprägungen für die Spalte *spalte*.  
 $sp_{max}$  bezeichnet die maximal mögliche Anzahl an Ausprägungen für die Spalte *spalte*.

*spalte (posnr)*

*posnr* ist eine positive ganze Zahl.

Wenn *posnr* kleiner oder gleich  $sp_{anz}$  ist, dann wird das *posnr*-te Spaltenelement der multiplen Spalte *spalte* in die Ausgabedatei geschrieben, andernfalls wird *spalte(posnr)* wie der NULL-Wert behandelt.

Ist *spalte* keine multiple Spalte oder *posnr* größer als  $sp_{max}$ , erfolgt eine Fehlermeldung.

*spalte (min .. max)*

Der Wert ist das Aggregat aus den Spaltenelementen *min* bis *max* der multiplen Spalte *spalte*.

Ist *spalte* keine multiple Spalte oder *min* nicht kleiner als *max*, dann erfolgt eine Fehlermeldung.

*posnr* bzw. *min .. max* nicht angegeben:

Ist *spalte* eine multiple Spalte, so ist die Angabe *spalte* gleichbedeutend mit der Angabe *spalte*(1 ..  $sp_{max}$ ).

*spalte*, *spalte* (*posnr*) bzw. *spalte* (*min .. max*)

darf nicht mehrfach mit sich überlappenden Ausprägungen als *ladespalte* angegeben werden.

FROM SPACE *space*

Name eines Anwender-Space, der sich im Zustand „recover pending“ befindet. Der Anwender-Space darf nicht defekt sein.

Diese Klausel dürfen Sie nur in Verbindung mit den Klauseln UNLOAD OFFLINE TABLE und TRANSFER\_FORMAT angeben.

FROM COPY\_FILE *datei* [PASSWORD *kennwort*]

BS2000-Dateiname eines auf Platte angelegten SESAM-Sicherungsbestandes eines Anwender-Space. Der SESAM-Sicherungsbestand darf nicht defekt sein. *datei* muss als alphanumerisches Literal angegeben werden.

Diese Klausel dürfen Sie nur in Verbindung mit den Klauseln UNLOAD OFFLINE TABLE und TRANSFER\_FORMAT angeben.

PASSWORD *kennwort*

BS2000-Kennwort, das bei CREATE CATALOG vergeben wurde. *kennwort* muss als alphanumerisches Literal angegeben werden.

Für die Angabe von *kennwort* gibt es folgende Möglichkeiten:

- 'C'*zeichenkette*"  
*zeichenkette* enthält vier abdruckbare Zeichen.
- 'X'*hex-zeichenkette*"  
*hex-zeichenkette* enthält acht sedezimale Zeichen.
- 'n'  
*n* bezeichnet eine ganze Zahl von - 2147483648 bis + 2147483647.

**INTO FILE** *datei* [**PASSWORD** *kennwort*]

Name der BS2000-Datei, in die die Daten geschrieben werden. *datei* muss als alphanumerisches Literal angegeben werden. Bei UNLOAD ONLINE darf *datei* keine Banddatei sein.

Die Ausgabedatei ist eine SAM-Datei, die bedarfsorientiert an der maximalen Satzlänge der auszugebenden Sätze formatiert ist. Die Berechnung der maximalen Satzlänge orientiert sich an der Definitionslänge und der Ausgabelänge der Spalten. Außerdem wird die Ausgabedatei schon in der zu erwartenden Größe angelegt, d.h. die Anzahl Sätze der zu entladenden Tabelle wird mitberücksichtigt.

SAM-Dateien können in Einheiten von 4KB Blöcken bis zu einer maximalen Blockgröße von 32 KB angelegt werden. Die Ausgabedatei wird in derjenigen Blockgröße angelegt, bei der der Verschnitt am geringsten ist. Dabei wird versucht, die Blockgröße der Ausgabedatei so groß wie möglich zu wählen. Der größtmögliche Verschnitt pro Satz beträgt also 4095 Bytes (= 4KB - 1 Byte).

Für alle UNLOAD-Formate mit Ausnahme von DELIMITER\_FORMAT und CSV\_FORMAT wird die Ausgabedatei mit dem CCSN der Datenbank (siehe Klausel CODE-TABLE auf [Seite 108](#)) angelegt. Wenn für die Datenbank kein CCSN verwendet wird, dann wird auch die Ausgabedatei ohne CCSN angelegt.

Im DELIMITER\_FORMAT und im CSV\_FORMAT wird der CCSN der Ausgabedatei über den Parameter EBCDIC/UTFE bestimmt (siehe [Seite 245](#)).

Kann die Ausgabedatei nicht angelegt werden, so wird die UNLOAD-Anweisung abgebrochen. Wird die UNLOAD-Anweisung abgebrochen, weil die theoretisch mögliche Länge eines Ausgabesatzes den maximalen Wert überschreitet, so empfiehlt es sich, mit der Hauptfunktion EXPORT TABLE zu arbeiten, die Daten mit UNLOAD DATA auszugeben oder das CSV\_FORMAT mit ESCAPE-Zeichen zu verwenden.

SESAM/SQL legt die Ausgabedatei unter dem angegebenen Dateinamen in der BS2000-Kennung an, in der der DBH abläuft. Soll die Ausgabedatei in einer anderen Kennung liegen, müssen Vorbereitungen dafür getroffen worden sein, siehe Abschnitt „Datenbankdateien und Jobvariablen auf fremden Benutzerkennungen“ im „[Basishandbuch](#)“.

**PASSWORD** *kennwort*

BS2000-Kennwort für eine kennwortgeschützte Ausgabedatei. *kennwort* muss als alphanumerisches Literal angegeben werden.

Für die Angabe von *kennwort* gibt es folgende Möglichkeiten:

- 'C'*zeichenkette*''  
*zeichenkette* enthält vier abdruckbare Zeichen.
- 'X'*hex-zeichenkette*''  
*hex-zeichenkette* enthält acht sedezimale Zeichen.
- 'n'  
*n* bezeichnet eine ganze Zahl von - 2147483648 bis + 2147483647.

(*ausgabedarstellung* , ...)

Liste der Ausgabedarstellungen.



Falls weder (*ausgabedarstellung*, ... ) noch TRANSFER\_FORMAT, LOAD\_FORMAT oder DELIMITER\_FORMAT angegeben ist, so werden alle Werte in der Standarddarstellung ausgegeben, die dem Datentyp der jeweils zugeordneten Spalte der Basistabelle *tabelle* entspricht.

Die Ausgabedarstellungen beschreiben Formate und Positionen der Ausgabewerte sowie die Darstellung des NULL-Werts in der Ausgabedatei. Falls eine Liste der Ausgabespalten (*ausgabespalte* , ...) angegeben ist, ordnet SESAM/SQL die angegebenen Ausgabedarstellungen der Reihe nach den Elementen der Liste der Ausgabespalten zu.

Es gelten folgende Regeln:

- Wenn bei UNLOAD TABLE nicht alle Spalten der Basistabelle *tabelle* in der Liste der Ausgabedarstellungen enthalten sind, so werden die Werte der fehlenden Spalten in der Standarddarstellung ausgegeben, die dem Datentyp der jeweiligen Spalte entspricht.
- Wenn bei UNLOAD TABLE nur die Liste der Ausgabedarstellungen angegeben ist, nicht jedoch die Liste der Ausgabespalten (*ausgabespalte* , ...), muss die Anzahl der Ausgabedarstellungen gleich der Anzahl der Spalten in der Tabelle *tabelle* sein. SESAM/SQL ordnet dann die angegebenen Ausgabedarstellungen der Reihe nach den Spalten der Tabelle *tabelle* zu.
- Werden bei der Konvertierung in die Ausgabedarstellung Zeichen oder Kommastellen abgeschnitten bzw. Kommastellen gerundet, so gibt SESAM/SQL eine Warnung aus. SESAM/SQL bricht die Ausführung von UNLOAD jedoch nicht ab, da das beschriebene Verhalten beabsichtigt sein kann.  
Tritt Datenverlust bei der Konvertierung auf, dann bricht SESAM/SQL die Ausführung von UNLOAD ab.
- Wenn Sie in der Liste der Ausgabespalten ein multiples Attribut nur über den Spaltennamen ansprechen oder wenn Sie einen Ausprägungsbereich angeben, so unterstellt SESAM/SQL, dass alle Ausprägungen dieser multiplen Spalte dasselbe Format haben. Daher ist in den genannten Fällen nur eine Formatangabe erforderlich, die sich dann auf die Ausgabedarstellung für eine Ausprägung bezieht.

Die Standarddarstellung der Daten bei UNLOAD ist im [Abschnitt „Standarddarstellung der Daten in der Ein- und Ausgabedatei“ auf Seite 33](#) beschrieben.

*ausgabedarstellung ::=*

**POSITION** (*posnr*)

Positive ganze Zahl, die die Position des Ausgabewerts in der Ausgabedatei angibt. *posnr* darf sich nicht auf eine Position außerhalb des Ausgabesatzes beziehen.

Positionsangaben, die zu sich überlappenden Ausgabefeldern führen, sind zulässig. SESAM/SQL prüft jedoch nicht, ob sich Ausgabefelder überlappen.

Die Angaben zu den Positionen im Ausgabesatz wertet SESAM/SQL entsprechend der Liste der Ausgabedarstellungen bzw. entsprechend der Liste der Ausgabespalten (*ausgabespalte*, ...) aus. Dies kann dazu führen, dass weiter rechts stehende Positionen in der Eingabedatei früher ausgewertet werden als weiter links stehende. Bei sich überlappenden Ausgabefeldern wird der Wert einer in der Liste der Ausgabespalten früher genannten Spalte durch den Wert einer später genannten Spalte überschrieben.

Bei UNLOAD TABLE sind außerdem die von SESAM/SQL intern berechneten Positionsangaben für die Spalten zu berücksichtigen, die in der Liste der Ausgabedarstellungen nicht genannt sind.

**POSITION** (\*)

POSITION(\*) muss angegeben werden, wenn eine genaue Positionsangabe nicht möglich oder nicht erwünscht ist. Die Anfangspositionen errechnen sich aus allen zuvor festgelegten Ausgabepositionen.

*datentyp*

Datentyp des Ausgabewerts.

Für *datentyp* dürfen Sie nicht FLOAT angeben. Anstatt FLOAT verwenden Sie REAL oder DOUBLE PRECISION.

Die Angabe *dimension* (Anzahl der Spaltenelemente bei multiplen Spalten) in *datentyp* ist nicht zulässig.

Fehlen die Angaben zu *länge* bzw. *stellen[,bruchteil]*, so setzt SESAM/SQL für *länge* bzw. *stellen* jeweils den Wert 1 und für *bruchteil* den Wert 0.

*datentyp* muss mit dem Datentyp der zugeordneten Ausgabespalte verträglich sein.

Wenn Sie sich die Werte für eine bestimmte Tabellenspalte in lesbarer Form ausgeben lassen wollen, müssen Sie den Datentyp CHARACTER mit ausreichender Längenangabe verwenden (siehe [Abschnitt „Lesbare Darstellung der Daten in Ein- und Ausgabedatei“ auf Seite 27](#)).

*datentyp* nicht angegeben:

SESAM/SQL verwendet den Datentyp der zugeordneten Ausgabespalte.

Wenn für eine Ausgabespalte vom Datentyp (NATIONAL) CHARACTER VARYING keine Ausgabedarstellung angegeben ist, erfolgt die Ausgabe in variabler Länge. SESAM/SQL stellt jedem Ausgabewert ein 2 Byte langes Längelfeld voran.



Das Längenfeld enthält bei NATIONAL CHARACTER VARYING die Länge in Bytes, nicht in Zeichen.

#### WHEN NULL THEN *nulldeskriptor*

Diese Klausel definiert für die zugehörige Ausgabedarstellung *ausgabedarstellung* die Darstellung des NULL-Werts in der Ausgabedatei. Die Klausel darf nur dann angegeben werden, wenn die der Ausgabedarstellung zugeordnete Ausgabespalte nicht als NOT NULL definiert ist.

Treten NULL-Werte in einer Ausgabespalte auf, für die keine WHEN NULL THEN-Klausel angegeben ist, so legt SESAM/SQL den NULL-Wert je nach Ausgabedarstellung wie folgt in der Ausgabedatei ab:

- bei (NATIONAL) CHARACTER:  
als Zeichenkette, bestehend aus Leerzeichen ' ' (X'40') bzw. N' ' (NX'0020')
- bei (NATIONAL) CHARACTER VARYING:  
das Längenfeld wird mit X'0000' belegt
- bei NUMERIC: X'F0'
- bei DECIMAL: X'0C'
- bei INTEGER, SMALLINT: binär Null
- bei REAL: X'00000000'
- bei DOUBLE PRECISION: X'0000000000000000'
- bei DATE: X'0001 0001 0001'
- bei TIME(3): X'0000 0000 0000 0000'
- bei TIMESTAMP(3): X'0001 0001 0001 0000 0000 0000 0000'

Bei Ausgabespalten, die mit (CALL-DML-)Defaultwertzeichen definiert sind, ist zu beachten:

Wird eine Ausgabedatei mit obigen Ersatzwerten (nicht-signifikanten Werten) als Eingabedatei für die Utility-Anweisung LOAD verwendet, so interpretiert SESAM/SQL diese nicht-signifikanten Werte ohne WHEN ... THEN NULL-Klausel bei LOAD nur dann als NULL-Wert, wenn das (CALL-DML-)Defaultwertzeichen das Standard-(CALL-DML-)Defaultwertzeichen ist. In solchen Fällen empfiehlt es sich, die Daten im UNLOAD\_FORMAT, TRANSFER\_FORMAT oder DELIMITER\_FORMAT zu entladen oder einen geeigneten *nulldeskriptor* in der WHEN NULL THEN ...-Klausel zu vereinbaren. Beim Laden der Daten mit LOAD müssen dann die entsprechenden Klauseln angegeben werden.

Bei einer Ausgabedarstellung, die sich auf mehrere Ausprägungen einer multiplen Spalte bezieht, dürfen Sie als *nulldeskriptor* nur ein Literal ohne Positionsangabe angeben. Aus einer Positionsangabe würde nicht ersichtlich, welche Ausprägungen den NULL-Wert enthalten.

*nulldeskriptor* ::=

*literal*

Literal.

SESAM/SQL wandelt den als *literal* angegebenen Wert datentyp-gerecht in die Ausgabedarstellung der zugeordneten Spalte um.

Bei der Ausgabedarstellung (NATIONAL) CHARACTER VARYING(*n*) darf *literal* maximal *n* Zeichen enthalten. Die Ausgabe erfolgt stets in der tatsächlichen Länge von *literal*.

POSITION(*posnr*)

Positive ganze Zahl, die die Position des NULL-Deskriptors in der Ausgabedatei angibt. *posnr* darf sich nicht auf eine Position außerhalb des Ausgabesatzes beziehen.

*alphanumerisches\_literal*

Alphanumerisches Literal

#### TRANSFER\_FORMAT

Die Daten werden in eine sogenannte Transferdatei ausgegeben.



Wenn Daten der Datentypen NATIONAL CHARACTER (VARYING) im TRANSFER\_FORMAT ausgegeben werden, dann kann die Transferdatei nicht mehr mit SESAM/SQL bis V4.0 geladen werden.

#### LOAD\_FORMAT

Die Daten werden in eine Datei ausgegeben, die als Eingabedatei für die Utility-Anweisung LOAD mit UNLOAD\_FORMAT geeignet ist.

Bei Ausgabedateien im LOAD\_FORMAT stellt SESAM/SQL jedem Ausgabewert einen Indikator vom Datentyp SMALLINT voran. Er zeigt an, ob der nachfolgende Ausgabewert als NULL-Wert zu interpretieren ist. Ein Ausgabewert ist als NULL-Wert zu interpretieren, wenn der zugehörige Indikator einen negativen Wert aufweist.

DELIMITER\_FORMAT TERMINATED BY *delimiter* {  $\left. \begin{array}{l} \text{EBCDIC} \\ \text{UTFE} \end{array} \right\}$  }

SESAM/SQL gibt die Daten im sogenannten Delimiter-Format aus. Im Delimiter-Format sind alle Werte im codierten Zeichensatz der Ausgabedatei dargestellt und mit einem DELIMITER-Zeichen abgeschlossen. Die auszugebenden Daten werden nötigenfalls gemäß dem CCSN der Datenbank in den Zeichensatz der Ausgabedatei konvertiert und ausgegeben. Genaueres zur Darstellung der Werte im Delimiter-Format siehe [Abschnitt „Lesbare Darstellung der Daten in Ein- und Ausgabedatei“ auf Seite 27](#).

Der NULL-Wert wird in der Ausgabedatei nicht dargestellt. Lediglich das den Nullwert abschließende DELIMITER-Zeichen wird dargestellt. Zwei aufeinanderfolgende DELIMITER-Zeichen sind also wie folgt zu interpretieren: das erste DELIMITER-Zeichen schließt den vorausgehenden Wert ab, das zweite DELIMITER-Zeichen schließt einen nicht dargestellten Nullwert ab.

Falls eine zu entladende Tabelle eine Spalte vom Datentyp (NATIONAL) CHARACTER VARYING enthält und einige Sätze Werte der Länge 0 haben (entspricht der leeren Zeichenkette ""), so werden diese Sätze in die UNLOAD-Fehlerdatei eingetragen, siehe Abschnitt „Fehlerdatei bei UNLOAD“ auf Seite 251.

Die (ggf. konvertierten und aufbereiteten) Werte werden in signifikanter Länge in die Ausgabedatei geschrieben. Jede Ausgabezeile wird in einem SAM-Satz geschrieben. Wird dabei die maximale Länge eines SAM-Satzes überschritten, dann wird die UNLOAD-Anweisung mit SQLSTATE abgebrochen.

#### *delimiter*

Definition des DELIMITER-Zeichens, das in der Ausgabedatei jeden Wert abschließen soll. Als DELIMITER-Zeichen sollte kein Wert gewählt werden, der Bestandteil der Ausgabewerte ist.

SESAM/SQL prüft, ob das DELIMITER-Zeichen in einem auszugebenden Wert auftritt. Dieser Satz wird nicht in die Ausgabedatei geschrieben, sondern in die UNLOAD-Fehlerdatei.

Die Angabe von *delimiter* sowie von *quote* und von *escape* (im CSV\_FORMAT, siehe Seite 247) hängt vom folgenden Parameter ab.

#### EBCDIC

Als CCSN der Ausgabedatei wird der CCSN der Datenbank (siehe Klausel CODETABLE auf Seite 108) eingetragen. Daten der Datentypen NATIONAL CHARACTER (VARYING) werden in den (EBCDIC-)Zeichensatz der Ausgabedatei konvertiert und ausgegeben. Daten der Datentypen CHARACTER (VARYING) werden unverändert ausgegeben. Daten anderer Datentypen werden in den (EBCDIC-)Zeichensatz der Ausgabedatei konvertiert und ausgegeben. (Der spezielle EBCDIC-Zeichensatz spielt hier keine Rolle, da bei der Konvertierung dieser (numerischen oder Zeit-) Datentypen die benötigten Zeichen alle im EBCDIC-Kern liegen und jeweils identisch dargestellt werden.)

Wenn für die Datenbank kein CCSN definiert ist, dann wird die Ausgabedatei ohne CCSN angelegt. Daten der Datentypen NATIONAL CHARACTER (VARYING) können in diesem Fall nicht ausgegeben werden. Daten der Datentypen CHARACTER (VARYING) werden unverändert ausgegeben. Daten anderer Datentypen werden nach EBCDIC konvertiert und ausgegeben.

Wenn eine Konvertierung nicht möglich ist, dann wird ein Eintrag in die UNLOAD-Fehlerdatei geschrieben.

*delimiter* sowie *quote* und *escape* (CSV\_FORMAT) sind in diesem Fall jeweils ein alphanumerisches Literal der Länge 1.

Beispiele: TERMINATED BY ';' oder TERMINATED BY X'5E'

Soll ein einfaches Hochkomma (') definiert werden, so müssen Sie vier einfache Hochkommata angeben (TERMINATED BY ''').

Aus Kompatibilitätsgründen kann auch die sedezimale Darstellung angegeben werden: 'X"xx"' (x: eine Stelle der sedezimalen Darstellung).



Ausgabedateien im Delimiter-Format mit CCSN EBCDIC sind CSV-Dateien (CSV: Comma Separated Values). Sie können in SESAM/SQL (siehe auch die Tabellenfunktion CSV im Handbuch „[SQL-Sprachbeschreibung Teil 1: SQL-Anweisungen](#)“ sowie die Utility-Funktionen LOAD und UNLOAD) und auch in externen Softwareprodukten verwendet werden.

## UTFE

Die Ausgabedatei wird mit dem CCSN UTFE angelegt. Daten der Datentypen CHARACTER (VARYING) werden gemäß dem CCSN der Datenbank (siehe Klausel CODE-TABLE auf [Seite 108](#)) in den Zeichensatz UTFE der Ausgabedatei konvertiert und ausgegeben. Daten anderer Datentypen werden in den Zeichensatz UTFE der Ausgabedatei konvertiert und ausgegeben.



Wenn Daten des Datentypen NATIONAL CHARACTER VARYING im DELIMITER\_FORMAT oder im CSV\_FORMAT in eine Ausgabedatei mit dem CCSN UTFE ausgegeben werden, dann kann der Ausgabewert gekürzt sein, wenn er nach der Konvertierung größer ist als die maximale Satzlänge der SAM-Datei.

Wenn für die Datenbank kein CCSN definiert ist, dann können Daten der Datentypen CHARACTER (VARYING) nicht ausgegeben werden. Daten anderer Datentypen werden in den Zeichensatz UTFE der Ausgabedatei konvertiert und ausgegeben.

Wenn eine Konvertierung nicht möglich ist, dann wird ein Eintrag in die UNLOAD-Fehlerdatei geschrieben.

*delimiter* sowie *quote* und *escape* (CSV\_FORMAT) sind in diesem Fall jeweils ein National-Literal der Länge 1, das auch in UTFE eine sedezimale Darstellung in einem Byte hat (Wertebereich NX'0000' ... NX'009F').

Beispiele: TERMINATED BY N';' oder TERMINATED BY NX'003B'  
oder auch: TERMINATED BY U&';' oder TERMINATED BY U&'003B'

Soll ein einfaches Hochkomma (') definiert werden, so müssen Sie vier einfache Hochkommata angeben (TERMINATED BY N'''' oder ...U&'').

CSV\_FORMAT DELIMITER *delimiter* [QUOTE *quote*] [ESCAPE *escape*]

{  $\left. \begin{array}{l} \text{EBCDIC} \\ \text{UTFE} \end{array} \right\}$  } [WITH HEADER]

SESAM/SQL gibt die Daten im sogenannten CSV-Format aus. Im CSV-Format sind alle Werte im codierten Zeichensatz der Ausgabedatei dargestellt und mit einem DELIMITER-Zeichen abgeschlossen. Die auszugebenden Daten werden nötigenfalls gemäß dem CCSN der Datenbank in den Zeichensatz der Ausgabedatei konvertiert und ausgegeben. Genauerer zur Darstellung der Werte im CSV-Format siehe [Abschnitt „Lesbare Darstellung der Daten in Ein- und Ausgabedatei“ auf Seite 27](#).

Der NULL-Wert wird in der Ausgabedatei nicht dargestellt, lediglich das den Nullwert abschließende DELIMITER-Zeichen. Zwei aufeinanderfolgende DELIMITER-Zeichen sind also wie folgt zu interpretieren: das erste DELIMITER-Zeichen schließt den vorausgehenden Wert ab, das zweite DELIMITER-Zeichen schließt einen nicht dargestellten Nullwert ab.

Wenn eine zu entladende Tabelle eine Spalte vom Datentyp (NATIONAL) CHARACTER VARYING enthält und einige Sätze Werte der Länge 0 haben (entspricht der leeren Zeichenkette ""), so werden diese Sätze in der Ausgabedatei durch zwei aufeinander folgende QUOTE-Zeichen dargestellt, wenn ein QUOTE-Zeichen angegeben ist. Wenn kein QUOTE-Zeichen angegeben ist, dann werden diese Sätze in die UNLOAD-Fehlerdatei eingetragen.

Die (ggf. konvertierten und aufbereiteten) Werte werden in signifikanter Länge in die Ausgabedatei geschrieben. Jede Ausgabzeile wird in einem SAM-Satz geschrieben. Wird dabei die maximale Länge eines SAM-Satzes überschritten, dann wird als letztes Zeichen des SAM-Satzes ein ESCAPE-Zeichen geschrieben und der Ausgabesatz im nächsten SAM-Satz fortgesetzt. Wenn kein ESCAPE-Zeichen angegeben ist, dann wird die UNLOAD-Anweisung mit SQLSTATE abgebrochen.

Wenn ein zu entladender Spaltenwert DELIMITER-Zeichen enthält, dann wird dieser Wert in der Ausgabedatei in QUOTE-Zeichen eingeschlossen, wenn ein QUOTE-Zeichen angegeben ist. Wenn ein ESCAPE-Zeichen angegeben ist, dann wird dem DELIMITER-Zeichen das ESCAPE-Zeichen vorangestellt. Wenn weder ein QUOTE- noch ein ESCAPE-Zeichen angegeben sind, dann wird der Satz in die UNLOAD-Fehlerdatei eingetragen.

Wenn ein zu entladender Spaltenwert QUOTE-Zeichen enthält, dann wird dem QUOTE-Zeichen ein ESCAPE-Zeichen vorangestellt, wenn ein ESCAPE-Zeichen angegeben ist. Wenn nur ein QUOTE-Zeichen angegeben ist, dann wird ein derartiger Wert in der Ausgabedatei in QUOTE-Zeichen eingeschlossen und das QUOTE-Zeichen wird verdoppelt.

Wenn ein zu entladender Spaltenwert ESCAPE-Zeichen enthält, dann wird dem ESCAPE-Zeichen in der Ausgabedatei ein ESCAPE-Zeichen vorangestellt.

Wenn ein zu entladender Spaltenwert NEWLINE-Zeichen enthält, dann wird der Spaltenwert in der Ausgabedatei in QUOTE-Zeichen eingeschlossen, wenn ein QUOTE-Zeichen angegeben ist. Wenn kein QUOTE-Zeichen angegeben ist, dann wird der Satz in die UNLOAD-Fehlerdatei eingetragen.

#### *delimiter*

Definition des DELIMITER-Zeichens, das in der Ausgabedatei jeden Wert abschließen soll. Als DELIMITER-Zeichen kann auch ein Wert gewählt werden, der Bestandteil der Ausgabewerte ist, wenn auch ein QUOTE-Zeichen und/oder ein ESCAPE-Zeichen angegeben wird.

#### *quote*

Definition des QUOTE-Zeichens, in das die Werte der Ausgabedatei eingeschlossen werden können.

#### *escape*

Definition des Entwertungszeichens (ESCAPE-Zeichen), mit dem ESCAPE-Folgen in der Ausgabedatei beginnen können.



*delimiter*, *quote* und *escape* müssen unterschiedliche Zeichen sein.

Die Angabe von *delimiter*, *quote* oder *escape* hängt vom Parameter EBCDIC/UTFE ab, siehe [Seite 245](#).

### WITH HEADER

Dieser Parameter bewirkt, dass im ersten Satz der Ausgabedatei eine Überschriftenzeile mit den Spaltennamen im CSV-Format ausgegeben wird. Dieser erste Satz hat das gleiche Format wie die folgenden Sätze mit den Ausgabedaten.

Bei multiplen Spalten wird für jede betroffene Ausprägung der Spaltenname ausgegeben. Die Ausprägungsnummer wird, in runden Klammern eingeschlossen, dem Spaltennamen angehängt.

Spaltennamen haben den Datentyp CHARACTER. Deshalb muss bei Angabe von UTFE für die Datenbank ein CCSN definiert sein.

Wenn bereits das Schreiben des ersten Satzes zu einem Eintrag in der Fehlerdatei führt, dann wird die Anweisung mit SQLSTATE abgebrochen.



Beim Laden von Anwenderdaten mit LOAD aus einer CSV-Eingabedatei muss die Überschriftenzeile mit SKIP FIRST *zahl* RECORDS überlesen werden, siehe [Seite 149](#).

Beim Lesen über die Tabellenfunktion CSV() kann die Überschriftenzeile durch den Parameter WITH ORDINALITY und einer entsprechenden WHERE-Klausel überlesen werden, siehe Handbuch „[SQL-Sprachbeschreibung Teil 1: SQL-Anweisungen](#)“.

**WHERE** *suchbedingung*

Es werden nur die Anwenderdaten ausgegeben, die die Suchbedingung erfüllen. In der Suchbedingung dürfen keine Benutzervariablen und keine Fragezeichen als Platzhalter verwendet werden.

Es können andere Tabellen als die angegebene Tabelle *tabelle* referenziert werden. Die referenzierten Tabellen dürfen jedoch keine Nur-CALL-DML-Tabellen sein.

Die Abarbeitung der Suchbedingung auf andere referenzierte Tabellen erfolgt in dem Standard-Isolationslevel der Applikation. Dieses gilt nicht, wenn Sie mit SET TRANSACTION oder dem Pragma ISOLATION LEVEL einen anderen Wert angegeben haben.

Bei WHERE *suchbedingung* sind weitere Pragmas wirksam (siehe Handbuch „[SQL-Sprachbeschreibung Teil 1: SQL-Anweisungen](#)“). Wenn die WHERE-Klausel angegeben ist, dann muss auch der Parameter ONLINE angegeben werden.

**ORDER BY**

Die ORDER BY-Klausel definiert die Reihenfolge, in der die gelesenen Sätze in die Ausgabedatei geschrieben werden sollen. Es wird zuerst nach den Werten der ersten angegebenen Spalte sortiert. Wenn Sätze vorkommen, die in der ersten Spalte nach den Vergleichsregeln gleiche Werte haben, dann werden diese gemäß der zweiten Sortierspalte sortiert usw. In SESAM/SQL sind NULL-Werte beim Sortieren kleiner als alle Nicht-NULL-Werte.

Die Reihenfolge der Sätze mit gleichen Werten in allen Sortierspalten ist undefiniert.

Wenn die ORDER BY-Klausel angegeben ist, dann muss auch der Parameter ONLINE angegeben werden.

*spalte*

Name der Spalte, nach der sortiert werden soll. *spalte* muss ein einfacher Spaltenname ohne Tabellennamen sein. Er muss zu der Ausgabemenge gehören, die mit der WHERE-Klausel erzeugt wird.

*spalte(posnr)*

Element einer multiplen Spalte, nach dem sortiert werden soll. *posnr* ist eine vorzeichenlose Ganzzahl, die die Positionsnummer des Spaltenelements in der multiplen Spalte angibt. Das Spaltenelement muss zu der Ausgabemenge gehören, die mit der WHERE-Klausel erzeugt wird.

*spaltennummer*

Nummer der Spalte, nach der sortiert werden soll.

*spaltennummer* ist eine vorzeichenlose Ganzzahl mit:  $1 \leq \text{spaltennummer} \leq \text{Anzahl der Ausgabespalten}$ .

*spaltennummer* kann eine einfache Spalte oder eine multiple Spalte mit Dimension 1 bezeichnen.

*ausdruck*

Sie können nicht nur nach den Ausgabespalten sortieren, sondern auch nach weiteren Ausdrücken, wie zum Beispiel `UPPER(spalte)`.

Dabei müssen folgende Bedingungen erfüllt sein:

- *ausdruck* darf nicht nur aus einem Literal bestehen
- *ausdruck* darf keine Unterabfrage und keine Mengenfunktion enthalten
- in *ausdruck* dürfen Spalten der Tabelle *tabelle* verwendet werden, auch wenn sie nicht in der Liste der Ausgabespalten vorkommen.

## ASC / DESC

Die Werte der zugehörigen Spalte werden aufsteigend (ASC) oder absteigend (DESC) sortiert.

## Fehlerdatei bei UNLOAD

SESAM/SQL legt beim UNLOAD bei Bedarf eine neue Datei an oder schreibt die bestehende fort. Der Dateiname lautet:

*catalog.space*.EXC.U bei UNLOAD OFFLINE bzw.  
*datei*.EXC.U bei UNLOAD ONLINE

*catalog* Name der Datenbank.

*space* Name des Anwender-Space, auf dem die Tabelle liegt, von der ausgegeben werden soll.

Bei einer partitionierten Tabelle wird der Spacename der ersten Partition verwendet. Die Fehlerdatei enthält die Fehlerinformationen für alle Partitionen der Tabelle.

*datei* Name der Datei, die in der Klausel INTO FILE angegeben ist.

EXC.U Standard-Endung des Namens der Fehlerdatei für UNLOAD-Anweisungen

Eine bereits existierende Fehlerdatei wird fortgeschrieben. Die Einträge aus verschiedenen UNLOAD-Anweisungen sind durch eine Überschriftszeile, die Datum und Uhrzeit protokollieren, getrennt. Wenn eine Fehlerdatei mindestens einen Satz enthält, wird eine Warnung ausgegeben.

Wenn *catalog* in einer DB-Kennung liegt, so wird die Fehlerdatei bei UNLOAD OFFLINE in der DB-Kennung angelegt, sofern die Vorbereitungen dafür getroffen worden sind, siehe Abschnitt „Datenbankdateien und Jobvariablen auf fremden Benutzerkennungen“ im „[Basishandbuch](#)“. Kann die Fehlerdatei nicht in der DB-Kennung angelegt werden, so wird sie in der DBH-Kennung angelegt.

Bei UNLOAD ONLINE wird die Fehlerdatei in der gleichen Kennung wie die Ausgabedatei *datei* angelegt. Wenn diese Kennung ungleich der DBH-Kennung ist, dann müssen die Vorbereitungen dafür getroffen worden sein, siehe Abschnitt „Datenbankdateien und Jobvariablen auf fremden Benutzerkennungen“ im „[Basishandbuch](#)“. Kann die Fehlerdatei nicht in dieser Kennung angelegt werden, so wird sie in der DBH-Kennung angelegt.

### *Aufbau der Fehlerdatei bei UNLOAD*

Die Fehlerdatei bei UNLOAD besteht aus zwei Kopfzeilen mit dem Anweisungsnamen und Datum/Uhrzeit der Anweisungsausführung sowie dem Namen der Ausgabedatei.

Dann folgen fehler-spezifische Wertezeilen mit folgendem Aufbau:

```
<zähler> <nummer> <sqlstate> <spalte> [<wert>]
```

Dabei bedeuten:

- <zähler> gibt an, nach welchem Satz der Ausgabedatei der in die Fehlerdatei geschriebene Satz in die Ausgabedatei einzuordnen wäre
- <nummer> gibt die Nummer des bearbeiteten Satzes an
- <sqlstate> gibt den SQLSTATE an
- <spalte> gibt die Tabellenspalte an
- <wert> gibt den Spaltenwert an (hexadezimal / abdruckbar)

Folgende Typen von Einträgen können in der UNLOAD-Fehlerdatei auftreten:

- Werte, die nicht transliteriert oder konvertiert werden können (SQLSTATE 22SB2); Werte, die beim CSV-Format NEWLINE-Zeichen enthalten ohne dass ein QUOTE-Zeichen definiert ist (SQLSTATE 22SB5).

Sie werden in der Fehlerdatei hexadezimal und abdruckbar in einer maximalen Länge von 32000 Byte (16000 Zeichen) dargestellt. Wenn in einem Satz mehrere Werte nicht transliteriert oder konvertiert werden können, dann wird für jeden dieser Werte ein Eintrag in die Fehlerdatei geschrieben. Durch die <nummer> kann festgestellt werden, dass sich die Fehlereinträge auf denselben Satz beziehen.

- Werte, die das DELIMITER-Zeichen enthalten (SQLSTATE 22SB3).  
Sie werden in der Fehlerdatei abdruckbar in einer maximalen Länge von 16000 Zeichen dargestellt.
- Spalten vom Datentyp (NATIONAL) CHARACTER VARYING, die einen Wert der Länge NULL enthalten (SQLSTATE 22SB4).  
Es wird ein Eintrag in die Fehlerdatei geschrieben, aber ohne <wert>.
- Bei UNLOAD mit selbstdefiniertem Format tritt ein Über- oder Unterlauf eines numerischen Wertes auf.  
Es wird der SQLSTATE 22003 in folgender Form in die Fehlerdatei eingetragen:  
<zähler> <nummer> 22003 <spalte>
- Bei UNLOAD mit selbstdefiniertem Format ist der Spaltenwert länger als der im Ausgabesatz definierte Bereich.  
Es wird die Warnung 01004 in folgender Form in die Fehlerdatei eingetragen:  
<zähler> <nummer> 01004 <spalte>
- Bei UNLOAD mit selbstdefiniertem Format werden Kommastellen gerundet oder abgeschnitten.  
Es wird die Warnung 01SB3 in folgender Form in die Fehlerdatei eingetragen:  
<zähler> <nummer> 01SB3 <spalte>



Wenn eine Warnung in die Fehlerdatei eingetragen wurde, dann wurde der Satz auch in die Ausgabedatei geschrieben. <zähler> +1 bezeichnet den Satz in der Ausgabedatei.

## UNLOAD-Formate

Die folgende Tabelle enthält eine Übersicht über die UNLOAD-Formate.

In den Beispielen für UNLOAD-Anweisungen sind nicht alle möglichen Klauseln berücksichtigt (z.B. FROM SPACE, FROM COPY FILE ...).

Format	Anweisung (Beispiel)	Entladene Spalten	Nullwerte	Weiterverarbeitung durch
TRANSFER_FORMAT	UNLOAD TABLE <i>tabelle</i> INTO FILE <i>datei</i> TRANSFER_FORMAT	Alle Spalten laut Reihenfolge von <i>tabelle</i>	Interne Darstellung (Y/N für alle Spalten am Ende eines Satzes)	LOAD-Anweisung mit TRANSFER_ FORMAT
	UNLOAD DATA <i>tabelle</i> ( <i>spalte, spalte, ...</i> ) INTO FILE <i>datei</i> TRANSFER_FORMAT	Nur die ange- gebenen Spal- ten in dieser Reihenfolge		
LOAD_FORMAT	UNLOAD TABLE <i>tabelle</i> INTO FILE <i>datei</i> LOAD_FORMAT	Alle Spalten laut Reihenfolge von <i>tabelle</i>	Interne Darstellung (negativer Indikator vor entsprechender Spalte; volle Attri- butlänge, außer bei (N)VARCHAR)	LOAD-Anweisung mit UNLOAD_ FORMAT
	UNLOAD DATA <i>tabelle</i> ( <i>spalte, spalte, ...</i> ) INTO FILE <i>datei</i> LOAD_FORMAT	Nur die ange- gebenen Spal- ten in dieser Reihenfolge		
DELIMITER_FORMAT	UNLOAD TABLE <i>tabelle</i> INTO FILE <i>datei</i> DELIMITER_FORMAT TERMINATED BY <i>delimiter</i>	Alle Spalten laut Reihenfolge von <i>tabelle</i>	Nur Delimiter- Zeichen	LOAD-Anweisung mit DELIMITER_ FORMAT bzw. Editor
	UNLOAD DATA <i>tabelle</i> ( <i>spalte, spalte, ...</i> ) INTO FILE <i>datei</i> DELIMITER_FORMAT TERMINATED BY <i>delimiter</i>	Nur die ange- gebenen Spal- ten in dieser Reihenfolge		

Tabelle 10: UNLOAD-Ausgabeformate

(Teil 1 von 2)

Format	Anweisung (Beispiel)	Entladene Spalten	Nullwerte	Weiterverarbeitung durch
CSV_FORMAT	UNLOAD TABLE <i>tabelle</i> INTO FILE <i>datei</i> CSV_FORMAT DELIMITER <i>delimiter</i> QUOTE <i>quote</i> ESCAPE <i>escape</i>	Alle Spalten laut Reihenfolge von <i>tabelle</i>	Nur DELIMITER-Zeichen	LOAD-Anweisung mit CSV_FORMAT bzw. Editor oder Kalkulationsprogramm
	UNLOAD DATA <i>tabelle</i> ( <i>spalte, spalte, ...</i> ) INTO FILE <i>datei</i> CSV_FORMAT DELIMITER <i>delimiter</i>	Nur die angegebenen Spalten in dieser Reihenfolge		
Standard-Format	UNLOAD TABLE <i>tabelle</i> INTO FILE <i>datei</i>	Alle Spalten laut Reihenfolge von <i>tabelle</i>	Blank oder Null in voller Attributlänge, außer bei (N)VARCHAR; bei LOAD nicht mehr als Nullwert erkennbar!	LOAD-Anweisung mit Standard- oder selbst definiertem Format
	UNLOAD DATA <i>tabelle</i> ( <i>spalte, spalte, ...</i> ) INTO FILE <i>datei</i>	Nur die angegebenen Spalten in dieser Reihenfolge		
Selbst definiertes Format	UNLOAD TABLE <i>tabelle</i> INTO FILE <i>datei</i> ( <i>ausgabedarstellung, ausgabedarstellung WHEN NULL THEN nulldescriptor, ...</i> )	Alle Spalten laut Reihenfolge von <i>tabelle</i>	Als <i>nulldescriptor</i> , falls dieser definiert ist; sonst wie beim Standard-Format	LOAD-Anweisung mit selbst definiertem Format bzw. Editor
	UNLOAD DATA <i>tabelle</i> ( <i>spalte, spalte, ...</i> ) INTO FILE <i>datei</i> ( <i>ausgabedarstellung, ausgabedarstellung WHEN NULL THEN nulldescriptor, ...</i> )	Nur die angegebenen Spalten in dieser Reihenfolge		
Mischung von Standard- und selbst definiertem Format	UNLOAD TABLE <i>tabelle</i> ( <i>spalte, spalte, ...</i> ) INTO FILE <i>datei</i> ( <i>ausgabedarstellung, ausgabedarstellung WHEN NULL THEN nulldescriptor, ...</i> )	Alle Spalten mit selbst definiertem Format für die angegebenen Spalten; Standard-Format für alle anderen Spalten	Als <i>nulldescriptor</i> , falls dieser definiert ist; sonst wie beim Standard-Format	LOAD-Anweisung mit selbst definiertem Format

Tabelle 10: UNLOAD-Ausgabeformate

(Teil 2 von 2)

## Beispiele

Die folgenden Beispiele geben Daten der Basistabelle KUNDE aus (siehe Beispiel zur Utility-Anweisung LOAD auf [Seite 163](#)). Die Tabelle KUNDE ist definiert im Schema AUFTRAGSVER der Datenbank AUFTRAGKUNDEN.

1. Die Daten werden (offline) im Delimiter-Format in die Ausgabedatei DAT.UNLOAD.D.KUNDE ausgegeben. Diese Ausgabedatei eignet sich als Eingabedatei für eine nachfolgende LOAD-Anweisung mit der Klausel DELIMITER\_FORMAT TERMINATED BY ';':

Wenn in der Tabelle KUNDE für PLZ kein Wert abgespeichert ist, wird an entsprechender Stelle in der Ausgabedatei nur das DELIMITER-Zeichen abgelegt.



```
UNLOAD TABLE auftragkunden.auftragsver.kunde
INTO FILE 'DAT.UNLOAD.D.KUNDE'
DELIMITER_FORMAT TERMINATED BY ';'
```

Nach Ausführung von UNLOAD hat die Ausgabedatei folgenden Inhalt:

```
100;Siemens AG;Otto-Hahn-Ring 6;81739;Muenchen;D;089/636-8;Elektro;
101;Login GmbH;Rosenheimer Str. 34;81667;Muenchen;D;089/4488870;PC Netzwerke;
102;JIKO GmbH;Posener Str. 12;30659;Hannover;D;0551/123874;Import-Export;
103;Plenzer Trading;Paul-Heyse-Str. 12;80336;Muenchen;D;089/923764;Fruechtehandel;
104;Jonas Fischladen;Hirschgartenstr. 12;12587;Berlin;D;016/5739921;Einzelhandel;
105;Pudelshop Anke;Am Muehlentor 26;41179;Moenchengladbach;D;040/873562;Dienstleistung;
106;Foreign Ltd.;26 West York St.;New York, NY;USA;001703/2386532;Handelsvertretung;
107;Externa & Co KG;Bernner Weg 78;3000;Bern 33;CH;;Anwaltskanzlei;
```

Die Anzahl der entladenen Sätze wird im Feld SQLrowcount der Diagnostic Area zurückgemeldet.

2. Die Daten werden online im Unload-Format in die Ausgabedatei DAT.UNLOAD.L.KUNDE ausgegeben. Dabei werden in der Tabelle nur die deutschen Kunden zur Ausgabe ausgewählt. Diese werden in der Ausgabedatei nach der Postleitzahl und (bei gleicher Postleitzahl) dem Firmennamen sortiert. Die Ausgabedatei eignet sich als Eingabedatei für eine nachfolgende Utility-Anweisung LOAD mit der Klausel UNLOAD\_FORMAT.

```
UNLOAD ONLINE TABLE auftragkunden.auftragsver.kunde
INTO FILE 'DAT.UNLOAD.L.KUNDE' LOAD_FORMAT
WHERE land = 'D' ORDER BY plz,firma
```

3. Die Daten werden (offline) in die Ausgabedatei DAT.UNLOAD.F.KUNDE ausgegeben. Die Werte des Primärschlüssels KNR der Tabelle KUNDE sowie die Werte von PLZ werden in der Ausgabedatei in lesbarer Form abgelegt. Dabei ist zu beachten, dass die einer Ausgabespalte vom Datentyp INTEGER zugeordnete CHARACTER-Ausgabedarstellung mindestens in der Länge CHARACTER(10) vereinbart sein muss. Wenn in der Tabelle KUNDE für PLZ kein Wert abgespeichert ist, wird an entsprechender Stelle in der Ausgabedatei die Zeichenkette „XXXXX“ abgelegt.



```
UNLOAD TABLE auftragkunden.auftragsver.kunde
(knr, firma, strasse, plz, ort, land, ktelefon, kinfo)
INTO FILE 'DAT.UNLOAD.F.KUNDE'
(PPOSITION(1) CHARACTER(10),
 PPOSITION(11) CHARACTER(40),
 PPOSITION(51) CHARACTER(40),
 PPOSITION(91) CHARACTER(5) WHEN NULL THEN 'XXXXX',
 PPOSITION(96) CHARACTER(40),
 PPOSITION(136) CHARACTER(3),
 PPOSITION(139) CHARACTER(25),
 PPOSITION(164) CHARACTER(50))
```

Nach der Ausführung von UNLOAD hat die Datei DAT.UNLOAD.F.KUNDE folgenden Inhalt:

100	Siemens AG			Otto-Hahn-Ring 6
81739	Muenchen	D	089/636-8	Elektro
101	Login GmbH			Rosenheimer Str.34
81667	Muenchen	D	089/4488870	PC Netzwerke
102	JIKO GmbH			Posener Str. 12
30659	Hannover	D	0551/123874	Import-Export
103	Plenzer Trading			Paul-Heyse-Str. 12
80336	Muenchen	D	089/923764	Fruechtehandel
104	Jonas Fischladen			Hirschgartenstr. 12
12587	Berlin	D	016/5739921	Einzelhandel
105	Pudelshop Anke			Am Muehlentor 26
41179	Moenchengladbach	D	040/873562	Dienstleistung
106	Foreign Ltd.			26 West York St.
XXXXX	New York, NY	USA	001703/2386532	Handelsvertretung
107	Externa & Co KG			Berner Weg 78
03000	Bern 33	CH		Anwaltskanzlei

**Siehe auch**

**LOAD, EXPORT/IMPORT TABLE**

---

## 4 Anhang

### 4.1 Syntaxübersicht

Im Folgenden sind die im vorliegenden Handbuch benötigten Regeln für Syntaxelemente in alphabetischer Reihenfolge zusammengestellt. Syntaxelemente, die nicht im vorliegenden Handbuch, sondern im Handbuch "[SQL-Sprachbeschreibung Teil 1: SQL-Anweisungen](#)" erklärt sind, sind durch den Zusatz „(Teil 1)“ gekennzeichnet.



In der Syntax kursiv gedruckte eckige Klammern sind Sonderzeichen und müssen in der Anweisung angegeben werden.

$$\textit{alphanumerisches\_literal} ::= \left\{ \begin{array}{l} '[\textit{zeichen} \dots]' [\{\textit{trenner} \dots' [\textit{zeichen} \dots]' \} \dots] \\ 'X' [\textit{hex hex}] \dots' [\{\textit{trenner} \dots' [\textit{hex hex}] \dots' \} \dots] \end{array} \right\} \text{ (Teil 1)}$$
$$\textit{ausgabedarstellung} ::= \text{POSITION} \left( \left\{ \begin{array}{l} \textit{zahl} \\ * \end{array} \right\} \right) [\textit{datentyp}] [\text{WHEN NULL THEN } \textit{nulldeskriptor}]$$
$$\textit{ausgabespalte} ::= \left\{ \begin{array}{l} \textit{spalte} \\ \textit{spalte}(\textit{zahl}) \\ \textit{spalte}(\textit{min}..\textit{max}) \end{array} \right\}$$
$$\textit{berechtigungsschlüssel} ::= \textit{einf\_name} \text{ (Teil 1)}$$
$$\textit{bs2000\_benutzer} ::= \left( \left\{ \begin{array}{l} \textit{rechnername} \\ * \end{array} \right\} \right), [*], \left\{ \begin{array}{l} \textit{bs2000\_benutzerkennung} \\ * \end{array} \right\}$$
$$\textit{catalog} ::= \textit{einf\_name} \text{ (Teil 1)}$$
$$\textit{catrec\_datei} ::= \textit{einf\_name} \text{ (Teil 1)}$$

$$\text{datentyp} ::= \left. \begin{array}{l}
 \left[ \left. \begin{array}{l} \{ [dimension] \\ (dimension) \} \end{array} \right] \text{CHARACTER} [(länge)] \\
 \left. \begin{array}{l} \{ \text{CHARACTER VARYING}(max) \\ \text{VARCHAR}(max) \} \\
 \left[ \left. \begin{array}{l} \{ [dimension] \\ (dimension) \} \right] \left. \begin{array}{l} \{ \text{NATIONAL CHARACTER} \\ \text{NCHAR} \} \right] [(cu\_länge \text{ [CODE\_UNITS]})] \\
 \left. \begin{array}{l} \{ \text{NATIONAL CHARACTER VARYING} \\ \text{NCHAR VARYING} \\ \text{NVARCHAR} \} \right] (cu\_max \text{ [CODE\_UNITS]}) \\
 \left[ \left. \begin{array}{l} \{ [dimension] \\ (dimension) \} \right] \left. \begin{array}{l} \{ \text{SMALLINT} \\ \text{INTEGER} \\ \text{NUMERIC}[(stellen[,bruchteil])] \\ \text{DECIMAL}[(stellen[,bruchteil])] \\ \text{REAL} \\ \text{DOUBLE PRECISION} \\ \text{FLOAT}[(stellen)] \\ \text{DATE} \\ \text{TIME}(3) \\ \text{TIMESTAMP}(3) \end{array} \right\}
 \end{array} \right\}
 \end{array} \right\}$$

*einf\_basistabellenname* ::= *einf\_name* (Teil 1)

*einf\_bedingungsname* ::= *einf\_name* (Teil 1)

*einf\_indexname* ::= *einf\_name* (Teil 1)

*einf\_name* ::=  $\left. \begin{array}{l} \{ regulärer\_name \\ spezialname \} \right\} \text{ (Teil 1)}$

*einf\_schemaname* ::= *einf\_name* (Teil 1)

*einf\_spacename* ::= *einf\_name* (Teil 1)

*einf\_stogroupname* ::= *einf\_name* (Teil 1)

*festpunktzahl* ::=  $\left[ \left. \begin{array}{l} \{ + \\ - \} \right] \left. \begin{array}{l} \{ \text{vorzeichenlose\_ganzzahl}[\text{.vorzeichenlose\_ganzzahl}] \\ \text{vorzeichenlose\_ganzzahl} [.] \\ \text{.vorzeichenlose\_ganzzahl} \end{array} \right\} \text{ (Teil 1)}$

*ganzzahl* ::=  $\left[ \left. \begin{array}{l} \{ + \\ - \} \right] \text{vorzeichenlose\_ganzzahl} [.] \text{ (Teil 1)}$

*gleitpunktzahl* ::= *festpunktzahl*  $\left[ \left. \begin{array}{l} \{ + \\ - \} \right] \text{vorzeichenlose\_ganzzahl} \text{ (Teil 1)}$

*index* ::= [[*catalog*.]*einf\_schemaname*.]*einf\_indexname* (Teil 1)

*integritätsbedingungsname* ::= [[*catalog*.]*einf\_schemaname*.]*einf\_bedingungsname* (Teil 1)

*ladedarstellung* ::= POSITION (  $\left\{ \begin{array}{c} \textit{zahl} \\ * \end{array} \right\}$  ) [*datentyp*][WHEN *nullbedingung* THEN NULL]

*ladeparameter* ::=  $\left\{ \begin{array}{l} \text{[NO] OVERWRITE} \\ \left\{ \begin{array}{l} \text{NO INDEX} \\ \text{GENERATE INDEX} \end{array} \right\} \\ \text{[NO] CONSTRAINT CHECK} \end{array} \right\}$

*ladespalte* ::=  $\left\{ \begin{array}{l} \textit{spalte} \\ \textit{spalte}(\textit{posnr}) \\ \textit{spalte}(\textit{min..max}) \end{array} \right\}$

*letzte\_partition* ::= PARTITION *partnr* [VALUE <=( )] ON SPACE *space* (Teil 1)

*literal* ::=  $\left\{ \begin{array}{l} \textit{alphanumerisches\_literal} \\ \textit{national\_literal} \\ \textit{spezial\_literal} \\ \textit{numerisches\_literal} \\ \textit{zeit\_literal} \end{array} \right\}$  (Teil 1)

*max* ::= *vorzeichenlose\_ganzzahl* (Teil 1)

*min* ::= *vorzeichenlose\_ganzzahl* (Teil 1)

*national\_literal* ::=

$\left\{ \begin{array}{l} \text{N}'[\textit{zeichen} \dots]'[\{\textit{trenner} \dots'[\textit{zeichen} \dots]'\} \dots] \\ \text{NX}'[4\textit{hex} \dots]'[\{\textit{trenner} \dots'[4\textit{hex} \dots]'\} \dots] \\ \text{U\&}'\left[ \begin{array}{c} \textit{zeichen} \\ \textit{esc 4hex} \\ \textit{esc+} \\ \textit{6hex} \end{array} \right] \dots'[\{\textit{trenner} \dots'[\begin{array}{c} \textit{zeichen} \\ \textit{esc 4hex} \\ \textit{esc+ 6hex} \\ \textit{esc esc} \end{array}]'\} \dots][\text{UESCAPE}'\textit{esc}'] \end{array} \right\}$  (Teil 1)

*nullbedingung* ::=  $\left\{ \begin{array}{l} \textit{spalte} \textit{vergleichs\_op} \textit{literal} \\ \text{POSITION} \left( \left\{ \begin{array}{c} \textit{zahl} \\ * \end{array} \right\} \right) \textit{vergleichs\_op} \left\{ \begin{array}{l} \textit{alphanumerisches\_literal} \\ \textit{national\_literal} \end{array} \right\} \end{array} \right\}$

$$\text{nulldeskriptor} ::= \left\{ \begin{array}{l} \textit{literal} \\ \text{POSITION}(\textit{zahl}) = \textit{alphanumerisches\_literal} \end{array} \right\}$$

$$\text{numerische\_funktion} ::= \left\{ \begin{array}{l} \text{ABS}(\textit{ausdruck}) \\ \text{CEIL}[\text{ING}](\textit{ausdruck}) \\ \text{FLOOR}(\textit{ausdruck}) \\ \text{MOD}(\textit{dividend}, \textit{divisor}) \\ \text{SIGN}(\textit{ausdruck}) \\ \text{TRUNC}(\textit{ausdruck}) \\ \left\{ \begin{array}{l} \text{CHAR\_LENGTH} \\ \text{CHARACTER\_LENGTH} \end{array} \right\}(\textit{ausdruck}[\text{USING} \left\{ \begin{array}{l} \text{CODE\_UNITS} \\ \text{OCTETS} \end{array} \right\} ]]) \\ \text{OCTET\_LENGTH}(\textit{ausdruck}) \\ \text{POSITION}(\textit{ausdruck} \text{ IN } \textit{ausdruck} [\text{USING} \text{CODE\_UNITS}]) \\ \text{JULIAN\_DAY\_OF\_DATE}(\textit{ausdruck}) \\ \text{EXTRACT}(\textit{bestandteil} \text{ FROM } \textit{ausdruck}) \end{array} \right\}$$

$$\text{numerisches\_literal} ::= \left\{ \begin{array}{l} \textit{ganzzahl} \\ \textit{festpunktzahl} \\ \textit{gleitpunktzahl} \end{array} \right\} \text{ (Teil 1)}$$

$$\text{partition} ::= \text{PARTITION } \textit{partnr} \text{ VALUE } \left\{ \begin{array}{l} < \\ \leq \end{array} \right\} (\textit{spaltenwert}, \dots) \text{ ON SPACE } \textit{space} \text{ (Teil 1)}$$

$$\text{partnr} ::= \textit{vorzeichenlose\_ganzzahl} \text{ (Teil 1)}$$

$$\text{posnr} ::= \textit{vorzeichenlose\_ganzzahl} \text{ (Teil 1)}$$

$$\text{pragma} ::= \text{--\%PRAGMA } \textit{pragma\_text}, \dots \text{ zeilenende} \text{ (Teil 1)}$$

$$\text{rec\_unit} ::= \left\{ \begin{array}{l} \text{COPY\_FILE } \textit{datei} \\ \text{COPY\_NUMBER } \textit{zahl} \\ \textit{zeitstempel\_literal} \\ \text{FOREIGN COPY} \end{array} \right\}$$

$$\text{regulärer\_name} ::= \textit{buchstabe} \left[ \left\{ \begin{array}{l} \textit{buchstabe} \\ \textit{ziffer} \\ - \end{array} \right\} \right] \dots \text{ (Teil 1)}$$

$$\text{replik} ::= \textit{catalog}$$

$$\text{replik\_beschreibung} ::= \text{REPLICATION } \textit{replik} \\ \quad [\text{PASSWORD } \textit{kennwort}] \\ \quad [\text{ON USER\_ID } \textit{bs2000\_benutzerkennung}] \\ \quad [\text{WITH CATREC } \textit{catrec\_datei} \\ \quad \quad \text{PASSWORD } \textit{kennwort} \text{ ON USER\_ID } \textit{bs2000\_benutzerkennung}]] \\ \quad \left\{ \begin{array}{l} \text{COPY} \\ \text{RENAME} \end{array} \right\}$$

*schema* ::= [catalog.]*einf\_schemaname* (Teil 1)

*space* ::= [catalog.]*einf\_spacename* (Teil 1)

$$space\_parameter\_liste ::= \left. \begin{array}{l} PRIMARY\ zahl \\ SECONDARY\ zahl \\ PCTFREE\ zahl \\ [NO]\ SHARE \\ [NO]\ DESTROY \\ NO\ LOG \end{array} \right\} \dots$$

*spalte* ::= *einf\_name* (Teil 1)

$$spaltenwert ::= \left. \begin{array}{l} alphanumerisches\_literal \\ national\_literal \\ numerisches\_literal \\ zeit\_literal \end{array} \right\} (Teil\ 1)$$

*spezialname* ::= "zeichen..." (Teil 1)

*stogroup* ::= [catalog.]*einf\_stogroupname* (Teil 1)

$$stogroup\_attributes ::= \left. \begin{array}{l} VOLUMES(volumename\ \dots)\ ON\ gerätetyp \\ PUBLIC \end{array} \right\} [ON\ katalogkennung]$$

$$suchbedingung ::= \left. \begin{array}{l} praedikat \\ suchbedingung \left. \begin{array}{l} \{AND\} \\ \{OR\} \end{array} \right\} suchbedingung \\ NOT\ suchbedingung \\ (suchbedingung) \end{array} \right\}$$

*systembenutzer\_kennung* ::=  $\left. \begin{array}{l} utm\_benutzer \\ bs2000\ benutzer \end{array} \right\}$

$$tabelle ::= \left. \begin{array}{l} [[catalog.]\i{einf\_schemaname}.\i{einf\_basistabellenname} \\ [[catalog.]\i{einf\_schemaname}.\i{einf\_viewname} \\ korrelation \end{array} \right\} (Teil\ 1)$$

$$utm\_benutzer ::= \left( \left. \begin{array}{l} rechnername \\ * \end{array} \right\} , \left. \begin{array}{l} utm\_anwendungsname \\ * \end{array} \right\} , \left. \begin{array}{l} utm\_benutzerkennung \\ * \end{array} \right\} \right)$$

$$vergleichs\_op ::= \left. \begin{array}{l} = \\ < \\ > \\ <= \\ >= \\ <> \end{array} \right\}$$

$$\text{voreinstellung} ::= \text{DEFAULT } \left. \begin{array}{l} \textit{alphanumerisches\_literal} \\ \textit{national\_literal} \\ \textit{numerisches\_literal} \\ \textit{zeit\_literal} \\ \text{CURRENT\_DATE} \\ \text{CURRENT\_TIME}(3) \\ \text{LOCALTIME}(3) \\ \text{CURRENT\_TIMESTAMP}(3) \\ \text{LOCALTIMESTAMP}(3) \\ \text{USER} \\ \text{CURRENT\_USER} \\ \text{SYSTEM\_USER} \\ \text{NULL} \\ \text{REF}(\textit{tabelle}) \end{array} \right\} \text{(Teil 1)}$$

*vorzeichenlose\_ganzzahl* ::= *ziffer*... (Teil 1)

$$\text{zeitfunktion} ::= \left. \begin{array}{l} \text{CURRENT\_DATE} \\ \text{CURRENT\_TIME}(3) \\ \text{LOCALTIME}(3) \\ \text{CURRENT\_TIMESTAMP}(3) \\ \text{LOCALTIMESTAMP}(3) \\ \text{DATE\_OF\_JULIAN\_DAY}(\textit{ausdruck}) \end{array} \right\} \text{(Teil 1)}$$

$$\text{zeit\_literal} ::= \left. \begin{array}{l} \text{DATE } \textit{'jahr-monat-tag'}$$

$$\text{TIME } \textit{'stunde:minute:sekunde'}$$

$$\textit{zeitstempel\_literal} \end{array} \right\} \text{(Teil 1)}$$

*zeitstempel\_literal* ::= `TIMESTAMP` *'jahr-monat-tag stunde:minute:sekunde'* (Teil 1)

## 4.2 SQL-Schlüsselwörter

In SESAM/SQL gibt es Wörter, die als Schlüsselwörter für SQL- und Utility-Anweisungen reserviert sind. Die reservierten Schlüsselwörter dürfen Sie in SQL- und Utility-Anweisungen sowie beim Arbeiten mit dem Utility-Monitor nicht als Namen von Views, Tabellen, Spalten usw. angeben, es sei denn, Sie geben das betreffende Schlüsselwort in Form eines Spezialnamens an.

Die Synonymverarbeitung des ESQL-Precompilers bietet eine komfortable Möglichkeit zum Ersetzen von Schlüsselwörtern bzw. zur Redefinition von Namen.

Mit Hilfe der Precompiler-Option SOURCE-PROPERTIES kann der Anwender den Parameter ESQL-DIALECT auf ISO, OLD oder ALL-FEATURES setzen. Dadurch wird entschieden, ob der SQL-Dialekt ISO, OLD oder FULL zu benutzen ist.

Die folgende Tabelle zeigt die reservierten Schlüsselwörter und ordnet sie den SQL-Dialekten zu, in denen sie reserviert sind.

Schlüsselwort	ISO	OLD	FULL
ABS	X		X
ABSOLUTE	X		X
ACTION	X		X
ADD	X		X
ALL	X	X	X
ALLOCATE	X		X
ALTER	X		X
AND	X	X	X
ANY	X	X	X
ARE	X		X
AS	X	X	X
ASC	X	X	X
ASSERTION	X		X
AT	X		X
AUTHORIZATION	X	X	X
AVG	X	X	X
BEGIN	X	X	X
BETWEEN	X	X	X
BIT	X		X

Tabelle 11: Reservierte Schlüsselwörter von SESAM/SQL

(Teil 1 von 10)

Schlüsselwort	ISO	OLD	FULL
BIT_LENGTH	X		X
BLOB	X		X
BOTH	X		X
BY	X	X	X
CALL	X		X
CASCADE	X		X
CASCADED	X		X
CASE	X		X
CAST	X		X
CATALOG	X		X
CEIL	X		X
CEILING	X		X
CHAR	X	X	X
CHARACTER	X	X	X
CHARACTER_LENGTH	X		X
CHAR_LENGTH	X		X
CHECK	X	X	X
CLOSE	X	X	X
COALESCE	X		X
COLLATE	X		X
COLLATION	X		X
COLUMN	X		X
COMMIT	X	X	X
CONNECT	X		X
CONNECTION	X		X
CONSTRAINT	X		X
CONSTRAINTS	X		X
CONTINUE	X	X	X
CONVERT	X		X
COPY			X
CORRESPONDING	X		X

Tabelle 11: Reservierte Schlüsselwörter von SESAM/SQL

(Teil 2 von 10)

Schlüsselwort	ISO	OLD	FULL
COUNT	X	X	X
CREATE	X	X	X
CROSS	X		X
CURRENT	X	X	X
CURRENT_CATALOG	X		X
CURRENT_DATE	X		X
CURRENT_ISOLATION_LEVEL			X
CURRENT_REFERENCED_CATALOG			X
CURRENT_SCHEMA	X		X
CURRENT_TIME	X		X
CURRENT_TIMESTAMP	X		X
CURRENT_USER	X		X
CURSOR	X	X	X
DATA	X		X
DATE	X		X
DATE_OF_JULIAN_DAY			X
DAY	X		X
DEALLOCATE	X		X
DEC	X	X	X
DECIMAL	X	X	X
DECLARE	X	X	X
DECLARE	X	X	X
DECRYPT	X	X	X
DEFAULT	X	X	X
DEFERRABLE	X		X
DEFERRED	X		X
DELETE	X	X	X
DESC	X	X	X
DESCRIPTOR	X		X
DIAGNOSTICS	X		X
DIRECTORY			X

Tabelle 11: Reservierte Schlüsselwörter von SESAM/SQL

(Teil 3 von 10)

Schlüsselwort	ISO	OLD	FULL
DISCONNECT	X		X
DISTINCT	X	X	X
DOMAIN	X		X
DOUBLE	X	X	X
DROP	X		X
ELSE	X		X
ENCRYPT	X	X	X
END	X	X	X
END-EXEC		X	
ESCAPE	X	X	X
EXCEPT	X		X
EXCEPTION	X		X
EXEC	X	X	X
EXECUTE	X	X	X
EXISTS	X	X	X
EXP	X		X
EXPORT			X
EXTERNAL	X		X
EXTRACT	X		X
FALSE	X		X
FETCH	X	X	X
FIRST	X	X	X
FLOAT	X	X	X
FLOOR	X		X
FOR	X	X	X
FOREIGN	X	X	X
FOUND	X	X	X
FROM	X	X	X
FULL	X		X

Tabelle 11: Reservierte Schlüsselwörter von SESAM/SQL

(Teil 4 von 10)

Schlüsselwort	ISO	OLD	FULL
GET	X		X
GLOBAL	X		X
GO	X	X	X
GOTO	X	X	X
GRANT	X	X	X
GROUP	X	X	X
HAVING	X	X	X
HEX_OF_VALUE	X	X	X
HOLD	X		X
HOUR	X		X
IDENTITY	X		X
IMMEDIATE	X	X	X
IMPORT			X
IN	X	X	X
INDICATOR	X	X	X
INITIALLY	X		X
INNER	X		X
INPUT	X		X
INSERT	X	X	X
INT	X	X	X
INTEGER	X	X	X
INTERSECT	X		X
INTERVAL	X		X
INTO	X	X	X
IS	X	X	X
ISOLATION	X		X
JOIN	X		X
JULIAN_DAY_OF_DATE			X

Tabelle 11: Reservierte Schlüsselwörter von SESAM/SQL

(Teil 5 von 10)

Schlüsselwort	ISO	OLD	FULL
KEY	X	X	X
LANGUAGE	X	X	X
LAST	X	X	X
LEADING	X		X
LEFT	X		X
LEVEL	X	X	X
LIKE	X	X	X
LIKE_REGEX	X		X
LOAD			X
LOCAL	X		X
LOCALTIME	X		X
LOCALTIMESTAMP	X		X
LOWER	X		X
MATCH	X		X
MAX	X	X	X
MIGRATE			X
MIN	X	X	X
MINUTE	X		X
MOD	X		X
MODIFY			X
MODULE	X	X	X
MONTH	X		X
NAMES	X		X
NATIONAL	X		X
NATURAL	X		X
NCHAR	X		X
NEW	X		X
NEXT	X	X	X
NO	X		X

Tabelle 11: Reservierte Schlüsselwörter von SESAM/SQL

(Teil 6 von 10)

Schlüsselwort	ISO	OLD	FULL
NORMALIZE	X		X
NOT	X	X	X
NULL	X	X	X
NULLIF	X		X
NUMERIC	X	X	X
NVARCHAR			X
OCTET_LENGTH	X		X
OF	X	X	X
OLD	X		X
ON	X	X	X
ONLY	X	X	X
OPEN	X	X	X
OPTION	X	X	X
OR	X	X	X
ORDER	X	X	X
OUTER	X		X
OUTPUT	X		X
OVERLAPS	X		X
PARTIAL	X		X
PERMIT		X	X
POSITION	X		X
POWER	X		X
PRECISION	X	X	X
PREPARE	X	X	X
PRESERVE	X		X
PRIMARY	X	X	X
PRIOR	X	X	X
PRIVILEGES	X	X	X
PROCEDURE	X	X	X
PUBLIC	X	X	X

Tabelle 11: Reservierte Schlüsselwörter von SESAM/SQL

(Teil 7 von 10)

Schlüsselwort	ISO	OLD	FULL
READ	X	X	X
REAL	X	X	X
RECOVER			X
REF	X		X
REFERENCES	X	X	X
REFRESH			X
RELATIVE	X		X
REORG			X
REP_OF_VALUE	X	X	X
RESTORE		X	X
RESTRICT	X		X
RETURN	X	X	X
REVOKE	X		X
RIGHT	X		X
ROLLBACK	X	X	X
ROWS	X		X
SCHEMA	X	X	X
SCOPE	X		X
SCROLL	X	X	X
SECOND	X		X
SECTION	X	X	X
SELECT	X	X	X
SESSION	X		X
SESSION_USER	X		X
SET	X	X	X
SIGN			X
SIZE	X		X
SMALLINT	X	X	X
SOME	X	X	X
SORTED			X

Tabelle 11: Reservierte Schlüsselwörter von SESAM/SQL

(Teil 8 von 10)

Schlüsselwort	ISO	OLD	FULL
SQL	X	X	X
SQLCODE		X	X
SQLERROR	X	X	X
SQLSTATE	X		X
SQRT	X		X
STORE		X	X
SUBSTRING	X		X
SUM	X	X	X
SYSTEM	X		X
SYSTEM_USER	X		X
TABLE	X	X	X
TEMPORARY	X	X	X
THEN	X		X
TIME	X		X
TIMESTAMP	X		X
TIMEZONE_HOUR	X		X
TIMEZONE_MINUTE	X		X
TO	X	X	X
TRAILING	X		X
TRANSACTION	X	X	X
TRANSLATE	X		X
TRANSLATION	X		X
TRIM	X		X
TRUE	X		X
TRUNC			X
UESCAPE	X		X
UNION	X	X	X
UNIQUE	X	X	X
UNKNOWN	X		X
UNLOAD			X

Tabelle 11: Reservierte Schlüsselwörter von SESAM/SQL

(Teil 9 von 10)

Schlüsselwort	ISO	OLD	FULL
UPDATE	X	X	X
UPPER	X		X
USAGE	X		X
USER	X	X	X
USING	X	X	X
VALUE	X		X
VALUES	X	X	X
VALUE_OF_HEX	X	X	X
VALUE_OF_REP	X	X	X
VARCHAR	X		X
VARYING	X		X
VIEW	X	X	X
WHEN	X		X
WHENEVER	X	X	X
WHERE	X	X	X
WITH	X	X	X
WITHOUT	X		X
WORK	X	X	X
WRITE	X	X	X
YEAR	X		X
ZONE	X		X

Tabelle 11: Reservierte Schlüsselwörter von SESAM/SQL

(Teil 10 von 10)

---

# Literatur

Die Handbücher finden Sie im Internet unter <http://manuals.ts.fujitsu.com>. Handbücher, die mit einer Bestellnummer angezeigt werden, können Sie in auch gedruckter Form bestellen.

**SESAM/SQL-Server (BS2000)**

**Basishandbuch**

Benutzerhandbuch

**SESAM/SQL-Server (BS2000)**

**SQL-Sprachbeschreibung Teil 1: SQL-Anweisungen**

Benutzerhandbuch

**SESAM/SQL-Server (BS2000)**

**CALL-DML Anwendungen**

Benutzerhandbuch

**SESAM/SQL-Server (BS2000)**

**Datenbankbetrieb**

Benutzerhandbuch

**SESAM/SQL-Server (BS2000)**

**Utility-Monitor**

Benutzerhandbuch

**SESAM/SQL-Server (BS2000)**

**Fachwörter und Masterindex**

Benutzerhandbuch

**SESAM/SQL-Server (BS2000)**

**Meldungen**

Benutzerhandbuch

**SESAM/SQL-Server (BS2000)**

**Performance**

Benutzerhandbuch

**ESQL-COBOL** (BS2000)

**ESQL-COBOL für SESAM/SQL-Server**

Benutzerhandbuch

**SESAM-DBAccess**

Server-Installation, Administration (nur auf dem Handbuch-Server verfügbar)

**BS2000OSD/BC**

**Kommandos**

Benutzerhandbuch

**ARCHIVE** (BS2000)

Benutzerhandbuch

**EDT** (BS2000)

**Anweisungen**

Benutzerhandbuch

**HSMS** (BS2000)

**Hierarchisches Speicher Management System**

Benutzerhandbuch

**SECOS** (BS2000)

**Security Control System - Access Control**

Benutzerhandbuch

**XHCS** (BS2000)

**8-bit-Code- und Unicode-Unterstützung im BS2000**

Benutzerhandbuch

---

# Stichwörter

Im Stichwortverzeichnis verweisen **halbfette** Seitenzahlen auf die Hauptfundstellen von Stichwörtern und *kursive* Seitenzahlen auf Beispiele. Es gilt folgende Sortierreihenfolge: Symbole vor Ziffern vor Buchstaben. Satzzeichen sind Symbole.

## A

abdruckbar

Darstellung (der Daten) **26**

ADD PARTITION (Utility) **81**

ADD PARTITION ON SPACE (Klausel) **81**

ADD STOGROUP (Klausel) **79**

ALL DATA (Klausel) **131, 137**

allgemeine Regeln

ARCHIVE-Parameterdatei **61**

HSMS-Parameterdatei **49**

ALLOWED-USAGE (HSMS-Operand) **39**

ALTER CATALOG (Utility) **73**

ALTER DATA FOR TABLE (Utility) **74**

ALTER MEDIA DESCRIPTION (Utility) **76, 79**

ALTER PARTITION (Utility) **84**

ALTER PARTITION ON SPACE (Klausel) **84**

ALTER PARTITIONING FOR TABLE **81, 84, 86**

ALTER PARTITIONING FOR TABLE (Utility) **80**

ALTER\_CODE\_TABLE (Klausel) **73**

Ändern

Datenbank-Eigenschaften **73**

Medientabelle **76, 79**

Anonymisieren

von Daten **74**

Anwender-Space **13, 217**

reorganisieren **227**

reparieren **178**

zurücksetzen **178**

Anwenderdaten

ausgeben **234, 255**

exportieren **129**

importieren **133**

laden **140, 163**

Arbeitsdatei

fremde Benutzerkennung **231**

Arbeitsdatei (REORG) **230**

ARCHIVE **97**

ARCHIVE-Lauf

Betriebsmittel reservieren **58**

steuern **56**

ARCHIVE-NAME (HSMS-Operand) **39**

ARCHIVE-Parameter

COMPRESS **60**

DEVICE **59**

DIRSAVE **59**

DRIVES **60**

ENVIRONMENT-ATTRIBUTES **60**

OPERATOR **59**

RETPD **60**

SAVE-ACL **59**

SPACE **60**

STREAM **59**

ARCHIVE-Parameterdatei

allgemeine Regeln **61**

Beispiel **64**

Beschreibung der Parameter **58**

erstellen **56**

Fehlerfälle **62**

Syntaxregeln **61**

Übersicht über die Parameter **57**

AT CATALOG (Klausel) **77, 116, 128, 176**

Attributformat OLDEST **155, 156**

Ausgabedarstellung **241**

Ausgabedatei **65, 234, 240**

fremde Benutzerkennung **65, 240**

Ausgabeformat **33, 234, 242**

Ausgabespalte [238](#), [241](#)

Ausgeben

Anwenderdaten [234](#), [255](#)

Ausprägung [150](#), [238](#)

automatisches Zählerfeld siehe Zählerfeld

## B

Backup-Archiv [37](#)

BACKUP-FILES (HSMS-Anweisung) [37](#), [50](#)

Basistabelle

archivieren [129](#)

importieren [133](#)

kopieren [129](#)

rekonstruieren [129](#)

reorganisieren [227](#)

Satznummern neu vergeben [129](#)

verlagern [129](#)

Beispiel [72](#)

ALTER MEDIA DESCRIPTION [79](#)

ALTER PARTITIONING FOR TABLE [89](#)

ARCHIVE-Parameterdatei [64](#)

CHECK CONSTRAINTS [93](#)

CHECK FORMAL [89](#), [96](#)

COPY [104](#)

CREATE CATALOG [73](#), [75](#), [114](#)

CREATE MEDIA DESCRIPTION [118](#)

CREATE REPLICATION [126](#)

Delimiter-Format [255](#)

DROP MEDIA DESCRIPTION [128](#)

EXPORT TABLE [132](#)

HSMS-Parameterdatei [49](#)

IMPORT TABLE [139](#)

LOAD [165](#)

MODIFY [176](#), [177](#)

RECOVER [218](#)

REFRESH REPLICATION [222](#)

REFRESH REPLICATION FOR SPACE [222](#)

REFRESH SPACE [226](#)

REORG [232](#)

UNLOAD [255](#), [256](#)

Benutzerkennung (BS2000) [67](#), [113](#), [231](#)

bei HSMS-Sicherung ändern [52](#)

Berechtigung

ALTER CATALOG [73](#), [74](#)

ALTER MEDIA DESCRIPTION [76](#)

CHECK CONSTRAINTS [91](#)

CHECK FORMAL [94](#)

COPY [97](#)

CREATE MEDIA DESCRIPTION [115](#)

DROP MEDIA DESCRIPTION [127](#)

EXPORT TABLE [130](#)

IMPORT TABLE [134](#)

LOAD [141](#)

MIGRATE [167](#), [170](#)

MODIFY [174](#), [175](#), [177](#)

RECOVER INDEX [217](#)

REORG [229](#)

UNLOAD [235](#)

Betriebsmittel

reservieren (ARCHIVE-Lauf) [58](#)

BS2000-Benutzerkennung [67](#), [113](#), [231](#)

BS2000-Kennwort [107](#), [121](#), [124](#), [159](#), [186](#), [194](#),  
[197](#), [204](#), [212](#), [222](#), [231](#), [239](#), [240](#)

BY\_ADD\_MIRROR\_UNIT [101](#)

BY\_SRDF\_TARGET [101](#)

## C

CALL-DML-Tabelle [169](#)

CALL-DML/SQL-Tabelle [170](#)

siehe Tabellenart

CAT-LOG-Datei [76](#), [97](#), [115](#), [127](#)

CAT-REC-Datei [76](#), [97](#), [114](#), [115](#), [127](#), [179](#), [206](#),  
[214](#), [221](#)

fremde Benutzerkennung [98](#)

Offline-update [103](#)

Online-update [103](#), [173](#), [177](#)

Replikat [119](#), [122](#)

CAT-REC-Kopie [51](#)

CATALOG\_SPACE (Klausel) [108](#), [229](#)

Catalog-Space [13](#), [14](#), [114](#)

anlegen [73](#), [75](#), [105](#), [114](#)

maximale Größe [105](#)

reorganisieren [227](#)

CCSN [129](#), [133](#), [140](#), [144](#), [234](#)

CHARACTER [34](#), [243](#)

CHARACTER VARYING [34](#), [243](#)

CHECK (Pragma) 24  
 CHECK CONSTRAINTS (Utility) 91, 93  
 CHECK FORMAL (Klausel) 102  
 CHECK FORMAL (Utility) 66, 94, 96  
   Replikat 94  
 check pending (Space-Zustand) 13, 24, 25  
 CODE\_TABLE (Klausel) 108  
 Code-Tabelle siehe codierter Zeichensatz  
 Coded Character Set 73, 105, 108  
 Coded Character Set Name 129, 133, 140, 144,  
 234  
 codierter Zeichensatz 73, 105, 108  
 Compound Key 169  
 COMPRESS (ARCHIVE-Parameter) 60  
 COMPRESS-FILES (HSMS-Operand) 40  
 CONSTRAINT CHECK (Ladeparameter) 153  
 COPY (Klausel) 125, 189, 200, 207, 215, 232  
 COPY (Utility) 97, 104, 166  
 copy pending (Space-Zustand) 13  
 COPY\_FILE (Klausel) 185, 204, 211, 225  
 COPY\_NUMBER (Klausel) 185, 204, 211, 225  
 COUNTING FIELD (Klausel) 160  
 CREATE CATALOG (Utility) 73, 75, 105, 114  
 CREATE MEDIA DESCRIPTION (Utility) 115,  
 118  
 CREATE REPLICATION (Utility) 119, 126  
 CREATE-ARCHIVE (HSMS-Anweisung) 37  
 CSV\_FORMAT (Klausel) 156, 247  
 CSV-Format 28, 30

## D

DOSTOGROUP 110, 118  
 DA\_LOGS 171, 172  
   modifizieren 175, 176  
 DA-LOG-Datei 76, 97, 115, 127  
 Darstellungsmittel 9  
 DATE 32, 35, 243  
 Datei  
   Ausgabe- 65, 234, 240  
   datenbank-spezifisch 76  
   Eingabe- 65, 140, 144  
   Fehler- 65, 95, 251  
   space-spezifisch 76

Daten  
   anonymisieren 74  
 Datenbank 73, 105  
   ändern Eigenschaften 73, 73  
   anlegen 105, 114  
   fremde Benutzerkennung 98, 105  
   reparieren 89, 178, 218  
 datenbank-spezifische Datei 76, 115, 127  
 Datenbankdatei  
   fremde Benutzerkennung 52, 65, 67, 95, 98,  
   105, 107, 120, 131, 158, 231, 240, 251  
 Datendarstellung  
   abdruckbar 26  
   Standard- 33  
   Übersicht 34–35  
 Datentyp 26, 33  
 DB-Kennung 98, 105, 107, 122, 179  
 DBH-Kennung 105  
 DCAM 113  
 DDL-TA-LOG-Datei 76, 115, 127  
 DECIMAL 34, 243  
 Default Storage Group 110, 118  
 Defaultwertzeichen 243  
 DELETE AGE (Klausel) 174, 175, 177  
 DELETE ALL (Klausel) 174, 177  
 DELETE DATE (Klausel) 174, 175, 177  
 DELIMITER 157  
 DELIMITER\_FORMAT (Klausel) 155, 244  
 Delimiter-Format 27, 29, 255  
 DELIMITER-Zeichen 27, 28, 29, 30, 155, 245,  
 248  
 DESTROY (Klausel) 109  
 DEVICE (ARCHIVE-Parameter) 59  
 DEVICE REQUEST (Klausel) 78, 117  
 DIRECTORY-NAME (HSMS-Operand) 40  
 DIRSAVE (ARCHIVE-Parameter) 59  
 DOUBLE PRECISION 33, 35, 243  
 DRIVES (ARCHIVE-Parameter) 60  
 DROP MEDIA DESCRIPTION (Utility) 127, 128  
 DROP PARTITION (Utility) 86  
 DROP PARTITION ON SPACE (Klausel) 86  
 DROP STOGROUP (Klausel) 79

### E

EDT 158  
Eingabedatei 65, 140, 144  
Einrichten  
    HSMS-Archiv 37  
Entwertungszeichen (ESCAPE) 157, 248  
ENVIRONMENT (HSMS-Operand) 39  
ENVIRONMENT-ATTRIBUTES (ARCHIVE-Parameter) 60  
Erstellen  
    ARCHIVE-Parameterdatei 56  
    HSMS-Parameterdatei 43  
ESCAPE 157, 248  
EXCEPT NO LOG INDEX SPACE (Klausel) 103  
Exception-Datei siehe Fehlerdatei  
EXPORT TABLE (Utility) 129, 132  
Export-Datei 129  
    CCSN 129, 131, 133  
    fremde Benutzerkennung 131  
    Tabelle erzeugen aus 133  
Exportieren  
    Anwenderdaten 129  
    Metadaten 129

### F

Fehlerdatei 65, 66, 95, 159, 251  
    fremde Benutzerkennung 65, 95, 158, 159, 251  
Fehlerfall  
    ARCHIVE-Parameterdatei 62  
    HSMS-Parameterdatei 49  
FOR SPACE (Klausel) 125  
FOREIGN 124  
FOREIGN COPY (Klausel) 225  
formale Korrektheit siehe Formalprüfung  
Formalprüfung (Tabelle, Index) 94, 96  
Formatbeschreibung (der Daten) 33  
Freiplatzreservierung 109, 227  
fremde Benutzerkennung  
    Arbeitsdatei 231  
    Ausgabedatei 65, 240  
    CAT-REC-Datei 98  
    Datenbank 98, 105  
    Export-Datei 131

    Fehlerdatei 65, 95, 158, 159, 251  
    Replikate 120  
Fremdkopie 119  
    RECOVER SPACE 182, 203  
FROM CATREC (Klausel) 124  
FROM COPY\_FILE (Klausel) 239  
FROM FILE (Klausel) 137  
FROM SPACE (Klausel) 239  
FULL (SQL-Dialekt) 263

### G

GENERATE INDEX (Ladeparameter) 153  
GENERATE INDEX ON NO LOG INDEX SPACE (Klausel) 212, 216

### H

HSMS-Anweisung  
    BACKUP-FILES 37, 50  
    CREATE-ARCHIVE 37  
    MODIFY-ARCHIVE 42  
    MODIFY-ARCHIVE-ATTRIBUTES 42  
    MODIFY-TAPE-CONTROL 37  
    RESTORE-FILES 37, 50  
    SHOW-ARCHIVE 51  
HSMS-Archiv  
    anlegen 37  
    Archiv-Verzeichnis ändern 42  
    Attribute ändern 42  
    Benutzerkennung ändern 52  
    einrichten 37  
    Miteigentümerschaft 52  
    System wechseln 54  
HSMS-ARCHIVE-USERID (HSMS-Parameter) 47  
HSMS-Lauf steuern 36  
HSMS-Operanden  
    ALLOWED-USAGE 39  
    ARCHIVE-NAME 39  
    COMPRESS-FILES 40  
    DIRECTORY-NAME 40  
    ENVIRONMENT 39  
    LOCATION 40  
    OPERATION-CONTROL 40  
    OWNER-FIELD 39

HSMS-Operanden  
 RETENTION-PERIOD 40  
 S2-DEVICE-TYPE 40  
 TAPE-CONTROL 41  
 USER-ACCESS 39

HSMS-Parameter  
 HSMS-ARCHIVE-USERID 47  
 MAIL 48  
 PARALLEL-RUNS 48  
 REORGANIZE-SPACE 48  
 REQUEST-NAME 47  
 SAVE-ACL 47  
 SAVE-CATREC-COPY 46  
 SAVE-DIRECTORY 47  
 SAVE-FILES 45

HSMS-Parameterdatei  
 allgemeine Regeln 49  
 Beispiel 49  
 Beschreibung der Parameter 45  
 erstellen 43  
 Fehlerfälle 49  
 Syntaxregeln 48  
 Übersicht über die Parameter 44

HSMS-Sicherung  
 Benutzerkennung wechseln 52  
 Pubset wechseln 52

**I**  
 IMPORT TABLE (Utility) 133, 139  
 Importieren  
 Anwenderdaten 133  
 Basistabelle 133  
 Index 133  
 neu aufbauen 217  
 INDEX (Klausel) 95  
 INTEGER 34, 243  
 Integritätsbedingung 91, 133, 141  
 prüfen 91, 93  
 INTO FILE (Klausel) 131, 240  
 INTO TABLE (Klausel) 149  
 ISO (SQL-Dialekt) 263

**J**  
 Jobvariable 120, 179

**K**  
 Kennwort (BS2000) 107, 121, 124, 159, 186,  
 194, 197, 204, 212, 222, 231, 240  
 Kennwortkatalog 167

**L**  
 Ladedarstellung 145  
 Ladeformat 33, 140  
 Laden  
 Anwenderdaten 140, 163  
 Ladeparameter 152  
 Ladespalte 150  
 Linkname 67  
 Literal 148  
 LOAD (Utility) 66, 91, 140, 163, 234  
 load running (Space-Zustand) 13  
 LOAD\_FORMAT (Klausel) 244  
 LOAD-Formate 161  
 LOCATION (HSMS-Operand) 40  
 LOG (Klausel) 102  
 Logging 102, 143, 166, 178, 179  
 ausschalten 109  
 einschalten 102  
 Logging-Datei 179

**M**  
 MAIL (HSMS-Parameter) 48  
 MEDIA STOGROUP (Klausel) 111  
 Mediensatz 127, 128  
 erzeugen 115, 118  
 Medientabelle 76, 79, 115, 118  
 ändern 76  
 Metadaten  
 exportieren 129  
 MIGRATE (Utility) 166  
 Migration 166, 170  
 MINIMIZE (Klausel) 232  
 Mischen  
 Spaltenwerte 74  
 Mischen von Spaltenwerten 75  
 Miteigentümer 52  
 Miteigentümerschaft 36, 52, 98, 107, 120, 189,  
 190, 200, 207, 215, 216, 228, 232  
 Sicherungen mit HSMS 52

MODIFY (Utility) [171](#), [176](#)  
MODIFY-ARCHIVE (HSMS-Anweisung) [42](#)  
MODIFY-ARCHIVE-ATTRIBUTES (HSMS-Anweisung) [42](#)  
MODIFY-TAPE-CONTROL (HSMS-Anweisung) [37](#)  
multiple Spalte [150](#), [238](#)

### N

Namenskonflikte (bei der Umstellung) [169](#)  
NATIONAL CHARACTER [34](#), [243](#)  
NATIONAL CHARACTER VARYING [34](#), [243](#)  
NEWLINE-Zeichen [28](#)  
nicht-signifikanter Wert [243](#)  
NO CHECK FORMAL (Klausel) [102](#)  
NO CONSTRAINT CHECK (Ladeparameter) [91](#), [153](#)  
NO CONSTRAINTS (Klausel) [139](#)  
NO DATA (Klausel) [132](#), [138](#)  
NO DESTROY (Klausel) [109](#)  
NO DEVICE REQUEST (Klausel) [78](#), [117](#)  
NO INDEX (Klausel) [138](#), [168](#), [186](#), [190](#), [194](#), [197](#), [201](#)  
NO INDEX (Ladeparameter) [152](#)  
NO LOG (Klausel) [109](#)  
NO OVERWRITE (Ladeparameter) [152](#)  
NO SHARE (Klausel) [78](#), [109](#), [117](#)  
NULL-Wert [148](#), [164](#), [243](#)  
NUMERIC [34](#), [243](#)  
Nur-CALL-DML-Tabelle [155](#), [156](#), [170](#)  
siehe Tabellenart

### O

OFFLINE (Klausel) [99](#), [237](#)  
Offline-Sicherung [97](#)  
OLD (SQL-Dialekt) [263](#)  
OLDEST [155](#), [156](#)  
ON SPACE (Klausel) [92](#), [217](#)  
ON TABLE (Klausel) [92](#), [217](#)  
ON USER\_ID (Klausel) [108](#), [122](#), [124](#), [189](#), [199](#), [206](#), [214](#), [222](#), [226](#)  
ONLINE (Klausel) [99](#), [237](#)  
Online-Sicherung [97](#)  
OPERATION-CONTROL (HSMS-Operand) [40](#)

OPERATOR (ARCHIVE-Parameter) [59](#)  
OVERWRITE (Ladeparameter) [152](#)  
OWNER-FIELD (HSMS-Operand) [39](#)

### P

PARALLEL-RUNS (HSMS-Parameter) [48](#)  
Partition [22](#)  
Verfügbarkeit [22](#)  
partitionierte Tabelle [22](#), [91](#), [133](#), [135](#), [138](#), [139](#), [149](#), [153](#), [159](#), [160](#), [170](#), [217](#), [237](#), [251](#)  
PASSWORD (Klausel) [107](#), [121](#), [124](#), [131](#), [137](#), [159](#), [168](#), [186](#), [194](#), [197](#), [204](#), [212](#), [222](#), [226](#), [231](#), [239](#), [240](#)  
PASSWORD\_CATALOG (Klausel) [167](#)  
Passwort siehe Kennwort (BS2000)  
PBI-Datei [76](#), [115](#), [127](#)  
PCTFREE (Klausel) [109](#), [227](#)  
POSITION [146](#), [242](#), [244](#)  
Pragma  
CHECK [24](#)  
Precompiler-Optionen  
SOURCE-PROPERTIES [263](#)  
Primärschlüssel [133](#)  
mehrspalzig [136](#)  
PRIMARY (Klausel) [77](#), [108](#), [116](#)  
Primärzuweisung  
festlegen [77](#), [116](#)  
Prüfen  
formale Korrektheit (Tabelle, Index) [94](#), [96](#)  
Integritätsbedingung [91](#), [93](#)  
PUBLIC (Klausel) [112](#), [123](#)  
Pubset, Wechsel bei HSMS-Sicherung [52](#)

### Q

QUOTE-Zeichen [28](#)

**R**

REAL [33](#), [35](#), [243](#)  
 Rechnername [113](#)  
 RECOVER (Utility) [89](#), [178](#), [218](#)  
 RECOVER CATALOG (Utility) [208](#)  
     mit Replikaten [213](#)  
     mit Sicherungsbeständen [208](#)  
 RECOVER CATALOG\_SPACE (Utility) [202](#)  
     mit Sicherungsbeständen [202](#), [205](#)  
 RECOVER INDEX (Utility) [217](#)  
 recover pending (Space-Zustand) [13](#)  
 RECOVER SPACE (Utility) [181](#), [195](#), [198](#)  
     mit Replikaten [187](#)  
     mit Sicherungsbeständen [181](#)  
 RECOVER SPACESET (Utility) [192](#)  
 RECOVERY\_UNITS [171](#), [172](#)  
     Einträge löschen [103](#)  
     modifizieren [174](#), [175](#), [176](#)  
 REFRESH REPLICATION (Utility) [219](#), [222](#)  
 REFRESH SPACE (Utility) [223](#), [226](#)  
 RENAME (Klausel) [125](#), [190](#), [200](#), [207](#), [216](#), [232](#)  
 REORG (Utility) [227](#), [232](#)  
 reorg pending (Space-Zustand) [14](#)  
 Reorganisieren  
     Anwender-Space [227](#)  
     Basistabelle [227](#)  
     Datenbank [227](#)  
 REORGANIZE-SPACE (HSMS-Parameter) [48](#)  
 Reparieren  
     Anwender-Space [178](#)  
     Catalog-Space [178](#)  
     Datenbank [89](#), [178](#), [218](#)  
     Space-Liste [178](#)  
     Space-Set [178](#)  
 Replikat [189](#), [199](#), [206](#), [214](#)  
     aktualisieren [219](#)  
     anlegen [119](#)  
     CHECK FORMAL [94](#)  
     erweitern [223](#)  
     fremde Benutzererkennung [120](#)  
 REQUEST-NAME (HSMS-Parameter) [47](#)  
 Reservieren  
     Betriebsmittel (ARCHIVE-Lauf) [58](#)  
 RESTORE-FILES (HSMS-Anweisung) [37](#), [50](#)

RETENTION-PERIOD (HSMS-Operand) [40](#)  
 RETPD (ARCHIVE-Parameter) [60](#)  
 Rücksetzen  
     siehe Zurücksetzen

**S**

S2-DEVICE-TYPE (HSMS-Operand) [40](#)  
 SAVE-ACL (ARCHIVE-Parameter) [59](#)  
 SAVE-ACL (HSMS-Parameter) [47](#)  
 SAVE-CATREC-COPY (HSMS-Parameter) [46](#)  
 SAVE-DIRECTORY (HSMS-Parameter) [47](#)  
 SAVE-FILE (HSMS-Parameter) [45](#)  
 Schlüsselwort  
     SQL [263](#)  
 SCOPE ALL (Klausel) [218](#)  
 SCOPE PENDING (Klausel) [186](#), [190](#), [194](#), [197](#),  
     [200](#), [212](#), [218](#)  
 SECONDARY (Klausel) [77](#), [109](#), [117](#)  
 SECOS [52](#)  
 SECURE-RESOURCE-ALLOCATION [58](#)  
 sedezimale Verschlüsselung  
     der SQL-Datentypen [34](#)  
 Sekundärzuweisung  
     festlegen [77](#), [117](#)  
 SESAM-Sicherungsbestand [185](#), [204](#), [211](#), [225](#),  
     [239](#)  
     BS2000-Dateiname [103](#)  
     erstellen [97](#), [104](#), [126](#)  
     verwalten [103](#)  
 SESAM/SQL-Server [7](#)  
 SESAM/SQL-Server-Tabelle [166](#)  
 SHARE (Klausel) [78](#), [109](#), [117](#)  
 SHOW-ARCHIVE (HSMS-Anweisung) [51](#)  
 Sicherungsbestand [211](#)  
 SKIP FIRST RECORDS (Klausel) [149](#)  
 SMALLINT [34](#), [243](#)  
 Sommerzeit [97](#)  
 SOURCE-PROPERTIES (Precomp.-  
     Option) [263](#)  
 Space  
     Zustand [13](#)  
 SPACE (ARCHIVE-Parameter) [60](#)  
 SPACE (Klausel) [95](#), [229](#)  
 space defect (Space-Zustand) [13](#), [14](#)

space o.k. (Space-Zustand) **13**  
Space-Liste **179**  
Space-Parameter (Klausel) **77, 108, 116**  
Space-Set **178, 191**  
    reparieren **178**  
space-spezifische Datei **76**  
Space-Zustände **13**  
    (Übersicht) **15–20**  
Spalte  
    multipel **150, 238**  
Spaltenwerte  
    mischen **74**  
Sperrung **13**  
Spezialname **263**  
SQL-Datenbankverzeichnis **105**  
SQL-Datentyp  
    sedezimale Verschlüsselung **34**  
SQL-Dialekt **263**  
SQL-Schlüsselwort **263**  
SQL-Tabelle **169, 170**  
Standarddarstellung (der Daten) **33**  
Steuern  
    ARCHIVE-Lauf **56**  
    HSMS-Lauf **36**  
STOGROUP (Klausel) **110, 118**  
Storage Group **110, 118**  
STREAM (ARCHIVE-Parameter) **59**  
Suchbedingung **132**  
symbolischer Rechnername **113**  
Syntaxelemente (Übersicht) **257–262**  
Syntaxregeln  
    ARCHIVE-Parameterdatei **61**  
    HSMS-Parameterdatei **48**  
System-Benutzerkennung **112**

## T

Tabelle **13, 217**  
    CALL-DML/SQL- **170**  
    exportieren **129**  
    importieren **133**  
    Nur-CALL-DML- **170**  
    Partitionen modifizieren **129**  
    partitioniert **22, 91, 133, 135, 138, 139, 149, 153, 159, 160, 170, 217, 237, 251**  
    SQL- **170**  
    Zustand check pending **13**  
TABLE (Klausel) **94**  
TAPE-CONTROL (HSMS-Operand) **41**  
Teilreplikat **119, 125, 219**  
TIME(3) **32, 35, 243**  
TIMESTAMP(3) **32, 35, 243**  
TO ... TABLE (Klausel) **169**  
TRANSFER\_FORMAT (Klausel) **154, 244**  
Transferdatei **158, 244**  
Trennzeichen (DELIMITER-Zeichen) **157**

## U

Umstellen  
    CALL-DML/SQL-Tabelle **170**  
    Datenbank **166**  
    Nur-CALL-DML-Tabelle **170**  
universeller Benutzer **105, 110, 112**  
UNLOAD (Utility) **66, 140, 234, 255**  
UNLOAD\_FORMAT (Klausel) **154**  
UNLOAD-Formate **253**  
UNRESTRICTED (Klausel) **175, 176**  
USER (Klausel) **112**  
USER-ACCESS (HSMS-Operand) **39**  
USING DIRECTORY (Klausel) **100**  
USING FILE (Klausel) **158, 230**  
USING MEDIA (Klausel) **123**  
USING PARTITION BY RANGE (Klausel) **135**  
USING SPACE (Klausel) **135, 169**  
USING STOGROUP (Klausel) **99**  
Utility-Anweisung **7, 11, 69, 70**  
Utility-Funktionen (Übersicht) **12**  
Utility-Monitor **219, 223**  
UTM-Anwendungsname **113**  
UTM-Benutzer **113**

**V**

Verfügbarkeit

logisch 22

Partition 22

physikalisch 22

VOLUMES ... ON (Klausel) 112, 123

Vorgängerversion 166

**W**

Wert

nicht-signifikant 243

WHEN ... THEN NULL (Klausel) 148, 243

WHEN NULL THEN (Klausel) 243

WHERE, Suchbedingung 132

Winterzeit 97

WITH CATREC (Klausel) 206, 214

WITH CONSTRAINTS (Klausel) 138

WITH INDEX (Klausel) 138, 168

**X**

XHCS 73, 108

**Z**

Zählfeld 160

Zeichen

DELIMITER 157

ESCAPE 157, 248

Zeichensatz

codiert 73, 105, 108

Zeitstempel 185, 188, 193, 197, 204, 211, 225

Zuladen siehe Laden

Zurücksetzen

Anwender-Space 178

Catalog-Space 178

Datenbank 178

Space-Liste 178

Space-Set 178

Zustand

siehe Space-Zustände

