Deutsch

FUJITSU Software BS2000

CIS V12.0 Manual 3

Kommandos

Benutzerhandbuch

April 2020

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 senden.

Zertifizierte Dokumentation nach DIN EN ISO 9001:2015

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 9001:2015 erfüllt.

Copyright und Handelsmarken

Inhaltsverzeichnis

1 Einleitung

1.1 Allgemeines

Die Beschreibung der CIS-Kommandos gibt dem CIS-Anwender detaillierte Informationen über ihre Anwendungsmöglichkeiten und Funktionen.

Zur Anwendung der CIS-Kommandos sind mindestens zwei Dateien notwendig:

Hauptdatei (HD)
Datenbeschreibungsdatei (DABEL)

In der Hauptdatei stehen die Daten (Informationen) und in der Datenbeschreibungsdatei die Datenbeschreibungen, die Format und Struktur der Daten festlegen und gleichzeitig Paßwortfunktionen besitzen (vgl. Manual-1: Dateien in der CIS-Datenbank).

Bei größeren Datenmengen ist der Aufbau einer Verweisdatei (VD) sinnvoll, um einen schnellen Zugriff auf die Daten zu ermöglichen. Es kann mit mehreren Haupt- und Verweisdateien gearbeitet werden (Multi-File-Betrieb).

Zu einer Datenbank können beliebig viele Datenbeschreibungen generiert werden, so daß mehrere Anwender mit verschiedenen Paßworten und unterschiedlichen Zugriffsmöglichkeiten auf diese eine Datenbank zugreifen können. Der Datenschutz ist bis zur kleinsten Informationseinheit, dem Feld, möglich, da der Anwender nur auf die Felder zugreifen kann, die in der für ihn freigegebenen Datenbeschreibung enthalten sind. Außerdem wird über das Paßwort die Anwendung bestimmter Kommandos erlaubt oder unterbunden.

Die CIS-Kommandosprache kann aktiv (über Terminal) oder passiv (über ein Anwenderprogramm) angewendet werden.

In diesem Handbuch wird bei jedem Kommando auf die zu verwendende Datenbeschreibung (Satz-, Transformations-, Bildbeschreibung oder Bildschirmmaske) verwiesen. Die Wirksamkeit einer angegebenen Datenbeschreibung (= Paßwort) wird erst durch die Angabe eines anderen Paßwortes aufgehoben.

Die Kommandos werden von einem Übersetzungsmodul auf ihre syntaktische und inhaltliche Richtigkeit geprüft. Sie können in Abhängigkeit der über die Datenbeschreibungsdatei zugewiesenen Sprache in Deutsch oder Englisch angegeben werden.

1.2 Zeigertechnik (Satzzeiger)

Diejenigen Kommandos, die sich direkt oder indirekt auf Sätze beziehen, wie ZEIGE, DRUCKE BLAETTERN oder AENDERN gehen von einer Lokalisierung des Satzes aus, auf den sich der entsprechende Befehl bezieht.

Der Satzzeiger lokalisiert den Satz entsprechend dem vorausgegangenen Kommando. Beim Satzzeiger kann es sich um einen

Zeiger im File Control Block der Hauptdatei	FCB-Zeiger,
Verweisdateizeiger	VD-Zeiger oder
Zielpunktlistenzeiger	ZPL-Zeiger

handeln.

Bemerkung:

Ein FCB-Zeiger ist im Gegensatz zu den transaktionsbezogenen VD- /ZPL- Zeigern taskbezogen. Folglich sollte bei Multitasking oder CISDBH-Betrieb der FCB-Zeiger nicht verwendet werden.

Entstehung der Zeiger

Der FCB-Zeiger entsteht zuerst. Nach jedem DVS-mäßigen Öffnen einer Hauptdatei zeigt er auf den ersten Satz der Hauptdatei.

Der VD-Zeiger entsteht bei einem GET-KI oder BLAETTERN-I Kommando.

Der ZPL-Zeiger entsteht mit einem SUCHE-, SORTIERE- oder AKTIVIERE-Kommando.

Explizite Positionierung der Zeiger

Die Zeiger können mit dem BLAETTERN-Kommando vor oder zurück positioniert werden.

Implizite Positionierung der Zeiger

Die Zeiger zeigen nach allen CIS-Kommandos, die Sätze verarbeiten, auf den zuletzt angesprochenen Satz.

Priorität der Zeiger

Der ZPL-Zeiger hat die höchste Priorität von den drei Zeigerarten, der FCB-Zeiger die geringste.

Löschen der Zeiger

Der FCB-Zeiger wird mit einem CLOSE-Kommando gelöscht.
Der VD-Zeiger wird mit einem CLOSE- oder IGNORIERE-I Kommando gelöscht.
Der ZPL-Zeiger wird mit einem CLOSE- oder IGNORIERE-Z Kommando gelöscht.

1.3 Anwendungsgebiete

1.3.1 Zielpunktlistenbehandlung

Zielpunktlisten werden in der Regel durch die Verwendung von Suchfragen (vgl. Seite 146) erstellt. Die Zielpunktliste enthält in aufsteigender Reihenfolge die Ordnungsbegriffe bzw. Adressen aller Sätze, die den Suchbedingungen genügen.

Die Reihenfolge der Ordnungsbegriffe bzw. Adressen in der Zielpunktliste kann mit dem SORT-Kommando geändert werden. Mit einer sortierten Zielpunktliste können keine Folgeverknüpfungen (UND, ODER, NICHT) ausgeführt werden.

Wird eine Zielpunktliste mehrmals zu verschiedenen Zeitpunkten benötigt, kann diese jederzeit für eine spätere Verwendung gesichert werden (vgl. SICHERN-E Seite 129 und SICHERN-V Seite 134). Je nach Operationsergänzung erfolgt die Sicherung in eine temporäre Datei oder in die Verweisdatei.

Dabei gelten folgende Unterschiede:

Temporäre Sicherung	Sicherung in der Verweisdatei
Die Sicherung ist nur temporär. D.h. die Zielpunktliste bleibt max. so lange gesichert, bis der CIS-Lauf beendet wird.	Die Sicherung ist permanent. D.h. die Zielpunktliste bleibt auch dann in der Verweisdatei bestehen, wenn der CIS-Lauf beendet ist.
Die gesicherte Zielpunktliste wird mit dem AKTIVIEREN-Kommando aktiviert. (vgl. Seite 40)	Die gesicherte Zielpunktliste wird mit dem SUCHE-Kommando aktiviert. (vgl. Seite 146)
Die gesicherte Zielpunktliste kann nur mit dem AKTIVIEREN-Kommando zur aktuellen Zielpunkt-liste werden. Diese aktuelle Liste kann, wenn die Ordnungsbegriffe aufsteigend sortiert sind, mit weiteren Folgefragen (UND, ODER, NICHT) verknüpft werden.	Die gesicherte Zielpunktliste kann mit dem SUCHE-Kommando und/oder mit den Folgefragen (UND, ODER, NICHT) verknüpft werden.
Die zu sichernde Zielpunktliste kann beliebig sortiert sein.	Die zu sichernde Zielpunktliste muß aufsteigend nach dem Ordnungsbegriff sortiert sein.
Der Satzzeigerstand wird mitgesichert.	Der Satzzeigerstand wird nicht mitgesichert.
Die Sicherung kann unter einer beliebigen Feldbezeichnung der aktuellen Datenbeschreibung erfolgen.	Zur Sicherung ist eine Pseudofeldbezeichnung notwendig, die in der Datenbeschreibung ange-geben sein muß.

Mit dem FREIGEBEN-Kommando (vgl. Seite 74) werden gesicherte Zielpunktlisten freigegeben. Bei Ende des CIS-Laufs (HALT-Kommando oder CLOSE,A-Aufruf) werden alle temporär gesicherten Zielpunktlisten von CIS automatisch gelöscht, die permanent in der Verweisdatei gesicherten Zielpunktlisten bleiben erhalten.

Die temporär gesicherten Zielpunktlisten können mit dem ZEIGE-E Kommando (vgl. Seite 205) abgefragt werden.

1.3.2 Datenausgabe

Die Ausgabekommandos ermöglichen es, Daten auf folgenden Medien auszugeben:

Terminal	ZEIGE-Kommando	vgl. Seite 205
Drucker	DRUCKE-Kommando	vgl. Seite 49
Datei	WRITE-Kommando	vgl. Seite 196

Für jede Ausgabeform ist die Anzahl der auszugebenden Einheiten über die Mengenangabe bestimmbar.

Auf Terminal oder Drucker können Daten der Hauptdatei, der Verweisdatei und der Datenbeschreibung ausgegeben werden. Die Ausgabeform der Daten ist entweder über das Kommando oder eine Bildbeschreibung steuerbar.

Bei der Dateiausgabe werden ausgewählte Daten zur getrennten Weiterverarbeitung bereitgestellt. Die erzeugten Dateien sind SAM- oder ISAM-Dateien (entsprechend SET-FILE-LINK Kommando).

1.3.3 Update-Kommandos

Die Update-Kommandos bewirken folgende Datenveränderungen in Haupt- und Verweisdatei:

PUT-Kommando	vgl. Seite 109	Aufnahme neuer Informationseinheiten (Satz, Abschnitt, Feld).
AENDERN-Kommando	vgl. Seite 35	Ändern bereits vorhandener Informations-einheiten.
LOESCHEN-Kommando	vgl. Seite 101	Löschen bereits vorhandener Informations-einheiten.

Die Kommandos sind jeweils auf Satz-, Abschnitts- oder Feldebene anwendbar. Ist beim Ändern oder Löschen von Wiederholabschnitten bzw. von Wiederholfeldern keine Multiplität explizit angegeben, so wird jeweils die erste Multiplität angesprochen.

Sind invertierte Felder vorhanden, wird die Verweisdatei durch die Update-Kommandos ebenfalls aktualisiert. Die Prüfung, ob die Felder invertiert sind, erfolgt über die (beim aktuellen Kommando) verwendete Datenbeschreibung. Es ist deshalb darauf zu achten, daß die verwendete Datenbeschreibung alle invertierten Felder enthält, für die ein Update erfolgen soll.

Eine Besonderheit stellen die Bildschirmkommandos dar. Damit können im Dialog über vorgegebene Bildschirmmasken folgende Datenveränderungen durchgeführt werden:

EINGEBEN-Kommando	vgl. Seite 68	Eingeben neuer Feldinhalte.
EINGEBEN-K	vgl. Seite 70	Satzweises Eingeben neuer Feldinhalte.
EINGEBEN-A	vgl. Seite 69	Anfügen neuer Abschnitte an bereits vorhandene Sätze mit MV-Format.
KORRIGIEREN-Kommando	vgl. Seite 100	Korrektur von Feldinhalten.

1.3.4 Mehrfachzugriffe

Mehrfachzugriffe auf den gleichen Satz sind zu vermeiden, wenn ein performanter Ablauf erreicht werden soll.

Mehrfachzugriffe bewirken einen erhöhten Aufwand an Lese-, Schreib- und/oder Update-Routinen. Beispielsweise wird bei jedem PUT,F nicht nur das Feld bzw. der Satz, sondern ein ganzer Block zurückgeschrieben. Bei entsprechender Blockung ist dieser zu schreibende Bereich sehr groß. Ist das Feld zusätzlich invertiert, so muß in der Verweisdatei gelesen und dann verändert werden.

Bei der Programmierung ist also darauf zu achten, daß viele Einzelzugriffe auf den gleichen Hauptdateisatz durch einen einzigen ersetzt werden.

Mehrere Einzel-zugriffe mit:	auf	zu ersetzen durch einen Zugriff mit:
GET,F	denselben Abschnitt (MV-Format) mehrere Abschnitte desselben Satzes (MV-Format) denselben Satz (V-Format)	GET,A GET,KP GET,KP
GET,A	denselben Satz (MV-Format)	GET,KP
PUT,F	denselben Abschnitt (MV-Format) mehrere Abschnitte desselben Satzes (MV-Format) denselben Satz (V-Format)	AE,A AE,K AE,K
PUT,A	denselben Satz (MV-Format)	AE,K
AE,F	denselben Abschnitt (MV-Format) mehrere Abschnitte desselben Satzes (MV-Format)	AE,A AE,K
AE,A	denselben Satz (MV-Format)	AE,K
L,F	denselben Abschnitt (MV-Format) mehrere Abschnitte desselben Satzes (MV-Format) denselben Satz (V-Format)	AE,A AE,K AE,K
L,A	denselben Satz (MV-Format)	AE,K

1.3.5 Transaktionskommandos

Diese Kommandos können nur bzw. müssen benützt werden, wenn mit

- CISKOOR (inlinked oder independent) und
- Transaktionsverarbeitung mit Before-Image-Sicherung

gearbeitet wird.

Zu den Transaktionskommandos gehören folgende Befehle:

- TRANSAKTION-Kommando vgl. Seite 171
- Verteilte Transaktionsverarbeitung vgl. Seite 177
- SPERRE-Kommando vgl. Seite 141
- STATUS-Kommando vgl. Seite 144

Die Transaktion ist die Menge der Kommandos, die für den Anwender eine logische Einheit darstellen.

Das TRANSAKTION-Kommando definiert den Anfang und das Ende einer Transaktion. Die Transaktion kann entweder normal beendet oder rückgesetzt werden.

Bei normalem Transaktionsende werden alle seit Transaktionsanfang durchgeführten Änderungen permanent.

Beim Rücksetzen einer Transaktion wird der Zustand der geänderten Datenbank(en) vor dem Zeitpunkt des Transaktionsanfangs wieder hergestellt.

Innerhalb einer Transaktion können bestehende Sätzen nur über gesperrte Sätze, Zielpunktlisten oder Dateien geändert werden, um zu verhindern, daß zur selben Zeit von einem weiteren Anwender ebenfalls Änderungen an denselben Sätzen vorgenommen werden. Die Datensicherheit ist dadurch gewährleistet. Nach Transaktionsende bzw. nach Rücksetzen einer Transaktion werden alle seit Transaktionsanfang gesperrten Zielpunktlisten bzw. gesperrten Dateien wieder freigegeben.

Ein SPERRE-Kommando vor dem Lesezugriff verhindert das Lesen von Sätzen, die gerade geändert werden.

Das STATUS-Kommando bietet die Möglichkeit, Auskünfte über gesperrte Dateien bzw. gesperrte Sätze einzuholen.

Beispiel einer Transaktion:

```
TR,A
SUCHE MITARBEITER = ...,DB.PERSO1
IM00    ANZAHL ZIELINFORMATIONEN: 1
SPERRE,Z
LOESCHE,1,K
SUCHE ERWSATZ = ...,DB.PERSO2
IM00    ANZAHL ZIELINFORMATIONEN: 5
SPERRE,Z
      Falls CM = "SP04" dann: TR,R
LOESCHE,$,K
TR,E
```

Im Beispiel wird in der Datenbank DB.PERS01 nach einem bestimmten Mitarbeitersatz gesucht, der anschließend gelöscht wird. Zu diesem Satz gibt es in der Datenbank DB.PERS02 Erweiterungssätze, die ebenfalls gelöscht werden sollen.

Nach dem Suchen in der Datenbank DB.PERS02 wird beim Sperren festgestellt, daß bereits auf einen Satz oder mehrere Sätze von einer anderen Transaktion zugegriffen wurde (SP04). Die Erweiterungssätze können nicht gelöscht werden, also ist auch die Löschung des Mitarbeitersatzes rückgängig zu machen (TR,R). Wird der zweite SPERRE,Z ohne Fehlermeldung ausgeführt, so werden die 5 Erweiterungssätze gelöscht.

Die Kommandos der verteilten Transaktionsverarbeitung werden im Zusammenhang mit dem 2-Phasen-Commit-Protokoll verwendet. Die Beschreibung dieses Protokolls sollte über die allgemeine Informatikliteratur beschafft werden. Eine kurze Abhandlung befindet sich im Manual 4.

Die verteilte Transaktionsverarbeitung von UTM ist in CIS (Modul CISCON, Programme CISDBH und CISKOOR) voll integriert.

1.4 Syntaxdarstellung im Handbuch

Abkürzungsmöglichkeiten in CIS-Kommandos:

Der Operationsteil des CIS-Kommandos kann bis auf Eindeutigkeit in Abhängigkeit aller CIS-Kommandos abgekürzt werden. Die kürzeste mögliche Abkürzung ist im Handbuch durch Fettdruck hervorgehoben.

Beispiel:

RECHNE kann wie folgt geschrieben werden:

```
RECHNE
RECHN
RECH
REC
RE
```

Bemerkung:

In den Beispielen dieses Handbuches wurde im allgemeinen die kürzeste Darstellungsform gewählt.

Feldbezeichnungen einer Datenbeschreibung können in Abhängigkeit der auftretenden Feldbezeichnungen bis auf Eindeutigkeit abgekürzt werden.

Metasprache für die Formatdarstellung der Kommandos

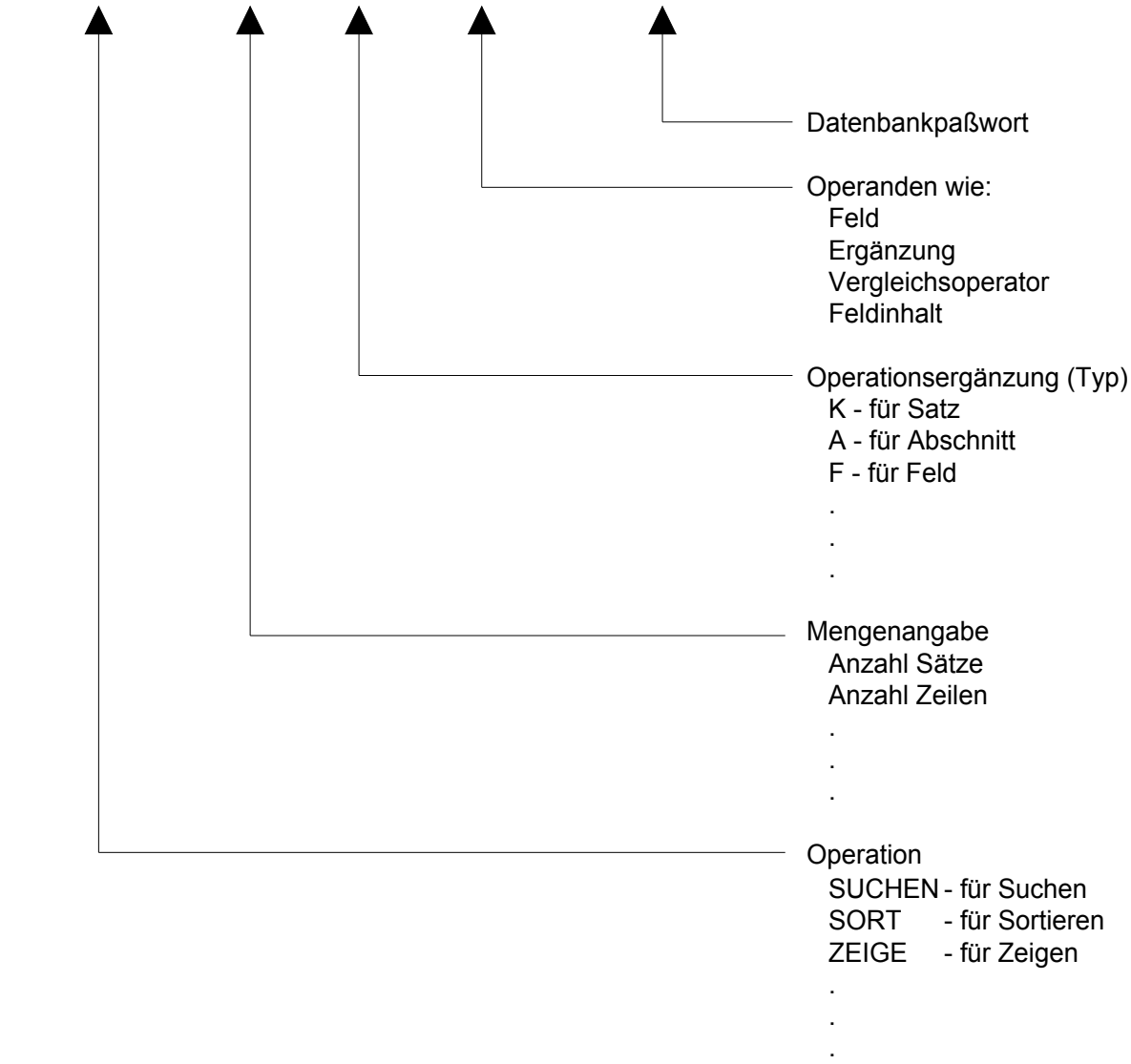
`[,{U/I},]`	Die Unterstreichung eines Wertes bezeichnet einen Standardwert der von CIS eingesetzt wird.
`[ ]`	Eckige Klammern schließen Wahlangaben ein, die weggelassen werden können.
`{UND/ODER/NICHT}`	Geschweifte Klammern schließen Alternativen ein. Aus den Angaben muß eine ausgewählt werden. Die Alternativen stehen nebeneinander und werden durch einen Schrägstrich / getrennt. Ist eine Alternative ein Standardwert (unterstrichen), dann ist keine Angabe notwenig, sofern der Standardwert gewünscht wird.
`{HD/VD}`	Der Schrägstrich trennt Alternativen.
`_`	Space
`]...`	Drei Punkte, auf einen Klammerausdruck folgend, bedeuten, daß die vorstehende Einheit (in eckigen Klammern eingeschlossen) mehrmals wiederholt werden kann.
`Operand...`	Drei Punkte auf einen Operanden folgend bedeuten, daß alle Angaben, die auf den vorhergehenden gleichen Operanden folgen, anzugeben sind.
`Operation...`	Drei Punkte auf eine Operation (mit Mengenangaben und Operationsergänzungen) folgend bedeuten, daß die Operation gekettet werden kann, d.h. daß die Operation mit den dazugehörenden Operandenangaben mehrmals wiederholt werden kann.
`,...` oder `_...`	Drei Punkte auf ein Komma bzw. ein Space folgend bedeuten, daß alle vor dem Komma bzw. Space stehenden Operanden mehrmals wiederholt werden können.

`NAME_-_ORT`	Feldangaben VON - BIS
	
	Bei Kommandos, in denen die Feldbezeichnungen mit Space als Trenner hintereinander angegeben werden, kann die Feldfolge durch eine erste Feldbezeichnung, Space, Bindestrich, Space und eine letzte Feldbezeichnung beschrieben werden.
	
	Beispiel:
	
	`ZEIGE,1,T,NAME_VORNAME_STRASSE_PLZ_ORT`
	
	wird ersetzt durch:
	
	`ZEIGE,1,T,NAME_-_ORT`
	
	Für die Bearbeitung ist die Reihenfolge der Felder in der Satzbeschreibung maßgebend. Wird mit Ergänzungen gearbeitet, so gelten die für das erste Feld gemachten Angaben für alle weiteren Felder.

1.5 Kommandostruktur

Ein CIS-Kommando besteht immer aus einer Operation und kann je nach Kommandotyp um
Mengenangabe,
Operationsergänzung,
Operandenangaben und
Datenbankpaßwort erweitert werden.

```
Operation[,Menge],Typ,[Operand][,DB.Paßwort]
```



Die Elemente werden in dieser Reihenfolge, durch ein Komma getrennt, angegeben. Stehen jedoch Operation und Operand unmittelbar hintereinander, so werden sie durch Space getrennt.

In jedem Kommando muß zuerst die Operation angegeben sein. Bei Operationen, in denen der Operand bereits durch die Operation hinreichend bestimmt ist, kann die Operationsergänzung weggelassen werden.

Die Angabe des Datenbankpaßwortes ist im ersten Kommando nach dem Laden von CIS notwendig, sowie in Fällen, in denen eine andere Datenbeschreibung benützt werden soll, als im vorhergehenden Kommando.

1.5.1 Operationen

Operationen sind z.B.

SUCHEN (= Suchen von Sätzen, die bestimmte logische Bedingungen erfüllen.)
ZEIGE (= Ausgabe von Informationen auf Bildschirm.)
GET (= Bereitstellen von Daten aus einer Hauptdatei.)
PUT (= Einfügen von Daten in eine Hauptdatei.)
.
.

1.5.2 Operationsergänzungen

Eine Operationsergänzung enthält stets eine nähere Beschreibung der Operation.

Beispiel:

`GET,KP,DB.SATZB1` die Operationsergänzung KP bedeutet die Bereitstellung eines Satzes.

`GET,F,PNR,DB.SATZB1` die Operationsergänzung F bedeutet die Bereitstellung eines Feldinhalts.

1.5.3 Operanden

Man kann zwei wesentliche Grundtypen von Operanden unterscheiden.

1. Tritt in einem Operanden ein Feldwert auf, so hat der Operand die folgende Form:

   Feldbezeichnung[(Ergänzung)] Vergleichsoperator Feldwert

   Beispiel 1: `PNR=4711`
   Beispiel 2: `ARZT(AM=2)=MEIER`

   Die Ergänzung muß dabei, wie das Beispiel 1 zeigt, nicht unbedingt angegeben sein.

   Auf Feldergänzungen wird im Abschnitt Operandenergänzungen (vgl. Seite 18) näher eingegangen. Die Beispiele 1 und 2 können z. Bsp. in den Kommandos:

   ```
   SUCHEN PNR>45   und
   PUT,F,ARZT(AM=2)=MEIER,DB.SATZBE
   ```

   stehen.

2. Im Operanden treten bei manchen Kommandos nur Feldbezeichnungen eventuell mit Feldergänzungen auf. Die Feldbezeichnungen werden stets durch ein oder mehrere Spaces voneinander getrennt.

   Der Operand hat also die folgende Struktur:

   ```
   Feldbezeichnung[(Ergänzung)]_...
   ```

   ```
   PNR
   PNR NAME ARZT
   PNR ARZT(AM=1) THERAPIE(AM=2)
   ```

1.5.4 Operandenergänzungen

Feldbezeichnungen und Abschnittsnamen, die in einem Operanden auftreten, können normalerweise durch die Angabe zusätzlicher Parameter ergänzt werden, um von der Definition bzw. von Standardwerten abzuweichen. Diese Ergänzungen werden durch Kommas voneinander getrennt. Der gesamte Ausdruck wird in runde Klammern eingeschlossen und unmittelbar hinter den Feld- bzw. den Abschnittsnamen geschrieben, auf den er sich bezieht.

AM={-n/n}	Legt die Multiplität n des Wiederholabschnitts fest, der angesprochen werden soll bzw. in dem die Felder liegen, die angesprochen werden sollen. Ist dieser Parameter nicht angegeben, so wird AM=1 angenommen. Vorzeichenlos bedeutet, daß die Reihenfolge der wiederholten Abschnitte von Beginn des Satzes in Richtung auf das Satzende betrachtet wird. Negatives Vorzeichen bedeutet die umgekehrte Richtung.
FM={-n/n}	Legt die Multiplität des Wiederholfeldes innerhalb eines Abschnitts oder Satzes fest. Für den FM-Parameter gelten analog die für den AM-Parameter gemachten Aussagen.
Bemerkung:	AM=$ und FM=$ sind nur in Kurzkommandoketten erlaubt (vgl. Seite 257).
AZ={n/$}	Der Parameter legt die Abschnittszyklen fest d.h. er gibt an, wieviele Abschnitte der gleichen Art zu berücksichtigen sind. $ bedeutet alle Abschnitte. Dabei erfolgt die Abarbeitung der Abschnitte immer in Richtung auf das Ende des Satzes. Sind z.B. die Parameter AM=4 und AZ=3 angegeben, so werden die Abschnitte mit den Multiplitäten 4, 5 und 6 berücksichtigt. Ist in der Ergänzung der AZ-Parameter nicht angegeben, so wird AZ=1 angenommen.
FZ={n/$}	Legt die Feldzyklen fest, d.h. wieviele Wiederholfelder zu berücksichtigen sind. $ bedeutet alle Felder. Dabei erfolgt die Abarbeitung stets in Richtung auf das Ende des Abschnitts bzw. des Satzes.
FL=n	Dieser Parameter erlaubt, die in der Satzbeschreibung festgelegte Feldlänge bei Ausgaben und Sortiervorgängen auf eine Länge von n Byte zu modifizieren.

FP={-n/n/$} Der FP-Parameter wird bei der Suche nach Wortteilen benötigt. Bei der Angabe FP=$ kann die gesuchte Zeichenreihe bei einem beliebigen Zeichen innerhalb des Feldwertes beginnen. FP=n bedeutet, daß die gesuchte Zeichenreihe mit dem n-ten Zeichen des Feldwertes von links beginnt, ein negatives Vorzeichen (FP=-n), daß sie mit dem n-ten Zeichen von rechts endet. Die Zeichenreihe selbst steht immer linksbündig.

DB=xxxxxx Der DB-Parameter ist als Feldergänzung beim Arbeiten in einem Fenster notwendig. Dabei steht xxxxxx für das Datenbankpaßwort, unter dem das Feld definiert wurde. (vgl. VERBINDE-Kommando - Seite 182)

Ist in einem CIS-Kommando dieser DB-Parameter angegeben, so muß ein Verbund-Paßwort gültig sein, oder das Kommando mit ',DB.Verbundpaßwort' abgeschlossen werden.

SO={UF/US/F/S} Der SO-Parameter wird beim SORTIERE-Kommando (vgl. Seite 136) verwendet. Für jedes zu sortierende Feld läßt sich eine eigene Sortierreihenfolge (fallend oder steigend) angeben.

UF Ergebnis der Sortierung in absteigender Reihenfolge entsprechend einer USER-Tabelle.

US Ergebnis der Sortierung in aufsteigender Reihenfolge entsprechend einer USER-Tabelle.

F Ergebnis der Sortierung in absteigender Reihenfolge.

S Ergebnis der Sortierung in aufsteigender Reihenfolge.

Der Anwender kann nach Rücksprache mit dem CIS-Team die CIS-Sortierungstabelle auf eigene Belange anpassen (USER-Tabelle).

1.5.5 Datenbankpaßwort

Das Datenbankpaßwort ist immer 6 Zeichen lang und ist identisch mit dem Namen der Datenbeschreibung oder Bildschirmmaske. Dem Paßwort werden zur Kennzeichnung immer die vier Zeichen ",DB." vorangestellt.

Beispiel:
```
SU REGION=MUENCHEN,DB.SATZBE
Z,1,M,DB.MASK01
WR,K,DATEI=STAMM.MUENCHEN,DB.TRANS1
```

Als Datenbeschreibung gelten:
- Satzbeschreibung
- Transformationsbeschreibung
- Bildbeschreibung
- Verbundbeschreibung

Es gibt immer nur eine aktuelle Datenbeschreibung. Wird eine Bildschirmmaske als Paßwort angegeben, dann ist die implizite Satzbeschreibung die aktuelle Datenbeschreibung (vgl. Manual-2: DIENSTPROGRAMME Bildschirmmasken).

Die folgenden Beispiele beziehen sich auf eine Kunden- und Auftragsdatenbank mit den Satzbeschreibungen KUNDEN und AUFTRG. Die zugehörigen Bildschirmmasken sind KUNDM1 und AUFTM1.

Beispiele:

`/START-PROGRAM FROM-FILE=CIS` `EI,K,DB.KUNDM1`	Gültige Bildschirmmaske: KUNDM1 Gültige Datenbeschreibung: KUNDEN.

`SU KUNDENNR=500123`
`Z,1,M (KUNDM1)`
`AE,1,F,KUNDE=MUELLER`
	Ohne Paßwortangabe, da KUNDEN und KUNDM1 aktuelle Paßwörter sind.

`SU AUFTRGNR=500123,DB.AUFTRG`	Die neue Datenbeschreibung AUFTRG bedeutet einen Datenbankwechsel, so daß die Maske KUNDM1 ungültig gesetzt wird.
`K,K,DB.AUFTM1`	Das Paßwort ist notwendig, da keine Maske gültig war.
`SU AUFTRGNR=600987` `Z,1,M (AUFTM1)`	Ohne Paßwortangabe, da AUFTRG und AUFTM1 aktuelle Paßwörter sind.
`EI,K,DB.KUNDM1` `SU KUNDE=BMW`	Die neue Maske KUNDM1 setzt implizit die Datenbeschreibung AUFTRG ungültig und aktualisiert KUNDEN.

1.6 Behandlung numerischer Feldinhalte

1.6.1 Eingabe numerischer Feldinhalte

Bei der Eingabe numerischer Werte sind folgende Aufbereitungsregeln zu beachten:

- Vorlaufende Spaces werden übergangen.
- Das Vorzeichen muß, wenn angegeben, als erstes signifikantes Zeichen (ungleich Space) eingegeben werden.
- Ein fehlendes Vorzeichen wird als + (positiv) interpretiert.
- Trenner zwischen ganzzahligem und gebrochenem Teil ist der Dezimalpunkt.
- Die eigentliche Ziffernfolge darf außer einem Dezimalpunkt nur Ziffern enthalten.
- Wird kein Dezimalpunkt angegeben, so wird die Position des Dezimalpunktes direkt hinter der letzten angegebenen Ziffer angenommen.
- Die interne Ausrichtung wird anhand des eingegebenen oder angenommenen Dezimalpunkts vorgenommen.
- Nach links oder rechts überlaufende Stellen werden ohne Fehlermeldung abgeschnitten.

1.6.2 Ausgabe numerischer Feldinhalte

Beim ZEIGE- bzw. DRUCKE-Kommando werden numerische Feldinhalte wie folgt aufbereitet:

- Aufbereiten ins externe Format bei den Feldbedeutungen H, B und P (gepacktes Feld).
- Einfügen des Dezimalpunktes, wenn Dezimalstellen definiert sind.
- Unterdrückung führender Nullen.
- Einfügung des evtl. negativen Vorzeichens vor die höchstwertige gültige Ziffer.
- Rechtsbündige Ausrichtung der Darstellung.
- Einfügen der Zeichenfolge '0.(00...)', wenn die Dezimalstellen größer oder gleich der maximalen Ziffernzahl definiert wurden.

Beim WRITE-Kommando wird nur entsprechend einer Transformationsanweisung aufbereitet.

1.6.3 Darstellung numerischer Feldinhalte in der CIS-Verweisdatei

Invertierte numerische Feldinhalte werden nicht 1:1 aus der Hauptdatei in die Verweisdatei übernommen. Bei negativen Werten wird ein 9er-Komplement gebildet und dieses in der CIS-VD gespeichert. Damit wird erreicht, daß negative Werte auch vor positiven Werten eingeordnet werden.

Ferner wird jede negative Zahl mit dem sedezimalen Zeichen 'D' und jede positive Zahl mit dem Zeichen 'F' dargestellt.

Beispiel für 2-stellig gepackte Zahlen (Feldbedeutung=P)

Werte	im Satz	in der VD
-2	X'002D'	X'D997'
-1	X'001B'	X'D998'
+1	X'001F'	X'F001'
+2	X'002E'	X'F002'
+2	X'002C'	X'F002'
+2	X'002A'	X'F002'

Beispiel für 2-stellig entpackte Zahlen (Feldbedeutung=R)

Wert	im Satz	in der VD
-2	X'F0D2'	X'D9F7'
-1	X'F0B1'	X'D9F8'
+1	X'F0F1'	X'F0F1'
+2	X'F0E2'	X'F0F2'
+2	X'F0C2'	X'F0F2'
+2	X'F0A2'	X'F0F2'

1.6.4 Darstellung numerischer Feldinhalte bei LEASY

LEASY bewertet numerische Feldinhalte bei der Sekundärindizierung nicht, d.h. LEASY bildet die Werte in der SI-Datei 1:1 ab. Damit sind die Werte nach ihrem binären Äquivalent sortiert und bei uneinheitlicher Vorzeichenverwendung können für den gleichen numerischen Wert mehrere SI-Sätze entstehen.

Beispiel für gepackte Zahlen (Feldbedeutung=P):

Werte	im Satz
-2	X'2D'
-1	X'1B'
+1	X'1F'
+2 (1)	X'2E'
+2 (2)	X'2C'
+2 (3)	X'2A'

In der SI-Datei stehen die Sätze in folgender Reihenfolge:

```
X'1B' = -1
X'1F' = +1
X'2A' = +2 (3)
X'2C' = +2 (2)
X'2D' = -2
X'2E' = +2 (1)
```

In der CIS-VD würden die Sätze in folgender Reihenfolge stehen:

```
X'D7' = -2
X'D8' = -1
X'F1' = +1
X'2E' = +2 (mit 3 Einträgen)
```

Beispiel für entpackte Zahlen (Feldbedeutung=R):

Wert	im Satz
-2	X'D2'
-1	X'B1'
+1	X'F1'
+2 (1)	X'E2'
+2 (2)	X'C2'
+2 (3)	X'A2'

In der SI-Datei stehen die Sätze in folgender Reihenfolge:

```
X'A2' = +2 (3)
X'B1' = -1
X'C2' = +2 (2)
X'D2' = -2
X'E2' = +2 (1)
X'F1' = +1
```

In der CIS-VD würden die Sätze in folgender Reihenfolge stehen:

```
X'D7'  = -2
X'D8'  = -1
X'F1'  = +1
X'F2'  = +2  (mit 3 Einträgen)
```

1.6.5 Suchen in der SI-Datei

Aus dem oben Dargestellten ergibt sich, daß bei Suchfragen, die numerische Feldinhalte enthalten, die interne Darstellung des Feldinhalts zu berücksichtigen ist.

Beispiel für gepackte Zahlen (Feldbedeutung=P):

`SU Feld = 123`	sucht nach = `X'123F'`	
`SU Feld = -123`	sucht nach = `X'123D'`	
`SU Feld = X'123v'`	sucht nach = `X'123v'`	wobei v jedes mögliche Hexa-Zeichen sein kann.

Das gleiche gilt für alle anderen Operatoren, wobei "logisch" verglichen wird:

```
X'1A' < X'1B' < X'1C' < X'1D' < X'1E' < X'1F'
```

Beispiel für entpackte Zahlen (Feldbedeutung=R):

`SU Feld=123`	sucht nach = `X'F1F2F3'`	
`SU Feld=-123`	sucht nach = `X'F1F2D3'`	
`SU Feld=X'F1F2v3'`	sucht nach = `X'F1F2v3'`	wobei v jedes mögliche Hexa-Zeichen sein kann.

Das gleiche gilt für alle anderen Operatoren, wobei "logisch" verglichen wird.

```
X'A1' < X'B1' < X'C1' < X'D1' < X'E1' < X'F1'
```

1.6.6 Balkendiagramm aus der SI-Datei

Die unterschiedliche Darstellungsweise der numerischen Feldinhalte bei LEASY und CIS wirkt sich auch bei der Ausgabe der Verweisdateibelegung aus.

`Z,n,VB,Feldbezeichnung`	bringt alle Feldinhalte in der Reihenfolge, wie sie in der SI-Datei stehen. Es werden jedoch die Werte mit richtigem Vorzeichen ausgegeben.

Beispiel für gepackte Zahlen (Feldbedeutung=P):

Werte in der SI-Datei:	Darstellung
`X'1B'...`	`-1...`
`X'1F'...`	` 1...`
`X'2A'...`	` 2...`
`X'2C'...`	` 2...`
`X'2D'...`	`-2...`
`X'2E'...`	` 2...`
`Z,n,VB,Feld*123`	setzt bei `X'123F'` auf
`Z,n,VB,Feld*-123`	setzt bei `X'123D'` auf
`Z,n,VB,Feld*X'123v'`	setzt bei `X'123v'` auf
	(v=jedes mögliche Hexa-Zeichen)

Beispiel für entpackte Zahlen (Feldbedeutung=R):

folgende Werte stehen in der SI-Datei:	Darstellung
`X'A2'...`	` 2...`
`X'B1'...`	`-1...`
`X'C2'...`	` 2...`
`X'D2'...`	`-2...`
`X'E2'...`	` 2...`
`X'F1'...`	` 1...`
`Z,n,VB,Feld*123`	setzt bei `X'F1F2F3'`auf
`Z,n,VB,Feld*-123`	setzt bei `X'F1F2D3'`auf
`Z,n,VB,Feld*X'F1F2v3'`	setzt bei `X'F1F2v3'`auf
	(v=jedes mögliche Hexa- Zeichen)

2 Kommandos

2.1 Tabellarische Übersicht der CIS-Kommandos

Auf den nächsten Seiten folgt die tabellarische Übersicht aller CIS-Kommandos.

Nach dem Operanden der Kommandos, außer bei HALT, VERBINDE und den Kommandos für die verteilte Transaktionsverarbeitung, kann jeweils das Datenbankpaßwort, DB.xxxxxx, getrennt durch ein Komma, angegeben werden.

Operation Erläuterung	Menge	Operations- ergänzung	Operand
AENDERN			
Abschnitt		,A	[,(Erg)]
Feld	[,n]	,F,	Feld[(Erg)]=Wert
Satz / Satz direkt		,{K/KD}	
Feld (gekettet)	[,n]	,KF,	Feld[(Erg)]=Wert[,Feld[(Erg)]=Wert]...
AKTIVIEREN			
VD-Zeiger/temporäre ZPL		{,E,/_/,I,}	Feld[(Erg)]=Wert
BLAETTERN			
direkt		,D,	Key{=/</>/:/*}Wert
indirekt im VD-Eintrag		,I,	Feld{=/</>/:/*}Wert[, Key{=/</>/:/*}Wert]
vor-/ rückwärts	,n	,{V/R}	
CLOSE			
aktuelle Datei			
alle Dateien		,A	
SYSLST			_SYSLST
HD/VD/ISISOUT			_{HD/VD/ISISOUT}=Name[,CLOSE...]...
DRUCKE			
Feldauskunft	[,n]	,A	
Bildausgabe	[,n]	,B	
Gruppensumme/tabellarisch/summarisch	[,n]	,{G/GT/GS},	Sortfeld[(Erg1)]_Feld[(Ergn)][_Feld...]...
tabellarisch	{[,n]	,T,/_}	{Feld[(Erg)]/Var[Kom]}_...
Verteilungstafel / summarisch	[,n]	,{V/VS},	Sortfeld[(Erg)]_Feld[(Erg)]{*/=/</>/:/#}Wert[,Feld...]...
Balkendiagramm	[,n]	,{VB/SB},	Feld[(Erg)][*Wert]
zeilenweise	[,n]	,Z,	{Feld[(Erg)]/Var[Kom]}_...
EINGEBEN			
Textende	[,n]	,{A/K}	
ENDE			
Textende			

Operation Erläuterung	Menge	Operationsergänzung	Operand
EXIT			
ADILOS/Test-ADILOS	,n	,{A/TA},	SYSTEM_EINGABE=Steuerdatei
Listenprogramm	,n	,L,	ILP=Programmname
FREIGEBEN			
ZPL: temporär/aus VD/VD-Zeiger		{_/,{E/V/I},}	Feld[(Erg)]=Wert
ZPL für VERBINDE		,J,	Feld
Suchergebnis		,S	
GET			
Abschnitt [mit Sperre]		,A[S],	Abschnittsname[(Erg)]
Erfassung über Bildschirm		,B	
Feld [mit Sperre]		,F[S],	Feld[(Erg)]
alle Feldbeschreibungen mit Trennern		,FAA	
Feldbeschreibung [mit Trennern]		,FB[A]	[,Feld[(Erg)]
1. Feldbeschreibung [mit Trennern]		,FE[A]	
Satz direkt [mit Sperre]		,KD[S],	Key{=/</>/:/*}Wert
Satz feldweise/nächster/rückwärts/Pointer [mit Sperre]	[,n]	,KF{N/R/P}[S],	Feld[(Erg)][_Feld...]...
Satz indirekt über VD [mit Sperre]		,KI[S],	Feld{=/</>/:/*}Wert[,Key{=/</>/:/*}Wert]
Nächster Satz/rückwärts [mit Sperre]		,{KN/KR}[S]	
Satz [mit Sperre]		,KP[S]	
Änderung über Bildschirm		,P	[(Erg)]
ZPL		,ZP	[,Key[(Erg)][_Key...]...
HALT			
CIS Beenden			
HILF			
Kommandoerläuterung			[_Kommando]
IGNORIEREN			
Suchergebnis/Fenster/ZPL/VD-Zeiger		{_/,{S/F/Z/I}}	
KORRIGIEREN			
Satz	[,n]	,K	

Operation Erläuterung	Menge	Operations- ergänzung	Operand
LOESCHEN			
Abschnitt	[,n]	,A,	Abschnittsart[(Erg)]
Wiederholfeld	[,n]	,F,	Feld[(Erg)]
Satz	[,n]	,K	
Satz direkt		,KD,	Key=Wert
OPEN			
Öffnen zum Update/Lesen		{_/,{U/I},}	{HD/VD}=Dateiname[OPEN...]...
PUT			
Abschnitt		,A	
Abschnitt feldweise		,AF,	Feld=Wert[,Feld]...
Wiederholfeld	[,n]	,F,	Feld[(Erg)]=Wert
Satz		,K	
Satz feldweise		,KF,	Feld=Wert[,Feld]...
RECHNE			
summ./zeilenw.	{,n,/[,n]	,{S/Z},/_}	Ergebnis[(Erg)]=Ausdruck[(Erg)][,[{S,/Z,}]Ergebnis...]...
SETZE			
externes/internes Format		,{E/I}	
groß/klein		,{G/K}	
LOGADR gesetzt/nicht gesetzt		,{LJ/LN}	
Modus normal/spezial/Abfrage		,{MN/MS/MQ}	
sequentiell		,S	
Zwischentrenner/Endetrenner		,{TZ/TE},	{C'c'/X'xx'}
Wildcard ein/viele Zeichen		,{WE/WV},	{C'c'/X'xx'}
SICHERN			
temporär/VD-Zeiger		{_/,{E/I},}	Feld[(Erg)]=Wert
ZPL für VERBINDE		,J,	Feld[(Erg)]
Suchergebnis		,S	
ZPL in VD		,V,	Pseudofeld=Wert

Operation Erläuterung	Menge	Operations- ergänzung	Operand
SORTIERE			
steigend/fallend/eindeutig		{_/,{S/F/E},}	Feld[(Erg)]_...
SPERRE			
Datei/ZPL		,{D/Z}	
Satz direkt		,KD,	Key=Wert
STATUS			
aktiver Dateien/aller TID´s		,{DG/DT}	
Sätze in aktiven HD/TID´s		,{SD/ST}	
Sätze in HD+TID/allen HD+TID´s		,{SA/SG}	
SUCHE			
Initialfrage		{_/,S,}	Feld[(Erg)]{*/=/#/</>/:}Wert[,{UND/ODER/NICHT}_...]...
Bereichsinitialfrage		{_/,S,}	Feld[(Erg)]{*/=/>}Wert1,BIS_Feld{:/=/>}Wert2,...
{UND/ODER/NICHT}			
Folgefrage		{_/,S,}	Feld[(Erg)]{*/=/#/</>/:}Wert[,{UND/ODER/NICHT}_...]...
Bereichsfolgefrage		{_/,S,}	Feld[(Erg)]{*/=/>}Wert1,BIS_Feld{:/=/>}Wert2,...
SYSTEM			
BS2000-Kommando			[_/Kommando]
TEXT			
			[_Text]
TRANSAKTION			
Anfang/Anfang lesen/Neu aufsetzen		,{A/AL/AR}	
Ende/Ende bedingt/Rücksetzen		,{E/EB/R}	
Rücksetzen bedingt		,RB	
TRANSAKTION **(Verteilte Transaktionsverarbeitung)**			
Anfang mit Status/Provisorisches Ende		,{AS/PE},	
Statusabfrage/Löschen Statuseintrag		,{SA/SL},	TID=xxxxxxxx
Beenden-/Rücksetzen im Zustand PTC		,{WE/WR},	TID=xxxxxxxx

Operation Erläuterung	Menge	Operations- ergänzung	Operand
VERBINDE			
auftragsbezogen/forciert/singulär		{_/,{A/F/S},}	Feld1(DB=aaaaaa)=Feld2(DB=bbbbbb)[_Feldn...]...,DB.vvvvvv
auftragsbez.forciert/auftragsbez.singulär		,{AF/AS},	Feld1(DB=aaaaaa)=Feld2(DB=bbbbbb)[_Feldn...]...,DB.vvvvvv
forciert singulär/auftragsbez.forciert singulär		,{FS/AFS},	Feld1(DB=aaaaaa)=Feld2(DB=bbbbbb)[_Feldn...]...,DB.vvvvvv
WARUM			
Erläuterung Fehlercode			_cccc
WIEDERHOLEN			
Letzte(s) Kommando(s)			[_n]
WRITE			
Satz/Originalsatz	[,n]	,{K/KO},	DATEI=Name
Selektiv/Selektiv mit Trenner/Transferdatei	[,n]	,{S/STT/TS},	DATEI=Name,Feld[(Erg)][{=Wert[,Feld...]/_Feld...}...
ZEIGE			
Feldauskunft/Bildausgabe/gesicherte ZPL und VD-Zeiger	[,n]	,{A/B/E}	
Gruppensumme/tabellarisch/summarisch	[,n]	,{G/GT/GS},	Sortfeld[(Erg)]_ Feld[(Erg)][_Feld...]...
Satz in Maske	[,n]	,M	
Tabellarisch/tabellarisch hexadezimal	[,n]	{,T,/,XT,/_}	{Feld[(Erg)]/Var[Kom]}_...
Verteilungstafel/summarisch	[,n]	,{V/VS},	Sortfeld[(Erg)]_Feld[(Erg)]{*/=/</>:/#}Wert[,Feld...]...
Balkendiagramm	[,n]	,VB,	Feld[(Erg)][*Wert]
Zeilenweise	[,n]	,Z,	{Feld[(Erg)]/Var[Kom]}_...

Operation Erläuterung	Operations- ergänzung	Operand
$D_ für CISDBH		
Kommunikation steuern	DENA	_ x
Beendigungsbedingung setzen	DEND	_ b
Information über angeschlossene Partner	DINFC	[_ p]
Information über aktive Transaktionen	DINFT	[_ p]
$D_ für CISKOOR		
Kommunikation steuern	KENA	_ x
Beendigungsbedingung setzen	KEND	_ b
Information über Before-Images	KINFB	[_ p]
Information über angeschlossene Partner	KINFC	[_ p]
Information über Sperren	KINFL	[_ F=Dateiname][_ p]
Information über Transaktion in PTC	KINFP	[_ p]
Information über Stati	KINFS	[_ p]
Information über aktive Transaktionen	KINFT	[_ p]
Anzahl Tasks vorgeben	KNTASKS	_ n
PTC beenden	KPEND	_ p
PTC rücksetzen	KPRESET	_ p
$P		
Programm Anfang/Ende	,{A/E}	
$T		
Terminal Anfang/Abbruch/Ende	,{A/D/E}	

2.2 AENDERN-Kommando

Mit dem Update-Kommando AENDERN können bereits vorhandene Daten geändert werden. Die Ausführung des Kommandos ist jeweils auf Satz-, Abschnitts- und Feldebene möglich. Ist beim Ändern von Wiederholabschnitten oder Wiederholfeldern keine Multiplität explizit angegeben, so wird der erste Abschnitt bzw. das erste Feld angesprochen.

Sind invertierte Felder vorhanden, wird die Verweisdatei ebenfalls aktualisiert. Die Prüfung, ob die Felder invertiert sind, erfolgt über die beim aktuellen Kommando verwendete Datenbeschreibung. Es ist deshalb darauf zu achten, daß die dabei verwendete Datenbeschreibung alle invertierten Felder, die geändert werden, enthält.

2.2.1 AENDERN-A (Abschnitt)

Anwendungsart: passiv

```
AENDERN,A[,(Erg)][,DB.xxxxxx]
```

AENDERN	Operation für Ändern.
A	Operationsergänzung für Abschnitt.
Ergänzung (vgl. Seite 18)	AM = Abschnittsmultiplität: Bei fehlender Angabe wird AM=1 angenommen.
xxxxxx	Satzbeschreibung

Vom aufrufenden Programm wird ein Abschnitt übergeben, der im lokalisierten Satz einen entsprechenden Abschnitt ersetzt. Der übergebene Abschnitt muß mindestens Abschnittslänge (AL) und Abschnittsart (AA) enthalten.

In der Regel geht einem AENDERN-A ein Leseaufruf wie z. B. GET-A voraus. Wenn der Abschnitt nach Inhaltsänderungen mit AENDERN-A zurückgeschrieben wird, darf sich auch die Abschnittslänge verändern, d. h. größer oder kleiner werden. Im AL-Feld steht die neue Länge des Abschnitts.

Für Wiederholabschnitte gilt: Wird eine Ergänzung angegeben, wird der entsprechende Abschnitt geändert. Fehlt die Ergänzung, wird der erste Abschnitt geändert.

2.2.2 AENDERN-F (Feld)

Anwendungsart: aktiv/passiv

```
AENDERN[,n],F,Feld[(Erg)]=Wert[,DB.xxxxxx]
```

AENDERN	Operation für Ändern.
n	Auf Sätze bezogene Mengenangabe. $ = alle Sätze. Bei fehlender Angabe wird 1 angenommen.
F	Operationsergänzung für Feld.
Feld	Name des zu ändernden Feldes.
Ergänzung (vgl. Seite 18)	AM = Abschnittsmultiplität AZ = Abschnittszyklen FM = Feldmultiplität FZ = Feldzyklen Wird nichts angegeben, so wird jeweils die erste Multiplität bzw. ein Zyklus angenommen.
Wert	Neuer Feldinhalt.
xxxxxx	Satzbeschreibung

Durch Angabe der Feldbezeichnung und Übergabe des neuen Feldinhaltes wird im lokalisierten Satz die Änderung vorgenommen.

Soll ein Feldinhalt mit Spaces (Leerzeichen) gefüllt werden, so ist Space in Hochkommas anzugeben.

Für die Eingabe numerischer Werte sind die Hinweise auf Seite 21 ff zu beachten.

Wird mit Transaktionen gearbeitet, so muß der Satz gesperrt sein.

2.2.3 AENDERN-K (Satz)

Anwendungsart: passiv

```
AENDERN,K[,DB.xxxxxx]
```

AENDERN	Operation für Ändern.
K	Operationsergänzung für Satz.
xxxxxx	Satzbeschreibung

Der vom aufrufenden Programm übergebene Satz ersetzt (überschreibt) den lokalisierten Satz in der Datenbank. Das bedeutet insbesondere:

- Die Satzlänge darf größer oder kleiner werden, d.h. es können Felder und/oder Abschnitte hinzugefügt oder gelöscht werden.
- Die Abschnittslänge kann ebenfalls größer oder kleiner werden.

Die Lokalisierung des Datenbanksatzes erfolgt nach folgenden Regeln (vgl. auch GET-Kommando - Seite 78):

- Ist eine Zielpunktliste vorhanden, so muß der im Satz eingesetzte Ordnungsbegriff dem aktuellen Ordnungsbegriff der Zielpunktliste (laut ZPL-Zeigerstand) entsprechen, da sonst die Änderung abgewiesen wird.

  Fehlermeldung: `UP07 ÄNDERN ORDNUNGSBEGRIFF NICHT ERLAUBT`

- Ist keine Zielpunktliste vorhanden, so ist für die Änderung der im Satz übergebene Ordnungsbegriff maßgebend.

Wird mit Transaktionen gearbeitet, so muß der Satz gesperrt sein.

2.2.4 AENDERN-KD (Satz direkt)

Anwendungsart: passiv

```
AENDERN,KD[,DB.xxxxxx]
```

AENDERN	Operation für Ändern.
KD	Operationsergänzung für Satz direkt.
xxxxxx	Satzbeschreibung

Der vom aufrufenden Programm übergebene Satz ersetzt (überschreibt) den lokalisierten Satz in der Datenbank. Das bedeutet insbesondere:

- Die Satzlänge darf größer oder kleiner werden, d.h. es können Felder und/oder Abschnitte hinzugefügt oder gelöscht werden.
- Die Abschnittslänge kann ebenfalls größer oder kleiner werden.

Die Lokalisierung des Datenbanksatzes erfolgt durch den im Satz übergebenen Ordnungsbegriff, d.h. eine eventuell bestehenden Zielpunktliste wird nicht berücksichtigt.

Wird mit Transaktionen gearbeitet, so muß der Satz gesperrt sein.

2.2.5 AENDERN-KF (Satz feldweise)

Anwendungsart: aktiv

```
AENDERN[,n],KF,Feld[(Erg)]=Wert[,Feld[(Erg)]=Wert]...[,DB.xxxxxx]
```

AENDERN	Operation für Ändern.
n	Auf Sätze bezogene Mengenangabe. $ = alle Sätze. Bei fehlender Angabe wird 1 angenommen.
KF	Operationsergänzung für Satz feldweise.
Feld	Name des zu ändernden Feldes.
Ergänzung (vgl. Seite 18)	AM = Abschnittsmultiplität AZ = Abschnittszyklen FM = Feldmultiplität FZ = Feldzyklen Wird nichts angegeben, so wird jeweils die erste Multiplität bzw. ein Zyklus angenommen.
Wert	Neuer Feldinhalt.
xxxxxx	Satzbeschreibung

Im lokalisierten Satz (bzw. lokalisierten Sätzen) werden die angegebenen Felder geändert. Die nicht angegebenen bleiben unverändert.

Dieses Kommando entspricht einem geketteten AE,F.

Wird mit Transaktionen gearbeitet, so muß der Satz gesperrt sein.

2.3 AKTIVIEREN-Kommando

Anwendungsart: aktiv/passiv

```
AKTIVIEREN{_/,E,/,I,}Feld[(Erg)]=Wert[,DB.xxxxxx]
```

AKTIVIEREN	Operation für Aktivieren.
E	Operationsergänzung für temporäre ZPL (Standardwert).
I	Operationsergänzung für VD-Zeiger.
Feld	Feldbezeichnung zur Bestimmung der gesicherten Zielpunktliste.
Ergänzung (vgl.Seite 18)	DB = Datenbankpaßwort
Wert	Wert, der bei SICHERN,E/I verwendet wurde.
xxxxxx	Satzbeschreibung

- AKTIVIEREN-E: Jede temporär gesicherte Zielpunktliste wird durch Angabe von Feld=Feldinhalt aktiviert. Sämtliche Zeigerstände werden so aktiviert, wie sie durch SICHERN-E (vgl. Seite 129) festgehalten wurden.

  Anmerkung: Dieses Kommando aktiviert nur temporär gesicherte Zielpunktlisten. Die Aktualisierung der permanent in der Verweisdatei gesicherten Zielpunklisten erfolgt über Suchfragen (vgl. Seite 134).

- AKTIVIEREN-I: Der mit SICHERN-I (vgl. Seite 131) gesicherte VD-Zeiger wird aktiviert.

Beispiel:

```
*
SUCHE GEHALT<3000
IM00    ANZAHL ZIELINFORMATIONEN: 1500
*
SICHERN,E,GEHALT=3
*
SUCHE GEHALT>3000
IM00    ANZAHL ZIELINFORMATIONEN: 620
*
AKTIVIERE,E,GEHALT=3
IM00    ANZAHL ZIELINFORMATIONEN: 1500
*
UND BERUF=DIPL.ING
IM00    ANZAHL ZIELINFORMATIONEN: 2
*
```

2.4 BLAETTERN-Kommando

2.4.1 BLAETTERN-D (direkt)

Anwendungsart: aktiv/passiv

```
BLAETTERN,D,Key{=/</>/:/*}Wert[,DB.xxxxxx]
```

BLAETTERN	Operation für Blättern.
D	Operationsergänzung für direktes Blättern.
Key	Ordnungsbegriff der Hauptdatei.
Wert	Feldinhalt
xxxxxx	Satzbeschreibung

Der Satzzeiger wird in der Zielpunktliste auf den Satz gesetzt, dessen Ordnungsbegriff die angegebene Bedingung als erster erfüllt. Kann die Bedingung nicht erfüllt werden, so erfolgt die Fehlermeldung:

```
AF09 SATZ NICHT VORHANDEN
```

Liegt keine Zielpunkliste vor, wird der FCB-Zeiger der Hauptdatei entsprechend der Bedingung positioniert.

2.4.2 BLAETTERN-I (indirekt in einer VD)

Anwendungsart: aktiv/passiv

```
BLAETTERN,I,Feld{=/</>/:/*}Wert1[,Key{=/</>/:/*}Wert2][,DB.xxxxxx]
```

BLAETTERN	Operation für Blättern.
I	Operationsergänzung für indirektes Blättern in der Verweisdatei.
Feld	Invertiertes Feld.
Wert1	Feldinhalt
Key	Ordnungsbegriff der Hauptdatei.
Wert2	Feldinhalt
xxxxxx	Satzbeschreibung

Entsprechend der Bedingung für das invertierte Feld wird ein Verweisdateisatz lokalisiert und der VD-Zeiger gesetzt. Die zweite Bedingung positioniert den Zeiger auf dem VD-Eintrag, wenn zu einem Wert1 mehrere Wert2 eingetragen sind.

Existiert eine ZPL, so wird der ZPL-Zeiger nicht verändert, er behält nach wie vor die höchste Priorität. D.h. ein CIS-Kommando mit dem ausgewertet wird (z.B. ZEIGE) verwendet den ZPL-Zeiger, nicht aber den VD-Zeiger. Ist dies nicht erwünscht, muß explizit mit IGNORIEREN-Z (vgl. Seite 99) die ZPL gelöscht werden.

Allgemeine Informationen zur Zeigertechnik bei CIS sind im Kapitel Zeigertechnik (vgl. Seite 8) dargestellt.

Mit dem IGNORIEREN-I-Kommando (vgl. Seite 97) kann der VD-Zeiger wieder ungültig gesetzt werden.

2.4.3 BLAETTERN-V/R (vorwärts/rückwärts)

Anwendungsart: aktiv/passiv

```
BLAETTERN,n,{V/R}[,DB.xxxxxx]
```

BLAETTERN	Operation für Blättern.
n	Auf Sätze bezogene Mengenangabe. $ = alle Sätze.
V	Operationsergänzung für vorwärts.
R	Operationsergänzung für rückwärts.
xxxxxx	Satzbeschreibung

Der Satzzeiger wird zur Lokalisierung eines Satzes vorwärts oder rückwärts positioniert.

Dabei wird als Satzzeiger entweder der ZPL-Zeiger, der VD-Zeiger (falls kein ZPL-Zeiger vorhanden) oder der FCB-Zeiger (falls weder ZPL-Zeiger noch VD-Zeiger vorhanden) verwendet.

Allgemeine Informationen zur Zeigertechnik bei CIS sind im Kapitel Zeigertechnik (vgl. Seite 8) aufgeführt.

Ist zum Zeitpunkt des Kommandoaufrufs eine Zielpunktliste vorhanden, so wird der Zeiger auf einen Ordnungsbegriff der Hauptdatei innerhalb dieser Zielpunktliste gesetzt. Der zu diesem Ordnungsbegriff gehörende Satz ist nun lokalisiert. Das nächste Kommando kann jetzt auf diesen Satz zugreifen.

Beispiel 1: Blättern in der Zielpunkliste

```
*
SUCHEN GEBMON=AUGUST,DB.JACOBI
IM00    ANZAHL ZIELINFORMATIONEN: 5
*
ZEIGE GEBMON NAME
GEBMON NAME
AUGUST KONRAD
AUGUST REGLER
AUGUST LANGE
AUGUST MEINHARD
AUGUST KIEHLMEIER
*
BLAETTERN,$,R
*
ZEIGE,2,T,GEBMON NAME
GEBMON NAME
AUGUST KONRAD
AUGUST REGLER
*
BLAETTERN,$,V
*
ZEIGE,1,T,GEBMON NAME
GEBMON NAME
AUGUST KIEHLMEIER
*
BL,$,R
*
SORT NAME
IM01    ANZAHL ZIELINFORMATIONEN: 5
*
ZEIGE,2,T,GEBMON NAME
GEBMON NAME
AUGUST KIEHLMEIER
AUGUST KONRAD
*
BLAETTERN,$,V
*
ZEIGE,1,T,GEBMON NAME
GEBMON NAME
AUGUST REGLER
*
```

Auf ersten Ordnungsbegriff der Zielpunktliste für nachfolgende Sortierung zurückblättern.

Auf letzten Ordnungsbegriff der Zielpunktliste vorblättern.

Auf letzten Ordnungsbegriff der Zielpunktliste vorblättern.

Beispiel 2: Blättern in der Verweisdatei

```
*
IGNORIEREN,Z
*
BLAETTERN,I,GEBMON=AUGUST
Z,2,T,GEBMON NAME
GEBMON NAME
AUGUST KONRAD
AUGUST REGLER
*
BLAETTERN,2,V
*
ZEIGE,2,T,GEBMON NAME
GEBMON NAME
AUGUST MEINHARD
AUGUST KIEHLMEIER
*
```

Letzte Zielpunktliste wird ignoriert VD-Zeiger wird gesetzt.

Beispiel 3: Blättern in der Hauptdatei

```
*
IGNORIEREN,Z
*
IGNORIEREN,I
Z,2,T,GEBMON NAME
GEBMON NAME
AUGUST MEINHARD
JUNI   STROER
*
BLAETTERN,10,V
*
ZEIGE,2,T,GEBMON NAME
GEBMON NAME
MAI    WAGNER
MAERZ  ZERNER
*
```

Letzte Zielpunktliste wird ignoriert. VD-Zeiger wird gelöscht.

2.5 CLOSE-Kommando

Anwendungsart: aktiv/passiv

```
CLOSE{{_/,A/_SYSLST}/_{HD/VD/ISISOUT}=Name}
     [,CLOSE...]...[,DB.xxxxxx]
```

CLOSE	Operation für CLOSE.
_	Space = aktuelle Datei (Standardwert).
A	Operationsergänzung für alle Dateien.
SYSLST	Operationsergänzung für die aktuelle SYSLST-Datei (UTM- und DBH-Betrieb).
HD	Bezeichnung für Hauptdatei.
VD	Bezeichnung für Verweisdatei.
ISISOUT	Bezeichnung für ISISOUT-Datei.
Name	Name der HD oder VD oder ISISOUT-Datei.
xxxxxx	Satzbeschreibung

Mit dem CLOSE-Kommando (expliziter CLOSE) werden offene Dateien geschlossen.

Für das CLOSE-Kommando sind verschiedene Schreibweisen möglich:

- Mit Angabe von Dateinamen:

  Die zu schließende Haupt-, Verweis- bzw. ISISOUT-Datei wird im CLOSE-Kommando angegeben. SYSLST bezeichnet den symbolischen Dateinamen der aktuellen SYSLST-Datei. Die angegebenen Dateien werden geschlossen. Bei Haupt- bzw. Verweisdatei muß dabei im A-Segment der aktuellen Satzbeschreibung der Parameter EOC=J eingetragen sein.

  CLOSE ISISOUT=... ist auch bei EOC=N zulässig.

  Beispiel 1:

  ```
  CLOSE HD=HD.PERS,CLOSE VD=VD.PERS,DB.PERSON
  ```

  Die Hauptdatei HD.PERS und die Verweisdatei VD.PERS werden geschlossen (der Parameter EOC=J ist in der Satzbeschreibung eingetragen). Haupt- und Verweisdatei können, wenn für die Anwendung sinnvoll, auch getrennt (Einzelkommandos) zu unterschiedlichen Zeitpunkten geschlossen werden.

Beispiel 2:

```
*
SU ...
*
SYS /SET-FILE-LINK FILE-NAME=NAME1,LINK-NAME=ISISOUT
*
WR, ... DATEI=NAME1
*
CLOSE ISISOUT=NAME1
*
SYS /SET-FILE-LINK FILE-NAME=NAME2,LINK-NAME=ISISOUT
*
WR, ... DATEI=NAME2
.
.
```

Die zunächst als Ausgabedatei zugewiesene Datei NAME1 wird geschlossen. Danach ist es möglich, sofort eine zweite Ausgabedatei zuzuweisen (hier: NAME2) und zu beschreiben.

- Nur Angabe einer Satzbeschreibung:

  Wird im Kommando nur eine Satzbeschreibung angegeben, so bedient sich der dann implizit ablaufende CLOSE der Datenbeschreibungsparameter HD=Dateiname zum Schließen der Hauptdatei bzw. des Parameters VD=Dateiname zum Schließen der Verweisdatei.

  Beispiel:

  Die zu schließenden Dateien werden der Datenbeschreibung DB.FELD01entnommen.

```
*
CLOSE,DB.FELD01
*
```

  Wird nur CLOSE eingegeben, so hat dies die gleiche Wirkung, wie oben beschrieben. Die Datei(en) der zuletzt verwendeten Datenbeschreibung wird (werden) geschlossen.

  Beispiel:

  Über das Paßwort der Suchfrage werden die benötigten Dateien implizit geöffnet. Die Dateien der aktuellen Datenbeschreibung DB.NAMEN1 werden geschlossen.

```
/START-PROGRAM FROM-FILE=CIS
SUCHE NAME=HUBER,DB.NAMEN1
IM00    ANZAHL ZIELINFORMATIONEN: 1
*
CLOSE
*
```

- Angabe der Operationsergänzung A:

  Wird im CLOSE-Kommando die Operationsergänzung A angegeben, so werden alle offenen Dateien, auch offene Sonderdateien wie z.B. DABEL, geschlossen. Die Angabe von DB.xxxxxx ist dabei nicht erforderlich.

  Am Ende eines Passiv-Programms ist ein CLOSE-A Aufruf Pflicht (als letzter CIS-Aufruf, nicht bei synchronisiertem UTM-Betrieb), im Aktivbetrieb wird dieser automatisch durch das HALT-Kommando ausgeführt.

  Beispiel:

  Alle offenen Dateien, auch offene Sonderdateien (z.B. DABEL) werden geschlossen.

```
*
CL,A
*
```

  Die Kurzkommando-Datei wird nicht geschlossen.

2.6 DRUCKE-Kommando

Das Kommando DRUCKE ermöglicht es, Daten auf Drucker auszugeben, wobei die Anzahl der auszugebenden Einheiten über die Mengenangabe bestimmbar ist.

Es können Daten der Hauptdatei, der Verweisdatei und der Datenbeschreibung ausgegeben werden.

Numerische Felder werden beim Drucken folgendermaßen aufbereitet:

- Entpacken bei Feldbedeutung P (gepacktes Feld).
- Einfügen des Dezimalpunkts, wenn Dezimalstellen definiert sind.
- Unterdrückung führender Nullen.
- Einfügen des evtl. negativen Vorzeichens vor die höchstwertige gültige Ziffer.
- Rechtsbündige Ausrichtung der Darstellung.
- Einfügen der Zeichenfolge '0.(00...)' wenn die Dezimalstellen größer oder gleich der maximalen Ziffernzahl definiert wurden.
- Ist die FL-Angabe (Feldlänge) zu klein, d.h. es kann nicht die ganze Zahl aufbereitet werden, so wird der Überlauf in der höchstwertigen Stelle durch einen * gekennzeichnet.

Zu jedem DRUCKE-Kommando gibt es auch ein entsprechendes ZEIGE-Kommando, das die Daten am Terminal ausgibt (vgl. Seite 205).

2.6.1 DRUCKE-A (Feldauskunft)

Anwendungsart: aktiv/passiv

```
DRUCKE[,n],A[,DB.xxxxxx]
```

DRUCKE	Operation für Ausgabe auf Drucker.
n	Auf Felder bezogene Mengenangabe. $ = alle Felder. Bei fehlender Mengenangabe wird '$' angenommen.
A	Operationsergänzung für Feldauskunft.
xxxxxx	Satzbeschreibung

Aus der angegebenen Satzbeschreibung wird folgende Information zur Verfügung gestellt:

- Feldbezeichnung
- Abschnittsart (AART)
- Wiederholabschnitt (W)
- Relative Feldadresse (ADR), bei V-Format bezogen auf Satzanfang, bei MV-Format bezogen auf den Abschnittsanfang.
- Verweisdatei-Segmentname (SEGM).
- Länge in Bytes (LNG), ggf. inkl. Dezimalstellen.
- Feldbedeutung (T)
- Art der Darstellung
- Anzahl der Dezimalstellen (DZ)
- Wenn ein Verbundpaßwort gültig ist, das Dateipaßwort, in dem das Feld definiert ist.

Beispiel:

```
*
D,$,A,DB.FLDAUS
*

FELDBEZEICHNUNG  AART W  ADR SEGM LNG T DARSTELLUNG DZ PASSWORT

EDV-NR           ABCD      9         3 O BINAER          FLDAUS
PSEUDO           ABCD     12 PSE     1 P ZEICHEN LB      FLDAUS
NAME             ABCD     13 NAM0   25   ZEICHEN LB
FIRMA            ABCD     38 FIRS   70 U ZEICHEN LB
FIRM1            ABCD    108        35 U ZEICHEN LB
FIRM2            ABCD    143        35 U ZEICHEN LB
GKL              BCDE W    9 GKL     1   ENTPACKT RB
UMSATZ           BCDE W   10         6   GEPACKT     2
PROD             CDEF      9 PRDS   15 W ZEICHEN LB
```

Die Reihenfolge der Feldbeschreibungen ist wie in der Datenbeschreibung definiert. Im Verbund sind die Felder aufsteigend nach Feldbezeichnung sortiert.

2.6.2 DRUCKE-B (Bildausgabe)

Anwendungsart: aktiv/passiv

```
DRUCKE[,n],B[,DB.xxxxxx]
```

DRUCKE	Operation für Ausgabe auf Drucker.
n	Auf Sätze bezogene Mengenangabe. $ = alle Sätze. Bei fehlender Mengenangabe wird '$' angenommen.
B	Operationsergänzung für Bild.
xxxxxx	Bildbeschreibung

Die Daten eines Satzes werden in Bilder, die mit einer Bildbeschreibung definiert worden sind, eingetragen und am Drucker ausgegeben.

2.6.3 DRUCKE-G (Gruppensummen)

Anwendungsart: aktiv/passiv

```
DRUCKE[,n],{G/GT/GS},Sortfeld[(Erg1)]_Feld[(Ergn)]
      [_Feld...]...[,DB.xxxxxx]
```

(vgl. auch die vereinfachte Syntax der Operandenangabe - Seite 15)

DRUCKE — Operation für Ausgabe auf Drucker.

n — Auf Ausgabezeilen bezogene Mengenangabe.
$ = alle Ausgabezeilen aller Sätze (eine Ausgabezeile entspricht jeweils einer Gruppensumme). Bei fehlender Mengenangabe wird '$' angenommen.

G — Operationsergänzung für Gruppensummen.

GT — Operationsergänzung für Gruppensummen mit tabellarischen Gesamtsummen.

GS — Operationsergänzung für Gruppenwechsel mit zeilenweisen und tabellarischen Gesamtsummen.

Sortfeld — Gruppenwechselbegriff

Erg_1 — Ergänzungen zum Gruppenwechselbegriff:

AM = Abschnittsmultiplität: Bei fehlender AM-Angabe wird AM=1 angenommen.
FM = Feldmultiplität: Bei fehlender FM-Angabe wird FM=1 angenommen.
FP = Feldposition (nur bei Feldbedeutung T): Bei fehlender FP-Angabe wird FP=1 angenommen.
FL = Feldlänge: Bei fehlender FL-Angabe wird die definierte Feldlänge verwendet.

SO = UF Ergänzung für Sortierung in absteigender Reihenfolge entsprechend einer USER-Tabelle.
SO = US Ergänzung für Sortierung in aufsteigender Reihenfolge entsprechend einer USER-Tabelle.
SO = F Ergänzung für Sortierung in absteigender Reihenfolge.
SO = S Ergänzung für Sortierung in aufsteigender Reihenfolge.

DB = Name der Satzbeschreibung: Die Angabe ist nur beim Arbeiten im Fenster zulässig.

`Feld` (2 bis n)	Die Felder müssen numerisch definiert sein (Felder mit Feldbedeutung R oder P). Bei Operationsergänzung GS muß die Anzahl der definierten Dezimalstellen für Feld 2 bis n gleich sein.
`Erg`$_n$	Ergänzungen für die Felder 2 bis n: AM = Abschnittsmultiplität: Bei fehlender AM-Angabe wird AM=1 angenommen. FM = Feldmultiplität: Bei fehlender FM-Angabe wird FM=1 angenommen. FP = Feldposition (nur bei Feldbedeutung T): Bei fehlender FP-Angabe wird FP=1 angenommen. FL = Feldlänge: Bei fehlender FL-Angabe wird die definierte Feldlänge verwendet. DB = Name der Satzbeschreibung: Die Angabe ist nur beim Arbeiten im Fenster zulässig.
`xxxxxx`	Satzbeschreibung

Die Zielpunktliste wird aufsteigend nach dem Gruppenwechselfeld (Sortfeld) sortiert. Solange der Inhalt des Gruppenwechselfeldes gleich bleibt, werden die Felder 2 bis n jeweils addiert. Ändert sich der Inhalt des Gruppenwechselfeldes, so wird eine Gruppensummenzeile ausgegeben (Summe von Feld 2 bis Feld n). Dieser Ablauf wird fortgesetzt, bis das Ende der Zielpunktliste erreicht ist oder bis die im Kommando angegebene Anzahl der Ausgabezeilen erreicht ist.

Bei Operationsergänzung GT werden nach der Ausgabe zusätzlich in einer Abschlußzeile die Gesamtsummen für die einzelnen Felder ausgegeben.

Bei Operationsergänzung GS werden die Gesamtsummen sowohl horizontal als auch vertikal gebildet.

Beispiele:

```
SU TOUR=65,BIS TOUR=70,DB.STATIK
IM00    ANZAHL ZIELINFORMATIONEN: 19
*
DRUCKE,$,T,TOUR UMSATZ01 - UMSATZ04
```

Ausdruck:

TOUR	UMSATZ01	UMSATZ02	UMSATZ03	UMSATZ04
69	543245.00	432.00	3424.00	432432.00
66	545.00	4311.50	5600.56	4800.50
66	4564.12	4511.09	5121.90	4300.90
66	2364.12	4119.09	4980.50	5089.90
66	4334.66	5009.50	4500.50	5129.12
67	4390.16	4809.56	4500.50	5321.77
67	4590.55	4809.56	380.08	6001.65
67	-12.65	-112.00	-98.90	6001.65
67	20906.34	41812.68	62719.02	83625.36
68	129.86	254.97	380.08	-2909.55
68	19128.59	38257.18	57385.77	76514.36
68	22684.09	45368.18	68052.27	90736.36
68	545.00	254.97	380.08	505.19
69	21972.99	43945.98	65918.97	87891.96
69	21901.88	43803.76	65705.64	87607.52
69	21830.77	43661.54	65492.31	87323.08
65	900.55	254.97	380.08	505.19
69	18630.82	37261.64	55892.46	74523.28
69	166695.39	296582.74	55679.13	74238.84

*

```
DRUCKE,$,G,TOUR UMSATZ01 - UMSATZ04
```

Ausdruck:

```
TOUR    UMSATZ01   UMSATZ02   UMSATZ03   UMSATZ04

  65      900.55     254.97     380.08     505.19
  66    11807.90   17951.18   20203.46   19320.42
  67    29874.40   51319.80   67500.70  100950.43
  68    42487.54   84135.30  126198.20  164846.36
  69   794276.85  465687.66  312112.51  844016.68
*
```

```
DRUCKE,$,GT,TOUR UMSATZ01 - UMSATZ04
```

Ausdruck:

```
TOUR    UMSATZ01   UMSATZ02   UMSATZ03   UMSATZ04

  65      900.55     254.97     380.08     505.19
  66    11807.90   17951.18   20203.46   19320.42
  67    29874.40   51319.80   67500.70  100950.43
  68    42487.54   84135.30  126198.20  164846.36
  69   794276.85  465687.66  312112.51  844016.68
SUMME  879347.24  619348.91  526394.95 1129639.08
*
```

```
DRUCKE,$,GS,TOUR UMSATZ01 - UMSATZ04
```

Ausdruck:

```
TOUR    UMSATZ01   UMSATZ02   UMSATZ03   UMSATZ04      SUMME

  65      900.55     254.97     380.08     505.19    2040.79
  66    11807.90   17951.18   20203.46   19320.42   69282.96
  67    29874.40   51319.80   67500.70  100950.43  249645.33
  68    42487.54   84135.30  126198.20  164846.36  417667.40
  69   794276.85  465687.66  312112.51  844016.68 2416093.70
SUMME  879347.24  619348.91  526394.95 1129639.08 3154730.18
*
```

2.6.4 DRUCKE-T (tabellarisch)

Anwendungsart: aktiv/passiv

```
DRUCKE[,n]{,T,/_}{Feld[Erg)]/Var[Kom]}_...[,DB.xxxxxx]
```

(vgl. auch die vereinfachte Syntax der Operandenangabe - Seite 15)

DRUCKE	Operation für Ausgabe auf Drucker.
n	Auf Sätze bezogene Mengenangabe. $ = alle Sätze. Bei fehlender Mengenangabe wird '$' angenommen.
T	Operationsergänzung für tabellarisch.
Feld	Name des auszugebenden Feldes.
Ergänzung	AM = Abschnittsmultiplität: Bei fehlender AM-Angabe wird AM=1 angenommen. AZ = Abschnittszyklen: $ = alle Abschnittszyklen. Bei fehlender AZ-Angabe wird AZ=1 angenommen. FM = Feldmultiplität: Bei fehlender FM-Angabe wird FM=1 angenommen. FZ = Feldzyklen: $ = alle Feldzyklen. Bei fehlender FZ-Angabe wird FZ=1 angenommen. FP = Feldposition (nur bei Feldbedeutung T): Bei fehlender FP-Angabe wird FP=1 angenommen. FL = Feldlänge: Bei fehlender FL-Angabe wird die definierte Feldlänge verwendet. DB = Name der Satzbeschreibung: Die Angabe ist nur beim Arbeiten im Fenster zulässig.
Variable	Rechenvariable: (vgl. Seite 115) %X0 ... %X9, %XA ... %XZ %Y0 ... %Y9, %YA ... %YZ
Kommentar	Zur besseren Dokumentation kann der Rechenvariablen ein Kommentar angefügt werden (max. 19stellige alphanumerische Zeichenfolge - vgl. Seite 118).
xxxxxx	Satzbeschreibung

Die Ausgabe der im Kommando angegebenen Felder erfolgt tabellarisch auf dem Drucker.

Die Ausgabelänge der Felder entspricht der in der Datenbeschreibung definierten Feldlänge, wenn sie nicht durch den Ergänzungsparameter FL modifiziert wird. Durch geeignete Feldergänzungen kann auf bestimmte Multiplitäten Bezug genommen werden.

Beispiele:

```
*
DRUCKE,3,T,FIRMA(FL=6) WARENTEXT(AZ=$)
```

Folgende Ausgabe wird erzeugt:

```
FIRMA  WARENTEXT

SPRAEN SCHOKOLADE
       PRALINEN
       KAKAOPULVER              Satz mit 5 Wiederholabschnitten.
       KEKSE
       KAUGUMMI
AILDIS                          Satz ohne Wiederholabschnitt.
SAUER  SALZSTANGEN
       ESSIGGURKEN              Satz mit 2 Wiederholabschnitten.
```

```
*
DRUCKE,$,T,FIRMA WARENTEXT(AM=-2,AZ=$)
```

Folgende Ausgabe wird erzeugt:

```
FIRMA        WARENTEXT

SPRAENGEL    KEKSE
             KAUGUMMI
AILDIS
SAUER        SALZSTANGEN
             ESSIGGURKEN
*
```

Die Multiplitätsangabe kann auch negativ sein. In diesem Beispiel wird auf alle Multiplitäten ab der zweitletzten Bezug genommen. Dies gilt auch für die FM-Angaben.

2.6.5 DRUCKE-V (Verteilungstafel)

Anwendungsart: aktiv/passiv

```
DRUCKE[,n],{V/VS},Sortfeld[(Erg)]_Feld[(Erg)]
          {*/=/</>/:/#}Wert[,Feld...]...[,DB.xxxxxx]
```

DRUCKE	Operation für Ausgabe auf Drucker.
n	Mengenangabe, die sich auf Druckzeilen bezieht. $ = alle Druckzeilen. Bei fehlender Mengenangabe wird '$' angenommen.
V	Operationsergänzung für Verteilungstafel.
VS	Operationsergänzung für Verteilungstafel mit Summenbildung.
Sortfeld	Ordinatenfeld der Verteilungstafel.
Feld	Abszissenfelder der Verteilungstafel.
Ergänzung	Sortfeld: FP = Feldposition (nur bei Feldbedeutung T). AM = Abschnittsmultiplität: Bei fehlender AM-Angabe wird AM=1 angenommen. FM = Feldmultiplität: Bei fehlender FM-Angabe wird FM=1 angenommen. DB = Name der Satzbeschreibung: Nur beim Arbeiten im Fenster. Feld: FP = Feldposition (nur bei Feldbedeutung T). AM = Abschnittsmultiplität: Bei fehlender AM-Angabe wird AM=1 angenommen. AZ = Abschnittszyklen: $ = alle Abschnittszyklen. Bei fehlender AZ-Angabe wird AZ=1 genommen. FM = Feldmultiplität: Bei fehlender FM-Angabe wird FM=1 angenommen. FZ = Feldzyklen: $ = alle Feldzyklen. Bei fehlender FZ-Angabe wird FZ=1 angenommen. DB = Name der Satzbeschreibung: Nur beim Arbeiten im Fenster. GR = Gruppenbildung in Intervallen.
Wert	Gesuchte Feldinhalte der Abszissenfelder.
xxxxxx	Satzbeschreibung

Die Verteilungstafel gibt Aufschluß über die mengenmäßige Beziehung zwischen verschiedenen Feldinhalten.

Die Werte des Ordinatenfeldes (Sortfeld) werden aufsteigend sortiert ausgegeben. Für jeden Feldinhalt der Ordinate wird die Häufigkeit der angegebenen Werte der Abszissenfelder ermittelt. Bei der Operationsergänzung VS werden zusätzlich vertikal und horizontal die Gesamtsummen gebildet.

Beispiel:

```
*
SU ALTER=50,BIS ALTER=65
IM00    ANZAHL ZIELINFORMATIONEN: 155
*
DRUCKE,$,VS,ALTER TYP=AL,TYP=GL,TYP=SBA
```

folgende Ausgabe wird erzeugt:

```
ALTER TYP=AL TYP=GL TYP=SBA SUMME

   50    2      3      9       14
   51    1      1     17       19
   53    0      0     31       31
   54    0      2     24       26
   56    0      2      1        3
   58    1      0     18       19
   59    0      3      4        7
   60    1      1      3        5
   61    0      1     15       16
   62    0      2     12       14
   63    1      0      0        1
SUMME    6     15    134      155
```

Am Bildschirm wird die Zahl der verarbeiteten Sätze angezeigt:

```
IM01    ANZAHL ZIELINFORMATIONEN: 155
```

Die Verschlüsselungen im Feld "Typ" haben folgende Bedeutung:

AL =	Abteilungsleiter
GL =	Gruppenleiter
SBA =	Sachbearbeiter

Mit der Ergänzung GR=I wird im Trefferfall für den entsprechenden Satz die Verarbeitung abgebrochen. Auf diese Weise lassen sich bei entsprechender Anordnung im Kommando Intervalle bilden:

Beispiel: `DR,$,VS,ALTER GEHALT(GR=I):4000,GEHALT(GR=I):5000...`

```
ALTER  GEHALT:4000  GEHALT:5000  GEHALT:6000  GEHALT:10000  SUMME

   50        10            4            2            1         17
   51         0            1            0            0          1
   .
   .
   .
SUMME        10            5            2            1         18
```

2.6.6 DRUCKE-VB (Verweisdatei-Balkendiagramm)

Anwendungsart: aktiv/passiv

```
DRUCKE[,n],VB,Feld[(Erg)][*Wert][,DB.xxxxxx]
```

DRUCKE	Operation für Ausgabe auf Drucker.
n	Mengenangabe, die sich auf invertierte Feldinhalte (VD-Sätze) bezieht. $ = alle Feldinhalte. Bei fehlender Mengenangabe wird '$' angenommen.
VB	Operationsergänzung für Verweisdatei-Balkendiagramm.
Feld	Name des invertierten Feldes, auf das sich die Ausgabe bezieht.
Ergänzung (Vgl. Seite 18)	DB = Name der Satzbeschreibung: Nur beim Arbeiten im Fenster.
*	Vergleichsoperator für größer/gleich.
Wert	Invertierter Feldinhalt, ab dem die Ausgabe erfolgen soll.
xxxxxx	Satzbeschreibung

Dieses Kommando stellt die Belegung der Verweisdatei dar. Es wird die Häufigkeitsverteilung der Feldinhalte zu dem angegebenen Feld dokumentiert.

Die Häufigkeit eines Feldinhalts (Anzahl der Verweise in der Verweisdatei) wird auf zwei Arten dargestellt. Zum einen als absoluter Zahlenwert (in der Spalte ANZAHL) und zum anderen als Balkendiagramm. Ist die maximale Häufigkeit größer als 100, so wird im Balkendiagramm auf den Maximalwert normiert, d.h. die Balkenlänge zu dem jeweiligen Feldinhalt ist relativ zur maximalen Häufigkeit dargestellt (Prozentangabe).

Am Ende jeder Druckseite wird eine Zwischensumme ausgegeben. Sie enthält die Anzahl aller bisher dargestellten Verweise und die Anzahl aller bisher aufgeführten Feldinhalte. Analog dazu wird am Ende der Liste die Endsumme ausgegeben.

Die Ausgabe kann auch erst ab einem bestimmten Feldinhalt, der im Feld *Wert angegeben wird (*für größer/gleich), erfolgen.

Beispiel: Verweisdatei-Balkendiagramm

Mit dem CIS-Kommando `DRUCKE,$,VB,LAENDER-KEY*300` wird auf SYSLST folgender Ausdruck erzeugt:

```
                                30.09.93  11:20:12      SEITE    1
VD-STATISTIK : FELD=LAENDER-KEY   AUFGESETZT BEI 300
               100% ENTSPRECHEN DER MAXIMALEN VERWEISZAHL:    474
FELDINHALT ANZAHL               20%                   40%  ... 100%

302          4 XXXXXXXXXXXXXXXX  I                     I     ... I
306         51 XXXXXXXXXX        I                     I     ... I
310         41 XXXXXXXXX         I                     I     ... I
314         64 XXXXXXXXXXXXXX    I                     I     ... I
318         68 XXXXXXXXXXXXXX    I                     I     ... I
322        100 XXXXXXXXXXXXXXXXXXXX                    I     ... I
 .                                                                .
 .                                                                .
 .                                                                .
400        474 XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX...XX
404        323 XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX... I
412        185 XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXI     ... I
 .                                                                .
 .                                                                .
 .                                                                .
608        172 XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX   I     ... I

ZWISCHENSUMME: 5546 VERWEISE FUER 58 UNTERSCHIEDLICHE FELDINHALTE
                                        DES FELDES: LAENDER-KEY
```

Bemerkung: Aus drucktechnischen Gründen sind die 132 Zeichen breiten Druckzeilen verkürzt dargestellt. Außerdem wird nur ein Auszug pro Druckseite aufgeführt.

Erläuterung:

Auf dieser 1. Druckseite werden die Feldinhalte 302 bis 608 des Feldes LAENDER-KEY ausgegeben, durch die Angabe *300 im DRUCKE-Kommando werden alle Feldinhalte kleiner 300 ignoriert. Die Anzahl der auftretenden Feldinhalte sind absolut (Spalte ANZAHL) und relativ mit Hilfe der Balkendarstellung angegeben.

Der Feldinhalt 400 kommt am häufigsten vor, insgesamt 474 Mal. Er wird in der Balkendiagrammdarstellung als Bezugspunkt herangezogen. Alle Balken sind demnach relativ zum Feldinhalt 400 zu betrachten.

Am Ende der Druckseite wird eine Zwischensumme gebildet. Die Anzahl der Verweise und die Anzahl der unterschiedlichen Feldinhalte in der Zwischensumme beziehen sich auf alle bisher dargestellen Feldinhalte (inklusiv derer, die auf den vorherigen Druckseiten ausgegeben wurden).

```
                                    30.09.93  11:20:12    SEITE     2
VD-STATISTIK : FELD=LAENDER-KEY      AUFGESETZT BEI 300
               100% ENTSPRECHEN DER MAXIMALEN VERWEISZAHL:     474
FELDINHALT ANZAHL                 20%                 40%   ... 100%

612         170 XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX   I       ... I
616         298 XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX... I
624         285 XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX... I
628         136 XXXXXXXXXXXXXXXXXXXXXXXXXXXXX           I       ... I
632         227 XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX... I
636         200 XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX    ... I
640         105 XXXXXXXXXXXXXXXXXXXXX                 I       ... I
644           3 X                   I                 I       ... I
647          13 XXX                 I                 I       ... I
649           3 X                   I                 I       ... I
 .                                                               .
 .                                                               .
 .                                                               .
743          33 XXXXXXX             I                 I       ... I
800         301 XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX... I
801           2                     I                 I       ... I
804         175 XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX  I       ... I
809           3 X                   I                 I       ... I
822           2                     I                 I       ... I
989          14 XXX                 I                 I       ... I

E N D S U M M E: 9917 VERWEISE FUER 99 UNTERSCHIEDLICHE FELDINHALTE
                                              DES FELDES: LAENDER-KEY
```

Erläuterung:

Auf dieser 2. Druckseite sind die restlichen Feldinhalte des Feldes LAENDER-KEY aufgeführt.

Am Ende der Druckseite wird die Endsumme ausgegeben.

```
                                   30.09.93  11:20:12    SEITE     3
VD-STATISTIK : FELD=LAENDER-KEY      AUFGESETZT BEI 300
               100% ENTSPRECHEN DER MAXIMALEN VERWEISZAHL:     474

          H A E U F I G S T E  F E L D I N H A L T E

FELDINHALT ANZAHL                20%                 40%   ... 100%

400          474 XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX...XX
404          323 XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX... I
390          321 XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX... I
800          301 XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX... I
616          298 XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX... I
732          296 XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX... I
624          285 XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX... I
508          275 XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX... I
632          227 XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX... I
484          220 XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX... I
528          211 XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX ... I
636          200 XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX    ... I
664          194 XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX     ... I
412          185 XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXI      ... I
604          182 XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX I      ... I
740          179 XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX I      ... I
512          177 XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX  I      ... I
804          175 XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX  I      ... I
608          172 XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX   I      ... I
612          170 XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX   I      ... I
736          156 XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX      I      ... I
706          150 XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX      I      ... I
.                                                                              .
.                                                                              .
.                                                                              .
436           88 XXXXXXXXXXXXXXXXXXXI                    I      ... I
382           87 XXXXXXXXXXXXXXXXXXX I                   I      ... I
520           86 XXXXXXXXXXXXXXXXXXX I                   I      ... I
334           86 XXXXXXXXXXXXXXXXXXX I                   I      ... I
352           80 XXXXXXXXXXXXXXXXXX  I                   I      ... I
330           80 XXXXXXXXXXXXXXXXXX  I                   I      ... I
669           79 XXXXXXXXXXXXXXXXXX  I                   I      ... I
440           79 XXXXXXXXXXXXXXXXXX  I                   I      ... I
666           78 XXXXXXXXXXXXXXXXX   I                   I      ... I
428           78 XXXXXXXXXXXXXXXXX   I                   I      ... I
456           76 XXXXXXXXXXXXXXXXX   I                   I      ... I
366           76 XXXXXXXXXXXXXXXXX   I                   I      ... I
370           75 XXXXXXXXXXXXXXXXX   I                   I      ... I
660           74 XXXXXXXXXXXXXXXXX   I                   I      ... I
302           74 XXXXXXXXXXXXXXXXX   I                   I      ... I
DIE 55 HAEUFIGSTEN FELDINHALTE DECKEN MIT EINER SUMME VON
           8484 VERWEISEN     85.5% ALLER VERWEISEINTRAEGE
```

Erläuterung:

Die vorletzte Druckseite, die beim DRUCKE-VB erzeugt wird, gibt eine Übersicht über die häufigsten Feldinhalte.

Auf dieser 3. Druckseite sind die 55 häufigsten Feldinhalte aufgeführt. Entsprechend den vorherigen Darstellungen ist auch hier die Balkenlänge auf den Feldinhalt "400" bezogen.

```
                                    30.09.93  11:20:12    SEITE     4
S U M M E N H A E U F I G K E I T ANZ. VERSCHIEDENER FELDINHALTE=99

    SORTIERKLASSE  ANZAHL                 20           40 ...    100

VERWEISZAHL<=   1    7 XXXXXXX            I            I ...      I
VERWEISZAHL<=   2   11 XXXXXXXXXXX        I            I ...      I
VERWEISZAHL<=   3   16 XXXXXXXXXXXXXXXX   I            I ...      I
VERWEISZAHL<=   4   16 XXXXXXXXXXXXXXXX   I            I ...      I
VERWEISZAHL<=   5   16 XXXXXXXXXXXXXXXX   I            I ...      I
VERWEISZAHL<=   6   17 XXXXXXXXXXXXXXXXX  I            I ...      I
VERWEISZAHL<=   7   17 XXXXXXXXXXXXXXXXX  I            I ...      I
VERWEISZAHL<=   8   17 XXXXXXXXXXXXXXXXX  I            I ...      I
VERWEISZAHL<=   9   17 XXXXXXXXXXXXXXXXX  I            I ...      I
VERWEISZAHL<=  10   17 XXXXXXXXXXXXXXXXX  I            I ...      I
VERWEISZAHL<=  19   19 XXXXXXXXXXXXXXXXXXXI            I ...      I
VERWEISZAHL<=  28   19 XXXXXXXXXXXXXXXXXXXI            I ...      I
VERWEISZAHL<=  37   20 XXXXXXXXXXXXXXXXXXXX            I ...      I
VERWEISZAHL<=  47   26 XXXXXXXXXXXXXXXXXXXXXXXXXX     I ...      I
.                                                        .
.                                                        .
.                                                        .
VERWEISZAHL<= 455   98 XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX...XXXXX I
VERWEISZAHL<= 464   98 XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX...XXXXX I
VERWEISZAHL<= 474   99 XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX...XXXXXXI
```

Erläuterung:

Am Ende der statistischen Auswertung wird die Summenhäufigkeit der aufgeführten Feldinhalte gebildet. Dazu wird die Menge der aufgetretenen (absoluten) Häufigkeiten (Spalte ANZAHL in den vorherigen Druckseiten) in Klassen eingeteilt. Eine Klasse enthält die Anzahl aller Feldinhalte, die höchstens so viele Verweise besitzen, wie die Klasse angibt.

Die erste Zeile bedeutet also, daß insgesamt 7 unterschiedliche Feldinhalte auftreten, die einen Eintrag in der Verweisdatei aufweisen. Die zweite Zeile sagt aus, daß 11 unterschiedliche Feldinhalte auftreten, die einen oder zwei Einträge in der Verweisdatei besitzen. Dementsprechend zeigt die letzte Zeile der Summenhäufigkeit, daß 99 (also alle) unterschiedliche Feldinhalte maximal 474 Einträge in der Verweisdatei besitzen.

Aus dem Vergleich zweier Sortierklassen läßt sich die Anzahl der Feldinhalte ermitteln, deren Verweisdateieinträge in einem bestimmten Intervall liegen. Betrachten wir beispielsweise die beiden Sortierklassen mit der Verweiszahl ≤ 10 und der Verweiszahl ≤ 37. Da 17 Feldinhalte höchstens 10 Verweise und 20 Feldinhalte höchstens 37 Verweise besitzen, treten 20 minus 17 also 3 Feldinhalte auf, die mindestens 11 und höchstens 37 Verweise aufweisen.

Insbesondere zeigt ein Vergleich der beiden letzten Sortierklassen, daß es genau einen Feldinhalt mit 474 Verweisen gibt.

Das Balkendiagramm, das die Werte in der Spalte ANZAHL verdeutlicht, zeigt in diesem Beispiel den absoluten Wert der Anzahl der Feldinhalte einer Sortierklasse. Erst bei einer Anzahl die größer als 100 ist wird wieder auf den größten auftretenden Wert normiert.

2.6.7 DRUCKE-SB (Statistik-Balkendiagramm der VD)

Anwendungsart: aktiv/passiv

```
DRUCKE[,n],SB,Feld[(Erg)][*Wert][,DB.xxxxxx]
```

DRUCKE	Operation für Ausgabe auf Drucker.
n	Mengenangabe, die sich auf invertierte Feldinhalte (VD-Sätze) bezieht. $ = alle Feldinhalte. Bei fehlender Mengenangabe wird '$' angenommen.
SB	Operationsergänzung für Statistik-Balkendiagramm der Verweisdatei.
Feld	Name des invertierten Feldes, auf das sich die Ausgabe bezieht.
Ergänzung (vgl. Seite 18)	DB = Name der Satzbeschreibung: Nur beim Arbeiten im Fenster.
*	Vergleichsoperator für größer/gleich.
Wert	Invertierter Feldinhalt, ab dem die Ausgabe erfolgen soll.
xxxxxx	Satzbeschreibung

Dieses Kommando arbeitet wie das DRUCKE-VB (vgl. Seite 60). Es werden aber nur die Balkendiagramme der häufigsten Feldinhalte und die Summenhäufigkeit ausgedruckt.

2.6.8 DRUCKE-Z (zeilenweise)

Anwendungsart: aktiv/passiv

```
DRUCKE[,n],Z,{Feld[(Erg)]/Var[Kom]}_...[,DB.xxxxxx]
```

(vgl. auch die vereinfachte Syntax der Operandenangabe - Seite 15)

DRUCKE	Operation für Ausgabe auf Drucker.
n	Auf Sätze bezogene Mengenangabe. $ = alle Sätze. Bei fehlender Mengenangabe wird '$' angenommen.
Z	Operationsergänzung für zeilenweise.
Feld	Name des auszugebenden Feldes.
Ergänzung	AM = Abschnittsmultiplität: Bei fehlender AM-Angabe wird AM=1 angenommen. AZ = Abschnittszyklen: $ = alle Abschnittszyklen. Bei fehlender AZ-Angabe wird AZ=1 angenommen. FM = Feldmultiplität: Bei fehlender FM-Angabe wird FM=1 angenommen. FZ = Feldzyklen: $ = alle Feldzyklen. Bei fehlender FZ-Angabe wird FZ=1 angenommen. FP = Feldposition (nur bei Feldbedeutung T): Bei fehlender FP-Angabe wird FP=1 angenommen. FL = Feldlänge: Bei fehlender FL-Angabe wird die definierte Feldlänge verwendet. DB = Name der Satzbeschreibung: Die Angabe ist nur beim Arbeiten im Fenster zulässig.
Variable	Rechenvariable: (vgl. Seite 115) %X0 ... %X9, %XA ... %XZ %Y0 ... %Y9, %YA ... %YZ
Kommentar	Zur besseren Dokumentation kann der Rechenvariablen ein Kommentar angefügt werden (max. 19stellige alphanumerische Zeichenfolge) - vgl. Seite 118.
xxxxxx	Satzbeschreibung

In der Ausgabe wird für jede angegebene Feldbezeichnung eine eigene Druckzeile erzeugt.

Beispiel:

```
*
DRUCKE,2,Z,FIRMA WARENTEXT(AM=-2,AZ=$)
```

folgende Ausgabe wird erzeugt:

```
FIRMA                SPRAENGEL

WARENTEXT 005/000    KAUGUMMI
WARENTEXT 006/000    KEKS
FIRMA                SUCHARD

WARENTEXT 003/000    PRALINE
WARENTEXT 004/000    SCHOKOLADE
```

Abschnitts-/Feldmultiplität

2.7 EINGEBEN-Kommando

Das Kommando kann nur mit einer Bildschirmmaske angewendet werden. Die auf dem Bildschirm erscheinende leere Maske wird mit entsprechenden Daten ausgefüllt und als CIS-Satz bzw. CIS-Abschnitt abgespeichert. Eventuell vorhandene Folgemasken werden nur beim Kommando EINGEBEN-K (Satz) nicht jedoch bei EINGEBEN-A (Abschnitt) automatisch aufgerufen.

Je nach Kommando werden am Ende des Bildschirms mehrere Möglichkeiten der Weiterverarbeitung angeboten. Durch Eingabe eines beliebigen Zeichens ungleich Space an der entsprechenden Position wird entweder der Eingabemodus beendet oder entsprechend fortgesetzt.

Eine der Positionen ist standardmäßig mit X vorbesetzt. Soll die vorbesetzte Funktion ausgeführt werden, so genügt es, das Formular einfach abzuspeichern. Soll eine andere Funktion ausgeführt werden, so muß die Vorbesetzung durch Space überschrieben und die gewünschte Position angekreuzt werden.

Werden versehentlich mehrere Positionen belegt, so wird das bereits ausgefüllte Formular nochmals gezeigt und gleichzeitig als letzte Zeile die Meldung EU04 FALSCH ANGEKREUZT ausgegeben. Das Formular kann jetzt korrigiert und gespeichert werden.

Bei der Eingabe numerisch definierter Felder sind die Hinweise im Kapitel "Behandlung numerischer Feldinhalte" (Seite 21 ff) zu beachten.

2.7.1 EINGEBEN-A (Abschnitt)

Anwendungsart: aktiv

```
EINGEBEN[,n],A[,DB.xxxxxx]
```

EINGEBEN	Operation für Eingeben.
n	Auf Sätze bezogene Mengenangabe. Bei fehlender Mengenangabe wird '$' (= alle) angenommen.
A	Operationsergänzung für Abschnitt.
xxxxxx	Bildschirmmaske

An einem bestehenden Satz (MV-Format) können neue Abschnitte angefügt werden. Über den ZPL-Zeiger wird der Satz lokalisiert. Ist keine ZPL vorhanden, so wird der FCB-Zeiger benützt.

Die in der Maske angezeigten Felder müssen alle zu einem Abschnitt gehören. Es kann mit keiner Folgemaske gearbeitet werden.

Am Ende der Maske werden vier Auswahlmöglichkeiten angezeigt:

BLAETTERN SATZ?	Der aktuelle Satz wird in der Haupt- bzw. Verweisdatei aktualisiert, der nächste Satz der Zielpunktliste wird gelesen und die leere Maske erscheint wieder auf dem Bildschirm.
WEITER ABSCHNITT?	Ein weiterer Abschnitt soll an den aktuellen Satz angefügt werden, ein Update erfolgt für den gerade eingegebenen Abschnitt.
BEENDEN?	Der aktuelle Satz wird in der Haupt- bzw. Verweisdatei aktualisiert, der Eingabemodus wird beendet.
ABBRUCH?	Der Eingabemodus wird abgebrochen. Es erfolgt kein Update für den aktuellen Abschnitt (Abschnitt wird nicht angelegt).

Wird BLAETTERN SATZ angekreuzt und ist das Ende der Zielpunktliste erreicht, so wird die Meldung EU03 KEIN SATZ (MEHR) VERFUEGBAR ausgegeben.

Mit der Mengenangabe n läßt sich die maximale Anzahl von Sätzen angeben, an die Abschnitte hinzugefügt werden sollen. Ist beispielsweise n=3 und die Anzahl der Zielinformationen (AZI) 10, so kann BLAETTERN SATZ nur 2 mal angekreuzt werden. Die restlichen 7 Sätze der ZPL bleiben unverändert. Bei fehlender Mengenangabe ist n=$ gesetzt, d.h. die gesamte ZPL kann abgearbeitet werden.

2.7.2 EINGEBEN-K (Eingeben Satz)

Anwendungsart: aktiv

```
EINGEBEN[,n],K[,DB.xxxxxx]
```

EINGEBEN	Operation für Eingeben.
n	Auf Sätze bezogene Mengenangabe (vgl. Seite 69). Bei fehlender Mengenangabe wird '$' (= alle) angenommen.
K	Operationsergänzung für Satz.
xxxxxx	Bildschirmmaske

Am Ende der jeweils letzten Bildschirmmaske werden drei Auswahlmöglichkeiten angezeigt:

FORTSETZEN?	Der neue Satz wird in die Haupt- bzw. Verweisdatei geschrieben, der nächste Satz kann eingegeben werden.
BEENDEN?	Der neue Satz wird gespeichert, der Eingabemodus wird beendet.
ABBRUCH?	Der Eingabemodus wird abgebrochen, der Satz wird nicht angelegt.

2.8 ENDE-Kommando

Anwendungsart: aktiv/passiv

```
ENDE[,DB.xxxxxx]
```

ENDE	Operation für Ende Textkommentar.
xxxxxx	Paßwort einer Datenbeschreibung.

Vor einem Listenausdruck kann zur Erläuterung ein Kommentar ausgegeben werden (vgl. TEXT-Kommando Seite 169). Mit dem ENDE-Kommando wird das Ende des Kommentars definiert (der Endebalken des Druck-Titelblattes wird in die SYSLST-Datei geschrieben). Aus der angegebenen, oder der zu diesem Zeitpunkt gültigen Datenbeschreibung wird für die Textaufbereitung ein eventuell angegebener FORM-Parameter im A-Segment (siehe Manual-2: CISGEN, A-Segment) ausgewertet.

Beispiel:

```
*
TEXT DIE FOLGENDEN DRUCKLISTEN
TEXT WURDEN MIT CIS ERSTELLT
ENDE
DRUCKE,100,T,NAME VORNAME WOHNORT TELEFON
*
```

Die erste Druckseite wird wie folgt aufgebaut: (verkürzte Darstellung)

```
CIS CIS CIS CIS CIS CIS CIS CIS CIS CIS CIS CIS CIS CIS CIS CIS CIS
 CIS CIS CIS CIS CIS CIS CIS CIS CIS CIS CIS CIS CIS CIS CIS CIS CI


DIE FOLGENDEN DRUCKLISTEN
WURDEN MIT CIS ERSTELLT


CIS CIS CIS CIS CIS CIS CIS CIS CIS CIS CIS CIS CIS CIS CIS CIS CIS
 CIS CIS CIS CIS CIS CIS CIS CIS CIS CIS CIS CIS CIS CIS CIS CIS CI
```

Auf den folgenden Druckseiten werden die angegebenen Feldinhalte tabellarisch ausgegeben.

2.9 EXIT-Kommando

2.9.1 EXIT-A/TA (ADILOS-Unterprogrammanschluß)

Anwendungsart: aktiv/passiv

```
EXIT,n,{A/TA},SYSTEM_EINGABE=Steuerdatei[,DB.xxxxxx]
```

EXIT	Operation für Unterprogrammanschluß.
n	Auf Sätze bezogene Mengenangabe. $ = alle Sätze.
A	Operationsergänzung für ADILOS.
TA	Operationsergänzung für TEST-ADILOS. Die Steuerdatei wird von ADILOS am Bildschirm ausgegeben.

Steuerdatei	Name einer DVS-Datei, die die Steueranweisungen entsprechend den ADILOS-Konventionen enthält. Die Steuerdatei muß folgende LESE-Anweisung enthalten:
	`LESEN EINGABE=UPROG`
	`UPROG PROGRAM=CISADILO`

xxxxxx	Satz- oder Transformationsbeschreibung.

ADILOS wird als Unterprogramm geladen. Der Name der Bibliothek, die die ADILOS-Module enthält, wird in CISVARI+X'251' mit der Länge 54 eingetragen. Fehlt der Eintrag (Space), so kann die Bibliothek mit

```
/SET-FILE-LINK FILE-NAME=Bibliotheksname,LINK-NAME=BLSLIB
```

zugewiesen werden.

2.9.2 EXIT-L (Listen-Unterprogrammanschluß)

Anwendungsart: aktiv/passiv

```
EXIT,n,L,ILP=Programmname[,DB.xxxxxx]
```

EXIT	Operation für Unterprogrammanschluß.
n	Auf Sätze bezogene Mengenangabe. $ = alle Sätze.
L	Operationsergänzung für CIS-Listenprogramm.
Programmname	ENTRY des zu ladenden Auswertungsprogramms.
xxxxxx	Satz- oder Transformationsbeschreibung.

Mit Hilfe des COBOL-REPORT Teils können auf einfache Weise Listen erzeugt werden. Diese Listenprogramme werden von CIS beim erstmaligen Aufruf über den ENTRY-Namen dynamisch gebunden. CIS übergibt dem geladenen Listen- bzw. Auswertungsprogramm jeweils einen Satz. Insgesamt werden soviele Sätze übergeben, wie in der Mengenangabe bestimmt wurde. Jeder Satz wird entweder so übergeben, wie er aus der Datenbank von CIS gelesen wurde, oder er kann in eine für das Auswertungsprogramm geeignete Form transformiert werden (Transformationsbeschreibung). Das Auswertungsprogramm benötigt keinen Dateizugriff für die auszuwertenden Daten. Die für die jeweilige Auswertung gewünschten Daten können durch geeignete Suchfragen gefunden werden. Soll die Übergabe sortiert erfolgen, kann vorher mit CIS sortiert werden. (siehe auch Manual-4: Programmschnittstelle ILP-Anschluß)

Beispiel:

```
*
SUCHE BRANCHE=MED,DB.FIRMDA
IM00    ANZAHL ZIELINFORMATIONEN: 28
*
SORT FIRMA
IM01    ANZAHL ZIELINFORMATIONEN: 28
*
EXIT,$,L,ILP=FIRMLIST,DB.FITRAN
*
```

2.10 FREIGEBEN-Kommando

2.10.1 FREIGEBEN-E/V (temporär/permanent gesicherte ZPL)

Anwendungsart: aktiv/passiv

```
FREIGEBEN{_/,{E/V},}Feld[(Erg)]=Wert[,DB.xxxxxx]
```

FREIGEBEN	Operation für Freigeben.
E	Operationsergänzung für temporäre Datei (vgl. Seite 129) - Standardwert.
V	Operationsergänzung für Verweisdatei (vgl. Seite 134).
Ergänzung (vgl. Seite 18)	DB = Name der Satzbeschreibung: Nur beim Arbeiten im Fenster.
Feld=Wert	Zur Bestimmung der gesicherten Zielpunktliste.
xxxxxx	Satzbeschreibung

Sollen temporär gesicherte Zielpunktlisten freigegeben werden, so ist das Kommando ohne Operationsergänzung oder mit Operationsergänzung E zu verwenden. Dabei bestimmen Feldbezeichnung und Feldinhalt die freizugebende Zielpunktliste.

Bei Ende des CIS-Laufs (Halt-Kommando bzw. CLOSE-A-Aufruf), werden alle temporär gesicherten Zielpunktlisten von CIS automatisch freigegeben.

Zum Löschen einer Zielpunktliste, die unter Verwendung eines Pseudofeldes in der Verweisdatei gesichert wurde, ist das Kommando mit Operationsergänzung V zu verwenden. Dabei bestimmen Feldbezeichnung und Feldinhalt die freizugebende Zielpunktliste.

Beispiele:

Freigeben (E)

```
*
FREIGEBEN GEHALT=3
*
AKTIVIEREN GEHALT=3
ZP13    ZPL NICHT GESICHERT
*
```

Freigeben (V)

```
*
FREIGEBEN,V,GEHALTBER=2000
*
SUCHE GEHALTBER=2000
IM00    ANZAHL ZIELINFORMATIONEN: 0
```

2.10.2 FREIGEBEN-I (temporär gesicherter VD-Zeiger)

Anwendungsart: aktiv/passiv

```
FREIGEBEN,I,Feld[(Erg)]=Wert[,DB.xxxxxx]
```

FREIGEBEN	Operation für Freigeben.
I	Operationsergänzung für VD-Zeiger (vgl. Seite 131).
Ergänzung (vgl. Seite 18)	DB = Name der Satzbeschreibung: Nur beim Arbeiten im Fenster.
Feld=Wert	Zur Bestimmung der gesicherten Zielpunktliste.
xxxxxx	Satzbeschreibung

Ein temporär gesicherter VD-Zeiger (vgl. Seite 131) wird freigegeben.

2.10.3 FREIGEBEN-J (ZPL für VERBINDE)

Anwendungsart: aktiv/passiv

```
FREIGEBEN,J,Feld[,DB.xxxxxx]
```

FREIGEBEN	Operation für Freigeben.
J	Operationsergänzung für ZPL zum VERBINDE
Feld	Zur Bestimmung der gesicherten Zielpunktliste.
xxxxxx	Satzbeschreibung

Die Zielpunktliste, die zum Verbinden bereitgestellt war, wird freigegeben. FR,J ist im Fenster nicht erlaubt, es muß ein Dateipaßwort gültig sein (vgl. Seite 132).

2.10.4 FREIGEBEN-S (Suchergebnis)

Anwendungsart: aktiv/passiv

```
FREIGEBEN,S[,DB.xxxxxx]
```

FREIGEBEN	Operation für Freigeben.
S	Operationsergänzung für Suchergebnis (vgl. Seite 133).
xxxxxx	Satzbeschreibung

Diese Kommando hebt das SICHERN-S Kommando (vgl. Seite 133) auf und gibt alle Suchergebnisse (Zielpunktlisten) frei, die nach dem SICHERN-S Kommando gesichert wurden. Damit ist auf diese Zielpunktlisten kein Zugriff mehr möglich.

2.11 GET-Kommando

Alle Kommandos im GET-Modus, bis auf GET-F, das nur in Kurzkommandos verwendet werden darf, und GET-FAA und GET-KF, die auch in Kurzkommandos angewendet werden können, sind nur in Anwenderprogrammen anwendbar. Sie übergeben dem aufrufenden Programm Daten, die aus der Hauptdatei oder der Datenbeschreibung stammen können.

Der Zugriff auf Daten der Hauptdatei kann - im Sinne von CIS - direkt durch Angabe des Schlüssels des Hauptdateisatzes, indirekt durch Gewinnung des Schlüssels des Hauptdateisatzes aus der vorliegenden Zielpunktliste bzw. aus dem Verweisdateieintrag, oder sequentiell, wenn keine Zielpunktliste vorliegt und der Schlüssel der Hauptdatei nicht angegeben ist, erfolgen.

2.11.1 GET-A (Abschnitt)

Anwendungsart: passiv

```
GET,A[S],Abschnittsname[(Erg)][,DB.xxxxxx]
```

GET	Operation für GET.
A	Operationsergänzung für Abschnitt.
AS	Mit Sperre.
Abschnittsname	Vierstellige Bezeichnung des Abschnitts.
Ergänzung (vgl. Seite 18)	AM = Abschnittsmultiplität. Bei fehlender AM-Angabe wird AM=1 angenommen.
xxxxxx	Satzbeschreibung

Dem aufrufenden Programm wird aus dem lokalisierten Satz der durch den Abschnittsnamen bezeichnete Abschnitt so übergeben, wie er in der Datenbank gespeichert ist (mit Abschnittslänge und Abschnittsart).

Bei der zusätzlichen Operationsergänzung AS wird der lokalisierte Satz gesperrt.

Neben dem Abschnittsnamen kann als weiteres Auswahlkriterium eine Multiplität angegeben werden. Die Multiplitätsangabe bewirkt, daß im Falle von Wiederholabschnitten genau dieser angegebene Abschnitt zur Verfügung gestellt wird. Der erste GET-Aufruf liest den lokalisierten Satz aus der Hauptdatei, alle weiteren GET-A Aufrufe beziehen sich auf diesen Satz. Ein Zugriff auf den nächsten Satz ist erst nach BLAETTERN,1,V bzw. mit GET-KN möglich.

2.11.2 GET-B (Satzerfassung über Bildschirm)

Anwendungsart: passiv

```
GET,B[,DB.xxxxxx]
```

GET	Operation für GET.
B	Operationsergänzung für Satzanlage über Bildschirm.
xxxxxx	Bildschirmmaske

Dieses Kommando gibt eine leere Bildschirmmaske auf das Datensichtgerät aus und liest nach Ausfüllen dieser Maske die Daten entsprechend der Satzbeschreibung in den vom aufrufenden Programm definierten Bereich, d.h. mit einem Feld für Satzlänge und evtl. Feldern für Abschnittslänge/Abschnittsart. Das bedeutet, daß die Felder entsprechend der Feldbedeutung aufbereitet und entsprechend der Feldadresse in der beschriebenen Länge an das Anwenderprogramm übergeben werden. Die Reihenfolge der Felder in der Bildschirmmaske muß nicht der Reihenfolge in der Satzbeschreibung entsprechen.

2.11.3 GET-F (Feld)

Anwendungsart: passiv (und aktiv über CISKURZ)

```
GET,F[S],Feld[(Erg)][,DB.xxxxxx]
```

GET	Operation für GET.
F	Operationsergänzung für Feld.
FS	Mit Sperre.
Feld	Feldbezeichnung
Ergänzung (vgl. Seite 18)	AM = Abschnittsmultiplität: Bei fehlender AM-Angabe wird AM=1 angenommen. FM = Feldmultiplität: Bei fehlender FM-Angabe wird FM=1 angenommen. DB = Name der Satzbeschreibung: Nur beim Arbeiten im Fenster.
xxxxxx	Satzbeschreibung

Dem aufrufenden Programm wird aus dem lokalisierten Satz das angesprochene Feld übergeben. Neben dem Feldnamen kann als zusätzliches Auswahlkriterium eine Abschnittsmultiplität und/oder eine Feldmultiplität angegeben werden. Dabei gibt es folgende Möglichkeiten:

- Das n-te Wiederholfeld im V-Satz.
- Das Feld im n-ten Wiederholabschnitt des MV-Satzes.
- Das n-te Wiederholfeld im festen Abschnitt des MV-Satzes.
- Das n-te Wiederholfeld im n-ten Wiederholabschnitt des MV-Satzes.

Bei der zusätzlichen Operationsergänzung FS wird der lokalisierte Satz gesperrt.

Das Feld wird so übergeben, wie es in der Datenbank gespeichert ist. Auf den nächsten Satz wird mit BL,1,V geschaltet.

2.11.4 GET-FAA (Alle Feldbeschreibungen mit Trennern)

Anwendungsart: aktiv

```
GET,FAA[,DB.xxxxxx]
```

GET	Operation für GET.
FAA	Operationsergänzung für Feldbeschreibungen mit Trennern.
xxxxxx	Satzbeschreibung

Ausgabe der D/E Segmente der Satzbeschreibung mit Zwischen- und Endetrennern (vgl. SETZE,{TE/TZ} auf Seite 127)

Beispiel mit verbundenen Satzbeschreibungen FMULTI und MESSKI:

```
ARTIKEL;BCDE;W;9;012;;12;12;;T;;;;FMULTI;«
AUSZEICHNUNG;;;77;AUS;0;1;1;;T;;;;MESSKI;«
BANKLEITZAHL;ZZZZ;W;19;;;12;12;W;T;;;;FMULTI;«
BEZEICHNUNG;BCDE;W;21;015;;34;34;W;T;;;;FMULTI;«
KONTONR;ZZZZ;W;9;;;10;10;;T;;;;FMULTI;«
NAME;ABCD;;12;001;S;30;30;0;T;;;;FMULTI;«
NAME;;;13;NAM;S;30;30;;T;;;;MESSKI;«
NATIONALITAET;;;91;NAT;0;15;15;;T;;;;MESSKI;«
ORT;ABCD;;46;003;;24;24;;T;;;;FMULTI;«
```

Der Endetrenner wurde in diesem Beispiel vom Standardwert C':' auf X'15' (Zeilenvorschub) gesetzt, um eine übersichtliche Darstellungsform zu erhalten.

Bedeutung der Felder:

Die einzelnen Felder werden durch ein Semikolon voneinander getrennt.

Feld 1	Feldbezeichnung
Feld 2	Abschnittsart (fehlt bei V-Format)
Feld 3	Wiederholzeichen (W) für Wiederholabschnitt.
Feld 4	Feldposition
Feld 5	Invertierkennzeichen (dreistellige Kurzkennung)
Feld 6	Sonderinvertierung (0, S, D, G, K)
Feld 7	Feldlänge wie in Satzbeschreibung definiert.
Feld 8	Feldlänge extern, abhängig vom Feldformat.
Feld 9	Logische Feldbedeutung (O, W, P, V...)
Feld 10	Feldformat (P, H, B, R, L, A, T)
Feld 11	Anzahl Kommastellen.
Feld 12	OCCURS-Angabe bei Feldgruppen.
Feld 13	INCREMENT-Angabe bei Feldgruppen.
Feld 14	Name der Datenbeschreibung.

2.11.5 GET-FBA (Feldbeschreibung mit Trennern)

Anwendungsart: aktiv

```
GET,FBA[,Feld[(Erg)]][,DB.xxxxxx]
```

GET	Operation für GET.
FBA	Operationsergänzung für Ausgabe einer Feldbeschreibung mit Trennern wie bei GET,FAA (vgl. Seite 81).
Feld	Name des Feldes, dessen Beschreibung zur Verfügung gestellt werden soll.
Ergänzung (vgl. Seite 18)	DB = Name der Satzbeschreibung. Nur beim Arbeiten im Fenster.
xxxxxx	Satz- oder Transformationsbeschreibung

Ausgabe der D/E Segmente des angegebenen Feldes. Wird das Feld nicht angegeben, so wird jeweils das nächste Feld der Satzbeschreibung ausgegeben.

2.11.6 GET-FEA (Erste Feldbeschreibung mit Trennern)

Anwendungsart: aktiv

```
GET,FEA [,DB.xxxxxx]
```

GET	Operation für GET.
FEA	Operationsergänzung für Ausgabe der ersten Feldbeschreibung mit Trennern wie bei GET,FAA (vgl. Seite 81).
xxxxxx	Satz- oder Transformationsbeschreibung.

Ausgabe der D/E Segmente des ersten Feldes der Satzbeschreibung.

2.11.7 GET-FB (Feldbeschreibung)

Anwendungsart: passiv

```
GET,FB[,Feld[(Erg)]][,DB.xxxxxx]
```

GET	Operation für GET.
FB	Operationsergänzung für Feldbeschreibung.
Feld	Name des Feldes, dessen Beschreibung zur Verfügung gestellt werden soll.
Ergänzung (vgl. Seite 18)	DB = Name der Satzbeschreibung. Nur beim Arbeiten im Fenster.
xxxxxx	Satz- oder Transformationsbeschreibung

Dem aufrufenden Programm wird aus der angegebenen Datenbeschreibung eine Feldbeschreibung übergeben. Es handelt sich dabei um einen Satz fester Länge (immer 42 Bytes). Der Aufbau des Satzes ist auf der nächsten Seite dargestellt.

Wurde im Kommando eine Feldbezeichnung angegeben, so wird die Feldbeschreibung genau dieses Feldes zur Verfügung gestellt. Ohne Angabe einer Feldbezeichnung erfolgt die Bereitstellung der Felder einzeln in Reihenfolge der Felder in der Datenbeschreibung. D.h. für jeden Kommandoaufruf wird die Feldbeschreibung nur eines Feldes zur Verfügung gestellt, beim nächsten Aufruf, die Feldbeschreibung des nächsten Feldes der Datenbeschreibung. Ist kein Feld mehr vorhanden, so wird mit FB01 quittiert.

Aufbau des übergebenen Satzes:

Assembler:	COBOL:			
DS H	PIC S9(4) COMP.		Länge der Feldbezeichnung	(2 Bytes)
DS CL16	PIC X(16).		Feldbezeichnung	(16 Bytes)
DS H	PIC S9(4) COMP.		Relative Feldadresse	(2 Bytes)
DS H	PIC S9(4) COMP.		Feldlänge	(2 Bytes)
DS CL2	PIC XX.	1)	Feldbedeutung	(2 Bytes)
DS CL4	PIC XXXX.	2)	Abschnittsart	(4 Bytes)
DS CL4	PIC XXXX.	3)	Segmentname	(4 Bytes)

Zusätzlich bei Satzbeschreibungen:

DS CL1	PIC X.		'W' aus Wiederholabschnitt	(1 Byte)
DS H	PIC S9(4) COMP.		Anzahl OCCURS WF-Gr *)	(2 Bytes)
DS CL1	PIC X.	4)	Versatz/Länge WF-Gr *)	(1 Byte)
DS CL6	PIC X(6).		Datenbeschreibung	(6 Bytes)

oder zusätzlich bei Transformationsbeschreibungen:

DS CL4	PIC XXXX.	Abschnittsart	(4 Bytes)
DS H	PIC S9(4) COMP.	Relative Feldadresse	(2 Bytes)
DS H	PIC S9(4) COMP.	Länge Transformation	(2 Bytes)
DS CL2	PIC XX.	Übertragungsvorschrift	(2 Bytes)

1) Feldinhalt: 1.Byte: Siehe D-Segment der Satzbeschreibung - Byte 15 (O,W,V,P,U).
2.Byte: 1.Halbbyte: X'0' - X'F' (Dezimalstelle bei P und R)
2.Halbbyte: X'2' für P
X'3' für H
X'4' für B
X'8' für R
X'9' für L
X'C' für A
X'D' für T

2) Feldinhalt: Bei V-Format: X'00000000'.

3) Feldinhalt: Wenn nicht invertiert: X'00000000'.

4) Dieses Feld muß in ein Binärfeld umgewandelt werden.

*) WF-GR: Wiederholfeldgruppe

2.11.8 GET-FE (Erste Feldbeschreibung)

Anwendungsart: passiv

```
GET,FE [,DB.xxxxxx]
```

GET	Operation für GET.
FE	Operationsergänzung für erste Feldbeschreibung.
xxxxxx	Satz- oder Transformationsbeschreibung.

Die Wirkung dieses Kommandos entspricht dem GET-FB Kommando (vgl. Seite 84), mit dem Unterschied, daß dem aufrufenden Programm die erste Feldbeschreibung der Datenbeschreibung übergeben wird.

2.11.9 GET-KD (Satz direkt)

Anwendungsart: passiv

```
GET,KD[S],Key{=/</>/:/*}Wert[,DB.xxxxxx]
```

GET	Operation für GET.
KD KDS	Operationsergänzung für Satz direkt. Mit Sperre
Key	Das als Ordnungsbegriff gekennzeichnete Feld der Datei.
Operator	= gleich < nächst kleinere > nächst größere : gleich oder der nächst kleinere * gleich oder der nächst größere
Wert	Schlüssel der Hauptdatei.
xxxxxx	Satzbeschreibung

Das Kommando GET,KD übergibt dem aufrufenden Programm **einen** Satz aus einer Hauptdatei durch direkten Zugriff über den Ordnungsbegriff.

Eine eventuell vorhandene ZPL wird nicht verändert oder zerstört.

Für die Datenübertragung gelten die gleichen Regeln wie bei GET-KP (vgl. Seite 92).

Die zusätzliche Operationsergänzung S sperrt den lokalisierten Satz. Die Operationsergänzung KS (gültig bis CIS-Version 10) wird noch unterstützt.

Ein Zugriff auf den nächsten Satz ist nach BL,1,V oder direkt mit GET-KN (vgl. Seite 91) möglich. Analoge Kommandos gelten für das Rückwärtslesen.

Ist eine Zielpunktliste vorhanden, so wird über den ZPL-Zeiger der nächste Satz gelesen. Soll der nächste Satz mit dem FCB-Zeiger (Satz mit dem nächst höheren Ordnungsbegriff) gelesen werden, so darf keine Zielpunktliste gültig sein. Sie muß vor dem GET-KD[S] mit dem Kommando IGNORIERE-Z (vgl. Seite 99) gelöscht werden. Dabei ist zu beachten, daß der DVS-Zeiger taskbezogen ist (vgl. Seite 8).

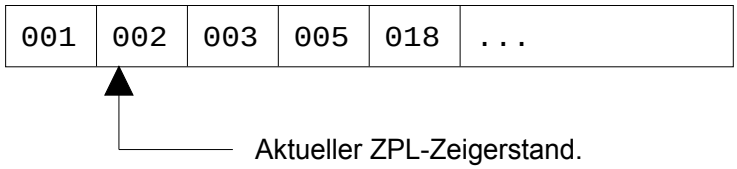
Beispiel:

1) Ohne Zielpunktliste:

In der Hauptdatei seien die Sätze mit den Ordnungsbegriffen ...100, 101, 102, 110, 111... vorhanden.

`GET,KD,OB=100`	Der Satz mit dem Ordnungsbegriff 100 wird zur Verfügung gestellt. Der FCB-Zeiger ist gültig positioniert (vgl. Seite 8).
`GET,KN`	Der Satz mit dem Ordnungsbegriff 101 wird zur Verfügung gestellt.
`GET,KN`	Der Satz mit dem Ordnungsbegriff 102 wird zur Verfügung gestellt.
`GET,KN`	Der Satz mit dem Ordnungsbegriff 110 wird zur Verfügung gestellt.
`GET,KD,OB>102`	Der Satz mit dem Ordnungsbegriff 110 wird zur Verfügung gestellt.
`GET,KN`	Der Satz mit dem Ordnungsbegriff 111 wird zur Verfügung gestellt.

2) Mit bestehender Zielpunktliste:



`GET,KD,OB=100`	Der Satz mit dem Ordnungsbegriff 100 wird zur Verfügung gestellt. Der ZPL-Zeiger hat Vorrang (vgl. Seite 8).
`GET,KN`	Der Satz mit dem Ordnungsbegriff 003 wird zur Verfügung gestellt.
`GET,KN`	Der Satz mit dem Ordnungsbegriff 005 wird zur Verfügung gestellt.

Soll GET-KD innerhalb einer Transaktion (vgl. Seite 171) ausgeführt werden und anschließend dieser Satz aktualisiert werden, so ist GET-KDS das performantere Kommando, da ein zusätzliches SPERRE-D Kommando (vgl. Seite 141) oder ein erneutes Suchen und Sperren mit dem SPERRE-Z Kommando (vgl. Seite 143) entfällt.

2.11.10 GET-KF (Satz feldweise)

Anwendungsart: aktiv

```
GET[,n],KF{N/R/P}[S],Feld[(Erg)][_Feld...]...[,DB.xxxxxx]
```

GET	Operation für GET.
n	Auf Sätze bezogene Mengenangabe. Bei fehlender Angabe wird 1 angenommen. $ = alle Sätze.
KFN[S]	Operationsergänzung für Lesen Feld nächster Satz [mit Sperre].
KFR[S]	Operationsergänzung für Lesen Feld vorhergehender (rückwärts) Satz [mit Sperre].
KFP[S]	Operationsergänzung für Lesen Feld des Satzes auf dem der Pointer steht [mit Sperre].
Feld	Feldbezeichnung
xxxxxx	Satzbeschreibung

Die Felder werden pro Satz mit Zwischentrennern aneinander gereiht. Führende und anhängende Spaces verschwinden.

Enthält ein Feld nur Spaces, so wird nur der Trenner ausgegeben.

Aktualisierungsfehler bringen ebenfalls nur den Trenner.

Am Ende eines jeden Satzes wird der Ende-Trenner eingefügt.

Beachte: Multiplitäten können nicht angegeben werden, sie werden immer vollständig abgearbeitet (AZ=$) (FZ=$). Das Kommando ist daher nur für lineare oder linearisierte Sätze zu empfehlen.

Beispiel: Zwischentrenner: C';'
Endetrenner: X'15' (Zeilenvorschub) Standardwert: C':'

```
GET,2,KFP,Name Straße

Meier Sepp;Dornweg;«
Kunz Karl;Leopoldstr.225;«
```

2.11.11 GET-KI (Indirekt über VD lesen)

Anwendungsart: passiv

```
GET,KI[S],Feld{=/</>/:/*}Wert[,Key{=/</>/:/*}Wert][,DB.xxxxxx]
```

GET	Operation für GET.
KI	Operationsergänzung für indirektes Lesen.
KIS	mit Sperre
Feld	Invertiertes Feld.
Key	Feld, das als Ordnungsbegriff definiert ist.
Wert	Wert im externen Format.
xxxxxx	Satzbeschreibung

Dieses Kommando ermöglicht das sequentielle Lesen von HD-Sätzen entlang der VD-Einträge. Der Aufsetzpunkt in der VD wird durch den Wert für das invertierte Feld festgelegt.

Ist kein Wert für den Ordnungsbegriff angegeben, so wird der erste VD-Eintrag zur Lokalisierung des Satzes herangezogen.

Das Kommando setzt den Zeiger in der Verweisdatei auf den gelesenen VD-Eintrag (vgl. Seite 8). Der Zeiger kann auch mit BLAETTERN,I (vgl. Seite 42) versetzt werden. Mit GET,KI,Ob>... kann weitergelesen, mit GET,KI,Ob<... kann rückwärts gelesen werden.

Eine bestehende Zielpunktliste wird mit GET,KI übergangen. Soll der VD-Zeiger für alle Ausgaben (GET, ZEIGE...) Gültigkeit haben, darf keine ZPL gültig sein (vgl. Seite 8).

2.11.12 GET-KN/KR (Nächster Satz/rückwärts)

Anwendungsart: passiv

```
GET,{KN/KR}[S][,DB.xxxxxx]
```

GET	Operation für GET.
KN	Operationsergänzung für nächsten Satz vorwärts.
KR	Operationsergänzung für nächsten Satz rückwärts.
S	mit Sperre
xxxxxx	Satz- oder Transformationsbeschreibung.

Durch den Aufruf von GET,KN wird der jeweils nachfolgende Satz zur Verfügung gestellt (die Syntax GET,N wird noch unterstützt). Bei den Kommandos GET-KP (vgl. Seite 92), GET-KI (vgl. Seite 90) und GET-KD (vgl. Seite 87) kann somit der nachfolgende bzw. vorhergehende Satz gelesen werden, ohne nach dem Kommando BLAETTERN,1,V nochmals die genannten Kommandos anwenden zu müssen, d.h. es ist nur ein statt zwei Kommandos notwendig.

Dabei ist entscheidend, welche Zeigerart vorliegt: ZPL-Zeiger, VD-Zeiger oder FCB-Zeiger (vgl. Seite 8).

Ist eine Zielpunktliste vorhanden, so wird über diese der nächste Satz gelesen. Soll aber der nächste Satz nach dem FCB-Zeiger (der Satz mit dem nächsthöheren Schlüssel) gelesen werden, so muß die bestehende Zielpunktliste mit dem Kommando IGNORIEREN-Z (vgl. Seite 99) ignoriert werden. Falls ein VD-Zeiger existiert, muß dieser mit IGNORIEREN-I (vgl. Seite 97) gelöscht werden.

Dies gilt analog für das Rückwärtslesen (GET,KR).

Mit der Operationsergänzung KNS/KRS läßt sich der gelesene Satz gleichzeitig sperren.

2.11.13 GET-KP (Satz)

Anwendungsart: passiv

```
GET,KP[S][,DB.xxxxxx]
```

GET	Operation für GET.
KP	Operationsergänzung für Satz.
KPS	Mit Sperre.
xxxxxx	Satz- oder Transformationsbeschreibung.

Das Kommando GET-KP übergibt dem aufrufenden Programm den lokalisierten Satz aus einer Hauptdatei. Dabei kann entweder der vollständige Datensatz übergeben werden, oder der Satz kann in ein von der Datenbankspeicherung abweichendes Format übertragen werden (Transformation). In jedem Fall enthält der Satz ein Feld mit der Satzlänge und evtl. Felder mit Abschnittslänge/Abschnittsart.

Speziell bei RECFORM=F muß im Anwenderprogramm berücksichtigt werden, daß der Satz im Speicher mit V-Format aufgebaut ist, d.h. mit Satzlänge gleich RECSIZE+4.

Bei einer Transformation erfolgt die Steuerung über eine Transformationsbeschreibung, deren Name beim GET-Aufruf als Paßwort verwendet wird.

Ein Zugriff auf den nächsten Satz ist erst nach BL,1,V oder direkt mit dem GET-KN (vgl. Seite 91) möglich. Für das Rückwärtslesen gelten die analogen Kommandos.

Die Operationsergänzung KPS sperrt den lokalisierten Satz.

Die Operationsergänzung K (gültig bis CIS-Version 10) wird noch unterstützt.

2.11.14 GET-P (Satzerfassung/ -änderung mit Bildschirmmaske)

Anwendungsart: passiv

```
GET,P[(Erg)][,DB.xxxxxx]
```

GET	Operation für GET.
P	Operationsergänzung für PUT Bildschirm.
Ergänzung (vgl. Seite 18)	AM = Abschnittsmultiplität: Bei fehlender AM-Angabe wird AM=1 angenommen.
xxxxxx	Bildschirmmaske

Das Kommando unterscheidet sich vom Kommando GET-B (vgl. Seite 79) lediglich darin, daß bei der Ausgabe auf das Datensichtgerät die Bildschirmmaske mit den im aufrufenden Programm definierten Daten (Satz) gefüllt ist. Das bedeutet, daß im Satzbereich (6. Übergabeparameter - vgl. Manual 4) sowohl Daten vom Anwenderprogramm an CIS als auch Daten von CIS an das Anwenderprogramm übergeben werden. Der Bereich enthält ein Feld mit der Satzlänge und evtl. Felder mit Abschnittslänge/Abschnittsart. Wird in der Bildschirmmaske nichts geändert, werden die Daten unverändert an das Anwenderprogramm zurückgegeben, ansonsten der geänderte bzw. ergänzte Satz.

Dadurch eignet sich dieses Kommando nicht nur zur Änderungsunterstützung, sondern es kann auch, wenn es sinnvoll ist, konstante Daten vor der Erfassung auszugeben, zur Datenerfassung verwendet werden.

2.11.15 GET-ZP (Zielpunktliste lesen)

Anwendungsart: passiv

```
GET,ZP[,KEY[(Erg)][_Key...]...[,DB.xxxxxx]
```

GET	Operation für GET.
ZP	Operationsergänzung für Zielpunktliste.
Key	Ordnungsbegriffe am Fenster beteiligter Datenbanken.
Ergänzung	DB=xxxxxx zur Unterscheidung nicht eindeutiger Feldnamen im Verbund.
xxxxxx	Satzbeschreibung

Das Kommando liest eine bestehende Zielpunktliste. Ab ZPL-Zeiger werden die Einträge der ZPL im Empfangsbereich zur Verfügung gestellt.

Ist der Empfangsbereich groß genug (Längenangabe im 5. Aufrufparameter - vgl. Manual 4), um alle Einträge aufzunehmen, quittiert CIS mit IM00. Bei zu kleinem Empfangsbereich wird die codierte Meldung GE06 ausgegeben. Der ZPL-Zeiger wird um die Anzahl der übergebenen Einträge implizit weitergeschaltet.

Folglich kann nach der Meldung GE06 ein erneuter GET-ZP-Aufruf abgesetzt werden, um die nächsten ZPL-Einträge zu erhalten.

Bei einer Zielpunktliste im Fenster ist im GET-ZP Kommando die Angabe der gewünschten Ordnungsbegriffe zwingend.

Die Reihenfolge der Ordnungsbegriffe im Kommando (nur im Fenster zulässig) bestimmt deren Reihenfolge in der Ausgabe.

Nach einem GET-ZP Aufruf steht der ZPL-Zeiger auf dem letzten Eintrag + 1, bei EOF auf dem letzten Eintrag. Danach kann beispielsweise mit dem BL-Kommando positioniert werden.

2.12 HALT-Kommando

Anwendungsart: aktiv

```
HALT
```

HALT	Operation für Halt.

Sämtliche offenen Dateien werden geschlossen, das Programm wird beendet.

2.13 HILF-Kommando

Anwendungsart: aktiv

```
HILF[_k][,DB.xxxxxx]
```

HILF	Operation für Hilfe.
k	CIS-Kommando. Es kann bis zur Eindeutigkeit verkürzt angegeben werden.
xxxxxx	Die DB-Angabe kann entfallen. Die Angabe DB.xxxxxx wirkt sich erst auf das nächste Kommando aus.

Mit diesem Kommando kann die Syntax von CIS-Kommandos erläutert werden.

Wird nur HILF angegeben, so wird eine Liste aller CIS-Kommandos auf dem Bildschirm ausgegeben. Bei der Angabe von HILF k wird das CIS-Kommando k näher erläutert.

2.14 IGNORIEREN-Kommando

Die einzelnen Arten des IGNORIEREN-Kommandos führen völlig unterschiedliche Funktionen aus und müssen daher getrennt betrachtet werden.

2.14.1 IGNORIEREN-F (Fenster)

Anwendungsart: aktiv/passiv

```
IGNORIEREN,F[,DB.xxxxxx]
```

IGNORIEREN	Operation für Ignorieren.
F	Operationsergänzung für Fenster.
xxxxxx	Satzbeschreibung

Mit diesem Kommando wird das derzeit gültige Fenster verlassen und in diejenige Datenbank verzweigt, die in der angegebenen Satzbeschreibung definiert ist.

Dabei wird aus der aktuell gültigen Zielpunktliste (ZPL) im Fenster eine neue ZPL gebildet. Diese ZPL enthält nur diejenigen Ordnungsbegriffe bzw. Wiedergewinnungsadressen, die in der (nun aktuellen) Datenbank vorkommen.

Beispiel:

```
*
VE KNDR(DB=KUNDEN)=AKNR(DB=AUFTRA),DB.AAAAAA          1)
IM10    ANZAHL RELATIONSZEILEN: 1020
*
SU AUFTRAGSWERT>10000                                 2)
IM10    ANZAHL RELATIONSZEILEN: 14
*
IG,F,DB.KUNDEN                                        3)
IM00    ANZAHL ZIELINFORMATIONEN: 3
*
```

1) Für alle Kunden mit zugehörigen Aufträgen wird eine Relation erstellt.
2) Erstellen einer Zielpunktliste innerhalb der Relation, in diesem Fall über alle Relationszeilen.
3) Fenster verlassen und in die Kunden-Datenbank verzweigen. Zielpunktliste mit allen Kunden, die einen Auftragswert größer 10000 aufweisen.

2.14.2 IGNORIEREN-I (Zeiger der VD)

Anwendungsart: aktiv/passiv

```
IGNORIEREN,I[,DB.xxxxxx]
```

IGNORIEREN	Operation für Ignorieren.
I	Operationsergänzung für Zeiger der VD.
xxxxxx	Satzbeschreibung

Dieses Kommando löscht den Zeiger in der Verweisdatei. Der VD-Zeiger wird explizit mit BLAETTERN-I (vgl. Seite 42) und implizit mit GET-KI (vgl. Seite 90) gesetzt.

Zum Beispiel darf kein Verweisdateizeiger existieren, wenn (ohne vorhandene ZPL) sequentiell in der HD gelesen werden soll.

2.14.3 IGNORIEREN-S (Suchergebnis)

Anwendungsart: aktiv/passiv

```
IGNORIEREN[,S][,DB.xxxxxx]
```

IGNORIEREN	Operation für Ignorieren.
S	Operationsergänzung für Suchergebnis.
xxxxxx	Satzbeschreibung.

Voraussetzung für dieses Kommando ist, daß die Zielpunktliste der vorletzten Suchfrage (Initial- oder Folgefrage) mit dem Kommando SICHERN-S (vgl. Seite 133) festgehalten wurde.

Durch das Kommando IGNORIEREN-S wird das letzte Suchergebnis ignoriert, d.h. die Zielpunktliste wird auf die vorletzte zurückgesetzt.

Beispiele:

```
SU WOHNORT=MUENCHEN
IM00    ANZAHL ZIELINFORMATIONEN: 100
*
IGNOR
SU06 KEINE GESICHERTE ZPL VORHANDEN FUER IG,S
*
SI,S
*
SU WOHNORT=MUENCHEN
IM00    ANZAHL ZIELINFORMATIONEN: 100
*
UND NAME=SCHMITT
IM00    ANZAHL ZIELINFORMATIONEN: 2
*
ZEIGE,2,Z,VORNAME
VORNAME     EGON
VORNAME     FRITZ
*
IGNOR,S
IM00    ANZAHL ZIELINFORMATIONEN: 100
*
UND NAME=SCHMITTEN
IM00    ANZAHL ZIELINFORMATIONEN: 1
*
ZEIGE,1,Z,VORNAME
VORNAME     LUISE
```

2.14.4 IGNORIEREN-Z (Zielpunktliste)

Anwendungsart: aktiv/passiv

```
IGNORIEREN,Z[,DB.xxxxxx]
```

IGNORIEREN	Operation für Ignorieren.
Z	Operationsergänzung für Zielpunktliste.
xxxxxx	Satzbeschreibung

Dieses Kommando löscht eine Zielpunktliste und setzt damit auch den ZPL-Zeiger (vgl. Seite 8) außer Kraft. Nachfolgende Satzzugriffe (z.B. für Ausgaben) erfolgen über den VD-Zeiger wenn dieser für die betreffende Datenbank definiert ist oder mit dem FCB-Zeiger sequentiell ab Dateianfang.

Beispiel:

```
*
IGNORIERE,Z
*
SORTIERE NAME
IM01    ANZAHL ZIELINFORMATIONEN: 100
*
```

Die Sätze der gesamten Hauptdatei werden aufsteigend nach NAME sortiert.

2.15 KORRIGIEREN-Kommando

Anwendungsart: aktiv

```
KORRIGIEREN[,n],K[,DB.xxxxxx]
```

KORRIGIEREN	Operation für Korrigieren.
K	Operationsergänzung für Satz.
n	Mengenangabe bezieht sich auf die zu korrigierenden Sätze. Bei fehlender Angabe wird $ (=alle) angenommen.
xxxxxx	Bildschirmmaske

Die auf dem Bildschirm erscheinende Maske ist mit den Daten des lokalisierten Satzes gefüllt. Numerische Feldinhalte werden dabei entsprechend der Feldbedeutung und Definition aufbereitet (eventuell negatives Vorzeichen, Unterdrückung führender Nullen, Dezimalpunkt).

Die Daten können nun verändert werden. Beim Korrigieren numerischer Feldinhalte gelten sinngemäß die Ausführungen des EINGEBEN-Kommandos (vgl. Seite 68).

Am Ende der Bildschirmmaske werden fünf Möglichkeiten angeboten:

WEITER MASKE?	Die Folgemaske (falls zu xxxxxx eine Folgemaske definiert wurde) wird mit Daten gefüllt ausgegeben.
WEITER SATZ?	Nach Aktualisierung des aktuellen Haupt- bzw. Verweisdateisatzes wird auf den nächsten Satz geblättert und am Bildschirm ausgegeben.
WEITER ABSCHNITT?	Bei Wiederholabschnitten (nur Sätze mit MV-Format) erscheint die gleiche Maske mit den Daten des nächsten Wiederholabschnitts des aktuellen Satzes.
ENDE?	Der Korrekturmodus wird beendet, der aktuelle Haupt- bzw. Verweisdateisatz wird aktualisiert.
ABBRUCH?	Der Korrekturmodus wird abgebrochen, es erfolgt keine Aktualisierung.

Durch Eingabe eines beliebigen Zeichens ungleich Space hinter der entsprechenden Position wird die jeweilige Funktion ausgeführt. Die Position WEITER SATZ? ist standardmäßig mit dem Zeichen X vorbesetzt.

2.16 LOESCHEN-Kommando

Mit dem Kommando LOESCHEN können bereits vorhandene Daten gelöscht werden. Die Ausführung des Kommandos ist jeweils auf Satz-, Abschnitts- oder Feldebene möglich. Ist beim Löschen von Wiederholabschnitten oder Wiederholfeldern keine Multiplität explizit angegeben, so wird der erste Abschnitt bzw. das erste Feld gelöscht.

Sind invertierte Felder vorhanden, werden die Verweisdateieinträge ebenfalls gelöscht. Die Prüfung, ob die Felder invertiert sind, erfolgt über die beim aktuellen Kommando verwendete Datenbeschreibung. Es ist deshalb darauf zu achten, daß die dabei verwendete Datenbeschreibung alle invertierten Felder enthält.

2.16.1 LOESCHEN-A (Abschnitt)

Anwendungsart: aktiv/passiv

```
LOESCHEN[,n],A,Abschnittsart[(Erg)][,DB.xxxxxx]
```

LOESCHEN	Operation für Löschen.
n	Auf Sätze bezogene Mengenangabe. $ = alle Sätze. Bei fehlender Mengenangabe wird 1 angenommen.
A	Operationsergänzung für Abschnitt.
Abschnittsart	Vierstelliger Name des zu löschenden Abschnitts.
Ergänzung (vgl. Seite 18)	AM = Abschnittsmultiplität: Bei fehlender AM-Angabe wird AM=1 angenommen. AZ = Abschnittszyklen: $ = alle Abschnittszyklen. Bei fehlender AZ-Angabe wird AZ=1 angenommen.
xxxxxx	Satzbeschreibung

Im jeweils lokalisierten Satz wird der Abschnitt gelöscht, der über die im Kommando angegebene Abschnittsart ermittelt worden ist.

Als Ergänzungsangabe sind AM=[-]n und AZ=n möglich, d.h. bei Wiederholabschnitten kann die Abschnittsmultiplität und ggf. die Anzahl der zu löschenden Wiederholabschnitte je Satz angegeben werden.

Beispiele:

```
*
LOESCHEN,A,ABCD
*
```

In einem Satz (Mengenangabe fehlt) wird der Abschnitt ABCD gelöscht. Falls ABCD ein Wiederholabschnitt ist, wird nur der erste Abschnitt gelöscht, da eine AM- bzw. AZ-Angabe fehlt.

```
*
LOESCHEN,A,ABCD(AM=-1)
*
```

In einem Satz (Mengenangabe fehlt) wird der letzte Wiederholabschnitt der Abschnittsart ABCD gelöscht.

```
*
LOESCHE,5,A,ABCD(AZ=3)
*
```

In fünf Sätzen werden jeweils ab der ersten Abschnittsmultiplität (AM-Angabe fehlt) drei Wiederholabschnitte gelöscht.

```
*
L,2,A,ABCD(AM=-3,AZ=$)
*
```

In zwei Sätzen werden jeweils ab der drittletzten Abschnittsmultiplität alle Wiederholabschnitte gelöscht.

2.16.2 LOESCHEN-F (Wiederholfeld)

Anwendungsart: aktiv/passiv

```
LOESCHEN[,n],F,Feld[(Erg)][,DB.xxxxxx]
```

LOESCHEN	Operation für Löschen.
n	Auf Sätze bezogene Mengenangabe. $ = alle Sätze. Bei fehlender Mengenangabe wird 1 angenommen.
F	Operationsergänzung für Wiederholfeld.
Feld	Name des zu löschenden Feldes.
Ergänzung (vgl. Seite 18)	AM = Abschnittsmultiplität: Bei fehlender AM-Angabe wird AM=1 angenommen. AZ = Abschnittszyklen: $ = alle Abschnittszyklen. Bei fehlender AZ-Angabe wird AZ=1 angenommen. FM = Feldmultiplität: Bei fehlender FM-Angabe wird FM=1 angenommen. FZ = Feldzyklen: $ = alle Feldzyklen. Bei fehlender FZ-Angabe wird FZ=1 angenommen.
xxxxxx	Satzbeschreibung

Das Kommando LOESCHEN-F kann nur auf Wiederholfelder angewendet werden. Im jeweils lokalisierten Satz wird durch Angabe der Feldbezeichnung das entsprechende Feld gelöscht.

Als Ergänzungsangaben sind AM=[-]n, AZ=n, FM=[-]n und FZ=n möglich. Fehlen Ergänzungsangaben wird jeweils der Wert 1 angenommen.

Beispiele:

```
*
LOESCHE,F,DIAGNOSE(FM=-1)
*
```

In einem Satz (Mengenangabe fehlt) wird der letzte Diagnoseeintrag gelöscht. Falls DIAGNOSE ein Wiederholfeld in einem Wiederholabschnitt ist, wird der letzte Diagnoseeintrag im ersten Wiederholabschnitt gelöscht, da AM- und AZ-Angabe fehlen.

```
*
LOESCHE,F,DIAGNOSE(AM=-1,FM=2)
*
```

Im letzten Wiederholabschnitt eines Satzes (Mengenangabe fehlt) wird der zweite Diagnoseeintrag gelöscht.

```
*
L,5,F,DIAGNOSE(AM=-3,AZ=2,FM=4,FZ=$)
*
```

In fünf Sätzen werden jeweils im dritt- und zweitletzten Wiederholabschnitt alle Diagnoseeinträge ab der vierten Multiplität gelöscht.

2.16.3 LOESCHEN-K (Satz)

Anwendungsart: aktiv/passiv

```
LOESCHEN[,n],K[,DB.xxxxxx]
```

LOESCHEN	Operation für Löschen.
n	Auf Sätze bezogene Mengenangabe. $ = alle Sätze. Bei fehlender Mengenangabe wird 1 angenommen.
K	Operationsergänzung für Satz.
xxxxxx	Satzbeschreibung

Entsprechend der im Kommando angegebenen Menge werden ein oder mehrere Sätze in der Datenbank gelöscht.

Beispiel:

```
*
SU RECHNUNGSDATUM=30.02.82
IM00    ANZAHL ZIELINFORMATIONEN: 20
*
LOESCHE,10,K
*                  Die ersten zehn Sätze der Zielpunktliste werden gelöscht.
BL,5,V
*
L,$,K
*                  Ab dem 15. Satz werden alle Sätze bis zum Ende der ZPL, d.h. also 6 Sätze
                   gelöscht.
IGNORIERE,Z
*                  Der erste Satz der Datenbank bzw der Satz, auf den der VD-Zeiger weist,
                   wird gelöscht (vgl. IGNORIEREN-Z Seite 99).
L,K
*
```

2.16.4 LOESCHEN-KD (Satz direkt)

Anwendungsart: aktiv/passiv

```
LOESCHEN,KD,Key=Wert[,DB.xxxxxx]
```

LOESCHEN	Operation für Löschen.
KD	Operationsergänzung für Satz direkt.
Key	Feldbezeichnung des als Ordnungsbegriff gekennzeichneten Feldes.
Wert	Schlüssel der Hauptdatei.
xxxxxx	Satzbeschreibung

Der angegebene Satz wird gelöscht, d.h. eine eventuell bestehende Zielpunktliste wird nicht berücksichtigt.

Wird mit Transaktionen gearbeitet, so muß der Satz gesperrt sein.

2.17 OPEN-Kommando

Anwendungsart: aktiv/passiv

```
OPEN{_/,{U/I},}{HD/VD}=Dateiname[,OPEN...]...[,DB.xxxxxx]
```

OPEN	Operation für OPEN.
U	Operationsergänzung für OPEN zum Update und Lesen.
I	Operationsergänzung für OPEN für nur lesenden Zugriff. Standardwert bei fehlender U- /I-Angabe.
HD	Bezeichnung für Hauptdatei.
VD	Bezeichnung für Verweisdatei.
Dateiname	Dateiname der Haupt- oder Verweisdatei.
xxxxxx	Satz- oder Transformationsbeschreibung.

Das OPEN-Kommando dient dem Öffnen der benötigten Haupt- bzw. Verweisdatei. Im A-Segment der aktuellen Satzbeschreibung muß EOC=J angegeben sein.

Wenn ohne OPEN-Kommando gearbeitet wird, bedient sich der dann implizit ablaufende OPEN der Datenbeschreibungsparameter:

HD=Dateiname - Öffnen einer Hauptdatei
VD=Dateiname - Öffnen einer Verweisdatei.

Die Vorteile des impliziten (verdeckten) OPEN bestehen darin, daß zur verwendeten Datenbeschreibung immer die zugehörigen Dateien geöffnet werden. Eine Fehlbedienung wird somit ausgeschlossen und der Datenschutz ist gewährleistet.

Der explizite OPEN mit direkter Zuweisung der Datei(en) sollte also nur in Sonderfällen verwendet werden.

Achtung: Beim expliziten OPEN bleibt die betreffende Datei solange geöffnet bis ein entsprechendes CLOSE-Kommando abgesetzt oder CIS beendet wird. Die Datei bleibt auch dann geöffnet, wenn ein neues OPEN-Kommando für eine weitere Datei gegeben wird. Eine Verbindung zwischen Paßwort und Dateien kann nicht aufgebaut werden. Beim Paßwortwechsel müssen also die gewünschten Dateien immer wieder mit einem expliziten OPEN zugewiesen werden. Ist die Datei noch nicht geschlossen, dient der OPEN nur der erneuten Zuweisung der Datei zum Paßwort.

Beispiele:

Impliziter OPEN

```
 *
SUCHE NAME=HUBER,DB.PERSON
IM00    ANZAHL ZIELINFORMATIONEN: 1
*
```

Die benötigten Dateien werden vor Start des SUCHE-Kommando über entsprechende Parameter, die in der Satzbeschreibung enthalten sind, mit INPUT geöffnet. Bei evtl. nachfolgenden Updatekommandos wird die Datei geschlossen und automatisch mit INOUT wieder geöffnet.

Expliziter OPEN

```
*
OPEN,U,HD=HD.PERS,OPEN,U,VD=VD.PERS,DB.PERSON
*
```

Hauptdatei und Verweisdatei werden für Update mit INOUT geöffnet.

2.18 PUT-Kommando

Mit dem PUT-Kommando können Daten in die Haupt- und Verweisdatei aufgenommen werden. Die Ausführung des Kommandos ist jeweils auf Satz-, Abschnitts- oder Feldebene möglich.

Sind invertierte Felder vorhanden, wird die Verweisdatei ebenfalls aktualisiert. Die Prüfung, ob die Felder invertiert sind, erfolgt über die beim aktuellen Kommando verwendete Datenbeschreibung. Es ist deshalb darauf zu achten, daß diese Datenbeschreibung alle invertierten Felder enthält.

2.18.1 PUT-A (Abschnitt)

Anwendungsart: passiv

```
PUT,A[,DB.xxxxxx]
```

PUT	Operation für PUT.
A	Operationsergänzung für Abschnitt.
xxxxxx	Satzbeschreibung

Der vom aufrufenden Programm übergebene Abschnitt wird an das physikalische Ende des lokalisierten Satzes angefügt. Der anzufügende Abschnitt muß vollständig, d.h. mit allen Abschnittsattributen wie Abschnittslänge (AL) und Abschnittsart (AA) übergeben werden. Mit diesem Kommando kann also entweder ein Abschnitt, der bisher noch nicht vorhanden ist, angelegt werden, oder aber es wird für einen Wiederholabschnitt die erste oder eine weitere Multiplität angefügt.

Wird mit Transaktionen gearbeitet, so muß der Satz gesperrt sein.

2.18.2 PUT-AF (Abschnitt feldweise)

Anwendungsart: aktiv

```
PUT,AF,Feld=Wert[,Feld]...[,DB.xxxxxx]
```

PUT	Operation für PUT.
AF	Operationsergänzung für Abschnitt feldweise.
Feld	Feldbezeichnung
Wert	Feldinhalt
xxxxxx	Satzbeschreibung

Mit diesem Kommando wird ein neuer Abschnitt an den lokalisierten Satz angefügt. Die Felder müssen alle im gleichen Abschnitt stehen. Nicht angegebene Felder werden mit Initialwerten gefüllt.

Wird mit Transaktionen gearbeitet, so muß der Satz gesperrt sein.

2.18.3 PUT-F (Wiederholfeld)

Anwendungsart: aktiv/passiv

```
PUT[,n],F,Feld[(Erg)]=Wert[,DB.xxxxxx]
```

PUT	Operation für PUT.
n	Auf Sätze bezogene Mengenangabe. Bei fehlender Angabe wird 1 angenommen.
F	Operationsergänzung für Feld.
Feld	Name des anzufügenden Wiederholfeldes.
Ergänzung (vgl. Seite 18)	AM = Abschnittsmultiplität: Bei fehlender AM-Angabe wird AM=1 angenommen.
Wert	Feldinhalt des zu speichernden Feldes.
xxxxxx	Satzbeschreibung

Dieses Kommando ist nur auf Wiederholfelder anwendbar. Das übergebene Feld wird jeweils an das Ende der entsprechenden Wiederholfeldkette angefügt. In der Ergänzung kann ggf. eine Abschnittsmultiplität angegeben werden.

Ist das Wiederholfeld ein numerisches Feld, sind bei der Eingabe die Hinweise im Kapitel "Behandlung numerischer Feldinhalte" (Seite 21) zu beachten.

Wird mit Transaktionen gearbeitet, so muß der Satz gesperrt sein.

Beispiele:

```
*
PUT,F,PRODUKTNR=4711
*
```

Im bestehenden Satz wird ein Wiederholfeld angefügt. Bei V-Format wird das Feld am Ende des Satzes angefügt. Bei MV-Format wird das Feld entweder am Ende des entsprechenden festen Abschnitts oder am Ende des ersten entsprechenden Wiederholabschnitts angefügt, da die AM-Angabe fehlt.

```
*
PUT,F,PRODUKTNR(AM=-1)=4711
```

Im letzen entsprechenden Wiederholabschnitt des MV-Format-Satzes wird ein Wiederholfeld angefügt.

2.18.4 PUT-K (Satz)

Anwendungsart: passiv

```
PUT,K[,DB.xxxxxx]
```

PUT	Operation für PUT.
K	Operationsergänzung für Satz.
xxxxxx	Satzbeschreibung

Der vom aufrufenden Programm übergebene Satz wird als neuer Satz in die Hauptdatei eingetragen. Die Satzlänge muß richtig gesetzt sein. Beim MV-Format müssen Abschnittslänge (AL) und Abschnittsart (AA) angegeben sein. Wenn LOGADR=N in der Datenbeschreibung angegeben ist, muß ein gültiger Datenbankschlüssel (Ordnungsbegriff) an der durch KEYPOS definierten Stelle vorhanden sein. Bei Angabe LOGADR=J wird von CIS an der durch KEYPOS definierten Stelle ein binärer Schlüssel in der durch KEYLEN definierten Länge eingetragen. Die Numerierung beginnt bei 1 und wird mit einer Schrittweite von 1 fortgesetzt.

Der von CIS eingesetzte Schlüssel wird im Sendebereich (5. Parameter - vgl. Manual 4) zurückgemeldet.

Wird mit Transaktionen gearbeitet, so wird der neue Satz automatisch gesperrt.

Speziell bei RECFORM=F muß im Anwenderprogramm sichergestellt sein, daß der Satz im Speicher mit V-Format aufgebaut ist, d.h. mit Satzlänge gleich RECSIZE+4.

2.18.5 PUT-KF (Satz feldweise)

Anwendungsart: aktiv

```
PUT,KF,Feld=Wert,...[,DB.xxxxxx]
```

PUT	Operation für PUT.
KF	Operationsergänzung für Satz feldweise.
Feld	Feldbezeichnung
Wert	Feldinhalt
xxxxxx	Satzbeschreibung

Methode, um Sätze im Aktiv/Server-Betrieb zu erzeugen.

Felder der Satzbeschreibung, die im Kommando nicht mit Werten gefüllt werden, sind mit formatgerechten Standardwerten belegt.

Es muß mindestens ein Feld angegeben werden.

2.19 RECHNE-Kommando

Anwendungsart: aktiv/passiv

```
RECHNE{,n,/[,n],{S/Z},/_}Ergebnis[(Erg)]=Ausdruck[(Erg)],
      [,{S/Z},Ergebnis...]...[,DB.xxxxxx]
```

RECHNE	Operation für Rechnen.
n	Auf Sätze bezogene Mengenangabe. $ = alle Sätze. Bei fehlender Mengenangabe wird 1 angenommen.
Z	Operationsergänzung für "zeilenweises Rechnen" (Standardwert).
S	Operationsergänzung für "summarisches Rechnen".
Ergebnis	a) Feld der aktuellen Datenbeschreibung mit der Feldbedeutung P, B, R oder A. Enthält es Sonderzeichen (- + * : /), so ist es in Hochkomma anzugeben. b) Rechenvariable: %X0 ... %X9, %XA ... %XZ. %Y0 ... %Y9, %YA ... %YZ.
Ausdruck	Allgemeiner Ausdruck, der Rechenvariablen, Felder (nur mit Feldbedeutung P, B, R, A), Konstanten und Rechenoperatoren enthalten kann.
Ergänzung (vgl. Seite 18)	AM = Abschnittsmultiplität FM = Feldmultiplität Als Ergänzungsangabe im Operanden, Ausdruck, kann zusätzlich noch der AZ- und/oder der FZ-Parameter angegeben werden. Dies bedeutet dann eine implizite Addition.
xxxxxx	Satzbeschreibung

2.19.1 %X-Rechenvariablen

Sie gelten global und sind mit dem Wert 0 initialisiert.

Initialisierung der Variablen %Xn: **RE**CHNE %Xn=Wert[,DB.xxxxxx]

n = 0,1, ... Z
Wert = numerische Angabe

Initialisierung aller Variablen: **RE**CHNE %X=0[,DB.xxxxxx]

Alle %X-Rechenvariablen werden auf Null gesetzt.

2.19.2 %Y-Rechenvariablen

Zu einer Zielpunktliste mit AZI>0 können %Y-Rechenvariablen

%Y0, %Y1, ... ,%YZ

definiert werden, die satzweise in der ZPL gehalten werden.

Es handelt sich um Rechenfelder mit maximal 29 Stellen. Die Standardformatierung (Länge 15 inklusiv 2 Nachkommastellen) kann mit Hilfe des RECHNE-Kommandos verändert werden:

RECHNE %Yn(l,k)=Wert

n = 0,1,... Z
l = Länge (≤ 29)
k = Nachkommastellen (< l)

Die Formatierung einer %Y-Variablen und deren Inhalte leben mit der ZPL. Zum Beispiel gehen die Inhalte nach einer erneuten Initialfrage verloren, wenn die alte ZPL nicht temporär gesichert wurde (vgl. SICHERN-E Seite 129).

Eine ZPL mit %Y-Variablen kann nicht weiter verknüpft werden, d.h. eine Folgefrage ist nicht erlaubt.

2.19.3 Algebraische Ausdrücke im RECHNE-Kommando

Ein algebraischer Ausdruck kann die vier Grundrechenarten mit folgenden Operatoren enthalten:

+ Addieren
- Subtrahieren
* Multiplizieren
/ Dividieren

Als Operanden sind Rechenvariablen, Felder und Konstanten zulässig. Klammerausdrücke sind auch geschachtelt möglich. Außerdem kann die Anzahl der Zielinformationen mit %AZI angesprochen werden. %AZI kann nur in algebraischen Ausdrücken verwendet werden, ist aber nicht als Ergebnisvariable zulässig. Analog gilt dies für die Systemkonstanten %SYSDAT und %SYSTIM sowie %SYSTSN, sofern die TSN nicht alphanumerisch ist.

Das Ergebnis des algebraischen Ausdrucks wird in einer Rechenvariablen oder in einem Feld abgespeichert. Wurde ein Feld angegeben, erfolgt in der Haupt- bzw. Verweisdatei ein Update des entsprechenden Datenfeldes.

Grundsätzlich lassen sich vier Operationsergänzungen des RECHNE-Kommandos unterscheiden:

```
RECHNE,Z,Feld=Ausdruck
RECHNE,Z,Rechenvariable=Ausdruck

RECHNE,S,Feld=Ausdruck
RECHNE,S,Rechenvariable=Ausdruck
```

Zeilenweises Rechnen

1. Operationsergänzung Z - Das Ergebnis wird in einem Feld abgelegt.

   Der im Kommando angegebene Operand bzw. Ausdruck wird berechnet und das Ergebnis dem Feld zugeordnet. Die im Ausdruck eventuell vorkommenden Felder werden vorher mit den Daten des aktuellen Satzes belegt.

   Über die Mengenangabe wird die Anzahl der Sätze angegeben, bei denen der Ausdruck neu berechnet und das Ergebnis dem jeweiligen Feld zugeordnet wird. Mit dem jeweiligen neuen Feldinhalt erfolgt ein Update in der Haupt- bzw. Verweisdatei.

Beispiel:

```
*
SU GEBJAHR=1930,BIS GEBJAHR=1960
IM00    ANZAHL ZIELINFORMATIONEN: 78
*
RE,$,Z,ALTER=1994-GEBJAHR
*
BL,$,R
*
Z,4,T,ALTER NAME

ALTER NAME
   47 MAIER
   27 HUBER
   30 MUELLER
   35 BAUMANN
```

ALTER ist ein Feld der Datenbeschreibung. Für die 78 mit der Suchfrage lokakisierten Sätze wird der Inhalt des Feldes ALTER mit dem RECHNE-Kommando ermittelt.

2. Operationsergänzung Z - Das Ergebnis wird in einer Rechenvariablen abgelegt.

Der im Kommando angegebene Ausdruck wird berechnet und der Rechenvariablen zugeordnet. Die im "Ausdruck" eventuell vorkommenden Felder werden vorher mit den Daten des aktuellen Satzes belegt.

Beispiel:

```
*
SU GEBJAHR=1930,BIS GEBJAHR=1960
IM00    ANZAHL ZIELINFORMATIONEN: 78
RE %Y1(2,0)=0
RE,$,Z,%Y1=1994-GEBJAHR
BL,$,R
Z,4,T,%Y1ALTER NAME
%Y1ALTER NAME
      47 MAIER
      27 HUBER
      30 MUELLER
      35 BAUMANN
*
```

Anmerkung: Den Rechenvariablen kann zur besseren Dokumentation der Ausgabe ein Kommentar angefügt werden (max. 19-stellige alphanumerische Zeichenfolge).

Folgende Zeichen sind erlaubt:

Ziffern:	`0 - 9`
Buchstaben:	`A - Z, Ä, Ö, Ü`
Sonderzeichen:	`-,+,!,",@,$,%,&,/,?,',_`

Folgende Zeichen sind nicht erlaubt: `=,#,*,:,'Space',(,)`

Summarisches Rechnen

1. Operationsergänzung S - Das Ergebnis wird in einem Feld abgelegt.

   Der im Kommando angegebene Operand bzw. Ausdruck, wird für jeden angesprochenen Satz berechnet und zum bestehenden Wert des jeweiligen Feldes addiert. Die im Ausdruck eventuell vorkommenden Felder werden vorher mit den Daten des aktuellen Satzes belegt.

   Mit diesem neuen Feld pro Satz erfolgt ein Update in der Haupt- bzw. Verweisdatei.

   Die Mengenangabe bestimmt, auf wieviele Sätze das RECHNE-Kommando angewendet wird.

   Beispiel:

   ```
   RE,S,SUMME=SUMME
   ```

   ist gleichbedeutend mit:
   ```
   RE,Z,SUMME=SUMME+SUMME  oder
   RE,Z,SUMME=SUMME*2
   ```

2. Operationsergänzung S - Das Ergebnis wird in einer Rechenvariablen abgelegt.

   Der im Kommando angegebene Ausdruck wird für jeden angesprochenen Satz berechnet und zum jeweilig bestehenden Wert der Rechenvariablen addiert.

   Die im Ausdruck eventuell vorkommenden Felder werden vorher mit den Daten des aktuellen Satzes belegt.

   Die Mengenangabe bestimmt, auf wieviele Sätze das RECHNE-Kommando angewendet wird.

Beispiel 1: Variante-1:

```
*
RE %XS=0
SU GESAMTPREIS=0
IM00    ANZAHL ZIELINFORMATIONEN: 5
RE,$,Z,GESAMTPREIS=ANZAHL*EINZELPREIS
BL,$,R
Z ANZAHL EINZELPREIS GESAMTPREIS
ANZAHL EINZELPREIS GESAMTPREIS
     1     5000.00     5000.00
     2     2500.00     5000.00
     3      100.00      300.00
     4       25.00      100.00
     5       20.00      100.00
BL,$,R
RE,5,S,%XS=GESAMTPREIS
Z,1,T,%XSUMME
%XSUMME
10500.00
```

Variante-2:

```
*
RE %XS=0
SU GESAMTPREIS=0
IM00    ANZAHL ZIELINFORMATIONEN: 5
RE,$,Z,GESAMTPREIS=ANZAHL*EINZELPREIS,S,%XS=GESAMTPREIS
BL,$,R
Z,$,T,ANZAHL EINZELPREIS GESAMTPREIS %XSUMME
ANZAHL EINZELPREIS GESAMTPREIS   %XSUMME
     1     5000.00     5000.00 105000.00
     2     2500.00     5000.00 105000.00
     3      100.00      300.00 105000.00
     4       25.00      100.00 105000.00
     5       20.00      100.00 105000.00
*
```

Beispiel 2:

```
*
SU GEBJAHR=1930,BIS GEBJAHR=1960
IM00    ANZAHL ZIELINFORMATIONEN: 78
RE %XD=0
RE %YA=(2,0)
RE,$,Z,%YA=1994-GEBJAHR,S,%XD=%YA
RE,Z,%XD=%XD/%AZI
BL,$,R
Z,4,T,%YALTER NAME %XD
%YALTER NAME            %XD
     47 MAIER         33.25
     27 HUBER         33.25
     30 MUELLER       33.25
     35 BAUMANN       33.25
```

2.20 SETZEN-Kommando

2.20.1 SETZEN-G/K (Groß-/Kleinschreibung)

Anwendungsart: aktiv/passiv

```
SETZEN,{G/K}[,DB.xxxxxx]
```

SETZEN	Operation für Setzen.
G/K	Operationsergänzung für Groß-/Kleinschreibung.
xxxxxx	Satzbeschreibung

Dieses Kommando dient dazu, die Groß-/Kleinschreibung in CIS, unabhängig vom Eintrag in CISVARI/1+X'18E', während des laufenden Betriebes einzuschalten.

Dialog-Ein-/Ausgaben von CIS werden stets über die Ein- /Ausgabetabellen transformiert (vgl. Manual-4).

G: Tabellenpaar 2 (CISI+X'400'/CISI+X'500') wird verwendet.

Standardmäßig, d.h. falls die Tabellen vom Anwender nicht verändert wurden, werden alle Kleinbuchstaben in Großbuchstaben umgesetzt.

K: Tabellenpaar 1 (CISI+X'200'/CISI+X'300') wird verwendet.

Standardmäßig, d.h. falls die Tabellen vom Anwender nicht verändert wurden, werden alle Zeichen so übernommen wie sie eingegeben wurden.

2.20.2 SETZEN-E/I (Externes Format/Internes Format)

Anwendungsart: passiv

```
SETZEN,{E/I}[,DB.xxxxxx]
```

SETZEN	Operation für Setzen.
E/I	Operationsergänzung für externes/internes Format.
xxxxxx	Satzbeschreibung

SETZEN,I ermöglicht es, die Eingabetransformation für Feldwerte zu umgehen. Felder können wie im jeweiligen Programm definiert und ohne Umsetzung in das externe Format in das CIS-Kommando geschrieben werden.

Beispiel: Suchkommando für komplizierten Compound-Key, der aus zwei gepackten Zahlen und 3 Byte Text besteht:

```
                    ■ ■ ■ ■ ■ ■ A B C     externes Format
SU COMP=00012C00004FC1C2C3                internes Format
```

SETZEN,E ist der Normalfall, d.h. die Werte müssen abdruckbar eingegeben werden und werden von der CIS-Syntax in die jeweiligen Formate umgesetzt.

2.20.3 SETZEN-LJ/LN (LOGADR)

Anwendungsart: aktiv/passiv

```
SETZEN,{LJ/LN}[,DB.xxxxxx]
```

SETZEN	Operation für Setzen.
LJ	Operationsergänzung für internes Setzen von LOGADR.
LN	Operationsergänzung damit LOGADR intern nicht gesetzt wird.
xxxxxx	Satzbeschreibung

Das Kommando kann nur bei binärem Ordnungsbegriff ≤ 4 Bytes benützt werden.

Das Kommando hat Priorität gegenüber der Angabe im A-Segment.

2.20.4 SETZEN-M (Modus)

Anwendungsart: aktiv

```
SETZEN,{MN/MS/MQ}[,DB.xxxxxx]
```

SETZEN	Operation für Setzen.	
MN	Operationsergänzung für Modus normal.	
MS	Operationsergänzung für Modus spezial.	
MQ	Operationsergänzung für die Abfrage des Modus.	IM00=N IM01=S
xxxxxx	Satzbeschreibung	

Mit diesem Kommando wird das Format der Ausgabenachricht festgelegt. Im Teilhaberbetrieb gilt das Format nur für die jeweilige Station.

Normal-Modus: Die Ausgabe ist entweder die Antwort auf das CIS-Kommando, eine AZI-Meldung oder eine Fehlermeldung.

Spezial-Modus: Der Ausgabe sind immer 4 Bytes CM vorangestellt.

Bei Antwort auf CIS-Kommando: `IM00` oder `IM01`

Bei AZI-Meldung: `CCCC000000000000000`

IM80 Anzahl ZI mit führenden
81 Nullen (15 - stellig)
90
91

Bei Fehlermeldung: `CCCCCCCC_ _ _ _Text`

Präfix normale Fehlermeldung

Nutzen: Ist die CIS-Station nicht ein Terminal, sondern ein Programm (wie z.B. ein Client in SINIX oder MS-Windows), so kann durch Abfrage der ersten vier Bytes erkannt werden, ob es sich um die erwartete Antwort, oder um eine Fehlermeldung handelt.

2.20.5 SETZEN-S (sequentiell)

Anwendungsart: aktiv/passiv

```
SETZEN,S[,DB.xxxxxx]
```

SETZEN	Operation für Setzen.
S	Operationsergänzung für sequentiell.
xxxxxx	Satzbeschreibung

Dieses Kommando bewirkt, daß die darauf folgende Initial- bzw. Folgefrage in der Hauptdatei sucht, auch wenn eine Verweisdatei vorhanden ist. Eine Ausnahme bilden Pseudofeld und Compound-Key, die nur in der Verweisdatei definiert sind. Sie können nur über die Verweisdatei gesucht werden.

Bei UND- oder NICHT-Folgefragen, wird nur die Menge der Zielinformationen der aktuellen Zielpunktliste berücksichtigt, d.h. für den Vergleich werden die Sätze der Hauptdatei, die durch die ZPL angegeben sind, gelesen.

Es kann also der Vergleich auf wenige HD-Sätze beschränkt werden, wenn die Datenmenge durch vorangehende Verknüpfungen bereits stark eingeschränkt ist und die folgende einschränkende Frage (UND/NICHT) eine große Menge Treffer von der VD bringen würde (geringe Streuung).

Die Anwendung des Kommandos vor einer ODER-Folgefrage ist sinnlos, da sie sich auf den gesamten Datenbestand bezieht.

Bei Stichwortinvertierung, Space- oder Nullwertunterdrückung kann mit Hilfe dieses Kommandos nach Feldinhalten, die nicht invertiert wurden, gesucht werden.

Die Wirksamkeit dieses Kommandos wird nach der nächsten Initial- oder Folgefrage aufgehoben.

Beispiel:

Initialfrage

```
*
SETZE,S
*
SUCHE A=1
IM01    ANZAHL ZIELINFORMATIONEN: 100              Suche in der HD
*
SUCHE A=1
IM00    ANZAHL ZIELINFORMATIONEN: 100              Suche über die VD
*
```

Folgefrage

```
*
SU NAME(FP=$)=SCHMI
IM00    ANZAHL ZIELINFORMATIONEN: 4312
*
UND WOHNORT=FREISING
IM00    ANZAHL ZIELINFORMATIONEN: 9
*
SE,S
*
UND VORNAME=XAVER
IM01    ANZAHL ZIELINFORMATIONEN: 1
*
SU VORNAME=XAVER
IM00    ANZAHL ZIELINFORMATIONEN: 10335
*
```

Wenn vor der letzten Folgefrage der Suchpfad nicht sequentiell gesetzt wird, müssen intern 10335 Verweise verglichen werden, ob das Feld VORNAME den Wert XAVER enthält. Mit SETZEN-S müssen nur die bereits lokalisierten neun Sätze verglichen werden (sie werden dabei direkt über den Ordnungsbegriff gelesen), was zu erheblicher Verkürzung der Antwortzeit führt.

2.20.6 SETZEN-TZ/TE (Zwischen-/Endetrenner)

Anwendungsart: aktiv/passiv

```
SETZE,{TZ/TE},{C'c'/X'xx'}[,DB.xxxxxx]
```

SETZEN	Operation für Setzen.	
TZ	Operationsergänzung für Zwischentrenner.	Standardwert: `C';'`
TE	Operationsergänzung für Endetrenner.	Standardwert: `C':'`
`C'c'`	Zeichen abdruckbar.	
`X'xx'`	Zeichen, hexadezimale Darstellung.	
`xxxxxx`	Satzbeschreibung	

Es werden für alle Aktivausgaben, wie z.B.: GET,KFP GET,FBA ..., die Trenner gesetzt.

2.20.7 SETZEN-WE/WV (Wildcard)

Anwendungsart: aktiv/passiv

```
SETZE,{WE/WV},{C'c'/X'xx'}[,DB.xxxxxx]
```

SETZEN	Operation für Setzen.
WE	Operationsergänzung für Wildcard ein Zeichen Standardwert: '?'
WV	Operationsergänzung für Wildcard viele Zeichen Standardwert: '*'
C'c'	Zeichen abdruckbar.
X'xx'	Zeichen, hexadezimale Darstellung.
xxxxxx	Satzbeschreibung

Beispiel: SU NAME=ME?ER

SU NAME=*ALT* (arbeitet wie bisher (FP=$)=ALT)

2.21 SICHERN-Kommando

Die einzelnen Arten des SICHERN-Kommandos dienen der temporären bzw. permanenten Sicherung von Suchergebnissen (Zielpunktlisten). Die einzelnen Kommandos unterscheiden sich aber in Anwendungs- und Ausführungsweise und müssen daher getrennt betrachtet werden.

2.21.1 SICHERN-E (Temporäre ZPL-Sicherung)

Anwendungsart: aktiv/passiv

```
SICHERN{,E,/_}Feld[(Erg)]=Wert[,DB.xxxxxx]
```

SICHERN	Operation für Sichern.
E	Operationsergänzung für temporär.
Feld=Wert	Kennzeichnung der Zielpunktliste.
Ergänzung (vgl. Seite 18)	DB = Name der Satzbeschreibung. Nur beim Arbeiten im Fenster.
xxxxxx	Satzbeschreibung

Eine erstellte Zielpunktliste wird in eine temporäre Datei gesichert, wobei der Stand des Zielpunktlistenzeigers mitgesichert wird. Eine derart gesicherte Zielpunktliste kann, im Gegensatz zu SICHERN-V (vgl. Seite 134), nur während des Ablaufs von CIS mit dem AKTIVIEREN-Kommando (vgl. Seite 40) wieder verwendet werden. Für die Sicherung kann jede Feldbezeichnung der aktuellen Datenbeschreibung benützt werden. Der angegebene Feldinhalt muß nur dem Format des verwendeten Feldes genügen. Wird ein numerisches Feld zur Sicherung verwendet, so sind die Hinweise im Kapitel "Behandlung numerischer Feldinhalte" (vgl. Seite 21) zu berücksichtigen.

Mit dem ZEIGE-E Kommando (vgl. Seite 208) läßt sich eine Auskunft über alle bisher temporär gesicherten ZPL's ausgeben.

Nicht mehr benötigte Zielpunktlisten sollten mit dem FREIGEBEN-Kommando (vgl. Seite 74) freigegeben werden.

Beispiel:

```
*
SUCHE GEHALT>2000,DB.PERSON
IM00    ANZAHL ZIELINFORMATIONEN: 1000
*
SICHERN GEHALT=1,DB.PERSON
*
SUCHE GEHALT<2000,DB.PERSON
IM00    ANZAHL ZIELINFORMATIONEN: 500
*
SICHERN,E,GEHALT=2
*
AKTIVIEREN GEHALT=1
*
```

Jetzt kann mit der 1. ZPL weitergearbeitet werden.

2.21.2 SICHERN-I (VD-Zeiger)

Anwendungsart: aktiv/passiv

```
SICHERN,I,Feld[(Erg)]=Wert[,DB.xxxxxx]
```

SICHERN	Operation für Sichern.
I	Operationsergänzung für indirekten VD-Zeiger.
Feld=Wert	Kennzeichnung der Sicherung.
Ergänzung (vgl. Seite 18)	DB = Name der Satzbeschreibung. Nur beim Arbeiten im Fenster.
xxxxxx	Satzbeschreibung

Falls ein VD-Zeiger existiert (vgl. BLAETTERN-I Seite 42 oder GET-KI Seite 90) wird er temporär gesichert und kann mit AKTIVIERE-I (vgl. Seite 40) zu einem späteren Zeitpunkt aktualisiert werden.

Mit dem ZEIGE-E Kommando (vgl. Seite 208) läßt sich eine Auskunft über alle bisher temporär gesicherten VD-Zeiger ausgeben.

2.21.3 SICHERN-J (ZPL für VERBINDE)

Anwendungsart: aktiv/passiv

```
SICHERN,J,Feld[(Erg)][,DB.xxxxxx]
```

SICHERN	Operation für Sichern.
J	Operationsergänzung für ZPL zum VERBINDE.
Feld	Kennzeichnung der Sicherung.
Ergänzung (vgl. Seite 18)	FP=Position im Textfeld. FL=(Teil-) Länge eines Textfeldes.
xxxxxx	Satzbeschreibung

SI,J ist nur mit Dateipaßwort gültig. Wird SI,J im Fenster verwendet, so wird die Fehlermeldung ZP22 ausgegeben. Eine Zielpunktliste wird zu Verbinden bereitgestellt.

Beispiel:

```
SU AGTNR=123456,DB.xxxxxx
IM00    ANZAHL ZIELINFORMATIONEN: 5
SI,J,AGTNR
SU WERT2>10000,DB.yyyyyy
IM00    ANZAHL ZIELINFORMATIONEN: 40
SI,J,KAGTNR(FP=3,FL=6)
VE AGTNR(DB=xxxxxx)=KAGTNR(DB=yyyyyy),DB.JOIN01
```

Durch das SI,J ist für alle beteiligten Datenbanken (xxxxxx, yyyyyy) eine Einschränkung gegeben. Statt der Verweisdatei werden die mit SI,J zur Verfügung gestellten Einträge zum Verbinden herangezogen.

2.21.4 SICHERN-S (Suchergebnis)

Anwendungsart: aktiv/passiv

```
SICHERN,S[,DB.xxxxxx]
```

SICHERN	Operation für Sichern.
S	Operationsergänzung für Suchergebnis.
xxxxxx	Satzbeschreibung

Der einmalige Aufruf dieses Kommandos sichert das jeweilige Suchergebnis (Zielpunktliste) einer Initial- oder Folgefrage temporär. Dadurch kann mit Hilfe des IGNORIEREN-S Kommandos (vgl. Seite 98) das vorletzte gesicherte Suchergebnis aktualisiert werden.

Dieses Kommando wird meist aktiv angewendet wenn das letzte Suchergebnis durch das vorletzte ersetzt werden soll.

Die Suchergebnisse werden solange gesichert bis ein FREIGEBE-S Kommando (vgl. Seite 77) gegeben wird.

Mit dem ZEIGE-E Kommando (vgl. Seite 208) läßt sich eine Auskunft über alle bisher temporär gesicherten ZPL's ausgeben.

2.21.5 SICHERN-V (Verweisdatei)

Anwendungsart: aktiv/passiv

```
SICHERN,V,Pseudofeld=Wert[,DB.xxxxxx]
```

SICHERN	Operation für Sichern.
V	Operationsergänzung für Verweisdatei.
Pseudofeld=Wert	Kennzeichnung der Zielpunktliste.
xxxxxx	Satzbeschreibung

Eine erstellte Zielpunktliste, die nicht im Fenster erstellt wurde, wird in der Verweisdatei gesichert.

Voraussetzung dafür ist, daß das zur Sicherung verwendete Feld in der Datenbeschreibung als Pseudofeld deklariert ist. Ist das Pseudofeld mit der logischen Feldbedeutung P definiert, so darf die Zielpunktliste nicht sortiert sein. Soll eine sortierte Zielpunktliste gesichert werden, muß das Pseudofeld mit der logischen Feldbedeutung S definiert sein.

Ein Pseudofeld kann an beliebiger Stelle in der Datenbeschreibung definiert sein. Es belegt keinen Speicherplatz im Hauptdateisatz, sondern dient nur der Sicherung in der Verweisdatei.

Der bei der Sicherung angegebene Feldinhalt muß den Konventionen der Feldbedeutung des Pseudofeldes genügen.

Die Sicherung ist permanent, d.h. die gesicherte Zielpunktliste wird im Gegensatz zur temporären Sicherung (vgl. Seite 129) nicht am Ende der Laufzeit von CIS automatisch gelöscht. Soll die Zielpunktliste gelöscht werden, muß sie explizit freigegeben werden (vgl. FREIGEBEN-Kommando Seite 74).

Bei Erstellung einer neuen VD verschwinden gesicherte ZPL's natürlich ebenfalls.

Bei der Sicherung wird der momentane ZPL-Zeigerstand nicht mitgesichert.

Die gesicherte Zielpunktliste wird mit dem SUCHE-Kommando aktiviert:

```
SUCHE Pseudofeld=Wert[,DB.xxxxxx]
```

Beispiel:

```
*
SUCHE GEHALT=1000,BIS GEHALT=2000,DB.PERSON
IM00    ANZAHL ZIELINFORMATIONEN: 763
*
SICHERN,V,PSEUDO=1
*
SU BERUF=SACHBEARBEITER
IM01    ANZAHL ZIELINFORMATIONEN: 1231
*
UND PSEUDO=1
IM00    ANZAHL ZIELINFORMATIONEN: 436
*
HALT

/START-PROGRAM FROM-FILE=CIS
SU PSEUDO=1,DB.PERSON
IM00    ANZAHL ZIELINFORMATIONEN: 763*
UND ORT(FP=$)=MUENCHEN
IM01    ANZAHL ZIELINFORMATIONEN: 375
*
```

PSEUDO ist ein Pseudofeld der aktuellen Datenbeschreibung.

Aktivierung der ZPL aus dem letzten CIS-Lauf.

2.22 SORTIERE-Kommando

Anwendungsart: aktiv/passiv

```
SORTIERE{_/,{S/F/E},}Feld[(Erg)]_...[,DB.xxxxxx]
```

SORTIERE	Operation für Sortieren.
S	Operationsergänzung für steigend (Standardwert).
F	Operationsergänzung für fallend.
E	Operationsergänzung für eindeutig.
Feld	Feldname des zu sortierenden Feldes.
Ergänzung (vgl. Seite 18)	AM = Abschnittsmultiplität: Bei fehlender AM-Angabe wird AM=1 angenommen. FM = Feldmultiplität: Bei fehlender FM-Angabe wird FM=1 angenommen FL = Feldlänge: Bei fehlender FL-Angabe wird in definierter Feldlänge sortiert. FP = Feldposition: FP=$ darf nicht verwendet werden. Bei fehlender FP-Angabe wird FP=1 angenommen. DB = Name der Satzbeschreibung: Nur beim Arbeiten im Fenster. SO = UF Ergänzung für Sortierung in absteigender Reihenfolge entsprechend einer USER-Tabelle. SO = US Ergänzung für Sortierung in aufsteigender Reihenfolge entsprechend einer USER-Tabelle. SO = F Ergänzung für Sortierung in absteigender Reihenfolge. SO = S Ergänzung für Sortierung in aufsteigender Reihenfolge.
xxxxxx	Satzbeschreibung

Mit dem SORT-Kommando können Sätze aufsteigend, absteigend oder eindeutig nach einem oder mehreren Feldern sortiert werden. Zusätzlich kann für jedes einzelne Feld eine Ergänzung angegeben werden, die die Sortierung für dieses Feld bestimmt. Diese SO-Ergänzung hat Priorität vor der Operationsergänzung.

Die Ergänzungen zu den Feldern können unterschiedlich sein. Es ist möglich, den SORT-Auftrag so zu formulieren, daß in der Datenbeschreibung definierte Angaben (Feldlänge oder Feldposition) modifiziert werden können.

Unabhängig vom ZPL-Zeigerstand wird die gesamte ZPL sortiert. Das Ergebnis des SORT ist eine Zielpunktliste, die für entsprechende Kommandos (z.B. Ausgabe) zur Adressierung verwendet wird. Eine Folgeverknüpfung mit dieser Zielpunktliste ist erst dann wieder möglich, wenn vor der Verknüpfung steigend nach dem Ordnungsbegriff (SORT Ordnungsbegriff) sortiert wird.

Ist als Operationsergänzung E angegeben (eindeutig sortieren), wird bei gleichen Feldinhalten nur jeweils der erste Satz der alten Zielpunktliste für die Erstellung der neuen Zielpunktliste verwendet. Die Sätze liegen dann, verringert um die Sätze mit gleichen Feldinhalten, aufsteigend sortiert vor.

Die USER-Tabelle ermöglicht es dem Anwender auch Umlaute und Sonderzeichen in gewünschter Reihenfolge zu sortieren. Die Vorgehensweise ist im Manual 4, Kapitel "Sortieren mit eigenen Benutzertabellen", beschrieben.

Beispiele:

Sortierung steigend

```
*
SU VORNAME=HANS
IM00    ANZAHL ZIELINFORMATIONEN: 98
Z,3,T,VORNAME NAME
VORNAME NAME
HANS    MEIER
HANS    GRUBER
HANS    MUELLER
SORT NAME(FL=3)
IM01    ANZAHL ZIELINFORMATIONEN: 98
Z,3,T,VORNAME NAME
VORNAME NAME
HANS    ALBERS
HANS    BRAUN
HANS    BRANDT
*
```

Es wurde nur nach den ersten 3 Bytes des Feldes NAME sortiert.

Sortierung fallend

```
*
SORT,F,NAME KUNDENNR(FP=3,FL=3)
IM01    ANZAHL ZIELINFORMATIONEN: 337
Z,3,T,NAME VORNAME KUNDENNR
NAME       VORNAME      KUNDENNR
ZECHBAUER  ANTON           43939
ZECHBAUER  XAVER           91931
WAGNER     HUGO            43875
*
```

Sortierung eindeutig

```
*
SU NAME=MUELLER
IM00    ANZAHL ZIELINFORMATIONEN: 3
*
ODER NAME=HUBER
IM00    ANZAHL ZIELINFORMATIONEN: 5
ODER NAME=MEIER
IM00    ANZAHL ZIELINFORMATIONEN: 7
Z,$,T,NAME VORNAME WOHNORT ORDNUNGSBEGRIFF
```

Die Feldinhalte werden unsortiert in der Reihenfolge ausgegeben, so wie die Sätze in der Hauptdatei stehen (vgl. Ordungsbegriffe).

```
NAME     VORNAME    WOHNORT   ORDNUNGSBEGRIFF
HUBER    KLAUS      GLONN                3
MUELLER  MARGOT     MUENCHEN            36
MEIER    ELISABETH  ANZING              89
MUELLER  HANS       MUENCHEN           123
MEIER    HANS       MUENCHEN           156
MUELLER  EVA        AUBING             178
HUBER    FRANZ      GAUTING            216
SORT,E,NAME
IM01    ANZAHL ZIELINFORMATIONEN: 3
Z,$,T,NAME VORNAME WOHNORT ORDNUNGSBEGRIFF
NAME     VORNAME    WOHNORT   ORDNUNGSBEGRIFF
HUBER    KLAUS      GLONN                3
MEIER    ELISABETH  ANZING              89
MUELLER  MARGOT     MUENCHEN            36
*
```

Nur jeweils der erste Satz mit dem gleichen Feldinhalt (Name) wird zur Erstellung der neuen Zielpunktliste verwendet. Die Zielpunktliste wird also um die vier Sätze mit jeweils gleichen Feldinhalten verringert.

```
SU NAME=MUELLER,OD NAME=HUBER,OD NAME=MEIER
IM00    ANZAHL ZIELINFORMATIONEN: 7
SO NAME VORNAME                                          Aufsteigende Sortierung
IM01    ANZAHL ZIELINFORMATIONEN: 7
Z,$,T,NAME VORNAME WOHNORT ORDNUNGSBEGRIFF
NAME     VORNAME    WOHNORT   ORDNUNGSBEGRIFF
HUBER    FRANZ      GAUTING               216
HUBER    KLAUS      GLONN                   3
MEIER    ELISABETH  ANZING                 89
MEIER    HANS       MUENCHEN              156
MUELLER  EVA        AUBING                178
MUELLER  HANS       MUENCHEN              123
MUELLER  MARGOT     MUENCHEN               36
SORT,E,NAME
IM01    ANZAHL ZIELINFORMATIONEN: 3
Z NAME VORNAME WOHNORT ORDNUNGSBEGRIFF
NAME     VORNAME    WOHNORT   ORDNUNGSBEGRIFF
HUBER    FRANZ      GAUTING               216
MEIER    ELISABETH  ANZING                 89
MUELLER  EVA        AUBING                178
*
```

```
SU NAME=MUELLER,OD NAME=HUBER
IM00    ANZAHL ZIELINFORMATIONEN: 5
SORT,E,NAME WOHNORT(FP=1,FL=1)
IM01    ANZAHL ZIELINFORMATIONEN: 3
Z NAME VORNAME WOHNORT ORDNUNGSBEGRIFF
NAME     VORNAME    WOHNORT   ORDNUNGSBEGRIFF
HUBER    KLAUS      GLONN                   3
MUELLER  EVA        AUBING                178
MUELLER  MARGOT     MUENCHEN               36
*
```

Sortierung steigend und fallend

```
*
SU VERTRETERNAME=BERGER,OD VERTRETERNAME=FREI
IM00    ANZAHL ZIELINFORMATIONEN: 5
Z,$,T,VERTRETERNAME UMSATZ ORDNUNGSBEGRIFF
VERTRETERNAME  UMSATZ  ORDNUNGSBEGRIFF
BERGER         200.00               2
FREI           100.00               7
FREI           400.00               9
BERGER         700.00              12
BERGER         100.00              15
SORT VERTRETERNAME(SO=S) UMSATZ(SO=F)
IM01    ANZAHL ZIELINFORMAIONEN: 5
Z,$,T,VERTRETERNAME UMSATZ ORDNUNGSBEGRIFF
VERTRETERNAME  UMSATZ  ORDNUNGSBEGRIFF
BERGER         700.00              12
BERGER         200.00               2
BERGER         100.00              15
FREI           400.00               9
FREI           100.00               7
```

Die Zielpunktliste (bzw. Hauptdatei) wird aufsteigend nach VERTRETERNAME sortiert und innerhalb des gleichen Vertreters absteigend nach UMSATZ. Die höchsten Umsätze des jeweiligen Vertreters werden also jeweils nach vorne sortiert.

2.23 SPERRE-Kommando

Das SPERRE-Kommando ist nur innerhalb von Transaktionen erlaubt und notwendig (vgl. Seite 171). Innerhalb einer Transaktion kann ein Update bereits vorhandener Daten nur über gesperrte Zielpunktlisten, Sätze oder Dateien erfolgen, d.h. vor einem Update muß ein SPERRE-Kommando oder ein anderes Kommando mit impliziter Sperrfunktion (wie z. B. GET-KS) durchgeführt worden sein.

Das SPERRE-Kommando wirkt nicht auf die Zielpunktliste bzw. Datei direkt, sondern auf ein erneutes SPERRE-Kommando.

Das hat zur Folge, daß zur selben Zeit von einem weiteren Anwender keine Änderungen an denselben Sätzen vorgenommen werden können (er müßte dazu ein SPERRE-Kommando absetzen). Die Datensicherheit ist dadurch gewährleistet.

Ein SPERRE-Kommando vor Lesezugriffen verhindert, daß Sätze gelesen werden, die möglicherweise gerade geändert werden.

Die Sperre wirkt solange, bis das Transaktionsende erreicht (d.h. kurze Transaktionen, nur die notwendigen Sätze sperren!) oder die Transaktion abgebrochen wird.

Ebenso können mit dem SUCHE-Kommando und mit dem GET-Kommando Sätze gesperrt werden (vgl. Seite 146 und 78).

2.23.1 SPERRE-D (Datei)

Anwendungsart: aktiv/passiv

```
SPERRE,D[,DB.xxxxxx]
```

SPERRE	Operation für Sperren.
D	Operationsergänzung für Datei.
xxxxxx	Satzbeschreibung

Mit diesem Kommando wird die gesamte Hauptdatei während der Transaktion gesperrt.

Die Datei wird bei Transaktionsende oder Transaktionsabbruch freigegeben.

Wenn diese Datei oder mindestens ein Satz der Datei schon von einer anderen Transaktion gesperrt ist, wird die Meldung SP04 zurückgemeldet.

2.23.2 SPERRE-KD (Satz direkt)

Anwendungsart: aktiv/passiv

```
SPERRE,KD,Key=Wert[,DB.xxxxxx]
```

SPERRE	Operation für Sperren.
KD	Operationsergänzung für Satz direkt.
Key	Ordnungsbegriff der Datei.
Wert	Schlüssel der Hauptdatei.
xxxxxx	Satzbeschreibung

Mit diesem Kommando wird der Satz mit dem Ordnungsbegriff=Wert, wie im Kommando angegeben, während der Transaktion gesperrt.

Es wird nicht geprüft, ob dieser Satz in der Hauptdatei auch tatsächlich vorhanden ist. Es wird geprüft, ob dieser Satz bereits von einer anderen Transaktion gesperrt wurde. Ist dieser Satz bereits gesperrt, so wird die Fehlermeldung SP04 ausgegeben.

Dieses Kommando ist z.B. notwendig, wenn mit dem Kommando GET-KD (vgl. Seite 87) und einem anschließenden AENDERN (Update) gearbeitet wird, oder generell, wenn die Verarbeitung nicht über eine Zielpunktliste erfolgt.

Der Satz wird bei Transaktionsende oder Transaktionsabbruch freigegeben.

2.23.3 SPERRE-Z (Zielpunktliste)

Anwendungsart: aktiv/passiv

```
SPERRE,Z[,DB.xxxxxx]
```

SPERRE	Operation für Sperren.
Z	Operationsergänzung für Zielpunktliste.
xxxxxx	Satzbeschreibung

Mit diesem Kommando wird die aktuelle Zielpunktliste gesperrt. Wenn mindestens ein Satz der Zielpunktliste oder die gesamte Datei bereits von einer anderen Transaktion gesperrt ist, wird die Meldung SP04 ausgegeben.

Es können beliebig viele Zielpunktlisten zu beliebigen Zeitpunkten gesperrt werden. Es sollten aber nur die gesperrt werden, bei denen es notwendig ist, da möglicherweise der Zugriff anderer Anwender unnötig verhindert wird.

Die Zielpunktlisten werden beim Ende oder Rücksetzen der Transaktion von CIS freigegeben.

Solange eine gesperrte Zielpunktliste nicht weiter bearbeitet wird (z.B. Sortieren, Folgesuchfrage ...), ist sie als gesperrt gekennzeichnet.

Nachfolgende Update-Kommandos mit dieser ZPL werden sofort durchgeführt. Ist jedoch kein Sperrkennzeichen mehr vorhanden, so wird CISKOOR veranlaßt zu prüfen, ob der entsprechende Ordnungsbegriff gesperrt ist. (Performance-Frage!!)

2.24 STATUS-Kommando

Das STATUS-Kommando bietet die Möglichkeit, Auskünfte über die Anzahl gesperrter Dateien bzw. gesperrter Sätze einzuholen. Voraussetzung für das Kommando ist, daß mit Transaktionen (vgl. Seite 171) gearbeitet wird.

2.24.1 STATUS-D (Dateien)

Anwendungsart: aktiv/passiv

```
STATUS,{DG/DT}[,DB.xxxxxx]
```

STATUS	Operation für Abfrage.
DG	Operationsergänzung für Dateien gesamt.
DT	Operationsergänzung für Dateien der aktuellen TID (Transaktions-ID).
xxxxxx	Satzbeschreibung

Mit diesem Kommando wird die Anzahl der gesperrten Dateien abgefragt. Die Anzahl wird über das Feld AZI (Anzahl Zielinformationen) gemeldet.

2.24.2 STATUS-S (Sätze)

Anwendungsart: aktiv/passiv

```
STATUS,{SD/ST/SA/SG}[,DB.xxxxxx]
```

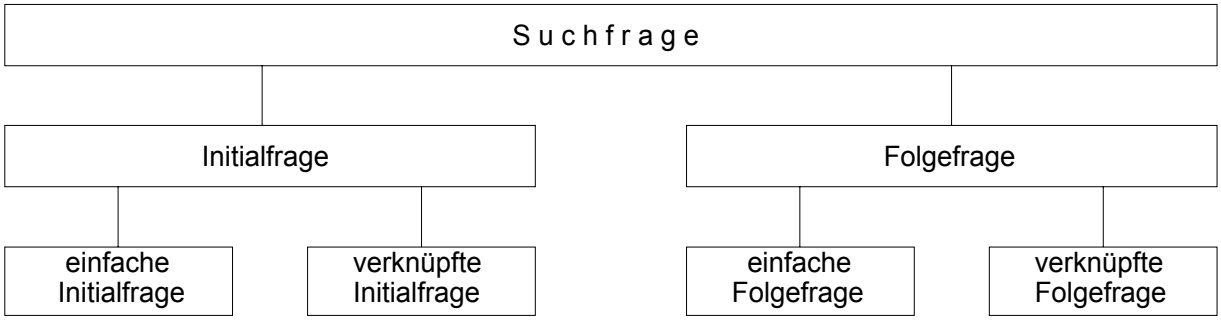
STATUS	Operation für Abfrage.
SD	Operationsergänzung für Sätze aus der aktuellen Hauptdatei - gilt für alle TID's (Transaktions-ID).
ST	Operationsergänzung für Sätze der aktuellen TID - gilt für alle Dateien.
SA	Operationsergänzung für Sätze aus der aktuellen Hauptdatei und der aktuellen TID.
SG	Operationsergänzung für Sätze gesamt - alle Dateien und TID's.
xxxxxx	Satzbeschreibung

Mit diesem Kommando wird die Anzahl der gesperrten Sätze abgefragt. Die Anzahl wird über das Feld AZI (Anzahl Zielinformationen) gemeldet.

2.25 SUCHE-Kommando

Das Ergebnis einer Suchfrage ist die Anzahl der gefundenen Sätze (Zielinformationen). Intern wird eine Zielpunktliste erstellt, die zur Adressierung der gefundenen Sätze dient.

Der Aufbau einer Suchfrage läßt sich aus folgender Struktur ersehen:

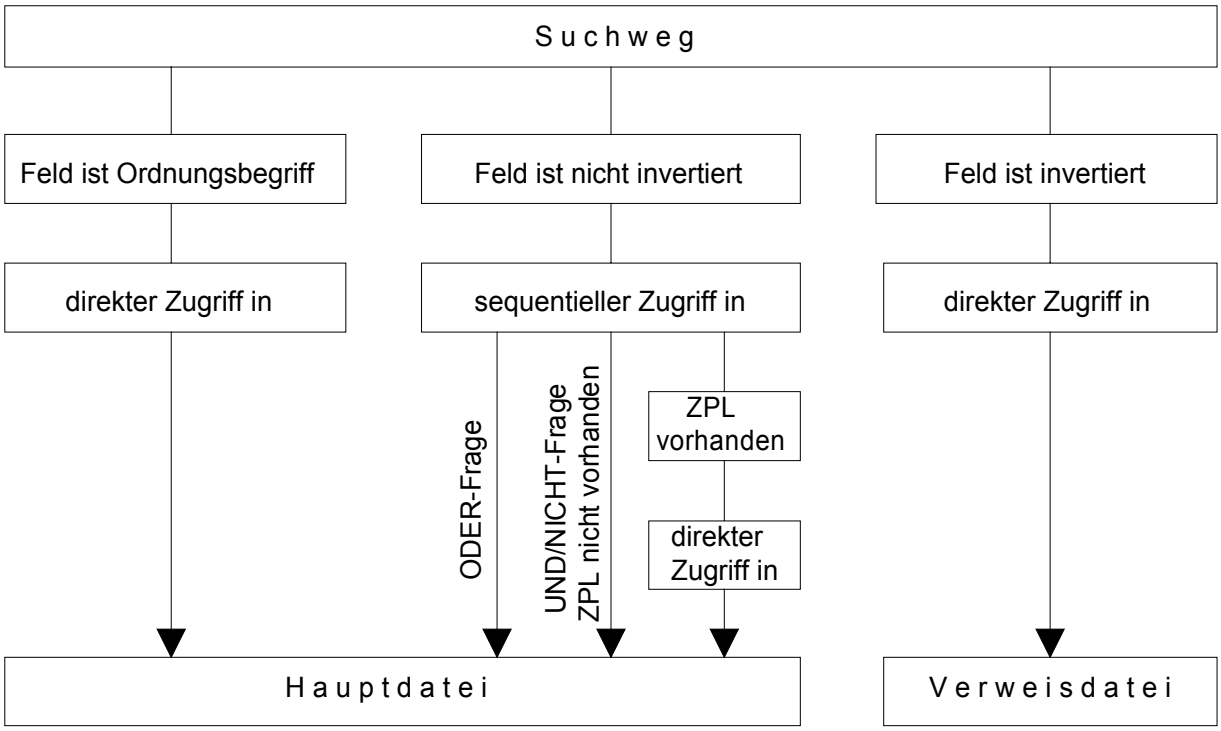


Frage	Art	Operator/Operatorenfolge
Initialfrage	einfache	`SUCHE ...`
	verknüpfte	`SUCHE ... , [ UND | ODER | NICHT ] ... , [ UND | ODER | NICHT ] ...`
Folgefrage	einfache	`[ UND | ODER | NICHT ] ...`
	verknüpfte	`[ UND | ODER | NICHT ] ... , [ UND | ODER | NICHT ] ...`

Die Anzahl der Verknüpfungen innerhalb einer Initial- oder Folgefrage ist abhängig von der Länge des Kommandos. Die Kommandolänge darf maximal 500 Bytes betragen und hängt von der maximalen Länge des 'formatierten Auftrags' ab, der in CISVARI+X'DE' mit dem Standardwert C'2048' vorbelegt ist und maximal C'8192' betragen kann.

Die Verknüpfungsoperatoren (UND, ODER, NICHT) können innerhalb einer verknüpften Frage unterschiedlich sein. Sie werden in der angegebenen Reihenfolge von links nach rechts abgearbeitet. Sowohl bei Initial- wie bei Folgefragen sind Bereichsangaben erlaubt.

Der Suchweg richtet sich nach den Angaben in der Datenbeschreibung. Er kann sequentiell, direkt oder indirekt sein. Bei verknüpften Suchfragen können gemischte Suchwege vorkommen. Folgendes Diagramm erläutert die möglichen Suchwege:



Bereichsfragen, d.h. Fragen mit einem Vergleichsoperator ungleich =, bewirken als erstes einen direkten Zugriff (auf HD bzw. VD). Anschließend wird sequentiell weitergearbeitet.

Der direkte Suchweg auf der HD (über den Ordnungsbegriff, keine Bereichsfrage) ist der schnellste, der sequentielle der langsamste, dazwischen liegt der direkte auf der VD.

Der zu suchende Wert wird durch das Feld, den Feldinhalt und den Vergleichsoperator bestimmt.

Vergleichsoperatoren können sein:

=	gleich
#	ungleich
<	kleiner
>	größer
:	kleiner oder gleich
*	größer oder gleich

Der Feldinhalt ist der Inhalt des Feldes in der Haupt- bzw. Verweisdatei. Ist ein Feld stichwortinvertiert, so wird der gesamte Feldinhalt durch die im A-Segment der Datenbeschreibung definierten Trenner in Stichworte aufgeteilt. Jede Zerlegung des Feldinhalts wird in der Verweisdatei eingetragen. Nur die Hauptdatei enthält den vollständigen Feldinhalt. Der Vorteil bei diesem Verfahren ist, daß beim Suchen nicht der gesamte Zeicheninhalt des Feldes eingegeben werden muß, um den Satz zu lokalisieren, sondern nur ein Stichwort (ein Bruchteil des vollständigen Feldinhalts).

Bei Stichwortinvertierung werden die als Trenner deklarierten Zeichen nicht invertiert. Soll nach ihnen gesucht werden, oder nach Zeichenfolgen (Strings), die Trennerzeichen enthalten, so muß der Suchweg vorher sequentiell gesetzt werden (vgl. Seite 125).

Ein Feldinhalt kann auch sedezimal angegeben werden:

Beispiel:

`SUCHE FELD=X'42150000'`	sucht nach der Gleitpunktzahl 21
`SUCHE FELD=X'123C4D'`	sucht nach einem Feldinhalt, der in Wirklichkeit aus 2 gepackten Zahlen besteht (könnte ein Compound Key sein).

Grundsätzlich, auch bei nicht invertierten Feldern, müssen Feldinhalte, die ein Komma enthalten, in Hochkommas eingeschlossen werden, da das Komma von CIS als Kommandotrennzeichen interpretiert wird.

Wird speziell nach Space gesucht, muß dieses ebenfalls in Hochkommas eingeschlossen werden.

Wird nach numerischen Feldinhalten gesucht, so wird der in der Suchfrage angegebene Wert dezimalpunktgerecht ausgerichtet, links und rechts überlaufende Stellen werden ohne Fehlermeldung abgeschnitten. D.h., handelt es sich z.B um ein einstellig, ganzzahlig definiertes Feld und es wird in der Suchfrage die Ziffernfolge 21.5 angegeben, so wird nach dem Feldinhalt 1 gesucht.

Werden in der Suchfrage keine Ergänzungen (Feldposition - FP, Abschnittsmultiplität - AM, oder Feldmultiplität -FM) angegeben, so gilt:

> Es werden alle Multiplitäten (Abschnitte und/oder Felder) durchsucht und geprüft, ob der tatsächliche Feldinhalt des in der Suchfrage angegebenen Feldes der Suchbedingung genügt.

Ist im A-Segment der Datenbeschreibung der Parameter SQLTAB=J angegeben, so wird bei der sequentiellen Suche die Zielpunktliste mit der Multiplität, in der der entsprechende Wert auftritt, ergänzt.

Eine derartige Zielpunktliste kann mit Folgefragen nicht weiter bearbeitet werden. Sie kann mehr Einträge enthalten, als Sätze in der Datei sind, da sie die Multiplitäten zählt (linearisiert).

Beispiel: 2 Sätze mit Wiederholfeldern.

KEY1	FELD1	. . .	1	7	12	4	1		
KEY2	FELD1	. . .	17	21	1	6	38	14	14

Wiederholfelder (W-FELD)

```
SUCHE W-FELD=1     AZI=3
Z KEY W-FELD       KEY1   1
                   KEY1   1
                   KEY2   1
```

Satz mit Wiederholabschnitt:

PERS	. . .	WARE	21652	240,00	HOSE	WARE	83519	89,00	HEMD

```
SU BEZEICHNUNG=HEMD          AZI=1
Z BEZEICHNUNG                HEMD
```

Eine Satzbeschreibung mit SQLTAB=J sollte sinnvollerweise nur e i n e Multiplität beschreiben, da alle nicht durch die Suchfrage eingeschränkten Multiplitäten wegen der Mehrfachspeicherung des Ordnungsbegriffes vervielfacht werden.

UND-/NICHT Folgefragen wirken einschränkend, d.h. die Anzahl der gefundenen Zielinformationen kann entweder kleiner oder gleich bleiben. Ist das im Suchargument angegebene Feld nicht in der Verweisdatei, so werden nur die gefundenen Zielinformationen durchsucht.

ODER-Folgefragen beziehen sich auf die gesamte Hauptdatei, d.h. die Anzahl der gefundenen Zielinformationen kann gleich oder größer werden.

Wenn mit Transaktionssicherung gearbeitet wird, kann jedes SUCHE-Kommando wahlweise mit Sperrfunktion aufgerufen werden. Wenn ein solches "Suchen mit Sperre" ausgeführt wird, werden die gefundenen Sätze gesperrt. Ein weiteres Sperrkommando ist weder notwendig noch sinnvoll. Die Sperre wird erst mit Transaktionsende wieder aufgehoben.

Ein "Suchen mit Sperre" entspricht einem gewöhnlichen Suchauftrag mit nachfolgendem SPERRE-Z Kommando. "Suchen mit Sperre" schließt die zeitliche Lücke zwischen den beiden sonst getrennten Kommandos und ist komfortabler.

Bei Initialfragen darf die Operationsergänzung S nur unmittelbar nach SUCHE bzw. bei Folgefragen nur beim ersten Verknüpfungsoperator der Suchfrage angegeben werden und gilt dann für den gesamten Einzelsuchauftrag (Initial- oder Folgefrage, einfach oder komplex).

2.25.1 Initialfrage

Einfache Initialfrage

Anwendungsart: aktiv/passiv

```
SUCHE{_/,S,}Feld{*/=/#/</>/:}Wert[,DB.xxxxxx]
```

SUCHE	Operation für Suchen.
S	Operationsergänzung für Suchen mit Sperre.
Feld	Name des zu suchenden Feldes.
Wert	Gesuchter Feldinhalt.
xxxxxx	Name der Satzbeschreibung

Beispiel:

```
*
SU,S,WOHNORT=MUENCHEN
IM00    ANZAHL DER ZIELINFORMATIONEN: 237
*
```

In einer Zielpunktliste (ZPL) werden die Schlüssel der gefundenen Sätze eingetragen und die Anzahl der Zielinformationen (AZI) als Suchergebnis ausgegeben.

Einfache Initialfrage mit Ergänzung

Anwendungsart: aktiv/passiv

```
SUCHE{_/,S,}Feld(Erg){*/=/#/</>/:}Wert[,DB.xxxxxx]
```

SUCHE	Operation für Suchen.
S	Operationsergänzung für Suchen mit Sperre.
Feld	Name des zu suchenden Feldes.
Ergänzung (vgl. Seite 18)	FP = Feldposition (nur bei Feldbedeutung T und A) AM = Abschnittsmultiplität FM = Feldmultiplität AZ = Abschnittszyklen FZ = Feldzyklen DB = Name der Satzbeschreibung: Nur beim Arbeiten im Fenster
Wert	Gesuchter Feldinhalt.
xxxxxx	Name der Satzbeschreibung.

Wortteilsuche

Mit Hilfe der Feldpositionsangabe ist es möglich, eine beliebige Zeichenfolge innerhalb eines beschriebenen Feldes zu suchen. Dabei spielt es keine Rolle, ob das Feld ein Ordnungsbegriff ist, ob es invertiert ist oder nicht.

Die Position (FP) innerhalb des durch die Feldbezeichnung bestimmten Feldes wird durch eine Zahl angegeben. Sie kann positiv oder negativ sein, d.h. der Suchvorgang wird von links oder von rechts beim betreffenden Feld begonnen.

FP=$ bedeutet, der angegebene Feldinhalt kann an beliebiger Stelle innerhalb des Feldes stehen.

Die Vergleichslänge wird implizit durch die Länge des in der Suchfrage angegebenen Feldinhaltes festgelegt.

Beispiele:

```
*
SU WORT(FP=$)=BAHN
...
*
```

Gefunden wird: BAHN
BAHNSCHRANKE
SCHNELLBAHN
TRAMBAHN
BAHNHOF
EISENBAHNFAHRT

```
*
SU WORT(FP=1)=BAHN
...
*
```

Gefunden wird: BAHN
BAHNSCHRANKE
BAHNHOF

```
*
SU WORT(FP=6)=BAHN
...
*
```

Gefunden wird: EISENBAHNFAHRT

```
*
SU WORT(FP=-1)=BAHN
...
*
```

Gefunden wird: BAHN
SCHNELLBAHN
TRAMBAHN

```
*
SU WORT(FP=-6)=BAHN
...
*
```

Gefunden wird: EISENBAHNFAHRT

Multiplitätensuche

Es ist möglich, sich bei der Suche auf eine bestimmte Multiplität zu beziehen.

Der Suchweg bei der Multiplitätensuche ist immer sequentiell.

Wird nur die Abschnittsmultiplität angegeben, kann jedes beliebige Feld innerhalb dieser Abschnittsmultiplität ein Treffer sein. Es werden also alle Felder dieses gewählten Abschnittes durchsucht.

Wird nur eine Feldmultiplität angegeben, werden alle Abschnittsmultiplitäten durchgesucht.

Bei Angabe von Abschnittszyklen werden, ausgehend von der angegebenen Abschnittsmultiplität (bei fehlendem AM-Parameter wird AM=1 angenommen), soviele Abschnitte durchsucht, wie der AZ-Parameter angibt.

Bei Angabe von Feldzyklen werden, ausgehend von der angegebenen Feldmultiplität (bei fehlendem FM-Parameter wird FM=1 angenommen), soviele Felder durchsucht, wie der FZ-Parameter angibt.

Beispiele:

Im Folgenden ist das Feld DIA ein Wiederholfeld in einem Wiederholabschnitt:

```
*
SU DIA(AM=3)=14712                  Im 3. Wiederholabschnitt werden alle Felder 'DIA'
*                                   geprüft.
*
SU DIA(FM=2)=14712                  Jedes 2. Wiederholfeld 'DIA' aus allen Wiederholab-
*                                   schnitten wird geprüft.
*
SU DIA(AM=2,FM=1)=14712             Das 1. Wiederholfeld im 2. Wiederholabschnitt wird
*                                   geprüft.
*
SU DIA=14712                        Alle Wiederholfelder 'DIA' in allen Wiederhol-
*                                   abschnitten werden geprüft.
*
SU DIA(AM=3,AZ=2)=14712             Im 3. und 4. Wiederholabschnitt werden alle Felder
*                                   'DIA' geprüft.
*
SU DIA(AM=3,AZ=2,FZ=4)=14712        Jedes 1. bis 4. Wiederholfeld 'DIA' im 3. und 4.
*                                   Wiederholabschnitt wird geprüft.
*
SU DIA(AZ=2,FZ=3)=14712             Das 1. bis 3. Wiederholfeld im 1. und 2. Wiederholab-
*                                   schnitt wird geprüft.
```

Wortteilsuche bei Multiplitäten

Bei Angabe einer Feldposition und einer Multiplität wird bei der angegebenen Multiplität eine Wortteilsuche durchgeführt.

Beispiel: Das Feld DIA ist ein Wiederholfeld in einem Wiederholabschnitt.

```
*
SU,S,DIA(AM=2,FM=1,FP=-1)=9
*
```

Das 1. Wiederholfeld DIA des 2. Wiederholabschnittes wird an der letzten Feldposition auf den Inhalt 9 geprüft. Die Sätze werden bis Transaktionsende gesperrt.

Verknüpfte Initialfrage

Anwendungsart: aktiv/passiv

```
SUCHE{_/,S,}Feld{*/=/#/</>/:}Wert,{UND_/ODER_/NICHT_}...
     [,DB.xxxxxx]
```

SUCHE	Operation für Suchen.
UND ODER NICHT	Verknüpfungsoperatoren.
S	Operationsergänzung für Suchen mit Sperre.
Feld	Name des zu suchenden Feldes.
Wert	Gesuchter Feldinhalt.
xxxxxx	Name der Satzbeschreibung.

Die verknüpfte Suchfrage kann beliebig komplex formuliert werden. Die Verknüpfungsoperatoren UND, ODER und NICHT bewirken keine (algebraische) Klammerung, sondern werden gleichrangig behandelt.

Beispiele:

```
*
SU,S,NAME=MUELLER,UND ORT=MUENCHEN
*
```

Alle MUELLER in MUENCHEN werden gesucht und gesperrt.

```
*
SU LAENDER-KEY=003,ODER LAENDER-KEY=013,NICHT KUNDE=NEU
*
```

Alle Stammkunden aus dem Land 003 oder 013 werden gesucht.

Verknüpfte Initialfrage mit Ergänzung

Anwendungsart: aktiv/passiv

```
SUCHE{_/,S,}Feld(Erg){*/=/#/</>/:}Wert,{UND_/ODER_/NICHT_}...
     [,DB.xxxxxx]
```

SUCHE	Operation für Suchen.
UND ODER NICHT	Verknüpfungsoperatoren.
S	Operationsergänzung für Suchen mit Sperre.
Feld	Name des zu suchenden Feldes.
Ergänzung (vgl. Seite 18)	FP = Feldposition: Nur mit Vergleichsoperator '=' und Feldbedeutung T oder A. AM = Abschnittsmultiplität FM = Feldmultiplität AZ = Abschnittszyklen FZ = Feldzyklen DB = Name der Satzbeschreibung: Nur beim Arbeiten im Fenster.
Wert	Gesuchter Feldinhalt.
xxxxxx	Name der Satzbeschreibung.

Die verknüpfte Suchfrage kann beliebig komplex formuliert werden. Die Verknüpfungsoperatoren UND, ODER und NICHT bewirken keine (algebraische) Klammerung, sondern werden gleichrangig behandelt.

Beispiele:

```
*
SU NAME(FP=$)=MAIER,UND OP=4711
*
SU NAME(FP=$)=MAIER,UND OP(AM=1)=4711
*
SU NAME=MAIER,UND OP(AM=1,FP=2)=4722
*
```

Bereichs-Initialfrage

Anwendungsart: aktiv/passiv

```
SUCHE{_/,S,}Feld{>/=/*}Wert1,BIS_Feld{</=/:}Wert2[,DB.xxxxxx]
```

SUCHE	Operation für Suchen.
S	Operationsergänzung für Suchen mit Sperre.
Feld	Name des zu suchenden Feldes.
Wert1	Gesuchter Feldinhalt (Anfangswert).
BIS	Verknüpfungsoperator.
Wert2	Gesuchter Feldinhalt (Endwert).
xxxxxx	Name der Satzbeschreibung.

Die im Kommando angegebenen Feldbezeichnungen müssen gleich sein. Der erste Feldinhalt (Wert1) muß "logisch kleiner" sein als der zweite Feldinhalt (Wert2).

Beispiele:

```
*
SU,S,NAME=HANS,BIS NAME=HERBERT
*
SU GEHALT>2500,BIS GEHALT<3000,UND GESCHLECHT=W
*
```

Anmerkung: Sämtliche bisher gezeigten Suchfragen lassen sich miteinander kombinieren. So kann auf eine Verknüpfung eine Bereichsfrage folgen oder umgekehrt.

Bereichs-Initialfrage mit Ergänzung

Anwendungsart: aktiv/passiv

```
SUCHE{_/,S,}Feld(Erg){>/=/*}Wert1,BIS_Feld(Erg){</=/:}Wert2
     [,DB.xxxxxx]
```

SUCHE	Operation für Suchen.
S	Operationsergänzung für Suchen mit Sperre.
Feld	Name des zu suchenden Feldes.
Ergänzung (vgl. Seite 18)	FP = Feldposition: Nur bei Feldbedeutung T oder A. AM = Abschnittsmultiplität FM = Feldmultiplität AZ = Abschnittszyklen FZ = Feldzyklen DB = Name der Satzbeschreibung: Nur beim Arbeiten im Fenster.
BIS	Verknüpfungsoperator
Wert1	Gesuchter Feldinhalt (Anfangswert).
Wert2	Gesuchter Feldinhalt (Endwert).
xxxxxx	Name der Satzbeschreibung.

Die im Kommando angegebenen Felder und Ergänzungen müssen gleich sein.

Bereichs-Initialfragen mit der Ergänzung FP sind nur zulässig, wenn das Feld nicht invertiert ist oder vorher SE,S (vgl. Seite 125) gegeben wurde.

Die Angabe FP=$ (beliebige Feldposition) ist nicht erlaubt.

Beispiele:

```
*
SU JAHR(FP=3)=73,BIS JAHR(FP=3)=75
```

Besser, da weniger fehlerträchtig, ist jedoch

```
SU JAHR(FP=3)=73,BIS JAHR=75
*
SU,S,DIA(AM=2,FP=1)=2,BIS DIA=4
*
SU,S,NAME=SCHMID,BIS NAME=SCHMITT,UND VORNAME=MARIA,NICHT ORT=BONN
```

Anmerkung: Sämtliche bisher gezeigten Suchfragen lassen sich miteinander kombinieren. So kann auf eine Verknüpfung eine Bereichsfrage folgen oder umgekehrt.

2.25.2 Folgefrage

Eine Folgefrage bezieht sich entweder auf eine vorangegangene Initialfrage oder auf eine vorangegangene Folgefrage. Die Anzahl der Folgefragen ist nicht begrenzt.

Einfache Folgefrage

Anwendungsart: aktiv/passiv

```
{UND/ODER/NICHT}{_/,S,}Feld{*/=/#/</>/:}Wert[,DB.xxxxxx]
```

UND ODER NICHT	Verknüpfungsoperatoren
S	Operationsergänzung für Suchen mit Sperre.
Feld	Name des zu suchenden Feldes.
Wert	Gesuchter Feldinhalt.
xxxxxx	Name der Satzbeschreibung.

Beispiel:

```
*
SU KLINIK=1
*
UND ARZT=KATZMEIER
*
NICHT,S,OPERATEUR=BECK
*
```

Einfache Folgefrage mit Ergänzung

Anwendungsart: aktiv/passiv

```
{UND/ODER/NICHT}{_/,S,}Feld(Erg){*/=/#/</>:}Wert[,DB.xxxxxx]
```

UND ODER NICHT	Verknüpfungsoperatoren
S	Operationsergänzung für Suchen mit Sperre.
Feld	Name des zu suchenden Feldes.
Ergänzung (vgl. Seite 18)	FP = Feldposition: Nur mit Vergleichsoperator '=' und Feldbedeutung T oder A. AM = Abschnittsmultiplität FM = Feldmultiplität AZ = Abschnittszyklen FZ = Feldzyklen DB = Name der Satzbeschreibung: Nur beim Arbeiten im Fenster.
Wert	Gesuchter Feldinhalt.
xxxxxx	Name der Satzbeschreibung.

Bei Angaben von Multiplitäten wird intern der Suchweg sequentiell gesetzt, auch wenn das Feld invertiert ist (vgl. Seite 125).

Die Wortteilsuche ist im allgemeinen, außer bei FP=1, zeitraubend, so daß, falls möglich, eine Vorauswahl getroffen werden sollte. Auf diese Weise wird ein sequentielles Durchsuchen der gesamten Daten vermieden.

Beispiele:

```
*
SU DIA=28700
IM00    ANZAHL ZIELINFORMATIONEN: 100
*
UND DIA(AM=2)=28700
IM01    ANZAHL ZIELINFORMATIONEN: 18
*
UND,S,DIA(AM=1,FP=$)=44
IM01    ANZAHL ZIELINFORMATIONEN: 3
*
```

Verknüpfte Folgefrage

Anwendungsart: aktiv/passiv

```
{UND/ODER/NICHT}{_/,S,}Feld{*/=/#/</>/:}Wert,
{UND_/ODER_/NICHT_}...[,DB.xxxxxx]
```

UND ODER NICHT	Verknüpfungsoperatoren
S	Operationsergänzung für Suchen mit Sperre.
Feld	Name des zu suchenden Feldes.
Wert	Gesuchter Feldinhalt.
xxxxxx	Name der Satzbeschreibung.

Die Verknüpfungen innerhalb einer Folgefrage werden in der angegebenen Reihenfolge abgearbeitet.

Beispiele:

```
*
NICHT ARZT=THUR,UND WOHNORT=MUENCHEN
IM01    ANZAHL ZIELINFORMATIONEN: 10
*
```

Das Ergebnis dieser verknüpften Folgefrage entspricht dem zweier Folgefragen.

```
*
NICHT ARZT=THUR
IM01    ANZAHL ZIELINFORMATIONEN: 837
*
UND WOHNORT=MUENCHEN
IM01    ANZAHL ZIELINFORMATIONEN: 10
*
```

Verknüpfte Folgefrage mit Ergänzung

Anwendungsart: aktiv/passiv

```
{UND/ODER/NICHT}{_/,S,}Feld(Erg){*/=/#/</>/:}Wert,
{UND_/ODER_/NICHT_}...[,DB.xxxxxx]
```

UND ODER NICHT	Verknüpfungsoperatoren
S	Operationsergänzung für Suchen mit Sperre.
Feld	Name des zu suchenden Feldes.
Ergänzung(vgl. Seite 18)	FP = Feldposition: Nur bei Feldbedeutung T oder A und Vergleichsoperator ' = '. AM = Abschnittsmultiplität FM = Feldmultiplität AZ = Abschnittszyklen FZ = Feldzyklen DB = Name der Satzbeschreibung: Nur beim Arbeiten im Fenster.
Wert	Gesuchter Feldinhalt.
xxxxxx	Name der Satzbeschreibung.

Beispiele:

```
*
UND DIAG(FP=1)=2,NICHT OP=12345
IM01    ANZAHL ZIELINFORMATIONEN: 105
*
NICHT,S,DIAG(FP=2,AM=3,FM=1)=3,UN ERG=1
IM01    ANZAHL ZIELINFORMATIONEN: 3
*
```

Bereichs-Folgefrage

Anwendungsart: aktiv/passiv

```
{UND/ODER/NICHT}{_/,S,}Feld{>/=/*}Wert1,BIS_Feld{</=/:}Wert2
                [,DB.xxxxxx]
```

UND ODER NICHT	Verknüpfungsoperatoren
S	Operationsergänzung für Suchen mit Sperre.
Feld	Name des zu suchenden Feldes.
Wert	Gesuchter Feldinhalt (Anfangswert).
BIS	Verknüpfungsoperator
Wert	Gesuchter Feldinhalt (Endwert).
xxxxxx	Name der Satzbeschreibung.

Die im Kommando angegebenen Feldbezeichnungen müssen gleich sein. Der erste Feldinhalt (Wert1) muß "logisch kleiner" sein als der zweite Feldinhalt (Wert2).

Beispiel:

```
*
ODER GEHALT=9000,BIS GEHALT=15000
*
```

Bereichs-Folgefrage mit Ergänzung

Anwendungsart: aktiv/passiv

```
{UND/ODER/NICHT}{_/,S,}Feld(Erg){>/=/*}Wert1,
                BIS_Feld(Erg){</=/:}Wert2[,DB.xxxxxx]
```

UND ODER NICHT	Verknüpfungsoperatoren
S	Operationsergänzung für Suchen mit Sperre.
Feld	Name des zu suchenden Feldes.
Ergänzung (vgl. Seite 18)	FP = Feldposition: Nur bei Feldbedeutung T oder A und Vergleichsoperator ' = '. AM = Abschnittsmultiplität FM = Feldmultiplität AZ = Abschnittszyklen FZ = Feldzyklen DB = Name der Satzbeschreibung: Nur beim Arbeiten im Fenster.
Wert1	Gesuchter Feldinhalt (Anfangswert).
BIS	Verknüpfungsoperator
Wert2	Gesuchter Feldinhalt (Endwert).
xxxxxx	Name der Satzbeschreibung.

Die im Kommando angegebenen Felder müssen gleich sein. Der erste Feldinhalt (Wert1) muß "logisch kleiner" sein als der zweite Feldinhalt (Wert2). Bereichsfolgefragen mit der Ergänzung FP sind nur sequentiell zulässig, d.h. wenn das Feld nicht invertiert ist oder vorher SE,S (vgl. Seite 125) gegeben wurde.

Die Angabe FP=$ (beliebige Feldpositionen) ist bei der Bereichsfrage nicht erlaubt.

Beispiele:

```
*
UND BETRIEBSNR(FP=1)=A,BIS BETRIEBSNR(FP=1)=B
*
NICHT DIA(FM=2)=45010,BIS DIA(FM=2)=55010
*
```

2.25.3 Abschnittsbezogene Suchfrage

Abschnittsbezogenes Suchen bedeutet, daß nur dann ein Satz in die Zielpunkliste aufgenommen wird, wenn alle Suchargumente in derselben Abschnittsmultiplität auftreten und die Suchbedingung(en) erfüllt ist (sind).

Die Abschnittsmultiplität selbst ist nicht anzugeben. Die Abschnittsmultiplität wird solange erhöht, bis die Bedingungen erfüllt sind, oder über alle Multiplitäten geprüft wurde.

Beim abschnittsbezogen Suchen wird stets sequentiell in der HD gesucht, auch wenn das angegebene Feld invertiert ist.

Die bisher erläuterten Suchfragen (Initial-, Folge- und Bereichsfragen)

```
SUCHE ...          SUCHE,S,...
UND ...            UND,S,...
ODER ...           ODER,S,...
NICHT ...          NICHT,S,...
```

werden durch die Operationsergänzung A auf das abschnittsbezogene Suchen gesetzt:

```
SUCHE,A,...        SUCHE,AS,...
UND,A,...          UND,AS,...
ODER,A,...         ODER,AS,...
NICHT,A,...        NICHT,AS,...
```

Regeln für den Aufbau einer abschnittsbezogenen Suchfrage:

- Nur Felder aus Wiederholabschnitten.
- Alle Felder aus der gleichen Abschnittsart.
- Mindestens 2 Verknüpfungen.
- Die Operationsergänzung A darf nur beim ersten Verknüpfungsoperator der Suchfrage angegeben sein.
- Kein ODER-Verknüpfungsoperator innerhalb der Suchfrage.

Auf eine abschnittsbezogene Suchfrage kann eine nichtabschnittsbezogene Suchfrage folgen und umgekehrt.

Beispiele abschnittsbezogener Suchfragen:

Alle Felder in den abschnittsbezogenen Suchfragen seien Felder aus einem Wiederholabschnitt.

Beispiel 1:

```
*
SU,A,DIAGNOSE=200,UND DAUER=10
IM01    ANZAHL ZIELINFORMATIONEN: 50
```

Es werden alle Sätze lokalisiert, bei denen innerhalb des gleichen Abschnittes beide Bedingungen erfüllt sind.

Beispiel 2:

```
*
SU,AS,DIAGNOSE=200,UND DAUER=10,UND ORT=BERLIN
IM01    ANZAHL ZIELINFORMATIONEN: 20
```

Es werden alle Sätze lokalisiert und gesperrt, bei denen alle 3 Bedingungen innerhalb derselben Abschnittsmultiplität erfüllt sind.

Wird das Kommando getrennt eingegeben,

```
SU,A,DIAGNOSE=200,UND DAUER=10
IM01    ANZAHL ZIELINFORMATIONEN: 50
UND,S,ORT=BERLIN
IM01    ANZAHL ZIELINFORMATIONEN: 40
```

so wird das sogenannte abschnittsbezogene Suchen nur für den 1. Teil des Kommandos ausgeführt. Eine abschnittsbezogene Verknüpfung zwischen DIAGNOSE, DAUER und ORT findet bei der UND-Folgefrage nicht statt.

In diesem Fall ist es sinnvoll, die Sperre nur bei der Folgefrage einzugeben. Würde auch oder nur bei der Initialfrage gesperrt, wären 50 Sätze gesperrt.

Beispiel 3:

```
*
SU ARZT=ROSENFELD
IM01    ANZAHL ZIELINFORMATIONEN: 50
UND,A,DIAG(FP=1)=2,NICHT OP=12345
IM01    ANZAHL ZIELINFORMATIONEN: 18
```

Erst die UND-Folgefrage wird abschnittsbezogen ausgeführt.

Beispiel 4:

```
*
SU,A,NAME=HANS,BIS NAME=HERBERT
```

Diese Suchfrage ist nicht erlaubt, da keine weitere UND-/NICHT-Verknüpfung angegeben wurde. Korrekt muß die Frage beispielsweise lauten:

```
SU,A,NAME=HANS,BIS NAME=HERBERT,UND ORT(FP=$)=MUENCHEN
IM01    ANZAHL ZIELINFORMATIONEN: 20
```

Beispiel 5:

```
*
SU ORT=MUENCHEN,OD ORT=KOELN,U,A,NAME=MEIER,N VORNAME=HANS
```

Falsche Fragestellung, die Operationsergänzung A darf nur beim 1. Operator der Suchfrage angegeben werden. Soll nur der NAME und VORNAME abschnittsbezogen gesucht werden, müssen 2 Fragen gestellt werden:

```
SU ORT=MUENCHEN,OD ORT=KOELN
IM01    ANZAHL ZIELINFORMATIONEN: 20
U,A,NAME=MEIER,N VORNAME=HANS
IM01    ANZAHL ZIELINFORMATIONEN: 10
```

Beispiel 6:

```
*
SU,A,ORT=BERLIN,ODER ORT=KOELN,UND NAME=MEIER
```

Diese Frage ist nicht erlaubt, da ODER innerhalb einer abschnittsbezogenen Suchfrage ein erneutes sequentielles Suchen in der gesamten HD bedeuten würde.

Sollen trotzdem alle MEIER in BERLIN oder KOELN gefunden werden, so müssen zwei Fragen gestellt werden:

```
SU,A,ORT=BERLIN,UND NAME=MEIER
IM01    ANZAHL ZIELINFORMATIONEN: 20
ODER,A,ORT=KOELN,UND NAME=MEIER
IM01    ANZAHL ZIELINFORMATIONEN: 40
```

2.26 SYSTEM-Kommando

Anwendungsart: aktiv

```
SYSTEM[_/Kommando][,DB.xxxxxx]
```

SYSTEM	Operation für Systemmodus.
/Kommando	BS2000-Kommando, das ausgeführt werden soll. Das Kommando kann maximal 200 Zeichen lang sein.
xxxxxx	Es muß keine Satzbeschreibung angegeben werden. Wird DB.xxxxxx trotzdem angegeben, so wirkt sich das erst auf das nächste Kommando aus.

Alle BS2000-Kommandos sind erlaubt außer:

- den Kommandos, die nicht per CMD-Makro aufgerufen werden können
- CALL-PROCEDURE, DO, START-PROGRAM, HELP-MSG-INFORMATION, LOAD-PROGRAM, LOGOFF

Wird nur SYSTEM eingegeben, so geht CIS im Teilnehmerbetrieb auf einen BREAKPOINT, im Teilhaberbetrieb wird SYSTEM ohne BS2000-Kommando zurückgewiesen.

2.27 TEXT-Kommando

Anwendungsart: aktiv/passiv

```
TEXT[_Text][,DB.xxxxxx]
```

TEXT	Operation für Textkommentar.
Text	Kommentar: Auch führende Spaces werden zum Kommentar gerechnet (Einrücken).
xxxxxx	Eine Satzbeschreibung muß nicht angegeben werden. Wird DB.xxxxxx trotzdem angegeben, so wirkt sich das erst auf das nächste Kommando aus.

Das TEXT-Kommando veranlaßt einen Listenausdruck auf einer oder mehreren separaten Seiten. Es wird dann angewendet, wenn ein Listenausdruck, erstellt mit einem DRUCKE-Kommando, ausführlicher kommentiert werden soll.

Bei der erzeugten Listenausgabe enthält die erste Seite den Kommentar und die folgenden Seiten die eigentliche Druckausgabe.

Der Kommentar wird mit dem Kommando TEXT eingeleitet. Es können beliebig viele Zeilen mit maximal 132 Zeichen eingegeben werden. Den Abschluß bildet das ENDE-Kommando (vgl. Seite 71).

2.28 TRANSAKTION-Kommando

Dieses Kommando muß beim Einsatz von CIS mit Datensicherung (Before-Image-Sicherung und Transaktionsverarbeitung) angewendet werden.

Die Transaktion ist die Menge der CIS-Kommandos, die für den Anwender eine logische Einheit darstellen.

Das TRANSAKTION-Kommando definiert den Transaktionsanfang, das normale Transaktionsende und im Fehlerfall die Wiederherstellung der Datenbank(en) in den Zustand vor dem Zeitpunkt des Transaktionsanfangs.

Innerhalb einer Transaktion können nur gesperrte Sätze geändert werden (vgl. Seite 141).

Eine Transaktion kann

- durch das CIS-Kommando TRANSAKTION-R und
- durch abnormales Beenden von CIS abgebrochen werden.

2.28.1 TRANSAKTION-A (Anfang)

Anwendungsart: aktiv/passiv

```
TRANSAKTION,A[,DB.xxxxxx]
```

TRANSAKTION	Operation für Transaktion.
A	Operationsergänzung für Anfang.
xxxxxx	Eine Satzbeschreibung muß nicht angegeben werden. Wird DB.xxxxxx trotzdem angegeben, so wirkt sich das erst auf das nächste Kommando aus.

Mit diesem Kommando wird der Anfang der Transaktion definiert.

2.28.2 TRANSAKTION-AL (Anfang lesen)

Anwendungsart: aktiv/passiv

```
TRANSAKTION,AL[,DB.xxxxxx]
```

TRANSAKTION	Operation für Transaktion.
A	Operationsergänzung für Anfang lesen.
xxxxxx	Eine Satzbeschreibung muß nicht angegeben werden. Wird DB.xxxxxx trotzdem angegeben, so wirkt sich das erst auf das nächste Kommando aus.

Mit diesem Kommando wird der Anfang einer Lese-Transaktion definiert.

In dieser Transaktion kann gesperrt werden, es können jedoch keine Änderungen durchgeführt werden. Wird dies trotzdem versucht, so wird die Fehlermeldung UP37 ausgegeben.

2.28.3 TRANSAKTION-AR (Neu aufsetzen)

Anwendungsart: aktiv/passiv

```
TRANSAKTION,AR[,DB.xxxxxx]
```

TRANSAKTION	Operation für Transaktion.
AR	Operationsergänzung für neu aufsetzen.
xxxxxx	Eine Satzbeschreibung muß nicht angegeben werden. Wird DB.xxxxxx trotzdem angegeben, so wirkt sich das erst auf das nächste Kommando aus.

Mit diesem Kommando wird die Transaktion auf ihren Anfangspunkt gesetzt:

- Alle Änderungen werden rückgängig gemacht.
- Alle Sperren werden aufgehoben.

Die Transaktion wird nicht beendet. Sie kann normal weitergeführt werden, z.B. mit Sperren und Änderungen. Sie muß mit TR,E beendet oder mit TR,R zurückgesetzt werden. TR,R setzt dann natürlich nur Änderungen ab dem Zeitpunkt, zu dem TR,AR gegeben wurde, zurück, da TR,AR selbst alle vorangegangenen Änderungen zurückgesetzt hat.

Da die Transaktion nicht beendet wird, kann dieses Kommando auch im synchronisierten UTM-Betrieb benützt werden.

2.28.4 TRANSAKTION-E (Ende)

Anwendungsart: aktiv/passiv

```
TRANSAKTION,E[,DB.xxxxxx]
```

TRANSAKTION	Operation für Transaktion.
E	Operationsergänzung für Ende.
xxxxxx	Eine Satzbeschreibung muß nicht angegeben werden. Wird DB.xxxxxx trotzdem angegeben, so wirkt sich das erst auf das nächste Kommando aus.

Mit diesem Kommando wird das normale Ende der Transaktion definiert.

Alle in der Datenbank bzw. den Datenbanken seit Transaktionsanfang durchgeführten Änderungen werden permanent, sie werden nicht mehr zurückgesetzt.

Alle seit Transaktionsanfang gesperrten Sätze, Zielpunktlisten oder Dateien werden freigegeben.

2.28.5 TRANSAKTION-EB (Ende bedingt)

Anwendungsart: aktiv/passiv

```
TRANSAKTION,EB[,DB.xxxxxx]
```

TRANSAKTION	Operation für Transaktion.
EB	Operationsergänzung für Ende bedingt.
xxxxxx	Eine Satzbeschreibung muß nicht angegeben werden. Wird DB.xxxxxx trotzdem angegeben, so wirkt sich das erst auf das nächste Kommando aus.

Mit diesem Kommando wird das normale Ende der Transaktion definiert, sofern eine Transaktion offen ist. (vgl. TR,E auf Seite 174). Ist keine Transaktion offen wird IM00 gemeldet.

Wird UTM-inlinked oder DBH-Betrieb gefahren, so werden alle transaktionsspezifischen Bereiche freigegeben. Diese Funktion sollte z.B. im UTM-Betrieb benützt werden, wenn der Anwender an einem bestimmten Terminal die Zusammenarbeit mit CIS beenden will. Der Platz in der Transaktionstabelle wird für ein anderes Terminal freigegeben.

2.28.6 TRANSAKTION-R (Rücksetzen)

Anwendungsart: aktiv/passiv

```
TRANSAKTION,R[,DB.xxxxxx]
```

TRANSAKTION	Operation für Transaktion.
R	Operationsergänzung für Rücksetzen.
xxxxxx	Eine Satzbeschreibung muß nicht angegeben werden. Wird DB.xxxxxx trotzdem angegeben, so wirkt sich das erst auf das nächste Kommando aus.

Alle in der Datenbank bzw. den Datenbanken seit Transaktionsanfang durchgeführten Änderungen werden zurückgesetzt.

Alle seit Transaktionsanfang gesperrten Sätze, Zielpunktlisten oder Dateien werden freigegeben.

2.28.7 TRANSAKTION-RB (Rücksetzen)

Anwendungsart: aktiv/passiv

```
TRANSAKTION,RB[,DB.xxxxxx]
```

TRANSAKTION	Operation für Transaktion.
RB	Operationsergänzung für Rücksetzen bedingt.
xxxxxx	Eine Satzbeschreibung muß nicht angegeben werden. Wird DB.xxxxxx trotzdem angegeben, so wirkt sich das erst auf das nächste Kommando aus.

Dieses Kommando entspricht dem TR,R. Es setzt eventuell offene Transaktionen zurück. Sollten keine Transaktionen offen sein, so reagiert es im Gegensatz zum TR,R nicht mit einer Fehlermeldung.

2.29 Verteilte Transaktionsverarbeitung

Soll ein Client Verbindung zu mehr als einem CIS-Server aufnehmen und in einer Transaktion, die zu mehr als einem Server Verbindung hat, Daten ändern, so spricht man von einer verteilten Transaktionsverarbeitung.

Das Transaktionsende muß nach dem 2-Phasen-Commit-Protokoll realisiert werden. Die Logik hierzu muß im Client liegen. Die notwendigen Befehle stellt CIS zur Verfügung.

Die Befehle sind für die Betriebsarten TIAM und UTM-unsynchronisiert realisiert. Für die Betriebsart UTM-synchronisiert ist die gesamte Logik und deren Zusammenarbeit mit UTM in CISCON enthalten.

Befehlsübersicht:	TR,AS	Anfang einer Transaktion.
	TR,PE	Provisorisches Ende.
	TR,SA	Statusabfrage
	TR,SL	Löschen des Statuseintrags.
	TR WE	Beenden der Transaktion.
	TR,WR	Rücksetzen der Transaktion.

2.29.1 TRANSAKTION-AS (Anfang einer Transaktion/Statusdatei)

Anwendungsart: aktiv/passiv

```
TRANSAKTION,AS,TID=xxxxxxxx
```

TRANSAKTION	Operation für Transaktion.
AS	Operationsergänzung für Anfang einer Transaktion mit Protokollierung in die Statusdatei.
TID	Transaktionskennung
Betriebsart:	TIAM UTM-unsynchronisiert

Anfang einer Transaktion wie bei TR,A. Es wird jedoch vermerkt, daß bei einem Warmstart von CISKOOR diese Transaktion in die Statusdatei geschrieben werden soll. Nach einem Warmstart kann dann der Zustand dieser Transaktion abgefragt werden. Es darf nicht vergessen werden, den Statuseintrag zu löschen. Dies geschieht nicht automatisch.

2.29.2 TRANSAKTION-PE (Provisorisches Ende)

Anwendungsart: aktiv/passiv

```
TRANSAKTION,PE,TID=xxxxxxxx
```

TRANSAKTION	Operation für Transaktion.
PE	Operationsergänzung für provisorisches Ende einer Transaktion.
TID	Transaktionskennung
Betriebsart:	TIAM UTM-unsynchronisiert

Die Transaktion geht in den Zustand "PETA" oder "PTC = Prepare to Commit".

2.29.3 TRANSAKTION-SA (Statusabfrage)

Anwendungsart: aktiv/passiv

```
TRANSAKTION,SA,TID=xxxxxxxx
```

TRANSAKTION	Operation für Transaktion.
SA	Operationsergänzung für Statusabfrage einer Transaktion.
TID	Transaktionskennung
Betriebsart:	TIAM UTM-unsynchronisiert

Statusabfrage für die Transaktion mit der TID xxxxxxxx. Das Kommando muß für eine TIAM-Transaktion von einem TIAM-Programm ausgehen. Für eine UTM-unsynchronisierte Transaktion muß das Kommando von der Anwendung (gleicher Name) gegeben werden, von der das Transaktions-Anfangskommando stammte.

Die Rückmeldung über CM bedeutet: IM01/IM11 : TA unbekannt
IM02/IM12 : TA rückgesetzt
IM03/IM13 : TA noch offen
IM04/IM14 : TA in PETA (PTC)

2.29.4 TRANSAKTION-SL (Löschen des Statuseintrags)

Anwendungsart: aktiv/passiv

```
TRANSAKTION,SL,TID=xxxxxxxx
```

TRANSAKTION	Operation für Transaktion.
SL	Operationsergänzung für Löschen des Statuseintrags.
TID	Transaktionskennung
Betriebsart:	TIAM UTM-unsynchronisiert

Löschen des Statuseintrags der Transaktion mit der TID "xxxxxxxx".

Betriebsart UTM-unsynchronisiert:	Ist die TID nicht angegeben, wird der Eintrag für die aktuelle Transaktion gelöscht. Das Kommando muß von der Anwendung (gleicher Name) gegeben werden, von der das Transaktions-Anfangskommando stammte.
Betriebsart TIAM:	Es muß die TID angegeben werden.

2.29.5 TRANSAKTION-WE (Beenden Transaktion nach Warmstart)

Anwendungsart: aktiv/passiv

```
TRANSAKTION,WE,TID=xxxxxxxx
```

TRANSAKTION	Operation für Transaktion.
WE	Operationsergänzung für Beenden der Transaktion nach Warmstart.
TID	Transaktionskennung
Betriebsart:	TIAM UTM-unsynchronisiert

Beenden der Transaktion mit der TID "xxxxxxxx", die nach dem Warmstart im Zustand PETA (PTC) war.

2.29.6 TRANSAKTION-WR (Rücksetzen nach Warmstart)

Anwendungsart: aktiv/passiv

```
TRANSAKTION,WR,TID=xxxxxxxx
```

TRANSAKTION	Operation für Transaktion.
WR	Operationsergänzung für Rücksetzen der Transaktion.
TID	Transaktionskennung
Betriebsart:	TIAM UTM-unsynchronisiert

Rücksetzen der Transaktion mit der TID "xxxxxxxx", die nach dem Warmstart im Zustand PETA (PTC) war.

2.30 VERBINDE-Kommando

Der Anwender steht häufig vor der Situation, daß Daten, die er gemeinsam bearbeiten will, in mehreren Dateien (Datenbanken) aufgeteilt sind. Über gemeinsame Feldinhalte können zusammengehörige Daten verknüpft und aus Sicht des Anwenders wie Gesamtsätze (Relationszeilen) behandelt werden. Innerhalb der Gesamtmenge dieser Relationszeilen (Relation, Fenster) kann sich der Anwender so bewegen als ob er eine CIS-Datenbank mit der entsprechenden Zahl Sätze bearbeiten würde.

Die Relation wird mit dem VERBINDE-Kommando erstellt. Dabei ist es unerheblich, ob die beteiligten Dateien von gleichem oder unterschiedlichem Typ sind (ISAM, SAM, UPAM) und ob die Sekundärindizes (Verweise) gleichen oder unterschiedlichen Typs sind (CIS-Verweisdatei, LEASY-SI-Datei).

Die Reihenfolge der Relationszeilen innerhalb der Relation ist aus Anwendersicht willkürlich und hängt nicht nur von der Formulierung des Kommandos ab.

Jede einzelne Relationszeile entspricht einem Satz in einer Datenbank und enthält alle Felder aller zusammengehörigen "echten" Sätze. Der interne Aufbau dieser Relationszeilen (virtuellen Sätze) ist aus Anwendersicht willkürlich und hängt nicht allein von der Formulierung des Kommandos ab. Soweit mit CIS Felder gezielt angesprochen werden oder der Satzaufbau ohne Bedeutung ist, ist das für den Anwender unerheblich. Das trifft auf alle zulässigen Aktiv-Anwendungen und auf viele Passiv-Kommandos zu. Bei allen Formen der Satzübergabe (betrifft nur Programme) bereitet der Anwender die virtuellen Sätze mit CISCOBMV auf (vgl. Manual-4: CISCOBMV).

Beispiel:

In drei Datenbanken (Dateien) mit den dazugehörenden Datenbeschreibungen (DB.KUNDEN, DB.AUFTRA und DB.ARTIKL) seien folgende Sätze enthalten:

Kundensätze
DB.KUNDEN

```
KNR NAME
 17 MAIER&CO
 23 KLEIN
 28 FEST
 38 BRAUN AG
 43 AMANN
```

Auftragssätze
DB.AUFTRA

```
AUFNR KNR ARTNR MG
   5    28  2345 15
   7    17  2502  3
   9    38  5625 12
   3    17  1244  3
```

Artikelsätze
DB.ARTIKL

```
   ANR BEZEICHNG
1000 TISCH
1111 HOCKER
1244 SESSEL
2500 BETT
2502 SCHRANK
5625 REGAL
6333 TRUHE
7999 LAMPE
```

Mit dem Kommando

```
VE KNR(DB=KUNDEN)=KNR(DB=AUFTRA) ARTNR=ANR(DB=ARTIKL),DB.JOIN01
```

entstehen folgende Relationszeilen:

```
 17 MAIER&CO     7    17  2502  3    2502 SCHRANK
 17 MAIER&CO     3    17  1244  3    1244 SESSEL
 38 BRAUNAG      9    38  5625 12    5625 REGAL
```

- Es entsteht keine Relationszeile für Kunden-Nr. 28 und Auftrag über Artikel-Nr. 2345, da kein Artikelsatz mit Artikel-Nr. 2345 vorhanden ist.

- Jede einzelne Zeile enthält alle Felder aller 3 zugehörigen "echten" Sätze. Der Inhalt des Kundensatzes 17 ist demzufolge in der 1. und in der 2. Relationszeile zweimal vorhanden.

2.30.1 VERBINDE-Kommando (Allgemein)

Anwendungsart: aktiv/passiv

```
VERBINDE{_/,{A/F/S/AF/AS/FS/AFS},}Feld1(DB=aaaaaa)
        =Feld2(DB=bbbbbb)[_Feldn...]...[,DB.vvvvvv]
```

VERBINDE	Operation für Verbinden.
A	Operationsergänzung für auftragsbezogenes Verbinden.
F	Operationsergänzung für forciertes Verbinden.
S	Operationsergänzung für singuläres Verbinden.
AF AS FS AFS	Operationsergänzungen kombiniert aus auftrags-bezogenem, forciertem und singulärem Verbinden.
Feldn	Felder, mit denen die Verbindung aufgebaut werden soll.
aaaaaa bbbbbb	Datenbankpaßworte, die jeweils eine Haupt- und Verweisdatei beschreiben.
DB.vvvvvv	Verbundpaßwort: Es wird vom Anwender frei vergeben und beschreibt die Relation (Fenster). Dieses Paßwort darf in der aktuellen Daten-beschreibungsdatei nicht existieren.

A Auftragsbezogenes Verbinden:

Wird die Operationsergänzung A im VERBINDE-Kommando angegeben, so werden nur Relationszeilen gebildet, die Sätze einer vorherigen Zielpunktliste enthalten. Die ZPL wird nicht verändert. Damit läßt sich die Anzahl der Relationszeilen auf bestimmte Sätze einschränken.

Beispiel:

```
SU KDNR=4711,DB.KUNDEN
*
VE,A,KDNR(DB=KUNDEN)=AKDNR(DB=AUFTRA),DB.JOJOJO
```

Es werden nur Relationszeilen gebildet, die die KDNR=4711 enthalten.

Die ZPL der Suchfrage SU KDNR=4711 bleibt erhalten und könnte z.B. per Paßwortwechsel (DB.KUNDEN) wieder verwendet werden:

```
VE,A,KDNR(DB=KUNDEN)=LKDNR(DB=LIEFER),DB.JOIN01
```

Eine Suchfrage innerhalb des Fensters überschreibt diese ZPL.

F Forciertes Verbinden:

Werte, die keine Entsprechung in der anderen Spalte haben, werden in den Verbund übernommen.

Beispiel:

KDR	ADR	ergibt 8 Relationszeilen KDR	ADR
1	1	1	1
7	1	1	1
12	1	1	1
	2	Dummy	2
	4	Dummy	4
	7	7	7
	10	Dummy	10
	12	12	12

Das forcierte Verbinden kann z.B. dazu verwendet werden, um Aufträge ohne Kunden zu ermitteln:

```
VE,F,KDR(DB=KUNDEN)=ADR(DB=AUFTRG),DB.xxxxxx
SU ADR#0,NICHT KDR#0
IG,F,DB.AUFTRG
```

AZI=3 Die drei Aufträge mit nicht (mehr) vorhandener Kundennummer sind gefunden.

S Singuläres Verbinden:

Bei m zu n Beziehungen wird nicht karthesisch multipliziert, sondern es entstehen n Realtionszeilen für m < n oder m Relationszeilen für n < m.

Regeln für die Verwendung der Datenbankpaßworte:

1. Jedes der Paßworte aaaaaa - nnnnnn muß eine gültige Satzbeschreibung aus der aktuellen Datenbeschreibungsdatei sein.

2. In jeder Gleichsetzung müssen auf beiden Seiten, d.h. links und rechts des Gleichheitszeichens, unterschiedliche Satzbeschreibungen angegeben werden. Demzufolge darf aaaaaa nicht gleich bbbbbb sein.

   Beispiel:

   ```
   VE KDNR(DB=KUNDEN)=AKDNR(DB=AUFNEU)=AKDNR(DB=AUFALT),DB.VERBXY
   ```

3. Datenbankpaßworte (DB=...) gelten solange bis ein anderes Paßwort angegeben wird. Links vom Gleichheitszeichen können also Paßworte u.U. weggelassen werden.

   Beispiel:

   ```
   VE KDNR(DB=KUNDEN)=AKDNR(DB=AUFTRA)
   POSITION=POSITION(DB=POSI01),DB.VERB01
   ```

   Das Kommando könnte auch so formuliert werden:

   ```
   VE KDNR(DB=KUNDEN)=AKDNR(DB=AUFTRA)
   POSITION(DB=AUFTRA)=POSITION(DB=POSI01),DB.VERB01
   ```

4. Werden im VERBINDE-Kommando zwei Zuordnungen angegeben, so muß mindestens ein Feld der zweiten Zuordnung aus derselben Satzbeschreibung stammen, wie eines der beiden Felder der ersten Zuordnung.

   Beispiele:

   ```
   VE KDNR(DB=KUNDEN)=AKDNR(DB=AUFTRA)
   POSITION(DB=AUFTRA)=POSITION(DB=POSI01),DB.VER999

   VE KDNR(DB=KUNDEN)=AKDNR(DB=AUFTRA)
   POSITION=POSITION(DB=POSI01),DB.VER999

   VE KDNR(DB=KUNDEN)=AKDNR(DB=AUFTRA)
   ORT(DB=KUNDEN)=ORT(DB=AUFTRA),DB.VERVER

   VE KDNR(DB=KUNDEN)=AKDNR(DB=AUFTRA)
   ORT=ORT(DB=KUNDEN),DB.VERVER
   ```

2.30.2 Voraussetzungen

1. Die Haupt- und Verweisdateien müssen in den A-Segmenten der beteiligten Datenbeschreibungen angegeben sein. Es sind alle Dateitypen, auch gemischt zulässig, die CIS bearbeiten kann (ISAM, SAM, UPAM). TYP=L (LEASY-SI-Datei) ist für die Verweisdatei zulässig.

2. Die Verbundfelder, d.h. die Felder, mit denen die Verbindung aufgebaut werden soll, müssen invertiert sein. Die Felder dürfen nicht stichwortinvertiert sein. Die Ergänzungen FP, AM, AZ, FM, FZ sind nicht zulässig. Die FL-Angabe ist nur bei Textfeldern erlaubt.

3. Das vom Anwender frei vergebbare Verbund-Paßwort (DB.vvvvvv) darf nicht in der aktuellen Datenbeschreibungsdatei existieren.

   Aufgrund des VERBINDE-Kommandos wird intern eine Verbunddatenbeschreibung gebildet, die alle Felder der beteiligten Datenbeschreibungen umfaßt und die Datenschutzparameter DSS und DSA enthält. Dabei werden die Parameter DSA und DSS nur dann für die Verbindung auf J gesetzt, wenn sie in allen beteiligten Satzbeschreibungen auch mit J angegeben sind.

4. Es dürfen maximal 11 Dateien miteinander verknüpft werden, also maximal 10 Gleichsetzungen.

2.30.3 Anwendung der Dateiverknüpfung

Zulässige CIS-Operationen bei gültigem Verbund-Paßwort

Ist bei einer zulässigen Operation die Operationsergänzung nicht angegeben, so sind alle Operationsergänzungen verwendbar.

Operation	zulässig	Bemerkung
AENDERN	N	
AKTIVIEREN	J	nicht Operationsergänzung I
BLAETTERN	J	
CLOSE	N	
DRUCKE	J	nicht Operationsergänzung B
EINGEBEN	N	
ENDE	J	
EXIT	J	nicht Operationsergänzung A
FREIGEBEN	J	nicht Operationsergänzung I
GET	J	nicht Operationsergänzungen A, B, P
HALT	J	
HILF	J	
IGNORIEREN	J	
KORRIGIEREN	N	
LOESCHEN	N	
OPEN	J	Dateinamen müssen im A-Segment stehen.
PUT	N	
RECHNE	J	Ergebnis nur in Rechenvariable
SETZEN	J	
SICHERN	J	nicht Operationsergänzungen I, V
SORTIERE	J	
SPERRE	N	
STATUS	J	
SUCHE	J	
SYSTEM	J	
TEXT	J	
TRANSAKTION	J	
VERBINDE	J	
WARUM	J	
WIEDERHOLEN	J	
WRITE	J	keine Transformation
ZEIGEN	J	nicht Operationsergänzung B
$D-Kommandos	J	
Kurzkommandos	J	

Sortierung

Nach Ablauf des VERBINDE-Kommandos ist aus Anwendersicht die Sortierung der Relationszeilen innerhalb der Relation undefiniert. Wird also eine bestimmte Sortierung, z.B. für eine nachfolgende Ausgabe gewünscht, so muß vom Anwender ein SORTIERE-Kommando angewendet werden.

Paßwortwechsel

Solange das Verbund-Paßwort gültig ist, bewegt sich der Anwender wie auf einer Datenbank. Auch diese virtuelle Datenbank hat einen Zeiger, der auf die jeweiligen Relationszeilen zeigt. Mit einem Paßwortwechsel kann jederzeit auf andere, echte Datenbanken gewechselt werden. Dabei werden die Relation und der Zeiger innerhalb der Relation nicht zerstört. Durch Angabe des Verbund-Paßwortes wird wieder in die Relation zurückgekehrt. Die Relation mit entsprechendem Zeigerstand steht wieder zur Verfügung.

Beispiel ohne Transaktionen:

```
*
VE KDNR(DB=KUNDEN)=AKDNR(DB=AUFTRA),DB.JOIN01
*
Z,1,T,EDV NAME, DB.JOIN01        Verbund-Paßwort
EDV  NAME
 13  MEIER
*
AE,1,F,NAME=MAIER,DB.KUNDEN      Wechsel in echte Datenbank + Update (FCB-Zeiger).
*
Z,1,T,EDV NAME, DB.JOIN01        Rückkehr ins Fenster.
EDV  NAME
 13  MAIER                       Zeiger steht auf derselben Relation.
```

Beispiel mit Transaktionen:

```
*
TR,A
*
VE KDNR(DB=KUNDEN)=AKDNR(DB=AUFTRA),DB.JOIN01
*
Z,1,T,EDV NAME,DB.JOIN01
EDV  NAME
 13  MEIER
*
SU EDV=13,DB.KUNDEN
*
SP,Z
*
AE,1,F,NAME=MAIER
*
Z,1,T,EDV NAME,DB.JOIN01
EDV  NAME
 13  MAIER
*
TR,E
```

Mehrere Relationen

Es ist möglich, Relationen temporär zu sichern. Dadurch können während eines CIS-Laufs mehrere Relationen aufgebaut und wechselweise bearbeitet werden.

- Eine Relation wird implizit mitgesichert, wenn eine Zielplunktliste zu dieser Relation gesichert wird.

- Die Relation wird implizit wieder aktiviert, wenn eine vorher gesicherte Zielpunktliste zu dieser Relation wieder aktiviert wird.

- Paßwortwechsel zwischen Verbunddatenbeschreibungen sind nur mit dem AKTIVIERE-Kommando oder dem VERBINDE-Kommando möglich.

Beispiel:

```
*
VE KDNR(DB=KUNDEN)=AKDNR(DB=AUFTRA),DB.KUNAUF                    1)
*
SU KEDV#0                                                        2)
*
SI,E,KEDV=1                                                      3)
*
VE POSITION(DB=AUFTRA)=POSITION(DB=POSIO1),DB.AUFPOS             4)
...                                                              5)
*
AK,E,KEDV=1,DB.KUNAUF                                            6)
```

1) Für alle Kunden mit zugehörigen Aufträgen wird eine Relation erstellt.

2) Erstellen einer Zielpunktliste innerhalb der Relation, in diesem Fall über alle Relationszeilen.

3) Sichern der Zielpunktliste. Dadurch wird die Relation implizit mitgesichert.

4) Erstellen einer neuen Verbindung.

5) Arbeiten in der neuen Relation.

6) Aktivieren der vorhin gesicherten Zielpunktliste. Dadurch wird implizit die dieser Zielpunktliste zugrundeliegende Relation zum aktuellen Fenster.

Wird für ein VERBINDE-Kommando ein Verbundpaßwort vergeben, mit dem bereits eine Zielpunktliste temporär gesichert ist, so erfolgt eine Fehlermeldung. Sämtliche gesicherten Zielpunktlisten zu einer Verbunddatenbeschreibung müssen erst per FREIGEBE-Kommando freigegeben werden, bevor dasselbe Paßwort für eine neue Verbindung vergeben werden kann.

Beispiel:

```
*
VE KNDR(DB=KUNDEN)=AKNR(DB=AUFTRA),DB.AAAAAA
IM10    ANZAHL RELATIONSZEILEN: 1020
*
SU KNDR=4711
IM10    ANZAHL RELATIONSZEILEN: 140
*
SI,E,AKDNR=1
*
VE POSI(DB=AUFTRA)=POSI(DB=POSI00),DB.AAAAAA
JO32    PW EXISTIERT ZU GESICHERTEM FENSTER
*
FR,E,AKDNR=1,DB.AAAAAA
*
VE POSI(DB=AUFTRA)=POSI(DB=POSI00),DB.AAAAAA
IM10    ANZAHL RELATIONSZEILEN: 3768
```

Gruppenbildung bei tabellarischer Ausgabe

Beim tabellarischen DRUCKEN/ZEIGEN werden gleiche Feldinhalte je nach Sortierung der sich wiederholenden Teilsätze in der Ausgabe unterdrückt.

Beispiel:

```
*
Z,$,T,KEDV NAME KDNR,DB.KUNDEN
KEDV    NAME KDNR
   1   HUBER    1
   2   MEIER 4711
   3 MUELLER 9999
*
Z,$,T AEDV ARTIKEL AKDNR,DB.AUFT00
AEDV ARTIKEL AKDNR
   1    HOSE     1
   2    HEMD  4711
   3 STRUMPF     1
   4   JACKE  9999
   5    HOSE  4711
   6 STIEFEL     1
VE KDNR(DB=KUNDEN)=AKDNR(DB=AUFT00),DB.KAJOIN
IM10    ANZAHL RELATIONSZEILEN: 6
*
Z,$,T,KEDV NAME KDNR AKDNR ARTIKEL AEDV
KEDV    NAME KDNR AKDNR ARTIKEL AEDV
   1   HUBER    1     1    HOSE    1
                      1 STRUMPF    3           1)
                      1 STIEFEL    6           1)
   2   MEIER 4711  4711    HEMD    2
                   4711    HOSE    5           1)
   3 MUELLER 9999  9999   JACKE    4
```

1) Gleiche Feldinhalte werden bei der Ausgabe unterdrückt. Im obigen Beispiel wird also KEDV=1, NAME=HUBER und KDNR=1 nur in der ersten Zeile ausgegeben.

Satz-Verarbeitung

Eine mit dem VERBINDE-Kommando erstellte Relation kann auch passiv bearbeitet werden. U.a. können die einzelnen Relationszeilen mit GET-K bzw. GET-KN ins Programm eingelesen werden (vgl. Manual-4: CISCOBMV).

2.31 WARUM-Kommando

Anwendungsart: aktiv

```
WARUM_cccc[,DB.xxxxxx]
```

WARUM	Operation für Warum.
cccc	Codierte Meldung.
xxxxxx	Eine Satzbeschreibung muß nicht angegeben werden: Wird DB.xxxxxx trotzdem angegeben, so wirkt sich das erst auf das nächste Kommando aus.

Dieses Kommando gibt zur codierten und verbalen Meldung zusätzliche Erläuterungen aus.

Beispiel:

```
AF03  FELDFEHLER
*
WA AF03
AF03   FELDFEHLER
       -ANGEGEBENE FELDBEZEICHNUNG NICHT IN DER
        VERWENDETEN DATENBESCHREIBUNG
       -ANGEGEBENE FELDBEZEICHNUNG MEHRDEUTIG
        (VORSICHT BEI GEBRAUCH VON ABKUERZUNGEN)
       -DATENBESCHREIBUNG OHNE FELDBEZEICHNUNGEN IM E-SEGMENT
        (MASKE BZW. TRANSFORMATIONSBESCHREIBUNG)
        ZUSATZ-INFO: FEHLERHAFTE FELDBEZEICHNUNG
*
```

2.32 WIEDERHOLEN-Kommando

Anwendungsart: aktiv

```
WIEDERHOLEN[_n][,DB.xxxxxx]
```

WIEDERHOLEN	Operation für Kommandowiederholung.
n	Auf Kommandos bezogene Mengenangabe. Maximal 2 gültige Ziffern. $ = alle Kommandos. Wird n nicht angegeben wird n = 1 angenommen, d.h. nur das letzte Kommando wird wiederholt.
xxxxxx	Eine Satzbeschreibung muß nicht angegeben werden. Wird DB.xxxxxx trotzdem angegeben, so wirkt sich das erst auf das nächste Kommando aus.

Das WIEDERHOLEN-Kommando gibt früher eingegebene Kommandos wieder auf dem Bildschirm aus. Soll z.B. ein Kommando mehrmals benützt werden und es wurde von einer Bildschirmausgabe überschrieben, so ist es möglich, dieses Kommando mit dem WIEDERHOLEN-Kommando wieder auf dem Bildschirm anzuzeigen. Auch eventuell fehlerhafte Kommandos können auf diese Weise nachkontrolliert werden. Die Anzahl der wiederholbaren Kommandos ist abhängig von der Länge der vorher eingegebenen Kommandos, da für die temporäre Speicherung (bis Programmende) ein Speicherbereich von ca. 250 Bytes zur Verfügung steht.

Beispiel:

```
*
SU ALTER=50,BIS ALTER=65
IM00    ANZAHL ZIELINFORMATIONEN: 7
*
UND ABTEILUNG=BUCHHALTUNG
IM00    ANZAHL ZIELINFORMATIONEN: 2
*
Z,2,T,ALTER NAME VORNAME
ALTER  NAME    VORNAME
   51  MAIER   ELISABETH
   53 MUELLER  HANS
*
*
WI 3
Z,2,T,ALTER NAME VORNAME
UND ABTEILUNG=BUCHHALTUNG
SU ALTER=50,BIS ALTER=65
*
```

Es sollen alle Personen im Alter zwischen 50 und 65 Jahren gezeigt werden. Um aber das SUCHE-Kommando, das durch die Ausgabe überschrieben wurde, nicht nochmals eingeben zu müssen, werden die 3 letzten Kommandos mit dem WIEDERHOLEN-Kommando auf den Bildschirm geholt. Das SUCHE-Kommando kann jetzt in der angezeigten Form übernommen und abgeschickt werden.

Nach dem SUCHE-Kommando kann auch das ZEIGE-Kommando leicht modifiziert werden (z.B. Anzahl ändern, um alle zu zeigen).

2.33 WRITE-Kommando

Mit dem Kommando WRITE können Daten in eine Datei ausgegeben werden. Die Anzahl der auszugebenden Sätze ist über die Mengenangabe bestimmbar. Der Datei muß mit dem SET-FILE-LINK Kommando der LINK-NAME 'ISISOUT' zugewiesen werden. Eine nicht vorhandene Datei wird mit SUPPORT=PUBLIC-DISK automatisch eingerichtet. Soll die Datei mit anderen bzw. zusätzlichen Operanden eingerichtet werden, so muß vor dem SET-FILE-LINK Kommando das CREATE-FILE Kommando gegeben werden.

```
/SET-FILE-LINK FILE-NAME=Dateiname,LINK-NAME=ISISOUT
[,ACCESS-METHOD={SAM/ISAM}],...
```

Mit dem Operanden ACCESS-METHOD wird die Zugriffsmethode (SAM oder ISAM) bestimmt. Wird der Operand nicht angegeben, so ist die Zugriffsmethode ISAM.

Während eines CIS-Laufs können maximal fünf Dateien beschrieben werden.

Beim ersten Ansprechen einer bereits vorhandenen Datei mit einem WRITE-Kommando, wird der Dateiinhalt gelöscht. Eine bestehende Datei kann also nur innerhalb des gleichen CIS-Laufs fortgeschrieben werden. Für ein und dieselbe Datei darf bei mehrmaligem Ansprechen während eines CIS-Laufs der Kommandotyp im WRITE-Kommando nicht gewechselt werden.

Werden mehrere ISISOUT-Dateien nacheinander beschrieben, muß das Kommando

```
CLOSE ISISOUT=Dateiname
```

angewendet werden, um beim Ausgabedateiwechsel die Fehlermeldung

```
WR19 SET-FILE-LINK KOMMANDO MIT LINK-NAME=ISISOUT FEHLT
```

zu unterdrücken (für KUKO's).

Ablaufbeispiel:

```
SU ...
SYS /SET-FILE-LINK FILE-NAME=name1,LINK-NAME=ISISOUT
WR...DATEI=name1
CL ISISOUT=name1
SYS /SET-FILE-LINK FILE-NAME=name2,LINK-NAME=ISISOUT
WR...
```

2.33.1 WRITE-K (Satz)

Anwendungsart: aktiv/passiv

```
WRITE[,n],K,DATEI=Name[,DB.xxxxxx]
```

WRITE	Operation für Schreiben in Datei.
n	Auf Sätze bezogene Mengenangabe. $ = alle Sätze. Bei fehlender Mengenangabe wird 1 angenommen.
K	Operationsergänzung für Satz.
Name	Name der Ausgabedatei (entsprechend SET-FILE-LINK Kommando).
xxxxxx	Satz- oder Transformationsbeschreibung.

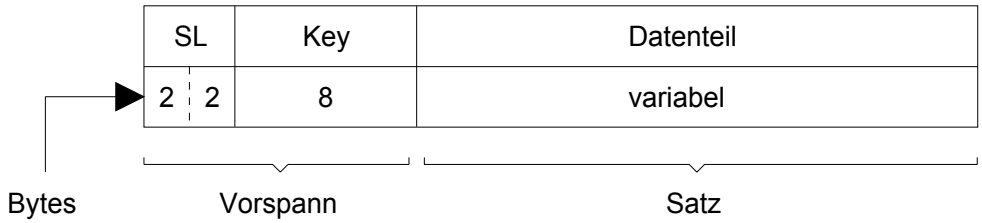
Die Sätze werden in die im Dateinamen benannte Datei geschrieben. Dabei gibt es zwei Möglichkeiten:

1. ISAM-Ausgabedatei:

   Die Mengenangabe bestimmt, wieviele Sätze in die Datei geschrieben werden.

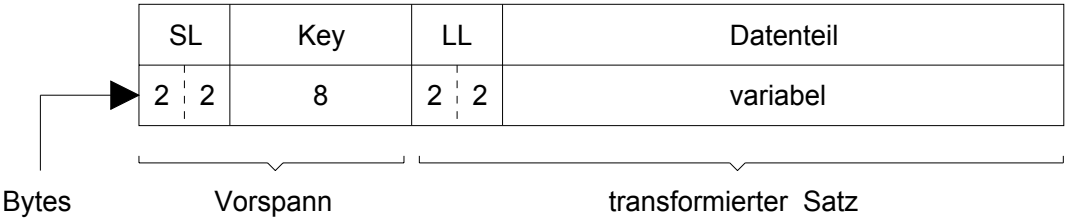
   Die Datei wird sequentiell, d.h. mit dem PUT-Makro geschrieben.

   Bei Verwendung einer Satzbeschreibung bleibt der Satzaufbau der Sätze gleich. Dabei wird jedem Satz ein 4-Byte langes Satzlängenfeld und ein 8-Byte langer Schlüssel, also insgesamt 12 Bytes, vorangestellt. Der Schlüssel wird numerisch entpackt mit der Schrittweite 1 aufsteigend vergeben:



   SL = Satzlänge

Bei Verwendung einer Transformationsbeschreibung kann der Satzaufbau, der sich an die 12 vorangestellten Bytes anschließt, verändert werden. Dem Datenteil wird ein 4 Byte langes Längenfeld vorangestellt:



SL = Satzlänge
LL = Länge des transformierten Satzes

Im Regelfall wird mit KEYLEN=8 und KEYPOS=5 gearbeitet. Eine so erzeugte Datei ist problemlos mit EDT, EDOR oder FORTRAN-Programmen zu verarbeiten.

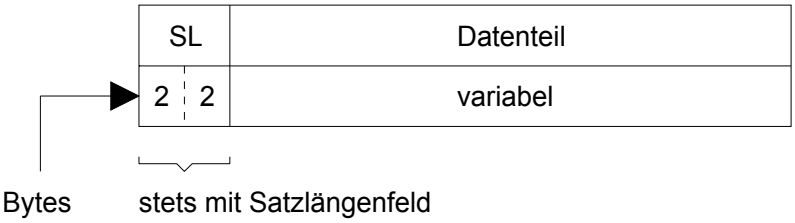
Weicht man von diesem Regelfall ab, so sind KEYPOS und KEYLEN so zu wählen, daß der ISAM-Schlüssel aufsteigend ist. Falls hierbei gleiche Schlüssel auftreten können, muß im SET-FILE-LINK Kommando DUPLICATE-KEY=YES angegeben sein.

2. SAM-Ausgabedatei:

Die Datei muß explizit mit SET-FILE-LINK . . . ACCESS-METHOD=SAM zugewiesen werden.

Wird mit RECFORM=F gearbeitet, so ist bei der RECSIZE-Angabe zu beachten, daß dem Satz noch 4 Bytes als Satzlängenfeld vorangestellt werden (RECSIZE=Daten + 4 Bytes).

Die Mengenangabe bestimmt, wieviele Sätze in die Datei geschrieben werden.



SL = Satzlänge

Bei Verwendung einer Satzbeschreibung bleibt der Satzaufbau gleich.

Bei Verwendung einer Transformationsbeschreibung kann der Satzaufbau verändert werden.

2.33.2 WRITE-KO (Originalsatz)

Anwendungsart: aktiv/passiv

```
WRITE[,n],KO,DATEI=Name[,DB.xxxxxx]
```

WRITE	Operation für Schreiben in Datei.
n	Auf Sätze bezogene Mengenangabe. $ = alle Sätze. Bei fehlender Mengenangabe wird 1 angenommen.
KO	Operationsergänzung für Originalsatz.
Name	Name der Ausgabedatei (entsprechend SET-FILE-LINK Kommando).
xxxxxx	Satz- oder Transformationsbeschreibung.

Es kann direkt in eine neue Datei geschrieben werden, die im Aufbau der aktuellen Hauptdatei völlig gleich ist.

Die Datei muß eine ISAM-Datei sein.

Die Datei wird wahlfrei, d.h. mit dem STORE-Makro geschrieben.

Bei Verwendung einer Satzbeschreibung bleibt der Satzaufbau der Sätze gleich. Im Gegensatz zu WRITE-K oder WRITE-S werden keine Daten vorangestellt.

Bei Verwendung einer Transformationsbeschreibung kann der Satzaufbau verändert werden.

Im Regelfall wird mit der gleichen KEYPOS- und KEYLEN-Angabe wie bei der aktuellen Hauptdatei gearbeitet. Eine so erzeugte Ausgabedatei kann problemlos als neue Hauptdatei verwendet werden. (Achtung: evtl. zugehörige Verweisdatei aufbauen).

Wird von diesem Regelfall abgewichen, so werden die ISAM-Schlüssel entsprechend der Abweichung aufgebaut.

2.33.3 WRITE-S (Selektiv)

Anwendungsart: aktiv/passiv

```
WRITE[,n],S,DATEI=Name,Feld[(Erg)][=Wert,][_Feld...]...
     [,DB.xxxxxx]
```

WRITE	Operation für Schreiben in Datei.
n	Auf Sätze bezogene Mengenangabe. $ = alle Sätze. Bei fehlender Mengenangabe wird 1 genommen.
S	Operationsergänzung für selektiv.
Name	Name der Ausgabedatei (entsprechend SET-FILE-LINK Kommando).
Feld	Name des zu schreibenden Feldes.
Ergänzung (vgl. Seite 18)	AM = Abschnittsmultiplität. Bei fehlender AM-Angabe wird AM=1 angenommen. AZ = Abschnittszyklen. $ = alle Abschnittszyklen. Bei fehlender AZ-Angabe wird AZ=1 angenommen. FM = Feldmultiplität. Bei fehlender FM-Angabe wird FM=1 angenommen. FZ = Feldzyklen. $ = alle Feldzyklen. Bei fehlender FZ-Angabe wird FZ=1 angenommen. FL = Feldlänge. Bei fehlender FL-Angabe wird das Feld mit der definierten Länge geschrieben. FP = Feldposition DB = Name der Satzbeschreibung. Nur beim Arbeiten im Fenster.
Wert	Es können Konstanten in den Ausgabesatz eingefügt werden bzw. bei fehlenden Multiplitäten ein festes (für CIS beschreibbares) Ausgabeformat erzeugt werden.
xxxxxx	Satzbeschreibung

Entsprechend der Reihenfolge der angegebenen Felder wird ein Ausgabesatz aufgebaut und in die im Kommando mit "Name" benannte Datei geschrieben. Die Felder werden nicht verändert. Es kann in eine ISAM- oder SAM-Datei geschrieben werden.

1. ISAM-Ausgabedatei:

   Die Mengenangabe bestimmt, aus wievielen Sätzen in die Datei geschrieben wird.

   Die Datei wird sequentiell, d.h. mit dem PUT-Makro geschrieben.

   Entsprechend der Reihenfolge der angegebenen Felder werden die Ausgabesätze aufgebaut. Jedem Satz wird ein 4-Byte langes Satzlängenfeld und ein 8 Byte langer Schlüssel, also insgesamt 12 Bytes, vorangestellt. Der Schlüssel wird numerisch entpackt mit der Schrittweite 1 aufsteigend vergeben.

   Im Regelfall wird mit KEYLEN=8 und KEYPOS=5 gearbeitet. Eine so erzeugte Datei ist problemlos mit EDT, EDOR oder FORTRAN-Programmen zu verarbeiten.

   Wird von diesem Regelfall abgewichen, so sind KEYPOS (größer / gleich 13, da stets 12 Bytes den eigentlichen Satzdaten vorangestellt werden) und KEYLEN so zu wählen, daß der ISAM-Schlüssel aufsteigend ist. Falls hierbei gleiche Schlüssel auftreten können, muß im SET-FILE-LINK Kommando DUPLICATE-KEY=YES angegeben sein.

2. SAM-Ausgabedatei:

   Die Datei muß explizit mit SET-FILE-LINK . . . ACCESS-METHOD=SAM zugewiesen werden.

   Wird mit RECFORM=F gearbeitet, so ist bei der RECORD-SIZE Angabe zu beachten, daß dem Satz noch ein 4 Byte langes Satzlängenfeld vorangestellt wird (RECSIZE=Daten + 4 Bytes).

   Die Mengenangabe bestimmt, wieviele Sätze in die Datei geschrieben werden.

   Entsprechend der Reihenfolge der angegebenen Felder werden die Ausgabesätze aufgebaut.

Beispiele:

```
SYS/SET-FILE-LINK FILE-NAME=AUSDAT1,
LINK-NAME=ISISOUT
*
SU FIRMA:10,DB.FIRMDB
IM00    ANZAHL ZIELINFORMATIONEN: 3
*
WR,$,S,DATEI=AUSDAT1,FIRMA NAME
*
```

Eine ISAM-Datei wird eingerichtet bzw. zugewiesen.

War die Datei schon vorhanden, so wird ihr Inhalt vor dem Beschreiben gelöscht. Ein im SET-FILE-LINK Kommando angegebenes OPEN-MODE=EXTEND ist unwirksam.

```
BL,$,R
WR,$,S,DATEI=AUSDAT1,FIRMA BESTELLUNG(AM=-2,AZ=2,FL=4)
*
```

Die Datei AUSDAT1 wird um 3 Sätze fortgeschrieben, wobei vom Wiederholfeld BESTELLUNG die beiden letzten Abschnitte in der Länge 4 übernommen werden.

```
SU FIRMA=12
IM00    ANZAHL ZIELINFORMATIONEN: 1
*
WR,S,DATEI=AUSDAT1,FIRMA NAME
*
SU NAME=HUBER
IM00    ANZAHL ZIELINFORMATIONEN: 5
*
```

Die Datei AUSDAT1 wird fortgeschrieben. Sie wird um einen Satz erweitert.

```
WR,$,K,DATEI=AUSDAT2
WR19 SET-FILE-LINK KOMMANDO FUER
     LINK-NAME=ISISOUT FEHLT
*
```

Das SET-FILE-LINK Kommando für die Datei AUSDAT2 wird angefordert.

```
SYS/SET-FILE-LINK FILE-NAME=AUSDAT2,
LINK-NAME=ISISOUT,ACCESS-METHOD=SAM
*
WR,$,K,DATEI=AUSDAT2
*
```

Die SAM-Datei AUSDAT2 wird eingerichtet bzw. zugewiesen. Die 5 Sätze werden in AUSDAT2 geschrieben, ein vorhandener Dateiinhalt wird vorher gelöscht.

```
CL ISISOUT=AUSDAT2
*
SU UMSATZKLASSE=1
IM00    ANZAHL ZIELINFORMATIONEN: 9
*
```

Die Datei AUSDAT2 wird geschlossen.

```
SYS/SET-FILE-LINK FILE-NAME=AUSDAT3,
LINK-NAME=ISISOUT
*
```

Das SET-FILE-LINK Kommando für die Datei AUSDAT3 wird eingegeben.

```
WR,$,KO,DATEI=AUSDAT3
*
```

Die 9 Sätze werden in die ISAM-Datei AUSDAT3 geschrieben. Ein vorhandener Dateiinhalt wird vorher gelöscht.

2.33.4 WRITE-STT (Selektiv mit Trennern)

Anwendungsart: aktiv/passiv

```
WRITE[,n],STT,DATEI=Name,Feld[(Erg)][_Feld...]...[,DB.xxxxxx]
```

WRITE	Operation für Schreiben in Datei.
n	Auf Sätze bezogene Mengenangabe. $ = alle Sätze. Bei fehlender Mengenangabe wird 1 genommen.
STT	Operationsergänzung für selektiv Transferdatei mit Trennern.
Name	Name der Ausgabedatei (entsprechend SET-FILE-LINK Kommando).
Feld	Name des zu schreibenden Feldes.
Ergänzung (vgl. Seite 18)	AM = Abschnittsmultiplität. Bei fehlender AM-Angabe wird AM=1 angenommen. FM = Feldmultiplität. Bei fehlender FM-Angabe wird FM=1 angenommen. FZ = Feldzyklen. Bei fehlender FZ-Angabe wird FZ=1 angenommen. DB = Name der Satzbeschreibung. Nur beim Arbeiten im Fenster.
xxxxxx	Satzbeschreibung

Entsprechend der Reihenfolge der angegebenen Felder wird ein Ausgabesatz aufgebaut und in die im Kommando unter Name benannte SAM-Datei geschrieben.

Fehlende Felder (z.B. nicht vorhandene Multiplitäten) fehlen im Ausgabesatz.

Die Felder werden pro Satz mit Zwischentrennern aneinander gereiht. Führende und anhängende Spaces verschwinden.

Enthält ein Feld nur Spaces, so wird nur der Trenner ausgegeben.

2.33.5 WRITE-TS (Transferdatei)

Ein wichtiger Aspekt für die Zusammenarbeit einzelner Standard-Software-Produkte (beispielsweise INFPLAN, ADILOS, BUGRAF) ist die Transferdatei. Sie trägt wesentlich zur horizontalen Integration der IDV-Bausteine im BS2000 bei.

Über diese Datei kann der Anwender Daten zwischen den Produkten, aber auch zwischen den Produkten und speziellen Anwenderprogrammen austauschen.

Die Transferdatei ermöglicht auch den Datentransfer vom BS2000 zu einem anderen System, um die Daten dort anderen Zielanwendungen zur Verfügung zu stellen (beispielsweise SIPLAN, MULTIPLAN, INFORMIX in SINIX). Mit dieser Methode arbeitet z. B. das Produkt MMC (Micro-Mainframe-Connection).

Anwendungsart: aktiv/passiv

```
WRITE[,n],TS,DATEI=Name,Feld[(Erg)][_Feld...]...[,DB.xxxxxx]
```

WRITE	Operation für Schreiben in Datei.
n	Auf Sätze bezogene Mengenangabe. $ = alle Sätze. Bei fehlender Mengenangabe wird 1 genommen.
TS	Operationsergänzung für Transferdatei.
Name	Name der Ausgabedatei (entsprechend SET-FILE-LINK Kommando).
Feld	Name des zu schreibenden Feldes.
Ergänzung (vgl. Seite 18)	AM = Abschnittsmultiplität. Bei fehlender AM-Angabe wird AM=1 angenommen. AZ = Abschnittszyklen. $ = alle Abschnittszyklen. Bei fehlender AZ-Angabe wird AZ=1 angenommen. FM = Feldmultiplität. Bei fehlender FM-Angabe wird FM=1 angenommen. FZ = Feldzyklen. Bei fehlender FZ-Angabe wird FZ=1 angenommen. DB = Name der Satzbeschreibung. Nur beim Arbeiten im Fenster.
xxxxxx	Satzbeschreibung

Das Kommando arbeitet analog dem WRITE-S (selektiv) Kommando. Entsprechend der Reihenfolge der angegebenen Felder wird ein Ausgabesatz aufgebaut und in die im Kommando unter Name benannte SAM-Datei weggeschrieben.

Die Felder werden nicht verändert. Fehlende Felder (z.B. nicht vorhandene Multiplitäten) fehlen im Ausgabesatz.

2.34 ZEIGE-Kommando

Das Kommando ZEIGE ermöglicht es, Daten auf dem Terminal auszugeben, wobei die Anzahl der auszugebenden Einheiten über die Mengenangabe bestimmbar ist.

Es können Daten der Hauptdatei, der Verweisdatei und der Datenbeschreibung ausgegeben werden.

Numerische Felder werden beim Zeigen wie folgt aufbereitet:

- Entpacken bei Feldbedeutung P (gepacktes Feld).
- Einfügen des Dezimalpunkts, wenn Dezimalstellen definiert sind.
- Unterdrückung führender Nullen.
- Einfügen des evtl. negativen Vorzeichens vor die höchstwertige gültige Ziffer.
- Rechtsbündige Ausrichtung der Darstellung.
- Einfügen der Zeichenfolge '0.(00...)', wenn die Dezimalstellen größer oder gleich der maximalen Ziffernzahl definiert wurden.
- Ist die FL-Angabe (Feldlänge) zu klein, d.h. es kann nicht die ganze Zahl aufbereitet werden, so wird der Überlauf in der höchstwertigen Stelle durch einen * gekennzeichnet.

Nach jeder Ausgabe wird der RDATA-Stern (*) für eine neue Eingabe angeboten.

Bei ZEIGE-Kommandos (ausgenommen ZEIGE-M) werden implizite Blätter-Funktionen angeboten:

Eingabe:	Blättern:
+DUE / DUE	Um einen Bildschirm vorwärts.
+n	Um n Sätze vorwärts.
-n	Um n Sätze rückwärt.s
++	Um alle Sätze vorwärts. Der letzte Satz wird gezeigt
--	Auf den Anfang. Der erste Bildschirm wird gezeigt.
-	Um einen Satz rückwärts.

Die Blätter-Funktionen beeinflußt das ursprünglich gegebene ZEIGE-Kommando nicht.

Für jedes ZEIGE-Kommando (außer E, M,...) gibt es auch ein entsprechendes DRUCKE-Kommando, das die Daten auf Drucker ausgibt (vgl. Seite 49).

2.34.1 ZEIGE-A (Feldauskunft)

Anwendungsart: aktiv

```
ZEIGE[,n],A[,DB.xxxxxx]
```

ZEIGE	Operation für Ausgabe auf Terminal.
n	Auf Felder bezogene Mengenangabe. $ = alle Felder. Bei fehlender Mengenangabe wird '$' angenommen.
A	Operationsergänzung für Feldauskunft.
xxxxxx	Satzbeschreibung

Aus der angegebenen Satzbeschreibung wird folgende Information zur Verfügung gestellt:

- Feldbezeichnung
- Abschnittsart (AART)
- Wiederholabschnitt (W)
- Relative Feldadresse (ADR), bei V-Format bezogen auf Satzanfang, bei MV-Format bezogen auf den Abschnittsanfang.
- Verweisdatei - Segmentname (SEGM).
- Länge in Bytes (LNG), ggf. inkl. Dezimalstellen.
- Feldbedeutung (T)
- Art der Darstellung
- Anzahl der Dezimalstellen (DZ)
- Wenn ein Verbundpaßwort gültig ist, das Dateipaßwort, in dem das Feld definiert ist.

Beispiel:

```
*
D,$,A,DB.FLDAUS
*

FELDBEZEICHNUNG  AART W  ADR SEGM LNG T DARSTELLUNG DZ PASSWORT

EDV-NR           ABCD      9         3 O BINAER           FLDAUS
PSEUDO           ABCD     12 PSE     1 P ZEICHEN LB       FLDAUS
NAME             ABCD     13 NAM0   25   ZEICHEN LB
FIRMA            ABCD     38 FIRS   70 U ZEICHEN LB
FIRM1            ABCD    108        35 U ZEICHEN LB
FIRM2            ABCD    143        35 U ZEICHEN LB
GKL              BCDE W    9 GKL     1   ENTPACKT RB
UMSATZ           BCDE W   10         6   GEPACKT     2
PROD             CDEF      9 PRDS   15 W ZEICHEN LB
```

Die Reihenfolge der Feldbeschreibungen ist wie in der Datenbeschreibung definiert. Im Verbund sind die Felder aufsteigend nach Feldbezeichnung sortiert.

2.34.2 ZEIGE-B (Bildausgabe)

Anwendungsart: aktiv

```
ZEIGE[,n],B[,DB.xxxxxx]
```

ZEIGE	Operation für Ausgabe auf Terminal.
n	Auf Sätze bezogene Mengenangabe. $ = alle Sätze. Bei fehlender Mengenangabe wird '$' angenommen.
B	Operationsergänzung für Bild.
xxxxxx	Bilddatenbeschreibung

Die Daten eines Satzes werden in Bilder, die mit einer Bildbeschreibung definiert worden sind, eingetragen und am Terminal ausgegeben.

2.34.3 ZEIGE-E (gesicherte Zielpunktlisten und VD-Zeiger)

Anwendungsart: aktiv

```
ZEIGE[,n],E[,DB.xxxxxx]
```

ZEIGE	Operation für Ausgabe auf Terminal.
n	Auf Ausgabezeilen bezogene Mengenangabe. $ = alle Zeilen. Bei fehlender Mengenangabe wird '$' angenommen.
E	Operationsergänzung für temporär gesicherte Zielpunktlisten und VD-Zeiger.
xxxxxx	Satzbeschreibung

Mit diesem Kommando können alle bisher temporär gesicherten Zielpunktlisten angezeigt werden, die mit dem SICHERN-E/S Kommando (vgl. Seite 129 und 133) gesichert wurden. Ein angegebenes Paßwort wird erst beim folgenden CIS-Kommando wirksam.

Folgende Daten werden angezeigt:
- Feldbezeichnung
- Feldinhalt
- Paßwort
- Anzahl der Zielinformationen (AZI)
- File-Nummer (EAM)
- Anzahl der belegten EAM-Blöcke
- ZPL im Fenster: VE = J
- VD-Zeiger: VE = I

Beispiel:

```
*
Z,$,E
FELDBEZ.    FELDINHALT          PASSWD    AZI    EAM    BLK    VE

FIRMA       A                   PERSON    27     29      1
NAME        01                  JOIN01    5      29      1      J
FIRMA       B                   PERSON    156    30      1
*
FREIGEBEN NAME=01,DB.JOIN01
*
Z,$,E
FELDBEZ.    FELDINHALT          PASSWD    AZI    EAM    BLK    VE

FIRMA       A                   PERSON    27     29      1
FIRMA       B                   PERSON    156    30      1
```

2.34.4 ZEIGE-G (Gruppensummen)

Anwendungsart: aktiv

```
ZEIGE[,n],{G/GT/GS},Sortfeld[(Erg1)]_Feld[(Ergn)][_Feld...]...
     [,DB.xxxxxx]
```

(vgl. auch die vereinfachte Syntax der Operandenangabe - Seite 15)

`ZEIGE` Operation für Ausgabe auf Terminal.

`n` Auf Ausgabezeilen bezogene Mengenangabe.
$ = alle Ausgabezeilen aller Sätze (eine Ausgabezeile entspricht jeweils einer Gruppensumme). Bei fehlender Mengenangabe wird '$' angenommen.

`G` Operationsergänzung für Gruppensummen.

`GT` Operationsergänzung für Gruppensummen mit tabellarischen Gesamtsummen.

`GS` Operationsergänzung für Gruppenwechsel mit zeilenweisen und tabellarischen Gesamtsummen.

`Sortfeld` Gruppenwechselbegriff

Erg_1 Ergänzungen zum Gruppenwechselbegriff:

AM = Abschnittsmultiplität: Bei fehlender AM-Angabe wird AM=1 angenommen.
FM = Feldmultiplität: Bei fehlender FM-Angabe wird FM=1 angenommen.
FP = Feldposition (nur bei Feldbedeutung T): Bei fehlender FP-Angabe wird FP=1 angenommen.
FL = Feldlänge: Bei fehlender FL-Angabe wird die definierte Feldlänge verwendet.

SO = UF Ergänzung für Sortierung in absteigender Reihenfolge entsprechend einer USER-Tabelle.
SO = US Ergänzung für Sortierung in aufsteigender Reihenfolge entsprechend einer USER-Tabelle.
SO = F Ergänzung für Sortierung in absteigender Reihenfolge.
SO = S Ergänzung für Sortierung in aufsteigender Reihenfolge.

DB = Name der Satzbeschreibung: Die Angabe ist nur beim Arbeiten im Fenster zulässig.

Feld (2 bis n)	Die Felder müssen numerisch definiert sein (Felder mit Feldbedeutung R oder P). Bei Operationsergänzung GS muß die Anzahl der definierten Dezimalstellen für Feld 2 bis n gleich sein.
Erg_n	Ergänzungen für die Felder 2 bis n: AM = Abschnittsmultiplität: Bei fehlender AM-Angabe wird AM=1 angenommen. FM = Feldmultiplität: Bei fehlender FM-Angabe wird FM=1 angenommen. FP = Feldposition (nur bei Feldbedeutung T): Bei fehlender FP-Angabe wird FP=1 angenommen. FL = Feldlänge: Bei fehlender FL-Angabe wird die definierte Feldlänge verwendet. DB = Name der Satzbeschreibung: Die Angabe ist nur beim Arbeiten im Fenster zulässig.
xxxxxx	Satzbeschreibung

Die Zielpunktliste wird aufsteigend nach dem Gruppenwechselfeld (Sortfeld) sortiert. Solange der Inhalt des Gruppenwechselfeldes gleich bleibt, werden die Felder 2 bis n jeweils addiert. Ändert sich der Inhalt des Gruppenwechselfeldes, so wird eine Gruppensummenzeile ausgegeben (Summe von Feld 2 bis Feld n). Dieser Ablauf wird fortgesetzt bis das Ende der Zielpunktliste erreicht ist oder bis die im Kommando angegebene Anzahl der Ausgabezeilen erreicht ist.

Bei Operationsergänzung GT werden nach der Ausgabe zusätzlich in einer Abschlußzeile die Gesamtsummen für die einzelnen Felder ausgegeben.

Bei Operationsergänzung GS werden die Gesamtsummen sowohl horizontal als auch vertikal gebildet.

Beim Arbeiten im Fenster ist zu berücksichtigen, daß gleiche Feldinhalte zu einer Feldbezeichnung in mehreren Relationszeilen auftreten können. Die Folge ist eine Verfälschung der Gruppensumme für diese Feldbezeichnung.

Beispiele:

```
SU TOUR=65,BIS TOUR=70,DB.STATIK
IM00    ANZAHL ZIELINFORMATIONEN: 19
*
ZEIGE,$,T,TOUR UMSATZ01 - UMSATZ04


TOUR    UMSATZ01   UMSATZ02   UMSATZ03   UMSATZ04

  69   543245.00     432.00    3424.00  432432.00
  66      545.00    4311.50    5600.56    4800.50
  66     4564.12    4511.09    5121.90    4300.90
  66     2364.12    4119.09    4980.50    5089.90
  66     4334.66    5009.50    4500.50    5129.12
  67     4390.16    4809.56    4500.50    5321.77
  67     4590.55    4809.56     380.08    6001.65
  67      -12.65    -112.00     -98.90    6001.65
  67    20906.34   41812.68   62719.02   83625.36
  68      129.86     254.97     380.08   -2909.55
  68    19128.59   38257.18   57385.77   76514.36
  68    22684.09   45368.18   68052.27   90736.36
  68      545.00     254.97     380.08     505.19
  69    21972.99   43945.98   65918.97   87891.96
  69    21901.88   43803.76   65705.64   87607.52
  69    21830.77   43661.54   65492.31   87323.08
  65      900.55     254.97     380.08     505.19
  69    18630.82   37261.64   55892.46   74523.28
  69   166695.39  296582.74   55679.13   74238.84
*
```

ZEIGE

```
*
BL,$,R
*
ZEIGE,$,G,TOUR UMSATZ01 - UMSATZ04


TOUR    UMSATZ01   UMSATZ02   UMSATZ03   UMSATZ04

  65      900.55     254.97     380.08     505.19
  66    11807.90   17951.18   20203.46   19320.42
  67    29874.40   51319.80   67500.70  100950.43
  68    42487.54   84135.30  126198.20  164846.36
  69   794276.85  465687.66  312112.51  844016.68
*
*
BL,$,R
*
ZEIGE,$,GT,TOUR UMSATZ01 - UMSATZ04


TOUR    UMSATZ01   UMSATZ02   UMSATZ03   UMSATZ04

  65      900.55     254.97     380.08     505.19
  66    11807.90   17951.18   20203.46   19320.42
  67    29874.40   51319.80   67500.70  100950.43
  68    42487.54   84135.30  126198.20  164846.36
  69   794276.85  465687.66  312112.51  844016.68
SUMME  879347.24  619348.91  526394.95 1129639.08
*
BL,$,R
*
ZEIGE,$,GS,TOUR UMSATZ01 - UMSATZ04


TOUR    UMSATZ01   UMSATZ02   UMSATZ03   UMSATZ04      SUMME

  65      900.55     254.97     380.08     505.19    2040.79
  66    11807.90   17951.18   20203.46   19320.42   69282.96
  67    29874.40   51319.80   67500.70  100950.43  249645.33
  68    42487.54   84135.30  126198.20  164846.36  417667.40
  69   794276.85  465687.66  312112.51  844016.68 2416093.70
SUMME  879347.24  619348.91  526394.95 1129639.08 3154730.18
*
```

2.34.5 ZEIGE-M (Maske)

Anwendungsart: aktiv

```
ZEIGE[,n],M[,DB.xxxxxx]
```

ZEIGE	Operation für Ausgabe auf Terminal.
n	Auf Sätze bezogene Mengenangabe. Bei fehlender Mengenangabe wird '$' (= alle) angenommen.
M	Operationsergänzung für Maskenausgabe.
xxxxxx	Bildschirmmaske

Das Kommando entspricht dem KORRIGIEREN-Kommando, jedoch findet kein Update statt.

2.34.6 ZEIGE-T (Tabellarisch)

Anwendungsart: aktiv

```
ZEIGE{_/[,n],{T/XT},}{Feld[Erg)]/Var[Kom]}_...[,DB.xxxxxx]
```

(vgl. auch die vereinfachte Syntax der Operandenangabe - Seite 15)

ZEIGE	Operation für Ausgabe auf Terminal.
n	Auf Sätze bezogene Mengenangabe. $ = alle Sätze. Bei fehlender Mengenangabe wird '$' angenommen.
T	Operationsergänzung für tabellarisch.
XT	Operationsergänzung für tabellarisch hexadezimal. Feldlängen werden verdoppelt. Kommazahlen werden ohne Komma dargestellt.
Feld	Name des auszugebenden Feldes.
Ergänzung	AM = Abschnittsmultiplität: Bei fehlender AM-Angabe wird AM=1 angenommen. AZ = Abschnittszyklen: $ = alle Abschnittszyklen. Bei fehlender AZ-Angabe wird AZ=1 angenommen. FM = Feldmultiplität: Bei fehlender FM-Angabe wird FM=1 angenommen. FZ = Feldzyklen: $ = alle Feldzyklen. Bei fehlender FZ-Angabe wird FZ=1 angenommen. FP = Feldposition (nur bei Feldbedeutung T): Bei fehlender FP-Angabe wird FP=1 angenommen. FL = Feldlänge: Bei fehlender FL-Angabe wird die definierte Feldlänge verwendet. DB = Name der Satzbeschreibung: Die Angabe ist nur beim Arbeiten im Fenster zulässig.
Variable	Rechenvariable: Zulässige Angaben sind: (vgl. Seite 115) %X0 ... %X9, %XA ... %XZ %Y0 ... %Y9, %YA ... %YZ
Kommentar	Zur besseren Dokumentation kann der Rechenvariablen ein Kommentar angefügt werden (max. 19stellige alphanumerische Zeichenfolge - vgl. Seite 118)
xxxxxx	Satzbeschreibung

Die Ausgabe der im Kommando angegebenen Felder erfolgt tabellarisch auf dem Terminal.

Die Ausgabelänge der Felder entspricht der in der Datenbeschreibung definierten Feldlänge, wenn sie nicht durch den Ergänzungsparameter FL modifiziert wird. Durch geeignete Feldergänzungen kann auf bestimmte Multiplitäten Bezug genommen werden.

Beispiele:

```
*
ZEIGE,$,T,FIRMA(FL=6) WARENTEXT(AZ=$)

FIRMA  WARENTEXT

SPRAEN SCHOKOLADE
       PRALINEN
       KAKAOPULVER            Satz mit 5 Wiederholabschnitten
       KEKSE
       KAUGUMMI
AILDIS                        Satz ohne Wiederholabschnitt
SAUER  SALZSTANGEN
       ESSIGGURKEN            Satz mit 2 Wiederholabschnitten
*
BL,$,R
*
ZEIGE,$,T,FIRMA WARENTEXT(AM=-2,AZ=$)

FIRMA        WARENTEXT

SPRAENGEL    KEKSE
             KAUGUMMI
AILDIS
SAUER        SALZSTANGEN
             ESSIGGURKEN
*
```

Die Multiplitätsangabe kann auch negativ sein. In diesem Beispiel wird auf alle Multiplitäten ab der zweitletzten Bezug genommen. Dies gilt auch für die FM-Angaben.

2.34.7 ZEIGE-V (Verteilungstafel)

Anwendungsart: aktiv

```
ZEIGE[,n],{V/VS},Sortfeld[(Erg)]_Feld[(Erg)]{*/=/</>/:/#}Wert
     [,Feld...]...[,DB.xxxxxx]
```

ZEIGE	Operation für Ausgabe auf Terminal.
n	Auf Ausgabezeilen bezogene Mengenangabe. $ = alle Druckzeilen. Bei fehlender Mengenangabe wird '$' angenommen.
V	Operationsergänzung für Verteilungstafel.
VS	Operationsergänzung für Verteilungstafel mit Summenbildung.
Sortfeld	Ordinatenfeld der Verteilungstafel.
Feld	Abszissenfelder der Verteilungstafel.
Ergänzung	Sortfeld: FP = Feldposition (nur bei Feldbedeutung T) AM = Abschnittsmultiplität: Bei fehlender AM-Angabe wird AM=1 angenommen. FM = Feldmultiplität: Bei fehlender FM-Angabe wird FM=1 angenommen. DB = Name der Satzbeschreibung: Nur beim Arbeiten im Fenster. Feld: FP = Feldposition (nur bei Feldbedeutung T) AM = Abschnittsmultiplität: Bei fehlender AM-Angabe wird AM=1 angenommen. AZ = Abschnittszyklen: $ = alle Abschnittszyklen. Bei fehlender AZ-Angabe wird AZ=1 genommen. FM = Feldmultiplität: Bei fehlender FM-Angabe wird FM=1 angenommen. FZ = Feldzyklen: $ = alle Feldzyklen. Bei fehlender FZ-Angabe wird FZ=1 angenommen. DB = Name der Satzbeschreibung: Nur beim Arbeiten im Fenster. GR = Gruppenbildung in Intervallen.
Wert	Gesuchte Feldinhalte der Abszissenfelder.
xxxxxx	Satzbeschreibung

Die Verteilungstafel gibt Aufschluß darüber, welche mengenmäßige Beziehung zwischen verschiedenen Feldinhalten besteht.

Die Werte des Ordinatenfeldes (Sortfeld) werden aufsteigend sortiert ausgegeben. Für jeden Feldinhalt der Ordinate wird die Häufigkeit der angegebenen Werte der Abszissenfelder ermittelt. Bei der Operationsergänzung VS werden zusätzlich vertikal und horizontal die Gesamtsummen gebildet.

Die intern verwendeten Summenzähler sind 5-stellig.

Beim Arbeiten im Fenster ist zu berücksichtigen, daß gleiche Feldinhalte zu einer Feldbezeichnung in mehreren Relationszeilen auftreten können. Die Folge ist eine Verfälschung der Gruppensumme für diese Feldbezeichnung.

Beispiel:

```
*
SU ALTER=50,BIS ALTER=65
IM00    ANZAHL ZIELINFORMATIONEN: 155
*
ZEIGE,$,VS,ALTER TYP=AL,TYP=GL,TYP=SBA
ALTER TYP=AL  TYP=GL  TYP=SBA SUMME

   50    2       3       9       14
   51    1       1      17       19
   53    0       0      31       31
   54    0       2      24       26
   56    0       2       1        3
   58    1       0      18       19
   59    0       3       4        7
   60    1       1       3        5
   61    0       1      15       16
   62    0       2      12       14
   63    1       0       0        1
SUMME    6      15     134      155
```

Die Verschlüsselungen im Feld "Typ" haben folgende Bedeutung: AL = Abteilungsleiter
GL = Gruppenleiter
SBA = Sachbearbeiter

Mit der Ergänzung GR=I wird im Trefferfall für den entsprechenden Satz die Verarbeitung abgebrochen. Auf diese Weise lassen sich bei entsprechender Anordnung im Kommando Intervalle bilden:

Beispiel: `Z,$,VS,ALTER GEHALT(GR=I):4000,GEHALT(GR=I):5000...`

```
ALTER  GEHALT:4000  GEHALT:5000  GEHALT:6000  GEHALT:10000  SUMME

   50        10           4           2           1         17
   51         0           1           0           0          1
   .
   .
   .
SUMME        10           5           2           1         18
```

2.34.8 ZEIGE-VB (Verweisdatei-Balkendiagramm)

Anwendungsart: aktiv

```
ZEIGE[,n],VB,Feld[(Erg)][*Wert][,DB.xxxxxx]
```

ZEIGE	Operation für Ausgabe auf Terminal.
n	Mengenangabe, die sich auf invertierte Feldinhalte (VD-Sätze) bezieht. $ = alle Feldinhalte. Bei fehlender Mengenangabe wird '$' angenommen.
VB	Operationsergänzung für Verweisdatei-Balkendiagramm.
Feld	Name des invertierten Feldes, auf das sich die Ausgabe bezieht.
Ergänzung (vgl. Seite 18)	DB = Name der Satzbeschreibung: Nur beim Arbeiten im Fenster.
*	Vergleichsoperator für größer/gleich.
Wert	Invertierter Feldinhalt, ab dem die Ausgabe erfolgen soll.
xxxxxx	Satzbeschreibung

Dieses Kommando stellt die Belegung der Verweisdatei dar. Es wird die Häufigkeitsverteilung der Feldinhalte zu dem angegebenen Feld dokumentiert.

Zu den verschiedenen Feldinhalten wird die Häufigkeit als Zahl ausgegeben. Im Gegensatz zum entsprechenden DRUCKE-Kommando (vgl. Seite 60), wird für die Häufigkeit keine Graphik erstellt. Am Ende der Ausgabe wird die Gesamtanzahl der verschiedenen angesprochenen Feldinhalte und die Gesamtanzahl der Verweise ausgegeben.

Die Ausgabe kann auch erst ab einem bestimmten Feldinhalt, der im Feld *Wert angegeben wird (*für größer/gleich), erfolgen.

Beispiel:

```
*
Z,12,VB,ORT*DARMSTADT,DB.ANWDA9
FELDINHALT(GEKUERZT)  * DARMSTADT                ANZAHL
DARMSTADT                                             1
DEGGENDORF                                            2
DEN                                                   2
DUDWEILER                                             1
DUEREN                                                1
DUESSELDORF                                           5
EINBECK                                               1
ERLANGEN                                             17
ESCHBORN                                              1
ESSEN                                                 3
FORCHHEIM                                             1
FRANKENTHAL                                           2
VERSCHIEDENE FELDINHALTE:        12  SUMME:         37
```

2.34.9 ZEIGE-Z (Zeilenweise)

Anwendungsart: aktiv

```
ZEIGE[,n],Z,{Feld[(Erg)]/Var[Kom]}_...[,DB.xxxxxx]
```

(vgl. auch die vereinfachte Syntax der Operandenangabe - Seite 15)

ZEIGE	Operation für Ausgabe auf Terminal.
n	Auf Sätze bezogene Mengenangabe. $ = alle Sätze. Bei fehlender Mengenangabe wird '$' angenommen.
Z	Operationsergänzung für zeilenweise.
Feld	Name des auszugebenden Feldes.
Ergänzung	AM = Abschnittsmultiplität: Bei fehlender AM-Angabe wird AM=1 angenommen. AZ = Abschnittszyklen: $ = alle Abschnittszyklen. Bei fehlender AZ-Angabe wird AZ=1 angenommen. FM = Feldmultiplität: Bei fehlender FM-Angabe wird FM=1 angenommen. FZ = Feldzyklen: $ = alle Feldzyklen. Bei fehlender FZ-Angabe wird FZ=1 angenommen. FP = Feldposition (nur bei Feldbedeutung T): Bei fehlender FP-Angabe wird FP=1 angenommen. FL = Feldlänge: Bei fehlender FL-Angabe wird die definierte Feldlänge verwendet. DB = Name der Satzbeschreibung: Die Angabe ist nur beim Arbeiten im Fenster zulässig.
Variable	Rechenvariable: Zulässige Angaben sind: (vgl. Seite 115) %X0 ... %X9, %XA ... %XZ %Y0 ... %Y9, %YA ... %YZ
Kommentar	Zur besseren Dokumentation kann der Rechenvariablen ein Kommentar angefügt werden (max. 19stellige alphanumerische Zeichenfolge) vgl. Seite 118.
xxxxxx	Satzbeschreibung

Die Ausgabe der im Kommando aufgeführten Felder erfolgt zeilenweise auf dem Terminal.

Beispiel:

```
*
ZEIGE,1,Z,FIRMA WARENTEXT(AM=-2,AZ=$)
FIRMA                SPRAENGEL
WARENTEXT 004/000    KEKSE
WARENTEXT 005/000    KAUGUMMI
*
```

Die Multiplitätsangabe kann auch negativ sein. In diesem Beispiel wird auf alle Multiplitäten ab der zweitletzten Bezug genommen. Das gilt auch für FM-Angaben.

2.35 $D-Kommando

Anwendungsart: aktiv

Mit den $D-Kommandos werden mehrere Administrationsfunktionen ausgeführt.

Die $D-Kommandos für CISKOOR und CISDBH werden im Manual-2 (Dienstprogramme) beschrieben.

2.36 $P-Kommando

Anwendungsart: passiv

Nur zu verwenden bei unsynchronisiertem Teilhaber-Betrieb!

```
$P,{A/E}
```

$P	Operation für Programm.
A	Ergänzung für Anfang.
E	Ergänzung für Ende.

Der Anwender definiert mit dieser Funktion für jede Anwendung den Programmanfang bzw. das Programmende.

Das $P-Kommando wird im Zusammenhang mit dem $T-Kommando (vgl. Seite 224) verwendet (siehe auch Manual-4: Unsynchronisierter CIS/UTM-Betrieb).

2.37 $T-Kommando

Anwendungsart: passiv

Nur zu verwenden bei unsynchronisiertem Teilhaber-Betrieb!

```
$T,{A/D/E}
```

$T	Operation für Terminal.
A	Ergänzung für Anfang.
D	Ergänzung für Verbindungsabbruch (disconnect).
E	Ergänzung für Ende.

Der Anwender definiert mit dieser Funktion für jede Anwendung den Terminalanfang, und das Terminalende.

$T,D darf unter UTM-Betrieb nicht verwendet werden.

Das $T-Kommando wird im Zusammenhang mit dem $P-Kommando (vgl. Seite 223) verwendet (siehe auch Manual-4: Unsynchronisierter CIS/UTM-Betrieb).

3 CISKURZ (Kurzkommando-Interpreter)

Bei der Anwendung der aktiven Version von CIS müssen häufig gleichartige Kommandos, die sich z.B. nur in einem Feldinhalt unterscheiden, formuliert werden. Bei Massenaufgaben kann sich dadurch ein erheblicher Bedienungsaufwand ergeben.

CISKURZ ermöglicht die Benutzung gespeicherter, vorformulierter Kommandos, wobei ggf. beim Aufruf die CISKURZ-Variablen durch aktuelle Werte ersetzt werden. Die gespeicherten Kommandos können zu einer Prozedur verkettet werden, zu deren Aufruf die Eingabe der Nummer des logisch ersten Kommandos ausreicht. Die Kommandofolge kann Schleifen enthalten und beliebig umfangreich sein. Sie wird in Kommandoschritten abgearbeitet.

Es können alle CIS-Aktiv Kommandos, GET-F als einziges Passiv-Kommado und spezielle CISKURZ-Kommandos (%-Kommandos, vgl. Seite 236) in Kurzkommandoketten verwendet werden. Sämtliche %-Kommandos werden direkt von CISKURZ bearbeitet, alle anderen werden an CISU (vgl. Manual-4: CISU-Benutzeranschluß) weitergereicht.

Mit CISKURZ kann der Bedienungsaufwand für das aktive CIS gesenkt werden. Damit wird gleichzeitig die Fehlerhäufigkeit durch Falscheingaben verringert.

CISKURZ bietet die Möglichkeit der internen Kopplung verschiedener Datenbanksätze durch ihre Feldinhalte. Diese Datenbanksätze können entweder in einer einzigen Datenbank oder in beliebig vielen verschiedenen Datenbanken gespeichert sein.

Der Kurzkommando-Interpreter CISKURZ ist in CIS-Aktiv enthalten.

Im folgenden werden nur die speziellen Möglichkeiten von CISKURZ beschrieben.

3.1 Bedienung

CISKURZ sucht alle angesprochenen Kurzkommandos in einer Datei, die folgende Eigenschaften haben muß:

```
FCBTYPE  = ISAM
RECFORM  = V
KEYPOS   = 5
KEYLEN   = 8
```

Diese Dateieigenschaften sind EDT- bzw. EDOR-kompatibel. Hat die Kommandodatei den Namen CIS.KUKO.BILD, ist kein explizites SET-FILE-LINK Kommando nötig. Hat die Kommandodatei einen anderen Namen, ist vor dem Start von CIS ein SET-FILE-LINK Kommando abzusetzen:

```
/SET-FILE-LINK FILE-NAME=Dateiname,LINK-NAME=KUKO
```

Der Wechsel zwischen mehreren Kommandodateien ist möglich. Dazu muß die aktuelle Kommandodatei geschlossen (vgl. %CLOSE-Kommando Seite 240) und die neue Kommandodatei mit einem SET-FILE-LINK Kommando zugewiesen werden (vgl. dazu auch das SYSTEM-Kommando Seite 168).

Ohne CIS zu beenden kann mit Hilfe des %KUKO-Kommandos (vgl. Seite 251) in den EDT als Unterprogramm verzweigt und die Kurzkommandokette geändert werden.

3.2 Kommandoformulierung

Die Kommandodatei wird mit den Dateiaufbereitungsprogrammen EDT oder EDOR erstellt. Die folgende Beschreibung bezieht sich auf die Bearbeitung mit dem Dienstprogramm EDT. Die EDT-Zeilennummer ist gleichzeitig die Kommandonummer. Dabei ist zu beachten, daß der Dezimalpunkt nur fiktiv ist, d.h. die EDT-Zeilennummer 0047.4711 entspricht der Kommandonummer 474711. Eine abzuändernde Kommandodatei ist mit dem EDT-Kommando

```
@GET 'Dateiname' N
```

einzulesen, damit die Neunumerierung im virtuellen Speicher durch den EDT unterbleibt.

Eine bestehende Kurzkommandodatei kann auch direkt in CIS bearbeitet werden (vgl. %KUKO-Kommando Seite 251).

Ein Kommando in der Kommandodatei beginnt auf Byte 13 (4 Bytes Satzlängenfeld + 8 Bytes Kommandonummer). Jedes Kommando in der Kommandodatei hat formal den Aufbau:

```
nnnnnnnnKommando[:Fs1]Terminator[Fs2][/*Kommentar*/]
```

`nnnnnnnn` Kommandonummer = EDT-Zeilennummer.

`Kommando` Als Kommando sind zugelassen:
- Alle CIS-Aktivkommandos.
- GET,F als einziges Passiv-Kommando (vgl. Seite 80).
- Spezielle CISKURZ-Kommandos (vgl. Seite 236).

Das jeweilige Kommando kann, wenn sinnvoll, mit Hilfe von Variablen modifiziert werden (vgl. Seite 231).

Variablen sind: & wird durch externe Werte ersetzt.
@ wird durch interne Werte ersetzt.

`Fs1` Der Fortsetzungsschlüssel Fs1 wird nur bei einigen speziellen CISKURZ-Kommandos angegeben (vgl. Seite 236).

Als Darstellung sind erlaubt:

E	Ende - es soll nicht weiter fortgesetzt werden.
N	Beim nächstfolgenden Kommando der Datei soll fortgesetzt werden.
nnnnnnnn	Kommandonummer des Forsetzungskommandos. Max. 8-stellig, führende Nullen können entfallen.

`Terminator` Als Terminator sind zulässig:

#	Meldung über Anzahl der Zielinformationen wird unterdrückt.
##	Meldung über Anzahl der Zielinformationen wird ausgegeben.

Der doppelte Terminator hat also nur bei Kommandos eine Bedeutung, die die Anzahl der Zielinformationen ausgeben.

`Fs2` Der Fortsetzungsschlüssel Fs2 kann bei jedem Kurzkommando angegeben werden.

Als Darstellung sind erlaubt:

E	Ende - es soll nicht weiter fortgesetzt werden.
N	Beim nächstfolgenden Kommando der Datei soll fortgesetzt werden.
nnnnnnnn	Kommandonummer des Fortsetzungsschlüssels, max. 8-stellig, führende Nullen können entfallen.

Wird nichts angegeben, so wird nicht weiter verzweigt (gleiche Bedeutung wie E).

`Kommentar` Beliebiger Text als Zeilenkommentar

3.2.1 Beispiele für Kommandoformulierungen

Beispiel 1:

Kommandozeile in einer Kommandodatei:

```
00000010SU NAME(FP=1)=&,DB.PERSNL##     /* SUCHAUFTRAG */
```

Das Kommando 10 enthält die Variable &. Diese wird während des Ablaufs durch einen externen Wert ersetzt (vgl. Seite 231). Die Anzahl der Zielinformationen wird aufgrund der zwei Terminatorzeichen ausgegeben. Es wird zu keinem weiteren Kurzkommando verzweigt, da kein Fortsetzungsschlüssel angegeben ist.

Beispiel 2:

Auszug aus einer Kommandodatei:

```
00000020SU BERUF=&,DB.PERSNL#N            /* INITIAL-SUCHE */
00000030UND DIENSTALTER*&,DB.PERSNL##N    /* EINSCHRÄNKUNG */
00000040Z,10,T,NAME ALTER DIENSTALTER#E   /* BS-AUSGABE    */
```

Bei Aufruf des Kommandos 20 werden alle Mitarbeiter mit einem bestimmten Beruf gesucht (Die Variable & wird während des Ablaufs durch einen externen Wert ersetzt, vgl. Seite 231). Die Anzahl der Zielinformationen wird nicht ausgegeben (nur ein Terminatorzeichen).

Nach Abarbeitung des Kommandos wird Kommando 30 ausgeführt. Das Dienstalter (&) wird durch einen externen aktuellen Parameter ersetzt und die neue Anzahl der Zielinformationen ausgegeben (zwei Terminatorzeichen).

Anschließend wird Kommando 40 ausgeführt. Von den Zielinformationen bzw. den ersten 10 Zielinformationen werden der Name, das Alter und das Dienstalter in Tabellenform gezeigt. Nach dem Kommando wird die Kommandokette beendet, da E als Fortsetzungsschlüssel angegeben ist. Der Fortsetzungsschlüssel muß in diesem Fall nicht angegeben sein, da hier nicht fortgesetzt werden soll.

Beispiel 3:

Auszug aus einer Kommandodatei

```
00000050SU BRANCHE=&,DB.BETRIB##N /*BRANCHE VON BS ANFORDERN*/
00000060SORT &,DB.BETRIB#N        /*SORTIERKRITERIUM VON BS*/
00000070WRITE,$,K,DATEI=AUSGABE#  /*AUSGABEDATEI SCHREIBEN*/
00000080SU UMSATZ*&,DB.BETRIB##60 /*MINDESTUMSATZ VON BS*/
```

Mit Kommando 50 werden alle Betriebe einer bestimmten Branche gesucht (Die Variable & wird im Ablauf durch einen externen Wert ersetzt, vgl. Seite 231), die Anzahl der Zielinformationen wird ausgegeben (zwei Terminatorzeichen).

Nach der Kommandoabarbeitung wird Kommando 60 ausgeführt, d.h. die Zielpunktliste wird aufsteigend nach einem anzugebenden Feld (Variable & wird im Ablauf ersetzt) sortiert, die Anzahl der Zielinformationen wird nicht ausgegeben (nur ein Terminatorzeichen).

Anschließend werden die lokalisierten Sätze in eine Datei geschrieben (Kommando 70) und die Kommandokette beendet.

Wird Kommando 60 direkt aufgerufen, wird eine zuvor erstellte Zielpunktliste sortiert und die dadurch adressierten Sätze in die Datei geschrieben. Es ist also nicht zwingend, mit dem ersten Kommando einer Prozedurkette zu beginnen.

Ebenso können verschiedene Prozedurketten in einem gemeinsamen Zweig enden. Wird z.B. Kommando 80 aufgerufen, so werden alle Betriebe mit einem Umsatz größer/gleich einem bestimmbaren Wert gesucht (Die Variable & wird während des Ablaufs durch einen externen Wert ersetzt - vgl. Seite 231), anschließend sortiert (Kommando 60) und in die Datei geschrieben (Kommando 70).

Es ist nicht zwingend notwendig, das Datenbank-Paßwort für jedes Einzelkommando anzugeben. Da aber jedes Kurzkommando in einer Kommandokette als erstes Kommando aufgerufen werden kann, soll das Datenbank-Paßwort nur in entsprechend geprüften Fällen weggelassen werden.

3.3 Variablensubstitution

CISKURZ kennt zwei Variablentypen: - externe Variable &
- interne Variable @

Innerhalb eines Kurzkommandos können die Variablen bis auf den Terminator an beliebiger Stelle stehen, beide Variablentypen sind innerhalb eines Kommandos zulässig.

Enthält ein Kurzkommando Variablen, so werden sie zuerst substituiert und dann wird das Kommando zur Syntaxprüfung weitergereicht.

1. Die externe Variable &

   Für die Belegung externer Variablen mit Inhalten ist der Anwender selbst zuständig. Dies kann auf zwei Arten geschehen:

   - Wertangabe(n) beim Aufruf der Kurzkommandokette (siehe auch Seite 234).
   - Eingabe des Wertes nach Aufforderung. Gibt es keine aktuellen Werte, so wird die Kurzkommandokette unterbrochen und eine Eingabe erwartet.

2. Die interne Variable @

   Im Gegensatz zur Variablen & kann der Anwender nicht selbständig für eine Substitution der Variablen @ sorgen. Eine interne Variable wird mit einem aktuellen gültigen Wert aus dem GET-F Puffer belegt.

   Der GET-F Puffer ist ein Bereich, der von CISKURZ verwaltet wird. Mit dem sonst nur passiv anwendbaren CIS-Kommando `GET,F,Feldname,DB.xxxxxx` (vgl. Seite 80) läßt sich dieser Bereich mit Daten belegen (siehe auch Beispiel 2 - Seite 233). GET-F Einträge können in beliebiger Menge abgelegt werden. Sie werden der Reihe nach zum Ersetzen (Substituieren) der Variablen @ verwendet.

   Alle Variablen @ werden der Reihe nach durch den jeweils gültigen GET-F Eintrag ersetzt. Sobald ein GET-F Eintrag zur Variablensubstitution verwendet wurde, wird er ungültig gesetzt, der nächste GET-F Eintrag des CISKURZ-Puffers wird jetzt zum aktuell gültigen.

   Jeder Variablen @ muß in der logischen Folge der Kommandokette ein gültiger GET-F vorausgehen. GET-F ohne nachfolgende Variable @ ist unschädlich aber wirkungslos.

   Nicht aktualisierte Variablen @ (vorausgegangener GET-F fehlt) führen zur Fehlermeldung:

```
KU27   KEIN/ZU WENIGE GET,F VOR '@'
```

Beispiele zur Variablensubstitution

Beispiel 1: Substitution externer Variablen

Kommandozeile:

```
00000054SU NAME=&,DB.xxxxxx##E
```

1. Ablaufmöglichkeit:

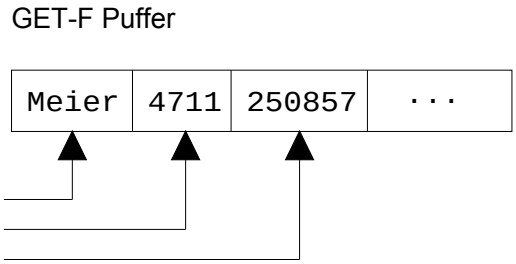
`*`
`54`
`ERSETZEN SIE VARIABLE`
`SU NAME=`
`*`
	Aufruf ohne Parametereingabe. Aufforderung zur Eingabe des aktuellen Parameters.

`MEIER`	Die Eingabe des Wertes ergibt intern das Kommando: `SU NAME=MEIER,DB.xxxxxx`
`IM00  ANZAHL ZIELINFORMATIONEN: 27` `*`	Trefferanzahl wird gemeldet.

2. Ablaufmöglichkeit:

`*` `54,MEIER`	Aufruf mit Parametereingabe. Die Kommandonummer wird mit einem aktuellen Wert (MEIER) eingegeben. Dieser ersetzt bei Abarbeitung des Kommandos durch CISKURZ die Variable &.
`IM00  ANZAHL ZIELINFORMATIONEN: 27`	Trefferanzahl wird gemeldet.

Beispiel 2: Substitution interner Variablen




```
00001000GET,F,NAME,DB.PERSON#N
00001010GET,F,Nummer#N
00001020GET,F,GEBURTSDATUM#N
00002000SU NAME=@,UND KENNZEICHEN=@,DB.STAMM1#N
00002010AE,1,F,DATUM=@#N
  .
  .
  .
```

Das Kommando 2000 bewirkt eine Suche in der STAMM-Datei nach dem Namen 'MEIER' mit dem Kennzeichen '4711'. Als gültiger Puffer-Eintrag ist nur noch der Wert '250857' vorhanden. Anschließend wird das Geburtsdatum im Stammsatz eingetragen.

Weitere Beispiele sind auf den Seiten 274 und 275 zu finden.

3.4 Aufruf

Nach dem Start von CIS können in beliebiger Reihenfolge CIS-Kommandos eingegeben und Kurzkommandoprozeduren gestartet werden.

Kurzkommandoprozeduren werden anhand der Kommandonummer identifiziert, d.h. ist das erste Zeichen numerisch, handelt es sich um eine Kurzkommandoprozedur, sonst wird angenommen, daß es ein normales CIS-Aktiv Kommando ist.

Syntax der Kurzkommandoeingabe:

```
Kdo-Nr.[,Wert]...
```

Die Werte ersetzen (substituieren) in der angegebenen Reihenfolge die externen Variablen (&) der Kurzkommandokette (vgl. Seite 231).

Werden zu viele Werte angegeben, so werden die überflüssigen nicht abgerufen.

Werden zu wenig Werte angegeben, so werden die restlichen externen Variablen der Kurzkommandokette mit dem letzten angegebenen Wert ersetzt.

Werden keine Werte angegeben, so werden sie im Ablauf angefordert.

3.5 Datenkopplung

Sinnvollerweise können Kurzkommandos nur solche CIS-Aufrufe enthalten, die aktiv zugelassen sind.

Um den Aufbau von komfortablen Kommandoketten und Kommandoschleifen zu gewährleisten, gibt es spezielle CISKURZ-Kommandos, die ab Seite 236 näher beschrieben werden. Mit CISKURZ ist Datenkopplung möglich. Feldinhalte können aus einem vorhandenen CIS-Satz geholt und an beliebiger Stelle in ein Kurzkommando eingefügt werden. Dabei werden die Feldinhalte mit dem sonst nur passiv anwendbaren CIS-Kommando `GET,F,Feld,DB.xxxxxx` im CISKURZ-Puffer gespeichert und damit für die spätere Substitution der internen Variablen @ zur Verfügung gestellt (vgl. Seite 231). Mit dem GET-F Kommando kann auch der Inhalt von temporären Rechenvariablen (%X1....) in den GET-F Puffer übertragen werden.

Bei der Kopplung ist Datenbank-/Datenbeschreibungswechsel erlaubt.

Beispiel zur Datenkopplung

Problemstellung

Anhand einer bestimmten Produktnummer, die in der Produktdatei gespeichert ist, soll der zugehörige Lieferant aus der Lieferantendatei gesucht und dessen Anschrift auf dem Bildschirm ausgegeben werden. Die Produktdatei heißt HD.PRODU1 und ist mit der Satzbeschreibung DB.PRODU1 beschrieben. Die Lieferantendatei heißt HD.LIEFER und ist mit der Satzbeschreibung DB.LIEFER beschrieben.

In der Produktdatei stehen u.a. folgende Daten:

- PRODUKTNUMMER
- PRODUKTBEZEICHN
- PRODUKTPREIS
- LIEFERANTEN-NR

In der Lieferantendatei stehen u.a. folgende Daten:

- NAME
- STRASSE
- PLZ
- ORT
- LIEFERANTEN-NR

Problemlösung mit einer Kurzkommandokette:

```
00000010SU PRODUKTNUMMER=&,DB.PRODU1#N  /* PRODUKTNUMMER SUCHEN*/
00000020GET,F,LIEFERANTEN-NR,DB.PRODU1#N/* LIEFERT.-NR MERKEN  */
00000030SU LIEFERANTEN-NR=@,DB.LIEFER#N /* LIEFERANT SUCHEN    */
00000040Z,1,T,NAME STRASSE PLZ ORT#E    /* ERGEBNIS AUSGEBEN   */
```

Kommando 10 sucht nach einer bestimmten PRODUKTNUMMER in der Produktdatei (die Variable & wird im Ablauf durch einen externen aktuellen Wert ersetzt, vgl. Seite 231). Das Kommando 20 übergibt die LIEFERANTEN-NR des lokalisierten Satzes an CISKURZ. Im weiteren Ablauf wird die Variable @ (Kommando 30) von CISKURZ durch den GET-F Eintrag (LIEFERANTEN-NR) ersetzt und damit in der Lieferantendatei danach gesucht. Kommando 40 gibt dann den NAMEN und die Anschrift (STRASSE PLZ ORT) aus, die Kommandokette wird beendet.

Falls die Felder PRODUKTNUMMER und LIEFERANTEN-NR invertiert sind, bietet sich folgende (performantere) Lösung an:

```
00000010SU PRODUKTNUMMER=&,DB.PRODU1#N
00000020VE,A,PRODUKTN(DB=PRODU1)=LIEFERA(DB=LIEFER),DB.JOJOJO#N
00000040Z,1,T,NAME STRASSE PLZ ORT#E
```

3.6 Spezielle CISKURZ-Kommandos

Alle CISKURZ-Kommandos beginnen mit dem Zeichen %.

Die speziellen CISKURZ-Variablen &, @ und (AM=$, FM=$) werden auch in den %-Kommandos durch aktuelle Einträge ersetzt. So können z.B. Kommentarzeilen zur Ausgabe aktueller Werte benützt werden.

3.6.1 Tabellarische Übersicht der CISKURZ-Kommandos

CISKURZ-Kommando	Erläuterung	K	D
%AZI{=/#/>/</>=/=>/<=/=</></<>}Anzahl :Fs1#[Fs2]	Anzahl Zielinformationen	X	
%CISGEN_Kommando	CISGEN-Aufruf		X
%CLOSE	Schließen KUKO		X
%CM=Code:Fs1#[Fs2]	CIS-Quittungen	X	
%GETx[xxx]	Aktualisierung Konstante	X	
%GET_BUFFER[+d]_{B/C/H/P/R}l[,dez]	Unbenannte Variable aus EB setzen	X	
%GOSUB:Fs1#[Fs2]	Unterprogrammaufruf	X	
%IF[{>/</=/:/*/#}]x[xxx]:Fs1#[Fs2]	Vergleich	X	
;Kommando-1[;Kommando-n]...	Kommandokettung		X
%_Kommentar	Externer Kommentar	X	
%*Kommentar	Interner Kommentar	X	
%KUKO	KUKO-Datei bearbeiten		X
%MODIFY_{BUFFER[+d]={B/C/H/P/R/X}m'Wert' /LENGHT=1}	Verändern Passiv-Puffer EB/LEB	X	X
%PASSIV_{OFF/ON}	Passivbetrieb aus/ein	X	X
%PAUS/%PEIN	Protokollierung aus/ein	X	X
%RANDOM_(min,max)	Zufallszahl erzeugen	X	
%RETURN	Rückkehr aus Unterprog.	X	
%SET[:AM=xxx][:FM=xxx][:Fs1]#[Fs2]	Multiplitätenbehandlung	X	
%SHOW_{BUFFER[+d]fl/LENGTH}	EB/LEB ausgeben		X
%STOP[:[Fs1][(As[:As]...)][:NO-END]] #[Fs2]	Unterbrechung	X	
%STOREx[xxx][=Konstante[,R]]	Konstanten speichern	X	
%SYS[_/BS2000-Kommando]	BS2000 Systemkdo.	X	

K: Als Kurzkommando in Kurzkommandoketten
D: Als Dialogkommando

3.6.2 %AZI (Anzahl der Zielinformationen)

```
nnnnnnnn%AZI{=/#/>/</>=/=>/<=/=</></<>}Anzahl:Fs1 Term.[Fs2]
```

nnnnnnnn	Kommandonummer
%AZI	Operation für die Abfrage der Anzahl Zielinformationen.
=	gleich
#	ungleich
>	größer
<	kleiner
>=	größer oder gleich
=>	gleich oder größer
<=	kleiner oder gleich
=<	gleich oder kleiner
><	ungleich
<>	ungleich
Anzahl	Anzahl abzufragender Zielinformationen.
:	Syntaxtrenner
Fs1	Fortsetzungsschlüssel (vgl. Seite 227) wenn der Vergleich wahr ist.
Term.	Terminator (vgl. Seite 228).
Fs2	Fortsetzungsschlüssel (vgl. Seite 228) wenn der Vergleich falsch ist.

Die %AZI-Variable sollte direkt nach dem CIS-Kommando, das eine ZPL liefert, abgefragt werden, denn jedes folgende CIS-Kommando setzt die Variable zurück. Demnach ist %AZI nach CIS-Kommandos, die keine AZI zurückmelden, nicht definiert und damit nicht abfragbar.

Beispiel:

Die Aufgabenstellung und die Datenbasis des folgenden Beispiels ist mit Beispiel 2 auf Seite 242 identisch.

Die Lösung über Kurzkommandoschleife wurde etwas abgeändert. Es wurden zwei Kommandos eingefügt (Kommando 15 und Kommando 110):

```
00000010SU PRODUKTPREIS=&,BIS PRODUKTPREIS=&,DB.PRODU1#N
00000015%AZI=0:105#N                /* FALSCHE PREISGRUPPE */
00000017%*ES WURDEN TREFFER ERZIELT (AZI>0)#N
00000020SI,E,PRODUKTBEZEICHN=ALLE,DB.PRODU1#N    /* F LOOP */
00000030GET,F,LIEFERANTEN-NR,DB.PRODU1#N    /* L-NR MERKEN */
00000040SU LIEFERANTEN-NR=@,DB.LIEFER#N    /* ZUGEHOER LIEF*/
00000050D,1,Z,LIEFERANTENNAME STRASSE PLZ ORT,DB.LIEFER#N
00000060AK,E,PRODUKTBEZEICHN=ALLE,DB.PRODU1#N /* DB-WECHSEL*/
00000070BL,1,V#N /* NAE PRODUKT */
00000080%CM=IM01:100#20  /* ENDE ? */
00000100% ALLE LIEFERANTEN AUSGEGEBEN#E        /*NORMAL END*/
00000105%*KEINE TREFFER (AZI=0)#N
00000110% KEINE PRODUKTE IN DIESER PREISGRUPPE VORHANDEN#E
```

Im Kommando 15 wird abgefragt, ob Zielinformationen vorhanden sind. Sind keine Treffer vorhanden, wird der Kommentar von Kommando 110 ausgegeben. Sind Zielinformationen vorhanden, wird bei Kommando 20 fortgesetzt.

3.6.3 %CISGEN (CISGEN-Aufruf)

```
nnnnnnnn%CISGEN_Kommando
```

nnnnnnnn	Kommandonummer
%CISGEN	Operation für CISGEN-Aufruf.
Kommando	CISGEN-Kommando

Zum Nachladen muß das SET-FILE-LINK Kommando

```
/SET-FILE-LINK FILE-NAME=MODLIB.CIS.n,LINK-NAME=BLSLIB
```

gegeben werden.

3.6.4 %CLOSE (Schließen der KUKO)

```
%CLOSE
```

%CLOSE Operation für das Schließen der Kurzkommandodatei.

Mit diesem Kommando wird die aktuelle Kurzkommandodatei (KUKO) geschlossen.

Es besteht die Möglichkeit, ohne CIS zu beenden, eine neue Kurzkommandodatei zuzuweisen.

Mit dem SYSTEM-Kommando (vgl. Seite 168) läßt sich nach dem %CLOSE ein SET-FILE-LINK Kommando absetzen (LINK-NAME=KUKO), mit dem eine neue Kurzkommandodatei zugewiesen wird.

Beispiel:

```
.
.
.
*%CLOSE
*SY /SET-FILE-LINK FILE-NAME=KUKO.NEU,LINK-NAME=KUKO
*
.
.
.
```

3.6.5 %CM (CIS-Quittungen)

```
nnnnnnnn%CM=Code:Fs1 Terminator[Fs2]
```

nnnnnnnn	Kommandonummer
%CM=Code	Operation für die Abfrage der codierten Meldung: Der Code ist die 4-stellige codierte Meldung, die von CIS nach jedem Verarbeitungsschritt ausgegeben wird (vgl. Manual-4: Meldungen).
:	Syntaxtrenner
Fs1	Fortsetzungsschlüssel (vgl. Seite 227)
Terminator	vgl. Seite 228
Fs2	Fortsetzungsschlüssel (vgl. Seite 228)

Ist die während der Verarbeitung erzeugte CIS-Quittung mit dem im Kurzkommando angegebenen Code identisch, so wird nach Fs1 verzweigt, bei Ungleichheit nach Fs2.

Bei CIS-Fehlerquittungen wird eine Kurzkommandokette normalerweise abgebrochen und die Quittung (codierte Meldung) mit der entsprechenden verbalen Meldung ausgegeben. Wird diese Quittung aber mit dem %CM-Kommando abgefragt, so unterbleiben Abbruch und Ausgabe.

CISKURZ-Fehlermeldungen (KUxx) können nicht abgefragt werden.

Beispiel 1

```
00000010SU FELD=&,DB.PROD01#N    /* SAETZE SUCHEN */
00000020SI,V,PSEUDO=1#N          /* SICHERN IN VD */
00000030%CM=ZP10:N#N             /* AKZEPTIERE: AZI = 0 */
```

In einer Kurzkommandoschleife wird nach einem bestimmten Feldinhalt gesucht. Die Zielpunktliste soll permanent in der Verweisdatei unter PSEUDO=1 gespeichert werden. PSEUDO ist ein Pseudofeld, mit dessen Hilfe bestimmte Bearbeitungsformen möglich sind (vgl. 'D-Segment Satzbeschreibung' und 'Zielpunktlistenbehandlung'). Wenn die Suchfrage keine Zielinformationen liefert, würde die Schleife normalerweise beim SICHERN-Kommando abgebrochen. Um das zu vermeiden, wird die entsprechende CIS-Quittung ("ZP10 KEINE ZPL ZUM SICHERN VORHANDEN") abgefragt und im Gleichheits- und Ungleichheitsfall zum nächsten Kommando verzweigt. Diese Anwendung des %CM-Kommandos wird auch im Fallbeispiel 1 (vgl. Seite 264) genutzt.

Beispiel 2

Problemstellung:

Es sollen die Anschriften aller Lieferanten, deren Produkte innerhalb angebbarer Preisgrenzen liegen, getrennt auf eine Liste ausgegeben werden. Die entsprechenden Daten stehen in der Produktdatei HD.PRODU1, beschrieben mit der Satzbeschreibung DB.PRODU1 und in der Lieferantendatei, beschrieben mit der Satzbeschreibung DB.LIEFER.

In der Produktdatei stehen u.a. folgende Daten:

- PRODUKTNUMMER
- PRODUKTBEZEICHN
- PRODUKTPREIS
- LIEFERANTEN-NR

In der Lieferantendatei stehen u.a. folgende Daten:

- LIEFERANTENNAME
- STRASSE
- PLZ
- ORT
- LIEFERANTEN-NR

Problemlösung über Kurzkommandoschleife:

```
00000010SU PRODUKTPREIS=&,BIS PRODUKTPREIS=&,DB.PRODU1#N
00000020SI,E,PRODUKTBEZEICHN=ALLE,DB.PRODU1#N
00000030GET,F,LIEFERANTEN-NR,DB.PRODU1#N
00000040SU LIEFERANTEN-NR=@,DB.LIEFER#N
00000050D,1,Z,LIEFERANTENNAME STRASSE PLZ ORT,DB.LIEFER#N
00000060AK,E,PRODUKTBEZEICHN=ALLE,DB.PRODU1#N
00000070BL,1,V#N
00000080%CM=IM01:100#20
00000100% ALLE LIEFERANTEN AUSGEGEBEN#E
```

Kommando 10 sucht in der HD.PRODU1 alle Produkte innerhalb eines angebbaren Preisintervalls. Mit Kommando 20 wird die Zielpunktliste temporär gesichert. Die erste LIEFERANTEN-NR wird im CISKURZ-Puffer gespeichert (Kommando 30), um sie für das nachfolgende SUCHE-Kommando sicherzustellen. Kommando 40 sucht nach der vorher gespeicherten LIEFERANTEN-NR in der Datei HD.LIEFER. Die Anschrift des gefundenen Lieferanten wird durch Kommando 50 ausgegeben. Die vorher gesicherte Zielpunktliste wird mit Kommando 60 wieder aktiviert. Blättern auf den nächsten Satz (Kommando 70). In Kommando 80 wird abgefragt, ob keine Zielinformationen mehr da sind. Ist die Zielpunktliste abgearbeitet, so wird der nachfolgende Kommentar (Kommando 100) ausgegeben und die Kette somit beendet. Sind noch Zielinformationen vorhanden, so wird nach Kommando 20 verzweigt und die Zielpunktliste mit aktuellem Satzzeiger wieder temporär gesichert. Danach wird die neue LIEFERANTEN-NR im CISKURZ-Puffer für das anschließende SUCHE-Kommando gespeichert usw..

3.6.6 %GET (Aktualisierung von Konstanten)

```
nnnnnnnn%GETx[xxx]Terminator[FS2]
```

nnnnnnnn	Kommandonummer
%GET	Operation für das Übertragen eines Wertes in den GET-F-Puffer.
x[xxx]	Speicherkennung (max. 4-stellige alphanumerische Zeichenfolge, so wie sie im vorherigen %STORE-Kommando angegeben wurde).
Terminator	vgl. Seite 228
Fs2	Fortsetzungsschlüssel (vgl. Seite 228).

Ein Wert, der mit dem %STORE-Kommando (vgl. Seite 262) unter einer Speicherkennung x[xxx] hinterlegt wurde, läßt sich mit %GETx[xxx] aus dieser Speicherkennung in den GET-F-Puffer übertragen.

Beispiel:

```
00000001%    *** BITTE DEN NAMEN EINGEBEN ***    #N
00000002%                                        #N
00000010%STOREA=&#N             /* NAME VON BS */
00000020%GETA#N
00000030SUCHE NAME=@,DB.PERSON#N/*IN AKTUELLER DATENBANK ?*/
00000040%AZI>0:90#N             /* JA */
00000050%GETA#N
00000060SUCHE NAME=@,DB.PERALT#N/* IN ARCHIV-DATENBANK ? */
00000070%AZI>0:110:#N           /* JA */
00000080%NAME NICHT GEFUNDEN#E
00000090%AKTUELLE MITARBEITER#N
00000100Z,$,T,NAME ...#E
00000110%FRUEHERE MITARBEITER#N
00000120Z,$,T,NAME ...#E
```

Mit Kommando 10 wurde eine einzugebende Konstante unter der Speicherkennung A hinterlegt, die mit Kommando 20 bzw. 50 wieder aktualisiert wird.

3.6.7 %GET-BUFFER (Unbenannte Variable setzen)

```
nnnnnnnn%GET_BUFFER[+d]_{B/C/H/P/R}l[,dez]Terminator[Fs2]
```

nnnnnnnn	Kommandonummer
%GET	Operation für unbenannte Variable @ aus EB setzen.
BUFFER	Interner Satzbereich
d	Distanz im internen Satzbereich, dezimal oder hexadezimal: X'd' Bei fehlender Angabe wird 0 angenommen.
B/C/H/P/R	Format der betreffenden Stelle, aus der eine unbenannte Variable erzeugt werden soll: B: binär C: Zeichen (Character) H: gepackt ohne Vorzeichen P: gepackt R: rechtsbündig
l	Länge des Feldes in Bytes.
dez	Anzahl Dezimalstellen (Nur bei den Formaten H,P und R möglich.)
Terminator	vgl. Seite 228
Fs2	Fortsetzungsschlüssel Fs2 (vgl. Seite 228).

Beispiel:

00000010%PASSIV ON	Schaltet Passiv-Verarbeitung ein.
00000020GET,K	Liest Satz von CIS in den internen Bereich.
00000030%GET B+4 R8	Von Stelle 4 in Länge 8 aus dem internen Bereich werden 8 Stellen als Zahl (R) in den @-Buffer gestellt.
00000040%GET B+42 C30	Von Stelle 42 in Länge 30 aus dem internen Bereich werden 30 Stellen Text in den @-Buffer gestellt.
00000050% @ @	Die beiden @-Sätze werden zur Ausgabe gebracht. z.B.: _+11_Text aus dem 1. Satz (30).
00000060BL,1,V	CIS-Satzzeiger wird um 1 weitergeschaltet.
00000070%CM=IM00:20	Solange der CIS-Zeiger nicht das Ende erreicht (IM01) wird auf das Kommando 20 verzweigt. Der nächste Satz wird in den Puffer gebracht.

3.6.8 %GOSUB (Unterprogramm-Aufruf)

```
nnnnnnnn%GOSUB:Fs1 Terminator[Fs2]
```

nnnnnnnn	Kommandonummer
%GOSUB	Operation für Unterprogrammaufruf.
:	Syntaxtrenner
Fs1	Kurzkommandoschlüssel, der den Anfang eines Unterprogramms kennzeichnet (vgl. Seite 227).
Terminator	vgl. Seite 228
Fs2	Kurzkommandoschlüssel, bei dem nach Abschluß des Unterprogramms fortgesetzt wird (vgl. Seite 228).

Innerhalb einer Kurzkommandokette können Teilstücke als Unterprogramm angesprungen werden.

Den Abschluß des Unterprogramms bildet das Kurzkommando %RETURN (vgl. Seite 256).

Unterprogramme können auch geschachtelt werden (vgl. Beispiel auf Seite 256).

3.6.9 %IF (Vergleich)

```
nnnnnnnn%IF[{>/</=/:/*/#}]x[xxx]:Fs1 Terminator[Fs2]
```

nnnnnnnn	Kommandonummer
%IF	Operation für Vergleich.
>	größer
<	kleiner
=	gleich
:	kleiner gleich
*	größer gleich
#	ungleich
x[xxx]	Speicherkennung: Max. 4-stellige alphanumerische Zeichenfolge, die mit keiner Ziffer beginnt.
:	Syntaxtrenner
Fs1	Sprungadresse, wenn der Vergleich wahr ist (vgl. Seite 227).
Terminator	vgl. Seite 228
Fs2	Sprungadresse, wenn der Vergleich falsch ist (vgl. Seite 228).

Fehlt der Vergleichsoperator, so wird als Vergleichsoperation ' = ' angenommen.

Das Kommando %IF vergleicht den letzten gültigen GET-F Feldinhalt (LIFO-Prinzip) mit dem Feldinhalt, der von einem entsprechendem %STORE-Kommando (vgl. Seite 262) gespeichert wurde. Die Speicherkennung muß in den beiden zusammengehörenden Kommandos gleich sein.

Ist der Vergleich der beiden Feldinhalte wahr, so wird nach Fs1 verzweigt, ansonsten nach Fs2.

Beispiel:

Bei jedem Vertreter sollen die getätigten Umsätze aller betreffenden Kunden als Gesamtsumme ausgegeben werden:

Kurzkommandokette:

```
00000010IG,Z,DB.KUNDEN#N    /* EVTL. ZPL LOESCHEN */
00000020SORT VERTRETER#N    /* GANZE DB SORTIEREN */
00000025RE %X=0#N           /* ALLE RECHENFELDER LOESCHEN */
00000030GET,F,VERTRETER#N   /* VERTRETER AUS 1. KUNDENSATZ*/
00000040%STOREA#N           /* MERKEN IN A */
00000050RE,1,S,%X1=UMSATZ#N /* VERTRETER-UMSATZ ADDIEREN */
00000060BL,1,V#N            /* NAECHSTER KUNDENSATZ */
00000070%CM=IM01:200#N      /* ENDE ? */
00000080GET,F,VERTRETER#N   /* VERTRETER AUS AKTUELLEM SATZ*/
00000090%IFA:50#N           /* VERTRETERWECHSEL ? */
00000100Z,1,Z,VERTRETER %X1GESAMTUMSATZ#25/* ERG. AUSGEBEN */
00000200Z,1,Z,VERTRETER %X1GESAMTUMSATZ#N /* ERG. AUSGEBEN */
00000210RE %X=0#E           /* ALLE RECHENFELDER LOESCHEN */
```

Kommando 10 bewirkt, daß alle Sätze der KUNDEN-Datei bearbeitet werden (denselben Effekt hätte auch ein SUCHE-Kommando). Die Sätze werden anschließend nach Vertretern sortiert (Kommando 20). Alle Rechenvariable (%X) werden prophylaktisch auf Null gesetzt. Der erste Vertreter wird unter der Speicherkennung A gespeichert (Kombination Kommando 30 und Kommando 40). Der Wert des Umsatzfeldes wird in die Rechenvariable %X1 addiert (Kommando 50). Mit Kommando 60 wird auf den nächsten Satz geblättert. Kommando 70 fragt ab, ob das Ende der Zielpunktliste erreicht ist. Ist das der Fall wird zu Kommando 200 verzweigt, sonst zu Kommando 80. Dieses stellt den aktuellen Wert des Feldes VERTRETER für die nachfolgende Vergleichsfrage im CISKURZ-Puffer zur Verfügung. Kommando 90 vergleicht den letzten mit GET,F sichergestellten Feldinhalt mit dem Wert, der durch das %STORE-Kommando unter der Speicherkennung A abgespeichert wurde. Sind die Werte gleich, beginnt die Schleife mit dem RECHNE-Kommando (Kommando 50) wieder von vorne. Bei Ungleichheit wird der Vertreter und die Gesamtumsatzsumme ausgegeben (Kommando 100) und nach Kommando 25 verzweigt (Gruppenwechsel, neuer Schleifendurchlauf).

Kommando 200 kann nur nach Abarbeitung von Kommando 70 angesprungen werden, und zwar dann, wenn kein Satz mehr vorhanden ist. Kommando 200 gibt den letzten bearbeiteten Vertreter mit seiner Gesamtumsatzsumme aus. Anschließend werden alle Rechenvariable wieder auf Null gesetzt und die Kette beendet.

3.6.10 ;Kommando (Kommandokettung)

```
;Kommando-1[;Kommando-n]...
```

Kommando CISKURZ-Kommando

Es können beliebig viele CIS-Kommandos und Kurzkommandos, getrennt durch ein Semikolon, aneinandergereiht werden. Diese werden so gespeichert als ob sie aus einer Kurzkommandoprozedur gelesen würden. Sie sind ab 1 mit einer Schrittweite von 1 durchnumeriert.

Terminator und FS2 werden nicht, FS1 wird angegeben. (Vgl. allgemeines Format auf Seite 227)

Beispiel:

```
;GET,F,NAME;%_@;BL,1,V;%CM=IM00:1
```

Solange der BL,1,V IM00 meldet verzweigt das letzte Kommando, %CM, auf das erste, GET,F.

Diese Art der Kettung ist nur am Kommando-Prompt "*" möglich, nicht aus der KUKO-Datei.

3.6.11 %Kommentar (Zeilen-, externer, interner Kommentar)

% externer Kommentar

```
nnnnnnnn%_Kommentar Terminator[Fs2]
```

nnnnnnnn	Kommandonummer
%	Operation für externen Kommentar.
Kommentar	Kommentar: Beliebige alphanumerische Zeichenfolge, die nicht mit * oder einem der Kurzkommandos beginnen darf.
Terminator	vgl. Seite 228.
Fs2	Fortsetzungsschlüssel (vgl. Seite 228). Bei einem Kommentar kann nur ein Fortsetzungsschlüssel angegeben werden.

Der externe Kommentar erläutert den Kommandoablauf und wird immer auf SYSOUT ausgegeben. Dieser Kommentar wird durch das Zeichen % eingeleitet. Da jedes spezielle CISKURZ-Kommando durch das Zeichen % eingeleitet wird, wird jedes ungültige CISKURZ-Kommando als Kommentar interpretiert.

Beispiel:

```
00000010% DIES IST EIN EXTERNER KOMMENTAR#E
```

Hinweis: Bei diesem Kommando ist es empfehlenswert zwischen dem %-Zeichen und dem eigentlichen Kommentar ein Space einzusetzen. Dadurch wird verhindert, daß Konflikte mit künftigen CISKURZ-Kommandos auftreten können.

%* interner Kommentar

```
nnnnnnnn%*Kommentar Terminator[Fs2]
```

nnnnnnnn	Kommandonummer
%*	Operation für internen Kommentar.
Kommentar	Kommentar: Beliebige alphanumerische Zeichenfolge.
Terminator	vgl. Seite 228.
Fs2	Fortsetzungsschlüssel (vgl. Seite 228): Bei einem internen Kommentar kann nur ein Fortsetzungsschlüssel angegeben werden.

Der interne Kommentar dient der Erläuterung der Kurzkommando-Datei und wird nur bei %PEIN (Kurzkommando-Testhilfe, vgl. Seite 254) auf SYSOUT ausgegeben.

Zeilenkommentar

Jede Kommandozeile kann beliebig kommentiert werden. Der Zeilenkommentar wird eingeschlossen zwischen /* und */.

Der Zeilenkommentar wird im Gegensatz zum externen und zum internen Kommentar nicht über den Bildschirm ausgegeben, auch dann nicht, wenn die Protokollierung eingeschaltet ist.

3.6.12 %KUKO (Kurzkommandodatei bearbeiten)

```
%KUKO
```

Mit diesem Kommando wird die aktuelle Kurzkommandodatei (KUKO) in den EDT eingelesen und in den EDT als Unterprogramm verzweigt.

Damit besteht die Möglichkeit die Kurzkommandokette zu bearbeiten, ohne CIS zu beenden.

Der EDT-Modus wird mit dem HALT-Kommando beendet, wobei die Kurzkommandodatei automatisch zurückgeschrieben wird.

Es kann immer nur eine Kurzkommandodatei bearbeitet werden. Vor Bearbeitung muß eine bereits eventuell geöffnete Kurzkommandodatei mit dem %CLOSE-Kommando (vgl. Seite 240) geschlossen werden. Die neu zu bearbeitende Kurzkommandodatei wird nach dem SYSTEM-Kommando (vgl. Seite 168) mit dem SET-FILE-LINK Kommando (LINK-NAME=KUKO) geöffnet.

Hinweis: Soll der automatische Update der KUKO verhindert werden, so ist der EDT von einer leeren Prozedurebene aus zu verlassen.

CIS meldet:

```
KU43 PROZEDUR-/KURZKOMMANDODATEI N I C H T GESCHRIEBEN
```

3.6.13 %MODIFY-BUFFER (Verändern Passiv-Buffer)

```
nnnnnnnn%MODIFY_{BUFFER[+d]={B/C/H/P/R/X}m'Wert'/LENGTH=l}
        Terminator[Fs2]
```

nnnnnnnn	Kommandonummer
%MODIFY	Operation für Veränderung Passiv-Buffer.
BUFFER	Operationsergänzung für den internen Bereich, der in CISKURZ zur Aufnahme des Satzes bei der Passivverarbeitung zur Verfügung gestellt wird.
d	Distanz im internen Satzbereich, dezimal oder hexadezimal X'd' Bei fehlender Angabe wird 0 angenommen.
B/C/H/P/R/X	Format, das an der angegebenen Adresse aus dem Eingabestring erzeugt werden soll: B: binär C: Zeichen (Character) H: gepackt ohne Vorzeichen P: gepackt R: rechtsbündig X: hexadezimal
m	Länge des Feldes in Bytes. Das Feld wird formatgerecht eingerichtet. Die Längenangabe ist nicht als Beschränkung zu verstehen. Beispiel: C2'AAAAA' überschreibt 5 Bytes mit A C5'AA' löscht 5 Bytes mit Space und trägt dann 2 Bytes mit AA ein. %MODIFY B+12=B3'10' Stelle 12 im Satz wird mit '00000A'überschrieben.
LENGTH	Operationsergänzung für die Länge des Zielbereichs.
l	Länge des Zielbereichs (LEB) (vgl. GET,K auf den Seiten 87, 90, 91 und 92). Entspricht dem LEB in der Passivprogrammierung. Vor jedem GET,K muß die Länge neu gesetzt werden: $0 < l \leq 32K$ LEB enthält nach Init 32 KB, sonst immer den Wert des letzten GET-Befehls. Die Länge kann auch hexadezimal in der Form X'l' (immer 4 Zeichen) angegeben werden. Wird der Satz beim GET,K gekürzt, weil $l <$ Satzlänge, wird die Meldung GE06 ausgegeben.
Terminator	Vgl. Seite 228
Fs2	Fortsetzungsschlüssel Fs2 (vgl. Seite 228).

3.6.14 %PASSIV-ON/OFF (Passivverarbeitung)

```
nnnnnnnn%PASSIV_{ON/OFF}Terminator[Fs2]
```

nnnnnnn	Kommandonummer
PASSIV	Operation für Passivverarbeitung.
ON/OFF	Einschalten/Ausschalten
Terminator	vgl. Seite 228
Fs2	Fortsetzungsschlüssel Fs2 (vgl. Seite 228).

3.6.15 %PEIN / %PAUS (Protokollierung ein-/ausschalten)

```
nnnnnnnn{%PEIN/%PAUS}Terminator[Fs2]
```

nnnnnnn	Kommandonummer
%PEIN	Operation für Protokollierung einschalten.
%PAUS	Operation für Protokollierung ausschalten.
Terminator	vgl. Seite 228
Fs2	Fortsetzungsschlüssel Fs2 (vgl. Seite 228).

Die Kurzkommandos, %PEIN und %PAUS, dienen dem Test von Kurzkommandos.

Die Eingabe %PEIN schaltet die Protokollierung der Kurzkommandos ein. Die Kurzkommandos werden mit dem Fortsetzungsschlüssel vor der Übergabe an das eigentliche CIS auf SYSOUT ausgegeben. Die Protokollierung bleibt so lange eingeschaltet, bis sie mit %PAUS wieder ausgeschaltet wird oder CIS mit HALT beendet wird.

Beide Kommandos können entweder direkt von der Datenstation eingegeben werden oder in der Kurzkommandodatei abgespeichert sein.

Eingabe vom Terminal:

%PEIN	Wird ohne weitere Angaben vor dem Start einer Kurzkommandokette eingegeben.
%PAUS	Soll die Protokollfunktion wieder ausgeschaltet werden, z.B. nach Ablauf der Kommandokette, wird das Kommando ohne weitere Angaben eingegeben.

3.6.16 %RANDOM (Zufallszahl)

```
nnnnnnnn%RANDOM_(min,max)Terminator[Fs2]
```

nnnnnnnn	Kommandonummer
%RANDOM	Operation für Zufallszahl erzeugen.
min	Untere Grenze der Zufallszahl.
max	Obere Grenze der Zufallszahl.
Terminator	vgl. Seite 228
Fs2	Fortsetzungsschlüssel Fs2 (vgl. Seite 228).

Die erzeugte Zufallszahl wird in den Puffer der unbenannten Variablen abgelegt und kann damit "@" ersetzen.

Beispiel: `;%RANDOM (1,2);% @` gibt z.B. +1 am Bildschirm aus.

oder:

`;%RANDOM (0,99);AEN,1,F,Zahl=@` schreibt in das Feld "Zahl" des aktuellen Satzes eine Zahl zwischen 0 und 99.

3.6.17 %RETURN (Rückkehr aus einem Unterprogramm)

```
nnnnnnnn%RETURN Terminator[Fs2]
```

nnnnnnnn	Kommandonummer
%RETURN	Operation für das Ende des Unterprogrammes.
Terminator	vgl. Seite 228
Fs2	Fortsetzungsschlüssel: Ist Fs2 nicht angegeben (normales Unterprogrammende), so wird bei der Fortsetzungsadresse des vorausgegangenen %GOSUB-Kommandos fortgesetzt (es wird also nur das Unterprogramm beendet, nicht das gesamte Kurzkommandoprogramm). Ist Fs2 angegeben, so wird bei diesem Schlüssel fortgesetzt. Eine Folgeadresse, die beim Unterprogrammaufruf (%GOSUB-Kommando - vgl. Seite 245) programmiert wurde, oder Sprungadressen, die durch die Verschachtelung von Unterprogrammen noch aktuell sind, werden bei Verwendung der unbedingten Sprungadresse ungültig.

Das Kurzkommando %RETURN bildet den Abschluß eines Unterprogramms, das mit %GOSUB (vgl. Seite 245) angesprungen wurde. Ein %RETURN mit Angabe des Fortsetzungsschlüssels verläßt alle Unterprogrammstufen.

Beispiel:

```
00000100 %GOSUB1000#N              /* 1. UNTERPROGRAMM */
00000200 %GOSUB2000#E              /* 2. UNTERPROGRAMM */
00001000 %*Unterprogramm 1#N
00001100SU BEZIRK=&,DB.KUNDEN#N
    .
    .
    .
00001600%GOSUB3000#N
00001700%RETURN#                  /* RUECKSPRUNG AUS UP 1 */
00002000%*Unterprogramm 2#N
    .
    .
    .
00002800%RETURN#                  /* RUECKSPRUNG */
00003000%*Unterprogramm 3#N
    .
    .
    .
00003500%RETURN#                  /* RUECKSPRUNG */
```

3.6.18 %SET (Multiplitätenbehandlung)

```
nnnnnnnn%SET[:AM=xxx][:FM=xxx][:Fs1]Terminator[Fs2]
```

nnnnnnnn	Kommandonummer
%SET	Operation für Setzen der Multiplität.
:	Syntaxtrenner
AM=xxx	Operationsergänzung für Abschnittsmultiplität: Für xxx kann eine max. 3-stellige Zahl mit Vorzeichen angegeben werden: xxx Die Multiplität wird absolut auf den Wert xxx gesetzt. +xxx Die Multiplität wird relativ um xxx erhöht. -xxx Die Multiplität wird relativ um xxx vermindert.
FM=xxx	Operationsergänzung für Feldmultiplität: Für xxx kann eine max. 3-stellige Zahl mit Vorzeichen angegeben werden: xxx Die Multiplität wird absolut auf den Wert xxx gesetzt. +xxx Die Multiplität wird relativ um xxx erhöht. -xxx Die Multiplität wird relativ um xxx vermindert.
Fs1	Fortsetzungsschlüssel (vgl. Seite 227)
Terminator	vgl. Seite 228
Fs2	Fortsetzungsschlüssel (vgl. Seite 228)

Wird innerhalb einer Kurzkommandokette in einem oder mehreren Kommandos die im Aktivbetrieb sonst unzulässige Angabe, AM=$ oder FM=$, angegeben, so wird für dieses Kurzkommando im CISKURZ-Kommandopuffer je ein Zähler für die Abschnitts- und Feldmultiplität mitgeführt. Diese Zähler werden beim ersten Ablauf des Kommandos angelegt und auf den Wert 1 gesetzt, falls sie nicht vorher explizit mit einem absoluten Wert besetzt wurden. Der Zählerstand kann mit dem %SET-Kommando manipuliert werden. Er wird im weiteren Ablauf in das betreffende Kommando eingetragen. Zählerstände werden in Abhängigkeit eines nicht angegebenen bzw. angegebenen Vorzeichens gesetzt bzw. errechnet. Ist kein Vorzeichen angegeben, wird der Zählerstand absolut auf den angegebenen Wert gesetzt, ist eine Vorzeichen angegeben, so wird der angegebene Wert zum aktuellen Zählerstand addiert bzw. davon subtrahiert.

Werden beim %SET-Kommando beide Fortsetzungsschlüssel angegeben, so werden die Zähler im unter Fs1 angegebenen Kommando gesetzt und zum unter Fs2 angegebenen Kommando verzweigt. Ist nur ein Fortsetzungsschlüssel (Fs2) angegeben, so werden die Zähler im unter Fs2 angegebenen Kommando gesetzt und anschließend dorthin verzweigt (vgl. Beispiel 2 auf Seite 270).

3.6.19 %SHOW-BUFFER/LENGTH (EB/LEB ausgeben)

```
nnnnnnnn%SHOW_{BUFFER[+d]fl/LENGTH}
```

nnnnnnnn	Kommandonummer
%SHOW	Operation für internen Satzbereich bzw. Länge des internen Satzbereichs ausgeben.
BUFFER	Operationsergänzung für interner Satzbereich.
d	Distanz im internen Satzbereich, dezimal oder hexadezimal: X'd'. Bei fehlender Angabe wird 0 angenommen.
f	Feldformat: C: Zeichen (Character) X: Hexadezimal
l	Länge des Feldes in Bytes.
LENGTH	Operationsergänzung für die Ausgabe der Größe des internen Satzbereichs.

3.6.20 %STOP (Unterbrechung)

```
nnnnnnnn%STOP[:[Fs1][(As[:As]...)][:NO-END]]Term.[Fs2]
```

nnnnnnnn	Kommandonummer	
%STOP	Operation für Unterbrechung.	
:	Syntaxtrenner. Er muß angegeben werden, wenn Fs1 und/oder Alternativschlüssel angegeben werden.	
Fs1	Fortsetzungsschlüssel (vgl. Seite 227). Fs1 legt die Fortsetzung bei Eingabe von BREAK fest.	
As	Alternativschlüssel:	Er legt fest, bei welchen Fortsetzungsschlüsseln durch explizite Eingabe der Kommandonummer fortgesetzt werden kann. Die Schlüssel müssen von einander durch den Doppelpunkt getrennt sein, sie müssen nicht aufsteigend angegeben werden.
NO-END	Die Eingabe E für Ende wird nicht akzeptiert, d.h. die Befehlsprozedur kann nicht verlassen werden.	
Term	Terminator vgl. Seite 228	
Fs2	Fortsetzungsschlüssel	(vgl. Seite 228): Er legt fest, bei welchem Fortsetzungsschlüssel, abweichend von den Alternativschlüsseln, bei der Eingabe von NEXT fortgesetzt wird (z. Bsp. Fehlerausgang).

%STOP unterbricht Kurzkommandoketten. Nach der Unterbrechung kann der Anwender die weitere Abarbeitung oder Beendigung der Kommandokette selbst steuern. Nach dem %STOP-Kommando wird der RDATA- Stern am Bildschirm ausgegeben. Die Fortsetzungsmöglichkeiten können und sollten also in der Kommandokette vor dem %STOP-Kommando mit Kommentaren erläutert werden.

Nach dem %STOP gibt es folgende Eingabemöglichkeiten:

END	Die Kette wird abgebrochen.
NEXT	Die Kette wird bei dem im Kommando angegebenen Fortsetzungsschlüssel Fs2 fortgesetzt.
nnnnnnnn	Die Bearbeitung wird bei der explizit eingegeben Kommandonummer (max. 8-stellig) fortgesetzt. Ist eine Liste von Alternativschlüsseln angegeben, so wird von CIS geprüft, ob die vom Anwender eingegebene Kommandonummer als Fortsetzungsalternative erlaubt ist. Ist dies nicht der Fall, so wird nach Fs2 verzweigt (Fehlerausgang).
BREAK	Die Bearbeitung setzt bei dem im Kommando angegebenen Fortsetzungsschlüssel Fs1 fort. Falls Fs1 nicht angegeben ist, wird bei Fs2 fortgesetzt.
andere Eingaben	Z.B. alphanumerische, bewirken ein Fortsetzen bei Fs2.

Beispiel:

```
00000010% BEI EINGABE VON:          AUFRUF VON:     #N
00000015%    - END                  ENDE DER KETTE  #N
00000020%    - 100                  SU KUNDE=       #N
00000025%    - 200                  SU AUFTRAG=     #N
00000030%    - 300                  SU LIEFERANT=   #N
00000050%STOP:(200:100:300)#400
00000100SU KUNDE=&##
00000200SU AUFTRAG=&##
00000300SU LIEFERANT=&##
00000400% UNERLAUBTE EINGABE; ES GIBT NUR #N
00000410% FOLGENDE MOEGLICHKEITEN:        #10
```

Kommando 10 bis Kommando 30 teilen dem Anwender seine Eingabemöglichkeiten über unbedingte Kommentare mit.

Nach Ablauf von Kommando 50 gibt es folgende Eingabemöglichkeiten:

END	Beendigung der Kurzkommando-Kette.
100	Fortsetzung bei Kommando 100.
200	Fortsetzung bei Kommando 200.
300	Fortsetzung bei Kommando 300.
andere	Fortsetzung bei Kommando 400. In diesem Fall wird nach dem Fehlerkommentar wieder das Auswahlmenü (Kommentare 10 - 30) ausgegeben und die Unterbrechung nochmals ausgeführt. Auch wenn eine Alpha-Zeichenfolge (z.B. bei BREAK oder HALT) eingegeben wird, wird bei 400 fortgesetzt.

Im obigen Beispiel wurde keine BREAK-Alternative festgelegt, weil dem Anwender nur die angegebenen Eingabemöglichkeiten erlaubt werden sollten. Soll für BREAK auch eine effektive Fortsetzungsalternative festgelegt werden (also bei Eingabe von **B**REAK nicht Fortsetzung über Fehler-Ausgang), so kann Kommando 50 wie folgt aussehen:

```
00000050%STOP:5000(200:100:300)#400
```

Bei Eingabe von **B**REAK würde jetzt ein Sprung nach Kommando 5000 durchgeführt.

3.6.21 %STORE (Speichern von Konstanten)

```
nnnnnnnn%STOREx[xxx][=Konstante[,R]]Terminator[Fs2]
```

nnnnnnnn	Kommandonummer
%STORE	Operation für Speichern.
x[xxx]	Speicherkennung: Max. 4-stellige alphanumerische Zeichenfolge, die mit keiner Ziffer beginnt.
Konstante	Angabe einer festen Zeichenfolge.
R	R muß angegeben sein, wenn die Konstante numerisch ist.
Terminator	vgl. Seite 228
Fs2	Fortsetzungsschlüssel (vgl. Seite 228)

Das Kommando kann Konstanten für Vergleiche ablegen. Dadurch ist es möglich, Datenbankwerte mit u.a. extern eingegebenen Werten zu vergleichen (vgl. dazu auch %IF-Kommando auf Seite 246 und das dort angegebene Beispiel).

Numerische Werte werden durch Format R gekennzeichnet. Dabei können die Vergleichslängen des durch den %STORE abgelegten Wertes und des beim %IF verwendeten GET-F Wertes unterschiedlich sein. Die Ausrichtung erfolgt nach dem angegebenen bzw. definierten Dezimalpunkt.

Soll mit Dezimalstellen gearbeitet werden, so ist die numerische Konstante mit dem Dezimalpunkt an der entsprechenden Stelle anzugeben. Wenn beim %STORE in der Rechen-Konstante kein Dezimalpunkt angegeben ist, wird der Dezimalpunkt nach der letzten Stelle angenommen. Ein negatives Vorzeichen kann angegeben werden.

Beispiel:

```
00000010%STOREA=123.45,R#N      Numerischer Wert mit zwei Dezimalstellen.
00000020%STOREB=-123.45,R#N     Negativer numerischer Wert mit zwei Dezimalstellen.
```

Fehlt die Angabe einer Konstanten im %STORE-Kommando, so wird der letzte gültige GET-F Wert (LIFO-Prinzip) unter der Speicherkennung xxxx abgespeichert.

3.6.22 %SYS (BS2000 Systemkommando)

```
nnnnnnnn%SYS[_/BS2000-Kommando]Terminator[Fs2]
```

nnnnnnnn	Kommandonummer
/%SYS	Operation für BS2000 Kommando.
Terminator	vgl. Seite 228
Fs2	Fortsetzungsschlüssel (vgl. Seite 228)

Das auszuführende Kommando kann maximal 200 Zeichen lang sein. AID- und IDA-Kommandos werden zurückgewiesen.

Fehlt ein BS2000-Kommando so wird in TIAM mit BKPT in die Kommandoebene BS2000 verzweigt.

Bei Syntaxfehlern und im Falle der Zurückweisung wird KU46 gesetzt und eine evtl. Kommandoprozedur unterbrochen. Fehler bei Kommandoausführung ist KU47 und ist in der Prozedur mit %CM abfragbar.

3.7 Fallbeispiele zur Anwendung von Kurzkommandos

Beispiel 1:

Ausgangsposition

Eine Firma führt unter anderem folgende vier Dateien:

- Vertreterdatei
- Kundendatei
- Auftragsdatei
- Lagerdatei

In diesem beschriebenen Fall sind diese Dateien nur logisch vier Dateien, physikalisch existieren nur zwei Dateien:

HD.KUKOTEST (CIS-Hauptdatei)
VD.KUKOTEST (CIS-Verweisdatei)

Die zugehörige Satzbeschreibung ist DB.DBNAME, die im folgenden auszugsweise abgebildet ist.

Die vier Dateien entsprechen folgenden Abschnitten der Datenbeschreibung:

Logische Datei	Entsprechender Abschnitt in der Satzbeschreibung
Vertreterdatei	VERT
Kundendatei	KUND
Auftragsdatei	AUFT
Lagerdatei	LGER

Satzbeschreibung:

SEGM	KOPFBESCHREIBUNG												
	NAME	PARAMETER IM A-SEGMENT											
AA	DBNAME	LOGADR=J,MAXDES=40,DSS=J,DSA=J,DSU=J, HD=HD.KUKOTEST,VD=VD.KUKOTEST											
FELDBESCHREIBUNG													
SEGM	QUELL-DEFINITION			LOG	DARST	TV	ZIEL-DEFINITION			SONDER FUNKTION		WA	SEGM	FELDBEZEICHNUNG TRANSFORMATIONS-ERGAENZUNG
	AA	ADR	LNG				AA	ADR	LNG	SF1	SF2			
3	4	8	12	15		18	21	25	29	32	36	0	5	46 (41-44 FREI)
DD	KOPF	9	3	OP	BT								EE	EDVNR
DD	KOPF	9	1		T					PSE			EE	PSEUDO
DDDD	VERT VERT VERT .	9 24 29	15 5 1		T T T					001 002			EEEE	VERTRETERNAME VNUMMER VERTRETERART .
DDDD	KUND KUND KUND KUND .	9 14 20 60	5 6 40 50		T T T T					003 004 005			EEEE	VERTRETERNUMMER KUNDENNUMMER NAME KUNDENANSCHRIFT .
DDDD	AUFT AUFT AUFT AUFT AUFT .	9 15 21 51 78	6 6 30 27 3		T T T T T					006 007			EEEE	AUFTRAGSNUMMER KDNR BESTELLDATEN LIEFERDATEN LAGERVERWEIS .
DDDD	LGER LGER LGER .	9 12 32	3 20 50		T T T					008			EEEE	LAGERNUMMER LAGERORT BESTANDSDATEN .
D	.												E	.

E N D E

Die logische Trennung der vier Dateien ist wie folgt realisiert:

Die Hauptdatei besteht aus einer bestimmten Gesamtanzahl von Sätzen, die vier verschiedene Strukturen haben können. Beim Aufbau der Datenbank wurde pro Abschnitt (VERT,KUND,AUFT,LGER) ein eigener Satz erzeugt. Alle Sätze enthalten als gemeinsamen Abschnitt nur jeweils den Kopfabschnitt, der den Ordnungsbegriff (EDVNR) enthalten muß. Es sind also nur folgende vier Strukturen jeweils eines Satzes, dargestellt in Abschnittsarten, möglich:

```
KOPF - Abschnitt
     +
VERT - Abschnitt
```

```
KOPF - Abschnitt
     +
KUND - Abschnitt
```

```
KOPF - Abschnitt
     +
AUFT - Abschnitt
```

```
KOPF - Abschnitt
     +
LGER - Abschnitt
```

Das Feld PSEUDO im Kopfabschnitt ist für die Struktur eines Satzes belanglos, es dient als Pseudofeld für bestimmte Anwendungen (vgl. auch D-Segment Satzbeschreibung und Zielpunktlistenbehandlung).

Die einzelnen Sätze der Hauptdatei sind über Felder mit gleichem Inhalt miteinander verknüpfbar. Inhaltsgleiche Felder sind:

Abschnittsname	Feldname	Feldname	Abschnittsname
VERT	VNUMMER	VERTRETERNUMMER	KUND
KUND	KUNDENNUMMER	KDNR	AUFT
AUFT	LAGERVERWEIS	LAGERNUMMER	LGER

Diese Felder sind auch in der abgebildeten Satzbeschreibung gekennzeichnet.

Erläuterungen:

Ein bestimmter Vertreter hat eine bestimmte Vertreternummer. Dieser Vertreter betreut mehrere Kunden. Die betreffenden Kundensätze enthalten alle dieselbe Vertreternummer. Jeder Kunde hat eine bestimmte Kundennummer. Alle Aufträge zu einem Kunden enthalten dieselbe Kundennummer (KDNR). Es gibt eine Anzahl verschiedener Auslieferungslager. Jedes Lager hat eine eigene Lagernummer. Dieselbe Lagernummer ist in den entsprechenden Auftragssätzen unter LAGERVERWEIS gespeichert.

Es gibt zwei Arten von Vertretern:

Art	Bedeutung	Schlüsselung VERTRETERART
Firmenkundenvertreter	betreut Firmenkunden (Großkunden)	F
Privatkundenvertrer	betreut Privatkunden	P

Problemstellung:

Ein bestimmtes Lager ist abgebrannt (Lagerort bekannt). Durch diesen Vorfall kommt es zu Lieferschwierigkeiten. Alle Kunden, bei denen die Lieferung von diesem Lager aus erfolgen sollte und von Firmenkundenvertretern betreut werden, sollen gesondert benachrichtigt werden, sie müssen also aus dem Datenbestand herausgesucht werden.

Problemlösung mittels Kurzkommandokette:

Die nachfolgende Kurzkommandokette wird mit internen Kommentaren erläutert. Diese stehen jeweils vor dem betreffenden Kurzkommando.

```
00000010%*   PROPHYLAKTISCH PSEUDOZIELPUNKTLISTE LOESCHEN    #N
00000020FR,V,PSEUDO=1,DB.DBNAME#N
00000030%*   ABFRAGE DER CIS-QUITTUNG ZP13, DAMIT DIE KETTE  #N
00000040%*   NICHT ABGEBROCHEN WIRD. DIESE QUITTUNG WIRD AUS-#N
00000050%*   GEGEBEN, WENN KEINE ZPL GESICHERT IST.          #N
00000060%CM=ZP13:N#N
00000070%*   PROPHYLAKTISCH PSEUDOZIELPUNKTLISTE LOESCHEN.   #N
00000080FR,V,PSEUDO=2#N
00000090%*   ABFRAGE DER CIS-QUITTUNG ZP13, DAMIT DIE KETTE  #N
00000100%*   NICHT ABGEBROCHEN WIRD. DIE QUITTUNG WIRD AUS-  #N
00000110%*   GEGEBEN, WENN KEINE ZPL GESICHERT IST.          #N
00000120%CM=ZP13:N#N
00000130%*   LAGERORT SUCHEN.                                #N
00000140SU LAGERORT=&#N
00000150%*   ABFRAGE DER ZIELINFORMATIONEN. BEI NULL WIRD    #N
00000160%*   DIE KETTE UEBER KOMMANDO 10000 BEENDET.         #N
00000170%AZI=0:10000#N
00000180%*   DIE LAGERNUMMER WIRD FUER DAS NACHFOLGENDE      #N
00000190%*   SUCHE-KOMMANDO IM CISKURZ-PUFFER GESPEICHERT.   #N
00000200GET,F,LAGERNUMMER#N
00000210%*   ALLE AUFTRAEGE MIT DIESER LAGERNUMMER           #N
00000215%*   (LAGERVERWEIS) SUCHEN.                          #N
00000230SU LAGERVERWEIS=@#N
00000240%*   WENN KEINE AUFTRAEGE (ZIELINFORMATIONEN) VOR-   #N
00000250%*   HANDEN SIND, KETTE UEBER KDO. 10000 BEENDEN.    #N
00000260%AZI=0:10000#N
00000270%*   ZIELPUNKTLISTE FUER WEITERE BEARBEITUNG SICHERN #N
00000280SI,E,EDV=1#N
00000290%*   ERSTE/NAECHSTE KUNDENNUMMER FUER WEITERE        #N
00000300%*   BEARBEITUNG IM CISKURZ-PUFFER SPEICHERN.        #N
00000310GET,F,KDNR#N
00000320%*   ERSTE/NAECHSTE KUNDENNUMMER SUCHEN.             #N
00000330SU KUNDENNUMMER=@#N
00000340%*   SUCHERGEBNIS MIT DEN BISHERIGEN LOKALISIERTEN   #N
00000350%*   KUNDEN DURCH 'ODER' VEREINIGEN (ERWEITERN).     #N
00000355ODER PSEUDO=1#N
```

```
00000360%*   NEUE ZPL IN DER VERWEISDATEI SICHERN.           #N
00000370SI,V,PSEUDO=1#N
00000380%*   DAMIT DIE KUKOKETTE NICHT ABGEBROCHEN WERDEN    #N
00000390%*   KANN, WIRD DIE CIS-QUITTUNG ZP10 ABGEFRAGT.     #N
00000400%*   DIESE QUITTUNG WIRD AUSGEGEBEN, WENN KEINE      #N
00000405%*   ZIELPUNKTLISTE ZUM SICHERN VORHANDEN IST        #N
00000410%CM=ZP10:N#N
00000420%*   DIE TEMPORAER GESICHERTE ZIELPUNKTLISTE DER     #N
00000425%*   AUFTRAEGE MIT DEM ZUTREFFENDEN LAGERORT WIRD    #N
00000430%*   WIEDER AKTIVIERT                                #N
00000440AK,E,EDV=1#N
00000450%*   AUF DEN NAECHSTEN AUFTRAG BLAETTERN.            #N
00000460BL,1,V#N
00000470%*   SIND KEINE AUFTRAEGE MEHR DA: NAECHSTES KOMMANDO#N
00000480%*   SONST NEUER SCHLEIFENDURCHLAUF (KOMMANDO 270).  #N
00000490%CM=IM01:N#270
00000500%*   ES SIND JETZT ALLE KUNDEN GEFUNDEN, DIE AUF-    #N
00000510%*   TRAEGE MIT DEM BETREFFENDEN LAGERORT HABEN.     #N
00000520%*   ES WIRD JETZT NACH DER VERTRETERART=F (FUER     #N
00000530%*   FIRMENVERTRETER) GESUCHT, UM DANN ZU DEN        #N
00000540%*   VERTRETERN DIE JEWEILIGEN KUNDEN ZU SUCHEN      #N
00000550SU VERTRETERART=F#N
00000560%*   SIND KEINE ZIELINFORMATIONEN VORHANDEN: UEBER   #N
00000570%*   KOMMANDO 10000 BEENDEN.                         #N
00000580%AZI=0:10000#N
00000590%*   ZPL FUER SCHRITTWEISE ABARBEITUNG SICHERN       #N
00000600SI,E,EDV=2#N
00000610%*   ERSTE/NAECHSTE VERTRETERNUMMER FUER FOLGENDES   #N
00000620%*   SUCHEN IM CISKURZ-PUFFER SPEICHERN.             #N
00000630GET,F,VNUMMER#N
00000640%*   KUNDEN ZUR JEWEILIGEN VERTRETERNUMMER SUCHEN.   #N
00000645SU VERTRETERNUMMER=@#N
00000650%*   GLEICHZEITIG MUSS DER KUNDE AUFTRAEGE MIT DEM   #N
00000660%*   BETREFFENDEN LAGERORT LAUFEN HABEN (SIND UNTER  #N
00000665%*   PSEUDO=1 IN DER VERWEISDATEI GESICHERT).        #N
00000667UND PSEUDO=1#N
00000669%*   DIE ZPL WIRD MIT DEN BEREITS GEFUNDENEN         #N
00000670%*   KUNDEN ERWEITERT, FUER DIE DIE VORHERIGEN       #N
00000675%*   BEDINGUNGEN GELTEN                              #N
00000680%*   (WURDEN SCHRITTWEISE BEI JEDEM SCHLEIFENDURCH-  #N
00000685%*   LAUF IN DER VD UNTER PSEUDO=2 GESICHERT).       #N
00000690ODER PSEUDO=2#N
00000700%*   DIE NEUE ZIELPUNKTLISTE (KUNDEN FUER DIE ALLE   #N
00000710%*   BEDINGUNGEN DER ABFRAGE ERFUELLT SIND) WIRD IN  #N
00000720%*   DER VERWEISDATEI UNTER PSEUDO=2 GESICHERT.      #N
00000730SI,V,PSEUDO=2#N
00000740%*   DAMIT DIE KUKOKETTE NICHT ABGEBROCHEN WERDEN    #N
00000750%*   KANN, WIRD DIE CIS-QUITTUNG ZP10 ABGEFRAGT;     #N
00000755%*   DIESE WIRD AUSGEGEBEN, WENN KEINE ZPL           #N
00000760%*   ZUM SICHERN VORHANDEN IST                       #N
00000770%CM=ZP10:N#N
```

```
00000780%*   DIE GESICHERTE ZPL DER FIRMENVERTRETER          #N
00000790%*   WIRD WIEDER AKTIVIERT.                          #N
00000800AK,E,EDV=2#N
00000810%*   AUF DEN NAECHSTEN FIRMENVERTRETER BLAETTERN.    #N
00000820BL,1,V#N
00000830%*   SIND KEINE VERTRETER MEHR DA:                   #N
00000840%*         - VERZWEIGEN NACH KOMMANDO 15000,         #N
00000850%*           SONST:                                  #N
00000860%*         - NEUER SCHLEIFENDURCHLAUF (KOMMANDO 590) #N
00000870%CM=IM01:15000#590
00000880%*   DIE NAECHSTEN KOMMENTARZEILEN (UNBEDINGTE       #N
00000885%*   KOMMENTARE) WERDEN AUSGEGEBEN, WENN KEIN KUNDE, #N
00000890%*   FUER DEN DIE GEFORDERTEN BEDINGUNGEN ZUTREFFEN, #N
00000900%*   VORHANDEN IST.                                  #N
00010000% KEIN KUNDE VORHANDEN #N
00010010% ----- E N D E ------ #E
00015000%*   ALLE KUNDEN, FUER DIE DIE GEFORDERTEN           #N
00015010%*   BEDINGUNGEN ZUTREFFEN, WERDEN GESUCHT           #N
00015015%*   SIE WURDEN VORHER SCHRITTWEISE                  #N
00015020%*   UNTER PSEUDO=2 IN DER VERWEISDATEI GESICHERT).  #N
00015030SU PSEUDO=2#N
00015040%*   SIND KEINE ZIELINFORMATIONEN VORHANDEN, SO WIRD #N
00015050%*   DIE KUKOKETTE UEBER KOMMANDO 10000 BEENDET      #N
00015060%AZI=0:10000#N
00015070%*   LISTE ERSTELLEN.                                #N
00015080D,$,T,NAME KUNDENANSCHRIFT#N
00015090% ----- E N D E ------ #E
```

Beispiel 2: Verwendung des %SET-Kommandos

Es liegt eine Lieferantendatei mit MV-Format vor. Der feste Abschnitt jedes Satzes (Abschnittsname: KOPF) enthält dabei lieferantenspezifische Daten wie z.B. Name und Anschrift des Lieferanten. Die Wiederholabschnitte des Satzes (Abschnittsname: PROD) enthalten die einzelnen Produkte dieses Lieferanten (siehe unten abgebildeten Auszug der Satzbeschreibung).

Satzbeschreibung mit MV-Format:

SEGM	KOPFBESCHREIBUNG												
	NAME	PARAMETER IM A-SEGMENT											
A	LIEFER	LOGADR=J,DSS=J,DSA=J,DSU=J,HD=HD.LIEFER											
	FELDBESCHREIBUNG												
SEGM	QUELL-DEFINITION			LOG	DARST	TV	ZIEL-DEFINITION			SONDER FUNKTION		WA	SEGM	FELDBEZEICHNUNG TRANSFORMATIONS-ERGAENZUNG
	AA	ADR	LNG				AA	ADR	LNG	SF1	SF2			
3	4	8	12	15		18	21	25	29	32	36	0	5	46 (41-44 FREI)
DDDDD	KOPF KOPF KOPF . . .	9 12 52	3 40 50	O 	B T T								EEEEE	EDVNR LIEFERANT ANSCHRIFT . . .
DDDDDD	PROD PROD PROD PROD . . .	9 24 32 38	15 8 6 3	 2 	T R R T							W W W W	EEEEEE	PRODBEZ PRODNUMMER PRODPREIS PREISEINHEIT . . .

E N D E

Es soll die Möglichkeit geschaffen werden, auch innerhalb der Wiederholabschnitte z.B. nach Preis oder Produktnummer zu sortieren.

Die nachfolgende Kurzkommandokette durchsucht alle vorhandenen Sätze nach Produktnummern. In diesem Beispiel wird nach ungleich Null gesucht, weil Null keine gültige Produktnummer ist. Jeder Wiederholabschnitt wird mittels WRITE-Kommando zusammen mit dem festen Abschnitt in eine ISAM-Datei (HD.LIENEU) geschrieben. Dieser ISAM-Datei wird eine neue Satzbeschreibung zugeordnet. Diese Satzbeschreibung beschreibt jeden ISAM-Satz als Satz mit V-Format, so daß u.a. nach jedem beliebigen, beschriebenen Feld sortiert werden kann. Vor Ablauf der Kette muß die neue Datei mit dem Systemkommando:

```
/SET-FILE-LINK FILE-NAME=HD.LIENEU,LINK-NAME=ISISOUT
```

zugewiesen werden.

Kurzkommandokette:

```
00000010%*   SCHLEIFENANFANG                                    #N
00000020%*   ES WERDEN ALLE PRODUKTNUMMERN MIT DER ERSTEN/   #N
00000030%*   NAECHSTEN ABSCHNITTSMULTIPLITAET                  #N
00000035%*   IN DER HD.LIEFER (MV-FORMAT) GESUCHT.            #N
00000100SU PRODUKTNUMMER(AM=$)#0,DB.LIEFER#N
00000110%*   WERDEN KEINE PRODUKTNUMMERN MEHR GEFUNDEN, WIRD #N
00000120%*   DIE SCHLEIFE UEBER KOMMANDO 600 BEENDET.         #N
00000200%AZI=0:600#N
00000210%*   SIND ZIELINFORMATIONEN VORHANDEN, WIRD MITTELS  #N
00000220%*   WRITE-KOMMANDO PRO WIEDERHOLABSCHNITT            #N
00000225%*   DER GESUCHTEN ABSCHNITTSMULTIPLITAET EIN SATZ   #N
00000230%*   IN DIE ISAM-DATEI HD.LIENEU GESCHRIEBEN. DER    #N
00000240%*   SATZ BEINHALTET JEWEILS DEN WIEDERHOLABSCHNITT  #N
00000250%*   DIESER MULTIPLITAET UND DEN ZUGEHOERIGEN FESTEN #N
00000260%*   ABSCHNITT DES ALTEN MV-SATZES.                   #N
00000300WR,$,S,DATEI=HD.LIENEU,LIEFERANT ANSCHRIFT PRODBEZ
(AM=$) PRODNUMMER(AM=$) PRODPREIS(AM=$) PREISEINHEIT(AM=$)   #N
00000310%*   DEN IM CISKURZ-PUFFER MITGEFUEHRTEN ZAEHLERSTAND#N
00000320%*   DES WRITE-KOMMANDOS UM EINS ERHOEHEN.#N
00000400%SET:AM=+1:300#N
00000410%*   DEN IM CISKURZ-PUFFER MITGEFUEHRTEN ZAEHLERSTAND#N
00000415%*   DES SUCHEKOMMANDOS UM EINS ERHOEHEN UND DORTHIN #N
00000420%*   VERZWEIGEN.                                      #N
00000500%SET:AM=+1#100
00000510%*   WERDEN KEINE PRODUKTNUMMERN MEHR GEFUNDEN, SO   #N
00000520%*   WIRD NACHFOLGENDER KOMMENTAR AUSGEGEBEN.         #N
00000600% E N D E   D E R   U M S E T Z U N G #E
```

Wird ein Wiederholfeld in einem Wiederholabschnitt gesucht, und wird der Wiederholabschnitt nicht gefunden, so kommt es zur Fehlermeldung AF18. Ist der Abschnitt noch vorhanden, wird aber kein Wiederholfeld mehr gefunden, so kommt es zur Fehlermeldung AF06. Beide Fehlermeldungen können, wenn die Programmlogik es erfordert, mit dem %CM-Kommando abgefragt werden.

Werden in einem Kommando, dessen Multiplitätenzähler während des Ablaufs mit %SET manipuliert werden sollen, externe Variablen erst während des Ablaufs ersetzt, so wird eine Fehlermeldung ausgegeben.

Im Anschluß wird die neue für die soeben beschriebene ISAM-Datei gültige Satzbeschreibung auszugsweise dargestellt. Der 8-stellige numerische Ordnungsbegriff (EDVNR) entspricht dem ISAM-Schlüssel der Datei.

Neue Satzbeschreibung mit V-Format:

SEGM	KOPFBESCHREIBUNG												
	NAME	PARAMETER IM A-SEGMENT											
A	LIENEU	LOGADR=J,DSS=J,DSA=J,DSU=J,HD=HD.LIENEU											
FELDBESCHREIBUNG													
SEGM	QUELL-DEFINITION			LOG	DARST	TV	ZIEL-DEFINITION			SONDER FUNKTION		WA	SEGM	FELDBEZEICHNUNG TRANSFORMATIONS-ERGAENZUNG
	AA	ADR	LNG				AA	ADR	LNG	SF1	SF2			
3	4	8	12	15		18	21	25	29	32	36	0	5	46 (41-44 FREI)
DD		5	8	O	R								E	EDVNR
DD		13	40		T								E	LIEFERANT
DD		53	50		T								E	ANSCHRIFT
DD		103	15		T								E	PRODBEZ
DD		118	8		R								E	PRODNUMMER
DD		126	6		2R								E	PRODPREIS
DD		132	3		T								E	PREISEINHEIT
							E N D E							

4 Systemkonstanten

Systemkonstanten sind in CIS intern definierte Felder, die gelesen aber nicht geändert werden können.

Sie lassen sich insbesonders in Kurzkommandoketten auswerten.

4.1 Übersicht über die Systemkonstanten

Name	Bedeutung	Länge	Feldbedeutung
%SYSDAT	Datum - Format JJMMTT	6	A
%SYSLST	Name der SYSLST-Datei der aktuellen Transaktion. Wurde der Systemkonstanten keine Datei zugewiesen, so ist deren Inhalt "SYSLST".	26	T
%SYSTID	Transaktions-Id	8	T
%SYSTIM	Uhrzeit - Format: hhmmss	6	A
%SYSTSN	TSN	4	A
%SYSUID	User-Id	8	T
%SYSTAI	Vollständige Transaktions-Id. (Rechner/Anwendung/Transaktion)	24	T

Bemerkung: Ist die TSN alphanumerisch, darf natürlich nicht mit ihr gerechnet werden.

4.2 Beispiele zur Verwendung von Systemkonstanten

Allgemeine Beispiele:

```
Z,1,Z,%SYSLST %SYSDAT %SYSTID

Z,1,T,%SYSTIM %SYSTSN %SYSUID

RE %X1=%SYSDAT

RE %X2=%SYSTIM

RE %X3=%SYSTSN
```

nicht erlaubt:

```
RE %SYSTIM=103000
```

Beispiel einer Kurzkommandokette unter UTM

Die nachfolgende Kurzkommandokette wird mit internen Kommentaren erläutert. Diese stehen jeweils vor dem betreffenden Kurzkommando.

```
00010000%*     Drucke-Kommando                  #N
00011000DR ...#N
00020000%*     Dateiname der SYSLST-Datei wird  #N
00021000%*     in @ gespeichert                 #N
00022000GET,F,%SYSLST#N
00030000%*     SYSLST-Datei schließen           #N
00031000CLOSE SYSLST#N
00040000%*     SYSLST-Datei drucken             #N
00041000SYS /PRINT @...#E
```

Beispiel einer Kurzkommandokette unter TIAM:

Die nachfolgende Kurzkommandokette wird mit internen Kommentaren erläutert. Diese stehen jeweils vor dem betreffenden Kurzkommando.

```
00010000%*   Ermitteln der TSN                              #N
00011000GET,F,%SYSTSN#N
00020000%*   Speichern der TSN unter A                      #N
00021000%STOREA#N
00030000%*   Ermitteln der Uhrzeit                         #N
00031000GET,F,%SYSTIM#N
00040000%*   Speichern der Uhrzeit unter B                 #N
00041000%STOREB#N
00050000%*   Lesen der TSN                                 #N
00051000%GETA#N
00060000%*   Lesen der Uhrzeit                             #N
00061000%GETB#N
00070000%*   Zuweisung der SYSLST-Datei                    #N
00071000%*   mit TSN und Uhrzeit im Namen                  #N
00072000SYS /ASSIGN-SYSLST TO-FILE=SYSLST.@.@#N
00080000%*   Drucke-Kommando                               #N
00081000DR ...#N
00090000%*   SYSLST-Datei schließen                        #N
00091000SYS /ASSIGN-SYSLST TO-FILE=*PRIMARY#N
00100000%*   Lesen der TSN                                 #N
00101000%GETA#N
00110000%*   Lesen der Uhrzeit                             #N
00111000%GETB#N
00120000%*   SYSLST-Datei ausdrucken                       #N
00121000SYS /PRINT-FILE FILE-NAME=SYSLST.@.@ ...#E
```