

STORE

Structured Mode-Syntax

```

STORE [RECORD] [IN] [FILE] view-name
      [PASSWORD=operand1]
      [CIPHER=operand2]
      [
        [
          USING
          GIVING
        ]
        NUMBER operand3
      ]

```

Reporting Mode-Syntax

```

STORE [RECORD] [IN] [FILE] view-name
      [PASSWORD=operand1]
      [CIPHER=operand2]
      [
        [
          USING
          GIVING
        ]
        NUMBER operand3
      ]
      {
        [USING] SAME [RECORD] [AS] [STATEMENT] [(r)]
        [
          SET
          WITH
        ]
        [operand4=operand5] ...
      }

```

Dieses Kapitel behandelt folgende Themen:

- Funktion
- Datenbankspezifische Anmerkungen
- Syntax-Beschreibung
- Beispiel

Eine Erläuterung der in dem Syntax-Diagramm verwendeten Symbole entnehmen Sie dem Abschnitt *Syntax-Symbole*.

Verwandte Statements: ACCEPT/REJECT | AT BREAK | AT START OF DATA | AT END OF DATA | BACKOUT TRANSACTION | BEFORE BREAK PROCESSING | DELETE | END TRANSACTION | FIND | GET | GET SAME | GET TRANSACTION DATA | HISTOGRAM | LIMIT | PASSW | PERFORM BREAK PROCESSING | READ | RETRY | UPDATE

Gehört zur Funktionsgruppe: *Datenbankzugriffe und Datenbankänderungen*

Funktion

Das Statement STORE dient dazu, auf einer Datenbank einen Datensatz hinzuzufügen.

Datenbankspezifische Anmerkungen

Adabas	Die Natural-Systemvariable *ISN enthält die Adabas ISN, die dem neuen Datensatz als Ergebnis der Ausführung des STORE-Statements zugewiesen wurde. Eine anschließende Referenz auf *ISN muss die Statement-Nummer des betreffenden STORE-Statements enthalten.
DL/I	<p>Mit dem STORE-Statement kann eine Segment-Ausprägung hinzugefügt werden.</p> <p>Ist der Datensatz mit einem Primärschlüssel definiert, so muss ein Wert für das Primärschlüssel-Feld angegeben werden.</p> <p>Bei einer GSAM-Datenbank müssen neue Datensätze am Ende der Datenbank hinzugefügt werden (aufgrund von GSAM-Beschränkungen).</p> <p>Die Natural-Systemvariable *ISN steht nicht zur Verfügung.</p>
SQL	<p>Mit dem STORE-Statement können Sie einer Tabelle eine Reihe hinzufügen. Die PASSWORD-, CIPHER- und GIVING NUMBER-Klauseln sind nicht erlaubt.</p> <p>Das STORE-Statement entspricht dem SQL-Statement INSERT.</p> <p>Die Natural-Systemvariable *ISN steht nicht zur Verfügung.</p>
VSAM	<p>Ist der Datensatz mit einem Primärschlüssel definiert, so muss ein Wert für das Primärschlüssel-Feld angegeben werden.</p> <p>Die Natural-Systemvariable enthält die Adabas-ISN bzw. VSAM-RBA oder -RRN, die dem neuen Datensatz als Ergebnis der STORE-Operation zugeteilt wurde. Um den Wert von *ISN zu erhalten, müssen Sie das betreffende STORE-Statement referenzieren (mittels Statement-Label oder Sourcecode- Zeilennummer).</p> <p>Bei VSAM-Dateien gilt *ISN nur für ESDS- und RRDS-Dateien.</p>

Syntax-Beschreibung

Operanden-Definitionstabelle:

Operand	Mögliche Struktur				Mögliche Formate										Referenzierung erlaubt	Dynam. Definition	
<i>operand1</i>	C	S			A											ja	nein
<i>operand2</i>	C	S				N										ja	nein
<i>operand3</i>		S				N	P		B							nein	ja
<i>operand4</i>		S	A		A	N	P	I	F	B	D	T	L			nein	nein
<i>operand5</i>	C	S	A		A	N	P	I	F	B	D	T	L			ja	nein

* Format B von *operand3* kann nur mit einer Länge von kleiner gleich 4 benutzt werden.

Syntax-Element-Beschreibung:

<i>view-name</i>	<p>Als <i>view-name</i> geben Sie den Namen eines Views an, der entweder in einem DEFINE DATA-Block oder in einer programmexternen Global oder Local Data Area definiert ist.</p> <p>Im Reporting Mode ist <i>view-name</i> der Name eines DDM, falls kein DEFINE DATA LOCAL-Statement benutzt wird.</p>
PASSWORD=<i>operand1</i>	<p>Die PASSWORD-Klausel gilt nur bei Adabas- und VSAM-Datenbanken.</p> <p>Sie dient dazu, ein Passwort (<i>operand1</i>) anzugeben, um Daten auf einer passwortgeschützten Datei speichern zu können. Das Passwort (<i>operand1</i>) kann als eine alphanumerische Konstante oder als eine alphanumerische Variable angegeben werden. Es kann aus bis zu 8 Zeichen bestehen und darf keine Sonderzeichen oder eingebettete Leerzeichen enthalten. Wenn das Passwort als eine Konstante angegeben wird, muss es in Apostrophen stehen.</p> <p>Weitere Informationen siehe die Statements FIND und PASSW.</p>
CIPHER=<i>operand2</i>	<p>Die CIPHER-Klausel gilt nur bei Adabas- und VSAM-Datenbanken.</p> <p>Diese Klausel wird benutzt, um einen Chiffrierschlüssel (<i>operand2</i>) bei der Aktualisierung von Daten einer Datei anzugeben, die verschlüsselt ist. Der Chiffrierschlüssel (<i>operand2</i>) kann als eine numerische Konstante mit 8 Stellen oder als eine Benutzervariable mit Format/Länge N8 angegeben werden.</p> <p>Weitere Informationen siehe das Statement FIND.</p>

<p>USING NUMBER <i>operand3</i></p> <p>or</p> <p>GIVING NUMBER <i>operand3</i></p>	<p>Diese Klausel kann nur bei Adabas- und VSAM-Datenbanken benutzt werden. Bei VSAM-Datenbanken gilt sie nur für VSAM-RRDS, wobei die vom Benutzer angegebene RRN (Relative Record Number) der ISN entspricht.</p> <p>Mit dieser Klausel können Sie für einen zu speichernden Datensatz eine eigene Adabas-ISN angeben. Ist die angegebene ISN bereits vergeben, wird das Programm abgebrochen und eine entsprechende Fehlermeldung ausgegeben, es sei denn, es ist eine ON ERROR-Verarbeitung definiert.</p>
<p>SET/WITH <i>operand4=operand5</i></p>	<p>Mit dieser Klausel können im Reporting Mode die Felder angegeben werden, für die Werte gespeichert werden sollen. Jedes in der Datei definierte Feld, das in der SET-Klausel nicht angegeben wird, erhält in dem neuen Datensatz einen Nullwert.</p> <p>Diese Klausel ist nicht erlaubt, wenn ein DEFINE DATA-Statement verwendet wird, da sich in diesem Fall das STORE-Statement immer auf den gesamten im DEFINE DATA-Statement definierten View bezieht.</p> <p>Hinweise für DL/I:</p> <p>Ein Segment mit variabler Länge wird mit der Länge gespeichert, die mindestens benötigt wird, um alle im STORE-Statement angegebenen Felder aufnehmen zu können. Die Segmentlänge unterschreitet jedoch nie die im SEGM-Makro der DBD definierte Mindestlänge. Es müssen für das Segment-Folgefild sowie für alle Folgefelder der <i>Ancestors</i> Werte angegeben werden. Es dürfen nur Werte für I/O-sensitive Felder angegeben werden. Wird ein multiples Feld oder eine Periodengruppe mit variabler Länge definiert, so werden am Ende des Segments nur die im STORE-Statement angegebenen Ausprägungen zu dem Segment hinzugeschrieben, welche die Segmentlänge bestimmen.</p>
<p>USING SAME (<i>r</i>)</p>	<p>Diese Klausel bewirkt im Reporting Mode, dass dieselben Feldwerte, die mit dem FIND-, GET- oder READ-Statement gelesen wurden, welches von dem STORE-Statement referenziert wird, als Werte des neuen Datensatzes gespeichert werden. Das Statement kann mit der Notation (<i>r</i>) mittels Sourcecode-Zeilenummer oder Statement-Label referenziert werden.</p> <p>Diese Klausel ist nicht erlaubt, wenn ein DEFINE DATA-Statement verwendet wird, da sich in diesem Fall das STORE-Statement immer auf den gesamten im DEFINE DATA-Statement definierten View bezieht.</p>

Beispiel

```

** Example 'STOEX1S': STORE (structured mode)
**
** CAUTION: Executing this example will modify the database records!
*****
DEFINE DATA LOCAL
1 EMPL-VIEW VIEW OF EMPLOYEES
  2 PERSONNEL-ID
  2 NAME
  2 FIRST-NAME
  2 MAR-STAT
  2 BIRTH
  2 CITY
  2 COUNTRY
*
1 #PERSONNEL-ID (A8)
1 #NAME (A20)
1 #FIRST-NAME (A15)
1 #BIRTH-D (D)
1 #MAR-STAT (A1)
1 #BIRTH (A8)
1 #CITY (A20)
1 #COUNTRY (A3)
1 #CONF (A1)
END-DEFINE
*
REPEAT
  INPUT 'ENTER A PERSONNEL ID AND NAME (OR ''END'' TO END)' //
    'PERSONNEL-ID : ' #PERSONNEL-ID //
    'NAME : ' #NAME /
    'FIRST-NAME : ' #FIRST-NAME

  /*
  /* VALIDATE ENTERED DATA
  /*
  IF #PERSONNEL-ID = 'END' OR #NAME = 'END'
    STOP
  END-IF
  IF #NAME = ' '
    REINPUT WITH TEXT 'ENTER A LAST-NAME' MARK 2 AND SOUND ALARM
  END-IF
  IF #FIRST-NAME = ' '
    REINPUT WITH TEXT 'ENTER A FIRST-NAME' MARK 3 AND SOUND ALARM
  END-IF
  /*
  /* ENSURE PERSON IS NOT ALREADY ON FILE
  /*
  FIND NUMBER EMPL-VIEW WITH PERSONNEL-ID = #PERSONNEL-ID
  IF *NUMBER > 0
    REINPUT 'PERSON WITH SAME PERSONNEL-ID ALREADY EXISTS'
    MARK 1 AND SOUND ALARM
  END-IF
  MOVE 'N' TO #CONF
  /*
  /* GET FURTHER INFORMATION
  /*
  INPUT
    'ADDITIONAL PERSONNEL DATA' //
    'PERSONNEL-ID : ' #PERSONNEL-ID (AD=IO) /
    'NAME : ' #NAME (AD=IO) /
    'FIRST-NAME : ' #FIRST-NAME (AD=IO) //
    'MARITAL STATUS : ' #MAR-STAT /

```

```

        'DATE OF BIRTH (YYYYMMDD) : ' #BIRTH           /
        'CITY                      : ' #CITY           /
        'COUNTRY (3 CHARACTERS)    : ' #COUNTRY        //
        'ADD THIS RECORD (Y/N)     : ' #CONF           (AD=M)
/*
/*   ENSURE REQUIRED FIELDS CONTAIN VALID DATA
/*
IF NOT (#MAR-STAT = 'S' OR = 'M' OR = 'D' OR = 'W')
    REINPUT TEXT 'ENTER VALID MARITAL STATUS S=SINGLE ' -
                'M=MARRIED D=DIVORCED W=WIDOWED' MARK 1
END-IF
IF NOT (#BIRTH = MASK(YYYYMMDD) AND #BIRTH = MASK(1582-2699))
    REINPUT TEXT 'ENTER CORRECT DATE' MARK 2
END-IF
IF #CITY = ' '
    REINPUT TEXT 'ENTER A CITY NAME' MARK 3
END-IF
IF #COUNTRY = ' '
    REINPUT TEXT 'ENTER A COUNTRY CODE' MARK 4
END-IF
IF NOT (#CONF = 'N' OR = 'Y')
    REINPUT TEXT 'ENTER Y (YES) OR N (NO)' MARK 5
END-IF
IF #CONF = 'N'
    ESCAPE TOP
END-IF
/*
/*   ADD THE RECORD
/*
MOVE EDITED #BIRTH TO #BIRTH-D (EM=YYYYMMDD)
/*
EMPL-VIEW.PERSONNEL-ID := #PERSONNEL-ID
EMPL-VIEW.NAME        := #NAME
EMPL-VIEW.FIRST-NAME  := #FIRST-NAME
EMPL-VIEW.MAR-STAT    := #MAR-STAT
EMPL-VIEW.BIRTH       := #BIRTH-D
EMPL-VIEW.CITY        := #CITY
EMPL-VIEW.COUNTRY     := #COUNTRY
/*
STORE RECORD IN EMPL-VIEW
/*
END OF TRANSACTION
/*
WRITE NOTITLE 'RECORD HAS BEEN ADDED'
/*
END-REPEAT
END

```

Ausgabe des Programms STOEX1S:

ENTER A PERSONNEL ID AND NAME (OR 'END' TO END)

PERSONNEL-ID : 90001100

NAME : JONES
 FIRST-NAME : EDWARD

Nach der Eingabe und Bestätigung der Personal-Schlüsseldaten werden zusätzliche Personal-Daten zur Eingabe angezeigt:

ADDITIONAL PERSONNEL DATA

PERSONNEL-ID : 90001100
NAME : JONES
FIRST-NAME : EDWARD

MARITAL STATUS :
DATE OF BIRTH (YYYYMMDD) :
CITY :
COUNTRY (3 CHARACTERS) :

ADD THIS RECORD (Y/N) : N

Äquivalentes Reporting-Mode-Beispiel: STOEX1R.