

# Flexible SQL

Dieses Kapitel behandelt folgende Themen:

- Flexible SQL benutzen
  - Textvariablen in Flexible SQL angeben
- 

## Flexible SQL benutzen

Zusätzlich zu der im bisherigen Verlauf dieses Kapitels beschriebenen SQL-Syntax haben Sie mit flexibler SQL die Möglichkeit, beliebige SQL-Syntax zu verwenden.

### Die Zeichen << und >>

Flexible SQL muss zwischen den Zeichen << und >> stehen. Sie kann beliebigen SQL-Text und *host-variables* enthalten. Mit flexibler SQL verwendete *host-variables* müssen als Präfix einen Doppelpunkt (:) haben.

Flexible SQL kann über mehrere Zeilen gehen und kann ganze oder teilweise Kommentarzeilen enthalten (vgl. `PROCESS SQL-Statement`).

Flexible SQL kann anstelle folgender SQL-Syntaxteile verwendet werden:

- *atom*
- *column-reference*
- *scalar-expression*
- *predicate*

Flexible SQL kann auch zwischen den Klauseln einer *select-expression* verwendet werden:

```
SELECT selection
  << ... >>
  INTO ...
  FROM ...
  << ... >>
  WHERE ...
  << ... >>
  GROUP BY ...
  << ... >>
  HAVING ...
  << ... >>
  ORDER BY ...
  << ... >>
```

### Anmerkung:

Der in flexibler SQL angegebene SQL-Text wird nicht vom Natural-Compiler verarbeitet, sondern (mit ausgetauschten *host-variables*) einfach in die SQL-Zeichenkette kopiert, die an das Datenbanksystem übergeben wird. Demzufolge werden Syntaxfehler in der flexiblen SQL erst zur Laufzeit erkannt, wenn die Datenbank das betreffende Statement ausführt.

**Beispiel 1:**

```
SELECT NAME
FROM SQL-EMPLOYEES
WHERE << MONTH (BIRTH) >> = << MONTH (CURRENT_DATE) >>
```

**Beispiel 2:**

```
SELECT NAME
FROM SQL-EMPLOYEES
WHERE << MONTH (BIRTH) = MONTH (CURRENT_DATE) >>
```

**Beispiel 3:**

```
SELECT NAME
FROM SQL-EMPLOYEES
WHERE SALARY > 50000
<< INTERSECT
  SELECT NAME
  FROM SQL-EMPLOYEES
  WHERE DEPT = 'DEPT10'
>>
```

## Textvariablen in Flexible SQL angeben

Innerhalb der flexiblen SQL können Sie auch sogenannte Textvariablen angeben.

<<:T:host-variable [ <b>LINDICATOR</b> :host-variable]>>
--

Die Syntax-Elemente sind im Folgenden beschrieben.

<b>:T:</b>	<p>Eine Textvariable ist eine <i>host-variable</i> mit dem Präfix <code>:T:</code>. Sie muss alphanumerisches Format haben.</p> <p>Zur Laufzeit wird eine Textvariable innerhalb eines SQL-Statements durch ihren Inhalt ersetzt, d.h. die in der Textvariablen enthaltene Textzeichenkette wird in die SQL-Zeichenkette eingefügt.</p> <p>Nach dem Ersetzen werden nachfolgende Leerzeichen aus der eingefügten Textzeichenkette entfernt.</p> <p>Sie müssen selbst darauf achten, dass sich aus dem Inhalt einer Textvariablen beim Einfügen ein syntaktisch korrektes SQL-Statement ergibt. Insbesondere darf eine Textvariable keine <i>host-variables</i> enthalten.</p> <p>Ein Statement, das eine Textvariable enthält, wird immer im dynamischen SQL-Modus ausgeführt.</p>
<b>LINDICATOR</b>	<p><b>LINDICATOR-Option:</b></p> <p>Nach der Textvariablen können Sie das Schlüsselwort <code>LINDICATOR</code> sowie eine Längenindikator-Variable (d.h. eine <i>host-variable</i> mit vorangestelltem Doppelpunkt) angeben.</p> <p>Die Längenindikator-Variable muss Format/Länge <code>I2</code> haben.</p> <p>Wenn Sie keine <code>LINDICATOR</code>-Variable angeben, wird der gesamte Inhalt der Textvariablen in die SQL-Zeichenkette eingefügt.</p> <p>Wenn Sie eine <code>LINDICATOR</code>-Variable angeben, werden nur die ersten <math>n</math> Zeichen (wobei <math>n</math> der Wert der <code>LINDICATOR</code>-Variablen ist) des Textvariableninhalts in die SQL-Zeichenkette eingefügt. Falls die Zahl in der <code>LINDICATOR</code>-Variablen größer als die Länge des Textvariableninhalts ist, wird der gesamte Textvariableninhalt eingefügt. Falls die Zahl in der <code>LINDICATOR</code>-Variablen negativ oder Null (0) ist, wird nichts eingefügt.</p> <p>Siehe auch allgemeine Informationen zu <i>host-variable</i>.</p>

### Beispiel mit Textvariable:

```

DEFINE DATA LOCAL
01 TEXTVAR (A200)
01 TABLES VIEW OF SYSIBM-SYSTABLES
    02 NAME
    02 CREATOR
END-DEFINE
*
MOVE 'WHERE NAME > ''SYS'' AND CREATOR = ''SYSIBM'' ' TO TEXTVAR
*
SELECT * INTO VIEW TABLES
FROM SYSIBM-SYSTABLES
<< :T:TEXTVAR >>
DISPLAY TABLES
END-SELECT
*
END

```

Das generierte SQL-Statement (wie mit dem Systemkommando LISTSQL angezeigt) sieht wie folgt aus:

```
SELECT NAME, CREATOR FROM SYSIBM.SYSTABLES:T: FOR FETCH ONLY
```

Das ausgeführte SQL-Statement sieht wie folgt aus:

```
SELECT TABNAME, CREATOR FROM SYSIBM.SYSTABLES  
WHERE TABNAME > 'SYS' AND CREATOR = 'SYSIBM'
```