

2 SQL Sprachelemente

Grundlegende Sprachelemente von SQL.

2.1 Übersicht

Themen des Kapitels – *SQL Sprachelemente*

Themen des Kapitels

- SQL (Structured Query Language)
- Schemata
- Datentypen

Im Kapitel *SQL Sprachelemente* werden die grundsätzlichen SQL Strukturen und Konstrukte beschrieben.

2. SQL Sprachelemente

2.1.1 SQL (Structured Query Language)

2.1.1.1 Allgemeines

SQL (Structured Query Language)

- SQL ist eine Sprache der dritten Generation
- Standardisiert durch ANSI

Die Datenbanksprache SQL (**S**tructured **Q**uery **L**anguage) wurde in den 70er Jahren von IBM entwickelt. Das ab 1981 *kommerziell* verfügbare System SQL/Data-System von IBM war das erste System, welches mit SQL als Abfragesprache auf den Markt kam. In den Folgejahren verbreitete sich SQL immer mehr, so dass auch andere Anbieter RDBMS nach und nach systemeigene Sprachen gegen SQL ersetzen. Zu Beginn der 80er Jahre begann das American National Standards Institute (ANSI) mit der Standardisierung (Normung) dieser Sprache. In den vergangenen Jahren kamen Ergänzungen und Modifikationen hinzu; 1992 wurde SQL2 festgeschrieben, das als ANSI-SQL92 bekannt ist.

2.1.1.2 SQL als Sprache

SQL als Sprache

- DDL (Data Definition Language)
- DML (Data Modification Language)
- DCL (Data Control Language)
- TC (Transaction Control)

SQL teilt sich grob in vier Bereiche auf:

- DDL (**D**ata **D**efinition **L**anguage) zur Definition von Tabellen und Wertebereichen. In diesem Bereich sind Befehle wie *CREATE TABLE*, *DROP TABLE*, *ALTER TABLE*, *CREATE VIEW*, *CREATE INDEX* usw. anzusiedeln.
- DML (**D**ata **M**anipulation **L**anguage) zur Veränderung von Tabelleninhalten und zur Suche. Hierzu gehören Befehle wie *INSERT*, *UPDATE*, *DELETE* und *SELECT* weitere.
- DCL (**D**ata **C**ontrol **L**anguage) zur Festlegung von Benutzern und deren (Zugriffs-) Rechte. Diese Befehle sind meist nur vom Administrator der Datenbank ausführbar. In diesen Bereich fallen u.a. *CREATE USER*, *DROP USER*, *ALTER USER*, *GRANT*, ...
- TC (**T**ransaction **C**ontrol) zur Steuerung der Transaktionen. In diesen Bereich fallen die Befehle *COMMIT*, *ROLLBACK* und *SAVEPOINT*.

2.1.1.3 SQL bei Oracle

SQL bei Oracle

- SQL*Plus
- PL/SQL (Prozedurales SQL)

Oracle hat bei ihrem Datenbank-System die Abfragesprache SQL um viele nützliche Funktionen erweitert, so dass weit mehr Möglichkeiten gegeben sind, als das bei reinem ANSI-SQL der Fall ist. SQL bei Oracle teilt sich in zwei Sprachen auf:

- SQL*Plus, die erweiterte SQL-Abfragesprache von Oracle.
- PL/SQL, die prozedurale Abfragesprache von Oracle.

SQL*Plus ist das bekannte SQL erweitert um Oracle-eigene Sprachelemente. PL/SQL ist eine prozedurale Sprache mit Pascal-ähnlichen Elementen wie Blockstruktur mit *DECLARE*, *BEGIN*, *END* und Steuerungselementen wie z.B. *IF-THEN*-Entscheidung, Schleifen und Ausnahmebehandlung.

2.2 Schema

Erläuterung des Schemas.

Schema

- Objekte der Datenbank sind immer einem Besitzer zugeordnet
- Man spricht dabei von einem Schema
- z.B. *scott.emp* -> die Tabelle *emp* ist im Schema *scott*
- Die Angabe des Schemas ist immer dann notwendig wenn die Tabelle nicht im eigenen Schema ist

Unter einem Schema versteht man die Menge aller Datenbank Objekte die einen bestimmten Besitzer (Oracle Benutzer) gehören.

Oracle Objekte sind z.B. Tabellen, Prozeduren, Funktionen, usw.

Wird ein Datenbankobjekt angesprochen, welches sich im Schema eines anderen Benutzer befindet muss dieser angegeben werden:

```
SELECT * FROM scott.emp
```

In diesem Beispiel würde die Tabelle *emp* im Schema des Benutzers *Scott* angesprochen.

2.3 Datentypen

Übersicht der verschiedenen Datentypen.

Datentypen

- CHAR
- NCHAR
- VARCHAR2
- NVARCHAR2
- NUMBER(p,s)
- DECIMAL
- FLOAT
- REAL
- INTEGER
- DATE

- LONG
- BLOB
- CLOB
- NCLOB
- BFILE
- RAW
- LONGRAW
- ROWID
- UROWID

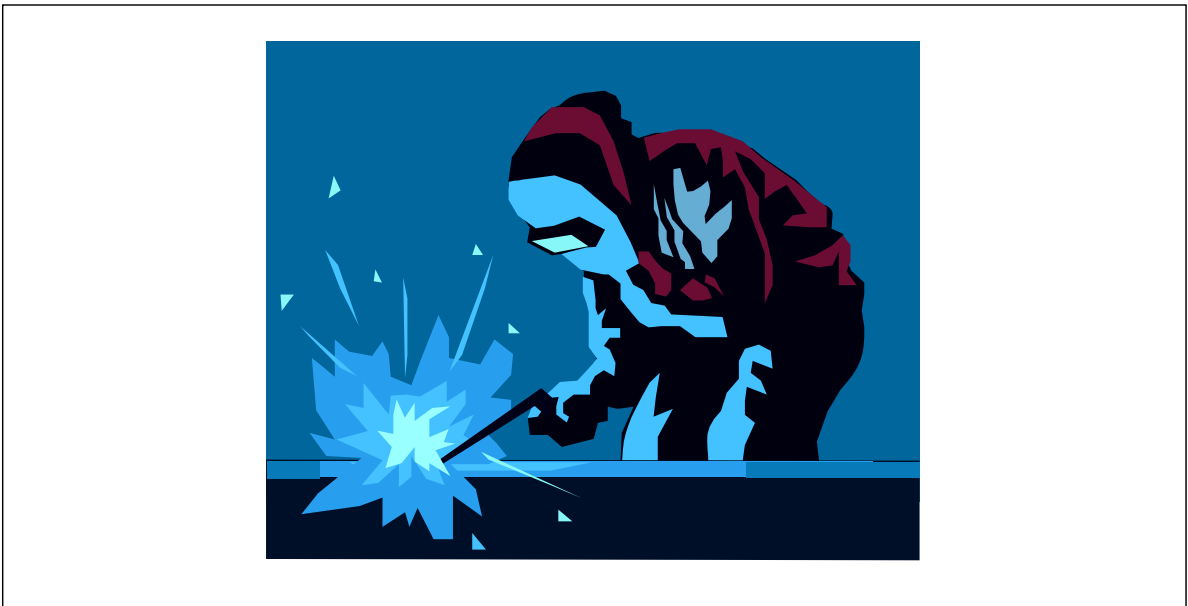
Datentyp	Code	Beschreibung
CHAR (Größe)	96	Zeichendaten fester Länge (maximal 2000 Bytes)
NCHAR (Größe)	96	Zeichendaten fester Länge abhängig vom nationalen Zeichensatz (maximal 2000 Bytes)
VARCHAR2 (Größe)	1	Zeichendaten variabler Länge (maximal 4000 Bytes)
NVARCHAR2 (Größe)	1	Zeichendaten variabler Länge abhängig vom nationalen Zeichensatz (maximal 4000 Bytes)

2. SQL Sprachelemente

LONG	8	Zeichendaten variabler Länge (maximal 2 GByte) – veraltet sollte nicht mehr benutzt werden
BLOB	113	Binary Large Object (maximal 4 GByte)
CLOB	112	Character Large Object (maximal 4 GByte)
NCLOB	112	Character Large Object abhängig vom nationalen Zeichensatz (maximal 4 GByte)
BFILE	114	Zeiger auf ein Large Binary File das außerhalb der Datenbank gespeichert wird (maximal 4 GByte)
NUMBER (p, s)	2	Precision $1 \leq p \leq 38$ Scale $-84 \leq s \leq 127$ (Bei negativen Werten für Scale wird bis zur entsprechenden Vorkommastelle gerundet)
DATE	12	Datum und Uhrzeit (01. Januar 4712 v. Chr. bis 31. Dezember 9999)
RAW (Größe)	23	Binäre Daten (maximal 2000 Byte)
LONG RAW	24	Binäre Daten variabler Länge (maximal 2 GByte) – veraltet sollte nicht mehr verwendet werden
ROWID	69	Eindeutige Adresse einer Zeile in der Datenbank.
UROWID (Größe)	208	Logische Adresse einer Zeile in einer Indexed Organized Table (maximal 4000 Byte)

2.4 Übung - Datentypen

Übung – Datentypen



In der Übung soll der Umgang mit den Datentypen erprobt werden.

1. Die Übung soll mit dem Programm *sqlplus.exe* durchgeführt werden. Die Anmeldung erfolgt mit dem Benutzer *userXX* und dem Passwort *password* erfolgen. Mit der View *USER_TABLES* können die Namen und Eigenschaften der Tabellen abgefragt werden.

2. Im ersten Schritt soll eine Tabelle erzeugt werden:

```
CREATE TABLE tab01 (numtest NUMBER(5,2));
```

3. Als nächstes kann eine *INSERT* auf die Tabelle durchgeführt werden:

```
INSERT INTO tab01 (numtest) VALUES (123.56);
```

4. Als letztes erfolgt die Anzeige mit:

```
SELECT * FROM tab01;
```