

LANDESBERUFSSCHULE 4 SALZBURG

Informatik

SQL-Statements

LBS 4

Inhalt

Einleitung	3
DDL (Data Definition Language).....	3
Befehle zum Erzeugen und entfernen von Tabellen.....	3
Beispiel:.....	3
Data Manipulation Language.....	4
Befehle	4
Beispiel: Datensätze einfügen in eine Datenbank	4
Beispiel: Datensätze ändern in einer Datenbank.....	4
Beispiel: Selektieren von Einträgen	4
JOIN	5
INNER JOIN	5
LEFT OUTER JOIN	5
RIGHT OUTER JOIN.....	6
Aufgabe:.....	6

Einleitung

Die Grundlage für relationale Datenbanken sind Tabellen mit ihren Spalten und Zeilen. In der Vertikalen sind die Spalten und in der Horizontalen die Zeilen angegeben. Eine Zeile (auch Tupel genannt) entspricht einem Element einer Tabelle, eine Spalte (auch Attribut genannt) einem Eintrag einer Tabelle.

Man unterscheidet zwischen Data Definition Language (DDL), Data Manipulation Language (DML) und der Data Control Language (DCL).

DDL (Data Definition Language)

Die Data Definition Language (DDL) ist ein Teil der Datenbanksprache SQL. Sie wird verwendet um Datenstrukturen und verwandte Elemente zu beschreiben, zu ändern oder zu entfernen. Ursprünglich bezog sich DDL auf Datenbanksysteme, der Begriff wird aber heute auch in anderen Zusammenhängen verwendet. Als Teil von SQL ist DDL die Datenbeschreibungssprache einer Datenbank.

Befehle zum Erzeugen und entfernen von Tabellen

`CREATE` (Tabelle erzeugen)

`ALTER` (Mit `ALTER TABLE` wird die Struktur einer Tabelle geändert)

`DROP` (`DROP COLUMN` – Spalte entfernen)

Beispiel:

```
CREATE TABLE <tabellenname>
(feldname datentyp{,feldname datentyp}
)
CREATE TABLE person(
persnr INT GENERATED ALWAYS as IDENTITY( START with 1,INCREMENT by 1),
firstname CHAR(45),
lastname CHAR(45),
age INT,
residence varchar(20),
CONSTRAINT PRIMARY KEY Primary Key(persnr)
)
```

Data Manipulation Language

Die Data Manipulation Language (DML, englisch für „Datenbearbeitungssprache“) ist ein weiterer Teil einer Datenbanksprache SQL. DML wird verwendet um Daten zu lesen, zu schreiben, zu ändern und zu löschen. DML ist die Datenver- oder Datenbearbeitungssprache einer Datenbank und schließt die Formulierung von Abfragen ein.

Befehle

- INSERT + VALUES ein einzelner Datensatz wird eingefügt
- INSERT + SELECT eine Menge von Datensätzen mit Hilfe einer Abfrage werden eingefügt
- UPDATE eine Menge von Datensätzen werden geändert, die Menge wird durch WHERE festgelegt
- DELETE eine Menge von Datensätzen werden gelöscht, die Menge wird durch WHERE festgelegt.
- TRUNCATE alle Datensätze einer Tabelle werden gelöscht
- SELECT

Beispiel: Datensätze einfügen in eine Datenbank

```
INSERT INTO person(firstname,lastname,age,residence)
VALUES (
'Max', 'Mustermann', 30, 'Salzburg')
```

Beispiel: Datensätze ändern in einer Datenbank

```
ALTER TABLE person
    add female boolean

DROP TABLE person
```

Beispiel: Selektieren von Einträgen

Nach dem SELECT kann ein einzelnes Attribut, mehrere Attribute oder der * Operator für alle Attribute folgen. Nach dem Schlüsselwort FROM wird die Tabelle(n) angegeben. Mit den Schlüsselwort WHERE kann das Ergebnis gefiltert werden.

```
SELECT Name FROM Tabelle WHERE a=?
```

JOIN

Ein JOIN verbindet bei der Abfrage mehrere Tabellen. Prinzipiell unterscheidet man zwischen

- INNER JOIN, EQUI JOIN, NATURAL JOIN (die Schnittmenge)
- FULL OUTER JOIN (alle Tupel der Tabellen)
- LEFT (OUTER) JOIN (Tupel der linken Tabelle und übereinstimmende der rechten Tabelle)
- RIGHT (OUTER) JOIN (Tupel der rechten Tabelle und Übereinstimmungen der linken Tabelle).

INNER JOIN

Beim INNER JOIN werden nur Spalten (Felder) angezeigt die identisch sind.

ID	Firma		ID	Produkt	L_NR		Firma	Produkt
1	Logitech		1	Maus	1		Logitech	Maus
2	Micro		2	Tastatur	1		Logitech	Tastatur
3	Netgear		3	Netzwerk	3		Netgear	Netzwerk
T1			T2			Ergebnis		

```
SELECT Firma, Produkt FROM T1 INNER JOIN T2 ON
(T1.ID = T2.L_NR)
```

LEFT OUTER JOIN

Beim LEFT OUTER JOIN werden auch Spalten der linken Tabelle in das Ergebnis einbezogen, bei denen es keine Übereinstimmung gibt. In der rechten Tabelle werden die Spalten mit NULL aufgefüllt.

ID	Firma		ID	Produkt	L_NR		Firma	Produkt
1	Logitech		1	Maus	1		Logitech	Maus
2	Micro		2	Tastatur	1		Logitech	Tastatur
3	Netgear		3	Netzwerk	3		Netgear	Netzwerk
T1			T2			Ergebnis		
						Micro	NULL	

```
SELECT Firma, Produkt FROM T1 LEFT (OUTER) JOIN T2 ON
(T1.ID = T2.L_NR)
```

RIGHT OUTER JOIN

Beim RIGHT OUTER JOIN werden alle Elemente der rechten Tabelle in das Ergebnis übernommen. Wenn kein Eintrag in der Linken Tabelle vorhanden ist wird er mit NULL aufgefüllt.

ID	Firma		ID	Produkt	L_NR		Firma	Produkt
1	Logitech		1	Maus	1		Logitech	Maus
2	Micro		2	Tastatur	1		Logitech	Tastatur
3	Netgear		3	Netzwerk	3		Netgear	Netzwerk
			4	Grafik	4		NULL	Grafik
T1			T2			Ergebnis		

```
SELECT Firma, Produkt FROM T1 RIGHT (OUTER) JOIN T2 ON
(T1.ID = T2.L_NR)
```

Aufgabe:

Erstellen Sie ein SQL-Skript welches das Datenmodell mit Inhalten füllt (mind. sieben Datensätze).

Führen Sie nachstehende Abfragen durch. Im Ergebnis soll/sollen

- alle Kunden mit ihrer Wohnadresse angezeigt werden
 - sortieren Sie diese aufwärts nach dem Vornamen (Firmennamen)
- die bestellten Produkte der Kunden angezeigt werden
 - die Bestellsumme einer Bestellung angezeigt werden
- alle Kunden deren Wohnort mit dem Buchstaben A oder a beginnt angezeigt werden
- alle Kunden die den Buchstaben ‚s‘ im Namen haben angezeigt werden
- alle Artikel die mehr als 10€ kosten angezeigt werden
- verwenden Sie die min max und avg Funktionen für die Bestellungen
- die Anzahl der verschiedenen Vornamen

Versuchen Sie eine „Stored Procedure“ mit und ohne Parameter zu erstellen.

Geben Sie die SQL-Skripte mit dem Datenbankmodell am Moodle ab

Allgemeine Infos:

<https://www.w3schools.com/sql/default.asp>