

Befehle in SQL

Inhalt

Der SQL Befehl allgemein

wichtige Befehle:

CREATE, USE, ALTER
INSERT, SELECT, UPDATE, DELETE
INNER JOIN, LEFT JOIN, etc.

Der SQL Befehl allgemein

Um mit dem Datenbankserver kommunizieren zu können bedarf es einer speziellen Sprache. Diese Sprache sind die SQL-Befehle.

Ein Befehl hat immer mehrere Bestandteile

BEFEHL tabelle [weitere Parameter] [evtl.
Einschränkungen];

Notation

Wenn wir SQL Befehle schreiben, schreiben wir diese
grundsätzlich GROß!

Feldnamen und Typen dagegen grundsätzlich klein.

Dies dient ins Besondere der Übersichtlichkeit.

Erstellen einer Datenbank

Ganz zu Beginn müssen wir zunächst eine Datenbank erstellen. Dies machen wir mit dem Befehl CREATE TABLE:

```
CREATE DATABASE test;
```

Dieser Befehl erstellt eine Datenbank mit dem Namen „test“.

Datenbank benutzen

Da ein SQL Server in aller Regel mehr als eine Datenbank verwaltet müssen wir ihm bei jeder neuen Sitzung mitteilen, welche Datenbank wir nutzen wollen.

Dies macht der Befehl USE

USE test;

im folgenden werden nun alle Befehle dieser Sitzung in der Datenbank „test“ ausgeführt.

Tabellen erstellen

Da unsere Datenbank nun noch völlig leer ist, soll unsere Aufgabe nun sein diese mit Tabellen zu füllen.

Eine Tabelle wird mit dem Befehl CREATE TABLE erstellt:

```
CREATE TABLE testtable (  
    id int(20) NOT NULL auto_increment,  
    name varchar(200) NOT NULL,  
    alter int(3) NOT NULL,  
    PRIMARY KEY(id));
```

Tabelle erstellen(2)

Dieser Befehl erstellt nun diese Tabelle:

Feldname	Type	Länge	Besonderheiten
id	int	20	Primary Key auto increment
name	varchar	200	-
alter	int	3	-

Diese kann nun mit Daten gefüllt werden.

Daten einfügen

Um Daten in eine Tabelle einzufügen verwendet man den Befehl INSERT INTO:

INSERT INTO (name,alter) VALUES ('Fritz Meier','40');

Dieser Befehl füllt die Tabelle mit den ersten Daten:

id	name	alter
1	Fritz Meier	40

Die id wird automatisch vergeben, da das Feld das Attribut auto_increment trägt.

Daten abrufen

Nun, da wir Daten in der Tabelle haben müssen wir diese auch wieder abrufen können:

```
SELECT [Felder] FROM [tabelle] WHERE [Bedingungen];
```

Als Beispiel diese Befehl:

```
SELECT * FROM testtable WHERE id='1';
```

Dieser Befehl ruft alle Felder aus der Tabelle auf, welche die id 1 haben. Da id Primärschlüssel ist, kann es nur ein solches Feld geben.

Daten ändern

Um Daten in einer Tabelle zu verändern verwenden wir den UPDATE Befehl:

```
UPDATE testtable SET name='Fritz Schulz' WHERE id='1';
```

Tabelle vorher:

id	name	alter
1	Fritz Meier	40

Tabelle nachher:

id	name	alter
1	Fritz Schulze	40

Daten Löschen

Um Daten wieder aus der Tabelle zu löschen verwenden wir nun den DELETE Befehl:

DELETE FROM testtable WHERE id='1'

dieser Befehl löscht unseren Datensatz wieder heraus.

Tabelle vorher:

id	name	alter
1	Fritz Meier	40

Tabelle nachher:

id	name	alter
----	------	-------

Wichtig: Die WHERE Klausel muss angegeben werden, da sonst alle Datensätze gelöscht werden würden.

Seltene Querys

Es gibt einige Querys, welche man recht selten benötigt. Hier gehören beispielsweise die folgenden dazu:

ALTER TABLE
LEFT JOIN

Verknüpfte Abfragen

Häufig ist es notwendig Daten aus mehreren Tabellen zu sammeln. Hierzu bietet sich ein JOIN Befehl an:

```
SELECT testtable.*,testtable2.* FROM testtable LEFT JOIN  
testtable2 ON (testtable.name=testtable2.tlname) WHERE  
testtable.id = ,l';
```

Dieser Befehl fragt alle Felder aus den Datenbanken testtable und testtable2 ab, bei welchen die testtable id='l' ist und die Felder name aus testtable sowie tlname aus testtable2 übereinstimmen.

LEFT JOIN (2)

testtable:

t1id	name	alter
1	Fritz Meier	40

testtable2:

t2id	t1name	geburtsort
1	Fritz Schulze	Kassel

Abfrageergebnis:

t1id	name	alter	t1name	geburtsort
1	Fritz Schulze	40	Fritz Schulze	Kassel