

HIT 2.Jahr	Datum:
AB Insert/Update/Delete	

MySQL-Anweisungen zum Einfügen/Ändern/Löschen von Datensätzen

1. Daten einfügen (Insert)

INSERT fügt einen Datensatz in eine vorhandene Tabelle ein. Es gibt zum Einen die Möglichkeit einen Datensatz anzulegen, indem die Werte der Attribute explizit angegeben werden.

Zum Ändern kann ein Datensatz angelegt werden, indem die Werte der Attribute aus einer oder mehreren anderen Tabellen ausgewählt werden.

Wir beschränken uns zunächst auf die 1.Variante, d.h. Wir legen die Attributwerte fest.

1.1 Insert mit Attributwerten

Die Formen INSERT ... VALUES und INSERT ... SET der Anweisung fügen Datensätze basierend auf explizit angegebenen Werten ein.

Es muss beachtet werden,

- dass alle Spaltenwerte, die mit NOT NULL definiert wurden, einen Wert erhalten müssen.
- dass bei Spalten, bei denen in der Definition ein DEFAULT-Wert festgelegt wurde, als Attributwert DEFAULT angegeben werden darf.
- dass für eine Spalte, die als Primärschlüssel mit AUTO_INCREMENT definiert wurde, automatisch ein Spaltenwert erzeugt wird.

```
INSERT [INTO] tabellen_name [(col_name,...)]
VALUES ({expr | DEFAULT},...)
```

Beispiel: INSERT INTO Kunde VALUES ('1', 'Meier', 'Hans'); oder

Fügt einen Datensatz in die Kundentabelle mit den Attributen K-Nr, Nachname, Vorname ein. Werden keine Spaltennamen angegeben, müssen die Attributwerte genau in der Reihenfolge angegeben werden, wie sie in der Tabellendefinition (beim Create) festgelegt wurden.

Beispiel: INSERT INTO Kunde (Nachname, Vorname)
VALUES ('Meier', 'Hans'),('Müller', 'Tim');

Durch AUTO_INCREMENT wird für den Primärschlüssel automatisch ein Wert erzeugt. Es können wie im Beispiel gleichzeitig mehrere Datensätze angelegt werden.

Hinweis: Mit der Anweisung Describe tbl_name kann die Tabellendefinition angezeigt werden.

```
INSERT [INTO] tabellen_name SET col_name={expr | DEFAULT}, ...
```

Beispiel: INSERT INTO Kunde SET Nachname='Meier', Vorname='Hans';

1.2 Insert mit Select

Die Form INSERT ... SELECT fügt Datensätze ein, wobei Spaltenwerte in einer oder mehreren anderen Tabellen ausgewählt werden.

```
INSERT [INTO] tabellen_name [(col_name,...)] SELECT ...
```

Hinweis: Die MySQL-Syntax ist hier in verkürzter Form aufgeführt und kann folgender Quelle entnommen werden: <http://dev.mysql.com/>.

HIT 2.Jahr	Datum:
AB Insert/Update/Delete	

2. Daten ändern/aktualisieren (Update)

Die SQL-Anweisung `UPDATE tabellen_name SET ...` ändert vorhandene Datensätze.

Die `SET`-Klausel gibt an, welche Spalten geändert werden und welche Werte diese erhalten sollen. Sofern angegeben, bestimmt die `WHERE`-Klausel die Bedingungen dafür, welche Datensätze geändert werden. Die Anzahl der geänderten Datensätze wird zurückgeliefert.

Mit `ORDER BY` kann bei einer Auswahl von Datensätzen eine Sortierung festgelegt werden. `ASCENDING`, kurz `ASC`, steht für aufsteigend in alphabetischer Reihenfolge, `DESCENDING`, kurz `DESC`, für absteigend.

Achtung: Ohne `WHERE`-Klausel werden **alle** Datensätze aktualisiert.

```
UPDATE tabellen_name
    SET col_name1=expr1 [, col_name2=expr2 ...]
    [WHERE where_condition]
    [ORDER BY ...]
```

Beispiel: Update Kunde
`SET Nachname='Meier-Berg', Vorname='Jan'`
`WHERE Nachname='Meier' AND Vorname='Hans';`

Ändert den Nachnamen von Hans Meier auf den Vornamen Jan und den Nachnamen auf Meier-Berg.

3. Daten löschen (Delete)

Delete-Anweisungen löschen Daten in einer (oder mehreren) Tabelle(n), wobei ein einschränkendes Selektionskriterium angegeben werden kann. Die Anzahl der gelöschten Datensätze wird zurückgeliefert.

Achtung: Ohne `Where`-Klausel werden alle Datensätze einer Tabelle gelöscht.

```
DELETE FROM tabellen_name
    [WHERE where_condition]
    [ORDER BY ...]
```

Beispiel: `DELETE FROM Kunde WHERE Nachname='Meier';`

Der Datensatz des Kunden mit dem Nachnamen Meier wird gelöscht.

4. Eingefügte Daten anzeigen (Select)

Mit einer einfachen `Select`-Anweisung können alle Daten einer Tabelle zeilenweise - entsprechend der Struktur der Tabelle - angezeigt werden. Der Platzhalter `'*'` sorgt dafür, dass die Werte aller Spalten angezeigt werden, ohne dass einzelne Spalten ausgewählt werden.

```
SELECT * FROM tabellen_name
```

Beispiel: `SELECT * FROM Kunde;`

Alle Datensätze der Tabelle Kunde werden angezeigt.

Weiterführende Links:

<http://dev.mysql.com/doc/> MySQL Reference Manuals

www.mysql.de MySQL Homepage