

SQL - DDL



Üben von DDL und DML

<https://www.jdoodle.com/execute-sql-online>

Your Code ...

```
1 create table calc(x int, y int);  
2  
3 insert into calc values(10, 25);  
4  
5 select x,y, (x+y) from calc;  
6
```

Ergebnis:

10|25|35

Befehlsgruppen in SQL

DDL

Data Definition
Language

- ✓ CREATE: Tabellen anlegen
- ✓ DROP: Tabellen löschen
- ✓ ALTER: Tabellenstruktur ändern

DML

Data Manipulation
Language

- ✓ SELECT: Datensätze wählen
- ✓ INSERT: Datensätze zufügen
- ✓ UPDATE: Datensätze ändern
- ✓ DELETE: Datensätze löschen

DSL

Data Security
Language

- ✓ BEGINTRANS: Transaktion beginnen
- ✓ COMMITTRANS: Transaktion erfolgreich beenden
- ✓ ROLLBACK: Transaktion abbrechen

DDL Befehl: Create

Funktion:

Erstellt eine Tabelle

Syntax:

```
CREATE TABLE table_name (  
    column1 datatype,  
    column2 datatype,  
    column3 datatype,  
    ....  
);
```

Beispiel:

```
CREATE TABLE Persons (  
    PersonID int,  
    LastName varchar(255),  
    FirstName varchar(255),  
    Address varchar(255),  
    City varchar(255)  
);
```

DDL Befehl: Create II

Datentype für CREATE TABLE :

CHAR(size)	Holds a fixed length string (can contain letters, numbers, and special characters). The fixed size is specified in parenthesis. Can store up to 255 characters
VARCHAR(size)	Holds a variable length string (can contain letters, numbers, and special characters). The maximum size is specified in parenthesis. Can store up to 255 characters. Note: If you put a greater value than 255 it will be converted to a TEXT type
TINYTEXT	Holds a string with a maximum length of 255 characters
TEXT	Holds a string with a maximum length of 65,535 characters
BLOB	For BLOBs (Binary Large Objects). Holds up to 65,535 bytes of data

DDL Befehl: Alter

Funktion:

Ändert die Tabellenstruktur: Fügt zu oder löscht Spalten oder ändert diese

Syntax:

```
ALTER TABLE table_name  
DROP COLUMN column_name;
```

Syntax:

```
ALTER TABLE table_name  
ADD column_name datatype;
```

Syntax:

```
ALTER TABLE table_name  
MODIFY column_name datatype;
```

Beispiel:

```
ALTER TABLE Persons  
ADD DateOfBirth date;
```

Tabelle

Name des
neuen Feldes

Datentyp

DDL Befehl: DROP

Funktion:

Löscht die Tabelle

Syntax:

```
DROP TABLE table_name;
```

Beispiel:

```
DROP TABLE Shippers;
```

DML Befehl: INSERT INTO

Funktion:

Fügt Datensätze in die Tabelle ein

Syntax:

```
INSERT INTO table_name (column1, column2, column3, ...)  
VALUES (value1, value2, value3, ...);
```

Syntax:

```
INSERT INTO table_name  
VALUES (value1, value2, value3, ...);
```

Diese Syntax kann verwendet werden, wenn sie **alle** Spalten mit Werten auffüllen

DML Befehl: INSERT INTO II

Beispiel 1:

```
INSERT INTO Personal (Vorname, Nachname, Position)
VALUES ('Harry', 'Washington', 'Trainee');
```

Beispiel 2:

```
INSERT INTO Personal
VALUES ('Harry', 'Washington', 'Trainee');
```

Beispiel 3:

```
INSERT INTO Kunden SELECT * FROM NeuKunden;
```

Fügt alle Datensätze von der Tabelle NeuKunden zu Kunden hinzu

DML Befehl: UPDATE

Funktion:

Ändert Werte in Feldern aufgrund Kriterien

Syntax:

```
UPDATE table_name  
SET column1 = value1, column2 = value2, ...  
WHERE condition;
```

Beispiel:

```
UPDATE Bestellungen  
SET Bestellmenge = Bestellmenge * 1.1, Fracht = Fracht * 1.03  
WHERE Bestimmungsland = 'USA';
```

DML Befehl: UPDATE II

- ✓ UPDATE erzeugt keine Ergebnismenge.
- ✓ Die Operation kann nicht rückgängig gemacht werden.
- ✓ Welche Datensätze werden aktualisiert:
 - Zunächst Ergebnisse einer Auswahlabfrage(SELECT) untersuchen, die dieselben Auswahlkriterien verwendet
 - Aktualisierungsabfrage(UPDATE) ausführen.
- ✓ Erstellen Sie immer Sicherungskopien Ihrer Daten.
- ✓ Wenn Sie einen falschen Datensatz aktualisieren, können Sie diesen mit Hilfe der Sicherungskopien wiederherstellen

DML Befehl: DELETE

Funktion:

Löscht Datensätze aus einer Tabelle, die den Kriterien in dem WHERE Abschnitt entsprechen

Syntax:

```
DELETE FROM table_name  
WHERE condition;
```

Beispiel:

```
DELETE FROM Customers  
WHERE CustomerName='Alfreds Futterkiste';
```

DML Befehl: DELETE II

- ✓ Es werden nur die Daten gelöscht, die Tabellenstruktur(Feldattribute und Indizes) bleiben erhalten!
- ✓ Bei 1:n-Beziehung: Operationen mit Löschoption in MS-Access löschen zusätzlich die Datensätze auf der n-Seite einer Beziehung, die mit dem Datensatz auf der 1-Seite, der durch die Abfrage gelöscht wird, in Beziehung stehen.

