



SQL

Structured Query Language

Teil2:

Data Manipulation Language



Die Datenbankabfragesprache SQL

1. Data Manipulation Language

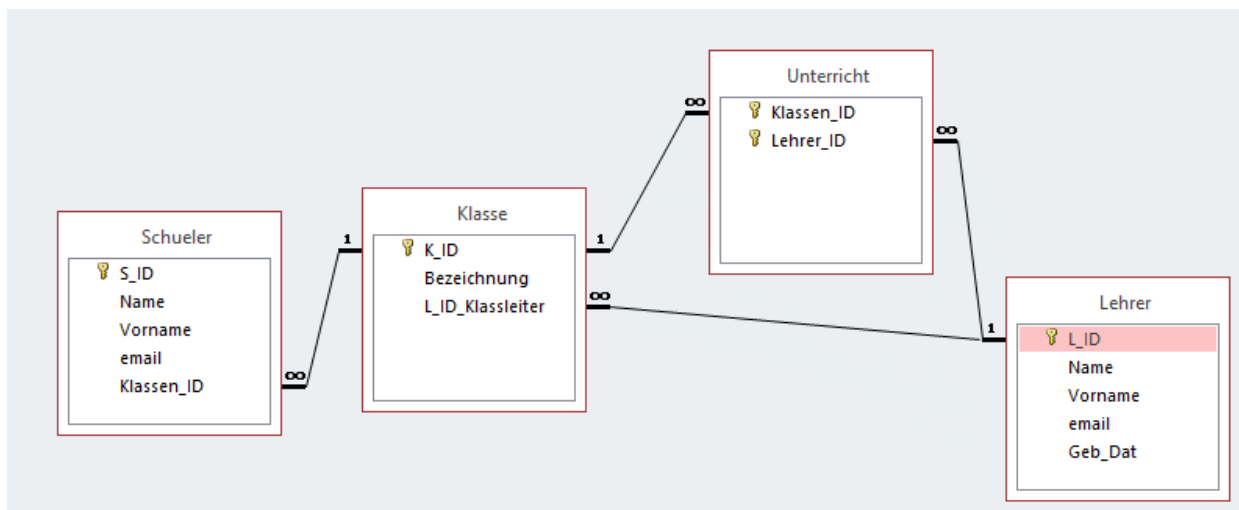
Die Data Manipulation Language (DML) ist eine Teilsprache von SQL und umfasst alle Anweisungen zur Beeinflussung der gespeicherten Daten. Grundlegende Operationen sind dabei: Eingabe, Änderung und Löschen von Daten.

2. Einfügen – INSERT Anweisung

Mit der INSERT-Anweisung werden Daten in Tabellen eingetragen. Dabei können sowohl ganze Zeilen als auch nur Teile von Tupeln (Datensätzen) mit Werten belegt werden. Die übrigen, nicht belegten Spalten bekommen jeweils den Standardwert NULL.

Die Daten können einmal durch eine SELECT-Anweisung aus einer anderen Tabelle ermittelt werden. Ebenfalls kann ein einzelner Satz durch Angabe der Felder und deren Werte eingefügt werden. Die Datentypen der Werte in der Werteliste müssen mit denjenigen der zugehörigen Spalte übereinstimmen. Die SELECT-Anweisung wird im Kapitel SQL genauer beschrieben.

Relationales Datenmodell der Schulverwaltung



Syntax

Variante 1:

Die Spaltennamen werden nicht angegeben, aber alle Felder müssen in der richtigen Reihenfolge gefüllt werden, notfalls mit „null“, dem Leer-Indikator.

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

Variante 2:

Nur die Attributnamen, die angegeben werden, werden in der entsprechenden Reihenfolge gefüllt. D.h. die Reihenfolge der Werte muss auch der Reihenfolge der Spalten in der Tabelle entsprechen. Es können auch Teilbereiche angegeben werden.

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

Variante 3:

Es können auch mehrere Datensätze in einer Anweisung erstellt werden.

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

Beispiele: Einfügen Datensatz in die Tabelle Lehrer

```
insert into Lehrer
values('WaFl','Wagner','Florian','wagnerf@itbspa.de','1900-12-12');
```

```
insert into Lehrer (L_ID, Name, Vorname, Email, Geb_Dat)
values('WaFl', 'Wagner', 'Florian', 'wagnerf@itbspa.de', null);
```

```
insert into Lehrer (L_ID, Name, Vorname)
values('WaFl','Wagner','Florian'),
('KaRi','Kaiser','Richard');    --Email und Geb_Dat sind leer
```

Teilweise oder ganze Auszüge aus einer Datenbank werden als Datenbankdump (dump = engl. für Auszug) bezeichnet. Derartige Auszüge werden für die Datensicherung oder Portierung der Datenbank erstellt.

Hinweis:

Die **Reihenfolge** ist wegen der referentiellen Integrität **unbedingt einzuhalten!**

In der Tabelle Klasse kann erst dann ein Datensatz mit einem Klassenleiter für die entsprechende Schulklasse erstellt werden, wenn es den Lehrer dazu gibt. Ein Schüler kann erst dann angelegt werden, wenn es eine Schulklasse gibt und die Zwischentabelle Unterricht braucht Lehrer und Schulklassen.

Beispiel: Datenbankdump für die Schulverwaltung

```
insert into Lehrer
values('KaRi','Kaiser','Richard','ri.kaiser@itbspa.de','1970-12-31') ;

insert into Lehrer
values('MaJo','Maderer','Josef','j.maderer@itbspa.de','1980-11-30');

insert into Lehrer (L_ID,Name, Vorname, Email, Geb_Dat)
values('PaMa','Paßberger','Markus','m.passberger@itbspa.de','1970-10-31');

insert into Klasse
values('13ITD','Fachinformatiker SI','MaJo');

insert into Klasse
values('13ITK','IT-Kaufleute','PaMa');

insert into Klasse
values('13ITA','Fachinformatiker SI/AE','MaJo');

insert into Schueler
values('13ITA007','Muster','Max','max.muster@musterfirma.de','13ITA');

insert into Schueler
values('13ITD007','Muster','Maria','maria.muster@musterfirma.de','13ITD');

insert into Schueler
values('13ITK007','Müller','Thomas','t.mueller@fcb.de','13ITK');

insert into Unterricht
values('13ITK','KaRi'),
('13ITD','KaRi'),
('13ITA','MaJo');
```

3. Ändern – UPDATE Anweisung

Mit der UPDATE-Anweisung können vorhandene Daten in Tabellen modifiziert werden. In den ausgewählten Zeilen werden die Werte neu berechnet bzw. zugewiesen und eingetragen.

Syntax

```
UPDATE Tabellen_Name
SET <Attribut1> = <WERT> [, <AttributN> = <WERT>]
[WHERE <Bedingung> ] ;
```

Beispiele: Änderung Datensatz in die Tabelle *Lehrer*

```
update Lehrer set email = 'f.wagner@itbspa.de'
where L_ID = 'WaFl' ;
```

```
update Lehrer set name = 'Flo', email = 'f.wagner@itbspa.de'
where L_ID = 'WaFl' ;
```

4. Löschen – DELETE Anweisung

Mit DELETE FROM werden Datensätze aus einer Datenbanktabelle gelöscht. Die Bedingung gibt an, um welche Datensätze es sich handelt. Fehlt die Bedingung, werden alle Zeilen einer Tabelle gelöscht.

Syntax

```
DELETE FROM Tabellen_Name
[WHERE <Bedingung> ] ;
```

Beispiel: Löschen von Datensätzen aus der Tabelle *Lehrer*

```
delete from Lehrer; --Alle Datensätze werden gelöscht
```

```
delete from Lehrer
```

```
where L_ID = 'WaFl'; --Datensatz vom Lehrer WaFl wird gelöscht
```

5. Nachschlagemöglichkeiten

Weitere Nachschlagemöglichkeiten finden Sie unter folgendem Link:

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

**Aufgabenstellung:**

- **Erzeugen** Sie in der Datenbank der kleinen Schulverwaltung **mindestens** folgende **Datensätze!**
 - **Achten** Sie dabei auf die richtige Reihenfolge der Eintragungen!
 - 10 Schüler:innen
 - 5 Klassen
 - 2 Schüler:innen pro Klasse zuweisen
 - 5 Lehrer
 - Jeder Lehrer hat eine Klassenleitung
 - 10 Unterrichtskombinationen
 - Lehrer unterrichten in mehreren Klassen
 - In einer Klasse unterrichten mehrere Lehrer