

Elementarische Anweisung: **INSERT**

SQL Teil 2



Insert

Syntax:

```
INSERT INTO <tabellenname> (attribut1, attribut2, ...)  
VALUES (value1, value2, ...);
```

attribut: Spalte

value : Werte zuweisen

Beispiel:

```
INSERT INTO autor (id, name, vorname)  
VALUES (4310, "peter", "müller");
```



Elementarische Anweisung: `SELECT`

SQL Teil 3



SELECT

Die erste DB Query

Syntax:

```
SELECT *  
FROM Entitätstyp;
```

Was bedeutet das?

Erkenntnis:

Die Abfrage muss genauer sein.



SELECT

Einzelne Attribute auswählen

Syntax:

```
SELECT atr1, atr2, ...  
FROM Entitätstyp;
```

Erkenntnis:

Es könnte sinnvoll sein, keine doppelten Ergebnisse auszugeben.



SELECT

Keine doppelten Ergebnisse

Syntax:

```
SELECT DISTINCT atr1, atr2, ...  
FROM Entitätstyp;
```

Erkenntnis:

Es wäre sinnvoll eine Bedingung in die Query zu integrieren.



SELECT

Bedingung integrieren

Syntax:

```
SELECT attr1, attr2, ...  
FROM Entitätstyp  
WHERE condition;
```

Was kann alles eine Bedingung sein?

Ähnlich wie in z.B. c#

Hinweis: Nur ein „=“



SELECT

Bedingung - Beispiele Logische Operatoren

Beispiele:

```
attr = "string"
```

```
attr = 1
```

```
attr > 0
```

```
attr = attr2
```

=	ist gleich
!= oder <>	ist ungleich
>	größer als
<	kleiner als
>=	größer gleich
<=	kleiner gleich
BETWEEN	Wert zwischen einer Spanne
LIKE	beinhaltet Muster
IN	Subquery



SELECT

Attribute und Entitätstypen in ``

Syntax:

```
SELECT attr1, attr2, ...  
FROM Entitätstyp;
```

möglich da keine Leerzeichen oder Sonderzeichen

```
attr1 ohne ``
```

nicht möglich da Leerzeichen. Syntax: `name`

```
`at tr1`
```



SELECT mit LIMIT s, r

Syntax:

```
LIMIT s, r
```

s: Mit welchem Eintrag gestartet werden soll (0 ist der Anfang)

r: Wie viele Einträge maximal

Beispiel:

```
SELECT *  
FROM `country`  
LIMIT 1, 3
```



SELECT mit ORDER BY

Syntax:

```
ORDER BY attr (type)
```

attr: Nach welchem Attribut sortiert werden soll

type: Nach welchem Typ sortiert wird

Beispiel:

```
SELECT *  
FROM `country`  
ORDER BY `Population` DESC
```

Hängt immer von dem zu sortierenden Datentyp ab.
Dies funktioniert automatisch.

DESC	Descending	Absteigend
ASC	Ascending	Aufsteigend



Extra Wissen: SELECT mit BETWEEN

Syntax:

```
SELECT *  
FROM `Entitätstyp`  
WHERE `Attribut` BETWEEN wert1 AND wert2;
```

Beispiel:

```
SELECT *  
FROM `city`  
WHERE `Population` BETWEEN 1 AND 100;
```



Extra Wissen: SELECT mit LIKE

<u>Syntax</u>	<u>Beschreibung</u>
<code>WHERE `Attribut` LIKE 'a%'</code>	Wahr wenn Wert ein a an erster Position hat
<code>WHERE `Attribut` LIKE '%a'</code>	Wahr wenn Wert ein a an letzter Position hat
<code>WHERE `Attribut` LIKE '%or%'</code>	Wahr wenn Wert ein or an irgendeiner position hat
<code>WHERE `Attribut` LIKE '_r%'</code>	Wahr wenn Wert ein a an zweiter Position hat
<code>WHERE `Attribut` LIKE 'a_%_%'</code>	Wahr wenn Wert mit einem a anfängt und mindestens drei Charaktere lang ist
<code>WHERE `Attribut` LIKE 'a%o'</code>	Wahr wenn Wert mit a anfängt und mit o aufhört



SELECT mit IN

Syntax:

```
SELECT * FROM Customers  
WHERE `Attribut` IN (<LISTE>);
```

<u>Name</u>	<u>Beispiel</u>
Tabelle	<code>IN (SELECT `Attribut` FROM `Entitätstyp`);</code>
Liste	<code>NOT IN ('Germany', 'France', 'UK');</code>



SELECT

Sonstige Syntax Hinweise

Strings:

```
"" dad ""
```

Beispiel:

```
SELECT * FROM `country`  
WHERE `Name` = ""dad""
```

Logische Operatoren für mehrere Bedingung:

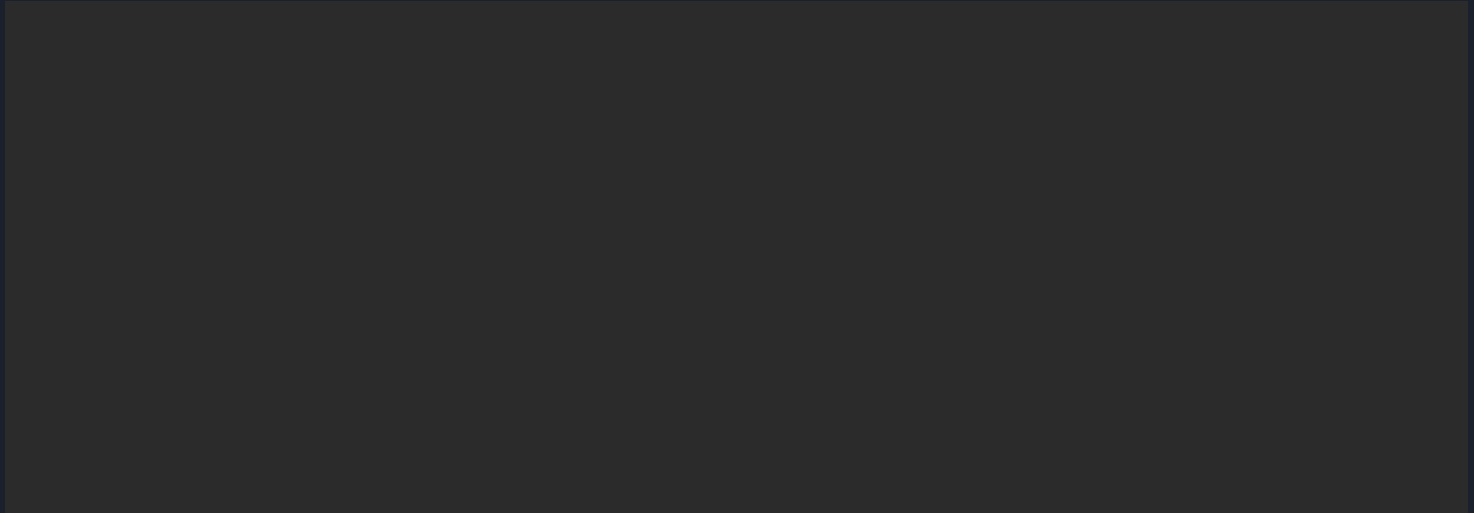
```
AND oder &&  
OR oder ||
```



SELECT

Viel Kleines ergibt ein Großes

Beispiel einer komplexeren Query:





Nächstes Thema: **DELETE, UPDATE / ALTER**

SQL Teil 4

Elementarische
Anweisung: **DELETE**





Delete

Syntax:

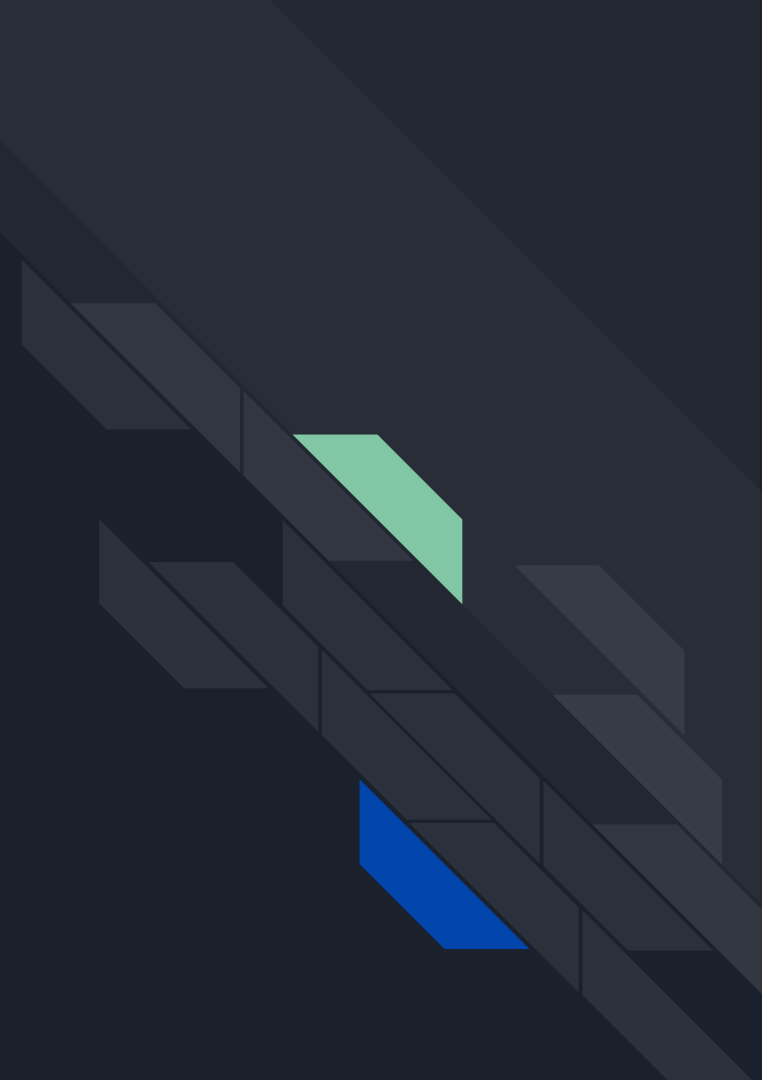
```
DELETE FROM <tabellenname>  
WHERE [bedingung];
```

DELETE : Löscht die ganze Zeile

Beispiel:

```
DELETE FROM `autor`  
WHERE `vorname` = "Heinrich" AND id = 4210;
```

Der Unterschied zwischen UPDATE und ALTER





ALTER / UPDATE

ALTER

- Bearbeitet die tatsächliche Struktur einer Datenbank (z.B. Attributename)
- Löscht, ändert oder fügt Elemente der Struktur hinzu.
- Auch der Datentyp eines Entitätstypen kann verändert werden.

UPDATE

- Ändert den Wert einer Zelle (z.B. den eines sich ändernden Komponenten wie ein Geldbetrag)
- Wählt Zellen anhand einer Bedingungen aus



Update

Syntax:

```
UPDATE <tabellenname>  
SET `attribut1` = value1, `attribut2` = value2, ...  
WHERE [bedingung]
```

`attribut`: Spalte

`value` : Werte aktualisieren

Beispiel:

```
UPDATE `autor`  
SET `name` = "Kramer", `vorname` = "Heinrich", ...  
WHERE `id` = 4311;
```



Alter

Syntax:

```
ALTER <tabellenname>  
ADD `attribut` DATENTYP,  
DROP TABLE/COLUMN name,  
MODIFY COLUMN name DATENTYP;
```

ALTER : Bearbeiten

DROP : Löscht die Spalte

MODIFY : Datentyp eines Attributs verändern

Beispiel:

```
ALTER `autor`  
ADD `telefonnr` INTEGER,  
DROP COLUMN `email`,  
MODIFY COLUMN `hausnr` VARCHAR(255);
```



Danke für eure
Aufmerksamkeit

Lesezeichen Empfehlung: 2, 8, 11, 13, 14, 15, 21, 23