

## Zusammenfassung: (2) SQL-Basics

Tabellen in Relationalen DBS

```
create table
insert
update
delete
select
```

Hinweis zu select:

- Alle Spalten mit \* abrufbar
- Vergleich in where - Klausel mit like und % möglich, z.B. ...where Nachname like 'M%' (findet alle Nachnamen, die mit M beginnen)

Tabellen anlegen mit **CREATE TABLE**  
Tabellen befüllen mit **INSERT INTO**  
Daten aktualisieren mit **UPDATE**  
Daten löschen mit **DELETE / DROP**

Beispiel:

```
CREATE TABLE personen (
  Nachname varchar(255),
  Vorname varchar(255),
  Land varchar(255),
  Geburtsjahr int(4) );
```

Bsp:

```
INSERT INTO personen
(Nachname, Vorname, Land, Geburtsjahr)
values
('Gurion', 'Ben', 'Israel', 1886)
```

Die wichtigsten Datentypen

varchar(255) entspricht einem String mit maximaler Länge von 255 Zeichen  
int(4) stellt eine 4-stellige(!) Integerzahl dar.

Eine Anmerkung zum NULL - Wert:

NULL ist in der Tat UNDEF, nicht etwa numerisch 0 und auch kein leerer String.

```
INSERT INTO personen
(Nachname, Vorname, Land, Geburtsjahr)
values
('Gurion', 'Ben', NULL, 1886)
```

UPDATE TABELLENNAME SET NAME = VALUE WHERE BEDINGUNG

Bsp.:

```
UPDATE personen set Geburtsjahr = 1886 where Nachname='Gurion';
```

Die where - Bedingung selektiert die Zeilen, die geändert werden sollen. Hier ist es nur die erste Zeile mit Nachnamen „Gurion“.

Würde man „where“ weglassen, würde das Geburtsjahr in allen Zeilen geändert werden.

Bemerkungen zur Syntax:

- Strings bitte immer in Hochkommata
- Integers auch, auch wenn es da kein Zwang ist
- Das abschließende ; ist kein SQL, aber u.U. auf der mySQL-Konsole notwendig

Bsp:

```
INSERT INTO personen (Nachname, Vorname, Land, Geburtsjahr)
values ('Gurion', 'Ben', 'Israel', '1886');
```

Löschen von Datensätzen (kompletten Zeilen):

```
DELETE FROM TABELLENNAME where BEDINGUNG
```

Löschen von Tabellen:

```
DROP table personen
```

Beispiel:

```
DELETE FROM PERSONEN where Nachname='Gurion';
```

Auch hier selektiert wieder „where“ die Zeile, die gelöscht werden soll.

```
DROP table PERSONEN;
```

SELECT SPALTENNAME FROM TABELLENNAME WHERE BEDINGUNG

SPALTENNAME wählt **Spalten**  
BEDINGUNG wählt **Zeilen**

```
SELECT Nachname, Geburtsjahr FROM PERSONEN WHERE Geburtsjahr < 1900;
```

Zeigt die Spalten „Nachname“ und „Geburtsjahr“ von allen Zeilen an, in denen das Geburtsjahr kleiner als 1900 ist.

```
select Name, Land from PERSONEN where Geburtsjahr < 1900
```

Ergebnis:

```
Gurion   Israel
Curie    Polen
```

Hinweise:

- Hinter select kann man auch \* schreiben, dann bekommt man alle Spalten angezeigt.
- In Where-Klauseln kann man auch ein '%' als Wildcard nutzen: select ... where Name like 'G%'

## Zusammenfassung: (3) SQL: Schlüssel

Primärschlüssel: Eindeutige Schlüssel

Fremdschlüssel: Schlüssel, die auf Primärschlüssel zeigen

→ Referenzielle Integrität: Fremdschlüssel müssen passenden Primärschlüssel haben

→ Erweiterung der Deklaration von Fremdschlüssels mit  
... on delete CASCADE  
... on update CASCADE

→ Möglichkeit der Verknüpfung von Tabellen bei SELECT  
select Name, Wahljahr from PERSONEN, AMTSZEITEN where ID=PID

Hinweis:

Haben verschiedene Tabellen gleiche Spaltennamen, so muss man z.B. select table1.Name, table2.Name from.. schreiben.

Personen				ID: Primärschlüssel
ID	Nachname	Vorname	Land	Geburtsjahr
1	Gurion	Ben	Israel	1886
2	Nasser	Gamal, Abdel	Ägypten	1918
3	Curie	Marie	Polen	1867
4	Gagarin	Juri	UdSSR	1934

```
create table Personen (
  ID int(10) PRIMARY KEY,
  Nachname varchar(255),...
```

Personen				
ID	Nachname	Vorname	Land	Geburtsjahr
1	Gurion	Ben	Israel	1886
2	Nasser	Gamal, Abdel	Ägypten	1918
3	Curie	Marie	Polen	1867
4	Gagarin	Juri	UdSSR	1934

Amtszeiten		
ID	gewählt	PersonID
10	1948	1
20	1954	2

```
create table Amtszeiten (
  ID int(10),
  gewählt int(10),
  PersonID int(10),
  FOREIGN KEY(PersonID) references Personen(ID)
)
```

```
select Nachname, Land, gewählt from
personen , amtszeiten
where ID = PersonID
```

Ergebnis		
Nachname	Land	gewählt
Gurion	Israel	1948
Nasser	Ägypten	1954