

Die Structured Query Language (SQL) dient zur Steuerung des Database Management- Systems (DBMS). Werden Tabelleninhalte angefordert (mit SELECT), so bekommt der Client Ergebnismengen zurückgeliefert.

Aufgabe :

Wenden Sie nun die ersten Abfragen auf Ihre Tabellen an :

Erstellen Sie Beispiele von kommentierten Abfragen und dokumentieren Sie diese in einem Word- Dokument. Am Ende der Praxisübung ist dem Lehrer ein mit Namen versehenes gedrucktes Exemplar auszuhändigen .

Bei irgendwelchen Unklarheiten schlagen Sie bitte zuerst in der Online- Hilfe Ihres DBMS nach.

1. Einfachste Abfrage (komplette Tabelle)

```
SELECT *  
FROM Tabellename;
```

2. Bestimmte Felder (Spalten) in gewünschter Reihenfolge :

```
SELECT Feldname1, Feldname2, .... etc.  
FROM Tabellename;
```

3. Filterung von Datensätzen (mit Zusatz : WHERE)

```
SELECT Feldname1, Feldname2, .... etc.  
FROM Tabellename  
WHERE Bedingung      ( =, <>, >, >=, <, <= ) // Z.B.: Preis > 10.75
```

4. Darstellung der Datensätze in bestimmter Reihenfolge (alphabetisch, numerisch)

```
SELECT Feldname1, Feldname2, .... etc.  
FROM Tabellename  
WHERE Bedingung      ( =, <>, >, >=, <, <= )  
ORDER BY Feldname1, Feldname2;      (mögl. Zusatz ASC, DESC)  
                                     (ascending = aufsteigend, d.i. von klein nach groß, descending = absteigend)
```

5. Der Abfrage können „virtuelle Spalten“ hinzugefügt und benannt werden (mit AS ...)

```
SELECT Feldname1, Feldname2, Berechneter_Ausdruck AS NeuFeldname  
FROM Tabellename  
WHERE Bedingung      ( =, <>, >, >=, <, <= )
```

```
Z.B. SELECT Anzahl * Preis AS Gesamtwert  
FROM .....
```

6. Den Tabellennamen kann ein Kürzel (= Alias) mit AS zugeordnet werden. Dadurch können die Feldnamen platzsparend vollständig korrekt bezeichnet werden.

```
SELECT M.Feldname1, M.Feldname2  
FROM Tabellename AS M  
WHERE Bedingung      ( =, <>, >, >=, <, <= )
```

7. Um die Wirkung einer Abfrage zu testen, können Tastatureingaben eingebunden werden (= sog. „parametrisierte Abfrage“). Alle SQL- Textabschnitte, die keinen Tabellen- oder Feldnamen bezeichnen, gelten als Parameter. Beispiel :

```
SELECT Feldname1, Feldname2
FROM Tabellennamen
WHERE Name = IrgendeinAusdruck      ( Texteingabe per Tastatur )
```

8. Komplexe Filterung . Im WHERE- Abschnitt der SQL- Anweisung können mit AND, OR und NOT komplexere Filtervorschriften ausgeführt werden.

```
SELECT Feldname1, Feldname2
FROM Tabellennamen
WHERE Name = 'X' AND Alter >= 20
```

9. Spezielle Filterung :

a) Textsuche mit dem LIKE- Operator (findet alle Ausdrücke, in denen ein X vorkommt):

```
WHERE Name LIKE '%X%' {Achtung! MS-Access benutzt statt des %-Zeichens}
                        {das Zeichen : * }
```

b) Abfrage auf Nullmarke :

```
WHERE Name IS NULL      { bzw. IS NOT NULL }
```

c) Filterung durch Festlegung eines zusammenhängenden Bereichs . Der Filterausdruck :

```
WHERE Feldname >= Untergrenze AND Feldname <= Obergrenze
```

kann verkürzt werden durch :

```
WHERE Feldname BETWEEN Untergrenze AND Obergrenze
(möglicher Zusatz: NOT, also: NOT BETWEEN ... AND ... )
```

d) Filterung durch Festlegung einer unzusammenhängenden Liste.

```
WHERE Vorname IN ('Harry', 'Gerda', 'Sibylle')
(möglicher Zusatz: NOT, also: NOT IN (...,... ) )
```

e) Sollten bei einer Filterung unerwünschterweise mehrere komplett gleiche Datensätze auftauchen, bzw. zu erwarten sein, kann hinter dem Schlüsselwort SELECT der DISTINCT-Zusatz angegeben werden (Bedeutung: Zeige nur verschiedene Datensätze), der Doppelungen oder Mehrfachanzeigen unterdrückt.

```
SELECT DISTINCT Feldname1, Feldname2, .... etc.
FROM Tabellennamen
WHERE Bedingung      ( =, <>, >, >=, <, <= )
```

10.) Begrenzung der zurückgegebenen Datensätze mit dem Schlüsselwort : TOP

```
SELECT TOP 3 Feldname1, Feldname2, //die ersten 3 Tupel
FROM Tabellennamen
```