Die SELECT-Anweisung zur Abfrage von Daten

Mit der SELECT-Anweisung können Daten einer oder mehrerer Tabellen flexibel abgefragt werden. In diesem Arbeitsblatt werden vorerst nur Daten aus einer einzigen Tabelle abgefragt.
Die Select-Anweisung besteht aus 3 Teilen:

- Der **Select-Teil** gibt die Spalten an, aus denen Daten dargestellt werden.
- Der **From-Teil** gibt an, welche Tabellen verwendet werden.
- Der **Where-Teil** gibt Bedingungen an, um die gesuchte Datenmenge einzuschränken.

1. Abfragen auf einer einzigen Tabelle ohne where-Teil:

```
SELECT     spalten_name1, spalten_name2,..
FROM       tabellen_name;
```

Beispiele für wichtige einfache Abfragen:

1. Der Name und der Vorname eines Kunden aus der Tabelle Kunde werden ausgegeben.

   `SELECT` Name, Vorname `FROM` kunde;

2. Alle Datensätze mit allen Spalten werden ausgegeben. Das Zeichen '*' steht als Platzhalter für alle Spaltennamen einer Tabelle.

   `SELECT * FROM` artikel;

3. Die Anzahl der Datensätze in der Tabelle wird ausgegeben.

   `SELECT COUNT(*) FROM` artikel;

4. Mit der Option **`DISTINCT`** können Duplikate aus dem Ergebnis entfernt werden.

   `SELECT DISTINCT` stadt `FROM` kunde;

2. Abfragen auf einer einzigen Tabelle mit einem WHERE-Teil:

Die Ergebnismenge wird mit Hilfe von **Bedingungen** (Kriterien)eingeschränkt:

```
SELECT     spalte1, spalte2,..
FROM       tbl_name
WHERE      bedingungen;
```

2.1. Abfragen mit Bedingung

Der folgende Befehl gibt die komplette Zeile aus, bei der in der Spalte ID eine 1 steht.

```
SELECT * FROM tbl_name WHERE ID = 1;
```

Wenn Sie nach einem Wort suchen, müssen Sie dieses in zwei Hochkommas setzen.

```
SELECT * FROM tbl_name WHERE spalte1 = 'wort';
```

Um die Datensätze zwischen 10 und 20 auszugeben können Sie `BETWEEN` verwenden.

```
SELECT * FROM tbl_name WHERE id BETWEEN 10 AND 20;
```

Um alle Datensätze zu finden, deren Spalte `NULL` enthält (bzw. nicht `NULL`)

```
SELECT * FROM tbl_name WHERE spalte1 IS NULL;     (IS NOT NULL)
```

2.2. Abfragen mit Platzhalter

Wenn Sie nicht genau wissen, an welcher Stelle das Wort steht, welches Sie suchen, können Sie den Platzhalter % verwenden. Der Platzhalter % steht für beliebig viele Zeichen.
Weiterhin müssen Sie das = durch LIKE ersetzen.

```
SELECT * FROM tbl_name WHERE spalte1 LIKE '%wort%';
```

Ein weiterer Platzhalter ist der Unterstrich. Dieser steht für genau ein Zeichen.

```
SELECT * FROM tbl_name WHERE spalte1 LIKE 'w__t';
```

2.3. Abfragen mit mehreren Bedingungen (verknüpfte Bedingungen)

Werden zwei Bedingungen mit einem `AND` verknüpft, müssen beide Bedingungen erfüllt werden.

```
SELECT * FROM tbl_name WHERE id > 10 AND id < 20;
```

Werden zwei Bedingungen mit einem `OR` verknüpft, muss mindestens eine Bedingung erfüllt sein.

```
SELECT * FROM tbl_name WHERE id = 10 OR id = 20;
```

Mit der Bedingung `NOT` kann man das Ergebnis einer Bedingung negieren.

```
SELECT * FROM tbl_name WHERE id < 10 AND NOT id = 5;
```

Werden mehrere Bedingungen miteinander verknüpft, müssen gegebenenfalls Teile der Abfrage in Klammern gesetzt werden.

```
SELECT * FROM tbl_name WHERE id < 10 OR (spalte1 = 'wort' AND id = 5);
```

2.4. Abfragen mit Sortierung

Bei der Sortierung wird das Attribut der Sortierung angegeben. Mit ASC und DESC wird angegeben, ob alphabetisch aufsteigend oder absteigend sortiert werden soll. Standardmäßig ist aufsteigend eingestellt.

Hier werden alle Mitglieder aus Kaiserslautern nach dem Namen sortiert, in diesem Fall absteigend (quasi alphabetisch rückwärts).

```
SELECT Vorname, Name, Ort
      FROM Mitglied
      WHERE Ort LIKE 'Kaiserslautern'
      ORDER BY Name DESC;
```