Optionen für SELECT-Statements

```
...
ORDER BY
  ...
```

Sorgt dafür, dass die Datensätze sortiert zurückgegeben werden. `ORDER BY` steht immer am Ende der Abfrage.

Beispiele:

`... ORDER BY SNachname ASC`

Sortiert aufsteigend nach dem Nachnamen. `ASC` steht für „ascending“, man kann `ASC` auch weglassen, da dies die Standard-Sortierreihenfolge ist.

`... ORDER BY SKlasse DESC`

Sortiert absteigend nach der Klasse. `DESC` steht für „descending“.

Man kann auch nach mehreren Spalten sortieren lassen. In diesem Fall wird zuerst nach der ersten angegebenen Spalte sortiert, wenn mehrere Datensätze hier den gleichen Wert haben, werden diese nach der zweiten Spalte sortiert und so weiter:

`... ORDER BY SKlasse DESC, SNachname ASC, SVorname`

Sortiert erst absteigend nach der Klasse. Schüler einer Klasse werden zuerst aufsteigend nach dem Nachnamen sortiert, falls sie den gleichen Nachnamen haben, werden sie aufsteigend nach dem Vornamen sortiert.

```
SELECT DISTINCT
  Spalten
FROM
  ...
```

Sorgt dafür, dass doppelte Datensätze aus dem Ergebnis entfernt werden. `DISTINCT` wird direkt nach `SELECT` eingefügt.

Anmerkung: Beim Herausfiltern der doppelten Ergebnisse werden *alle* Spalten betrachtet, die ausgewählt werden. Die Kombination aus allen Spalten muss eindeutig sein.

Beispiele:

`SELECT DISTINCT SKlasse FROM schueler`

Ergebnis: 5, 6, 7, 8, 9, 10, 11, 12, 13

`SELECT DISTINCT SNachname, SKlasse FROM schueler`

Ergebnis: 30 Datensätze, da zwar die Klasse bei einigen Schülern gleich ist, jedoch die Kombination aus Nachnamen und Klasse jeweils unterschiedlich sind.

```
SELECT
  COUNT(*)
FROM
  ...
```

Liefert die Anzahl der Datensätze zurück.

Die Abfrage kann auch mit einem `WHERE` eingeschränkt werden, in diesem Fall wird die Anzahl der Datensätze zurückgeliefert, auf die die `WHERE`-Bedingung zutrifft.

Beispiele:

`SELECT COUNT(*) FROM schueler`

Ergebnis: Die Anzahl aller Datensätze

`SELECT COUNT(*) FROM schueler WHERE SKlasse <= 7`

Ergebnis: Die Anzahl der Datensätze, bei denen SKlasse kleiner oder gleich 7 ist.

Anstelle von `COUNT(*)` kann man auch den Namen einer Spalte angeben, z.B. `COUNT(SVorname)`, das Ergebnis ändert sich dadurch nicht.

```
SELECT
  MAX(Spalte),
  MIN(Spalte),
  AVG(Spalte),
  SUM(Spalte)
...
```

`MAX` liefert den größten Wert, der in der gewählten Spalte steht.
`MIN` liefert den kleinsten Wert, der in der gewählten Spalte steht.
`AVG` liefert den Mittelwert aller Werte, die in der gewählten Spalte stehen.
`SUM` liefert die Summe aller Werte, die in der gewählten Spalte stehen.

Man kann die Datensätze natürlich noch zusätzlich mit `WHERE` einschränken.

Diesen Aggregationsfunktionen muss man immer eine konkrete Spalte

geben, mit der sie arbeiten können, * funktioniert hier nicht.
Wenn die WHERE-Bedingung auf keinen Datensatz zutrifft, ist das Ergebnis der Funktionen NULL (ein „leerer Wert“, nicht zu verwechseln mit der Zahl 0).
MIN und MAX funktionieren auch mit Textfeldern, in diesem Fall wird der alphabetisch erste bzw. letzte Wert zurückgegeben. AVG und SUM liefern bei Textfeldern die Zahl 0.