

## SESSION #2:

### Die vier Grundbefehle in SQL - "SPICKZETTEL"

| Befehl        | Erklärung   | Beispiele   |
|---------------|---|---|
| <b>INSERT</b> | Befehl, um Daten in die Datenbank einzufügen                                | INSERT INTO `tabellenname`<br>(`spaltenname1`, `spaltenname2`)<br>VALUES ('wert1.1', 'wert2.1'),<br>(`wert1.2` , `wert2.2`)       |
| <b>SELECT</b> | Befehl, um Daten aus der Datenbank anzuzeigen                               |   |
| <b>UPDATE</b> | Befehl, um Daten in der Datenbank zu aktualisieren                          | UPDATE `tabellenname` SET<br>`spaltenname` = `wert` (bearbeitet<br>alle Zeilen der Spalte → beachte die<br>Filterfunktionen s.U.) |
| <b>DELETE</b> | Befehl, um Daten aus der Datenbank zu löschen                               |   |
| FROM          | Spezifiziert die Datenquelle  | SELECT * FROM `tabellenname`<br><br>SELECT `spaltenname`,<br>`spaltenname2` FROM<br>`tabellenname`                                |
| *             | Platzhalter für "Alles" - kann z.B. durch einen Spaltennamen ersetzt werden |   |
| ,             | Trennzeichen für die Abfrage mehrerer Felder oder mehrerer Befehle          | SELECT `spaltenname1`,<br>`spaltenname2` FROM<br>`tabellenname`   |
| AS            | Definiert den Namen der Spalte in der Ausgabe                               | SELECT `spaltenname` AS<br>beliebiger_Name FROM<br>`tabellenname`   |
| WHERE         | Filtert die Werte   | SELECT * FROM `tabellenname`<br>WHERE `spaltenname` = `wert`<br>oder:   |

|          |  |  |
|----------|--|--|
|          |  | DELETE FROM `tabellenname`<br>WHERE `spaltenname` = `wert`                                     |
| WHERE IN | Filtert nach mehreren Werten gleichzeitig  | SELECT * FROM `tabellenname`<br>WHERE `spaltenname` IN ('wert1',<br>'wert2', 'wert3', 'wert4') |
| LIKE     | Filtert alle Einträge, wo ein bestimmter Wert (z.B. ein Wort) vorkommt           | SELECT `spaltenname` FROM<br>`tabellenname` WHERE<br>`spaltenname` LIKE `wert`                 |
| NOT      | Zeigt alle Einträge bis auf die, wo ein bestimmter Wert (z.B. ein Wort) vorkommt | SELECT `spaltenname` FROM<br>`tabellenname` WHERE<br>`spaltenname` NOT LIKE `wert`             |
| %        | Platzhalter für einen Wert: beliebiges kann folgen                               | SELECT `spaltenname` FROM<br>`tabellenname` WHERE<br>`spaltenname` LIKE `w%`                   |
| _        | Platzhalter für genau 1 Zeichen  | SELECT `spaltenname` FROM<br>`tabellenname` WHERE<br>`spaltenname` LIKE `w_rt`                 |
| COUNT    | Funktion: Zählt die Anzahl der Einträge ("Wie viele Zeilen gibt es?")            | SELECT COUNT (*) FROM<br>`tabellenname`  |
| MIN      | Funktion: Gibt den kleinsten Wert (Minimum) zurück                               | SELECT MIN (`spaltenname`)<br>FROM `tabellenname`  |
| MAX      | Funktion: Gibt den größten Wert (Maximum) zurück                                 | SELECT MAX (`spaltenname`)<br>FROM `tabellenname`  |
| AVG      | Funktion: Gibt den Mittelwert (Average) zurück                                   | SELECT AVG (`spaltenname`)<br>FROM `tabellenname`  |
| SUM      | Funktion: Gibt die Summe zurück  | SELECT SUM (`spaltenname`)<br>FROM `tabellenname`  |
| AND      | Logischer Operator: Prüft, ob Bedingung A und Bedingung B zutreffen              | SELECT * FROM `tabellenname`<br>WHERE `spaltenname` > 100 AND<br>`spaltenname` < 500           |

|          |  |   |
|----------|--|---|
| OR       | Logischer Operator: Prüft, ob Bedingung A oder Bedingung B zutrifft  | SELECT * FROM `tabellenname`<br>WHERE `spaltenname` > 100 OR<br>`spaltenname` < 500   |
| DISTINCT | Prüft, wie viele unterschiedliche Einträge es in den Spalten gibt bzw. entfernt Dubletten                            | SELECT DISTINCT `spaltenname`<br>FROM `tabellenname`<br>Oder zB SELECT COUNT<br>(DISTINCT `spaltenname`) FROM<br>`tabellenname`   |
| GROUP BY | Gruppirt die Ausgabe   | SELECT `spaltenname` FROM<br>`tabellenname` GROUP BY<br>`spaltenname`<br>oder z.B. SELECT `spaltenname`,<br>COUNT(*) AS anzahl FROM<br>`tabellenname` WHERE<br>`spaltenname2` < 5 GROUP BY<br>`spaltenname` |
| HAVING   | Filtert die Ausgabe nach erfolgten Berechnungen/ Gruppierungen   | SELECT `spaltenname`, COUNT(*)<br>AS name FROM `tabellenname`<br>WHERE `spaltenname` > 5 GROUP<br>BY `spaltenname` HAVING name ><br>2;  |
| ORDER BY | Ordnet die Werte (z.B. aufsteigend (ASC) oder absteigend (DESC))   | SELECT * FROM `tabellenname`<br>ORDER BY `spaltenname` ASC,<br>`spaltenname2` ASC   |
| LIMIT    | Limitiert die Anzahl der Einträge (vgl. WHERE; "Wo soll die Ausgabe starten und wo enden?")                          | SELECT * FROM `tabellenname`<br>ORDER BY `spaltenname` ASC<br>LIMIT 2, 4 ("Überspringe 2 Zeilen<br>und gib die nächsten vier Zeilen<br>aus")  |
| --       | Kennzeichnet einzeilige Kommentare<br>→ alles was danach in der Zeile steht<br>wird nicht interpretiert              |   |
| /*...*/  | Kennzeichnet meist mehrzeilige<br>Kommentare → alles was zwischen<br>"/*" und "*/" steht wird nicht<br>interpretiert |   |

|  |  |  |
|--|--|--|
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Anmerkung: Die Befehle können nahezu beliebig kombiniert werden

**Konvention:**

- Befehle groß schreiben, um direkt zu sehen, dass es sich um einen Befehl handelt, Namen von Tabellen etc. klein,...
- `...` für Feldbezeichnungen (Name einer Tabelle, Spalte ,...);
- '...' für Werte (nur erforderlich bei Text);
- → Konventionen können, müssen in der Regel aber nicht eingehalten werden. Sie erleichtern das Lesen des Codes für einen selbst und andere.
  - "where spaltenname = wert" ist dasselbe wie "WHERE `spaltenname` = 'wert'"