

SESSION #1:

Einführung in Datenbanken und SQL - "SPICKZETTEL"

Befehl	Erklärung	Beispiele
SELECT	Befehl, um Daten aus der Datenbank anzuzeigen	
FROM	Spezifiziert die Datenquelle	SELECT * FROM `tabellenname` SELECT `spaltenname`, `spaltenname2` FROM `tabellenname`
*	Platzhalter für "Alles" - kann z.B. durch einen Spaltennamen ersetzt werden	
,	Trennzeichen für die Abfrage mehrerer Felder oder mehrerer Befehle	SELECT `spaltenname1`, `spaltenname2` FROM `tabellenname`
AS	Definiert den Namen der Spalte in der Ausgabe	SELECT `spaltenname` AS beliebiger_Name FROM `tabellenname`
WHERE	Filtert die Werte	SELECT * FROM `tabellenname` WHERE `spaltenname` = 'wert'
WHERE IN	Filtert nach mehreren Werten gleichzeitig	SELECT * FROM `tabellenname` WHERE `spaltenname` IN ('wert1', 'wert2', 'wert3', 'wert4')
LIKE	Filtert alle Einträge, wo ein bestimmter Wert (z.B. ein Wort) vorkommt	SELECT `spaltenname` FROM `tabellenname` WHERE `spaltenname` LIKE 'wert'
NOT	Zeigt alle Einträge bis auf die, wo ein bestimmter Wert (z.B. ein Wort) vorkommt	SELECT `spaltenname` FROM `tabellenname` WHERE `spaltenname` NOT LIKE 'wert'

%	Platzhalter für einen Wert: beliebiges kann folgen	SELECT `spaltenname` FROM `tabellenname` WHERE `spaltenname` LIKE 'w%'
_	Platzhalter für genau 1 Zeichen	SELECT `spaltenname` FROM `tabellenname` WHERE `spaltenname` LIKE 'w_rt'
COUNT	Funktion: Zählt die Anzahl der Einträge ("Wie viele Zeilen gibt es?")	SELECT COUNT (*) FROM `tabellenname`
MIN	Funktion: Gibt den kleinsten Wert (Minimum) zurück	SELECT MIN (`spaltenname`) FROM `tabellenname`
MAX	Funktion: Gibt den größten Wert (Maximum) zurück	SELECT MAX (`spaltenname`) FROM `tabellenname`
AVG	Funktion: Gibt den Mittelwert (Average) zurück	SELECT AVG (`spaltenname`) FROM `tabellenname`
SUM	Funktion: Gibt die Summe zurück	SELECT SUM (`spaltenname`) FROM `tabellenname`
AND	Logischer Operator: Prüft, ob Bedingung A und Bedingung B zutreffen	SELECT * FROM `tabellenname` WHERE `spaltenname` > 100 AND `spaltenname` < 500
OR	Logischer Operator: Prüft, ob Bedingung A oder Bedingung B zutrifft	SELECT * FROM `tabellenname` WHERE `spaltenname` > 100 OR `spaltenname` < 500
DISTINCT	Prüft, wie viele unterschiedliche Einträge es in den Spalten gibt bzw. entfernt Dubletten	SELECT DISTINCT `spaltenname` FROM `tabellenname` Oder zB SELECT COUNT (DISTINCT `spaltenname`) FROM `tabellenname`
GROUP BY	Gruppiert die Ausgabe	SELECT `spaltenname` FROM `tabellenname` GROUP BY `spaltenname` oder z.B. SELECT `spaltenname`, COUNT(*) AS anzahl FROM `tabellenname` WHERE `spaltenname2` < 5 GROUP BY `spaltenname`

HAVING	Filtert die Ausgabe nach erfolgten Berechnungen/ Gruppierungen	SELECT `spaltenname`, COUNT(*) AS name FROM `tabellenname` WHERE `spaltenname` > 5 GROUP BY `spaltenname` HAVING name > 2;
ORDER BY	Ordnet die Werte (z.B. aufsteigend (ASC) oder absteigend (DESC))	SELECT * FROM `tabellenname` ORDER BY `spaltenname` ASC, `spaltenname2` ASC
LIMIT	Limitiert die Anzahl der Einträge (vgl. WHERE; "Wo soll die Ausgabe starten und wo enden?")	SELECT * FROM `tabellenname` ORDER BY `spaltenname` ASC LIMIT 2, 4 ("Überspringe 2 Zeilen und gib die nächsten vier Zeilen aus")
--	Kennzeichnet einzeilige Kommentare → alles was danach in der Zeile steht wird nicht interpretiert	
/*...*/	Kennzeichnet meist mehrzeilige Kommentare → alles was zwischen "/*" und "*/" steht wird nicht 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'"