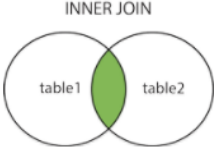
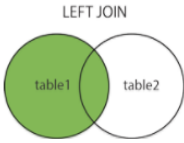
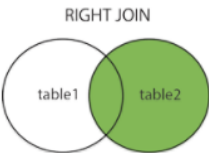
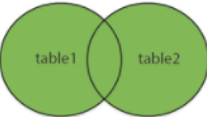


SESSION #3: SQL-Abfragen mit JOINS - "SPICKZETTEL"

Befehl	Erklärung	Beispiele
INSERT	Befehl, um Daten in die Datenbank einzufügen	INSERT INTO `tabellenname` (`spaltenname1`, `spaltenname2`) VALUES ('wert1.1', 'wert2.1'), (`wert1.2` , `wert2.2`)
SELECT	Befehl, um Daten aus der Datenbank anzuzeigen	
UPDATE	Befehl, um Daten in der Datenbank zu aktualisieren	UPDATE `tabellenname` SET `spaltenname` = `wert` (bearbeitet alle Zeilen der Spalte → beachte die Filterfunktionen s.U.)
DELETE	Befehl, um Daten aus der Datenbank zu löschen	
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` oder:

		DELETE FROM `tabellenname` WHERE `spaltenname` = `wert`
WHERE IN	Filtert nach mehreren Werten gleichzeitig	SELECT * FROM `tabellenname` WHERE `spaltenname` IN ('wert1', 'wert2', 'wert3', 'wert4')
BETWEEN	Filtert Einträge, die zwischen zwei Werten liegen	SELECT * FROM `tabellenname` WHERE `spaltenname` BETWEEN 'wert1' AND 'wert2';
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`
INNER JOIN 	Verknüpft zwei Tabellen und zeigt die Werte an, die in beiden Tabellen vorkommen Info: "INNER JOIN" ist die ausführliche Schreibweise, nur "JOIN" reicht aber auch :)	SELECT `tabellenname1`.`spaltenname1`, `tabellenname2`.`spaltenname2`, `tabellenname1`.`spaltenname3` FROM `tabellenname1` INNER JOIN `tabellenname2`
ON	Filtert die Werte. Im Gegensatz zu WHERE werden die Werte direkt gefiltert, während die Tabellen verknüpft werden und nicht erst nachträglich	SELECT `tabellenname1`.`spaltenname1`, `tabellenname2`.`spaltenname2` FROM `tabellenname1` INNER JOIN `tabellenname2` ON `tabellenname1`.`spaltenname_FremdID` = `tabellenname2`.`spaltenname_PrimärID`;
LEFT JOIN 	Verknüpft zwei Tabellen und zeigt alle Werte an, die in Tabelle 1 vorkommen und die entsprechenden Werte aus Tabelle 2 oder NULL	SELECT `tabellenname1`.`spaltenname1`, `tabellenname2`.`spaltenname2` FROM `tabellenname1` LEFT JOIN `tabellenname2` ON `tabellenname1`.`spaltenname_FremdID` = `tabellenname2`.`spaltenname_PrimärID`
RIGHT JOIN 	Verknüpft zwei Tabellen und zeigt alle Werte an, die in Tabelle 2 vorkommen und die entsprechenden Werte aus Tabelle 1 oder NULL ACHTUNG! Fehler auf w3schools. Anfragen funktionieren nur unter diesem Link . Mehr dazu auf den Lernkarten.	SELECT `tabellenname1`.`spaltenname1`, `tabellenname2`.`spaltenname2` FROM `tabellenname1` RIGHT JOIN `tabellenname2` ON `tabellenname1`.`spaltenname_FremdID` = `tabellenname2`.`spaltenname_PrimärID`
FULL JOIN	Verknüpft zwei Tabellen und zeigt alle	SELECT

<p>FULL OUTER JOIN</p> 	<p>Werte aus Tabelle 1 und Tabelle 2 an. (Kreuzprodukt)</p> <p>ACHTUNG! Fehler auf w3schools. Mehr dazu auf den Lernkarten.</p>	<pre>`tabellenname1`.`spaltenname1`, `tabellenname2`.`spaltenname2` FROM `tabellenname1` FULL OUTER JOIN `tabellenname2` ON `tabellenname1`.`spalte_PrimaryID` =`tabellenname2`.`spalte_FremdID`</pre>
<p>GROUP BY</p>	<p>Gruppiert die Ausgabe</p>	<pre>SELECT `spaltenname` FROM `tabellenname` GROUP BY `spaltenname` oder z.B. SELECT `spaltenname`, COUNT(*) AS anzahl FROM `tabellenname` WHERE `spaltenname2` < 5 GROUP BY `spaltenname`</pre>
<p>HAVING</p>	<p>Filtert die Ausgabe nach erfolgten Berechnungen/ Gruppierungen</p>	<pre>SELECT `spaltenname`, COUNT(*) AS name FROM `tabellenname` WHERE `spaltenname` > 5 GROUP BY `spaltenname` HAVING name > 2;</pre>
<p>ORDER BY</p>	<p>Ordnet die Werte (z.B. aufsteigend (ASC) oder absteigend (DESC))</p>	<pre>SELECT * FROM `tabellenname` ORDER BY `spaltenname` ASC, `spaltenname2` ASC</pre>
<p>LIMIT</p>	<p>Limitiert die Anzahl der Einträge (vgl. WHERE; "Wo soll die Ausgabe starten und wo enden?")</p>	<pre>SELECT * FROM `tabellenname` ORDER BY `spaltenname` ASC LIMIT 2, 4 ("Überspringe 2 Zeilen und gib die nächsten vier Zeilen aus")</pre>
<p>--</p>	<p>Kennzeichnet einzeilige Kommentare → alles was danach in der Zeile steht wird nicht interpretiert</p>	
<p>/*...*/</p>	<p>Kennzeichnet meist mehrzeilige Kommentare → alles was zwischen "/*" und "*/" steht wird nicht interpretiert</p>	

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’”