

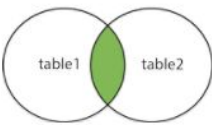
SESSION #4:

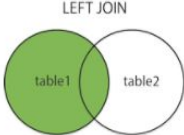
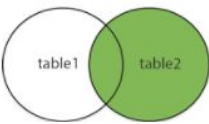
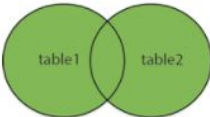
Erstellen einer Datenbank - "SPICKZETTEL"

Befehl	Erklärung	Beispiele
CREATE TABLE	Erstellt eine Tabelle in der Datenbank	CREATE TABLE `tabellenname` (`spaltenname1` datentyp, `spaltenname2` datentyp)
ALTER TABLE (...) ADD	Erweitert die Tabelle	ALTER TABLE `tabellenname` ADD `spaltenname1` datentyp
integer	Speichern von ganzen Zahlen	25
float	Speichern von Kommazahlen	25,75
currency	Speichern von Geldbeträgen	25,75€
boolean	Speichern von Wahrheitswerten	TRUE, FALSE
timestamp	Speichern von Datum- & Zeitangaben	(JJJJ.MM.TT HH:MM:SS)
date	Speichern von Datumsangaben	(JJJJ.MM.TT)
time	Speichern von Zeitangaben	(HH:MM:SS)
varchar (n)	Speichern von Zeichenketten (n Zeichen)	Norbert Schmidt 2
longtext	Speichern von Zeichenketten (bis zu 4,294,967,295 Zeichen)	1. Es war einmal ein kleines süßes Mädchen, das hatte jedermann lieb, der sie nur ansah, am allerliebsten a
DROP TABLE	Löscht eine Tabelle aus der Datenbank	DROP TABLE `tabellenname`
PRIMARY KEY	Setzt eine Spalte als Primärschlüssel	CREATE TABLE `tabellenname` (`spaltenname1` datentyp PRIMARY KEY, `spaltenname2` datentyp)
AUTO- INCREMENT	Definiert, dass die Spalte auto-incremented wird	CREATE TABLE `tabellenname` (`spaltenname1` datentyp

		AUTOINCREMENT, `spaltenname2` datentyp)
NOT NULL	Definiert, dass die Zellen dieser Spalte nicht leer sein dürfen	CREATE TABLE `tabellenname` (`spaltenname1` datentyp, `spaltenname2` datentyp NOT NULL)
UNIQUE	Definiert, dass in zwei Zellen derselben Spalte niemals derselbe Wert stehen darf	CREATE TABLE `tabellenname` (`spaltenname1` datentyp, `spaltenname2` datentyp, CONSTRAINT UC_spaltenname2 UNIQUE (`spaltenname2`))
DEFAULT	Definiert, dass einer Spalte ein Standardwert zugewiesen wird, wenn keiner angegeben wurde	CREATE TABLE `tabellenname` (`spaltenname1` datentyp, `spaltenname2` datentyp DEFAULT 'wert')
CHECK	Definiert, welche Werte in die Tabelle eingetragen werden können	ALTER TABLE `tabellenname` ADD `spaltenname1` datentyp CHECK (`spaltenname1` <='wert1')
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 mehrere 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_Fre

		mdID`=`tabellenname2`.`spaltenname_PrimärID`;
<p>LEFT JOIN</p> 	Verknüpft zwei Tabellen und zeigt alle Werte an, die in Tabelle 1 vorkommen und die entsprechenden Werte aus Tabelle 2 oder NULL	<pre>SELECT `tabellenname1`.`spaltenname1`, `tabellenname2`.`spaltenname2` FROM `tabellenname1` LEFT JOIN `tabellenname2` ON `tabellenname1`.`spalte_FremdID` = `tabellenname2`.`spalte_PrimärID`</pre>
<p>RIGHT JOIN</p> 	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.	<pre>SELECT `tabellenname1`.`spaltenname1`, `tabellenname2`.`spaltenname2` FROM `tabellenname1` RIGHT JOIN `tabellenname2` ON `tabellenname1`.`spalte_FremdID` = `tabellenname2`.`spalte_PrimärID`</pre>
<p>FULL JOIN</p> 	Verknüpft zwei Tabellen und zeigt alle Werte aus Tabelle 1 und Tabelle 2 an. (Kreuzprodukt) ACHTUNG! Fehler auf w3schools. Mehr dazu auf den Lernkarten.	<pre>SELECT `tabellenname1`.`spaltenname1`, `tabellenname2`.`spaltenname2` FROM `tabellenname1` FULL OUTER JOIN `tabellenname2` ON `tabellenname1`.`spalte_PrimärID` = `tabellenname2`.`spalte_FremdID`</pre>
GROUP BY	Gruppiert die Ausgabe	<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>
HAVING	Filtert die Ausgabe nach erfolgten Berechnungen/ Gruppierungen	<pre>SELECT `spaltenname`, COUNT(*) AS name FROM `tabellenname` WHERE `spaltenname` > 5 GROUP BY `spaltenname` HAVING name > 2;</pre>

- 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'"