# Database

* SQL => structured query language
* DBMS => database management system
* RDBMS => relational database management system (MYSQL)

## MYSQL

* DDL -> data definition language
* CREATE DATABASE `databaseName`
* DROP DATABASE `databaseName`
* CREATE TABLE `tableName` (  
  `columnName1` Datatype Constraints Extras,

`columnName2` Datatype Constraints Extras  
)

* DROP TABLE `tableName`
* ALTER TABLE `tableName` ADD `columnName` Datatype constraints extras
* ALTER TABLE `tableName` CHANGE `columnName` `newColumnName` Datatype Constraints Extras
* ALTER TABLE `tableName` DROP `columnName`
* DML => data manipulation language
* INSERT INTO `tableName` (`columnName1`,`columnName4`) VALUES (‘value1’,’value4’) , ()  
  INSERT INTO `tableName` VALUES (DEFAULT,’value1’,value2,…)
* UPDATE `tableName` SET `columnName1` = ‘value1’ , `columnName2` = ‘value2’  
  UPDATE `tableName` SET `columnName1` = ‘value1’ , `columnName2` = ‘value2’ WHERE condition
* DELETE FROM `tableName`
* DELETE FROM `tableName` WHERE condtion
* DQL => data query language
* SELECT `columnName1`,`columnName2` AS `newColumnName` FROM `tableName`
* SELECT \* FROM `tableName` WHERE condition
* Operators (logical , comparison , arithmetic)
* WHERE `columnName` BETWEEN miniNumber AND maxNumber
* WHERE `columnName` IN(value1,value2,…)
* WHERE `columnName` IS || IS NOT NULL
* Aggregate functions (MIN,MAX,SUM,AVG,COUNT,GROUP\_CONCAT)
* GROUP BY `columnName`
* ORDER BY `columnName` ASC||DESC , `columnName2` ASC||DESC
* HAVING condition
* LIMIT value,offset
* (INNER-LEFT-RIGHT) JOIN
* Read About (GROUP\_CONCAT,CONCAT,IF(),ROUND(),Difference between HAVING & where, LIMIT value,offset ,UNION , UNION ALL ,subquery, VIEW (difference between view & table), WHERE LIKE)
* Examples
* # SELECT \* FROM `users` WHERE `email` = 'asmaa@gmail.com' AND `password` = "asmaa@123456789"
* # SELECT \* FROM `users` WHERE `id` < 5 OR `id` > 7
* # SELECT \* FROM `users` WHERE (`id` >= 1 AND `id` < 5) OR (`id` > 7 AND `id` <= 11)
* # SELECT \* FROM `users` WHERE !(`id` >= 5 AND `id` <= 7)
* # SELECT \* FROM `users` WHERE `id` BETWEEN 5 AND 7;
* # SELECT \* FROM `users` WHERE `id` IN(1,4,7,9);
* # SELECT \* FROM `users` WHERE `verification\_code` IS NOT NULL
* # SELECT `id`,`first\_name`,`last\_name`,`salary` ,`bonus`, (`salary`+`bonus`) \* 0.9 AS `total\_salary` FROM `users`
* /\* SELECT

COUNT(`id`) AS `users`,

COUNT(`verification\_code`) AS `users\_has\_code`,

COUNT(`id`) - COUNT(`verification\_code`) AS `users\_hasn't\_code`

FROM `users` \*/

* /\* SELECT

MAX(`salary`) AS `highest\_salary`,

MIN(`salary`) AS `lowest\_salary`,

#AVG(`salary`) AS `average\_salary`

SUM(`salary`) / COUNT(`id`) AS `average\_salary`

FROM

`users`

* /\* SELECT `gender` , COUNT(`id`) AS `number\_of\_users`

FROM `users`

GROUP BY `gender` \*/

* /\*SELECT `gender` , MAX(`salary`) AS `max\_salary`

FROM `users`

GROUP BY `gender`\*/

* /\*SELECT `last\_name` , COUNT(`id`) AS `count\_of\_childs`

FROM `users`

GROUP BY `last\_name`

HAVING `count\_of\_childs` > 1

ORDER BY `count\_of\_childs` DESC , `last\_name` DESC\*/

* #SELECT \* FROM `users` ORDER BY `created\_at` DESC LIMIT 1
* SELECT

`products`.\*,

AVG(`reviews`.`rate`) AS `avg\_rate`,

COUNT(`reviews`.`product\_id`) AS `total\_rates`

FROM `products`

LEFT JOIN `reviews`

ON `products`.`id` = `reviews`.`product\_id`

GROUP BY `products`.`id`Z

* SELECT

`users`.`first\_name`,

`users`.`last\_name`,

`reviews`.`rate`,

`reviews`.`comment`,

`products`.`name\_en`

FROM `users`

JOIN `reviews`

ON `users`.`id` = `reviews`.`user\_id`

JOIN `products`

ON `products`.`id` = `reviews`.`product\_id`

* SELECT

`categories`.`name\_en` AS `category\_name\_en`,

`subcategories`.`name\_en` AS `subcategory\_name\_en`

FROM

`categories`

INNER JOIN `subcategories` ON `categories`.`id` = `subcategories`.`category\_id`

* SELECT

`products`.\*,

`brands`.`name\_en` AS `brand\_name\_en`,

`subcategories`.`name\_en` AS `subcategory\_name\_en`,

`categories`.`id` AS `category\_id`,

`categories`.`name\_en` AS `category\_name\_en`

FROM

`products`

LEFT JOIN `brands`

ON `brands`.`id` = `products`.`brand\_id`

LEFT JOIN `subcategories`

ON `subcategories`.`id` = `products`.`subcategory\_id`

LEFT JOIN `categories`

ON `categories`.`id` = `subcategories`.`category\_id`

* Ex::

**SELECT** `tableName`.`columnName` , aggregate() AS `newName`

**FROM** `tableName`

**JOIN** `tableName1**`**

**ON** `forgeinKey` = `primaryKey`

**WHERE** condition

**GROUP BY** `columnName`

**HAVING** condition

**ORDER BY** `tableName`.`columnName` ASC | DESC , `tableName`.`columnName1` ASC | DESC

**LIMIT** value