-SQL => structured query language

-DBMS => database management system

-RDBMS => relational database management system

# 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 `newColumnName` datatype constraints extras
* ALTER TABLE `tableName` CHANGE `oldColumnName` `newColumnName` datatype constraints extras
* ALTER TABLE `tableName` DROP `columnName`
* ALTER TABLE `childTableName` ADD CONSTRAINT `constraintName` FOREIGN KEY(`FKColumn`) REFERENCES `parentTableName`(`primaryKey`) ON DELETE RESTRICT ON UPDATE CASCADE
* TRUNCATE TABLE `tableName`

🡺 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 `columnName` AS `newColumnName` ,`columnName2` FROM `tableName`
* SELECT \* FROM `tableName`
* SELECT \* FROM `tableName` WHERE condition (comparison operators , logical operators)
* WHERE `columnName` BETWEEN mini AND max
* WHERE `columnName` IN(value1,value2,…)
* Arthematic operators (+,-,\*,/,%)
* WHERE `columnName` IS|IS NOT NULL
* WHERE `columnName` LIKE ‘\_%’
* Query UNION Query
* Aggregates Functions (MAX,MIN,COUNT,SUM,AVG,GROUP\_CONCATE)
* GROUP BY `columnName`
* Search on : GROUP\_CONCATE , bulitin function mysql , difference between having & where , LIMIT & offset , difference between union & union all , difference between table & view
* **SELECT** `tableName`.`columnName` , AggregateFunc(`tableName`.`columnName`) AS `new`

**FROM** `tableName`

**JOIN `**tableName**`**

**ON `**parent**`.`**primary**` =** `child`.`foreign`

**WHERE condition**

**GROUP BY `**tableName**`.`**columnName**`**

**HAVING condition**

**ORDER BY** `columnName` ASC|DESC , `columnName2` ASC|DESC

**LIMIT value,offset**

🡺 Examples

* # SELECT \* FROM `users` WHERE (`id` BETWEEN 1 and 3) or ( `id` BETWEEN 7 and 9)
* # SELECT \* FROM `users` WHERE `id` < 4 OR `id` > 6
* # SELECT \* FROM `users` WHERE `id` NOT BETWEEN 4 AND 6
* # SELECT \* FROM `users` WHERE `id` IN(1,2,4,5,7,8,9)
* # SELECT \* FROM `users` WHERE `id` NOT IN(3,6)
* # SELECT \* FROM `users` WHERE `id` = 1 OR `id` = 2
* # SELECT CONCAT(`first\_name`," ",`last\_name`) AS `full\_name` , `email` AS `user\_email`,`gender` FROM `users` WHERE `gender` = 'm'
* SELECT

CONCAT(`first\_name`, " ", `last\_name`) AS `full\_name`,

`salary` + `bonus` AS `basic\_salary`,

(`salary` + `bonus`) \* 0.1 AS `insurance`,

(`salary` + `bonus`) \* 0.14 AS `tax`,

(`salary` + `bonus`) \* 0.76 AS `net\_salary`

FROM

`users`

UNION

SELECT

'#' AS `total`,

SUM(`salary` + `bonus`) AS `basic\_salary`,

SUM((`salary` + `bonus`) \* 0.1 ) AS `insurance`,

SUM((`salary` + `bonus`) \* 0.14) AS `tax`,

SUM((`salary` + `bonus`) \* 0.76) AS `net\_salary`

FROM

`users`

* SELECT

COUNT(`id`) AS `users\_count`,

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

COUNT(`id`) - COUNT(`verification\_code`) AS `users\_has\_not\_code`

FROM

`users`

* SELECT `phone` FROM `users` WHERE `verification\_code` IS NOT NULL
* SELECT \* FROM `users` WHERE `first\_name` LIKE 'a\_%'
* SELECT `status` , COUNT(`id`) AS `users\_count`

FROM `users`

GROUP BY `status`

* SELECT

`gender`,

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

MIN(`salary`) AS `mini\_salary`

FROM

`users`

GROUP BY

`gender`

* SELECT `status` , COUNT(`id`) AS `users\_count`

FROM `users`

GROUP BY `status`

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

`products`.\*,

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

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

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

FROM `products`

LEFT JOIN `subcategories`

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

LEFT JOIN `categories`

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

LEFT JOIN `brands`

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

* SELECT
* CONCAT(`users`.`first\_name` , ' ' , `users`.`last\_name` ) AS `full\_name`,

COUNT(`carts`.`product\_id`) AS `cart\_count`

FROM `users`

LEFT JOIN `carts`

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

GROUP BY `users`.`id`

ORDER BY `cart\_count` DESC , `full\_name` ASC