# Queries

1. CREATE DATABASE `databaseName` (CREATE DATABASE IF NOT EXISTS `dataBaseName` )
2. DROP DATABASE `databaseName` (DROP DATABASE IF EXISTS `databaseName`)
3. CREATE TABLE `tableName` (

`columnName1` datatype constaints extra,

`columnName2` datatype constaints extra

)

1. DROP TABLE `tableName`
2. ALTER TABLE `tableName` ADD `columnName` datatype constaints
3. ALTER TABLE `tableName` DROP `columnName`
4. ALTER TABLE `tableName` CHANGE `columnName` `new\_columnName` datatype constraints
5. ALTER TABLE `tableName` ADD CONSTRAINT `constraintName` FOREIN KEY(`foreinKeyName`)   
   REFERENCES `tableName`(`primaryKeyName`) ON DELETE CASCADE ON UPDATE SET NULL
6. INSERT INTO `tableName` VALUES (‘’,’’,,DEFAULT) , (‘’,’’,,DEFAULT)
7. INSERT INTO `tableName` (`columnName1`,…) VALUES (‘’,….) , (‘’,….)
8. DELETE FROM `tableName`
9. DELEtE FROM `tableName` WHERE condition
10. UPDATE `tableName` SET `columnName` = ‘value1’ , `columnName2` = ‘value2’
11. UPDATE `tableName` SET `columnName` = ‘value1’ , `columnName2` = ‘value2’ WHERE conditon

# DAY-2

1. SELECT \* FROM `tableName`
2. SELECT `tableName`.`columnName1` AS `newName` , `tableName`.`columnName2` FROM `tableName`
3. SELECT \* FROM `tableName` WHERE condition
4. Operators (arthimatic , logical , comparison )
5. # single line comment , /\* \*/ multi line
6. WHERE `columnName` BETWEEN min AND max
7. WHERE `columnName` IN(‘vlaue’,’value2’,’’)
8. WHERE `columnName` IS NULL || IS NOT NULL
9. SELECT DISTINCT `columnName` FROM `tableName`
10. Aggregates Funcs (SUM,MIN,MAX,COUNT,GROUP\_CONCAT)
11. GROUP BY `columnName`
12. ORDER BY `columnName` ASC || DESC , `columnName1` ASC || DESC
13. HAVING condition => virtual column
14. LEFT JOIN , JOIN , RIGHT JOIN  
    ON `parentTable`.`primaryKey` = `childTable`.`forienKey`
15. WHERE `columnName` LIKE ‘%a\_’

% => 0 , one , multiple characters

\_ => one missed character

1. CREATE VIEW `viewName` AS ( Query )
2. DROP VIEW `viewName`
3. Search on ‘UNION’
4. 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

## Examples

# SELECT \* FROM `users` WHERE `id` > 5 OR `id` < 3

# SELECT \* FROM `users` WHERE `bonus` >= 2 AND `bonus` <=5

# SELECT \* FROM `users` WHERE `bonus` BETWEEN 2 AND 5

# SELECT \* FROM `users` WHERE `first\_name` = 'ahmed' OR `first\_name` = 'galal' OR `first\_name` = 'alaa'

# SELECT \* FROM `users` WHERE `first\_name` IN('ahmed','galal','alaa')

# SELECT `users`.`id`,`users`.`last\_name` FROM `users` WHERE `code` is NOT NULL

# SELECT CONCAT(`first\_name`, ' ',`last\_name`) AS `full\_name` , `bonus` FROM `users`

# SELECT \* FROM `users` WHERE DATE\_FORMAT(`created\_at`,'%Y-%m-%d') = CURDATE()

# SELECT DISTINCT `last\_name` FROM `users`

/\* SELECT

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

MAX(`bonus`) AS `max\_bonus`,

MIN(`bonus`) AS `min\_bonus`,

SUM(`bonus`) AS `sum\_bonus`,

AVG(`bonus`) AS `avg\_bonus`

FROM

`users`

\*/

/\* SELECT

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

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

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

FROM

`users`

\*/

/\*

SELECT

`last\_name`,

ROUND(AVG(`bonus`)) AS `family\_avg`

FROM

`users`

GROUP BY

`last\_name`

HAVING `family\_avg` <> 0

ORDER BY

`family\_avg` ASC,

`last\_name` DESC

\*/

/\* SELECT `gender` , COUNT(`id`) AS `users\_count` , MAX(`bonus`) AS `max\_bonus`

FROM `users`

GROUP BY `gender`

\*/

/\* SELECT

CONCAT(

`users`.`first\_name`,

' ',

`users`.`last\_name`

) AS `fullname`,

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

FROM

`users`

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

GROUP BY

`users`.`id` \*/

/\* SELECT

`products`.\*,

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

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

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

FROM

`products`

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

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

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

\*/

SELECT \* FROM `users` WHERE `first\_name` LIKE 'a\_\_%'

# % => 0 , one , multiple characters

# \_ => one missed character