# Day1

1. ; => end of query
2. Comment => single line (#) , multi-line ( /\* \*/ )
3. CREATE DATABASE databaseName (CREATE DATABASE IF NOT EXISTS databaseName)
4. DROP DATABASE databaseName (DROP DATABASE IF EXISTS databaseName)
5. CREATE TABLE `tableName` (  
    `columnName` DataType Constraints Extra,

`columnName1` DataType Constraints Extra,  
)

1. DROP TABLE `tableName`
2. ALTER TABLE `tableName` ADD `columnName` ` DataType Constraints Extra
3. ALTER TABLE `tableName` CHANGE `columnNameToChange` `newColumnName` ` DataType Constraints Extra
4. ALTER TABLE `tableName` DROP `columnName`
5. ALTER TABLE `chlidTable` ADD CONSTRAINT `constraintName` FORIEN KEY(`forienKeyColumn`) REFERENCES `parentTable`(`primaryKey`) ON DELETE CASCADE ON UPDATE SET NULL
6. INSERT INTO `tableName` (`columnName1`,….) VALUES (‘value1’,….) , (‘value1’,….)
7. INSERT INTO `tableName` VALUES (DEFAULT , ‘value1’,….) , (DEFAULT , ‘value1’,….)
8. UPDATE `tableName` SET `columnName` = ‘value’ , `columName2` = 12 WHERE conditon
9. UPDATE `tableName` SET `columnName` = ‘value’
10. DELETE FROM `tableName` WHERE codition
11. DELETE FROM `tableName`
12. TRUNCATE TABLE `tableName`

# DAY2

1. SELECT \* FROM `tableName`
2. SELECT `columnName1`,`columnName2` FROM `tableName`
3. SELECT \* FROM `tableName` WHERE condition
4. Conditions (logical , comparison)
5. WHERE `columnName` BETWEEN min AND max
6. WHERE `columnName` IN(value1,value2,…)
7. WHERE `columnName` IS NULL || IS NOT NULL
8. WHERE `columnName` LIKE ‘a\_%’
9. Aggregates (MIN,MAX,COUNT,AVG,SUM,GROUP\_CONCAT)

**Ex1::**  
SELECT

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

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

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

FROM

`users`

**Ex2::**  
SELECT

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

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

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

FROM

`users`

**Ex3::**

SELECT COUNT(`id`) AS `users\_count` FROM `users` WHERE `status` = 1

1. GROUP BY `columnName`
2. ORDER BY `columnName` DESC || ASC , `columnName2` DESC || ASC
3. HAVING condition
4. IF(condition on column , true , false)

**Ex1::**

SELECT

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

`last\_name`

FROM

`users`

GROUP BY

`last\_name`

HAVING `avg\_bonus` > 4

ORDER BY

`avg\_bonus` DESC , `last\_name` DESC

LIMIT 1

**Ex2::**

SELECT

COUNT(`id`) AS `useers\_verified\_count`,

IF(`status` = 1 , 'verified' , 'not verified') AS `status`

FROM

`users`

GROUP BY

`status`

1. INNER JOIN , LEFT JOIN , RIGHT JOIN

**Ex1::**

|  |  |
| --- | --- |
| FULL NAME | Cart\_count |
| Galal husseny | 5 |
| Ahmed Mohamed | 3 |
| Donya ahmed | 0 |

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

**EX2::**

SELECT

`products`.\*,

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

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

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

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

FROM

`products`

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

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

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

1. Query UNION Query
2. CREATE VIEW viewName` AS ( Query )
3. DROP VIEW `viewName`
4. SELECT DISTINCT `columnName` FROM `tableName`
5. Build In funcs => mysql

**EX1::**

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

**EX:2:**

**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