MYSQL(basics)

➢ MySQL Data Types:

Numeric Data Type:

Data Type Syntax	Description
INT	A normal-sized integer that can be signed or unsigned.You can specify a width of up to 11 digits.
TINYINT	A very small integer that can be signed or unsigned. You can specify a width of up to 4 digits.
SMALLINT	A small integer that can be signed or unsigned. You can specify a width of up to 5 digits.
MEDIUMINT	A medium-sized integer that can be signed or unsigned.You can specify a width of up to 9 digits.
BIGINT	A large integer that can be signed or unsigned.You can specify a width of up to 20 digits.
FLOAT(m,d)	A floating-point number that cannot be unsigned. You can define the display length (m) and the number of decimals (d). This is not required and will default to 10,2, where 2 is the number of decimals and 10 is the total number of digits (including decimals). Decimal precision can go to 24 places for a float.
DOUBLE(m,d)	A double precision floating-point number that cannot be unsigned. You can define the display length (m) and the number of decimals (d). This is not required and will default to 16,4, where 4 is the number of decimals. Decimal precision can go to 53 places for a double. Real is a synonym for double.
DECIMAL(m,d)	An unpacked floating-point number that cannot be unsigned. In unpacked decimals, each decimal corresponds to one byte. Defining the display length (m) and the number of decimals (d) is required. Numeric is a synonym for decimal.

Date and Time Data Type:

Data Type Syntax	Maximum Size	Explanation
DATE	Values range from '1000-01-01' to '9999-12-31'.	Displayed as 'yyyy-mm-dd'.
DATETIME	Values range from '1000-01-01 00:00:00' to '9999-12-31 23:59:59'.	Displayed as 'yyyy-mm-dd hh:mm:ss'.
TIMESTAMP(m)	Values range from '1970-01-01 00:00:01' UTC to '2038-01-19 03:14:07' TC.	Displayed as 'YYYY-MM-DD HH:MM:SS'.
TIME	Values range from '-838:59:59' to '838:59:59'.	Displayed as 'HH:MM:SS'.
YEAR[(2|4)]	Year value as 2 digits or 4 digits.	Default is 4 digits.

String Data Types:

Data Type Syntax	Maximum Size	Explanation
CHAR(size)	Maximum size of 255 characters.	Where size is the number of characters to store. Fixed-length strings. Space padded on right to equal size characters.
VARCHAR(size)	Maximum size of 255 characters.	Where size is the number of characters to store. Variable-length string.
TINYTEXT(size)	Maximum size of 255 characters.	Where size is the number of characters to store.
TEXT(size)	Maximum size of 65,535 characters.	Where size is the number of characters to store.
MEDIUMTEXT(size)	Maximum size of 16,777,215 characters.	Where size is the number of characters to store.
LONGTEXT(size)	Maximum size of 4GB or	Where size is the number of characters to store.

	4,294,967,295 characters.	
BINARY(size)	Maximum size of 255 characters.	Where size is the number of binary characters to store. Fixed-length strings. Space padded on right to equal size characters. (introduced in MySQL 4.1.2)
VARBINARY(size)	Maximum size of 255 characters.	Where size is the number of characters to store. Variable-length string. (introduced in MySQL 4.1.2)

Large Object Data Types (LOB) Data Types:

Data Type Syntax	Maximum Size
TINYBLOB	Maximum size of 255 bytes.
BLOB(size)	Maximum size of 65,535 bytes.
MEDIUMBLOB	Maximum size of 16,777,215 bytes.
LONGTEXT	Maximum size of 4gb or 4,294,967,295 characters.

➢ MySQL Database:

1. CREATE DATABASE:

To create database syntax is

CREATE DATABASE database_name;

You can check the created database by the following query:

SHOW DATABASES;

2. USE DATABASE:

use SQL command **USE** to select a particular database ,syntax is

USE database_name;

3.MySQL Drop Database:

You can drop/remove/delete by using drop database command.

Syntax is : **DROP DATABASE** database_name;

➢ Table & Views:

1.CREATE TABLE:

syntax for creating a MySQL table in the database is

CREATE TABLE table_name (column_name column_type...);

See the created table:

SHOW tables;

See the table structure:

DESCRIBE table_name;

2.ALTER TABLE:

It is used when you want to change the name of your table or any table field. It is also used to add or delete an existing column in a table.The ALTER statement is always used with **"ADD", "DROP" and "MODIFY**" commands according to the situation.

a)ADD a column in the table:

Syntax is : ALTER TABLE table_name
ADD new_column_name column_definition
[**FIRST** | **AFTER** column_name];

Column_definition is data_type and null or not null etc,,.
Last line in syntax is **optional**.if it is not specified the data will be added at the end of the column.

See the recently added columns:
SELECT* **FROM table_name;**

b)Add multiple columns in the table:

Syntax is:ALTER TABLE table_name
ADD new_column_name column_definition
[**FIRST** | **AFTER** column_name],
ADD new_column_name column_definition
[**FIRST** | **AFTER** column_name], ... ;

c)MODIFY column in the table:

The MODIFY command is used to change the column definition of the table.
Syntax:ALTER TABLE table_name
MODIFY column_name column_definition
[**FIRST | AFTER** column_name];

d)DROP column in table:

The DROP command is used to delete the column in the table.
Syntax:ALTER TABLE table_name
DROP COLUMN column_name;

e)RENAME column in table:

Syntax is: **ALTER TABLE** table_name
CHANGE **COLUMN** old_name new_name
column_definition
[**FIRST** | **AFTER** column_name]

f)RENAME table:

Syntax is: **ALTER TABLE** table_name
RENAME **TO** new_table_name;

3.TRUNCATE TABLE:

MYSQL TRUNCATE statement removes the complete data without removing its structure.

Syntax : **TRUNCATE TABLE** table_name;

4.DROP TABLE:

To remove the complete data with its structure.

Syntax is: **DROP TABLE** table_name;

5.MYSQL VIEW:

In MySQL, View is a virtual table created by a query by joining one or more tables.

1.CREATE VIEW:

A VIEW is created by SELECT statements. SELECT statements are used to take data from the source table to make a VIEW.

CREATE [OR REPLACE] **VIEW** view_name **AS**
SELECT columns
FROM tables
[**WHERE** conditions];

OR REPLACE is optional,It is used when a VIEW already exist. If you do not specify this clause and the VIEW already exists, the CREATE VIEW statement will return an error.

[WHERE conditions] also optional.

TO see created/updated view: **SELECT** * **FROM** view_name;

2.UPDATED VIEW: Syntax is

ALTER VIEW view_name **AS**
SELECT columns
FROM table
WHERE conditions;

3.DROP VIEW: Syntax is

DROP VIEW [IF EXISTS] view_name;

IF EXISTS is optional.If you do not specify this clause and the VIEW doesn't exist, the DROP VIEW statement will return an error.

➢ MySQL Queries:

1.INSERT RECORD:

Insert records in Partial values syntax is

INSERT INTO table_name (field1, field2,...fieldN)

VALUES (value1, value2,...valueN);

Syntax for all fields are

INSERT INTO table_name **VALUES** (value1, value2,...valueN);

2.UPDATE RECORD:

UPDATE statement is used to update data of the MySQL table within the database.

The **UPDATE** statement is used with the SET, and WHERE clauses. The **SET** clause is used to change the values of the specified column. We can update single or multiple columns at a time. The **WHERE** clause is used to specify the condition and can update selected rows in a table, but it is optional. syntax is

UPDATE table_name

SET field1=new-value1, field2=new-value2, ...

[**WHERE** Clause]

3.DELETE RECORD:

MySQL DELETE statement is used to delete data from the MySQL table within the database.Syntax is

DELETE FROM table_name

WHERE (Condition specified);

4.SELECT RECORD:

Used to fetch data from the one or more tables in MySQL. We can retrieve records of all fields or specified fields.

Syntax for specified fields:

SELECT expressions

FROM tables [**WHERE** conditions];

Syntax for all fields:

SELECT * **FROM** tables [**WHERE** conditions];

➢ MySQL Clauses:

1.WHERE CLAUSE:

MySQL WHERE Clause is used with SELECT, INSERT, UPDATE and DELETE clause to filter the results. It specifies a specific position where you have to do the operation.

Syntax is **WHERE** condition**;**

2.DISTINCT CLAUSE:

It is used to remove duplicate records from the table and fetch only the unique records. The DISTINCT clause is only used with the SELECT statement.

Syntax is **SELECT DISTINCT** expressions

FROM tables

[**WHERE** conditions];

3.FROM CLAUSE:

The MySQL FROM Clause is used to select some records from a table. It can also be used to retrieve records from multiple tables using JOIN condition.

Syntax is

```
FROM table1
[ { INNER JOIN | LEFT [OUTER] JOIN| RIGHT [OUTER] JOIN } table2
ON table1.column1 = table2.column1 ]
```

NOTE: table1 and table2: specify tables used in the MySQL statement. The two tables are joined based on table1.column1 = table2.column1.

Ex:

```
SELECT *  FROM officers
WHERE officer_id <= 3;
```

4.ORDER BY CLAUSE:

The MYSQL ORDER BY Clause is used to sort the records in ascending or descending order.

Syntax is

```
SELECT expressions
FROM tables
[WHERE conditions]
ORDER BY expression [ ASC | DESC ];
```

NOTE:Here where,asc and dsc are optional.

5.GROUP BY CLAUSE:

The MYSQL GROUP BY Clause is used to collect data from multiple records and group the result by one or more column. It is generally used in a SELECT statement.

You can also use some aggregate functions like COUNT, SUM, MIN, MAX, AVG etc. on the grouped column.

Syntax is

```
SELECT expression1, expression2, ... expression_n,
aggregate_function (expression)
FROM tables  [WHERE conditions]
GROUP BY expression1, expression2, ... expression_n;
```

Parameters: expression1, expression2, ... expression_n: It specifies the expressions that are not encapsulated within an aggregate function and must be included in the GROUP BY clause.
aggregate_function: It specifies a function such as SUM, COUNT, MIN, MAX, or AVG etc. tables: It specifies the tables, from where you want to retrieve the records. There must be at least one table listed in the FROM clause. WHERE conditions: It is optional. It specifies the conditions that must be fulfilled for the records to be selected.

6.HAVING CLAUSE:

HAVING Clause is used with GROUP BY clause. It always returns the rows where condition is TRUE.It is used to restrict the groups of returned rows. It shows only those groups in result set whose conditions are TRUE.

Syntax is

```
SELECT expression1, expression2, ... expression_n,
 aggregate_function (expression)
```

FROM tables
[**WHERE** conditions] **GROUP BY** expression1, expression2, ... expression_n
HAVING condition;

➢ MySQL Conditions:

1. AND CONDITION:

It is used with SELECT, INSERT, UPDATE or DELETE statements to test two or more conditions in an individual query.
Syntax is:

WHERE condition1 AND condition2
... AND condition_n;

2. OR CONDITION:

It specifies that if you take two or more conditions then one of the conditions must be fulfilled to get the records as result.
Syntax is

WHERE condition1 OR condition2
... OR condition_n;

3. AND-OR CONDITION:

We can use AND & OR condition both together with the SELECT, INSERT, UPDATE and DELETE statement. While combine these conditions, you must be aware where to use round brackets so that the database know the order to evaluate each condition.
Syntax is

WHERE condition1 AND condition2
... OR condition_n;

4. AND-OR CONDITION:

5. AND-OR CONDITION: