

MySQL Server



Day 1 :

➤ SQL : Structured Query Language

Used for Storing, Manipulating & Retrieving Data from Database.

1. Can Create Database/Tables in Database
2. Can execute Queries against Database as per requirement
3. Can retrieve data from a database
4. Can insert data into Database
5. Can Update records in Database
6. Can Delete Records from Database/Tables

Terms in Database :

Fields- Columns

Records- Rows

Resources : <https://www.w3schools.com/>



Day 1 :

1. Create Database :

Syntax : CREATE DATABASE DB_Name;

E.g. I want to create Database with name KTCTC → **SQL Query :** CREATE DATABASE KTCTC;

2. Create Tables :

Syntax : CREATE TABLE table_Name (

Column1_Name datatype,

Column2_Name datatype,

Column3_Name datatype,

.....

)



Day 1 :

3. Insert Into Statement : It is used for inserting new records in Table.

Syntax :

- A. With Specifying Column Names & Values in statement : **INSERT INTO table_Name (column1_Name, column2_Name,...) VALUES (value1,value2,...);**
- B. With Specifying only values in statement : **INSERT INTO table_Name VALUES (value1,value2,...);** → **Values must be inserted as per order of columns in table**
- C. Adding multiple records using single Query : **INSERT INTO table_Name VALUES (value1,value2), (value1,value2), (value1,value2),.....(value1,value2);**



Day 1 :

4. Select Statement: Select Statement is used to retrieve data from Database tables

Retrieved data is stored in result table(temporary) and known as **result set**.

Syntax :

A. SELECT * FROM table_Name;

* is used whenever user wants to retrieve all fields related to all records present in Database Table.

B. SELECT column1_Name, coulmn2_Name,...from table_Name;

Whenever user wants to retrieve selective fields for all records present in DB table, instead of * in select statement use particular column names.

5. Select Distinct Statement: Select Distinct Statement is used to retrieve only distinct/different data from Database tables

Result set returns only distinct values present in specified column.

Syntax : SELECT DISTINCT(column1_Name) from table_Name;



Day 2 :

6. Where Clause:

Where clause is used in Select/Update/Delete statement

to filter records based on conditions specified in statement.



Syntax :

A. **SELECT * FROM table_Name WHERE condition;**

B. **SELECT column1_Name,column2_Name,... From table_Name WHERE condition;**

Operator	Description	Operator	Description
=	Equal to	LIKE	Search for a pattern
>	Greater than	IN	To specify multiple possible values in column
<	Less than	!=	Not Equal to
>=	Greater than or equal to	BETWEEN	Between a certain range
<=	Less than or equal to	AND	Logical AND between 2 conditions
OR	Logical OR between 2 conditions	NOT	Logical not before condition

Day 3 :

7. ORDER BY Keyword:



Order by is used to sort result set in ascending(ASC)/descending(DESC) order

- a. **Default Sort Order :** Ascending
- b. ORDER BY keyword will be used along with WHERE clause in statement.
- c. Sorting can be applied on multiple fields/columns at a time

Syntax :

- A. **SELECT * FROM table_Name ORDER BY column_Name ASC/DESC;**
OR **SELECT * FROM table_Name WHERE condition ORDER BY column_Name ASC/DESC;**
- B. **SELECT column1_Name,column2_Name,... From table_Name ORDER BY column_Name ASC/DESC;**
OR **SELECT column1_Name,column2_Name,... From table_Name WHERE condition ORDER BY column_Name ASC/DESC;**
- C. **SELECT * FROM table_Name ORDER BY column1_Name ASC, column2_Name DESC;**

Day 3 :

8. SQL Select LIMIT clause:

LIMIT clause is used to specify number of records to return from table.



- A. LIMIT clause can be used along with WHERE clause in statement.
- B. LIMIT clause can be used along with WHERE & ORDER BY in combination

Syntax :

A. **SELECT * FROM table_Name **LIMIT number;****

OR SELECT * FROM table_Name WHERE condition **LIMIT number;**

OR SELECT * FROM table_Name WHERE condition ORDER BY column_Name ASC/DESC **LIMIT number;**

B. **SELECT column1_Name,... From table_Name **LIMIT number;****

OR SELECT column1_Name,... From table_Name WHERE condition **LIMIT number;**

**OR SELECT column1_Name,... From table_Name WHERE condition ORDER BY column_Name ASC/DESC
LIMIT number;**

Day 3 :

9. Aggregate Functions in SQL:

Returns single value as per function used on specified field/column.



A. Aggregate functions can be used along with WHERE clause.

A. **MIN()**- Returns smallest value from specified field

Syntax : SELECT **MIN(column_Name)** from table_Name WHERE condition;

B. **MAX()**- Returns largest value from specified field

Syntax : SELECT **MAX(column_Name)** from table_Name WHERE condition;

C. **COUNT()**- Returns number of records matches specified condition in WHERE clause.

Syntax : SELECT **COUNT(column_Name)** from table_Name WHERE condition;

D. **AVG()**- Returns average value of a numeric column.

Syntax : SELECT **AVG(column_Name)** from table_Name WHERE condition;

E. **SUM()**- Returns sum of numeric column.

Syntax : SELECT **SUM(column_Name)** from table_Name WHERE condition;

Day 4 :

10. SQL JOINs:



Used to combine records from 2 or more tables based on common fields/relations between them.

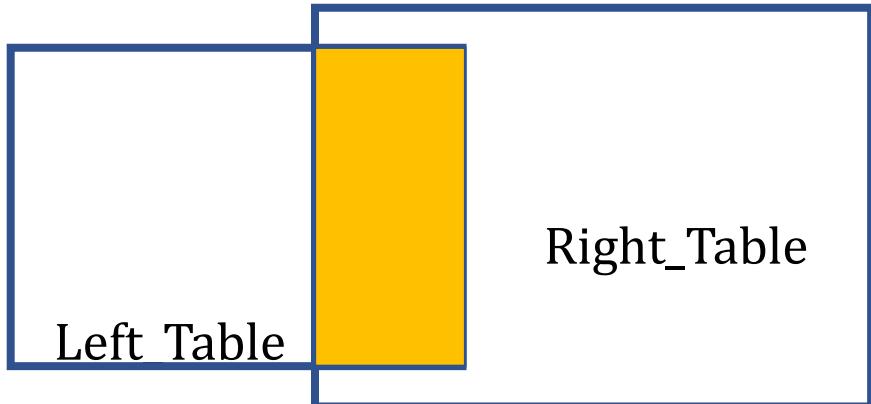
Types of JOINs :

1. **INNER JOIN** : Returns those records having matching fields in both the tables
2. **LEFT JOIN** : Returns all records from left table & matching records from right table
3. **RIGHT JOIN** : Returns all records from right table & matching records from left table
4. **FULL JOIN** : Returns all the records from both tables

Day 4 :

10. A. INNER JOIN :

Returns those records having matching fields in both the tables



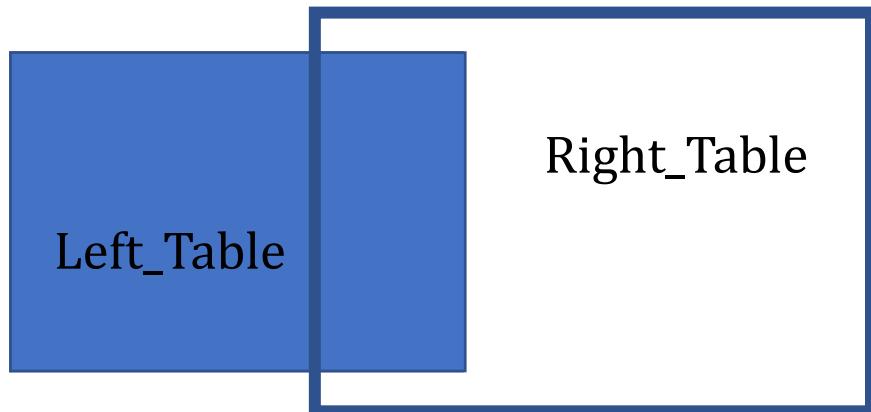
Syntax : `SELECT * FROM Left_Table INNER JOIN Right_Table ON Left_Table.column_Name=Right_Table.column_Name;`

OR `SELECT column1_Name,..... FROM Left_Table INNER JOIN Right_Table ON
Left_Table.column_Name=Right_Table.column_Name;`

Day 4 :

10. B. LEFT JOIN : Returns all records from left table & matching records from right table

: For unmatched records in right table, NULL values are returned in result set.



Syntax : SELECT * FROM **Left_Table** **LEFT JOIN** **Right_Table** ON **Left_Table.column_Name=Right_Table.column_Name**;

OR SELECT column1_Name,..... FROM **Left_Table** **LEFT JOIN** **Right_Table** **ON**
 Left_Table.column_Name=Right_Table.column_Name;

Table A: RN, name 1,2

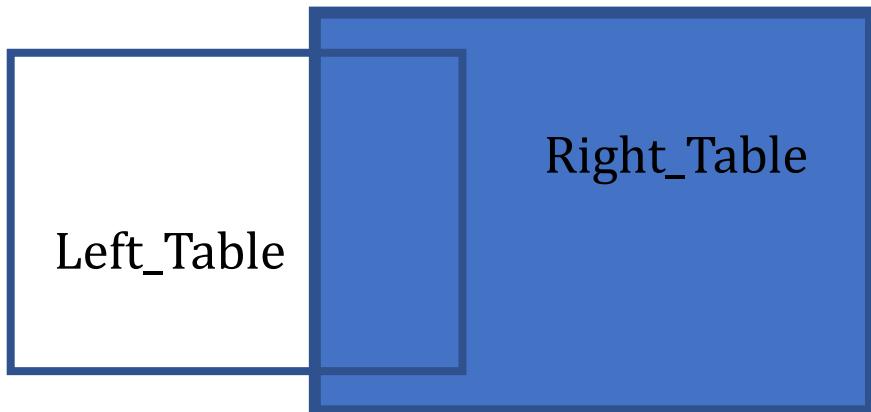
Table B: RN,score 1



Day 4 :

10. C. **RIGHT JOIN** : Returns all records from right table & matching records from left table

: For unmatched records in left table, NULL values are returned in result set.



Syntax : `SELECT * FROM Left_Table RIGHT JOIN Right_Table ON Left_Table.column_Name=Right_Table.column_Name;`

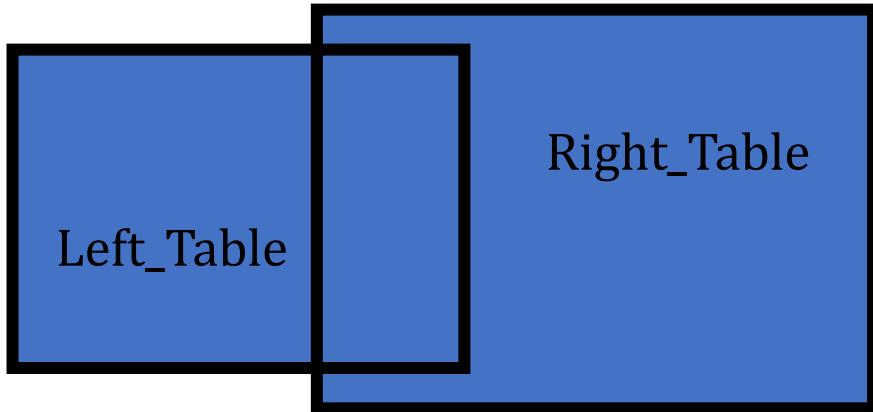
OR `SELECT column1_Name,..... FROM Left_Table RIGHT JOIN Right_Table ON
Left_Table.column_Name=Right_Table.column_Name;`



Day 4 :

10. D. **FULL JOIN** : Returns all the records from both tables

: For unmatched records, NULL values are returned in result set.



Syntax : SELECT * FROM **Left_Table** **FULL JOIN** **Right_Table** ON **Left_Table.column_Name=Right_Table.column_Name**;

OR SELECT column1_Name,..... FROM **Left_Table** **FULL JOIN** **Right_Table** **ON**
Left_Table.column_Name=Right_Table.column_Name;



Day 5 :

11. Group By Statement : This is used to group rows with similar values.

: This is often used with aggregate functions.

Syntax : SELECT columnName1,... FROM table_Name **GROUP BY column_Name;**

OR SELECT columnName1,... FROM table_Name WHERE condition **GROUP BY column_Name;**

OR SELECT columnName1,... FROM table_Name WHERE condition **GROUP BY column_Name ORDER BY column_Name;**

OR SELECT columnName1,... FROM table_Name WHERE condition **GROUP BY column_Name ORDER BY column_Name LIMIT No;**

WHERE-->GROUP BY--> ORDER BY-->LIMIT



Day 5 :

12. Having clause: This is used in place of where clause if condition is having aggregate function.

Syntax : SELECT columnName1,... FROM table_Name GROUP BY column_Name **HAVING column_Name;**

OR SELECT columnName1,... FROM table_Name GROUP BY column_Name **HAVING column_Name;**

OR SELECT columnName1,... FROM table_Name GROUP BY column_Name **HAVING column_Name ORDER BY column_Name;**

OR SELECT columnName1,... FROM table_Name GROUP BY column_Name **HAVING column_Name ORDER BY column_Name LIMIT No;**

WHERE-->GROUP BY -> Having --> ORDER BY-->LIMIT



Day 5 :

13. Delete Statement: This is used delete records form table.

Always used with where clause to delete records having specified condition in statement to insure unnecessary deletion of records from table.

Syntax : DELETE FROM table_Name ;-----Deletes All records from table.

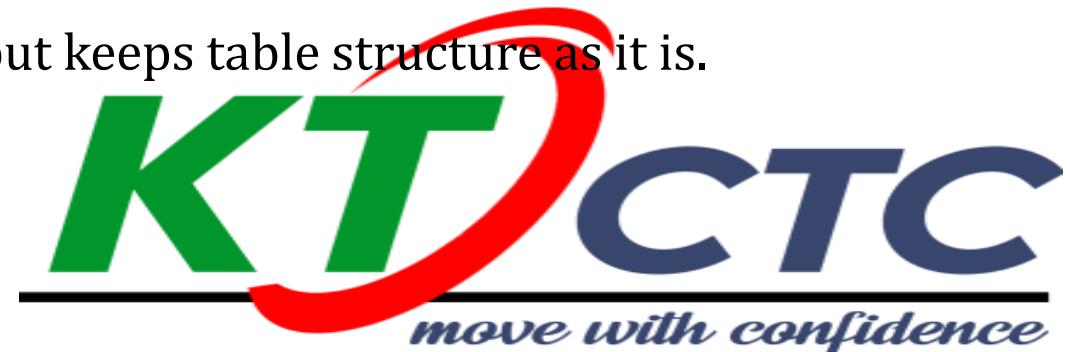
DELETE FROM table_Name WHERE condition;

14. DROP Statement: Used to Delete table from DB-----Loss of all information inside table

Syntax: DROP TABLE table_Name;

15. Truncate Statement : Used to delete all data from table , but keeps table structure as it is.

Syntax: TRUNCATE TABLE table_Name;



Day 6 :

16. UNION & UNION ALL Operator : It is used to combine 2 or more tables or Result set of any select statement.

- Both tables/results sets on which UNION is used must have same number of columns with same datatypes(Columns must be in same order).

Syntax:

1. SELECT column_Name,... FROM table1

UNION

SELECT column_Name,... FROM table2;

2. SELECT column_Name,... FROM table1

UNION ALL

SELECT column_Name,... FROM table2;

UNION- It only returns rows with unique values.

UNION ALL- It returns all records from both tables including duplicates also.



Day 6 :

17. SQL Constraints :

Constraints are used to specify rules on fields/columns in DB table.



Always used while using CREATE/ALTER statement for table creation/modification.

Syntax: CREATE TABLE table_Name(

column_Name datatype constraints,

.....

)

1. **NOT NULL** : This will not allow NULL value in that column.
2. **UNIQUE** : This will not allow insertion of duplicate values in that column.
3. **PRIMARY KEY** : This will not allow NULL & Duplicate values in that column./Single Primary key in table
4. **FOREIGN KEY** : This will uniquely identifies records from other table/ Allows duplicate as well NULL values in that column./More than one foreign keys are possible in a table
5. **CHECK** : This is used to insure that all values getting into that column satisfies specific conditions.
6. **DEFAULT** : This sets default value in that column, if it is missed while insertion of data in table.

Day 6 :

18. ALTER TABLE: This statement is used to add/delete/modify table structure/column.

- Also used to add/delete/modify constraints on columns

Syntax: ALTER TABLE table_Name **ADD COLUMN** column_Name datatype constraints;

OR ALTER TABLE table_Name **DROP COLUMN** column_Name;

OR ALTER TABLE table_Name **MODIFY COLUMN** column_Name datatype constraints;



Day 6 :

T1 : What is SQL?→ SQL stands for structured Query Language, which is a database tool used to create & access database to support any software application.

T2 : What are the different types of statements supported by SQL?→ There are 3 types of statements supported by SQL.

- 1. DDL**-Data Definition Language -Used to define structure of DB-CREATE/ALTER/DROP Statements
- 2. DML**- Data Manipulation Language- Used to manipulate records in DB-
INSERT/UPDATE/DELETE/SELECT
- 3. DCL**-Data Control Language –Used to set various permissions on DB.

T3 : What is Difference between DROP & DELETE?

T4 : What is Difference between DROP & TRUNCATE/DELETE & TRUNCATE?

T5 : Syntax for Various statements?

T6 : Use of WHERE & HAVING clause.

T7 : To write Scenario based SQL Queries→ EmpId, EmpName,City, Dept,Project,Salary



Day 6 :

T8 : What is JOIN & its types along with Syntax.

T9 : Write JOIN Queries based on scenarios.

T10. Which aggregate functions can be used in SQL statements?

T11. Write SQL queries using LIKE operator based on scenarios

T12. What is difference between IN—Multiple OR—Used on single column & OR operators?

T13. Write SQL statement to get Nth Highest/Nth Lowest salary?



10. Wildcards used in LIKE operator:



Symbols	Description
%	Represents zero or any number of characters
_ : underscore	Represents single character
[as]%	Represents any single character within the brackets
[^as]%	Represents any character not in the bracket
[a-m] % : dash	Represents a range of characters