**MySQL - Introduction What is a Database?**

A database is a separate application that stores a collection of data. Each database has one or more distinct APIs for creating, accessing, managing, searching and replicating the data it holds. A database is an organized collection of data It is a collection of schemas, tables, queries, reports, views, and other objects. Database designers typically organize the data to model aspects of reality in a way that supports processes requiring information, such as (for example) modelling the availability of rooms in hotels in a way that supports finding a hotel with vacancies. Other kinds of data stores can also be used, such as files on the file system or large hash tables in memory but data fetching and writing would not be so fast and easy with those type of systems.

**Define RDBMS & its Terminology**

A Relational DataBase Management System (RDBMS) is a software that

* Enables you to implement a database with tables, columns and indexes.
* Guarantees the Referential Integrity between rows of various tables.
* Updates the indexes automatically.
* Interprets an SQL query and combines information from various tables.

**RDBMS Terminology**

1. **Table** − A table is a matrix with data. A table in a database looks like a simple spreadsheet.
2. Column − One column (data element) contains data of one and the same kind, for example the column postcode.
3. Row − A row (= tuple, entry or record) is a group of related data, for example the data of one subscription.
4. Redundancy − Storing data twice, redundantly to make the system faster.
5. Primary Key − A primary key is unique. A key value can not occur twice in one table. With a key, you can only find one row.
6. Foreign Key − A foreign key is the linking pin between two tables.
7. Compound Key − A compound key (composite key) is a key that consists of multiple columns, because one column is not sufficiently unique.
8. Index − An index in a database resembles an index at the back of a book.
9. Referential Integrity − Referential Integrity makes sure that a foreign key value always points to an existing row

**MySQL**

MySQL is the most popular Open Source Relational SQL Database Management System. It is the best RDBMS being used for developing various web-based software applications. MySQL is developed, marketed and supported by MySQL AB, which is a Swedish company. MySQL is a fast, easy-to-use RDBMS being used for many small and big businesses. MySQL is developed, marketed and supported by MySQL AB, which is a Swedish company. MySQL is becoming so popular because of many good reasons −

* MySQL is released under an open-source license. So you have nothing to pay to use it.
* MySQL is a very powerful program in its own right. It handles a large subset of the functionality
* of the most expensive and powerful database packages.
* MySQL uses a standard form of the well-known SQL data language.
* MySQL works on many operating systems and with many languages including PHP, PERL, C, C++, JAVA, etc.
* MySQL works very quickly and works well even with large data sets.
* MySQL is very friendly to PHP, the most appreciated language for web development.
* MySQL supports large databases, up to 50 million rows or more in a table. The default file size limit for a table is 4GB, but you can increase this (if your operating system can handle it) to a theoretical limit of 8 million terabytes (TB).
* MySQL is customizable. The open-source GPL license allows programmers to modify the MySQL software to fit their own specific environments.

**1. CREATE DATABASE Statement**

CREATE DATABASE database\_name;

**2.CREATE TABLE Statement**

CREATETABLEtable\_name

(

Col\_name1 datatype(),Col\_name2 datatype(),**…** Col\_namen datatype(),

);

**Example**

CREATE TABLE users (

id INTEGER PRIMARY KEY,

username TEXT NOT NULL,

email TEXT NOT NULL,

password TEXT NOT NULL,

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);

****ALTER :****  
This command is used to add, delete or change columns in the existing table. The user needs to know the existing table name and can do add, delete or modify tasks easily.

**Syntax:**

ALTER TABLE table\_name

ADD column\_name datatype;

**Example:**

ALTER TABLE users

ADD COLUMN phone\_number VARCHAR(20);

**RENAME**

**Syntax**

ALTER TABLE table\_name

CHANGE COLUMN old\_column\_name new\_column\_name data\_type;

**Example:**

ALTER TABLE users

CHANGE COLUMN phone\_number contact\_number VARCHAR(20);

****DROP** :**  
This command is used to remove an existing table along with its structure from the Database.

****Syntax****

DROP TABLE table\_name;

**Example:**

DROP TABLE Student\_info;

### **DROP A COLUMN**

**Syntax:**

ALTER TABLE table\_name

DROP COLUMN column\_name;

**Example:**

ALTER TABLE users

DROP COLUMN phone\_number;

**DESCRIBE**

The **DESCRIBE** statement is commonly used to retrieve metadata about a database object, such as a table.

**Syntax:**

SHOW COLUMNS FROM table\_name;

**Example:**

SHOW COLUMNS FROM users;

# **DML Commands in SQL**

DML is an abbreviation of **Data Manipulation Language**.

The DML commands in Structured Query Language change the data present in the SQL database. We can easily access, store, modify, update and delete the existing records from the database using DML commands.

****Following are the four main DML commands in SQL:****

1. SELECT Command
2. INSERT Command
3. UPDATE Command
4. DELETE Command

#### **1. INSERT**

It inserts or adds new rows or records in the existing table.

**Syntax:**

INSERT INTO table\_name(column1,column2,column3, ...)  
VALUES(value1,value2,value3, ...); ;

**Example:**

INSERT INTO users (username, email, password) VALUES

('john\_doe', 'john@example.com', 'hashed\_password\_123'),

('jane\_doe', 'jane@example.com', 'hashed\_password\_456');

**2.SQL SELECT Statement**

The SELECT statement is the most commonly used command in Structured Query Language. It is used to access the records from one or more database tables and views. It also retrieves the selected data that follow the conditions we want.

**1.SELECT Statement**

SELECT column1, column2....columnN FROM table\_name;

**2. DISTINCT Clause**

SELECT DISTINCT Column1,column2,etc

FROM Table name

**3.WHERE Clause**

**Example**

SELECT column1, column2, column2, etc

FROM table\_name

WHERE column = 'Software Engineer' AND salary > 80000;

**4.AND/OR Clause**

**Example**

-- Select employees who are 'Software Engineer' or have a salary > 80000

SELECT employee\_id, employee\_name, job\_title, salary

FROM employees

WHERE job\_title = 'Software Engineer' OR salary > 80000;

**5. IN Clause**

**Example**

-- Select employees with job title 'Software Engineer' or 'Data Scientist'

SELECT employee\_id, employee\_name, job\_title, salary

FROM employees

WHERE job\_title IN ('Software Engineer', 'Data Scientist');

**6. BETWEEN Clause**

**Example:**

-- Select employees with salary between 50000 and 80000

SELECT employee\_id, employee\_name, job\_title, salary

FROM employees

WHERE salary BETWEEN 50000 AND 80000;

**7.LIKE Clause**

**Example**

SELECT employee\_id, employee\_name, job\_title, salary

FROM employees

WHERE employee\_name LIKE 'J%';;

**8.ORDER BY Clause**

**Example:**

SELECT employee\_id, employee\_name, job\_title, salary

FROM employees

ORDER BY salary DESC;

**9. GROUP BY**

**T**he GROUP BYclause in SQL is used to group rows that have the same values in specified columns into summary rows. It is often used in conjunction with aggregate functions (such as **SUM**, **COUNT**, **AVG**, etc.) to perform operations on each group of rows.

**Example:**

SELECT job\_title, SUM(salary) AS total\_salary

FROM employees

GROUP BY job\_title;

**10.COUNT Clause**

**Example:**

SELECT job\_title, COUNT(\*) AS employee\_count

FROM employees

GROUP BY job\_title;

**11. HAVING Clause**

**Example:**

SELECT job\_title, AVG(salary) AS average\_salary

FROM employees

GROUP BY job\_title

HAVING AVG(salary) > 70000;

Update Query :

UPDATE command will modify any field value of any MySQL table.

* You can update one or more field altogether.
* You can specify any condition using the WHERE clause.
* You can update the values in a single table at a time.

The WHERE clause is very useful when you want to update the selected rows in a table.

**Syntax**

UPDATE table\_name SET field1 = new-value1, field2 = new-value2 [WHERE Clause ]

**Example:**

UPDATE employees

SET job\_title = 'Senior Software Engineer'

WHERE employee\_id = 1;

**Using Where Clause**

UPDATE employees

SET salary = salary \* 1.1 -- Give a 10% salary raise

WHERE job\_title = 'Software Engineer';

**DELETE Query**

If you want to delete a record from any MySQL table, then you can use the SQL command DELETE FROM. The WHERE clause is very useful when you want to delete selected rows in a table. If the WHERE clause is not specified, then all the records will be deleted from the given MySQL table.

**Syntax**

DELETE \* FROM table\_name DELETE FROM table\_name [WHERE Clause]

 You can specify any condition using the WHERE clause.

 You can delete records in a single table at a time.