**Introduction to database:**

You deal with data every day…

When you want to listen to your favourite songs, you open your playlist from your smartphone. In this case, the playlist is a database.

When you take a photo and upload it to your account on a social network like Facebook, your photo gallery is a database.

When you browse an e-commerce website to buy shoes, clothes, etc., you use the shopping cart database.

Databases are everywhere. So what is a database?  By definition, a database is merely a structured collection of data.

The data relating to each other by nature, e.g., a product belonged to a product category and associated with multiple tags. Therefore, we use the term **relational database**.

In the relational database, we model data like products, categories, tags, etc., using tables. A table contains columns and rows. It is like a spreadsheet.

A table may relate to another table using a relationship, e.g., one-to-one and one-to-many relationships.

Because we deal with a significant amount of data, we need a way to define the databases, tables, etc., and process data more efficiently. Besides, we want to turn the data into information.

And this is where SQL comes to play.

SQL – the language of the relational database

SQL stands for the structured query language.

**What is MySQL**

**MySQL? What?**

My is the daughter’s name of the [MySQL’s co-founder, Monty Widenius](https://en.wikipedia.org/wiki/Michael_Widenius).

The name of MySQL is the combination of My and SQL, MySQL.

MySQL is a database management system that allows you to manage relational databases. It is open source software backed by Oracle. It means you can use MySQL without paying a dime. Also, if you want, you can change its source code to suit your needs.

Even though MySQL is open source software, you can buy a commercial license version from Oracle to get premium support services.

MySQL is pretty easy to master in comparison with other database software like Oracle Database, or Microsoft SQL Server.

MySQL can run on various platforms UNIX, Linux, Windows, etc. You can install it on a server or even in a desktop. Besides, MySQL is reliable, scalable, and fast.

The official way to pronounce MySQL is *My Ess Que Ell, not My Sequel.*However, you can pronounce it whatever you like, who cares?

If you develop websites or web applications, MySQL is a good choice. MySQL is an essential component of the LAMP stack, which includes Linux, Apache, MySQL, and PHP.

**Features of MySQL – What is MySQL?**

* **Ease of Management –**The software very easily gets downloaded and also uses an event scheduler to schedule the tasks automatically.
* **Robust Transactional Support –**Holds the ACID (Atomicity, Consistency, Isolation, Durability) property, and also allows distributed multi-version support.
* **Comprehensive Application Development –**MySQL has plugin libraries to embed the database into any application. It also supports stored procedures, triggers, functions, views and many more for application development. You can refer to the [***RDS Tutorial***](https://www.edureka.co/blog/rds-aws-tutorial/), to understand Amazon’s RDBMS.
* **High Performance –**Provides fast load utilities with distinct memory caches and table index partitioning.
* **Low Total Cost Of Ownership –**This reduces licensing costs and hardware expenditures.
* **Open Source & 24 \* 7 Support –**This RDBMS can be used on any platform and offers 24\*7 support for open source and enterprise edition.
* **Secure Data Protection –**MySQL supports powerful mechanisms to ensure that only authorized users have access to the databases.
* **High Availability –**MySQL can run high-speed master/slave replication configurations and it offers cluster servers.
* **Scalability & Flexibility –**With MySQL you can run deeply embedded applications and create data warehouses holding a humongous amount of data.

MySQL Installation :

====== ========

<https://www.onlinetutorialspoint.com/mysql/install-mysql-on-windows-10-step-by-step.html>

Basic MySQL Queries:

**Create Database:**

**Syntax:**

**CREATE** **DATABASE** database\_name;

**Example:**

Let's take an example to create a database name "employees"

**CREATE** **DATABASE** employees;

SHOW DATABASES;

**MySQL SELECT Database**

SELECT Database is used in MySQL to select a particular database to work with. This query is used when multiple databases are available with MySQL Server.

You can use SQL command USE to select a particular database.

**Syntax:**

USE database\_name;

**Example:**

Let's take an example to use a database name "customers".

USE customers;

**MySQL Drop Database**

You can drop/delete/remove a MySQL database easily with the MySQL DROP DATABASE command. It deletes all the tables of the database along with the database permanently. It throws an error, if the database is not available.

**Syntax:**

DROP DATABASE database\_name;

**Example:**

Let's take an example to drop a database name "employees"

DROP DATABASE employees;

SHOW DATABASES;

Here, you can see that the database "employees" is removed.

**Tables:**

# **MySQL CREATE TABLE**

The MySQL CREATE TABLE command is used to create a new table into the database. A table creation command requires three things:

* Name of the table
* Names of fields
* Definitions for each field

**Syntax:**

**CREATE** **TABLE** table\_name (column\_name column\_type...);

[**next →**](https://www.javatpoint.com/mysql-alter-table)[**← prev**](https://www.javatpoint.com/mysql-drop-database)

# **MySQL CREATE TABLE**

The MySQL CREATE TABLE command is used to create a new table into the database. A table creation command requires three things:

* Name of the table
* Names of fields
* Definitions for each field

**Syntax:**

Following is a generic syntax for creating a MySQL table in the database.

**CREATE** **TABLE** table\_name (column\_name column\_type...);

**Example:**

Here, we will create a table named "cus\_tbl" in the database "customers".

**CREATE** **TABLE** cus\_tbl(

   cus\_id **INT** NOT NULL AUTO\_INCREMENT,

  cus\_firstname **VARCHAR**(100) NOT NULL,

   cus\_surname **VARCHAR**(100) NOT NULL,);

**Note:**

1. Here, NOT NULL is a field attribute and it is used because we don't want this field to be NULL. If you will try to create a record with NULL value, then MySQL will raise an error.
2. The field attribute AUTO\_INCREMENT specifies MySQL to go ahead and add the next available number to the id field.

**See the created table:**

Use the following command to see the table already created:

**SHOW tables;**

**See the table structure:**

Use the following command to see the table already created:

**DESCRIBE cus\_tbl;**

1) ADD a column in the table

**Syntax:**

**ALTER** **TABLE** table\_name

**ADD** new\_column\_name column\_definition

[ **FIRST** | **AFTER** column\_name ];

**Example:**

In this example, we add a new column "cus\_age" in the existing table "cus\_tbl".

Use the following query to do this:

**ALTER** **TABLE** cus\_tbl

**ADD** cus\_age **varchar**(40) NOT NULL;

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 ];

**Example:**

In this example, we modify the column cus\_surname to be a data type of varchar(50) and force the column to allow NULL values.

**Use the following query to do this:**

**ALTER** **TABLE** cus\_tbl

**MODIFY** cus\_surname **varchar**(50) NULL;

DROP column in table

**Syntax:**

**ALTER** **TABLE** table\_name

**DROP** **COLUMN** column\_name;

Let's take an example to drop the column name "cus\_address" from the table "cus\_tbl".

**Use the following query to do this:**

**ALTER** **TABLE** cus\_tbl

**DROP** **COLUMN** cus\_address;

RENAME column in table

**Syntax:**

**ALTER** **TABLE** table\_name

CHANGE **COLUMN** old\_name new\_name

column\_definition

[ **FIRST** | **AFTER** column\_name ]

**Example:**

In this example, we will change the column name "cus\_surname" to "cus\_title".

**Use the following query to do this:**

**ALTER** **TABLE**  cus\_tbl

CHANGE **COLUMN** cus\_surname cus\_title

**varchar**(20) NOT NULL;

RENAME table

**Syntax:**

**ALTER** **TABLE** table\_name

RENAME **TO** new\_table\_name;

**Example:**

In this example, the table name cus\_tbl is renamed as cus\_table.

**ALTER** **TABLE** cus\_tbl

RENAME **TO** cus\_table;

# **MySQL TRUNCATE Table**

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

The TRUNCATE TABLE statement is used when you want to delete the complete data from a table without removing the table structure.

**Syntax:**

**TRUNCATE** **TABLE**  table\_name;

**Example:**

This example specifies how to truncate a table. In this example, we truncate the table "cus\_tbl".

**TRUNCATE** **TABLE**  cus\_tbl;

# **MySQL DROP Table**

MYSQL DROP table statement removes the complete data with structure.

**Syntax:**

**DROP** **TABLE**  table\_name;

**Example:**

This example specifies how to drop a table. In this example, we are dropping the table "cus\_tbl".

**DROP** **TABLE**  cus\_tbl;

# **MySQL Queries**

A list of commonly used MySQL queries to create database, use database, create table, insert record, update record, delete record, select record, truncate table and drop table are given below.

## 1) MySQL Create Database

MySQL create database is used to create database. For example

1. **create** **database** db1;

## 2) MySQL Select/Use Database

MySQL use database is used to select database. For example

1. use db1;

## 3) MySQL Create Query

MySQL create query is used to create a table, view, procedure and function. For example:

**CREATE** **TABLE** customers

(id **int**(10),

**name** **varchar**(50),

 city **varchar**(50),

**PRIMARY** **KEY** (id )

);

## 4) MySQL Alter Query

MySQL alter query is used to add, modify, delete or drop colums of a table. Let's see a query to add column in customers table:

**ALTER** **TABLE** customers

**ADD** age **varchar**(50);

## 5) MySQL Insert Query

MySQL insert query is used to insert records into table. For example:

**insert** **into** customers **values**(101,'rahul','delhi');

## 6) MySQL Update Query

MySQL update query is used to update records of a table. For example:

**update** customers **set** **name**='bob', city='london' **where** id=101;

## 7) MySQL Delete Query

MySQL update query is used to delete records of a table from database. For example:

**delete** **from** customers **where** id=101;

## 8) MySQL Select Query

Oracle select query is used to fetch records from database. For example:

**SELECT** \* **from** customers;

## 9) MySQL Truncate Table Query

MySQL update query is used to truncate or remove records of a table. It doesn't remove structure. For example:

**truncate** **table** customers;  

**10) MySQL Drop Query**

MySQL drop query is used to drop a table, view or database. It removes structure and data of a table if you drop table. For example:

**drop** **table** customers;