

MALAVIYA NATIONAL INSTITUTE OF TECHNOLOGY JAIPUR

Getting Started with MySQL

Practice Session

Objectives:

- Basics of SQL
 - How to Install and work on MySQL through PhpMyadmin
 - Getting familiar with SQL
 - SQL Environment
 - SQL Data types
 - Executing some basic queries on MySQL
-

- **How to Install and work on MySQL through PhpMyadmin**

In case not installed MySQL and PhpMyadmin follow the below instructions:

1. Sudo apt-get -y update

2. Install Apache

sudo apt-get install apache2

3. Install MySQL

sudo apt-get install mysql-server.

when it will ask for login and password :

login: **root** password: **root**

4. Install PHP

sudo apt-get install php5 libapache2-mod-php5

5. In order to configure PhpMyadmin follow the following instructions:

apt-get -y install phpmyadmin

During the installation process follow the instruction appear on the screen:

Select option **Apache2** and use the appropriate login and password as already entered previously.

6. After Configuration of Apache over:

open the following file : **sudo /etc/apache2/apache2.conf**

Add the following to the bottom of the file **apache2.conf**:

Include /etc/phpmyadmin/apache.conf

7. Restart Server

```
sudo /etc/init.d/apache2 restart
```

8. Check Apache

Open a web browser and navigate to <http://localhost/>. You should see a message saying It works!

9. In case myPHP Apache not works execute the below command:

```
sudo dpkg --reconfigure Apache restart
```

A) Introduction to SQL:

SQL (Structured Query Language) is the standard for relational database management systems (RDBMS) .It is an ANSI (American National Standards Institute) standard, interactive and programming language for getting information from database and updating it. Queries take the form of a command language that lets you select, insert, update, find out the location of data and so forth. SQL is not case sensitive

What can SQL do:-

- It lets you access and manipulate databases.
- It can execute queries against a database
- It can retrieve data from a database
- It can insert records in a database
- It can update records in a database
- It can delete records from a database
- It can create new databases
- It can create new tables in a database
- It can create stored procedures in a database
- It can create views in a database
- It can set permissions on tables, procedures, and views

A) Getting Familiar with SQL

1) SQL Environment:

- **Catalog** - A set of schemas that, when put together, constitute the description of a database. In figure two version of catalog shown Production (live) version and development version.
- **Schema**- The structure that contains descriptions of objects created by a user (base tables, views, constraints)
- **Data Definition Language (DDL)**- Commands that define a database, including creating, altering, and dropping tables and establishing constraints
- **Data Manipulation Language (DML)** -Commands that maintain and query a database. DML is used for inserting, deleting and updating data in a database.
- **Data Control Language (DCL)**-Commands that control a database, including administering privileges and committing data

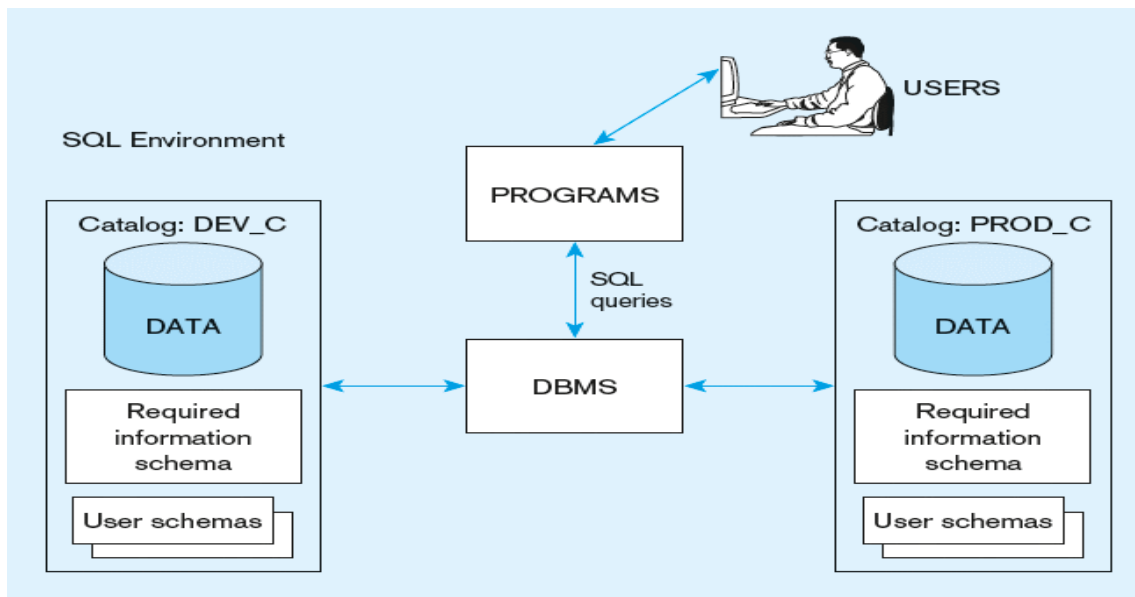
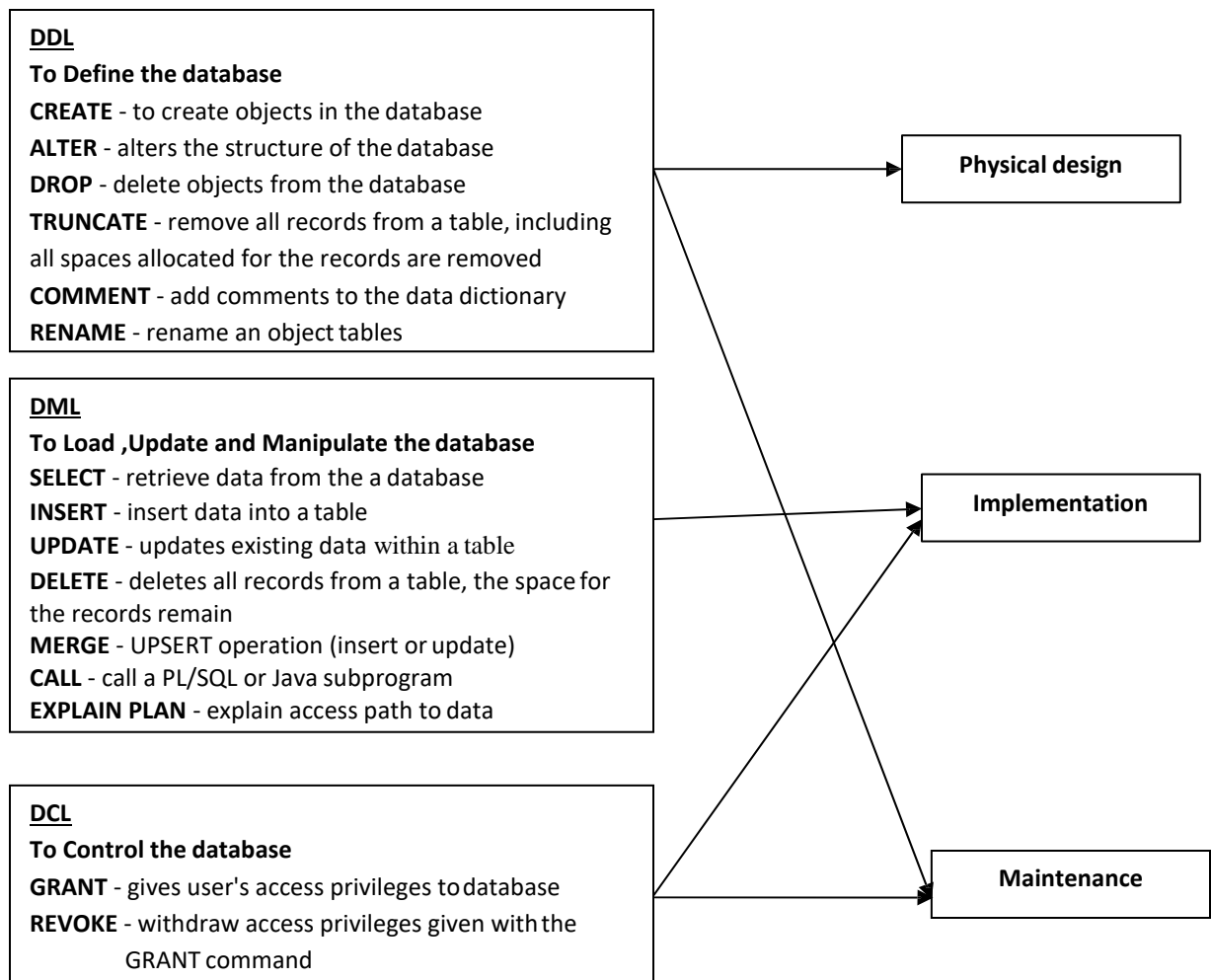


Fig. A simplified schematic of a typical SQL environment

The following table gives a brief overview for the different types of languages (i.e. DDL, DML, and DCL) as mentioned above and the database development process:



2) List of few SQL Data types:

| DATA TYPE | FORMAT | DESCRIPTION |
|-----------------|--|--|
| String types | CHAR(n) VARCHAR(n) VARCHAR2(n) LONG | <ul style="list-style-type: none">fixed length character data , n characters long, Maximum length=2000 bytesvariable length character data , maximum 4000 bytesVariable length character data, up to 4 GB. Maximum 1 |
| Numeric types | FLOAT(m [,n]) INT (b) | <ul style="list-style-type: none">Here, m is the total number of digits in the number and n signifies number of decimal digits.A normal-sized integer that can be signed or unsigned |
| Date/Time types | DATE TIME DATETIME | <ul style="list-style-type: none">Holds fixed-length date/time in 'YYYY-MM-DD.' format.Displayed as 'HH:MM:SS'.Displayed as 'yyyy-mm-dd hh:mm:ss'. |

Note: - Above data types are used more frequently. There are many other data types in MySQL apart from those mentioned above.

Explore more about data types of MySQL @ <https://dev.mysql.com/doc/refman/8.0/en/data-types.html>

3) How to executing basic queries on MySQL

- a) First, we will list all the objects, available in your database by executing the following SQL query :-

SHOW DATABASES;

SHOW SCHEMAS;

This will list all the tables, views, and synonyms etc. that are available to you.

b) Create database `database_name`:

```
CREATE DATABASE database_name;    //to create a database named database_name  
// Replace database_name with the name of the database that you want to create.
```

c) Now you can work with the new created database above:

```
USE databse_name; //database already created above;
```

- d) Now, You create a table using CREATE statement by executing the following query:-

```
CREATE TABLE students  
( Idno CHAR (12),  
  Name VARCHAR (15),  
  DoB DATE,  
  CGPA float (4, 2),  
  Age int (3)  
);
```

A **student** table created having columns as below with empty data values:

| students | | | | |
|----------|------|-----|------|-----|
| Idno | Name | DoB | CGPA | Age |
| | | | | |

- e) To view table structure ie. all of the fields, definition and formats of the created table

DESCRIBE table-name;

It will display all the **meta-data** of the created table.

To see all of the fields and their formats of the table **students** execute:

DESCRIBE student;

//carefully observe the meta-data of the **students** table

- f) Now if we want to insert data into the table ,you can do this using **INSERT** statement as follows:-

**INSERT INTO students (Idno,Name,DoB,CGPA,Age)
VALUES ('2015H112151P','Sagun Sinha', '1985-08-21',8.5,26);**

Caution: Don't execute same INSERT statement more than once, otherwise same data values will be inserted into the table as many times you execute it.

- g) To find out what tables the selected database contains

SHOW TABLES;

To retrieve all data records/rows from a table execute:

SELECT * FROM table-name;

Now, execute the statement below and observe the outcomes:

SELECT * FROM students;

Task 1: Similar to (3.d) above, now you insert few more records in the **students** table whose data values are given below.

| Idno | Name | DoB | CGPA | Age |
|--------------|------------|------------|------|-----|
| 2006HS12559P | Jagat | 1989-08-21 | 8.51 | 23 |
| 2006PH12998G | Mohan Rao | 1991-10-17 | 6.75 | 21 |
| 2012C6PS777H | Prem Chand | 1995-03-28 | 10.0 | 18 |

You can also insert the values inside the table by the following insert statement without mentioning the field names as given below.

INSERT INTO students VALUES ('2010H112151G','maduri Gupta', '1978-07-31',5.5,26);

Caution: Whenever, you are inserting data values this way, you must ensure that you give the field values in proper order, i.e. order in which you have created in the table.

Now, execute below statement and observe the outcomes for correct insertions:

SELECT * FROM students;

There is, yet another way to insert values **in an interactive way**. Try the following query and give the field values when prompted for.

**INSERT INTO students (idno,name,dob,cgpa,age)
VALUES ('&idno','&name','&dob','&cgpa','&age);**

h) To **add new column** in the table such as phone number

```
ALTER TABLE students
ADD (PhoneNo NUMBER(12));
```

Execute below statement and observe whether the new column added to the table or not:

```
SELECT * FROM students;
```

Now, execute **DESCRIBE students;** //To observe the description of the students table.

Task 2: i) Insert into students table one student record with his phone number. Check whether it is inserted into table?

ii) Insert into students table one row having the student name larger than 15 characters and observe whether this record goes into table or not? Why?

i) To modify any existing column in the table such as the **Name** from 15 char to 30 char longer execute:

```
ALTER TABLE students MODIFY (Name VARCHAR(30));
```

Now, execute **DESCRIBE students;** //observe the description of students table.

Execute **SELECT * FROM students;** //observe output data

Now, repeat **Task 2(ii)** again and observe the outcome.

Now suppose we want to retrieve data from the table students, then there are different possibilities:

a) To retrieve specific column data from the table then following SQL statement will be executed:-

```
SELECT name,age,PhoneNo, FROM students;
```

This will list specific columns i.e. Name, Age, PhoneNo only from student table.

b) To retrieve a group of rows w.r.t. a particular value of an attribute then below statement will be executed:

i) **SELECT * FROM students WHERE age>21;**

ii) **SELECT * FROM students WHERE name= 'Sagun Sinha';**

Task 3: Now run above two statements which list only few selected columns instead of all column of students table.

a) If there are duplicate rows or column in the table, then, to retrieve only the distinct values following statement will be executed:

```
SELECT DISTINCT Age FROM students;
```

Now, execute: **SELECT Age FROM students;** //observe difference to previous outcomes.

b) To retrieve data of a table in a sorted order, try these:

```
SELECT name,ID FROM students ORDER BY name;
```

```
SELECT name,ID FROM students ORDER BY name DESC;
```

```
SELECT name,age,cgpa FROM students ORDER BY name DESC, age ASC;
```

TASK:

- **Explore and read more about the data types and their syntax available in MySQL.**
[You can also read more about the data types from Help ==> Search in the Help menu]
- Insert into students table
 - One student record with his phone number larger than 12 digits. Observe whether this record goes into table or not? Why?
 - One student record with his CGPA larger than 4 digits as specified during the duration table has created. Observe the outcome of your query?
- Create a table **EMPLOYEE** (*Eid,Ename,dob,Degree,Pay,Department,Address*)
 - List meta-data description of the table.
 - Populate the table with few sample data records.
 - Execute different types of select queries to retrieve data from table.