

1. Introduction to MySQL

MySQL is the most popular Open Source Relational SQL database management system. MySQL is one of the best RDBMS being used for developing web based software applications.

1.1 What is 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. Other kinds of data stores can 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. So now a days we use relational database management systems (RDBMS) to store and manager huge volume of data. This is called relational database because all the data is stored into different tables and relations are established using primary keys or other keys known as foreign keys.

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.

1.2 RDBMS Terminology:

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

1.3 Advantages of MYSQL

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.

Execute simple SQL commands using MySQL Client:

You can connect to your MySQL server by using MySQL client using mysql command. At this moment you do not need to give any password as by default it will be set to blank.

So just use following command

```
sudo /opt/lampp/bin/mysql -u root
[sudo] password for davinci:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 1
Server version: 5.1.41 Source distribution

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

It should be rewarded with a mysql> prompt. Now you are connected to the MySQL server and you can execute all the SQL command at mysql> prompt as follows.

```
mysql> show databases;
+-----+
| Database          |
+-----+
| information_schema |
| cdcoll             |
| mysql              |
| phpmyadmin         |
| test               |
+-----+
5 rows in set (0.11 sec)

mysql>
```

Post-installation Steps:

MySQL ships with a blank password for the root MySQL user. As soon as you have successfully installed the database and client you need to set a root password as follows:

```
[root@host]# mysqladmin -u root password "new_password";
```

Now to make a connection to your MySQL server you would have to use following command:

```
[root@host]# mysql -u root -p
Enter password:*****
```

Unix users will also want to put your MySQL directory in your PATH, so you won't have to keep typing out the full path every time you want to use the command-line client. For bash, it would be something like:

```
export PATH=$PATH:/usr/bin:/usr/sbin
```

Running MySQL at boot time:

If you want to run MySQL server at boot time then make sure you have following entry in /etc/rc.local file
/etc/init.d/mysqld start

Also, you should have mysqld binary in /etc/init.d/ directory.

Running and Shutting down MySQL Server:

First check if your MySQL server is running or not. You can use following command to check this:

```
ps -ef | grep mysqld
```

If your MySQL is running then you will see mysqld process listed out in your result. If server is not running then you can start it by using following command:

```
root@hoat# cd /usr/bin  
./safe_mysqld &
```

Now if you want to shutdown an already running MySQL server then you can do it by using following command:

```
root@hoat# cd /usr/bin  
./mysqladmin -u root -p shutdown  
Enter password: *****
```

Administrative MySQL Command:

Here is the list of important MySQL command which you will use time to time to work with MySQL database:

USE Databasename : This will be used to select a particular database in MySQL workarea.

SHOW DATABASES: Lists the databases that are accessible by the MySQL DBMS.

SHOW TABLES: Shows the tables in the database once a database has been selected with the use command.

SHOW COLUMNS FROM tablename: Shows the attributes, types of attributes, key information, whether NULL is permitted, defaults, and other information for a table.

SHOW INDEX FROM tablename: Presents the details of all indexes on the table, including the PRIMARY KEY.

SHOW TABLE STATUS LIKE tablename\G: Reports details of the MySQL DBMS performance and statistics.

MySQL PHP Syntax

MySQL works very well in combination of various programming languages like PERL, C, C++, JAVA and PHP. Out of these languages, PHP is the most popular one because of its web application development capabilities.

PHP provides various functions to access MySQL database and to manipulate data records inside MySQL database. You would require to call PHP functions in the same way you call any other PHP function.

The PHP functions for use with MySQL have the following general format:

```
mysqli_function(value,value,...);
```

The second part of the function name is specific to the function, usually a word that describes what the function does. The following are two of the functions which we will use in our tutorial
mysqli_connect(\$connect);

```
mysqli_query($connect,"SQL statement");
```

Following example shows a generic syntax of PHP to call any MySQL function.

```
<html>
<head>
<title>PHP with MySQL</title>
</head>
<body>
<?php
$retval = mysqli_function(value, [value,...]);
if( !$retval )
{
die ( "Error: a related error message" );
}
// Otherwise MySQL
or PHP Statements
?>
</body>
</html>
```

Starting from next chapter we will see all the important MySQL functionality along with PHP.

2 Managing databases in MySQL

2.1 Creating Databases

To create database in MySQL, you use the CREATE DATABASE command as follows:

```
CREATE DATABASE [IF NOT EXISTS] database_name;
```

CREATE DATABASE command will create the database with the given name you specify. The IF NOT EXISTS is an optional parameter. The IF NOT EXISTS part prevented the user from error if there is a database with the given name exists already.

For example to create a student database, give the command like this:

```
CREATE DATABASE student;
```

You can see the following statements on the screen:

```
sudo /opt/lampp/bin/mysql -u root
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 1
Server version: 5.1.41 Source distribution

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

2.2 Showing Databases

Show database statement shows all databases in your database server. For example

```
SHOW DATABASES;
```

The output is :

```

davinci@Monalisa: ~
davinci@Monalisa:~$ sudo /opt/lampp/bin/mysql -u root
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 2
Server version: 5.1.41 Source distribution

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> clear;
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql      |
| phpmyadmin  |
| test       |
+-----+
5 rows in set (0.04 sec)

mysql>

```

2.3 selecting database to work with

To select a database which you plan to work with, you can use 'USE' statement as follows:

```
USE databasename;
```

Now execute the statement in the terminal as:

```

davinci@Monalisa: ~
mysql> create database student;
Query OK, 1 row affected (0.01 sec)

mysql> use student
Database changed
mysql>

```

2.4 Removing Database

Removing a database means removal physically. All the data and related objects inside the database are permanently deleted and cannot be undone. So be careful in using it.

The command used for this is :

```
DROP DATABASE [IF EXISTS] database_name;
```

Like CREATE DATABASE statement, IF EXISTS part is an optional part to prevent you from removing database which does not exist.

3. WORKING WITH DATABASE TABLES

SQL (Structured Query Language) is a database language designed for managing data in relational database management system.

SQL statements can be categorized into

- a) Data Definition Language(DDL)
- b) Data Manipulation Language (DML)

3.1 Data Definition Language (DDL)

The DDL manages table and index structure. The most basic items of DDL are the CREATE, ALTER, RENAME, DROP and TRUNCATE statements.

3.1.1 CREATING TABLES.

To create table we use the CREATE TABLE statement. The typical form of SQL CREATE TABLE statement is as follows:

```
CREATE TABLE [IF NOT EXISTS] table_name(column_list, ...)
```

- MySQL supports IF NOT EXISTS after CREATE TABLE statement to prevent you from error of creating table which already exists on the database server.
- table_name is the name of table you would like to create. After that, you can define a set of columns which is usually in this form: column_name data_type(size) [NOT] NULL

```
mysql> create table if not exists st_table(student_id int(11) not null, st_name varchar(20) not null, age int(11), dept_id int(11), PRIMARY KEY (student_id));
```

```
Query OK, 0 rows affected (0.13 sec)
```

First, you specify table name st_table after CREATE TABLE statement. Then you list the columns of the table with their attributes such as data type, size, NOT NULL. And finally you specify the primary key of the table; in this case the primary key is student_id if the table has more than one primary key, you can separate them by a comma.

SHOWING AND DESCRIBING TABLES IN A DATABASE

In order to show all tables in a database, you use SHOW TABLES statement. By executing this MySQL will return all tables name in the current selected database you are working with.

To see the metadata of the table you have created use DESCRIBE command as follows:

```
DESCRIBE table_name;
```

To see the metadata of st_table we have created execute the statement

```
DESCRIBE st_table;
```

see the example below:

```
mysql> describe st_table
-> ;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| student_id | int(11)       | NO   | PRI | NULL    |       |
| st_name    | varchar(20)   | NO   |     | NULL    |       |
| age       | int(11)       | YES  |     | NULL    |       |
| dept_id   | int(11)       | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.04 sec)

mysql> █
```

ALTERING TABLE STRUCTURES

To alter existing table use ALTER TABLE

```
ALTER [IGNORE] TABLE table_name options [, options..]
options are:
ADD [COLUMN] create_definition [ FIRST | AFTER col_name]
or ADD [COLUMN] (create_definition, create_definition,...)
or ADD INDEX [index_name](index_col_name,...)
or ADD PRIMARY KEY (index_col_name,...)
or ADD UNIQUE [index_name](index_col_name,...)
or ADD FULLTEXT [index_name](index_col_name,...)
or ADD [CONSTRAINT symbol] FOREIGN KEY [index_name](index_col_name,...)
[reference_definition]
or ALTER[COLUMN] col_name {SET DEFAULT literal | DROP DEFAULT}
or CHANGE [ COLUMN] old_col_name create_definition [FIRST | AFTER col_name]
or MODIFY [COLUMN] create_definition [FIRST | AFTER col_name]
or DROP [COLUMN] col_name
or DROP PRIMARY KEY
or DROP INDEX index_name
or DISABLE KEYS
or ENABLE KEYS
or RENAME [TO] new_table_name
or ORDER BY col_name
or table_options
```

- ⤴ The CHANGE and MODIFY are the same, they allow you to change the definition of the column or its position in the table.
- ⤴ The DROP COLUMN will drop the column from the table permanently, if the table contains data, all the data of that column will be lost.
- ⤴ The RENAME clause allows to change the table name to the new one.

DELETING TABLES

To delete a table use the command DROP TABLE. You can delete multiple tables also. IF EXISTS option used to prevent from deleting table which does not exist.

```
DROP TABLE [IF EXISTS] table_name [table_name...]
```

EMPTY TABLE'S DATA

In some cases, you want to delete all the table in a fast way. This can be done by using TRUNCATE command as follows:

```
TRUNCATE TABLE table_name
```

DATA MANIPULATION LANGUAGE

This is used to query, add, delete and update the data in the table.

QUERYING DATA FROM DATABASE TABLES IN MYSQL

In order to retrieve data from MYSQL database server use SQL SELECT statement. The syntax is

```
SELECT column_name1, column_name2,..
FROM tables
[WHERE conditions]
```

```
[GROUP BY group]
[HAVING group_conditions]
[ORDER BY sort_columns]
[LIMIT limits];
```

The SQL SELECT statement has many optional parts which you can use. FROM, WHERE, GROUP BY, HAVING, ORDER BY and LIMIT has to be in the order above.

To select all columns in a table you can use (*) notation instead of listing all column name. For example, if you need to query all the columns in st_table , just execute this query:

```
SELECT * FROM st_table;
```

The output is given below:

```
mysql> select * from st_table
-> ;
+-----+-----+-----+-----+
| student_id | st_name | age | dept_id |
+-----+-----+-----+-----+
|          100 | ram     | 25 | 1       |
|          101 | krish  | 25 | 1       |
|          103 | ganesh  | 25 | 2       |
|          105 | lalitha | 19 | 3       |
|          106 | radha   | 20 | 3       |
|          107 | kalaivani | 21 | 4       |
+-----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql>
```

The SQL SELECT statement also allows you to view some partial data of a table by listing columns name after the SELECT keyword. For example if you need to view only student_id and dept_id of all students you can give like this:

```
SELECT student_id, dept_id from st_table;
```

The output is given below:

```
mysql> select student_id,dept_id from st_table;
+-----+-----+
| student_id | dept_id |
+-----+-----+
|          100 | 1       |
|          101 | 1       |
|          103 | 2       |
|          105 | 3       |
|          106 | 3       |
|          107 | 4       |
+-----+-----+
6 rows in set (0.00 sec)

mysql>
```

WHERE clause:

The WHERE clause of the SELECT statement retrieve particular rows from the table which match its conditions or search criteria. You use WHERE clause to filter the records you want to see. For example to find the students list belonging to department id=1

```
SELECT * from st_table where dept_id=1;
```



```
Database changed
mysql> select * from st_table where dept_id=1;
+-----+-----+-----+-----+
| student_id | st_name | age | dept_id |
+-----+-----+-----+-----+
|          100 | ram    | 25 |        1 |
|          101 | krish  | 25 |        1 |
+-----+-----+-----+-----+
2 rows in set (0.01 sec)

mysql> █
```

SORTING RESULT WITH ORDER BY

The ORDER BY clause allows you to sort the result set on one or more columns in ascending or descending order. To sort the result set in ascending order you use ASC and in descending order you use DESC keywords. By default, the ORDER BY will sort the result set in ascending order. For example to sort the student name in ascending order, execute the command:

```
SELECT * FROM st_table ORDER BY st_name;
```

```
mysql> mysql> select * from st_table order by st_name;
+-----+-----+-----+-----+
| student_id | st_name | age | dept_id |
+-----+-----+-----+-----+
|          103 | ganesh  | 25 |        2 |
|          107 | kalavani | 21 |        4 |
|          101 | krish   | 25 |        1 |
|          105 | lalitha | 19 |        3 |
|          106 | radha   | 20 |        3 |
|          100 | ram     | 25 |        1 |
+-----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql> █
```

DISTINCT

With DISTINCT keyword, you can eliminate the duplicated result from the SELECT statement.

```
SELECT DISTINCT column_name, FROM table_name;
```

RETRIEVING DATA IN A RANGE

SQL BETWEEN allows you to retrieve values within a specific range. The usage of SQL BETWEEN is as follows:

```
SELECT column_list FROM table_name WHERE column_1 BETWEEN lower_range AND upper_range;
```

For example the command

```
SELECT * FROM st_table where student_id between 101 and 103;
```

The above command will extract the following records as in figure below:

```
mysql> select * from st_table where student_id between 101 and 103;
+-----+-----+-----+-----+
| student_id | st_name | age | dept_id |
+-----+-----+-----+-----+
|          101 | krish   | 25  |        1 |
|          103 | ganesh  | 25  |        2 |
+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql> █
```

You can use the following command to get the same result

```
SELECT column_list FROM table_name WHERE column_1 >=lower_range AND column_1 <=upper_range
```

```
SELECT * FROM st_table WHERE student_id >=101 AND student_id<=103;
```

```
administrator@ubuntu: /
+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql> select * from st_table where where student_id >=101 and student_id<=103;
+-----+-----+-----+-----+
| student_id | st_name | age | dept_id |
+-----+-----+-----+-----+
|          101 | krish   | 25  |        1 |
|          103 | ganesh  | 25  |        2 |
+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

USING SQL LIKE OPERATION

SQL LIKE allows you to perform pattern matching in your column in a database table. SQL LIKE is often used with SELECT statement in WHERE clause. MySQL provides you two wildcard characters for use with LIKE, the percentage % and underscore_.

- ⤴ Percentage (%) wildcard allows you to match any string of zero or more characters.
- ⤴ Underscore(_) allows you to match any single character.

MYSQL scans the whole sttable and find matching for the like characters and returns the result as in figure.

```
mysql> select st_name from st_table where st_name like 'k%';
+-----+
| st_name |
+-----+
| krish   |
| kalaivani |
+-----+
2 rows in set (0.02 sec)

mysql> select st_name from st_table where st_name like 'k_';
Empty set (0.00 sec)

mysql> select dept_id from st_table where dept_id like '1_';
Empty set (0.00 sec)

mysql> select st_name from st_table where st_name like 'k%';
+-----+
| st_name |
+-----+
| krish   |
| kalaivani |
+-----+
2 rows in set (0.00 sec)

mysql> select st_name from st_table where student_id like '_02';
Empty set (0.00 sec)

mysql> select st_name from st_table where student_id like '_01';
+-----+
| st_name |
+-----+
| krish   |
+-----+
1 row in set (0.00 sec)

mysql>
```

```
mysql> insert into dept values(1,'csc');
Query OK, 1 row affected (0.05 sec)

mysql> insert into dept values(2,'physics');
Query OK, 1 row affected (0.05 sec)

mysql> insert into dept values(3,'chemistry');
Query OK, 1 row affected (0.05 sec)

mysql> insert into dept values(4,'maths');
Query OK, 1 row affected (0.16 sec)

mysql> select * from dept;
+-----+-----+
| dept_id | dept_name |
+-----+-----+
| 1       | csc       |
| 2       | physics  |
| 3       | chemistry |
| 4       | maths    |
+-----+-----+
4 rows in set (0.00 sec)

mysql>
```

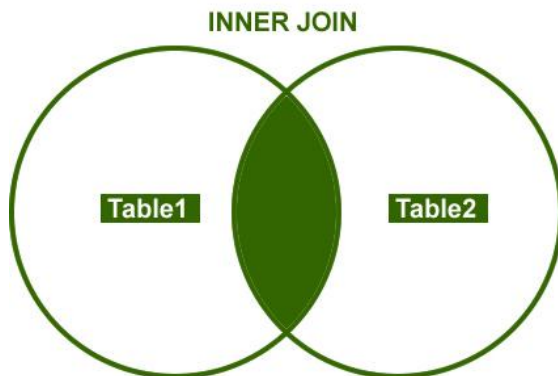
SQL Joins

SQL Joins are used to relate information in different tables. A Join condition is a part of the sql query that retrieves rows from two or more tables. A SQL Join condition is used in the SQL WHERE Clause of select, update, delete statements.

SQL contains different types of Joins we will see each concept with example by using above tables.

INNER JOIN

This join returns rows when there is at least one match in both the tables.



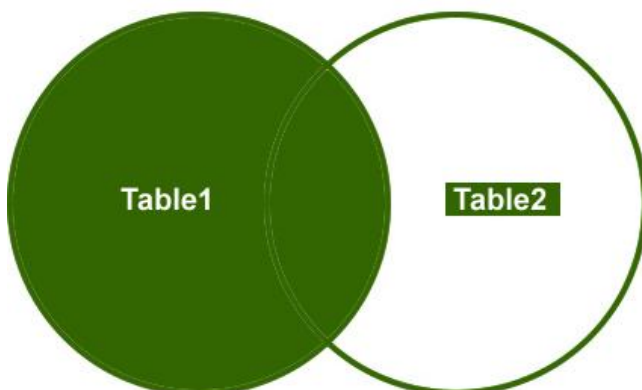
Example:

```
at line 1
mysql> clear
mysql> select st_table.student_id,st_table.st_name,dept.dept_name
-> from st_table
-> inner join dept
-> on st_table.dept_id=dept.dept_id;
+-----+-----+-----+
| student_id | st_name | dept_name |
+-----+-----+-----+
| 100 | ran | csc |
| 101 | krish | csc |
| 103 | ganesh | physics |
| 105 | lalitha | chemistry |
| 106 | radha | chemistry |
| 107 | kalaivani | maths |
+-----+-----+-----+
6 rows in set (0.00 sec)
mysql>
```

OUTER JOIN

There are three different Outer Join methods.

LEFT OUTER JOIN: This join returns all the rows from the left table in conjunction with the matching rows from the right table. If there are no columns matching in the right table, it returns NULL values



Example

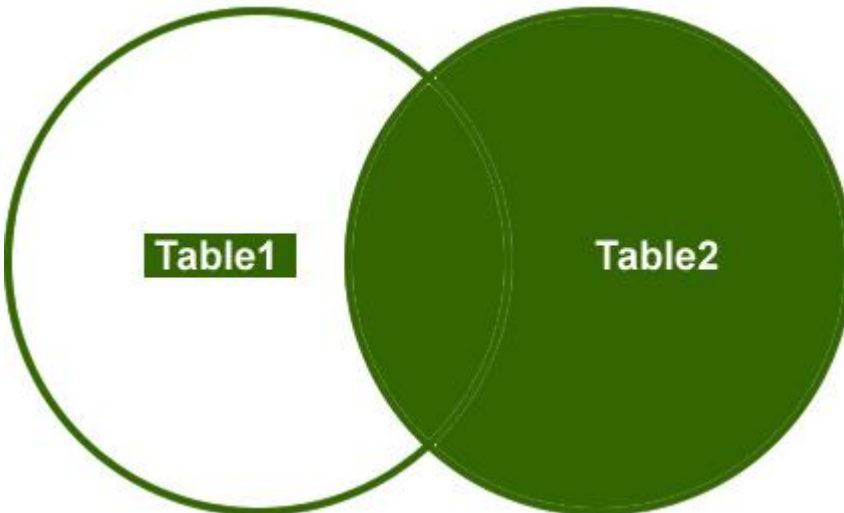
```
mysql> select st_table.student_id,st_table.st_name,dept.dept_name,dept.dept_id
+-----+-----+-----+-----+
| student_id | st_name | dept_name | dept_id |
+-----+-----+-----+-----+
| 107 | kalaivani | maths | 4 |
+-----+-----+-----+-----+
6 rows in set (0.03 sec)

mysql> insert into st_table values(120,'parvathi',20,7);
Query OK, 1 row affected (0.06 sec)

mysql> select st_table.student_id,st_table.st_name,dept.dept_name,dept.dept_id
-> from st_table
-> left join dept
-> on st_table.dept_id=dept.dept_id;
+-----+-----+-----+-----+
| student_id | st_name | dept_name | dept_id |
+-----+-----+-----+-----+
| 100 | ram | csc | 1 |
| 101 | krish | csc | 1 |
| 103 | ganesh | physics | 2 |
| 105 | lalitha | chemistry | 3 |
| 106 | radha | chemistry | 3 |
| 107 | kalaivani | maths | 4 |
| 120 | parvathi | NULL | NULL |
+-----+-----+-----+-----+
7 rows in set (0.00 sec)

mysql>
```

RIGHT OUTER JOIN: This join returns all the rows from the right table in conjunction with the matching rows from the left table. If there are no columns matching in the left table, it returns NULL values.

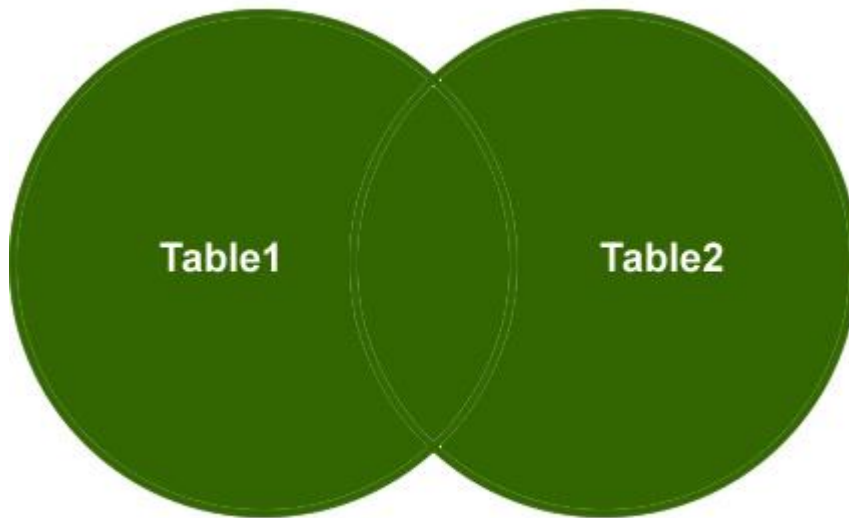


```
mysql> select st_table.student_id,st_table.st_name,dept.dept_name,dept.dept_id
+-----+-----+-----+-----+
| student_id | st_name | dept_name | dept_id |
+-----+-----+-----+-----+
| 100 | ram | csc | 1 |
| 101 | krish | csc | 1 |
| 103 | ganesh | physics | 2 |
| 105 | lalitha | chemistry | 3 |
| 106 | radha | chemistry | 3 |
| 107 | kalaivani | maths | 4 |
| NULL | NULL | economics | 5 |
| NULL | NULL | bba | 10 |
+-----+-----+-----+-----+
8 rows in set (0.00 sec)

mysql>
```

FULL OUTER JOIN: This join combines left outer join and right outer join. It returns row from either table when the conditions are met and returns null value when there is no match.

FULL OUTER JOIN



```
SELECT *  
FROM Table1 t1  
FULL OUTER JOIN Table2 t2  
ON t1.Col1 = t2.Col1
```

The above result can be achieved with left join and right join union as below:

```
mysql> (select s.student_id,s.st_name,d.dept_name,d.dept_id  
-> from st_table as s  
-> left join dept as d  
-> on s.dept_id=d.dept_id)  
-> union  
-> (select s.student_id,s.st_name,d.dept_name,d.dept_id  
-> from st_table as s  
-> right join dept as d  
-> on s.dept_id=d.dept_id)  
-> ;
```

student_id	st_name	dept_name	dept_id
100	ram	csc	1
101	krish	csc	1
103	ganesh	physics	2
105	lalitha	chemistry	3
106	radha	chemistry	3
107	kalaivani	maths	4
120	parvathi	NULL	NULL
NULL	NULL	economics	5
NULL	NULL	bba	10

SQL- Aggregate functions:

The SQL Aggregate Functions are functions that provide mathematical operations. If you need to add, count or perform basic statistics, these functions will be of great help.

The functions include:

- **count()** - counts a number of rows
- **sum()** - compute sum
- **avg()** - compute average
- **min()** - compute minimum

- **max()** - compute maximum

The AVG() Function

The AVG() function returns the average value of a numeric column.

SQL AVG() Syntax

```
SELECT AVG(column_name) FROM table_name
```

```
mysql> select avg(age) from st_table;
+-----+
| avg(age) |
+-----+
| 22.1429 |
+-----+
1 row in set (0.03 sec)
mysql>
```

SQL COUNT() Function

The COUNT() function returns the number of rows that matches a specified criteria.

SQL COUNT(column_name) Syntax

The COUNT(column_name) function returns the number of values (NULL values will not be counted) of the specified column:

```
SELECT COUNT(column_name) FROM table_name
```

SQL COUNT(*) Syntax

The COUNT(*) function returns the number of records in a table:

```
SELECT COUNT(*) FROM table_name
```

```
mysql> select count(age) from st_table;
+-----+
| count(age) |
+-----+
| 7 |
+-----+
1 row in set (0.00 sec)
mysql>
```

SQL COUNT(DISTINCT column_name) Syntax

The COUNT(DISTINCT column_name) function returns the number of distinct values of the specified column:

```
SELECT COUNT(DISTINCT column_name) FROM table_name
```

```
mysql> select count(distinct age) from st_table;
+-----+
| count(distinct age) |
+-----+
|          4          |
+-----+
1 row in set (0.00 sec)

mysql>
```

The MAX() Function

The MAX() function returns the largest value of the selected column.

SQL MAX() Syntax

SELECT MAX(column_name) FROM table_name

```
mysql> select max(age) from st_table;
+-----+
| max(age) |
+-----+
|        25 |
+-----+
1 row in set (0.00 sec)

mysql>
```

The MIN() Function

The MIN() function returns the smallest value of the selected column.

SQL MIN() Syntax

SELECT MIN(column_name) FROM table_name

```
mysql> select min(age) from st_table;
+-----+
| min(age) |
+-----+
|        19 |
+-----+
1 row in set (0.00 sec)

mysql>
```

The SUM() Function

The SUM() function returns the total sum of a numeric column.

SQL SUM() Syntax

SELECT SUM(column_name) FROM table_name


```
mysql> select sum(age) from st_table;
+-----+
| sum(age) |
+-----+
|      155 |
+-----+
1 row in set (0.00 sec)

mysql>
```

Connecting to MYSQL

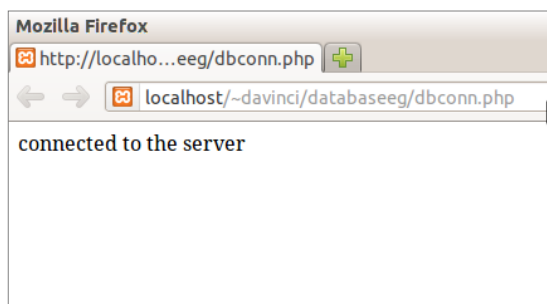
to use any database in PHP – and in other programming languages as well – you first need to connect to the database server. Connecting to a MYSQL database in PHP requires the following information. You need to get the following data to establish the connection:

- ^ hostname
- ^ mysql database username
- ^ mysql database password
- ^ database name

After connecting to the MySQL database server, you need to select the required database. Of course if the connection to their server failed then it makes no sense to continue the script so we can exit.

```
<?php
$db_handle=mysqli_connect("localhost","root","","student")
    or die("unable to connect to database");
echo "connected to the server";
?>
```

As you can see, the `mysqli_connect` function establishes the connection to the MySQL server and selects the database. On execution you will get the following output:



Closing the Function

The connection will be closed automatically when the script ends. To close the connection before, use the `mysqli_close()` function.

```
mysqli_close($db_handle);
```

Retrieving data from the database

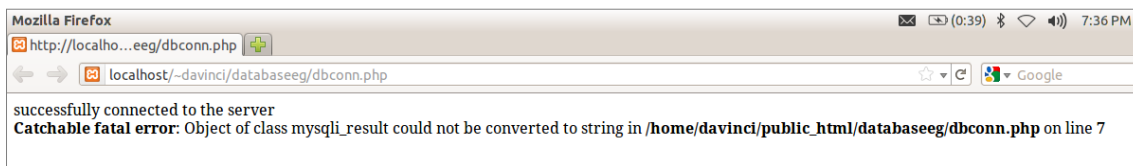
The steps to be followed are:

- 1) open the connection using `mysqli_connect`
- 2) select the respective database.
- 3) Using `SELECT` sql statement and `mysqli_query()` function select the required table.
- 4) Store the result of query and display it as output.

The following program establishes these step

```
<?php
$db_handle=mysqli_connect("localhost","root","","student")
    or die("unable to connect to database");
echo "successfully connected to the server";
$query = "select * from st_table";
$result_table=mysqli_query($db_handle,$query) or die("unable to execute ". $query);
echo $result_table;
?>
```

The above program is not enough for printing the output. When you execute the above program you will get the following output



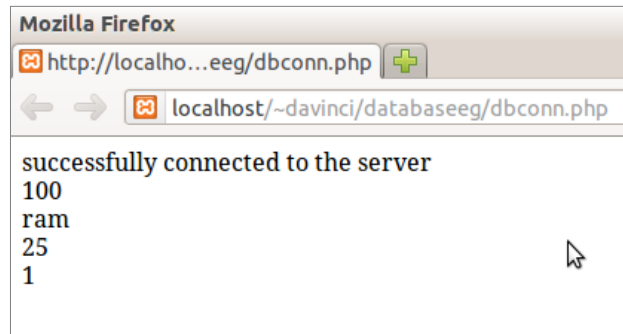
The above steps are enough for getting result from mysql server. But you need more functions to display the result in a proper format. They are:

- ♣ `mysqli_fetch_array()`- Fetch a result row as an associative array.
- ♣ `mysqli_fetch_row()` - Fetch a result row as an enumerated array (numeric)
- ♣ `mysqli_fetch_array()` - Fetch a result row as an associative array, a numeric array, or both

All the above three functions convert one record of result to an array and later you can use this array as you want.

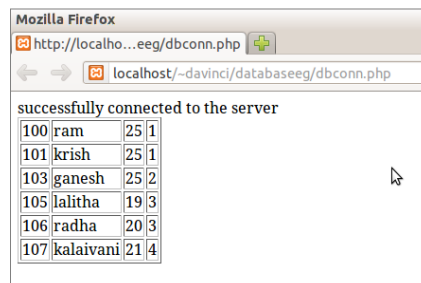
```
<?php
$db_handle=mysqli_connect("localhost","root","","student")
    or die("unable to connect to database");
echo "successfully connected to the server";
$query = "select * from st_table";
$result_table=mysqli_query($db_handle,$query) or die("unable to execute ". $query);
$row=mysqli_fetch_array($result_table);
echo "<br>";
echo $row['student_id']. "<br>";
echo $row['st_name']. "<br>";
echo $row['age']. "<br>";
echo $row['dept_id']. "<br>";
?>
```

The output you can see this:



The above program will extract only one record from data base. This is because the `mysqli_fetch_array()` will extract only one record. In order to extract all the records which are matchin a criteria you have to give within loops. Not only that the above program output is not in a desired format. To display it neetly you can use tables as in the following example

```
<?php
$db_handle=mysqli_connect("localhost","root","","student")
    or die("unable to connect to database");
echo "successfully connected to the server";
$query = "select * from st_table";
$result_table=mysqli_query($db_handle,$query) or die("unable to execute ". $query);
echo "<table border='1'>";
while($row=mysqli_fetch_array($result_table))
{
    echo "<tr>";
    echo "<td>" . $row['student_id']. "</td>";
    echo "<td>". $row['st_name']. "</td>";
    echo "<td>". $row['age']. "</td>";
    echo "<td>". $row['dept_id']. "</td>";
    echo "</tr>";
}
echo "</table>";
?>
```



To display the field name also in the output enter the following as

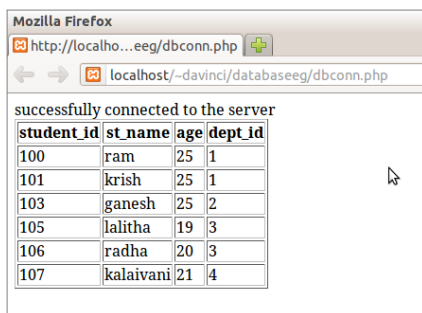
```
$fields=mysqli_fetch_fields($result_table);
foreach($fields as $col)
{
    echo "<th>". $col->name. "</th>";
```

```
}
```

In the above code `mysqli_fetch_fields()` will return the field names. Then we can use `foreach` loop to iterate through the array `$fields` which contains the field name and display them on the output. After entering this the program will be as follows:

```
<?php
$db_handle=mysqli_connect("localhost","root","","student")
    or die("unable to connect to database");
echo "successfully connected to the server";
$query = "select * from st_table";
$result_table=mysqli_query($db_handle,$query) or die("unable to execute ". $query);
$fields=mysqli_fetch_fields($result_table);
echo "<table border='1'>";
foreach($fields as $col)
{
    echo "<th>".$col->name."</th>";
}
while($row=mysqli_fetch_array($result_table))
{
    echo "<tr>";
    echo "<td>". $row['student_id']. "</td>";
    echo "<td>". $row['st_name']. "</td>";
    echo "<td>". $row['age']. "</td>";
    echo "<td>". $row['dept_id']. "</td>";
    echo "</tr>";
}
echo "</table>";
?>
```

The output of the above program is :



student_id	st_name	age	dept_id
100	ram	25	1
101	krish	25	1
103	ganesh	25	2
105	lalitha	19	3
106	radha	20	3
107	kalaivani	21	4

To insert data into database through PHP

```
Insert into st_table values(108,'sumathi',20,4);
```

```
<?php
$db_handle=mysqli_connect("localhost","root","","student")
    or die("unable to connect to database");
echo "successfully connected to the server <br>";
```

```

$query = "Insert into st_table values(108,'sumathi',20,4)";
$result_table=mysqli_query($db_handle,$query) or die("unable to execute ". $query);
if(mysqli_affected_rows($db_handle)==1)
{
    echo " student data inserted successfully";
}
else
{
    echo " unable to insert data<br>";
}
?>

```

The output is :



To find out the result status of the insert statement, we use the function **mysqli_affected_rows()** which returns the number of rows affected by the last INSERT, UPDATE, REPLACE or DELETE query.

If the return value is equal to 1 (since only 1 insert statement is issued in the above example) then the insert got executed successfully.

UPDATING A RECORD

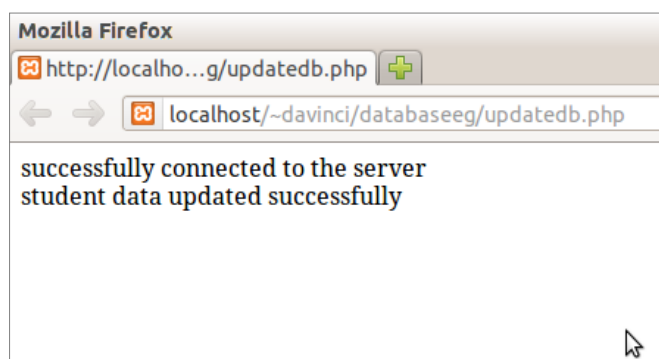
To update an existing data in the database execute the following program:

```

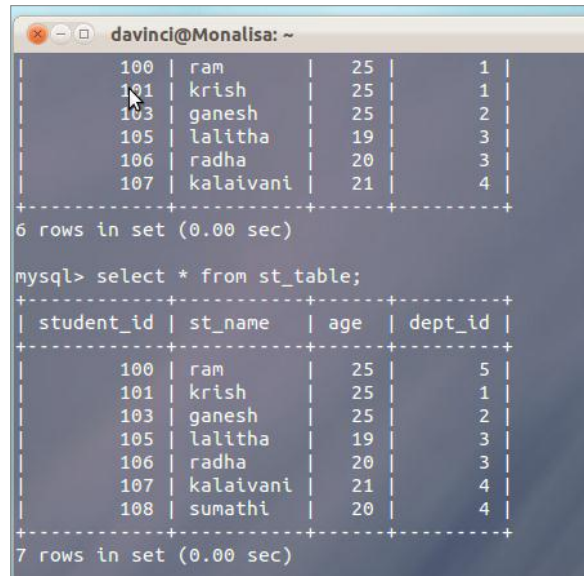
<?php
$db_handle=mysqli_connect("localhost","root","","student")
    or die("unable to connect to database");
echo "successfully connected to the server <br>";
$query = "update st_table set dept_id = 5 where student_id = 100";
$result_table=mysqli_query($db_handle,$query) or die("unable to execute ". $query);
if(mysqli_affected_rows($db_handle)==1)
{
    echo " student data updated successfully";
}
else
{
    echo " unable to update data<br>";
}
?>

```

Now you can check whether the data is updated by running



dbconn.php or in terminal by using select statement as in figure below:



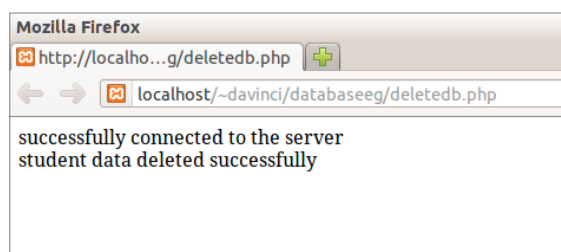
```
mysql> select * from st_table;
+-----+-----+-----+-----+
| student_id | st_name | age | dept_id |
+-----+-----+-----+-----+
| 100 | ram | 25 | 1 |
| 101 | krish | 25 | 1 |
| 103 | ganesh | 25 | 2 |
| 105 | lalitha | 19 | 3 |
| 106 | radha | 20 | 3 |
| 107 | kalaivani | 21 | 4 |
+-----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql> select * from st_table;
+-----+-----+-----+-----+
| student_id | st_name | age | dept_id |
+-----+-----+-----+-----+
| 101 | krish | 25 | 1 |
| 103 | ganesh | 25 | 2 |
| 105 | lalitha | 19 | 3 |
| 106 | radha | 20 | 3 |
| 107 | kalaivani | 21 | 4 |
| 108 | sumathi | 20 | 4 |
+-----+-----+-----+-----+
7 rows in set (0.00 sec)
```

DELETING A RECORD

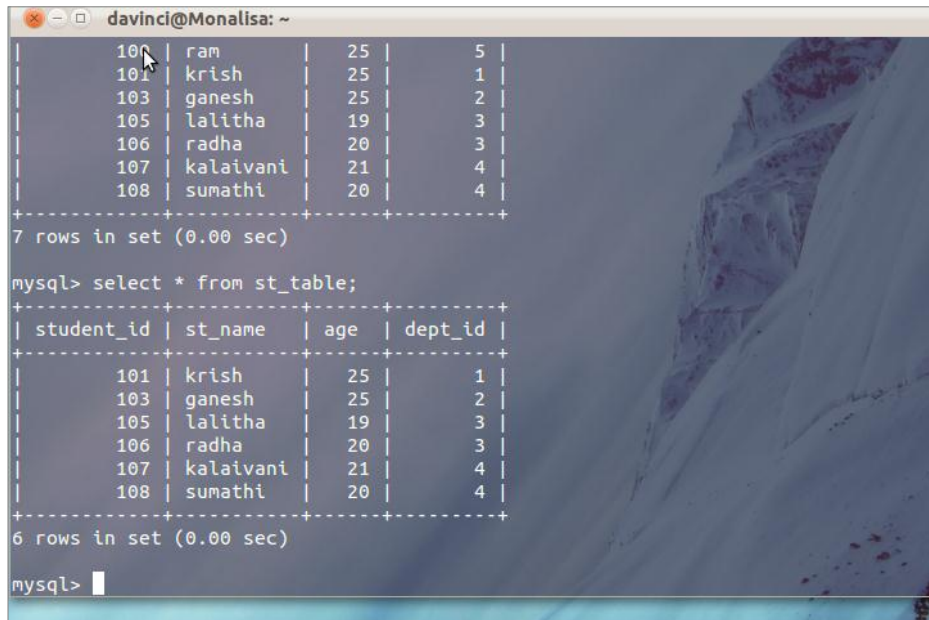
To delete the record

```
<?php
$db_handle=mysqli_connect("localhost","root","","student")
    or die("unable to connect to database");
echo "successfully connected to the server <br>";
$query = "delete from st_table where student_id = 100";
$result_table=mysqli_query($db_handle,$query) or die("unable to execute ". $query);
if(mysqli_affected_rows($db_handle)==1)
{
    echo " student data deleted successfully";
}
else
{
    echo " unable to delete data<br>";
}
?>
```



You can view the data as

you can see that the record with student_id =100 has been deleted in the following figure:



```
davinci@Monalisa: ~  
+-----+-----+-----+-----+  
| 100 | ram      | 25 | 5 |  
| 101 | krish    | 25 | 1 |  
| 103 | ganesh   | 25 | 2 |  
| 105 | lalitha  | 19 | 3 |  
| 106 | radha    | 20 | 3 |  
| 107 | kalaivani | 21 | 4 |  
| 108 | sumathi  | 20 | 4 |  
+-----+-----+-----+-----+  
7 rows in set (0.00 sec)  
  
mysql> select * from st_table;  
+-----+-----+-----+-----+  
| student_id | st_name | age | dept_id |  
+-----+-----+-----+-----+  
| 101 | krish    | 25 | 1 |  
| 103 | ganesh   | 25 | 2 |  
| 105 | lalitha  | 19 | 3 |  
| 106 | radha    | 20 | 3 |  
| 107 | kalaivani | 21 | 4 |  
| 108 | sumathi  | 20 | 4 |  
+-----+-----+-----+-----+  
6 rows in set (0.00 sec)  
  
mysql> |
```

config.php

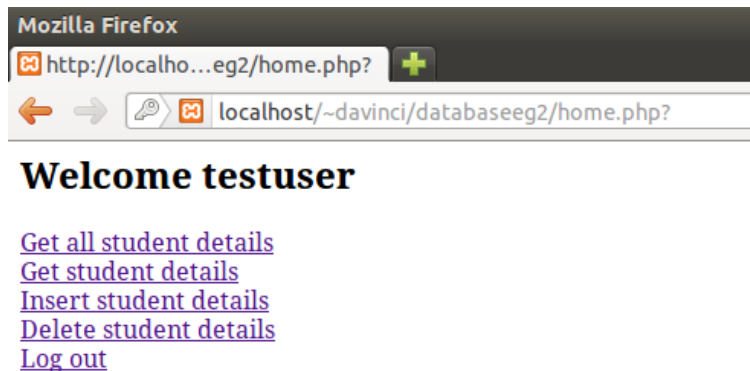
```
<?php  
$host="localhost";  
$db_user="root";  
$db_pass="";  
$db_name="student";  
  
?>
```

authorize.php

```
<?php  
session_start();  
if(isset($_SESSION['isAuth']))  
{  
    $username=$_SESSION['name'];  
    echo "<h2> Welcome $username</h2>";  
}  
else  
{  
    echo "<a href='login.php'> PLEASE LOGIN</a>";  
    exit();  
}  
?>
```

Home.php

```
<?php
require_once 'authorize.php';
?>
<a href="selectdb.php"> Get all student details</a><br>
<a href="selectstudent.php"> Get student details</a><br>
<a href="insertdb.php"> Insert student details</a><br>
<a href="deletedb.php"> Delete student details</a><br>
<a href="logout.php"> Log out</a><br>
```



login.php

```
<?php
if(isset($_POST['submit']))
{
    require_once 'config.php';
    $dbhandle=mysqli_connect($host,$db_user,$db_pass,$db_name)
        or die("unable to connect");
    $username=mysqli_escape_string($dbhandle,$_POST['username']);
    $password=mysqli_escape_string($dbhandle,$_POST['password']);
    if(empty($username)||empty($password))
    {
        exit("please enter username and passwird");
    }
    $query="select * from login where username='$username' and password='$password'";
    $login_result=mysqli_query($dbhandle,$query)
        or die("unable to execute".$query);
    if(mysqli_num_rows($login_result)==1)
    {
        session_start();
        $_SESSION['name']=$username;
        $_SESSION['isAuth']=true;
        header('Location:home.php?.SID);
        exit();
    }
    else
    {
        echo "wrong username/password combination <br>";
    }
}
```



```

}
}

?>
<html>
  <head>
    <title>
      PHP LOGIN FORM
    </title>
  </head>
  <body>
    <form action="<?php echo $_SERVER['PHP_SELF']?>" method="post">
      username:<input type="text" name="username" value=""/><br>
      password:<input type="password" name="password" value=""/><br>
      <input type="submit" name="submit" value="Login"/>
    </form>
  </body>
</html>

```

The output is :

insertdb.php

```

<?php

require_once 'authorize.php';
?>

<?php
if(isset($_GET['submit']))
{
require_once 'config.php';
$dbhandle=mysqli_connect($host,$db_user,$db_pass,$db_name)
  or die("unable to connect");
$sid=mysqli_escape_string($dbhandle,$_GET['sid']);
$name=mysqli_escape_string($dbhandle,$_GET['sname']);
$age=mysqli_escape_string($dbhandle,$_GET['age']);
$deptid=mysqli_escape_string($dbhandle,$_GET['deptid']);
if(!filter_var($sid,FILTER_VALIDATE_INT)
  || !filter_var($age,FILTER_VALIDATE_INT)
  || !filter_var($deptid,FILTER_VALIDATE_INT))

```

```

{
    exit("please enter integer values");
}
$query="insert into st_table values($sid,$sname,$age,$deptid)";

$st_table=mysqli_query($dbhandle,$query)
    or die("unable to execute".$query);
if(mysqli_affected_rows($dbhandle)==1)
{
    echo "student data inserted successfully";
}
else
{
    echo "unable to insert data";
}
mysqli_close($dbhandle);
}

?>
<html>
<head>
<title>
    student details
</title>
</head>
<body>
<form action="<?php echo $_SERVER['PHP_SELF'];>" method="get">
<table>
<tr>
<td>Please enter student id:</td>
<td> <input type="text" name="sid" value=""/></td>
</tr>
<tr>
<td>Please enter student name:</td>
<td> <input type="text" name="sname" value=""/></td>
</tr>
<tr>
<td>Please enter student age:</td>
<td> <input type="text" name="age" value=""/></td>
</tr>
<tr>
<td>Please enter student dept id:</td>
<td> <input type="text" name="deptid" value=""/></td>
</tr>
</table>
<input type="submit" name="submit" value="Insert data"/>

</form>
</body>
</html>

```

Selectdb.php

```
<?php
require_once 'authorize.php';
?>

<?php
if(isset($_GET['submit']))
{
require_once 'config.php';
$dbhandle=mysqli_connect($host,$db_user,$db_pass,$db_name)
    or die("unable to connect");
$sid=mysqli_escape_string($dbhandle,$_GET['sid']);
$name=mysqli_escape_string($dbhandle,$_GET['sname']);
$age=mysqli_escape_string($dbhandle,$_GET['age']);
$deptid=mysqli_escape_string($dbhandle,$_GET['deptid']);
if(!filter_var($sid,FILTER_VALIDATE_INT)
    || !filter_var($age,FILTER_VALIDATE_INT)
    || !filter_var($deptid,FILTER_VALIDATE_INT))
{
    exit("please enter integer values");
}
$query="insert into st_table values($sid,$name,$age,$deptid)";

$stmt=mysqli_query($dbhandle,$query)
    or die("unable to execute".$query);
if(mysqli_affected_rows($dbhandle)==1)
{
    echo "student data inserted successfully";
}
else
{
    echo "unable to insert data";
}
mysqli_close($dbhandle);
}
```

```

?>
<html>
  <head>
    <title>
      student details
    </title>
  </head>
  <body>
    <form action="<?php echo $_SERVER['PHP_SELF']?>" method="get">
      <table>
        <tr>
          <td>Please enter student id:</td>
          <td><input type="text" name="sid" value=""/></td>
        </tr>
        <tr>
          <td>Please enter student name:</td>
          <td><input type="text" name="sname" value=""/></td>
        </tr>
        <tr>
          <td>Please enter student age:</td>
          <td><input type="text" name="age" value=""/></td>
        </tr>
        <tr>
          <td>Please enter student dept id:</td>
          <td><input type="text" name="deptid" value=""/></td>
        </tr>
      </table>
      <input type="submit" name="submit" value="Insert data"/>
    </form>
  </body>
</html>

```

Selectstudent.php

```

<?php
require_once 'authorize.php';
?>
<?php
if(isset($_GET['submit']))
{
require_once 'config.php';
$dbhandle=mysqli_connect($host,$db_user,$db_pass,$db_name)
  or die("unable to connect");
$sid=mysqli_escape_string($dbhandle,$_GET['sid']);
if(!filter_var($sid,FILTER_VALIDATE_INT))
{
  exit("please enter integer");
}
}

```

```

}
$query="select * from st_table where student_id=".$sid;
$stmt=mysqli_query($dbh,$query)
    or die("unable to execute".$query);

$column=mysqli_fetch_fields($stmt);
echo "<table border=1>";
foreach($column as $col)
{
    echo "<th>".$col->name."</th>";
}
while($row=mysqli_fetch_array($stmt))
{
    echo "<tr>";
    echo "<td>".$row['student_id']."</td>";
    echo "<td>".$row['st_name']."</td>";
    echo "<td>".$row['age']."</td>";
    echo "<td>".$row['dept_id']."</td>";
    echo "</tr>";
}
echo "</table>";
mysqli_close($dbh);
exit();
}
?>
<html>
<head>
<title>
student details
</title>
</head>
<body>
<form action="<?php echo $_SERVER['PHP_SELF']?>" method="get">
please enter student id:<input type="text" name="sid" value=""/><br>
<input type="submit" name="submit" value="Get Details"/>

</form>
</body>
</html>

```

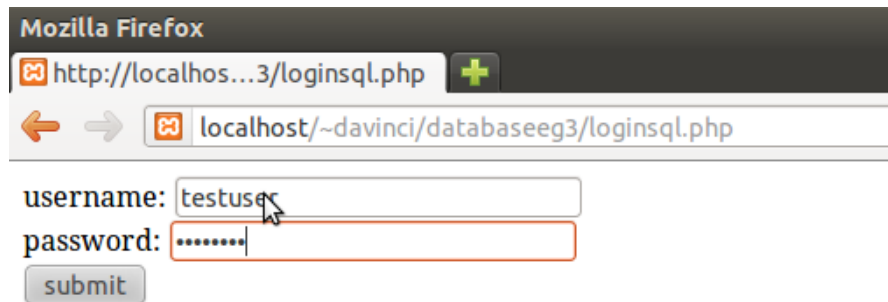
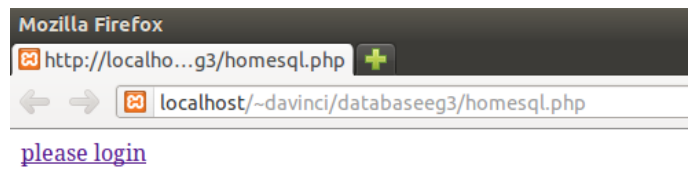
homesql.php

```

<?php
require_once 'authorizesql.php';
?>
<a href="selectsql.php"> get all user details</a><br>
<a href="insertsql.php"> insert new user details</a><br>
<a href="deletesql.php">delete user details</a><br>

```

```
<a href="logoutsq.php"> logout</a><br>
```



Php and mysql with session example

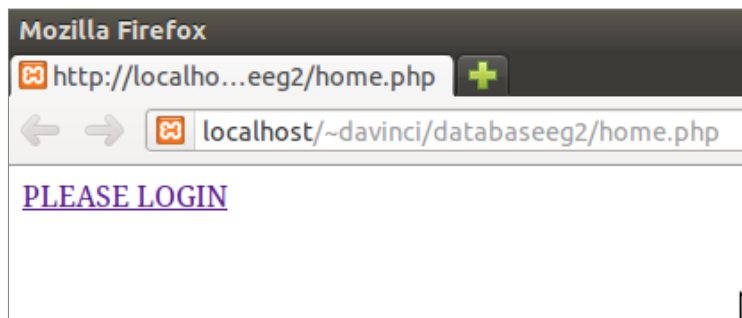
we are going to write a menu like program where you do all the above operation dynamically. In this the connection information is given as separate file and included wherever needed
config.php

This file contains the following codes:

```
<?php
$host="localhost";
$db_user="root";
$db_password="";
$db_name="student";
?>
```

Authorize.php

```
<?php
session_start();
if(isset($_SESSION['isAuth']))
{
    $username=$_SESSION['name'];
    echo "<h2> Welcome $username</h2>";
}
else
{
    echo "<a href='login.php'> PLEASE LOGIN</a>";
    exit();
}
?>
```



Home.php

```
<?php
require_once 'authorize.php';
?>
<a href="selectdb.php"> Get all student details</a><br>
<a href="selectstudent.php"> Get student details</a><br>
<a href="insertdb.php"> Insert student details</a><br>
<a href="updatedb.php"> update student details</a><br>
<a href="delete.php"> Delete student details</a><br>
<a href="logout.php"> Log out</a><br>
```

Login.php

```
<?php
if(isset($_POST['submit']))
{
    require_once 'config.php';
    $dbhandle=mysqli_connect($host,$db_user,$db_pass,$db_name)
        or die("unable to connect");
    $username=mysqli_escape_string($dbhandle,$_POST['username']);
    $password=mysqli_escape_string($dbhandle,$_POST['password']);
    if(empty($username)||empty($password))
    {
        exit("please enter username and passwird");
    }
    $query="select * from login where username='$username' and password='$password'";
    $login_result=mysqli_query($dbhandle,$query)
        or die("unable to execute".$query);
    if(mysqli_num_rows($login_result)==1)
    {
        session_start();
        $_SESSION['name']=$username;
        $_SESSION['isAuth']=true;
        header('Location:home.php?'.SID);
        exit();
    }
    else
    {
```

```

    echo "wrong username/password combination <br>";
}
}

?>
<html>
  <head>
    <title>
      PHP LOGIN FORM
    </title>
  </head>
  <body>
    <form action="<?php echo $_SERVER['PHP_SELF'];" method="post">
      username:<input type="text" name="username" value=""/><br>
      password:<input type="password" name="password" value=""/><br>
      <input type="submit" name="submit" value="Login"/>
    </form>
  </body>
</html>

```

Selectdb.php

```

<?php
require_once 'authorize.php';
?>
<?php
require_once 'config.php';
$dbhandle=mysqli_connect($host,$db_user,$db_pass,$db_name)
    or die("unable to connect");
echo "connected sucesfully";
$query="select * from st_table";
$stmt_table=mysqli_query($dbhandle,$query)
    or die("unable to execute".$query);
$column=mysqli_fetch_fields($stmt_table);
echo "<table border=1>";
foreach($column as $col)
{
    echo "<th>".$col->name."</th>";
}
while($row=mysqli_fetch_array($stmt_table))
{

```



```

echo "<tr>";
echo "<td>".$row['student_id']."</td>";
echo "<td>".$row['st_name']."</td>";
echo "<td>".$row['age']."</td>";
echo "<td>".$row['dept_id']."</td>";
echo "</tr>";

}
echo "</table>";
mysqli_close($dbhandle);

?>

```

Selectstudent.php

```

<?php
require_once 'authorize.php';
?>
<?php
if(isset($_GET['submit']))
{
require_once 'config.php';
$dbhandle=mysqli_connect($host,$db_user,$db_pass,$db_name)
    or die("unable to connect");
$sid=mysqli_escape_string($dbhandle,$_GET['sid']);
if(!filter_var($sid,FILTER_VALIDATE_INT))
{
    exit("please enter integer");
}
$query="select * from st_table where student_id=".$sid;
$stmt=mysqli_query($dbhandle,$query)
    or die("unable to execute".$query);

$column=mysqli_fetch_fields($stmt);
echo "<table border=1>";
foreach($column as $col)
{
    echo "<th>".$col->name."</th>";
}
while($row=mysqli_fetch_array($stmt))
{
    echo "<tr>";
    echo "<td>".$row['student_id']."</td>";
    echo "<td>".$row['st_name']."</td>";
    echo "<td>".$row['age']."</td>";
}
}

```

```

echo "<td>".$row['dept_id']."</td>";
echo "</tr>";

}
echo "</table>";
mysqli_close($dbhandle);
exit();
}
?>
<html>
  <head>
    <title>
      student details
    </title>
  </head>
  <body>
    <form action="<?php echo $_SERVER['PHP_SELF']?>" method="get">
      please enter student id:<input type="text" name="sid" value=""/><br>
      <input type="submit" name="submit" value="Get Details"/>

    </form>
  </body>
</html>

```

Insertdb.php

```

<?php

require_once 'authorize.php';
?>

<?php
if(isset($_GET['submit']))
{
require_once 'config.php';
$dbhandle=mysqli_connect($host,$db_user,$db_pass,$db_name)
  or die("unable to connect");
$sid=mysqli_escape_string($dbhandle,$_GET['sid']);
$name=mysqli_escape_string($dbhandle,$_GET['sname']);
$age=mysqli_escape_string($dbhandle,$_GET['age']);
$deptid=mysqli_escape_string($dbhandle,$_GET['deptid']);
if(!filter_var($sid,FILTER_VALIDATE_INT)
  || !filter_var($age,FILTER_VALIDATE_INT)
  || !filter_var($deptid,FILTER_VALIDATE_INT))
{
  exit("please enter integer values");
}
$query="insert into st_table values($sid,$name,$age,$deptid)";

$st_table=mysqli_query($dbhandle,$query)

```

```

        or die("unable to execute".$query);
if(mysqli_affected_rows($dbhandle)==1)
{
    echo "student data inserted successfully";
}
else
{
    echo "unable to insert data";
}
mysqli_close($dbhandle);
}

?>
<html>
<head>
<title>
    student details
</title>
</head>
<body>
<form action="<?php echo $_SERVER['PHP_SELF']?>" method="get">
<table>
<tr>
<td>Please enter student id:</td>
<td> <input type="text" name="sid" value=""/></td>
</tr>
<tr>
<td>Please enter student name:</td>
<td> <input type="text" name="sname" value=""/></td>
</tr>
<tr>
<td>Please enter student age:</td>
<td> <input type="text" name="age" value=""/></td>
</tr>
<tr>
<td>Please enter student dept id:</td>
<td> <input type="text" name="deptid" value=""/></td>
</tr>
</table>
<input type="submit" name="submit" value="Insert data"/>

</form>
</body>
</html>

```

Updatedb.php

```
<?php
```

```

require_once 'authorize.php';
?>
<html>
  <body>
    <?php
require_once 'config.php';
if(isset($_GET['getdata']))
{
  $sid=$_GET['sid'];
  $st_data=get_student_data($sid);
  display_form($st_data);
}
elseif(isset($_POST['update']))
{
  if(process_update($_POST)==1)
  {
    echo "student data updated successfully<br>";
  }
  elseif(process_update($_POST)==0)
  {
    echo "nothing to update <br>";
  }
  else
  {
    echo "unable to update<br>" ;
  }
}
?>

<form action="<?php echo $_SERVER['PHP_SELF']?>" method="get">
  Enter student id:
  <input type="text" name="sid" value=""/>
  <input type="submit" name="getdata" value="submit"/>

</form>
<?php
function process_update($update_form)
{
  // require_once 'config1.php';

$dbhandle=mysqli_connect($GLOBALS['host'],$GLOBALS['db_user'],$GLOBALS['db_pass'],
  $GLOBALS['db_name']) or die("unable to connect");
  $sid=mysqli_escape_string($dbhandle,$update_form['sid']);
  $sname=mysqli_escape_string($dbhandle,$update_form['sname']);
  $age=mysqli_escape_string($dbhandle,$update_form['age']);
  //$emailid=mysqli_escape_string($dbhandle,$update_form['emailid']);
  $deptid=mysqli_escape_string($dbhandle,$update_form['deptid']);
  if(!filter_var($sid,FILTER_VALIDATE_INT)
    || !filter_var($age,FILTER_VALIDATE_INT)
    || !filter_var($deptid,FILTER_VALIDATE_INT))
  {
    exit("please enter integer values");
  }
}

```

```

}
$query="update st_table set st_name='sname','."age=$age,".
    "dept_id=$deptid where student_id=$sid";
$stmt_table=mysqli_query($dbh,$query)
    or die("unable to execute".$query);
$affected_rows=mysqli_affected_rows($dbh);
mysqli_close($dbh);
return $affected_rows;
}

function get_student_data($sid)
{
    //require_once 'config1.php';

    $dbh=mysqli_connect($GLOBALS['host'],$GLOBALS['db_user'],$GLOBALS['db_pass'],
        $GLOBALS['db_name']) or die("unable to connect");
    $sid=mysqli_escape_string($dbh,$sid);
    if(!filter_var($sid,FILTER_VALIDATE_INT))
    {
        exit("please enter integer values");
    }
    $query="select * from st_table where student_id=".$sid;
    $student_table=mysqli_query($dbh,$query)
        or die("unable to execute");
    $student_data=mysqli_fetch_array($student_table);
    mysqli_close($dbh);
    return($student_data);
}

function display_form($student_data)
{
    ?>
    <form action="<?php echo $_SERVER['PHP_SELF']?>" method="post">
    <input type="hidden" name="sid" value="<?php echo $student_data[0]?>" />
    <input type="text" name="sname" value="<?php echo $student_data[1]?>" />
    <input type="text" name="age" value="<?php echo $student_data[2]?>" />
    <input type="text" name="deptid" value="<?php echo $student_data[3]?>" />
    <input type="submit" name="update" value="update data" />
    </form>
    <?php
}
?>
</body>
</html>

```

```

<?php
require_once 'authorize.php';
if(isset($_POST['submit']))
{
require_once 'config.php';
$dbhandle=mysqli_connect($host,$db_user,$db_pass,$db_name)
    or die("unable to connect");
$sid=mysqli_escape_string($dbhandle,$_POST['sid']);
if(!filter_var($sid,FILTER_VALIDATE_INT))
{
    exit("please enter integer");
}
$query="delete from st_table where student_id=".$sid;
$student_table=mysqli_query($dbhandle,$query)
    or die("unable to execute".$query);
if(mysqli_affected_rows($dbhandle)==1)
{
    echo "student data deleted successfully";
}
else
{
    echo "unable to delete data";
}
mysqli_close($dbhandle);
}

?>
<html>
<head>
<title>
    student details
</title>
</head>
<body>
<form action="<?php echo $_SERVER['PHP_SELF'];?>" method="post">
    please enter student id:<input type="text" name="sid" value=""/><br>
    <input type="submit" name="submit" value="Delete Data"/>

</form>
</body>
</html>

```

Logout.php

```

<?php
session_start();
unset($_SESSION['name']);
unset($_SESSION['isAuth']);
session_destroy();
header('Location:login.php');
?>

```

Mozilla Firefox
http://localhost...eg2/home.php? +
localhost/~davinci/databaseeg2/home.php? Google

Welcome testuser

[Get all student details](#)
[Get student details](#)
[Insert student details](#)
[update student details](#)
[Delete student details](#)
[Log out](#)

Mozilla Firefox
http://localhost...2/selectdb.php +
localhost/~davinci/databaseeg2/selectdb.php Google

Welcome testuser

connected successfully

student_id	st_name	age	dept_id
110	sname	25	5
101	krish	25	1
103	ganesh	25	2
106	radha	20	3
107	kalaivani	21	4
108	sumathi	20	4
111	sujatha	25	5

Mozilla Firefox
http://localhost...2/selectdb.php
localhost/~davinci/databaseeg2/selectdb.php

Welcome testuser

connected sucesfully

student_id	st_name	age	dept_id
110	sname	25	5
101	krish	25	1
103	ganesh	25	2
105	lalitha	19	3
106	radha	20	3
107	kalaivani	21	4
108	sumathi	20	4

student details - Mozilla Firefox
student details
localhost/~davinci/databaseeg2/selectstudent.php

Welcome testuser

please enter student id:

Mozilla Firefox
http://localhost...it=Get+Details
localhost/~davinci/databaseeg2/selectstudent.php?sid=105&submit=Get+Details

Welcome testuser

student_id	st_name	age	dept_id
105	lalitha	19	3

Mozilla Firefox
http://localho...2/updatedb.php
localhost/~davinci/databaseeg2/updatedb.php

Welcome testuser

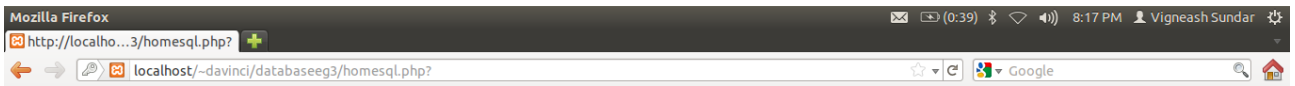
Enter student id:

Welcome testuser

Enter student id:

Welcome testuser

please enter student id:

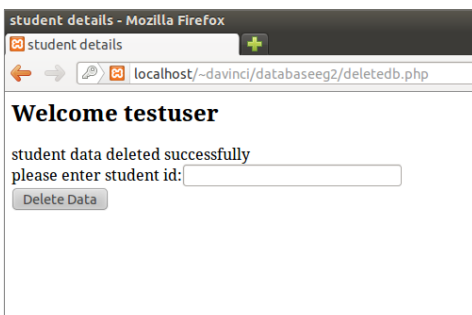


welcome testuser

- [get all user details](#)
- [insert new user details](#)
- [delete user details](#)
- [logout](#)



Loginsql.php



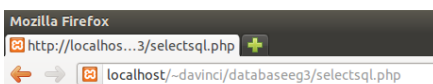
```
<?php
if(isset($_POST['submit']))
{
    require_once 'configsql.php';
    $dbhandle=mysqli_connect($host,$db_user,$db_pass,$db_name) or die("unable to connect");
    $username=mysqli_escape_string($dbhandle,$_POST['username']);
```

```

$password=mysqli_escape_string($dbhandle,$_POST['password']);
if(empty($username) || empty($password))
{
    exit("please enter username and password");
}
$query="select * from login where username='$username' and password='$password'";
$login_result=mysqli_query($dbhandle,$query) or die("unable to execute".$query);
if(mysqli_num_rows($login_result)==1)
{
    session_start();
    $_SESSION['name']=$username;
    $_SESSION['isAuth']=true;
    header('location:homesql.php?',SID);
    exit();
}
else
{
    echo "wrong username/password combination";
}
}
?>

<html>
<body>
<form action="<?php echo $_SERVER['PHP_SELF']?>" method="post">
username:
    <input type="text" name="username" value=""/>
    <br>
password:
    <input type="password" name="password" value=""/>
    <br>
    <input type="submit" name="submit" value="submit"/>
</form>
</body>
</html>

```



welcome testuser

connected successfully

student_id	st_name	age	dept_id
110	sname	25	5
101	krish	25	1
103	ganes	25	2
106	radha	20	3
107	kalaivani	21	4
108	sumathi	20	4
111	sujatha	25	5