



Path: <http://localhost/phpmyadmin>



- * PHP is providing `php_mysql.dll` library with no. of functionalities to connect with MySQL database.
- * MySQL is an open source relational Database management syst. supporting number of objects like Tables, views, etc.
- * MySQL is implemented by MySQL AB corporation in 1995. Now, it is maintaining by Oracle Corporation.
- * The default username of mysql db is "root" and it does not contain any "password".

~~MySQL Differenciation~~

02. OCT. 2019

* Database :

A Database is a collectⁿ of database objects. Database we are using to maintain Tables, stored procedures, etc.

* Table:

Table is a collectⁿ of Rows and columns.

* MySQL Interaction With Command Prompt:

We can connect with MySQL database

through a command prompt by executing `mysql.exe` file what is available in XAMPP/MySQL/Bin Folder.
`C:/xampp/mysql/bin>mysql.exe -u root`
`C:/xampp/mysql/bin>mysql.exe -u root & create database dbnew;`
`mysql> Create database dbnew;` `mysql> Use dbnew;`
`mysql> Use dbnew;`

Columns: Fields.

Rows: Records

* phpMyAdmin:

It is a Graphical User Interface used to connect with MySQL database. It is available with XAMPP download. The URL address to open `phpMyAdmin` is `http://localhost/phpMyAdmin`.

1) Insert:

By using this option we can insert records in Table.

2) Browse:

By using this option, we can Browse the Table Records.

3) Structure:

To change the structure of a Table. If u want to add new ~~fields~~ fields, delete existing fields, Add constraints, delete constraints and to rename columns.

4) SQL:

By using this option, we can execute our SQL statements.

5) Search:

By using this option, we can search the Records in Ascending or Descending order based on ~~any~~ columns (columnnames).

6) Operations:

By using this option, we can change the Tablename, to move Table into another database, to change the table engine etc.

7) Export:

By using this option we can Export database Tables as sql, pdf, Excel, etc. file formats.

8) Import :

By using this option, we can Import the exported file.

9) Empty:

By using this option, we can delete the table Records.

10) Drop:

By using this option, we can delete the table structure.

* MySQL Interaction With PHP:

PHP 5.0 is providing php-mysql.dll library to connect with mysql database. This library is providing more functions.

1) mysql_connect:

By using this fⁿ, we can create a connⁿ betⁿ php prog. and mysql db. It contains 3 Arguments. i) ServerName, ii) Username, And iii) Password.

2) mysql_select_db :

To select a db from mysql server, arguments are database name And connection id.

3) mysql_query:

To execute a sql query in my-sql database. Arguments are sql statement And connection id.

4) mysql_error:

To Get the Error Message if Any occurred at the time of mysql stat. Execution.

5) mysql_errno:

To Get the Error number if any occurred at the time of mysql stat. Executⁿ.

* Prog: To create a connⁿ betⁿ MySQL and PHP.

Ex: <?php \$con = mysql_connect("localhost", "root", ""); // Userid, Blank password, 03-Oct-2019

if(\$con=mysql_connect("localhost", "root", "")) SAVE AS: Connect.php

{ echo("Connected"); "
"; }

echo \$con;

} else { Resource datatype . }

echo mysql_error();

}

O/p: Connected
Resource id #2

Ex: <?php Prog. To add New Record

mysql_connect ("localhost", "root", "");

mysql_select_db ("db_bar6pm");

if(mysql_

Ex: <?php
 \$con=mysql_connect("localhost","root","");
 if(mysql_query("create database db-new",\$con))
 echo "Database is created";
 else
 echo mysql_error();
 ?>

SAVE AS: DatabaseCreated.php (77)

O/P : Database is created
 Connection id
ON Double Click (Refresh)
O/P :

Ex: <?php *** Prog. to create a Table in a Database**

```
$con=mysql_connect("localhost","root","");
mysql_select_db("db-new",$con);
if(mysql_query("Create Table tb-new(sno int)",$con))
echo "Created";
else
    echo mysql_error();
?>
```

SAVE AS: TableCreated.php

O/P : Created

ON Double click (Refresh)

Table 'tb-new' already exists.

* Datatypes Available in MySQL Server:

Basically, we have 3 types of Datatypes

- 1) Text Types
- 2) Number Types
- 3) Date Types

I) Text Types: occupies ~~100~~ Bytes

i) CHAR: ~~data~~ By using this type, we can store ~~a~~ fixed set of characters from 0 to 255 characters long.

ii) VARCHAR: ~~data~~ By using this type, we can store a variable set from 0 to 255 characters long

iii) TINYTEXT: A string with a max^m length of 255 characters.

iv) TEXT: A string with a max^m length of 65535 characters.

v) BLOB: A string with a max^m length of 65535 characters. (same as TEXT)
 But can also store, Binary Data.

vi) MEDIUMTEXT: A string with a max^m length of 16777215 characters.

vii) MEDIUMBLOB: It is same as ~~tinytext~~ MEDIUMTEXT. But, it can also store Binary Data.

viii) LONGTEXT: String with a max^m length of 4294967295 characters.

ix) LONGBLOB: Same as ~~tinyblob~~ LONGBLOB. We can also store Binary Data.

II) Number Types:

i) TINYINT:

Store the signed values from -128 to 127, and 0 to 255 Unsigned values.

ii) SMALLINT:

Signed: -32768 to 32767
Unsigned: 0 to 65535.

iii) MEDIUMINT:

Signed: -8388608 To 8388607
Unsigned: 0 To 16777215.

iv) INT:

Signed: -2147483648 To 2147483647

Unsigned: 0 To 4294967295.

v) BIGINT:

Signed: -9223372036854775808

To 9223372036854775807

(9.2e+17)

vi) Unsigned:

0 To 18446744073709551615

~~Date: 04 Oct 20~~

7) LTRIM:

To delete LHS spaces of a string.

8) RTRIM:

To delete RHS spaces of a string.

Ex: Select concat(trim("Scott"), "smith")

O/P: Scottsmith

9) SQRT:

To get the square Root of \$ip value

Ex: Select sqrt(16) O/P: 4

~~10) ASCII:~~ By using this f?, we can get the ASCII value of \$ip value.

vi) FLCAT:

A small no. with a floating Decimal pt.

Ex: Select ascii('a')

* Count: To get Total no. of Records

vii) DOUBLE:

A Large no. with a Floating Decimal pt.

Ex: Select count(empid) from emp

To get the character of an ASCII value.

viii) DECIMAL:

A Double stored as a String, allowing for a fixed Decimal point.

Ex: select char(98) O/P: b

III) Date

1) DATE YYYY-MM-DD

~~9881168666 Umang Nagpur mod:~~ To Get the Remainder of \$ip value

Ex: Select mod(10,5) O/P: 0

2) DATETIME YYYY-MM-DD HH:MM:SS

15) ADDDATE:

By using this, we can add some dates to \$ip date.

Ex: Select ADDDATE('2012-01-02', INTERVAL 31 DAY);

16) curdate, current_date:

By using this f?

We can get the current date inform? curtime,

17) current_time(): To get the current time inform?. O/P: 2012-10-10 19:24:38

18) Current_timestamp(): It is combination of current_date & current_time used to display Both Date & Time Inform?.

Ex: i) Select curdate();
ii) Select curtime();
iii) Select current_timestamp(); O/P: 2012-10-10 19:24:38

1) Abs:

By using this function we can get, the Absolute value of \$ip value.

(Converts -value to +value & +value to -value)

Ex: Select abs(-10) O/P: 10.

2) SIN:

To get the sin value of \$ip value

Ex: Select sin(0) O/P: 0

Ex: Select sin(90) O/P: 0.89399666360056

27) MD5 (Message Digest 5)

03.OCT.2019

By using this fⁿ, we can convert the \$fp value as 32 characters length alphanumeric string. ex: Select MD5("Scott") 14 AUGUST 2019

5) Round:

Rounds

Converts a floating point number to its nearest integer value.

Ex: Select round (10.40) O/P : 10

*now():

Same as current_timestamp

Select round (10.90) O/P : 11

Select round (10.50) O/P : 11

6) Floor:

By using this fⁿ, we can rounds a floating point number to its nearest lowest integer value.

Ex: Select floor (10.99) O/P : 10

7) ceil, ceiling:

Rounds a floating point number to its nearest highest integer value.

04.OCT.2019

Ex: Select ceil (10.40)

O/P : 11

*sum:

To get all Records of Specified column.

18) rand:

Returns a random number betⁿ 0 and 1. Ex: Select sum(emppid) from emp

04.OCT.2019

Ex: Select rand

O/P : 0.58637....

20) min:

To get minimum value from group of values.

04.OCT.2019

Ex: Select min(sno) from tab_user

Ex: Select pow(2,2)

21) max:

Select min(emppid) from emp

22) Concat:

To get maxⁿ value from group of values. Ex: Select max(emppid) from emp

22) Concat:

Join two or more than two string values.

Ex: Select concat(cname, username) from jos_user O/P :

Ex: Select concat(uname, pwd) from new-tab O/P:

Ex: Select concat("Scott", "Smith", "John"). O/P: ScottSmithJohn

23) CHAR_LENGTH, CHARACTER_LENGTH:

By using this fⁿ, we can get the total number of characters of a string.

04.OCT.2019

i) Select char_length(uname) from new-tab O/P :

ii) SELECT CHARACTER_LENGTH("NILESH") O/P:

24) Reverse:

Display the string in Reverse direction.

04.OCT.2019

Ex: Select reverse ("Scott") O/P: tcoos

25) Substring:

We get the substring of a string contains 2 arguments.

04.OCT.2019

First character number & Second is Total number of character.

Ex: Select substring("scott", 2, 3) O/P: cot

04.OCT.2019

Select substring("Welcome", 4, 4) O/P: Come

26) CRC82 (Cyclic Redundancy Check): Bit Encrypted Numeric

By using this fⁿ, we can convert the \$fp String as 32 characters length value.

Ex: Select crc82("scott") O/P:

Program: 1

```
<? PHP
if ($con = mysql_connect("localhost", "root", ""))
    echo "Connected";
else
    echo mysql_error();
echo $con;
?>
```

Program: 2

```
<? PHP
$con = mysql_connect ("localhost", "root", "");
if (mysql_select_db ("test", $con))
    echo "Selected";
else
    echo mysql_error();
?>
```

```
<?php
if(isset($_POST['submit']))
{
    $db_name=$_POST['dbname'];
    $db_name=$_POST['dbname'];
    mysql_connect("localhost", "root", "");
    if(mysql_select_db($db_name))
        {
            $query="insert into db_name values('".$_POST['dbname']."')";
            if(mysql_query($query))
                echo "Created";
            else
                echo mysql_error();
        }
    }
?>
```

At: DBName:
Create

Program: 3

```
<? PHP
$con = mysql_connect ("localhost", "root", "");
if (mysql_query ("create database db1", $con))
    echo "DB created";
else
    echo mysql_error();
?>
```

Program: 4

```
<? PHP
$mysql = mysql_connect ("localhost", "root", "");
if(mysql_query ("create database db2", $con))
    mysql_select_db ("test");
mysql_query ("create table tb1_newtab (uname varchar(100), pwd varchar(100))");
$sqlstt = "insert tb1_newtab values ('scottaa', 'scott123')";
mysql_query ($sqlstt);
?>
```

```
<?php
$con = mysql_connect ("localhost", "root", "");
mysql_select_db ("test");
mysql_query ("create table tb1_batch (uname varchar(100), pswd varchar(100), sno int)", $con);
$sqlstt = "insert tb1_batch values ('N1', 'L01')";
mysql_query ($sqlstt);
echo "Row is added";
?>
```

mysql_fetch_row:

By using this F() we can fetch a record from resultset and returns the Obj as Numeric Array. At the same time it locates the resultant pointer at the begining of Next Record.

```
<table border='0' cellpadding='0' cellspacing='0'>
<tr>
<th style='background-color: lightyellow; color: red'> SNO </th>
<th style='background-color: lightyellow; color: red'> Password </th>
<?php
$no=1;
mysql_connect("localhost", "root", "");
mysql_select_db("test");
$data=mysql_query("select * from new_tab");
while($rec=mysql_fetch_row($data))
{
if($no%2==0)
$color="lightblue";
else
$color="lightgreen";
$no++;
echo 'K' . $no . '
<td style="background-color: ' . $color . '">' . $rec[0] . '</td> ' . $rec[1] . '<td>' . $rec[2] . '</td>';
}
?>
```

Run -

<http://localhost/xyz.php>

rahu1pandey, stn@gmail.com

id: satyamphp@gmail.com

Pwd: phoassword

id: stbatch11@gmail.com

Pwd: stbatch

Father: Ramdars Singh

Entry Singh

8109332193 Mob NO:

C/o Sonu Devi

Nayee Dadhi

Karvate, Chandwali, Up.

232104

DOB: 06. April. 1989

pravesh26@gmail.com

Database Engines in MySQL: Database Engine supported by MySQL internal process (82)
A DB engine is an underlying program using to run the SQL statements in DB languages. Diff. types of DB engines are available in MySQL

* Different Types of DB Engines

i) myisam:

It is a default storage engine supports MySQL. Every table by default creates with this engine.

If we create any table with myisam engine, it creates 3 files to maintain the table information? The filenames are same as table name contains diff. types of extensions.

ii) .frm:

This extension file maintains table format.

iii) .myd:

It maintains Table Data.

iv) .myi:

It maintains Table Index.

All these files are available in Data folder of MySQL folder.

* SQL stat. to create table with myisam Engine

→ Create Table HbL_new1(sno int) engine=MyISAM

Create Table <tablename> engine=MyISAM

MyISAM stores the information in Low Level Binary Format. That's why we can use this tables with any operating system.

To see the DB engines available with MySQL we need to execute Show Engines statement: show engines

Information in Low Level Binary Format that's why it is comparable with any type of OS.

MyISAM can store max³² limit upto 4GB data. It supports max³² dataspace and also it executes select statement very fast.

* Drawbacks of MyISAM:

To show table status SQL stat. is: show table status

Why

i) This engine does not support Relationships b/w the tables that's we don't use this engine with complex Enterprise level Applications.

ii) We can not get the data back easily if database server is crashed.

iii) It is faster at the time of retrieving data. But it is slow while inserting & updating data.

* INNODB:

i) It is Transaction Safe Engine supports the relationships b/w the tables.

These tables we can used to develop complex Enterprise level Application.

iii) This engine supports the data recovery option, If the database server is crashed.

iv) It is very faster at the time of inserting & updating the records.

v) We can easily retrieve the data back, if db server is crashed.

* SQL stat. to create table with innodb engine:

Create Table <tablename> (col1 datatype) engine=innodb

Create Table HbL_new1(sno int) engine=innodb

3) MEMORY (heap):

This Engine can stores the data in Memory locatⁿ of database server. It is 30% more faster than MyISAM Engine. But the Data is delete when we stop the Database Server.

4) BERKELEY: ~~Create Table~~

By using this Engine we can maintain the Relationships Betⁿ the Tables. It is same as innodb But this Engine Occupies more memory locatⁿ and we cannot move this Tables easily from one locatⁿ to Another locatⁿ. That's why we don't use these engines in applicat^s.

~~constraint supported by different engines~~

* Constraints Supported File MySQL DB:

05.OCT.2012

Constraints are nothing but the condⁿ what we can apply on Database Columns. MySQL is supporting different Types of Constraints

1) NOT NULL:

By using this constraint, we can Restrict the Null values in Database columns. This Constraint doesn't allow Null-values.

Ex: Create Table Tbl-new(sno int not null)

2) Unique:

By using this Constraint, we can Restrict the duplicate values in DB columns.

Ex: Create Table Tbl-new4(sno int unique) Ex: Create Table Tbl-new2(sno int unique, uname varchar(100))

3) primary key:

By using this constraint, We can Restrict a column with Null & Duplicate values. It is a combinatⁿ of not Null & Unique.

Ex: Create Table Tbl-new4(sno int primary key)

4) auto_increment:

By using this constraint we can increase the column value with a new Record without passing any explicit value. The auto increment column should be primary key column. Specifying variable

Ex: Create Table Tbl-new5(sno int primary key auto_increment, uname varchar(100))

5) default:

By using this constraint, we can set the default value to the table column.

Ex: Create Table Tbl-new7(sno int default 100, uname varchar(100) default 'SCOTT')

6) Foreign key:

By using this constraint, We can refer a table column with another table primary key column. (Refers value of another column from primary key column)

To create the Relationship Betⁿ primary key & foreign key column, The Table should be innodb tables.

Ex: Create Table `tbl-parent` (`pid int primary key, pname varchar(100)`) engine=innodb
 Create Table `tbl-child` (`cid int, cname varchar(100), pid int, foreign key (pid)`
 references `tbl-parent` (`pid`) engine=innodb

<u>tbl-parent</u>		<u>tbl-child</u>		
<u>pid</u>	<u>pname</u>	<u>cid</u>	<u>cname</u>	<u>pid</u>
1	-			
2	-			

{ Unique key allows only one Null value }

SQL statements:

- * SQL statements supported by MySQL Database.
- * SQL statements are divided into 4 different Types.
 - 1) DML (Data Manipulation Language)
 - 2) DCL (Data Control Language)
 - 3) TCL (Transact Control Language)
 - 4) DDL (Data Definition Language)

DML statements:

By using this statement, we can manage the data of Database Objects

a) SELECT:

By using this statement, we can select Table Records.

Ex: `select * from tbl-user` → '*' means all columns

i) `select uname, pwid from tbl-user`

ii) `select * from tbl-user where uname='scott'`

iii) `select * from tbl-user where uname like 's_'` → that 2 underscore means it shows

clip starting with s & 2 underscores after 's'.

iv) `select * from tbl-user limit 0,2`

v) `select * from tbl-user where uname like 's%'`

b) INSERT:

By using this statement, we can insert Records in Table.

Ex:

i) `insert tbl-user values (123, 'abc', 'abc')`

ii) `insert tbl-user (sno) values (1234)` iii) `insert into tbl-user (uname) values ('smith')`

iv) `insert into tbl-user values (123, 'abc', 'abc'), (456, 'xyz', 'xyz')`

v) `insert into tbl-user values (100, 'scott'), (200, 'smith')`

PHP

```
mysql_connect ("localhost", "root", "");
mysql_select_db ("test");
$sqlstmt = "insert into tbl-user values (333, 'x', 'y');"
```

```
mysql_query ($sqlstmt);
```

```
echo "Row Added";
```

```
}
```

MySQL_fetch_row():

This fn fetches a Record from Resultset returns as an Array.

Total number of Array Elements are equal to the number of table columns.

This fn returns the arr as Numeric Array.

key starts from "0" and ends with

Total no. of Elements values are the column values of the Records.

<?php

```
mysql_connect("localhost", "root", " ");
mysql_select_db("test");
$data=mysql_query("select * from table_user");
while($row=mysql_fetch_row($data))
```

=x=

<style>

```
th {
    background-color: lightblue;
    color: red;
}
```

</style>

<table border='2'>

<th> Sno </th>

<th> Uname </th>

<th> Pwd </th>

<?php

```
mysql_connect("localhost", "root", " ");
mysql_select_db("test");
$data=mysql_query("select * from tbl_user");
$count=1;
```

while(\$row=mysql_fetch_row(\$data))

```
{
    if ($count%2==0)
        $color="lightyellow";
    else
```

\$color="lightgreen";

\$count++;

// print_r(\$row);

```
echo "<tr style='background-color: $color'><td>$row[0]</td><td>$row[1]</td><td>$row[2]</td>"
```

}

{}

```

<?php
if(isset($_POST['sub']))
{
    $sno=$_POST['sno']; $uname=$_POST['un']; $pwd=$_POST['pwd'];
    mysql_connect("localhost", "root", "");
    mysql_select_db("test");
    if(mysql_query("insert into new
                    values('$sno','$uname','$pwd')"))
        echo "Row is Inserted";
    else
        echo mysql_error();
}

<form method="post" action=" ">
sno:<input name="<del>sno"><br> Username:<input name="un"> <br> <input name="pwd"

<input type="submit" name="sub" value="Insert">
</form>

```

~~Ex:~~

* mysql_num_fields (Resultset) ~~getFields~~

→ By using this fⁿ we can get the total no. of Resultset fields. ~~getFields~~

* mysql_field_name (resultset, fieldIndexNumber):
By using this fⁿ, we can get the name of a field from a Resultset

The Arguments are resultset, fieldIndexNumber

Ex: <table border="2">

```

<?php
if(isset($_POST['sub']))
{
    $tname=$_POST['t1'];
    mysql_connect("localhost", "root", "");
    mysql_select_db('test');
    $data=mysql_query("select * from $tname");
    $fc=mysql_num_fields($data);
    for($i=0; $i<$fc; $i++)
    {
        $fname=mysql_field_name($data, $i);
        echo "<th>$fname</th>";
    }
    while($rec=mysql_fetch_row($data))
    {
        echo "<tr>";
        for($i=0; $i<$fc; $i++)
        {
            echo "<td>$rec[$i]</td>";
        }
    }
}

```

</table>

```

<form method="post" action=" ">
<Table name:<input name="t1"><br>
<input type="submit" name="sub"
       value="Get Records">
</form>

```

Ex: <?php

```

mysql_connect("localhost", "root", "");
mysql_select_db("test");
$data=mysql_query("select * from
                  test");
echo mysql_field_name($data, 2);
}

```

* mysql_num_rows()

By using this fⁿ, we can get the Total no. of Records from mysql resultset. Contains 1 arguments name of Resultset and

```
<?php
mysql_connect("localhost", "root", " ");
mysql_select_db("test");
$data = mysql_query("select * from tbl-user");
echo mysql_num_rows($data);
?>
```

* mysql_fetch_assoc():

By using this fⁿ, We can fetch the records from Resultset & Returns the O/p as an Associative Array. Array Keys are columnnames, Array Values are column values.

Ex:

```
<?php
mysql_connect("localhost", "root", " ");
mysql_select_db("test");
$data = mysql_query("select * from tbl-user");
$rec = mysql_fetch_assoc($data);
//print_r($rec);
echo $rec['uname'];
?>
```

* mysql_fetch_array():

To read the records from resultset & returns the O/p as both numeric Array & associative array. It is a combination of mysql_fetch_row(), mysql_fetch_assoc(), ~~mysql_fetch_array()~~ Ex: \$data = mysql_query("select * from tbl-user");
\$rec = mysql_fetch_array(\$data);

* mysql_fetch_object():

^{std} Reads a Record from Resultset & Returns the O/p as an object This object belongs to a Base class i.e. ~~stdClass~~. stdClass.

Ex:

```
<?php
mysql_connect("localhost", "root", " ");
mysql_select_db("test");
$data = mysql_query("select * from tbl-user");
$rec = mysql_fetch_object($data);
print_r($rec);
echo $rec->uname;
?>
```

* mysql_fetch_field():

(88)

By using this fn, we can fetch a field from Resultset
and returns the O/P ~~as~~ that can be an array. (To fetch complete informⁿ of a field as
Ex: <?php Array.)

```
mysql_connect("localhost", "root", "");  
mysql_select_db("test");  
$field=mysql_fetch_field($data);  
print_r($field);  
$field=mysql_fetch_field($data);  
print_r($field);
```

Note:
Ex: <?php Other way of that
session_start(); prog. see
?>

<table border='1'> On last page on
<?php Dated 30-Oct-2013
mysql_connect("localhost", "root", "");
mysql_select_db("information_schema");
if(isset(\$_REQUEST['qs']))
{\$sindex=\$_REQUEST['qs'];}
else
{
\$sindex=0;
\$data=mysql_query("select * from character_sets");
\$_SESSION['tr']=mysql_num_rows(\$data);
}
\$tabrec=\$_SESSION['tr'];
\$totrec=5;
\$data=mysql_query("select * from character_sets limit \$sindex, \$totrec");
while(\$rec=mysql_fetch_row(\$data))
{
echo "<tr><td>\$rec[0]</td> \$rec[1]</td> \$rec[2]</td></tr>";
}
echo "</table>";
\$previousindex=\$sindex-\$totrec;
if(\$sindex!=0)
{
echo " previous ";
}
\$nextIndex=\$sindex+\$totrec;
if(\$nextIndex<\$tabrec)
echo " next ";
}

```
<?php  
mysql_connect("localhost", "root", "");  
mysql_select_db("test");  
$data=mysql_query("select * from  
tbl_user");  
print_r(mysql_fetch_field($data));  
print_r(mysql_fetch_field($data));  
?>
```

delete:

By using this stat, we can delete a record or set of Records from a Table.

Ex: delete from tbl-user where sno=100;

update:

To update the Records of a Table.

Ex: i) update tbl-user set uname='scott123' where sno=10

ii) update tbl-user set uname='scott123', pwd='abcd' where sno=10;

DDL Statements:

By using these stat. we can define the structure of database objects, To create, alter, drop etc. we can go for ~~these~~ DDL statements.

09.OCT.2013

1> create:

By using this statement, we can create DB objects.

Ex: i) create user scott@localhost

ii) create user smith@localhost identified by 'smith123'

Ex: 5) alter table tbl-new drop primary key
Ex: 4) alter table tbl-new add constraint primary key(sno)

2> alter:

By using this statement, we can change the structure of a Table.

Ex: alter table tbl-new add column uname varchar(100)

alter table tbl-new add column address varchar(100) first

3> drop:

By using this statement, we can drop db objects like table, user, etc.

We can also drop the table columns

Ex: drop table tbl-new

4> truncate:

By using this statement, We can delete the all Records of Table.

Ex: truncate table tbl-new

5> rename:

By using this option, We can change the Table name.

Ex: rename table tbl-new to abcd

6> mysql_list_dbs():

This fn Returns list of databases as a Resultset.

Ex:

<select> select Tag is used for drop-down list.
<?php

mysql_connect("localhost", "root", "");

\$data=mysql_list_dbs();

while(\$db=mysql_fetch_row(\$data))

{

} echo "<option>\$db[0]</option>";

}

</select>

2) mysql_list_tables:

To get the list of Tables available in mysql database.

Ex: <select>

```
<?php
mysql_connect("localhost", "root", "");
$data = mysql_list_tables("db_new");
while ($table = mysql_fetch_row($data))
{
    echo "<option>$table[0]</option>";
}
</select>
```

Ex: 2) <form method="post" action=" " >

Databases : <select name="drpdb">

```
<?php
$ext = $_POST['drpdb'];
mysql_connect("localhost", "root", "");
$data = mysql_list_dbs();
while ($db = mysql_fetch_row($data))
{
    if ($ext == $db[0])
        echo "<option selected>$db[0]</option>";
    else
        echo "<option>$db[0]</option>";
}
</select>
<input type="submit" name="sub" value="GetTables" />
<?php
if (isset($_POST['sub'])) || isset($_POST['sub2']))
{
    $pname = $_POST['drptable'];
    mysql_connect("localhost", "root", "");
    $dbname = $_POST['drpdb'];
    $tables = mysql_list_tables($dbname);
    echo "<br>Tables : <select name='drptable'>";
    while ($rec = mysql_fetch_row($tables))
    {
        if ($pname == $rec[0])
            echo "<option selected>$rec[0]</option>";
    }
}
```

```

    else
        echo "<option> $rec[0] </option>";
    }
}

<input type="submit" value="Get Records" name="sub2" />

<?php
if (isset($_POST['sub2']))
{
    mysql_connect ("localhost", "root", " ");
    $dbname = $_POST['drpdb'];
    $tname = $_POST['drptable'];
    mysql_select_db ($dbname);
    $dataq = mysql_query ("select * from $tname");
    echo "<table border='2'>";
    $fc = mysql_num_fields ($dataq);
    for ($i=0; $i<$fc; $i++)
    {
        $fname = mysql_field_name ($dataq, $i);
        echo "<th>$fname";
    }
    while ($rec=mysql_fetch_row ($dataq))
    {
        echo "<tr>";
        for ($i=0; $i<$fc; $i++)
        {
            echo "<td>$rec[$i]</td>";
        }
    }
}
</form>

```

Ex:3 <?php

```

if (isset($_POST['sub1']))
{
    $tname = $_POST['txtable'];
    mysql_connect ("localhost", "root", " ");
    mysql_select_db ("test");
    $data = mysql_query ("select * from $tname");
    echo "<table border='2'>";
    $fc = mysql_num_fields ($data);
    for ($i=0; $i<$fc; $i++)
    {
        $fn = mysql_field_name ($data, $i);
        $fn = ucfirst($fn);
        echo "<th>$fn</th>";
    }
    while ($rec = mysql_fetch_row ($data))
    {
        echo "<tr>";
        for ($i=0; $i<$fc; $i++)
        {
            echo "<td>$rec[$i]</td>";
        }
    }
}
<? Form method="POST" action=" " >
<Table name:<input type="text" name="txtable">
<br/><input type="submit" value="Get Records" name="sub">
</form>

```

^ 1.0.1. (* ~~ENTER~~ ~~INPUT~~ ~~RECORDS~~

<script> src="jquery-1.3.js" </script>

SAVE AS: MySqlDelMulRec.php 11.OCT.2012

<style>

```
.cls1 { background-color: lightyellow; }
```

</style>

<script>

```
function applystyle(t) { t.style.backgroundColor = "lightyellow"; }
```

```
function removestyles(t, col) { t.style.backgroundColor = col; }
```

```
function chkfun(x) { // frm.elements show all elements in the form const frm = document.elements; for (i=0, i<const.length; i++) { if (const[i].type == "checkbox") const[i].checked = x; }}
```

```
function fun2(sno) { "MySql" location = Del.php?ss=" + sno }
```

</script>

<form action="DelAll.php" method="post" id="frm1">

<table border='2'>

<?php

```
mysql_connect("localhost", "root", ""); mysql_select_db("test"); $data = mysql_query("select * from tb1_cities"); $count = 0; while ($rec = mysql_fetch_row($data)) { if (($count % 2) == 0) $color = "lightblue"; else $color = "silver"; $count++; }
```

it is used for
one or more
the same page.
it is within the same
page.

```
echo "<tr style='background-color: $color; onmouseover='applystyle(this)' onmouseout='removestyles(this, '$color')'><td><input type='checkbox' name='$rec[0]'><td>$rec[1]<td>$rec[2]<td><img src='close.png' onclick='fun2($rec[0])'>" } }>
```

</table>

<input type="submit" name="sub" value="Delete All" />

<form>

```
<a href="#" onclick="chkfun(true)"> Check All <a href="#" onclick="chkfun(false)"> Uncheck All </a>
```

for go to previous page

de1all.php

<input type="button" SAVE AS: MySqlDelAll.php

<?php

```
mysql_connect("localhost", "root", ""); mysql_select_db("test"); foreach($_POST as $k => $v) { mysql_query("delete from tb1_cities where sno=$k"); }
```

header("location: ~~MySqlDelMulRec.php~~");

```

<?php      SAVE AS: MySqlDel.php    del.php
$x=$_REQUEST['ss'];
mysql_connect("localhost", "root", "");
mysql_select_db("test");
mysql_query("delete from tb_city where sno=$x");
header ("location: MySqlDel.php");
}

```

* program to delete multiple Records.

Ex: <script>

```

function Funchk (val)
{
arr = document.getElementById ('frm1').elements
for (i=0; i<arr.length; i++)
{
if (arr[i].type = "checkbox")
{
arr[i].checked=val
}
}

function Fundel (sn)
{
location ='del.php?' +sn
}

</script>
<Form method="post" action = "delall.php" id="frm1">
<Table border='1'>
<?php
mysql_connect("localhost", "root", "");
mysql_select_db("test");
$data=mysql_query ("select * from tb_city");
while ($rec =mysql_fetch_row ($data))
{
echo "<tr><td><input type='checkbox' name= $rec[0]>
<td> $rec[1] <td> $rec [2] <td>
<img src='b_drop.png' onclick='fundel($rec[0])'>";
}
</tr>
</table>
<input type="Submit" value="Delete All" >
</form>
<a href="#" onclick='funchk(true)'>Check All</a>
<a href="#" onclick = 'funchk(false)'>Uncheck All </a>

```

```

<?php
$qs=$_SERVER['QUERY_STRING'];
mysql_connect("localhost", "root", " ");
mysql_select_db("test");
mysql_query("delete from tbl-city where cid=$qs");
header("location:get.php");
?>

default
<?php
mysql_connect("localhost", "root", " ")
mysql_select_db("test");
foreach($_POST as $k=>$v)
{
    mysql_query("delete from tbl-city where id=$k");
}
header("location:get.php");
?>

```

* mysql_pconnect:

It is the same as mysql_connect, it creates ~~Persistent~~ conn^{Persistent}

first, It checks whether any conn^P is created or not. If any conn^P is already created, it uses the connect^P. Otherwise, it creates a new conn^P. The conn^P will close when we stop the server.

Ex: <?php
if(mysql_pconnect("localhost", "root", " "));
echo 'Connected';
else
echo "Not Connected";
?>

revoke insert on *.* from scott@localhost
revoke insert, select on *.* from scott@localhost
revoke all privileges on *.* from scott@localhost

16 Oct 2019 (95)

* <meta http-equiv="refresh" Content="5" /> <body bgcolor="lightblue">
By using this, ~~will~~ Refresh same page after every 5 seconds automatically means reloading after every 5 seconds.

* ~~Registration With Client-side Validation~~ Page AutoReload with Metatag.

* Registration with Client-side validation & Redirect to another page when

Javascript is Disabled:

Step I

→ <noscript> Reg.php

```
<meta http-equiv="refresh" Content="0; error.html">
</noscript>

<script src="jquery-1.3.js"></script>
<script src="jquery.validate.js"></script>
<style>
    .cls1 {
        color: red;
    }
</style>
<script>
    function fun1()
    {
        $("#frm1").validate({errorClass:'cls1'})
    }
</script>
<body onload="fun1()">
<form method="post" action="regval.php" id="frm1">
<table>
    <tr><td>Username <td><input name='txtuser' id="txtuser" class="required"
        onblur="chkuser(this.value)"><td> <span id="sp1"></span>
    <tr><td> Password <td><input type="password" name='txtpwd' class="required"
        minlength="10" id="pwd">
    <tr><td> Re-Password <td><input type="password" name='txtrpwd' equalTo="#pwd">
    <tr><td> Email <td> <input type="text" name='txtemail' class="required email">
    <tr><td> Age <td><input name='txtage' class="required number">
    <tr><td> <input type="submit" name="sub" value="Register">
</table>
</form>
```

```

<script>
function chkuser(val)
{
    $('#sp1').html('<img src=spinner-grey.gif width=20>')
    $.post("validate.php", {qs:val}, function(xy)
    {
        if(xy==0)
        {
            $('#sp1').css("color", "green");
            $('#sp1').html("Username is Available")
        }
        else
        {
            $('#sp1').css("color", "red");
            $('#xtuser').val("");
            $('#sp1').html("username is Not Available")
        }
    })
}
</script>

```

error.htm
 please enable javascript Go to Reg
<validate.php>

```

<?php
$qs = $_REQUEST['qs'];
mysql_connect('localhost', "root", "");
mysql_select_db("test");
$data=mysql_query("select * from tbl_user where uname='$qs' ");
sleep(3);
echo mysql_num_rows($data);
?>

```

A SERVER-SIDE REGISTRATION

18.OCT.2019 (97)

```

<?php
    include "conn.php";
    $uname = $_POST['txtnum'];
    $pwd = $_POST['txtpwd'];
    $pwd = "abc". $pwd. "$%&"; } important
    $pwd = crc32(md5($pwd)); } part
    —connect();
    if(mysql_query("insert into tbl-uservalues ('$uname', '$pwd')"))
    {
        echo "Account is Created";
    }
    else
    {
        if(mysql_errno() == 1062)
            echo "Username is already Registered";
        else
            echo mysql_error();
    }
}

```

19.OCT.2019

Reg.php

```

<?php
if(isset($_POST['sub']))
{
    $uname = $_POST['txtnum'];
    $pwd = $_POST['txtpwd'];
    $email = $_POST['txtemail'];
    $pwd = "abc". $pwd. "&#";
    $pwd = sha1(md5($pwd));
    mysql_connect("localhost", "root", "");
    mysql_select_db("test");
    if(mysql_query("insert into tbl_uservalues ('$uname', '$pwd', '$email', 0")))
    {
        $randno = rand(1, 1000);
        $randno = $uname . $randno;
        mysql_query("insert into tbl_activationvalues ('$uname', '$randno')");
        $str = <<<abc
Registration completed successfully to active ur account click on

```

```

<a href='activate.php?rn=$randomno'> ACTIVATION LINK </a>
abc;
mail($email,"Activation Link",$str,"From:admin@mysite.com\r\nContent-type:
echo "Registration Completed";
}
?>
<form method="post" action=" " >
Username: <input type="text" name='txtnuname'> <br>
password: <input type="text" name="txtpwd"> <br>
Email: <input type="text" name="txtemail"> <br>
<input type="Submit" name="sub" Value="Register">
</form>

```

Activate.php

```

<?php
$q = $_REQUEST['rn'];
mysql_connect("localhost","root","");
mysql_select_db("test");
$data = mysql_query("select uname from tbl_activation where acode ='$q' ");
$rec = mysql_fetch_row($data);
$uname = $rec[0];
mysql_query("Update tbl-user set active=1 where uname = '$uname' ");
echo "Activation completed successfully";
?>

```

Login.php

```

<script src="jquery-1.3.js"></script>
<script src="cookie.js"></script>
<script>
function readcookie(uname)
{
$("#txtn2").val($.cookie(uname))
}
function fun1()
{
if (document.getElementById('chk1').checked)
{
un = $("#txtn1").val()
pw = $("#txtn2").val()
}

```

```

    $cookie(un, pw, {expires:10})
}

</script>
<?php
if(isset($_POST['sub']))
{
    $uname = $_POST['txtruser'];
    $pwd = $_POST['txtpwd'];
    $pwd = "abc". $pwd. "$$#";
    $pwd = sha1(md5($pwd))

mysql_connect("localhost", "root", " ");
mysql_select_db("test");
$sqlstmt = "select * from tb1_user where uname = '$uname' and pwd = '$pwd'";
$data = mysql_query($sqlstmt);
if(mysql_num_rows($data) == 1)
{
    $data1 = mysql_query("select * from tb1_user where uname = '$uname' and active = 1");
    if(mysql_num_rows($data1) == 0)
        echo "Activate Account";
    else
        echo "Welcome To User";
}
else
    echo "Invalid User";
}

<form method="post" action=" " >
    Username: <input name="txtruser" id="txtr1" onblur="readcookie(this.value)"><br>
    Password: <input name="txtpwd" id="txtp1" type="password"> <br>
    <input type="checkbox" id="Chk1"> Remember Me <br>
    <input type="submit" name="sub" value="Login" onclick="fun1()"/></form>

```

```

<script>
function funupdate(uname)
{
location='update.php?' + uname
}
</script>
<table border='1'>
<?php
mysql_connect('localhost','root','');
mysql_select_db("test");
$data=mysql_query("select * from tbl_user");
while($rec=mysql_fetch_row($data))
{
echo "<tr><td> $rec[0]
<td> $rec[1]
<td> $rec[2]
<td> $rec[3]
<td> $rec[4]
<td><input type='button' value='update'
      onclick=funupdate('$rec[0]')>";
}
</table>

```

Update.php

```

<?php
session_start();
$qs=$_SERVER['QUERY_STRING'];
$_SESSION['oun']= $qs;
mysql_connect("localhost","root","");
mysql_select_db("test");
$data=mysql_query("select * from tbl_user
                  where uname='$qs' ");
$rec=mysql_fetch_row($data);
$str <<<abc

```

```

<form method="post" action="save.php">
abc;

```

```

for ($i=0; $i<count($rec); $i++)
{

```

```

$fname=mysql_field_name($data,$i);
$fname=ucfirst($fname);
$str = "$fname:<input type='text' name=$fname VALUE=$rec[$i]<br>";

```

10. OCT. 2013 100

```

$str = "<input type='Submit' value='Save'>";
echo $str;
}

```

Save.php

```

<?php
session_start();
$str="update tbl_user set";
$count=count($_POST);
$i=0;
foreach($_POST as $k=>$v)
{
    $i++;
    $str.= $k."=". "'$v'";
    if($i!=$count)
        $str.=",";
}

```

```

$uname=$_SESSION['oun'];
$str="where uname='$uname'";
mysql_connect("localhost","root","");
mysql_select_db("test");
mysql_query($str);
header("location: get.php");
}

```

Up

Active: <input type="text" name="txtactive" value=\$rec[4]>

<input type="submit" name="sub" value="Save">
</form>
abc;
echo \$str;
?>

Save.php

```
<?php  
session_start();  
$uname=$_POST['txtuser'];  
$pwd=$_POST['txtpwd'];  
$email=$_POST['txtmail'];  
$age=$_POST['txtage'];  
$active=$_POST['txtactive'];  
$ou=$_SESSION['oun'];  
$sqlstmt="update tb1_user set uname='$uname',  
pwd='$pwd', email='$email', age='$age',  
active=$active where uname='$ou'";  
mysql_connect("localhost", "root", "");  
mysql_select_db("test");  
mysql_query($sqlstmt);  
header("location: get.php");  
?>
```

Update.php

```
<?php  
session_start();  
$qs=$_SERVER['QUERY_STRING'];  
$_SESSION['oun']=$qs;  
mysql_connect("localhost", "root", "");  
mysql_select_db("test");  
$data=mysql_query("select * from tb1_user  
where uname='$qs'");  
$rec=mysql_fetch_row($data),  
$str<<< abc  
<form method="post" action="Save.php">  
<Username:> <input type="text" name="txtuser"  
value=$rec[0]> <br>  
password: <input type="text" name="txtpwd"  
value=$rec[1]> <br>  
Email : <input type="text" name="txtmail"  
value=$rec[2]> <br>  
Age: <input type="text" name="txtage"  
value=$rec[3]> <br>
```

* mysql_close()

By using this fⁿ, we can close the opened database connⁿ.

```
<?php  
mysql_connect("localhost", "root", "");  
mysql_close();  
mysql_select_db("test");  
?>
```

* Joins:

By using joins we can records from multiple tables based on cond's.

Joins are mainly divided into 3 types.

i) Inner Join 2) Outer Join 3) Cross or Cartesian Join.

1) Inner Join:

Inner Join is again divided into 3 Types.

- a) Equi-join, b) Non Equi Join c) Self Join.

2) Equi Join:

If u want the records from a particular Table, based on equal condⁿ, comes under Equi-Join.

Ex: i) Select emp.ename, dept.dname from emp, dept where emp.deptno = dept.deptno

OR

ii) Select emp.ename, dept.dname from emp inner join dept on emp.deptno = dept.deptno

OR

iii) Select e.ename, d.dname from emp e, dept d where e.deptno = d.deptno

Using inner join (Speed more Executⁿ)

Using Alias name

3) Non-Equi Join:

If u want to get the records from multiple Tables without Equal Condⁿ, we can go for Non-Equi-Join.

Ex:

Select e.name, s.grade from emp e, salgrade s where e.sal between s.lowsal and s.hightsal

OR

Select e.name, s.grade from emp e inner join salgrade s on e.sal between s.lowsal and s.hightsal.

4) Self Join:

If u want to get records from single table using cond's, we can go for self join.

Ex: Select e.ename as Employee, m.ename as Manager from emp e, emp m where e.mgr = m.empid

Select distinct m.ename as Manager from emp e, emp m where e.mgr=m.empid

OR

Select m.ename as Manager from emp e, where e.mgr=m.empid.

* Select e.ename, tbl-city.cname, d.dname, dept.d

Select e.ename, c cname, d dname from emp e, tbl-city c ,dept d
where e.deptno = d.deptno and e.cityid = c.cid.

OR

~~Select e.ename, c cname, d dname from emp e join dept d on
e.deptno = d.deptno join tbl-city c on e.cityid = c.cid~~

II) Outer Join:

To get the matched Records from one Table matching and Unmatched Records from another Table. It is divided into 3 types

a) Left Outer Join:

To get matching and Unmatching records from LHS Table, Matching Records from RHS.

Ex: Select e.ename, d.dname from emp e left outer join dept d on e.deptno = d.deptno.

b) Right Outer Join:

To get the matching Records from LHS, Matching & Unmatching Records from RHS.

Ex: Select e.ename, d.dname from emp e right outer join dept d on e.deptno = d.deptno

c) Fuzz Outer Join:

It is a combination of Left Outer Join & Right Outer Join using an Union keyword, We can combine these 2 Joins.

Ex: Select e.ename, d.dname from emp e left outer join dept d on e.deptno = d.deptno union select e.ename, d.dname from emp e right outer join dept d on e.deptno = d.deptno.

Cross OR Cartesian Join:

This join we're using to get the Records from multiple tables without any cond?

Ex: Select * from emp, dept

Views:

View is a Virtual Table of Physical Table. Views we're using to call the common columns no. of times from a Table.

Views we're using to call the columns from a Table. Instead of using multiple calls of specified columns from a Table we can go for a View.

View doesn't occupy memory. It contains select statement for every call of view, gets the Records from a Table.

Types of Views:> Simple View:

If we create any view by getting the records from single table comes under simple view.

x: Create view vw_view1 as select empid, ename, sal from emp

Create view vw_view1 as select empid as 'Employee Id', ename as 'Employee Name', sal from emp

> Complex View:

If we create any view by getting the records from multiple tables comes under complex views.

x: Create view vw_view3 as select e.ename, e.sal, e.cityid, d.dname from emp e, dept d where e.deptno = d.deptno

* Order By:

By using this ~~Statement~~ class we can change the order of table records. We can get the records in Ascending or Descending order by using this class.

Ex: Select * from dept order by deptno asc

Select * from dept order by deptno desc

By using php:

```
Table border='2'>
?php
if(isset($_REQUEST['qs'])) {
    $ord=$_REQUEST['qs'];
} else
    $ord="desc";
if($ord=="asc")
    $ord="desc";
else
    $ord="asc";
```

```
mysql_connect("localhost", "root", "");
mysql_select_db("test");
$data = mysql_query ("select * from dept order by
deptno $ord");
echo "<H><a href='Ord.php?qs=$ord'>Dept No</a>";
echo "<H>Dept Name";
while ($rec=mysql_fetch_row($data))
{
    echo "<tr><td>$rec[0]<td>$rec[1]</tr>";
}
</table>
```

* mysql2_connect:

It is same as mysql library to connect with mysql database in the php program. But it is providing more functionalities. It requires connectionid as first argument for every sql stat. Execut.

Ex:

```
<?php
$con = mysqli_connect("localhost", "root", " ");
mysqli_select_db($con, "test");
$data = mysqli_query($con, "Select * from dept");
while ($rec = mysqli_fetch_row($data))
{
    echo $rec[0];
    echo $rec[1];
}
?>
```

* Sql injection:

26. OCT. 2012

Sql injection is the concept of passing the sql statements to the s/p controls by the user. User can pass this sql statements to hack the information from a Database.

User can execute their own sql statements to get the information from db of server, Add the username and passwords, ~~Re~~move the DB informⁿ, etc.

User will use s/p controls, querystring, etc. to Hack the informⁿ from DB.

~~User will use some special characters to pass their sql statement thr. the s/p controls.~~

To provide security to the Applicatⁿ, we need to follow these steps

Step1: Don't Accept lengthy stat. from user, thr. the s/p controls.

Step2: Use JavaScript Validation to check whether user is passing Invalid characters thr. the s/p controls.

Step3: Used addslashes, put the backslashes in front of single & double quotes. We can also use mysql_real_escape_string, to put the sql slashes in front of single and double quotes.

Step4: Use magic_quotes_gpc Configuration setting. It is used to put the backslashes in front of single and double quotes. The value should be "ON".

Configuration setting: magic_quotes_gpc=ON

```

<?php
if(isset($_POST['sub']))
{
    $uname=$_POST['txtnum'];
    $pwd=$_POST['txtpwd'];
    mysql_connect("localhost","root","");
    mysql_select_db("test");
    $uname=mysql_real_escape_string($uname);
    $pwd=mysql_real_escape_string($pwd);
    $sqlstr="select * from tb1_user where
        uname='".$uname."' and pwd='".$pwd."'";
    echo $sqlstr."<br>";
    $data=mysql_query($sqlstr);
    if(mysql_num_rows($data)==1)
    {
        echo "Valid";
    }
    else
        echo "Invalid";
}
?>
<form method="post" action=" " >
    Username:<input type='text' name='txtnum'>
    <br/>
    password:<input type='password' name='txtpwd'>
    <br/>
    <input type='submit' name='sub' value='Login'>
</form>

```

SAYE A :

O/P:

* mssql:

(106)

By using this library we can
create a conn? Bet? php applicat?
and sql server database.

This library is providing no.
of fns.

1) mssql_connect():

This f? used to
create a conn? Bet? php applicat?
and sql server database.

2) mssql_select_db():

To select a DB
From a sql server.

3) mssql_query():

To execute an sq?
Statement in mssql server.

```

<?php
    if(mssql_connect("Welcome-pc","sa",
        "123")) {
        echo "Connected";
    }
    else
        echo "Not Connected";
    $data=mssql_query("select * from
        tb1_user");
    while($rec=mssql_fetch_row($data))
    {
        echo $rec[0];
        echo $rec[1];
    }
?>

```

4) mssql_init():

By using this f?, we can
initialize sq? statements.

5) mssql_execute():

To Execute the
initialized sq? statements

write procedure

Store procedure:

27. Oct. 2013 (07)

It is a db inmemory object used to executes multiple SQL

Store procedures we're using to increase the performance of Applications.
Store procedures execute only for 1st call. From 2nd call onwards, get the info from previously executed code.

To create the store procedure, The syntax is

Syntax:

```
Create procedure <name> as begin  
sql stat.  
sql stat.  
end
```

To call the store procedure from MySQL we should use mysqli library.

Ex:

```
delimiter // (To clear Mem. Buffer)  
Create procedure sp_get()  
Begin  
select * from emp;  
END //
```

Ex:

```
Ex: <?php  
$con=mysqli_connect("localhost","root","","");  
mysqli_select_db($con,"test");  
$data=mysqli_query($con,"call sp_get()");  
echo mysqli_error();  
while($rec=mysqli_fetch_row($data))  
{  
echo $rec[0];  
echo $rec[1];  
echo "<br>";  
}  
?>
```

prog. To see records Next, previous
on clicking Next and previous
button.

30.OCT.2019

```
table border='2'>
<php
mysql_connect("localhost", "root", " ");
mysql_select_db("information_schema");
if(isset($_REQUEST['ind']))
$ind = $_REQUEST['ind'];
else
$ind = 0;
$data = mysql_query("select * from character_sets");
$_SESSION['tr'] = mysql_num_rows($data);
//SESSION_DESTROY();
echo "Totrec". $_SESSION['tr'];
$totrec = 5;
$data = mysql_query("select * from character_sets
limit $ind, $totrec");
while ($rec = mysql_fetch_row($data))
{
echo "<tr><td>$rec[0]</td>$rec[1]</td>$rec[2]</td>";
}
echo "</table>";
$pind = $ind - $totrec;
if($pind >= 0)
echo "<a href='PreNxtRec.php?ind=$pind'> PRE </a>";
$mind = $ind + $totrec;
if($nind < $_SESSION['tr'])
echo "<a href='PreNxtRec.php?ind=$nind'> NEXT </a>";
?
}

```

SAVE AS: PreNxtRec.php

Opp:

* Program of Registration form with check Availability, validation

109

```
<noscript>
<meta http-equiv="refresh" content="0; 'error.html' ">
</noscript>
<script src="jquery-1.3.js"></script>
<script src="jquery.validate.js"></script>
<script>
    function chkuser(un)
    {
        document.getElementById('sp1').innerHTML = '<img src=spinner_grey.gif width=20>';
        $.post("check.php", {un: un}, function(data)
        {
            if (data == 0)
            {
                document.getElementById('sp1').style.color = "green";
                document.getElementById('sp1').innerHTML = "Username is Available";
            }
            else
            {
                document.getElementById('sp1').style.color = "red";
                document.getElementById('sp1').innerHTML = "Username is Not Available";
            }
        });
    }

    function fun1()
    {
        $('#frm1').validate();
    }
</script>
<body onload="fun1()">
<form method="post" action="reg.php" id="frm1">
<table>
    <tr> <td> Username <td> <input name='txtruser' class="required" onblur="chkuser(this.value)"> <td> <span id="sp1"> </span>
    <tr> <td> Password <td> <input type="password" id="pwd" name='txtpwd' class="required" minlength="5" >
    <tr> <td> RetypePwd <td> <input type="password" name='txtrpwd' equalTo="#pwd" >
    <tr> <td> Email <td> <input name='txtemail' class="email required" >
    <tr> <td> <input type="submit" name="sub" value="Register" >
</table>
</form>
```

crg.php

```
<?php
$user = $_POST['txtruser'];
$pwd = $_POST['txtpwd'];
$pwd = sha1(md5($pwd));
$email = $_POST['txtemail'];
mysql_connect("localhost", "root", "");
mysql_select_db("test");
if(mysql_query("insert into tb1-user values ('$user', '$pwd', '$email', 0)"))
{
    echo "Registration is completed";
}
else
    echo "Username is already Registered";
?>
```

check.php

```
<?php
$qs = $_REQUEST['tv'];
mysql_connect("localhost", "root", "");
mysql_select_db("test");
$data = mysql_query("select * from tb1-user where uid='$qs'");
echo mysql_num_rows($data);
?>
```

JOOMLA

Installation:

Copy JOOMLA zip file and paste
in htdocs and also extract their,

Rename extracted folder and open at
 browser by typing URL address

`http://localhost/projectfoldername`

default Username and password
⇒ admin

Joomla

(189)
03.OCT.2012
96.OCT.2012

* Joomla: XAMPP 1.5.7. Copy Joomla S/W & paste in htdocs & extract their Joomla.

ii) Joomla is an open source content Management syst. software

implemented by "Mambo group". (Initially Joomla called Mambo Technology, after it becomes as Joomla).

iii) Joomla is providing no. of predefined modules. By using these Modules we can customize the content of Website based on our Requirements.

iv) Joomla Administrator module is providing all manager to customize website content.

v) First we need to install Joomla project in our syst. Then we can add & Remove the modules.

vi) Joomla is available as Open Source licence with diff. version like Joomla 1.x, Joomla 1.5.7, and Joomla 1.5.8, and 2.x etc.

First, we need to install Joomla project in our syst. It requires no. of installations?

* Installation Steps of Joomla: Step 1: After installation, it provides one website. Using this we can manage the content. Joomla website requires RAR.

i) Extract Joomla S/W and place in the htdocs locatⁿ of XAMPP Folder.

ii) Open the Website in Browser by <http://localhost/pre> name of Extracted file in htdocs

(Some supporting tables to run the project, we have to specify database name, Username & password at the time of installation).

Step 1: Language:

choose the language what u want to use at the time of Joomla installation.

(English - United States)

Step 2: Preinstallation check:

In this step, we can see the recommended configuration setting value, we need to change them.

Joomla will still operate even if your settings do not match.

(No need to change, Click on Next)

Step 3: License:

Displays the license informⁿ of Joomla. Click on Next ←

Step 4: Database Configuration: Joomla installs some supporting tables. Joomla is providing some database tables. To run the website we need to specify database server name, userid & passwords. host

hostname: localhost
username: root
password: _____
database name: any

Step 5: FTP Configuration:

Enter FTP servername, Username & password. If we r working

LINUX

with ~~Windows~~ OS no need to specifies FTP configuration settings. If we r working

Step 6: Main Configuration: with windows OS.

By using configuration, we can enter the sitename, ~~site~~ email id, admin username & admin password. Click on install sample data which install

Some ~~sample~~ data i.e. included in the Joomla installation package.

Step 7: Finish:

This is Final step, we need to delete the installation folder to run the

Joomla website. After installation to open url address: <http://localhost/projectfoldername>

To open administrator module URL address is <http://localhost/projectfoldername/administrator>

Go C:/XAMPP/htdocs/myproj & delete installation/~~site~~ project folder

Joomla is providing diff. types of Manager, to manage the content of websites like, Frontpage Manager, Article Manager, Etc.

1) Front Page Manager: By using this option, we can manage Frontpage Articles. To set enable and disable the Front Page articles.

Using this option, Article and uses the directory structure just one template.

2) Template Manager:

By Using this option, we can manage the Templates of Joomla Website. This option is available in the Extensions Menu.

To add New Template Click on Install/Uninstall option of Extension Menu. By default 3 Types of Templates are available. We can also

install new Templates. We can install new Templates in Joomla project using Install/Uninstall option.

3) Module Manager: Can install new Templates in Joomla project using Install/Uninstall option of Extensions Menu. By using this option, we can enable/disable the install option.

modules of Joomla Website. Modules are nothing But Menus, polls, Login form etc.

Module Manager is available in Extensions Menu.

Joomla user groups are divided into 2 types:

Frontend

Backend

Frontend users work with the Joomla website.

Backend users can work with

Plugin Manager:

By using plugin manager, we can enable or disable the plugins. All installed Plugins are explained in this world.

* index.php

It is a file available in Template folder contains the design script of Front page. We can change the Header and Footer inform by using this file.

By using this file we can change the title text of Joomla website.

It is available in project folder.

(Open MyFolder → index.php → Open with Editplus →

Template
is installed
successfully

We can select
the Template
From Template
Manager window

We can select that
Template as a
default Template.

All installed templates
are storing in Templates
folder of Joomla project
Folder. We can also copy &
paste the Templates directly
in this location.

* Article Manager:

By using Article Manager, we can create new articles & we can ~~read~~ edit the existing articles. Joomla is providing TINYMCE editor to edit the article contents. This Editor is providing more options like bold, italic, font styles, etc. to edit article contents. By Default, we cannot execute php script thru the articles. Execute the php script we need to install "Sourcererplugin".

* Installation of Sourcererplugin:

We can install the Sourcererplugin using install/uninstall option of Extensions Menu. Once, the Sourcerer is installed, we can find the source button or Article Manager TINYMCE Editor by using this we can execute html, javascript and php script. We should place this script within the ~~script~~ source declaration syntax.

{source}...{/source}

Go to control panel → Article Manager → Add new Article Manager → New option →

Title: MyArt

Frontpage

Publ.

Alias:

Section:

Print

Extensions → Install/Uninstall → Sourcerer → Next Step

Ex: {source}

```
<?php
$x=100;
$y=200;
$z=$x+$y;
echo "value is $z";
?>
```

{/source}

akindic plugin:

By using this plugin, we can display the content of website in different Indian languages. It is an open source plugin. We can directly install this plugin in joomla website. Path: Install/Install

copy this plugin & paste in plugins folder of project folder / plugin/editors/tinymce/jscripts/tiny_mce folder.

edit file tinymce.php Add 2 lines in tinymce.php file those are :

```
$plugins[ ]='akindicplugin';
$buttons1[ ]=$buttons2[ ]='akindicplugin'; } } gmp
```

~~These lines should follow by a statement i.e. \$buttons2[]='forecolor'~~

These lines we should add followed by a statement i.e. \$buttons2[]='forecolor' once, this plugin is added successfully, we can findout an icon on 'tinymce' editor with a text 'IT'. By using this ~~edit~~ icon we can select different Indian languages

* Article Parameters:

2 Types of parameters are available those are article parameters and Advanced parameters.

By using Article parameters we can change the access level of Article, article created date and time, article published date and time, etc.

By using Advanced parameters we can hide the "pdf" icon, print icon, Mail icon etc. of an Article.

* Metadata Information:

By using this option, we can provide the Meta keywords and description to the Articles.

collect of menu items

* Menu Manager:

By using Menu Manager, we can create new menus. Menus are used to maintain the menu items. menu items we can use to open the internal articles and external webpages.

05.OCT.2012

Module Manager:

If u want to enable or disable the menus, we can go for Module Manager. Module Manager is available in Extensions menu.

Steps To Create Menus And Menu Items:

i) Open Menu Manager, Enter Unique Name, Title, Module Title per new menu.

Steps To Open Internal Links / External Links in Menu Items:

ii) Open menu when u want to create menu item & select a New option from menu item manager of that menu.

iii) Select the link type (Internal link / External link) what you want to use with this menu item. If it is an Article, Click on internal link. Select Article Layout from Articles.

iv) Enter menu item title and then select Article from Basic parameters option.

v) If u want to open external webpage, Click on External link Then Enter Link title and link page.

Menu Item Edit Options:

1) Display in: By using this option, we can change the menu of menu item.

2) Parent item:

By using this option, we can set a parent item to the child item.

3) Published:

To display & hide the menu item.

4) Order:

To change the order of menu item.

5) Access Level:

To specify the access full scope to the link, we have 3 default access specifiers
(1) Public (2) Registered (3) Special.

6) on click, open in:

(194)

To specify the window where U want to open the link page.

We can open the parent window with Browser Navigator, New Window without/with Browser Navigator.

* Login FORM:

By using Login Form, We can Register the New User Accounts as well as we can Login to the Website using Existing User Accounts. When New user Account has been created, it Sends Activat? Email. By using Activat? Link user can Activate his Account.

* User Manager:

By using this option of Admin Module, We can enable and disable the User Accounts. Administrator can also create the New user accounts & He can Edit the Existing user accounts. Myproject / com_user / register / default.php

See all user accounts what are registered in a website

* Polls: By using this option (Poll Manager), We can create the new polls and we can edit existing polls. This option is available in Content Menu. Polls we r using to get the Opinions from users by displaying a question.

Steps To create New Polls:

- 1) Open poll Manager from Content option, Then select New Button, Enter poll title & options. Logtime is the time interval bet? 1 poll & another poll.
- 2) Enable the poll by clicking on published option.
- 3) To display a poll as today poll, Go to the Module Manager, Select Poll option and Then select poll from Module parameters dropdown list.

* Search:

(sections, categories, articles etc)

By using search module, we can search the contents of a website. We can enable and disable the search module, we can go for module manager.

* Media Manager:

By using media manager we can create a new folder in the images folder as well as we can upload the images into the website.

* Concept To create New Control in Registration Form:

Step1: By default, JOOMLA Registrat? form is providing the predefined controls if U want to add New controls, we need to added a file i.e. default.php file. This file is available in project folder / components / com_user / view / register / tmpl folder.

By default Contains the design script of Registrat? form, we need to add New controls at this locat?.

Step2:

To store the Data of Column in Database Table, we need to create a ~~new~~ column in "jos_users" table. The columnname & the controlname should be same.

Step 3:

Add a variable in user.php file. The variable name should be same as control name.
This file is available in project folder/libraries/Joomla/database/table folder.

Var \$params = null;
Var \$Address = null;

* Language Manager:

By using this option, we can change the Language option of Administrator Module. We can change the Left and Right position of module by using position property of that module. To change the posistion, open the module manager, select Section Manager module & then select posistion of that module.

* Section Manager: By using this option, we can create the sections in Joomla project.

It is used to divide a project into different sections.

* Category Manager:

By using this option, we can ~~not~~ create the categories within the sect's

* Global configurations:

By using this option, we can change the configuration settings of a system, size and server.

A) Site Settings:

a) site offline:

To display the website in OFFLINE state

b) OFFLINE MSG:

To change the OFFLINE Text. (We can specify the msg what we want to display when the site is OFFLINE)

c) Site Name:

To change the Name of ^{Web} site

d) Default WYSIWYG Editor:

To change the Editor ~~Type~~ of Article Manager.

By default, it displays "Editor-TinyMCE" editor

e) List Length:

To increase / decrease the total no. of list items per page.

B) Metadata Settings:

By using this option, we can provide the Metakeywords &

Description to the Article.

* SEO Settings:

* Search Engine Friendly URLs:

By using this option, we can enable or disable search engine friendly URL addresses. By default, it is 'Disabled'.

* User setting:

Allow & Stop user Registration, new user activation?

* System Setting:

By using this setting, we can change Log folder location, user setting, session setting.

(195)

Allow User Registration: To allow and stop User Registration.
To allow (Stop User Account Activation)

New User Registration Time: To specify Registration file new user.

All user comes under Registered user. To change the user type of registered user

New User Account Activation: By using this option we can allow & stop the activation link for new users.

* Session Settings: To change the session lifetime and session Handler.

* Media Settings:

By using this option, we can change the Media folder settings.

We can specify max ^{file} size to upload, we can specify the extensions to the specified file. local time settings
(Path) (don't change) To change the FTP settings, Database settings And mail settings.

* FTP Settings: can
(We don't work with FTP settings)

* Locale Settings: To change the Local Time settings.

* Database Settings: To change the DB, servername, userid, password etc. & Also we can change the prefix word.

* Mail Settings: To change the From ^{Name}, from address etc.

* User Types: (User Groups)
User types are mainly divided into 2 types.

- 1) Front-end User Type
- 2) Back-end User Type

1) Front-End User Type: It can access the Website

2) Back-End User Type: It can manage the admin module of Website.

Back-End User Type again divided into 3 types.

a) Super Administrator:

b) Administrator: This user can work with the all modules of Joomla project

c) Administrator: Admin can work with very few features options. Administrator cannot work with Language Manager, Global Configuration, Template Manager etc.

c) Manager: Manager has very few roles. He can't work with Language Manager, Menu Manager
Global Configuration, Tools, Etc.

(197)

* Configuration.php: It is available in project folder. By using this file, we can maintain Global settings. We can also use Configuration settings in Manager. But finally, the values will store at this location.

* Tools

i) Write Message: It is an option available in Tools used to write the mails to the backend users of Joomla project.

ii) Read Message: To Read the Messages.

only Super Administrator can Read and Write the Messages, what he got from another user.

iii) Mass Mail: By using this option, super administrator can send the mails to the groups of Joomla website.

* JFactory: It is a class in Joomla package used to get the info of Logged In User. It is providing getUser() method to get the details of Logged In User. This method is providing username, password, etc. properties. ^{email} ^{to get complete info of logged in user} ^{without features} ~~are available in joomla user table.~~ getUser() is a method used to maintain these properties.

{source}

```
<?php
$user = JFactory::getUser();
$username = $user->username;
$em = $user->email;
echo $username;
echo $em;
?>
```

This should type in Notepad and placed in Article Manager

{/source}

* Page Title: By using this option, we can hide & change the title of front page. It is available in Home menu item of Main Menu. We should change the options of parameters.

* Artisteer:

It is a licensed Tool used to create the templates of Joomla, Drupal etc. These Tool will work with the support of .NET Framework.

* Plugin Manager:

By using Plugin Manager we can enable or disable the Joomla plugins. It is available in Extensions' Menu.

*Steps to Add New Controls in Registration Form:

30.OCT.2014 (198)

- i) User Registration form is available in default.php page which is located in project folder | components | components | com_user | views | register | tmpl.
- ii) We should add controls in this page.
- iii) Add a column in database jos_users table, The columnname should be same as controlname.
- iv) Add a variable in user.php file By assigning null value, The variable name should be same as control name. This file is available in project folder /libraries/joomla/database/Table Folder.

*Prog. of Categorizedwise List

```
→ <select multiple size="10">
  <optgroup label="PHP" style="color: Red">
    <option> Appress </option>
    <option> Black Book </option>
  </optgroup>
</select>
```

*Prog. To play video files.

```
→ <embed src="e:/2.avi" autoplay="false" playcount="2" width="200" height="200">
```

DRUPAL

Document Root

DRUPAL

DRUPAL

(260)

DRUPAL

Sitename: localhost

DRUPAL is a content management system providing no. of predefined modules used to manage the content of website. DRUPAL is open source SW implemented by "Dries" as a message board.

Versions like 5.x, 6.x, 7.x, etc.

DRUPAL is available with different environments. First we need to install this project then we can manage the content of website. By default no. of predefined modules are available in drupal. By using these modules we can manage Installation : (DRUPAL 7.1.5) ~~manually or~~

1) choose profile:

By using this option, we can select the installation type.

2 Types of installations are available

- a) Standard &
- b) Minimal

2) choose Language:

Select the language what u want to use.

3) Verify Requirements:

This option checks the configuration settings what it requires

If all configuration settings ~~match~~ matches, it will redirect to setup db options.

4) DB Setup:

Enter the DB Name, Username, Password, To install the Drupal supporting tables.

5) Configure site:

To Enter the site info like sitename, emailid, admin username, password and email address.

To open the Drupal Website, The URL address is <http://localhost/projectFolderName>

* Content:

Content is used to display Text on Browser. By default, 2 types of contents are available. Posts, Blogs, Articles etc. These contents are available in

1) Articles & Adminstrator menu. To create new content the option is "Create

2) Basic Page Content". In drupal, By default 2 types of articles are available

those are page and story find

By using "Add Content", we can Add New content Using ~~the~~ content we can find the Existing content.

* Options of Content:

1) Text Format:

By using this option, we can specify the type of ~~script~~ format what u want to execute. The available formats are 3 types. Those are

- 1) Filter HTML
- 2) Full HTML
- 3) Plain Text

- 1) Filter HTML supports very few <HTML> Tags.
- 2) Full HTML supports all <HTML> Tags.
- 3) Plain Text doesn't support any <HTML> Tags.

By Default, php script doesn't execute ^{in drupal}. To execute php script we need to install a plugin called php code. By using modules option. We can enable the php code.

Select Modules and check php code then click on save configuration. Once, the php code is installed, we can find out php ^{Code} option in format types. By using this, we can enable the php code in articles.

* Menu Settings:

To create Menus for Articles,

* Revision Information:

To create the New Revisions for DRUPAL contents.

Revisions are used to maintain the Updates what we done on ~~DRUPAL~~ DRUPAL content.

* Comment Settings:

To Enable and Disable the comments.

* Authoring Informations:

By using this option, we can change the Author name, and Article author date and time inform?

* Publishing Options:

To publish and Unpublished Article, promote the Article on Frontpage etc. We r using this option. We can also display article as sticky article or Frontpage.

* Structure:

By using this option, We can change the structure of Application.

It is providing 4 options.

1) Blocks

2) Content Types

3) Menus etc.

4) Themes

Blocks:

By using this option, we can change the regions of Website. Different Types of Regions are available Those r header, sidebarFirst, sidebarSecond, footer etc.

Menus:

By using this option we can create Menus and Menu items. Menu items we r using to open the internal links as well as External links.

Themes:

By using this option, we can change the DRUPAL Website Themes. we can also upload new themes. And customize the Existing themes.

By using configure option, we can change the colors of themes.

Diff. Types of predefined color sets are available. We can select any one of them and also we can customize our own colors.

We can also enable or disable Logo, Sitename, SiteSlogon etc.

Content Management:Content Type:

To create new content type and to ~~added~~ Edit the existing Content Type. By default, DRUPAL 6.8 we have 2 Types of Content Types. These are page and story.

★ Post Settings:

To specify no. of posts on main page and length of Trimmed posts

User Management:

By using this option, we can customize user accounts It also we can create the new user Roles, we can change the User permissions etc.

Roles:

By using this option we can add New user Roles.

Permissions:

To allow and Deny the permissions, to access the DRUPAL Modules By User

Profile:

By using this option, we can add new controls in the User's Registration form it is available in User Management Menu item. First, we need to install the profiles module using modules Menu. By using Modules option, we can install the profiles module. Once, It is install successfully, we can find out profiles menu item in User Management.

Steps to

* Steps To Add New control in Registration Form:

(204)

- 1) Click on profiles from User Management Then select control type (single line Text Field, Multiline Text Field etc). And Enter Title, Category, Name with the New control. Then select Disable in User Registration Form Option & then click on Save Configuration.

User Settings:

By using this option, we can allow and stop the public Registrations, email activation. We can also change the email settings.

Environment:
Local where no. of people communicate

Blogs: By using Blogs option, we can allow the User to create

Book Module:

18.OCT.2012

By using Book Module, we can create the Books and Book pages in DRUPAL website. Once, we added the Book pages, user cannot see the books what we added. To show the books to the user, we need to Enable Book Navigation Option. It is allowing the user to see the Books & Book Pages.

By Default, User can't create Books, To allow the User to create Books need to provide permission to user.

Polls:

(Father of computer: Charles Babbage)

By using this module, we can capture polls from user of diff. Topics in the form of multiple choice Questions.

Search:

By using this module, we can enable the Search options.

Reports:

By using this option, we can see the recent log entries, access denied errors, page Not found Errors Etc. Joomla Using Configuration.php

* Once, The project is Uploaded, To connect with the DB we have to Change the DB servername, username and password in settings.php file, and ~~default~~.settings.php file. These files are available in project folder/sites/default folder.