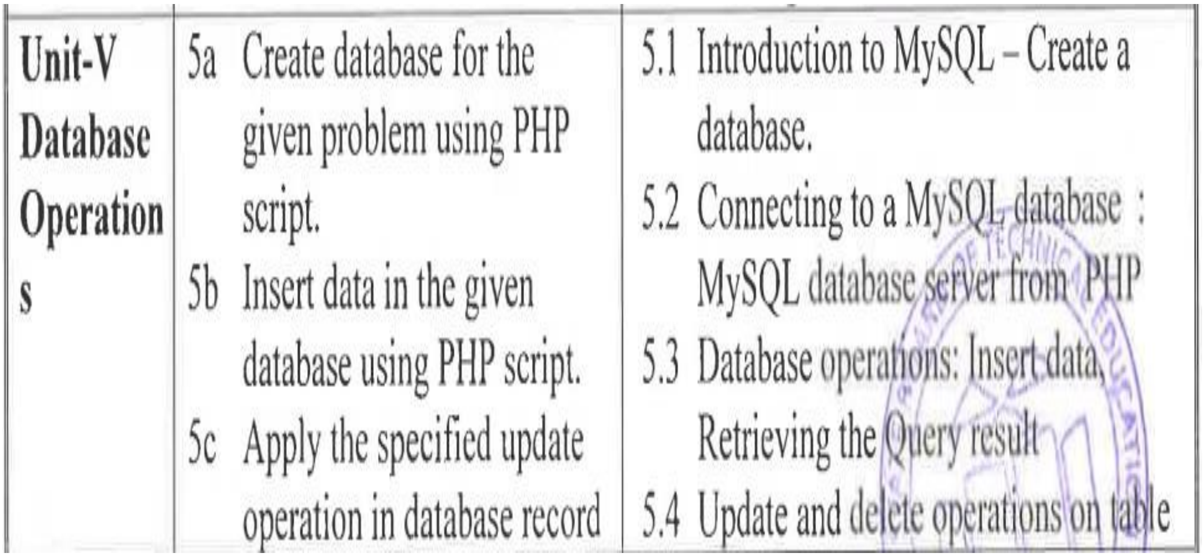

Unit-V Database Operation s	5a Create database for the given problem using PHP script. 5b Insert data in the given database using PHP script. 5c Apply the specified update operation in database record	5.1 Introduction to MySQL – Create a database. 5.2 Connecting to a MySQL database : MySQL database server from PHP 5.3 Database operations: Insert data, Retrieving the Query result 5.4 Update and delete operations on table

Marks: 14 Marks

5.1 Introduction to MySQL- create a database.

- MySQL is a database system used on the web
- MySQL is a database system that runs on a server
- MySQL is ideal for both small and large applications
- MySQL is very fast, reliable, and easy to use
- MySQL uses standard SQL
- MySQL compiles on a number of platforms
- MySQL is free to download and use
- MySQL is developed, distributed, and supported by Oracle Corporation

1.Create database

```
<?php
$servername = "localhost";
$username = "root";
$password = "";/* Put your password here*/
/* Create connection*/
$conn = mysqli_connect($servername, $username, $password);
/* Check connection*/
if (!$conn) {
   die("Connection failed: " . mysqli_connect_error());
}
/* Create database*/
$sql = "CREATE DATABASE stud";
if (mysqli_query($conn, $sql)) {
   echo " $sql created successfully";
}else{
   echo "Error creating database: " . mysqli_error($conn);
}
mysqli_close($conn);
```

```
?>
```

mysqli_connect(host, username, password, dbname, port, socket)

Parameter	Description
host	Optional. Specifies a host name or an IP address
username	Optional. Specifies the MySQL username
password	Optional. Specifies the MySQL password
dbname	Optional. Specifies the default database to be used
port	Optional. Specifies the port number to attempt to connect to the MySQL server
socket	Optional. Specifies the socket or named pipe to be used

The die():- is an inbuilt function in PHP. It is used to print message and exit from the current php script. It is equivalent to the exit() function in PHP.

Syntax :

die($message)

Parameters : This function accepts only one parameter and which is not mandatory to be passed.

$message : This parameter represents the message to be printed while exiting from script.

5.2 Connection to a Mysql database

Php will work with virtually all database software,including oracle and sybase but most commonly used is freely available MYSQL database

php 5 and later can work with MYSQL database using:

1) MYSQLi extension(“the “i” stands for improved”),and

2) PDO(php data objects)

Should I Use MySQLi or PDO?

1. Both MySQLi and PDO have their advantages:
2. PDO will work on 12 different database systems, whereas MySQLi will only work with MySQL databases.
3. So, if you have to switch your project to use another database, PDO makes the process easy.
4. You only have to change the connection string and a few queries. With MySQLi, you will need to rewrite the entire code - queries included.
5. Both are object-oriented, but MySQLi also offers a procedural API.
6. Both support Prepared Statements. Prepared Statements protect from SQL injection, and are very important for web application security.

There are **three ways** of working with MySQl and PHP

MySQLi (object-oriented)

MySQLi (procedural)

PDO

Connecting to MySQL database using PHP

There are 3 ways in which we can connect to MySQl from PHP as listed above and described below:

1. Using **MySQLi object-oriented** procedure: We can use the MySQLi object-oriented procedure to establish a connection to MySQL database from a PHP script.

Syntax:

```
<?php
$servername = "localhost";
$username = "username";
$password = "password";

// Creating connection
$conn = new mysqli($servername, $username, $password);

// Checking connection
if ($conn->connect_error) {
  die("Connection failed: " . $conn->connect_error);
}
echo "Connected successfully";
?>
```

Output:

```
Connected successfully
```

Explanation: We can create an instance of the mysqli class providing all the necessary details required to establish the connection such as host, username, password etc. If the instance is created successfully then the connection is successful otherwise there is some error in establishing connection.

2. Using **MySQLi procedural procedure** : There is also a procedural approach of MySQLi to establish a connection to MySQL database from a PHP script as described below.

Syntax:

```
<?php
$servername = "localhost";
$username = "username";
$password = "password";

// Creating connection
$conn = mysqli_connect($servername, $username, $password);

// Checking connection
if (!$conn) {
  die("Connection failed: " . mysqli_connect_error());
}
```

```
echo "Connected successfully";
?>
```

Output:

```
Connected successfully
```

Explanation: In MySQLi procedural approach instead of creating an instance we can use the mysqli_connect() function available in PHP to establish a connection. This function takes the information as arguments such as host, username , password , database name etc. This function returns MySQL link identifier on successful connection or FALSE when failed to establish a connection.

3. Using **PDO procedure**: PDO stands for PHP Data Objects. That is, in this method we connect to the database using data objects in PHP as described below:

Syntax:

```
<?php
$servername = "localhost";
$username = "username";
$password = "password";

try {
   $conn = new PDO("mysql:host=$servername;dbname=myDB", $username, $password);
   // setting the PDO error mode to exception
   $conn->setAttribute(PDO::ATTR_ERRMODE,  PDO::ERRMODE_EXCEPTION);
   echo "Connected successfully";
   }
catch(PDOException $e)
   {
   echo "Connection failed: " . $e->getMessage();
   }
?>
```

Output:

```
Connected successfully
```

Explanation:The exception class in PDO is used to handle any problems that may occur in our database queries. If an exception is thrown within the try{ } block, the script stops executing and flows directly to the first catch(){ } block.

Closing A Connection

When we establish a connection to a MySQL database from a PHP script , we should also disconnect or close the connection when our work is finished. Here we have described the syntax of closing the connection to a MySQL database in all 3 methods described above. We have assumed that the reference to the connection is stored in $conn variable.

1. Using **MySQLi object oriented procedure**

Syntax

```
$conn->close();
```

2. Using **MySQLi procedural procedure**

Syntax

```
mysqli_close($conn);
```

3. **Using PDO procedure**

Syntax

```
$conn = null;
```

5.3 Database operations

1. Insert operation

```
<?php
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "stud";

// Create connection
$conn = mysqli_connect($servername, $username, $password, $dbname);
// Check connection
if (!$conn) {
  die("Connection failed: " . mysqli_connect_error());
}

$sql = "INSERT INTO co (firstname, lastname, email)
VALUES ('Mohit', 'Jadhawani', 'mohit@gmail.com')";

if (mysqli_query($conn, $sql)) {
  echo "New record created successfully";
} else {
  echo "Error: " . $sql . "<br>" . mysqli_error($conn);
}

mysqli_close($conn);
?>
```

2. Retrieve Data

```
<?php
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "stud";

// Create connection
$conn = mysqli_connect($servername, $username, $password, $dbname);
// Check connection
if (!$conn) {
  die("Connection failed: " . mysqli_connect_error());
}

$sql = "SELECT id, firstname, lastname,email,reg_date FROM co";
$result = mysqli_query($conn, $sql);

if (mysqli_num_rows($result) > 0) {
  // output data of each row
  while($row = mysqli_fetch_assoc($result)) {
      echo "id: " . $row["id"]. " - Name: " . $row["firstname"]. " " . $row["lastname"]."-Email:"."
".$row["email"]."-Time"." ".$row["reg_date"]."<br>";
  }
} else {
  echo "0 results";
}

mysqli_close($conn);
?>
```

3. Update data

```
<?php
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "stud";

// Create connection
$conn = mysqli_connect($servername, $username, $password, $dbname);
// Check connection
if (!$conn) {
  die("Connection failed: " . mysqli_connect_error());
}

$sql = "UPDATE co SET lastname='khan', firstname='shah rukh' WHERE id=3";

if (mysqli_query($conn, $sql)) {
  echo "Record updated successfully";
} else {
  echo "Error updating record: " . mysqli_error($conn);
}

mysqli_close($conn);
?>
```

4. Delete Data.

```
<?php
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "stud";

// Create connection
$conn = mysqli_connect($servername, $username, $password, $dbname);
// Check connection
if (!$conn) {
  die("Connection failed: " . mysqli_connect_error());
}

// sql to delete a record
$sql = "delete FROM co WHERE id=2";

if (mysqli_query($conn, $sql)) {
  echo "Record deleted successfully";
} else {
  echo "Error deleting record: " . mysqli_error($conn);
}

mysqli_close($conn);
?>
```

[1]

[1]