SNS COLLEGE OF ENGINEERING

Kurumbapalayam (Po), Coimbatore – 641 107

An Autonomous Institution

Accredited by NBA – AICTE and Accredited by NAAC – UGC with 'A' Grade
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING-IOT Including CS&BCT

COURSE NAME : 19SB602 FULL STACK DEVELOPMENT FOR NEXT GENERATION IOT

III YEAR / VI SEMESTER

Unit III- **CORE PHP WITH MODEL–VIEW–CONTROLLER**

Topic : Exception, MySQL

PHP Exception Handling

Exception handling is **used to change the normal flow of the code execution** if a specified error (exceptional) condition occurs. This condition is called an **exception.**

The **current code state is saved**

The code execution will switch to a **predefined (custom) exception handler function**

Depending on the situation, the handler **may then resume the execution from the saved code state**, terminate the script execution or continue the **script from a different location in the code**

Different error handling methods:

Basic use of **Exceptions**

Creating a **custom exception handler**

Multiple **exceptions**

Re-throwing an exception

Setting a **top level exception handler**

Note: Exceptions should only be used with error conditions, and should not be used to jump to another place in the code at a specified point.

Basic Use of Exceptions

When an **exception is thrown**, the **code following it will not be executed**, and PHP will try to **find the matching "catch" block.**

If an exception is not caught, a fatal error will be issued with an "Uncaught Exception" message.

Lets try to throw an exception without catching it:

```
<?php
//create function with an exception
function checkNum($number) {
  if($number>1) {
    throw new Exception("Value must be 1 or below");
  }
  return true;
}
//trigger exception
checkNum(2);
?>
```

Try, throw and catch

To avoid the **error from the example above**,

we need to **create the proper code to handle an exception**.

Proper exception code should include:

try - A function using an exception should be in a "try" block. If the exception does not trigger, the code will continue as normal. However if the exception triggers, an exception is "thrown"

throw - This is how you trigger an exception. Each "throw" must have at least one "catch"

catch - A "catch" block **retrieves an exception** and **creates an object containing the exception information**

```php
<?php
//create function with an exception
function checkNum($number) {
  if($number>1) {
    throw new Exception("Value must be 1 or
below");
  }
  return true;
}
```

```
//trigger exception in a "try" block
try {
  checkNum(2);
  //If the exception is thrown, this text will not be shown
  echo 'If you see this, the number is 1 or below';
}

//catch exception
catch(Exception $e) {
  echo 'Message: ' .$e->getMessage();
}
?>
```

Rules for exceptions

Code **may be surrounded in a try block**, to help catch potential exceptions

Each try block or "throw" must have at **least one corresponding catch block**

Multiple **catch blocks can be used to catch different classes of exceptions**

Exceptions can be thrown (or re-thrown) in a catch block **within a try block**

MySQL

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**

MySQL is named after co-founder Monty Widenius's daughter:My

The data in a MySQL database are stored in tables. **A table is a collection of related data**, and it **consists of columns and rows.**

Databases are **useful for storing information categorically.** A company may have a database with the following tables:

Employees
Products
Customers
Orders

PHP + MySQL Database System

PHP combined with MySQL are cross-platform (you can develop in Windows and serve on a Unix platform)

Database Queries

A query is a **question or a request.**

We can **query a database for specific information** and have a **record set returned.**

Look at the following query (using standard SQL):

```
SELECT LastName FROM Employees
```

The query above selects all the data in the "LastName" column from the "Employees" table.

PHP Connect to MySQL

PHP 5 and later can work with a MySQL database using:

MySQLi extension (the "i" stands for improved)

PDO (PHP Data Objects)

Earlier versions of PHP used the MySQL extension. However, this extension was deprecated in 2012.

Open a Connection to MySQL

Before we can **access data in the MySQL database**, we need to be able to **connect to the server**:

Example (MySQLi Object-Oriented)Get your own PHP Server

```
<?php
$servername = "localhost";
$username = "username";
$password = "password";
// Create connection
$conn = new mysqli($servername, $username, $password);
// Check connection
if ($conn->connect_error) {
  die("Connection failed: " . $conn->connect_error);
}
echo "Connected successfully";
?>
```

Create a MySQL Database Using MySQLi and PDO

The CREATE DATABASE statement is used to create a database in MySQL.

The following examples create a database named "myDB":

Example (MySQLi Object-oriented)

```
<?php
$servername = "localhost";
$username = "username";
$password = "password";
// Create connection
$conn = new mysqli($servername, $username, $password);
// Check connection
if ($conn->connect_error) {
  die("Connection failed: " . $conn->connect_error);
}
```

```
// Create database
$sql = "CREATE DATABASE myDB";
if ($conn->query($sql) === TRUE) {
  echo "Database created successfully";
} else {
  echo "Error creating database: " . $conn->error;
}

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

Create a MySQL Table Using MySQLi and PDO

The CREATE TABLE statement is used to create a table in MySQL.

We will create a table named "MyGuests", with five columns: "id", "firstname", "lastname", "email" and "reg_date":

```
CREATE TABLE MyGuests (
id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
firstname VARCHAR(30) NOT NULL,
lastname VARCHAR(30) NOT NULL,
email VARCHAR(50),
reg_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP
)
```

Insert Data Into MySQL Using MySQLi and PDO

After a database and a table have been created, we can start adding data in them.

Here are some syntax rules to follow:

The SQL query must be quoted in PHP
String values inside the SQL query must be quoted
Numeric values must not be quoted
The word NULL must not be quoted
The INSERT INTO statement is used to add new records to a MySQL table:

INSERT INTO table_name (column1, column2, column3,...)
VALUES (value1, value2, value3,...)

Delete Data From a MySQL Table Using MySQLi and PDO

The DELETE statement is used to delete records from a table:

DELETE FROM table_name WHERE some_column = some_value

Any Query????

Thank you……