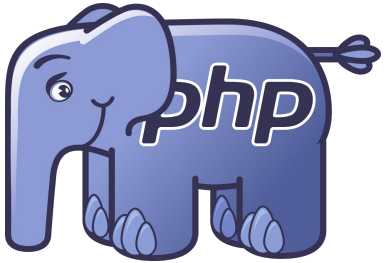
Information Technology Institute



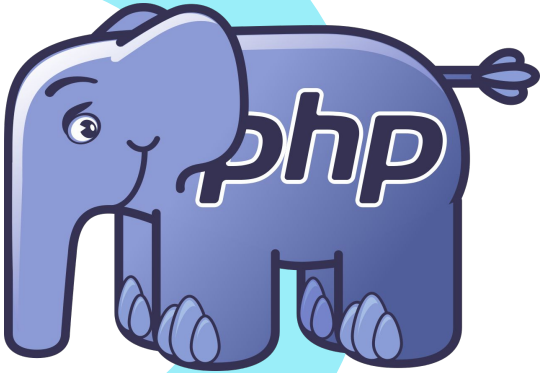
Day04

Prepared by:
Noha Shehab
Teaching Assistant
Information Technology Institute (ITI)

Agenda

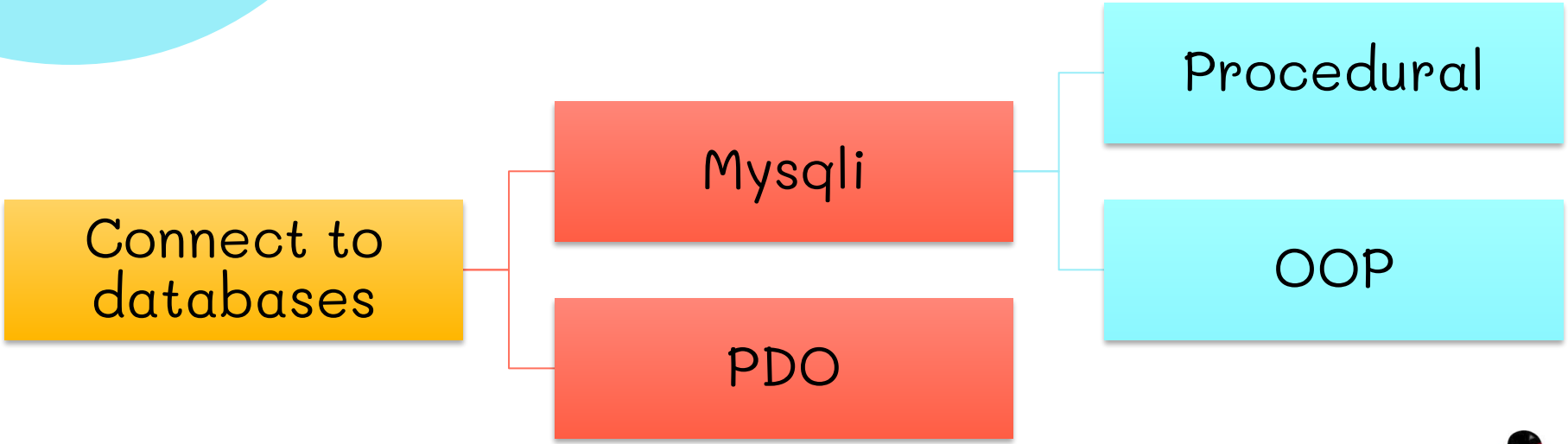
- Connection to databases
- Accessing MySQL Using mysqli
- PDO
- Date and Time Functions

Connection to databases



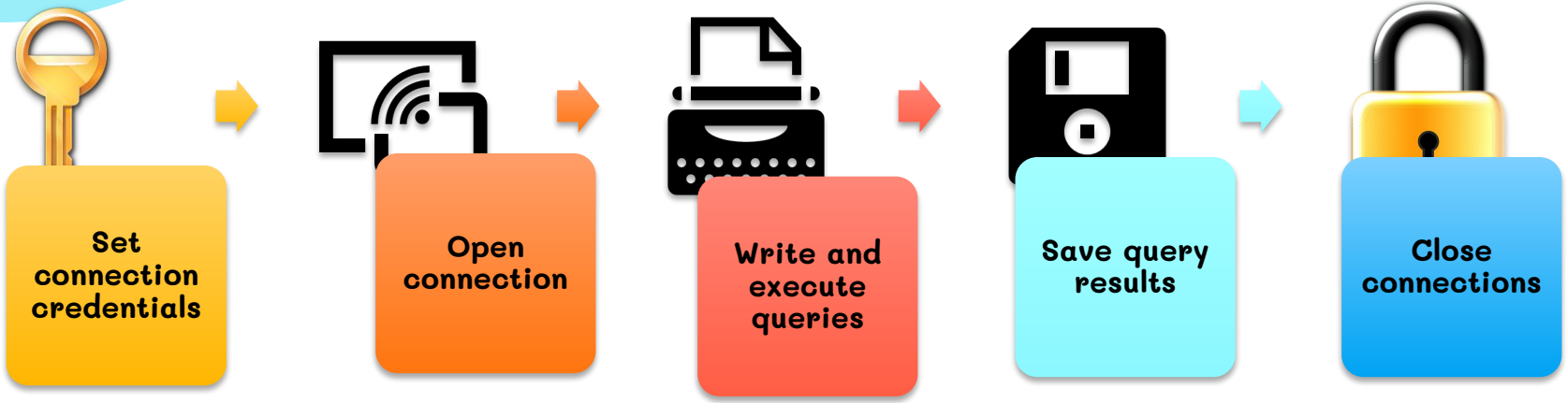
Connection to databases

- To connect to a database you can use either of 2 ways

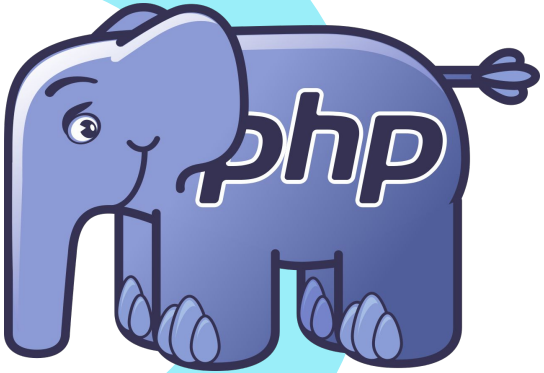


Connection to databases

- Let's connect„



Connecting using Mysqli



Mysqli

- The mysqli interface is an improvement (it means "MySQL Improvement extension") of the mysql interface.
- mysqli is deprecated on php 5.5 and removed in php 7.
- Support procedural style and OOP style.
- Important note: You must put your code inside a try-catch block

Mysqli

- **Procedural programming**:
  - Instructs a computer how to complete a task that you want it to do in logical steps using Functions (in mysqli mysqli_connect(), mysqli_query(), mysqli_fetch_assoc(), mysqli_free_result())

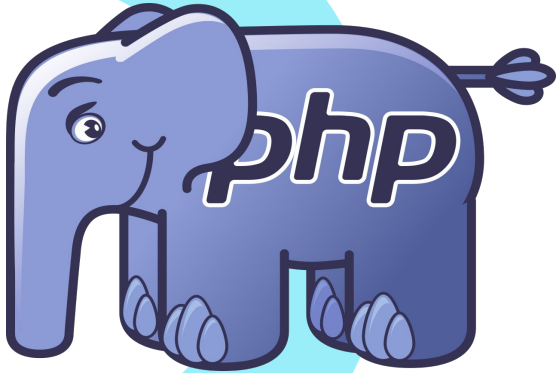
- **OOP**:
  - You have a common classes that handles the task you want and you takes objects from the class so you can access the methods inside class using it.
    Class mysqli

Mysqli

- **Procedural programming**:
  - Instructs a computer how to complete a task that you want it to do in logical steps using Functions (in mysqli mysqli_connect(), mysqli_query(), mysqli_fetch_assoc(), mysqli_free_result())

- **OOP**:
  - You have a common classes that handles the task you want, and you takes objects from the class so you can access the methods inside class using it.
    Class mysqli

Mysqli-Procedural



Connection to database

- Use the function mysqli_connect()

```php
<?php
define("DB_HOST", "localhost");
define("DB_USER", "root");
define("DB_PASSWORD", "");
define("DB_DATABASE", "osgr2");
try {
    $conn = mysqli_connect(DB_HOST, 'root', DB_PASSWORD, DB_DATABASE, 3308);
    var_dump($conn);

     // checking errors
    if (mysqli_connect_errno()) {
        trigger_error(mysqli_connect_error());
        echo "Failed to connect to MySQL: " . mysqli_connect_error();
    }
} catch (Exception $e) {
    echo 'Connection failed: ' . $e->getMessage();
}
```

```
object(mysqli)[1]
  public 'affected_rows' => int 0
  public 'client_info' => string 'mysqlnd 5.0.12-dev
  public 'client_version' => int 50012
  public 'connect_errno' => int 0
  public 'connect_error' => null
  public 'errno' => int 0
  public 'error' => string '' (length=0)
  public 'error_list' =>
    array (size=0)
      empty
  public 'field_count' => int 0
```

Executing query

- Use the function mysqli_query($conn,$query)

```php
$respro = mysqli_query($conn,"select * from students");
var_dump($respro);
```

```
object(mysqli_result)[2]
  public 'current_field' => int 0
  public 'field_count' => int 5
  public 'lengths' => null
  public 'num_rows' => int 16
  public 'type' => int 0
```

- Get the number of rows returned:

```php
$rowCount= mysqli_num_rows($respro);
var_dump($rowCount);
```

```
Day04\MysqliProcedural.php:20:int 16
```

Fetching result rowset

- Fetching rows as associative arrays.

```
'Student_id' => string '8' (length=1)
'Student_name' => string 'Noha' (length=4)
's_add' => string 'test address' (length=1
'email' => string 'noha@gmail.com' (length
'phone' => string '0240495rr' (length=9)
```

```php
while ($row = mysqli_fetch_assoc($respro)) {
    var_dump($row);
    printf ("%s (%s)\n", $row["Student_id"], $row["Student_name"]);
}
```

- Fetching rows as objects:

```
object(stdClass)[4]
  public 'Student_id' => string '8' (length=1)
  public 'Student_name' => string 'Noha' (lengt
  public 's_add' => string 'test address' (leng
  public 'email' => string 'noha@gmail.com' (le
  public 'phone' => string '0240495rr' (length=
```

```php
$result = mysqli_query($conn, "select * from
while ($obj = mysqli_fetch_object($result)) {
    var_dump($obj);
    printf ("%s (%s)\n", $obj->Student_id, $obj->Student_name);
}
```

Fetching result rowset

- mysqli_fetch_row(): retrieve the results in an enumerated array

```php
$result = mysqli_query($conn, "select * from students");
while ($obj = mysqli_fetch_row($result)) {
    var_dump($obj);
}
```

```
array (size=5)
  0 => string '6' (length=1)
  1 => string 'Test' (length=4)
  2 => string 'Menofia' (length=7)
  3 => string 'hhh@gg.com' (length=10)
  4 => string '22345544' (length=8)
```

- Free result set `mysqli_free_result($result);`

Inserting new data

```php
$sql = 'INSERT INTO students (student_name, student_add,email,phone)
    VALUES ("Thursday","Thursday@iti.com", 5445, "Day 4 php")';

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

- Mysqli_insert_id():function returns with the last inserted id.

Updating data

```php
$sql = "UPDATE students set student_name='Mysqli' WHERE  Student_id=28";

if (mysqli_query($conn, $sql)) {
    echo "Record update successfully"."<br>";
    echo mysqli_insert_id($conn)."<br>";
    Var_dump($conn)

} else {
    echo "Error: " . $sql . "<br>" . mysqli_error($conn);
}
```

```
C:\wamp64\www\PHPSmart\Day04\MysqliProcedural.php:62:
object(mysqli)[1]
  public 'affected_rows' => int 1
  public 'client_info' => string 'mysqlnd 5.0.12-dev
  public 'client_version' => int 50012
  public 'connect_errno' => int 0
  public 'connect_error' => null
  public 'errno' => int 0
  public 'error' => string '' (length=0)
  public 'error_list' =>
    array (size=0)
```

Deleting data, Closing connection

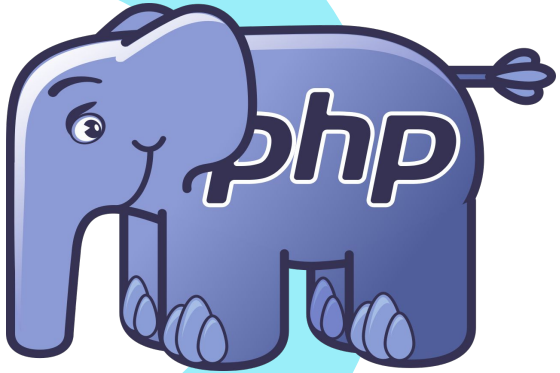
```php
$sql = "DELETE FROM students WHERE  Student_id=26";

if (mysqli_query($conn, $sql)) {
    echo "Query executed successfully"."<br>";
    echo mysqli_insert_id($conn)."<br>";
} else {
    echo "Error: " . $sql . "<br>" . mysqli_error(
}
var_dump($conn);
```

```
Query executed successfully
0

C:\wamp64\www\PHPSmart\Day04\MysqliProcedural.php:69:
object(mysqli)[1]
  public 'affected_rows' => int 0
  public 'client_info' => string 'mysqlnd 5.0.12-dev
  public 'client_version' => int 50012
  public 'connect_errno' => int 0
  public 'connect_error' => null
  public 'errno' => int 0
  public 'error' => string '' (length=0)
```

- Query executed successfully doesn't mean that the value is deleted or not.

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

Mysqli-OOP



Connection to database

- Use new mysqli -> to get an object of mysqli to create the connection.

```php
try {
    $conn2 = new mysqli(DB_HOST, DB_USER,
        DB_PASSWORD, DB_DATABASE,3308);
    // #escaped seq used to make data comptable
     with mySQL but not securing it
    $welcometext='welcome to oop';
    $escaped =$conn2->real_escape_string ($welcometext);
    if ($conn2->connect_errno) {
        trigger_error($conn2->connect_error);
        printf("Connect failed: %s\n", $conn2->connect_error);
        exit();
    }
} catch (Exception $e) {
    echo 'Connection failed: ' . $e->getMessage();
}
```

```
object(mysqli)[1]
  public 'affected_rows' => int 0
  public 'client_info' => string 'mysqlnd 5.0.12-dev
  public 'client_version' => int 50012
  public 'connect_errno' => int 0
  public 'connect_error' => null
  public 'errno' => int 0
  public 'error' => string '' (length=0)
  public 'error_list' =>
    array (size=0)
```

Executing query

- Use the object $conn to call the query function

```php
$resoop= $conn2->query("select * from students");
var_dump($resoop);
```

```
object(mysqli_result)[2]
  public 'current_field' => int 0
  public 'field_count' => int 5
  public 'lengths' => null
  public 'num_rows' => int 16
  public 'type' => int 0
```

- Also notice the num_rows property

```php
$rowCount= $resoop->num_rows;
var_dump($rowCount);
```

```
Day04\MysqliProcedural.php:20:int 16
```

Fetching result rowset

- Fetching rows as associative arrays.

```
while ($row = $resoop->fetch_assoc()) {
    var_dump($row);
    printf ("%s (%s)\n", $row["Student_id"], $row["Student_name"]);
}
```

```
'Student_id' => string '8' (length=1)
'Student_name' => string 'Noha' (length=4
's_add' => string 'test address' (length=
'email' => string 'noha@gmail.com' (lengt
'phone' => string '0240495rr' (length=9)
```

- Fetching rows as objects:

```
if ($result = $conn2 -> query("select * from students")) {
    while ($obj = $result -> fetch_object()) {
        printf("%s (%s)\n", $obj->Student_id , $obj->Student_name);
    }
}
```

```
object(stdClass)[4]
  public 'Student_id' => string '8' (length=1)
  public 'Student_name' => string 'Noha' (length=4
  public 's_add' => string 'test address' (length=
  public 'email' => string 'noha@gmail.com' (lengt
  public 'phone' => string '0240495rr' (length=9)
```

Fetching result rowset

- mysqli_fetch_row(): retrieve the results in an enumerated array

```php
while ($obj = $result -> fetch_row()) {
    var_dump($obj);
}
```

```
array (size=5)
  0 => string '6' (length=1)
  1 => string 'Test' (length=4)
  2 => string 'Menofia' (length=7)
  3 => string 'hhh@gg.com' (length=10)
  4 => string '22345544' (length=8)
```

- Free result set

```php
$result -> free_result();
```

Inserting new data

```php
$sql = 'INSERT INTO students (student_name, student_add,email,phone)
    VALUES ("Nohaaaa","Noha@iti.com", "mansoura", 7800909)';

if($conn2->query($sql)){
    echo "New record created successfully using oop"."<br>";
}else{
    echo "Error: " . $sql . "<br>" .$conn2->connect_error;
}
```

- Return with the last inserted id as following

```php
var_dump($conn2->insert_id);
```

Updating data

```php
$sql = "UPDATE students set Student_name='Mostafa' WHERE  Student_id=10";
if ($conn2->query($sql)) {
    echo "Record update successfully"."<br>";
} else {
    echo "Error: " . $sql . "<br>" . $conn2->connect_error;
}
```

```
object(mysqli)[1]
  public 'affected_rows' => int 1
  public 'client_info' => string 'mysqlnd 5.0.12-dev
  public 'client_version' => int 50012
  public 'connect_errno' => int 0
  public 'connect_error' => null
  public 'errno' => int 0
  public 'error' => string '' (length=0)
  public 'error_list' =>
    array (size=0)
```

Deleting data, Closing connection

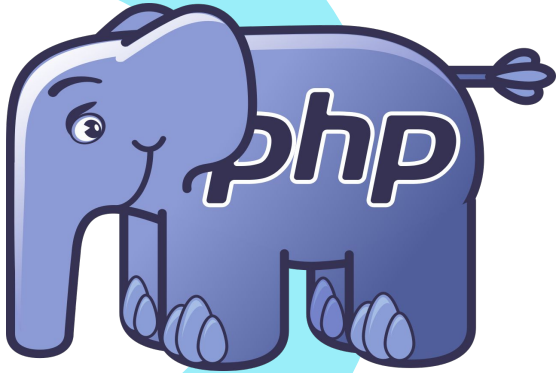
```php
$sql = "DELETE FROM students WHERE  Student_id=12";

if ($conn2->query($sql)) {
    echo "Record deleted successfully"."<br>";
} else {
    echo "Error: " . $sql . "<br>" .$conn2->connect_error;
}
```

```
object(mysqli)[1]
  public 'affected_rows' => int 0
  public 'client_info' => string 'mysqlnd 5.0.12-dev
  public 'client_version' => int 50012
  public 'connect_errno' => int 0
  public 'connect_error' => null
  public 'errno' => int 0
  public 'error' => string '' (length=0)
```

- Query executed successfully doesn't mean that the value is deleted or not.

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

Prepared statements



Prepared statement

- The basic concept of a prepared statement is that you send a template of the query you want to execute to **MySQL and then send the data separately**.
- You can send multiple lots of the same data to the same prepared statement; this capability is particularly useful for **bulk inserts**.
- They are useful for speeding up execution when you are performing large numbers of the same query with different data
- Prepared statements are the recommended solution for the prevention of SQL injection.

Prepared statement

- In mysqli procedural
  - Supports ? Place holder.

```php
$sql = "insert into students (student_name,email) values(?, ?)";
if ($stmt = mysqli_prepare($conn, $sql)) {
    $name="nohaaash";
    $email="noha@iti.com";
    mysqli_stmt_bind_param($stmt, "ss", $name,$email);
    $result = mysqli_stmt_execute($stmt);
    // Fetch data here
    mysqli_stmt_close($stmt);
}
```

Prepared statement

- In mysqli object oriented
  - Supports ? Place holder.

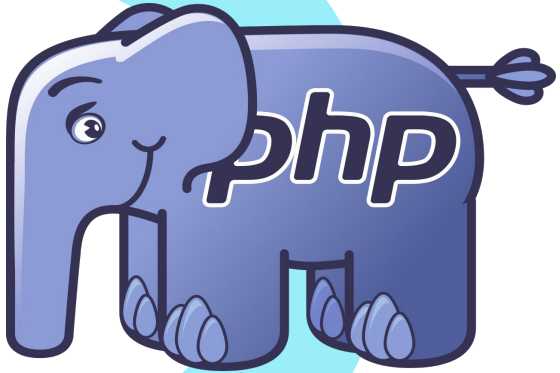
```php
$sql = "insert into students (student_name,email) values(?, ?)";
if ($stmt = $conn2->prepare($sql)) {
    $name="nohaaash";
    $email="noha@iti.com";
    $stmt->bind_param($stmt, "ss", $name,$email);
    $result = $stmt->execute();
    // Fetch data here
    $stmt->close();
}
```

Parameter types...

- You need to define the parameter type when you pass it to the bind_param function

Character	Description
i	corresponding variable has type integer
d	corresponding variable has type double
s	corresponding variable has type string
b	corresponding variable has type binary

PHP Data Object(PDO)



PHP Data Object(PDO)

- The PDO (**PHP Data Objects**) extension allows developers to connect to numerous different types of databases and execute queries against them in a uniform, object-oriented manner.

- PDO provides a data-access abstraction layer, which means that regardless of which database you're using, you use the same functions to issue queries and fetch data.

- PDO supports a wide range of databases such as MS SQL Server, Informix, Oracle, MySQL, PostgreSQL, SQLite

Let's connect

- Take an instance from the PDO class to start the connection

```php
$dsn = 'mysql:dbname=osgr2;host=127.0.0.1;port=3308;'; #port number
$user = 'root';
$password = '';
try {
    $db = new PDO($dsn, $user, $password);
    var_dump($db);

}catch (PDOException $e) {
    echo 'Connection failed: ' . $e->getMessage();
}
```

```
object(PDO)[1]
```

Select statements

- Take an instance from the PDO class to start the connection

```php
$query= "SELECT * FROM students where `student_id` >10;";
$stmt=$db->prepare($query);
$stmt->execute();
$result=$stmt->fetchAll();
var_dump($result);
```

```
array (size=10)
  'Student_id' => string '31' (length=2)
  0 => string '31' (length=2)
  'Student_name' => string 'Nohaa' (length=5)
  1 => string 'Nohaa' (length=5)
  'Student_add' => string 'Mansoura' (length=8)
  2 => string 'Mansoura' (length=8)
  'email' => string 'nshehab@iti.gov.eg' (length=18)
  3 => string 'nshehab@iti.gov.eg' (length=18)
  'phone' => string '123456789' (length=9)
  4 => string '123456789' (length=9)
```

- Display error of the statement

```php
$stmt->errorInfo()
```

PDO and place holders

- PDO supports two kinds of placeholders:
  - Named placeholders. A colon(:), followed by a distinct variable name
  - Traditional SQL positional placeholders, represented as ?

? placeholder

```
$query="Insert INTO students
   (student_name, student_email) Values(?,?)";
$stmt=$db->prepare($query);
$stmt->execute(["Nohaaash",'nshehab@iti.gov.eg']);
$result=$stmt->rowCount();
$id=$db->lastInsertId();
```

Colon : placeholder

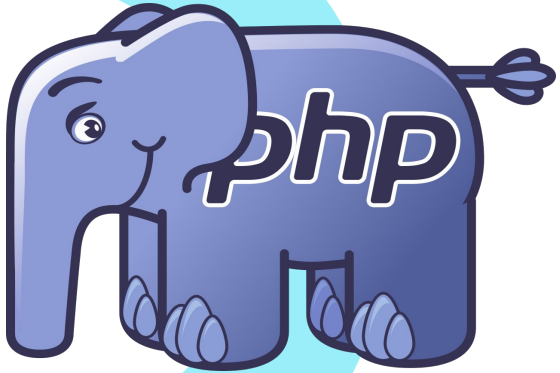
- $stmt->bindParam
- $stmt->bindValue

```php
$query="Insert INTO students (student_name, email)
            Values(:student_name,:student_email)";
$stmt=$db->prepare($query);
$student_name="nohaaaaaaaaaaa";

$stmt->bindParam(":student_name",$student_name, PDO::PARAM_STR);

$stmt->bindValue(":student_email","nshehab@iti.gov.eg");
$stmt->execute();
$result=$stmt->rowCount();
```

PDO Transaction

```php
try {
    $statement = $db->prepare("UPDATE students
        SET student_name = :student_name WHERE student_id=:student_id ");
    $db->beginTransaction();
    $statement->execute(["student_name"=>'OS', "student_id"=>'40']);
    $statement->execute(["student_name"=>'Application', "student_id"=>'45']);

    $db->commit();
}
catch (PDOException $e) {
    if ($db->inTransaction()) {
        $db->rollback();
    }
    throw $e;
}
```

Dates



Date in PHP

- Date() function takes two parameters, one of them optional.
- The first one is a format string, and the second, optional one is a Unix timestamp.
- If you don't specify a timestamp, date() will default to the current date and time.

```
var_dump(date('jS F Y'));
```

```
date.php:3:string '25th March 2021'
```

- Check the full pattern options here.

Date in PHP

- Date() function takes two parameters, one of them optional.
- The first one is a format string, and the second, optional one is a Unix timestamp.
- If you don't specify a timestamp, date() will default to the current date and time.

```
var_dump(date('jS F Y'));
```

```
date.php:3:string '25th March 2021'
```

- Check the full pattern options here.

Date timestamp?

- Unix systems store the current time and date as a 32-bit integer containing the number of seconds since midnight, January 1, 1970, GMT.

- They do have similar problems, though, because they can represent only a limited span of time using a 32-bit integer. If your software needs to deal with events before 1902 or after 2038, you will be in trouble.

Date timestamp?

- If you want to convert a date and time to a Unix timestamp, you can use the mktime() function. It has the following prototype:

```php
var_dump(mktime( hour: 0, minute: 0, second: 0, month: 3, day: 26, year: 2021));
```

```
date.php:10:int 1616716800
```

Get timezone?

- New Datetime() instance..

```php
$date = new DateTime();
$timeZone = $date->getTimezone();
var_dump($timeZone);
echo $timeZone->getName()."<br>";
```

```
object(DateTimeZone)[2]
  public 'timezone_type' => int 3
  public 'timezone' => string 'UTC' (length=3)

UTC
```

```php
$time=time();
var_dump($time);
var_dump(date('U'));
var_dump(getdate()); // returns w
```

```
C:\wamp64\www\PHPSmart\Day04\date.php:18:int 1616713212

C:\wamp64\www\PHPSmart\Day04\date.php:19:string '1616713212' (length=10)

C:\wamp64\www\PHPSmart\Day04\date.php:20:
array (size=11)
  'seconds' => int 12
  'minutes' => int 0
  'hours' => int 23
  'mday' => int 25
  'wday' => int 4
  'mon' => int 3
  'year' => int 2021
  'yday' => int 83
  'weekday' => string 'Thursday' (length=8)
  'month' => string 'March' (length=5)
  0 => int 1616713212
```

Validate dates

- checkdate() function to check whether a date is valid.
- This capability is especially useful for checking user input dates.

```
var_dump(checkdate(2, 29, 2020)); // valid
var_dump(checkdate(2, 29, 2021)); // not valid
```

Format timestamp

- strftime() : format a timestamp according to the system's locale (the web server's local settings)

```
echo strftime('%A')."<br>";
echo strftime('%X')."<br>";
echo strftime('%c')."<br>";
echo strftime('%y')."<br>";
```

Thursday
23:02:44
Thu Mar 25 23:02:44 2021
21

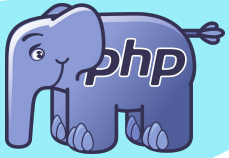
Calculating times in PHP

- A simple way to work out the length of time between two dates in PHP is to use the difference between Unix timestamps.

```php
$bdayunix = mktime  (0, 0, 0, 9,5, 2019);
$nowunix = time(); // get unix ts for  today
$ageunix = $nowunix - $bdayunix;
$time=$ageunix / (365 * 24 * 60 *  60);
var_dump($time);
```

```
date.php:37:float 1.5560797501268
```

Lab 04



- 1- Store the data submitted in this form in the database
- 2- when you submit the data, you will be redirected to a new page that display all users
- Implement the edit, delete operation for each row

Thanks ^^

Noha Shehab
nshehab@iti.gov.eg