PHP y SQL

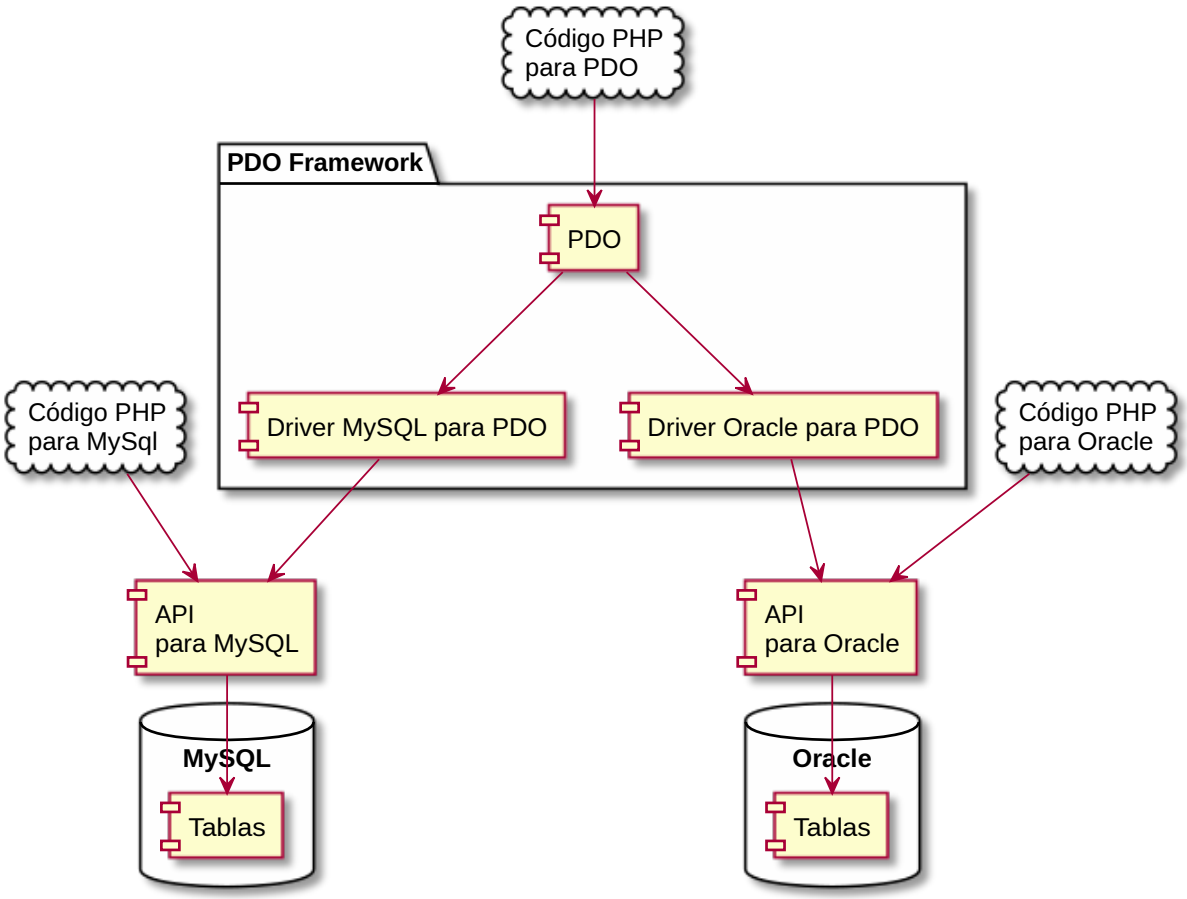
Álvaro González Sotillo

19 de diciembre de 2024

Índice

1. API, *driver* y PDO

- *API*: conjunto de funciones/objetos que permiten conectar a una base de datos SQL
- PDO: conjunto de funciones que pueden acceder a cualquier base de datos, siempre que haya un *driver* para ello
  - El *driver* adapta una API al PDO
- Por ser más simple, utilizaremos el API mysqli



2. `mysqli`: API orientada a objetos o procedimental

- `mysqli` puede usarse de dos formas
  - Como métodos de un objeto conexión
  - Como funciones que reciben una conexión
- No se recomiena mezlarlas

```php
// ORIENTADO A OBJETOS
$conexion = new mysqli("ejemplo.com", "usuario", "contraseña", "basedatos");
$conexion->query("DROP TABLE IF EXISTS test");

// PROCEDIMENTAL
$conexion = mysqli_connect("ejemplo.com", "usuario", "contraseña", "basedatos");
mysqli_query($conexion,"DROP TABLE IF EXISTS test");
```

3. Conexión

- `mysqli_connect`
  - Utiliza opciones por defecto
  - Recibe servidor, usuario, contraseña, base de datos y puerto.
- `mysqli_init -> mysqli_options -> mysqli_real_connect`
  - Permite la configuración de varios parámetros, como el *timeout* de la conexión.
- Al finalizar, `mysqli_close`

3.1. Conexiones persistentes

- Cuando el *script* PHP termina, se cierran todas las conexiones
- Esto puede ser poco eficiente, ya que se debe abrir otra conexión la vez siguiente
- Una conexión persistente puede ser más eficiente. Cada API tiene su forma de conseguirla
  - En `mysqli`, con `p:` delante del nombre de *host*

4. Sentencias DDL

- Con el método `query`

```
$conn->query('
    create table clientes(
      id int auto_increment primary key,
      nombre varchar(255),
      apellidos varchar(255)
    );
');
```

4.1. Ejercicios

1. Crea una tabla como la del ejemplo. Comprueba que se ha creado desde phpMyAdmin o línea de comandos
2. Crea un formulario que pregunte por el nombre de tabla, el nombre y el tipo de tres campos. El botón de submit creará esa tabla.

5. Sentencias DML

- Pasos generales
  1. Lanzar la sentencia
     - Sentencia directa
     - O bien, sentencia preparada
  2. Recoger los resultados
     - El cliente puede requerir todos los datos de una vez, y libera al servidor. Esto puede sobrecargar la memoria del cliente.
     - El cliente puede ir pidiendo datos según necesita. El servidor debe mantener datos en memoria.
  3. O procesar el error

5.1. Todo en memoria del cliente

```
$resultado = $mysqli->query("SELECT id FROM test ORDER BY id ASC");

$resultado->data_seek(0);
while ($fila = $resultado->fetch_assoc()) {
    echo " id = " . $fila['id'] . "\n";
}
```

5.2. Todo en memoria del cliente, en un array

```
$resultado = $mysqli->query("SELECT id FROM test ORDER BY id ASC");

$filas = $mysqli->fetch_all(MYSQLI_ASSOC);
foreach ($filas as $fila) {
    echo " id = " . $fila['id'] . "\n";
}
```

5.3. Como *stream*, sin liberar al servidor

```
$mysqli->real_query("SELECT id FROM test ORDER BY id ASC");
$resultado = $mysqli->use_result();

while ($fila = $resultado->fetch_assoc()) {
    echo " id = " . $fila['id'] . "\n";
}
```

5.4. Control de errores

```
$result = $sql->query("select ...");
if($result) {
    // HA FUNCIONADO
}
else {
    echo "Código de error mysql: $sql->errno";
    echo "Mensaje de error mysql: $sql->error";
}
```

```
try{
    $sql->query( "create ...");
    // HA FUNCIONADO
}
catch(Exception $error){
    echo "Error SQL con try-catch:" . htmlspecialchars( $error->getMessage() ) . "<br>\n";
    echo "Error numérico SQL:" . $conexion->errno . "<br>\n";
    foreach( $conexion->error_list as $error_description ){
        var_dump($error_description );
    }
}
```

- Lista de códigos de error

5.5. Ejercicios

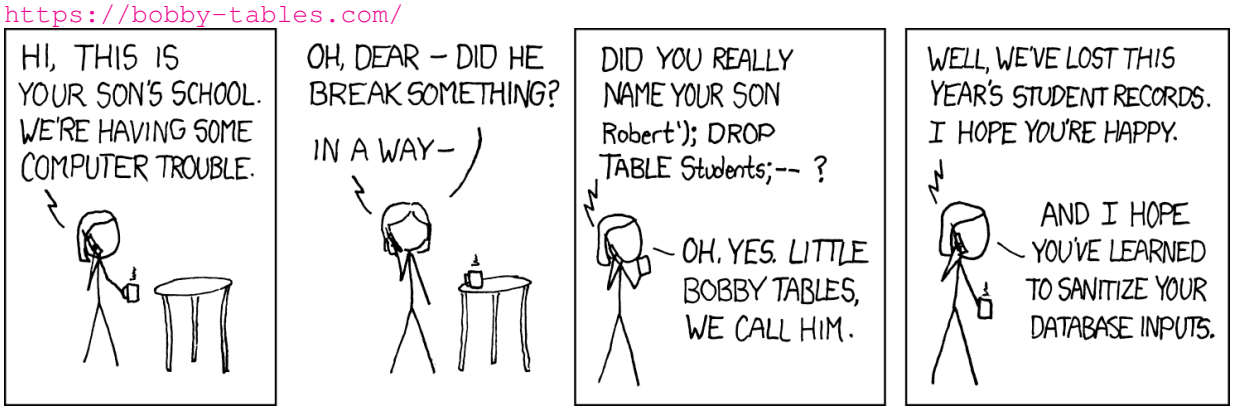
1. Crea una página donde se listen todos los clientes existentes, en forma de tabla.
2. Crea un formulario para insertar un nuevo cliente. Tras insertarse, se mostrarán todos los clientes ya insertados.

6. Sentencias preparadas

- Las sentencias suelen ser cadenas dinámicas
  - Los datos a seleccionar/borrar/actualizar dependen de parámetros
- Existen la tentación de constuir *strings* con las sentencias

```
$id = $_GET["id"];
$mysqli->query("delete from USUARIOS where id='$id'");
```

¿Qué ocurre si `id` es `' or '1'='1' --` ?
https://bobby-tables.com/



6.1. `prepare`, `bind`

https://www.w3schools.com/php/php_mysql_prepared_statements.asp

- Es la única forma segura de pasar cadenas y números a una sentencia SQL
  - `mysqli_real_escape_string` puede tener problemas
- Pasos:
  1. Se prepara la sentencia
  2. Se enlazan los parámetros
  3. Se ejecuta la query
  4. En su caso, se recogen resultados

```php
// prepare sql and bind parameters
 $stmt = $conn->prepare("
   INSERT INTO MyGuests (firstname, lastname, email)
   VALUES (?, ?, ?)"
 );

 $firstname = "John";
 $lastname = "Doe";
 $email = "john@example.com";
 stmt->bind_param("sss", $firstname, $lastname, $email);
 $stmt->execute();
```

```php
    $stmt = $conexion->prepare("select firstname, lastname, email
                                from MyGuests
                                where email like ?");
    $stmt->bind_param("s",$filtro_email);
    $stmt->execute();
    $result = $stmt->get_result();

    while ($row = $result->fetch_assoc()) {
        // Igual que con conexion->query()
        ...
    }
```

6.2. Tipos de datos

`i`	integer
`d`	double, float
`s`	string
`b`	BLOB

6.3. ¿Y las fechas?

- Como cadenas
  - `date`: Formatea a texto un tiempo UNIX
  - `time`: Consigue el tiempo UNIX actual
  - `strtotime`: Pasa de texto a tiempo UNIX
- Precaución: El timezone

```php
$fecha = date('Y-m-d H:i:s');
echo $fecha . "\n";
$unix_time = time();
echo $unix_time . "\n";
$str = strtotime('2024-12-06 19:14:40');
echo $str . "\n";
```

6.4. Ejercicios

1. Crea un formulario para insertar un nuevo cliente. Tras insertarse, se mostrará el nuevo identificador creado
2. Añade un campo `fecha_de_alta` al cliente. Inclúyelo en los listados y las inserciones.

7. Transacciones

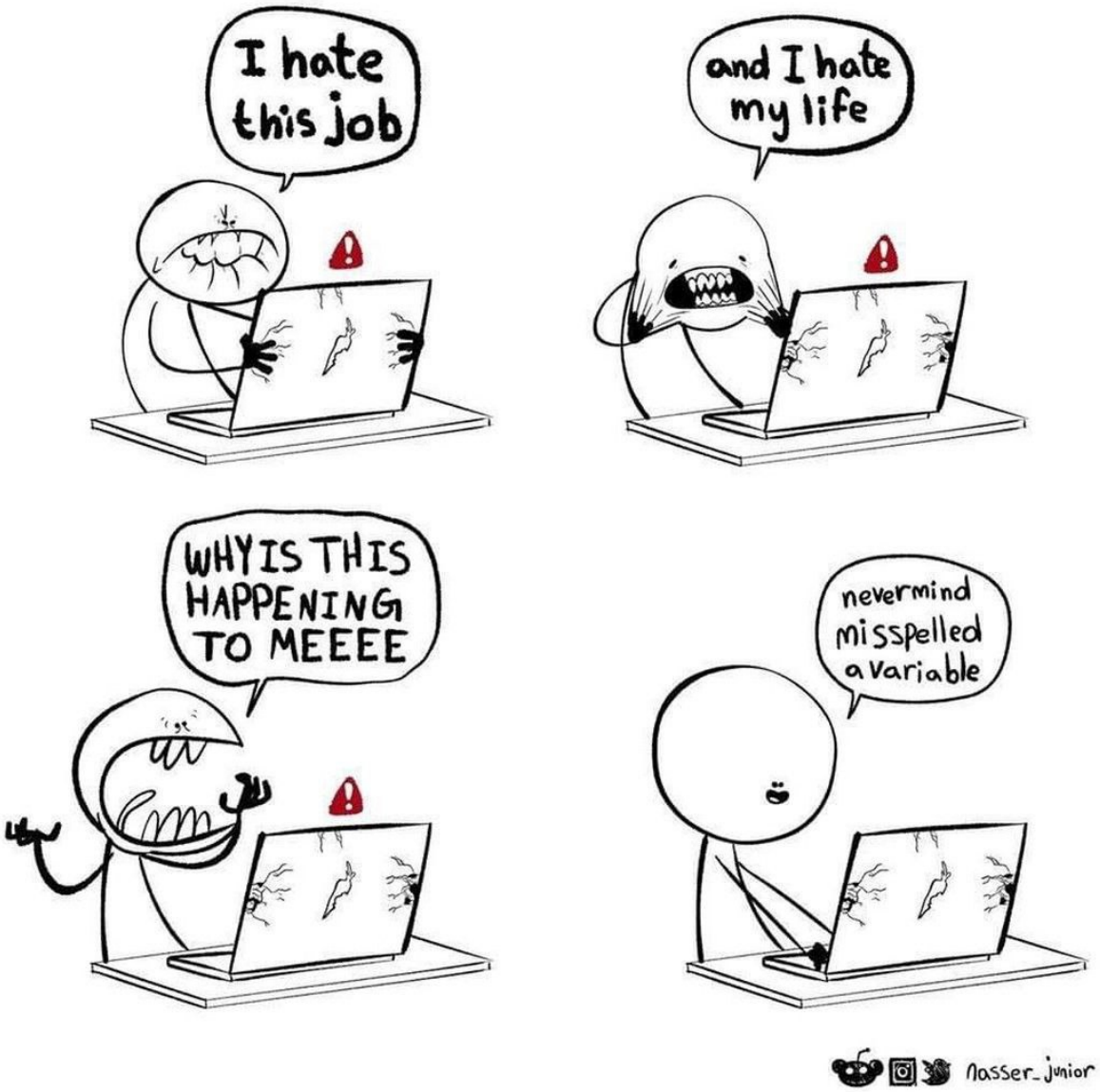
```php
$mysqli = new mysqli("ejemplo.com", "usuario", "contraseña", "basedatos");
$mysqli->autocommit(false);

$mysqli->query("INSERT INTO test(id) VALUES (1)");
$mysqli->rollback();

$mysqli->query("INSERT INTO test(id) VALUES (2)");
$mysqli->commit();
```

8. Sobre todo...

No hay que desesperar



9. Referencias

- Formatos:

    - Transparencias
    - PDF
    - Página web
    - EPUB
- Creado con:
    - Emacs
    - org-re-reveal
    - Latex
- Alojado en Github