

Apuntes PHP para el examen:

Dos vías para entrar a bbdd en php: Mysqli (My SQL Improved) y PDO (PHP Data Objects).

conexión procedural mysqli→ `mysqli_connect(parametros de acceso a la base de datos);`

conexión orientada a objetos mysqli→ `new mysqli(params);`

conexión PDO Orientada a objetos→ `new PDO("mysql:host=...;dbname=db", 'username', 'password');`

Para habilitar a mysqli a enviar reportes de error:

SIEMPRE ANTES DE ABRIR LA BASE DE DATOS.

`mysqli_report(MYSQLI_REPORT_ERROR | MYSQLI_REPORT_STRICT);`

Esquema de conexión y ejecución de comandos SQL:

1 Fase de **Conexión**→ Abrimos la bbdd.

2 Fase de ejecución→ query a la bdd y Fetch si es necesario (cuando queremos datos concretos).

3 Fase de desconexión→ Cierre de la BBDD.

En la fase de conexión se establece la variable con los datos de conexión:

`$con = mysqli_connect(parametros de acceso a la base de datos);`

En la fase de ejecución establecemos una query y la ejecutamos:

`$query = 'SELECT last_name, first_name FROM employees';` Preparacion de la query

`$table = $conn->query($query);` consulta dada la conexión

`foreach ($table as $row) {`

`foreach ($row as $column) {`

`echo $column . ' ';`

`}`

`} Fetch en caso de necesitar datos concretos (Podemos hacer consultas que devuelvan true o false dependiendo de las necesidades de nuestro código)`

`$table->fetch_assoc()` nos devuelve un array asociativo con los nombres de columna como clave y strings como valor asociado a dichas claves para cada campo. se ha de ejecutar por filas como vemos en el foreach mas arriba.

`$table->fetch_array();` se puede utilizar de la misma manera sólo que el contenido nos lo da en claves numérica y para acceder a ellos no podemos usar el nombre de la columna si no el indice numérico del array que equivaldría a esa posición en la tabla.

Se pueden ejecutar consultas preparadas, lo cual es lo idóneo para evitar inyecciones sql:

se consiguen mediante 3 métodos que hay que usar juntos:

```
$stmt= $mysqli->prepare($query); // $mysqli es la instancia de la conexión.  
$stmt->bind_param("isb", $datoInt, $datoString, $datoBool)--> el primer parámetro es en  
orden, el tipo de dato de la variable o valor que le pasamos en los subsecuentes  
parámetros.  
$stmt->execute();
```

Las transacciones en bdd nos permite agrupar y revertir consultas en caso de error

3 métodos:

`begin_transaction()` inicia la transacción.

`commit()` Aplica los cambios de la transacción.

`rollback()` revierte los cambios en caso de error sin deshacer el commit

Estos métodos se aplican sobre la instancia de la conexión `$mysqli->commit();`

`$mysqli->autocommit(true);` se puede instanciar antes del `begin_transaction()` para no tener que hacer commit en caso de que la conexión sea exitosa