

1.1 Introducció

- PHP a través de les seves APIs suporta la majoria de gestors de bases de dades que hi ha al mercat.
- El gestor de BBDD més utilitzat amb PHP és MySQL.
- En aquest apartat veurem les funcions de l'API de PHP que permeten treballar amb MySQL. Cal destacar que totes les funcions comencen amb el prefix `mysql_...`
- Aquesta API avui en dia està totalment obsoleta, existeix una nova API adaptada a programació orientada a objectes `mysqli...`
- Aquesta API permet treballar de manera procedimental i de manera POO, nosaltres per ara veurem la primera forma... més endavant veurem la 2a
- Quan es té una plana php que consulta a una base de dades es poden identificar 4 accions:
 - o Connexió al gestor de la base de dades
 - o Selecció de la base de dades a utilitzar
 - o Execució de la sentència SQL (Consulta/Inserció/Esborrat de dades)
Si es fa una consulta → Recollir i tractar les dades retornades per la BBDD
 - o Tancament de la connexió

1.2 Connexió a la base de dades

- La funció que permet connectar amb PHP a una base de dades MySQL és `mysqli_connect(...)`

```
resource mysqli_connect($servidor[:port], $usuari, $clau,$bbdd);
```

- Aquesta funció retorna un **identificador de connexió** de tipus enter. Si es produeix un error el valor que retorna és `FALSE`.
- Els arguments que rep aquesta funció són:
 - o `$servidor` → és un string que conté l'adreça ip (o url) del servidor on està la base de dades
 - o `$port` → port pel què s'accedeix a la base de dades.
Per defecte MySQL utilitza el port 3306, en aquest cas no cal especificar-lo.
 - o `$usuari` → és un string que conté el nom de l'usuari MySQL que fa la connexió
L'usuari que hi ha creat per defecte a MySQL és: root
 - o `$clau` → és un string que conté la clau de l'usuari (opcional)
 - o `$bbdd` → és un string que conté la base de dades a la que ens connectarem

- Exemple:

```
<?php
    $servidor="localhost";           // "127.0.0.1" o la IP o nom del domini
    $usuari="root";                  // usuari de la bbdd
    $clau="";                        // password
    $bbdd="DAW2";
    $connexio=mysqli_connect($servidor, $usuari, $clau,$bbdd);
?>
```

- Observació:
 - o Sovint es guarden aquestes línies de codi en un fitxer que es pot anomenar `connexio.php`, i després fer un `include()` o `require()` d'aquest fitxer.

1.3 Execució de sentències SQL

- La funció que permet executar una sentència SQL és `mysqli_query(...)`

```
resource mysqli_query($connexio, $sql);
```

- Els arguments que rep són:
 - o `$connexio` → identificador de connexió, que indica a quina connexió s'ha d'enviar la sentència SQL. L'identificador de connexió és el `resource` que ens ha retornat anteriorment la funció `mysqli_connect(...)`
 - o `$sql` → string que conté la sentència SQL a executar
- La funció pot retornar 2 tipus de dades:
 - o INSERT, DELETE → Retorna un valor booleà (TRUE o FALSE) en funció si la sentència s'ha executat bé o malament
 - o SELECT → retorna un identificador de consulta, a partir del qual es podran recuperar totes les dades retornades pel SELECT
- Exemple: INSERT

```
<?php
...
$sql="insert into clients (nom, cognom, adreça)
      values ('$nom','$cognom', '$adreça')";
mysqli_query($connexio, $sql);
?>
```

- Exemple: SELECT

```
<?php
...
$sql="select nom, cognom from clients";
```

```
$res= mysqli_query($conn,$sql);
?>
```

- o Observació
Ara amb altres funcions caldria extreure les dades què ha retornat el SELECT.

1.4 Funcions per recuperar les dades d'un SELECT

- La funció `mysqli_fetch_assoc(...)` permet volcar en un array totes les dades que retorna una consulta

```
array mysqli_fetch_assoc($id_res);
```

- Aquesta funció retorna un array associatiu que conté totes les dades retornades per la consulta SQL.
- Les posicions de l'array associatiu es correspondran amb els noms de les columnes del SELECT que s'ha executat.
- Exemple:

```
<?php
...
$sql = "select * from clients";
$id_res = mysqli_query($connexio,$sql);
while($fila = mysqli_fetch_assoc($id_res)){
    echo $fila["nom"];
    echo $fila["cognom"];
    ...
    //també ho podríem recórrer amb un foreach(...)
}
?>
```

- Observació:
Aquesta segurament és la manera més fàcil de recuperar les dades, fins i tot seria més fàcil si dins el bucle, féssim un `foreach(...)` per recuperar totes les dades de `$fila`.
Existeixen altres funcions que permeten obtenir les dades d'altres maneres.

1.5 Comptar número de files retornades

- Funció `mysqli_affected_rows(...)`
 - o Aquesta funció permet comptar el número de files que ha retornar un SELECT anteriorment executat.
 - o Compte... perquè el paràmetre que cal enviar a la funció és la connexió (i no el resultat)

```
int mysqli_affected_rows($connexio);
```

1.6 Tancament de la connexió

- Un cop executades totes les sentències SQL es recomana tancar la connexió amb la base de dades utilitzant la funció `mysqli_close(...)`
`bool mysqli_close ($connexio);`
- Aquesta funció rep com a paràmetre l'identificador de connexió. Com que normalment només es té una connexió alhora a la base de dades, si no es s'indica cap valor la connexió també es tanca correctament.

1.7 Exemples

Exemple 1:

- A continuació es mostra un exemple de com treballar amb Mysql, utilitzant el mètode tradicional (procedimental):

```
<?php
$servidor="localhost";
$usuari="root";
$password="";
$bbdd="bbddprova";
$connexio = mysqli_connect($servidor, $usuari, $password, $bbdd);
if(mysqli_connect_errno()){
    echo "ERROR MySQL: ".mysqli_connect_error();
    exit();
}else{
    $sql = "select * from cotxe";
    $resultat = mysqli_query($connexio, $sql);
    while($fila = mysqli_fetch_assoc($resultat)){
        echo $fila["marca"]." ";
        echo $fila["model"]."<BR>";
        //...
        //També es podria fer un foreach(...) de $fila
    }
    mysqli_close($connexio);
}
?>
```

Exemple 2:

```
<?php
$con=mysqli_connect("127.0.0.1","root","","ASIX2");
//EXEMPLE SELECT
$sql="select * from alumne";
$resultat=mysqli_query($con,$sql);
while($fila=mysqli_fetch_assoc($resultat)){
    foreach($fila as $key => $value){
        echo "$key: $value";
        echo "<br>";
    }
    echo "<br>";
}
//EXEMPLE INSERT
$sql="insert into alumne values ('Jaume','Viladoms','jv@jv.com','600112233')";
mysqli_query($con,$sql);
echo mysqli_affected_rows($con);
?>
```

1.8 Upload fitxers a la BBDD

1.8.1 Pujar el fitxer a la base de dades

Per pujar un fitxer a la base de dades, podem seguir els següents passos:

1. Cal afegir el camp de tipus BLOB (Binary Large Object) o TinyBlob, MediumBlob, Blob o LongBlob (per no tenir problemes d'espai, es recomana LONGBLOB) a la base de dades, que és on es guardarà la imatge en format binari. També es recomana afegir un camp que guardi el TYPE o tipus del fitxer (quan treballem amb imatges acostuma a ser image/png, image/jpg...).
2. Quan es puja el fitxer al servidor des d'un formulari, no caldrà fer el `move_uploaded_file(...)` sinó que haurem de posar en una variable de PHP, tot el contingut del fitxer.

Això es pot fer de 2 maneres:

- a. Utilitzant la funció `file_get_contents()` per posar tot el contingut del fitxer en una variable:

```
$dadesImatge= file_get_contents($nom_fitxer_tmp);
```
 - b. Obrint el fitxer en la seva ruta temporal, llegint totes les seves dades de cop i guardant-les en una variable:

```
$f = fopen($nom_fitxer_tmp,"r");
$dadesImatge = fread($f, filesize($nomtmp));
fclose($f);
```
3. Un cop tenim les dades guardades en una variable, caldrà protegir el joc de caràcters del fitxer, per a fer això tenim 2 opcions:
 - a. Protegir amb la funció → `mysqli_real_escape_string()`

```
$dadesImatge= mysqli_real_escape_string($connexio, $dadesImatge);
```
 - b. Protegir amb la funció antiga → `addslashes(...)`

```
$dadesImatge= addslashes($dades);
```
 4. A partir d'aquí ja es pot fer l'insert a la bbdd, per exemple:

```
$sql="insert into taula(foto, tipus) values('$dadesImatge', '$tipus');"
```

1.8.2 Mostrar imatges de la base de dades

- Per a mostrar una imatge de la base de dades, hem de recuperar la imatge en binari de la base de dades i posar-la dins l'etiqueta `` amb el següent format:

```
echo '';
```

```
$p="<img src=\"data:\". $tipus .";base64,\".base64($dadesImatge).\">";
```

- Observacions

-
- o Cal tenir en compte que \$tipus conté el TYPE del fitxer → [image/jpg](#), [imatge/png...](#)
 - o Podeu trobar més informació <https://www.w3docs.com/snippets/html/how-to-display-base64-images-in-html.html>

Exercici 1

- a. Crea una base de dades anomenada DAW
- b. Crea una taula anomenada Persona, amb la següent estructura:
`Persona(codi, nom, cognoms, email, telefon)`
- c. Entra dades en aquesta taula
- d. Fes un script en PHP que mostri totes les dades que hi ha a la taula PERSONA

Exercici 2

- a. Fes una plana que a través d'un formulari demani tots els atributs d'una PERSONA (codi, nom, cognoms, email i telèfon) i al clicar el botó, guardi totes aquestes dades a través d'un INSERT a la taula PERSONA.
- b. Fes que en la mateixa plana també es faci el llistat de totes les persones que hi ha a la taula PERSONA.