

## Entorns de Desenvolupament



### 21. PHP i MySQL



Unió Europea  
Fons social europeu  
L'FSE inverteix en el teu futur



## 21. PHP i MySQL

Es pot utilitzar **PHP** per accedir **bases de dades** que estiguen en marxa en un servidor donat, sent MySQL la que més s'utilitza amb aquest llenguatge de programació. MySQL és una base de dades **relacional** (les dades es guarden en taules), és **lliure** (Llicència GPL), està desenvolupada actualment per la companyia Oracle (tot i que Oracle té altres tipus de bases de dades propietàries) i utilitza **SQL estàndard**.

PHP proporciona un conjunt d'elements en la referència del llenguatge que permeten treballar amb facilitat amb aquesta base de dades.

PHP 5 i les seues versions posteriors poden treballar amb una base de dades MySQL utilitzant:

- L'extensió **mysqli** (la "i" significa improved), de manera **procedimental (amb funcions)** o **orientada a objectes**.
- **PDO** (PHP Data Objects), Objectes de dades PHP.

Qualsevol de les dues opcions proporciona les **mateixes opcions** per treballar amb PHP i MySQL. En tot cas, hem de tenir en compte que:

- Les dues tècniques són orientades a objectes, però MySQLi també ofereix una **API de procediment**.
- Les dues **protegeixen de la injecció de SQL**, un aspecte molt important per a la seguretat de les aplicacions web.
- **PDO** permet treballar en **12 sistemes de bases de dades diferents**, mentre **MySQLi només** funciona amb bases de dades **MySQL**. Per tant, si per alguna raó s'ha de canviar el projecte per utilitzar una altra base de dades, PDO fa que el procés més fàcil. Només cal canviar la cadena de connexió i algunes consultes. Amb MySQLi, s'haurà de tornar a escriure el codi complet - consultes incloses.

En els exemples que veurem utilitzarem només l'extensió **mysqli**, però mostrant les possibles solucions amb la versió orientada a objectes i procedimental.

## 1. Connexió amb una base de dades MySQL

Abans de poder accedir a les dades a la base de dades MySQL, hem de ser capaços de **connectar amb el servidor**:

Veiem un exemple utilitzant les expressions **procedimentals**:

```
<?php
$servidor = "localhost";
$usuari = "root";
$contrasenya = "";
try { // Creem la connexió
    $connexio = mysqli_connect($servidor, $usuari, $contrasenya);
    echo "Connectat correctament";
} catch (Exception $e) { // hi ha hagut algun error
    die("Error en connectar a la base de dades: ".$e);
}
?>
```

Ara fem el mateix utilitzant la notació **orientada a objectes**:

```
<?php
$servidor = "localhost";
$usuari = "root";
$contrasenya = "";
try { //creem connexió
    $connexio = new mysqli($servidor, $usuari, $contrasenya);
    echo "Connectat correctament";
} catch (Exception $e) { //hi ha hagut algun error
    die("Error en connectar a la base de dades: ".$e);
}
?>
```

## Tancar la connexió amb una base de dades MySQL

La connexió es tancarà automàticament quan acabe l'execució del codi de l'script. Però si volem tancar la connexió abans, hem d'utilitzar:

En la versió procedimental:

```
mysqli_close($connexio);
```

Per exemple:

```
...
try { //creem connexió
    $connexio = new mysqli($servidor, $usuari, $contrasenya);
    echo "Connectat correctament";
} catch (Exception $e) { //hi ha hagut algun error
    die("Error en connectar a la base de dades: ".$e);
}
mysqli_close($connexio);
```

En la versió orientada a objectes:

```
$connexio->close();
```

Per exemple:

```
...
try { //creem connexió
    $connexio = new mysqli($servidor, $usuari, $contrasenya);
    echo "Connectat correctament";
} catch (Exception $e) { //hi ha hagut algun error
    die("Error en connectar a la base de dades: ".$e);
}
$connexio->close();
```

## 2. Crear base de dades i taula

Una vegada establerta la connexió, tan sols cal realitzar les diferents **consultes SQL** desitjades sobre una **base de dades triada** i obtenir-ne el resultat.

Això sí, **primer** haurem de **seleccionar** la **base de dades** a utilitzar dins del servidor. Suposem que inicialment ni tan sols existeix la **base de dades**, de manera que haurem d'executar la instrucció que permet **crear-la**.

**Procedimentalment**, s'utilitza la instrucció:

```
mysqli_query($connexio,$instruccio_sql);
```

On **\$instrucció\_sql** ha de contenir una instrucció en sql vàlida.

```
<?php
$servidor = "localhost";
$usuari = "root";
$contrasenya = "";
try { // Creem la connexió
    $connexio = mysqli_connect($servidor, $usuari, $contrasenya);
    echo "Connectat correctament";
} catch (Exception $e) { // hi ha hagut algun error
    die ("Error en connectar a la base de dades: ".$e);
}
// Creem la base de dades, introduïm la instrucció SQL en la variable
$sql = "CREATE DATABASE dawprova";
try{ // Executem la instrucció, comprovant si hi ha error
    if (mysqli_query($connexio, $sql))
        echo "Base de dades creada amb èxit";
} catch (Exception $e) { //si hi ha hagut algun error
    echo "Error en crear la base de dades: " . mysqli_error($connexio);
}
```

//Tanquem la connexió abans d'acabar

```
mysqli_close($connexio);  
?>
```

En la versió orientada a Objectes:

```
<?php  
$servidor = "localhost";  
$usuari = "root";  
$contrasenya = "";  
try { //creem connexió  
    $connexio = new mysqli($servidor, $usuari, $contrasenya);  
    echo "Connectat correctament";  
} catch (Exception $e) { //hi ha hagut algun error  
    die("Error en connectar a la base de dades: ".$e);  
}  
// Creem la base de dades, introduïm la instrucció SQL en la variable  
$sql = "CREATE DATABASE dawprova";  
try{ // Executem la instrucció, comprovant si hi ha error  
    if ($connexio->query($sql) === TRUE)  
        echo "Base de dades creada amb èxit";  
} catch (Exception $e) { //si hi ha hagut algun error  
    echo "Error en crear la base de dades: " . $connexio->error;  
}  
//Tanquem la connexió abans d'acabar  
$connexio->close();  
?>
```

Ara, per executar instruccions sql sobre una base de dades existent, primer l'hem de triar, **afegint un nou paràmetre a la connexió**. Per exemple, si volguérem crear una taula dins de la base de dades **dawprova** que hem creat abans, tindríem, en la versió procedimental:

```
<?php
$servidor = "localhost";
$usuari = "root";
$contrasenya = "";
$basedades = "dawprova"; //nom de la base de dades a la que connectem
try { // Creem la connexió
    $connexio=mysqli_connect($servidor,$usuari,$contrasenya,$basedades);
    echo "Connectat correctament";
} catch (Exception $e) { // hi ha hagut algun error
    die ("Error en connectar a la base de dades: ".$e);
}
// Creem la base de dades
// Introduïm la instrucció SQL en la variable $sql per a crear la taula
$sql = "CREATE TABLE usuaris (
id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
nom VARCHAR(30) NOT NULL,
cognoms VARCHAR(30) NOT NULL,
email VARCHAR(50),
data TIMESTAMP
)";
try { // Executem la instrucció, comprovant si hi ha error
    if (mysqli_query($connexio, $sql))
        echo "Taula usuaris creada amb èxit";
} catch (Exception $e) { //si hi ha hagut algun error
    echo "Error en crear la taula: " . mysqli_error($connexio);
}
//Tanquem la connexió abans d'acabar
mysqli_close($connexio);
?>
```

| [orientat a objectes](#):

```
<?php
$servidor = "localhost";
$usuari = "root";
$contrasenya = "";
$basedades = "dawprova";
try { //creem connexió
    $connexio = new mysqli($servidor, $usuari, $contrasenya,$basedades);
    echo "Connectat correctament";
} catch (Exception $e) { //hi ha hagut algun error
    die("Error en connectar a la base de dades: ".$e);
}
// Creem la base de dades
// Introduïm la instrucció SQL en la variable $sql per a crear la taula
$sql = "CREATE TABLE usuaris (
    id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
    nom VARCHAR(30) NOT NULL,
    cognoms VARCHAR(30) NOT NULL,
    email VARCHAR(50),
    data TIMESTAMP
)";
try { // Executem la instrucció, comprovant si hi ha error
    if ($connexio->query($sql) === TRUE)
        echo "Taula usuaris creada amb èxit";
} catch (Exception $e) { //si hi ha hagut algun error
    echo "Error en crear la taula: " . $connexio->error;
}
//Tanquem la connexió abans d'acabar
$connexio->close();
?>
```



### 3. Operacions sobre taules

Una vegada disposem de base de dades i taules per treballar podem realitzar operacions de selecció, inserció, modificació i eliminació.

#### Inserir Dades

En aquest cas, hi ha algunes regles de sintaxi a seguir:

- La **consulta SQL** ha d'estar **entre cometes dobles** en PHP.
- Els **valors de cadena** dins de la consulta SQL han d'estar **entre cometes simples**.
- Els **valors numèrics** no han de portar **cap tipus de cometes**.
- La paraula **NULL** **tampoc** ha d'estar **entre cometes**.

La instrucció **INSERT INTO** s'utilitza per afegir nous registres a una taula de MySQL:

```
INSERT INTO nom_taula (columna1, columna2, columna3, ...)  
VALUES (valor1, valor2, valor3, ...)
```

Si tenim algun camp que siga **autoincremental**, podem obtenir el valor de l'últim element inserit amb:

```
$ultim_id = mysqli_insert_id($connexio); //procedimental  
  
o  
  
$ultim_id = $connexio->insert_id; //orientat a objectes
```

En format **procedimental**:

```
<?php  
$servidor = "localhost";  
$usuari = "root";  
$contrasenya = "";  
$basedades = "dawprova";  
try { // Creem la connexió  
    $connexio=mysqli_connect($servidor,$usuari,$contrasenya,$basedades);
```

```
    echo "Connectat correctament";
} catch (Exception $e) { // hi ha hagut algun error
    die ("Error en connectar a la base de dades: ".$e);
}
// Introduïm la instrucció SQL en la variable $sql
$sql = "INSERT INTO usuaris (nom, cognoms, email, data)
VALUES ('Anna', 'Rovira', 'anna@gmail.com', now())";
// Executem la instrucció, comprovant si hi ha hagut algun tipus d'error
if (mysqli_query($connexio, $sql)) {
    $ultim_id = mysqli_insert_id($connexio);
    echo "Nou registre creat amb èxit. Últim id: ".$ultim_id;
} else {
    echo "Error: ". $sql . "<br/>" . mysqli_error($connexio);
}
mysqli_close($connexio); //Tanquem la connexió abans d'acabar
?>
```

I en format **orientat a objectes**:

```
<?php
$servidor = "localhost";
$usuari = "root";
$contrasenya = "";
$basedades = "dawprova";
try { //creem connexió
    $connexio = new mysqli($servidor, $usuari, $contrasenya,$basedades);
    echo "Connectat correctament";
} catch (Exception $e) { //hi ha hagut algun error
    die("Error en connectar a la base de dades: ".$e);
}
// Introduïm la instrucció SQL en la variable $sql
$sql = "INSERT INTO usuaris (nom, cognoms, email,data)
VALUES ('Anna', 'Rovira', 'anna@gmail.com', now())";
```

```
// Executem la instrucció, comprovant si hi ha hagut algun tipus d'error
if ($connexio->query($sql) === TRUE) {
    $ultim_id = $connexio->insert_id;
    echo "Nou registre creat amb èxit. Posició: ".$ultim_id;
} else {
    echo "Error: ". $sql . "<br/>" . $connexio->error;
}
//Tanquem la connexió abans d'acabar
$connexio->close();
?>
```

Tenim una instrucció que permet **executar més d'una instrucció sql separades per comes**, en lloc d'utilitzar com fins ara

```
mysqli_query($connexio, $sql); //procedimental
o
$connexio->query($sql); //orientat a objectes
```

Suposant que ara **\$sql** és una composició d'instruccions com per exemple:

```
$sql = "INSERT INTO usuaris (nom, cognoms, email, data)
VALUES ('Manuel', 'González Ferri', 'manuel@gmail.com',now());";
$sql .= "INSERT INTO usuaris (nom, cognoms, email, data)
VALUES ('Maria', 'Sanchis Coll', 'maria@gmail.com',now());";
$sql .= "INSERT INTO usuaris (nom, cognoms, email, data)
VALUES ('Gemma', 'Jiménez Serna', 'gemma@gmail.com',now());";
```

Executariem les ordres:

```
mysqli_multi_query($connexio, $sql) //procedimental
o
$connexio->multi_query($sql) //orientat a objectes
```

## Seleccionar dades

La instrucció **SELECT** s'utilitza per **seleccionar dades** d'una o més taules:

```
SELECT nom_columna(es) FROM nom_taula
```

O podem utilitzar el caràcter **\*** per seleccionar **totes les columnes** d'una taula:

```
SELECT * FROM nom_taula
```

El **procediment** per seleccionar les dades en l'exemple que veurem més endavant serà el següent:

- En primer lloc, **creem una consulta SQL** que selecciona les columnes ID, nom i cognoms de la taula usuaris.
- Després s'**executa la consulta** i posa les **dades resultants en una variable** anomenada **\$resultat**.
- A continuació, amb **\$resultat->num\_rows** o **mysqli\_num\_rows(\$resultat)** **comprovem si hi ha més de zero files retornades**.
- Si es retornen més de zero files, amb **\$row = mysqli\_fetch\_assoc(\$result)** o **\$row = \$resultat->fetch\_assoc()** es posen tots els **resultats en un array associatiu** que es pot recórrer. Els índex de l'array es corresponen amb els noms dels camps de la taula.
- El bucle **while** **recorre el conjunt de resultats i dóna eixida** a les dades de les columnes **id**, **nom** i **cognoms**.

Veiem el resultat en versió **procedimental**:

```
<?php
$servidor = "localhost";
$usuari = "root";
$contrasenya = "";
$basedades = "dawprova";
try { // Creem la connexió
    $connexio=mysqli_connect($servidor,$usuari,$contrasenya,$basedades);
    echo "Connectat correctament";
} catch (Exception $e) { // hi ha hagut algun error
```

```
    die ("Error en connectar a la base de dades: ".$e);
}
// Creem la consulta select
$sql = "SELECT id, nom, cognoms FROM usuaris";
$resultat = mysqli_query($connexio, $sql);
// Executem la instrucció, comprovant si hi ha hagut algun tipus d'error
i si n'ha retornat més de 0
if (mysqli_num_rows($resultat) > 0) {
// Dades d'eixida de cada fila
    while($row = mysqli_fetch_assoc($resultat)) {
        echo "id: ".$row["id"]." - Nom: ".$row["nom"]." ".$row["cognoms"]. "<br/>";
    }
} else { // No s'ha trobat cap resultat
    echo "0 resultats";
}
//Tanquem la connexió abans d'acabar
mysqli_close($connexio);
?>
```

I en versió [orientat a objectes](#):

```
<?php
$servidor = "localhost";
$usuari = "root";
$contrasenya = "";
$basedades = "dawprova";
try { //creem connexió
    $connexio = new mysqli($servidor, $usuari, $contrasenya,$basedades);
    echo "Connectat correctament";
} catch (Exception $e) { //hi ha hagut algun error
    die("Error en connectar a la base de dades: ".$e);
}
```

```
// Creem la consulta select
$sql = "SELECT id, nom, cognoms FROM usuaris";
$resultat = $connexio->query($sql);
// Executem la instrucció, comprovant si hi ha hagut algun tipus d'error
i si n'ha retornat més de 0
if ($resultat->num_rows > 0) {
    // Dades d'eixida de cada fila
    while($row = $resultat->fetch_assoc()) {
        echo "id: ".$row["id"]." - Nom: ".$row["nom"]." ".$row["cognoms"]."<br>";
    }
} else { // No s'ha trobat cap resultat
    echo "0 resultats";
}
// Tanquem la connexió abans d'acabar
$connexio->close();
?>
```

## Eliminar dades

La sentència **DELETE** s'utilitza per **eliminar registres d'una taula**:

```
DELETE FROM nom_taula
ON alguna_columna = algun_valor
```

Per exemple, en la taula que hem creat:

Id	Nom	Cognoms
19	Manuel	González Ferri
20	Maria	Sanchis Coll
21	Gemma	Jiménez Serna

Si volguérem eliminar l'element amb **id = 20**, tindríem el següent codi, en **procedimental**:

```
<?php
$servidor = "localhost";
$usuari = "root";
$contrasenya = "";
$basedades = "dawprova";
try { // Creem la connexió
    $connexio=mysqli_connect($servidor,$usuari,$contrasenya,$basedades);
    echo "Connectat correctament";
} catch (Exception $e) { // hi ha hagut algun error
    die ("Error en connectar a la base de dades: ".$e);
}
$sql = "DELETE FROM usuaris WHERE id=20"; // sql per eliminar un registre
if (mysqli_query($connexio, $sql)) { // Executem la instrucció,
    echo "Registre eliminat correctament";
} else {
    echo "Error eliminant registre" . mysqli_error($connexio);
}
mysqli_close($connexio); // Tanquem la connexió abans d'acabar
?>
```

I ara **orientat a objectes**:

```
<?php
$servidor = "localhost";
$usuari = "root";
$contrasenya = "";
$basedades = "dawprova";
try { //creem connexió
    $connexio = new mysqli($servidor, $usuari, $contrasenya,$basedades);
    echo "Connectat correctament";
} catch (Exception $e) { //hi ha hagut algun error
    die("Error en connectar a la base de dades: ".$e);
}
```

```
// sql per eliminar un registre
$sql = "DELETE FROM usuaris WHERE id=20";

// Executem la instrucció
if ($connexio->query($sql) === TRUE) {
    echo "Registre eliminat correctament";
} else {
    echo "Error eliminant registre" . $connexio->error;
}
//Tanquem la connexió abans d'acabar
$connexio->close();
?>
```

## Actualitzar dades

La instrucció **UPDATE** s'utilitza per actualitzar els registres existents en una taula:

```
UPDATE nom_taula
SET columna1 = valor, columna2 = valor2, ...
ON alguna_columna = algun_valor
```

Per exemple, en la taula que hem creat:

Id	Nom	Cognoms
19	Manuel	González Ferri
21	Gemma	Jiménez Serna

En format procedimental:

```
<?php
$servidor = "localhost";
$usuari = "root";
$contrasenya = "";
$basedades = "dawprova";
```



```
try { // Creem la connexió
    $connexio=mysqli_connect($servidor,$usuari,$contrasenya,$basedades);
    echo "Connectat correctament";
} catch (Exception $e) { // hi ha hagut algun error
    die ("Error en connectar a la base de dades: ".$e);
}
// sql per eliminar un registre
$sql = "UPDATE usuaris SET cognoms='Carpio Molina' WHERE id=10";
// Executem la instrucció
if (mysqli_query($connexio, $sql)) {
    echo "Registre actualitzat correctament";
} else {
    echo "Error actualitzant registre" . mysqli_error($connexio);
}
// Tanquem la connexió abans d'acabar
mysqli_close($connexio);
?>
```

| orientat a objectes:

```
<?php
$servidor = "localhost";
$usuari = "root";
$contrasenya = "";
$basedades = "dawprova";
try { //creem connexió
    $connexio = new mysqli($servidor, $usuari, $contrasenya,$basedades);
    echo "Connectat correctament";
} catch (Exception $e) { //hi ha hagut algun error
    die("Error en connectar a la base de dades: ".$e);
}
// sql per eliminar un registre
$sql = "UPDATE usuaris SET cognoms='Carpio Molina' WHERE id=10";
```

```
// Executem la instrucció
if ($connexio->query($sql) === TRUE) {
    echo "Registre actualitzat correctament";
} else {
    echo "Error actualitzant registre" . $connexio->connect_error;
}
// Tanquem la connexió abans d'acabar
$connexio->close();
?>
```

## Limitar selecció de dades

MySQL proporciona una clàusula **LIMIT** que s'utilitza per especificar el nombre de registres per tornar.

La clàusula **LIMIT** fa que siga fàcil de codificar els resultats de diverses pàgines o la **paginació amb SQL**, i és molt útil en taules grans. La devolució d'una gran quantitat de registres pot tindre un impacte en el rendiment.

Suposem que volem seleccionar els registres de l'1 - 30 (ambdós inclosos) d'una taula anomenada "Comandes". La consulta SQL tindria el següent aspecte:

```
$sql = "SELECT * FROM comandes LIMIT 30";
```

Quan s'executa la consulta SQL anterior, ha de tornar els primers 30 registres.

Si volguérem seleccionar els registres del 16 al 25 (ambdós inclosos), podem fer-ho utilitzant l'opció MySQL: **OFFSET** (desplaçament)

La consulta SQL següent diu "retorna 10 registres, començant en el registre 16 (OFFSET 15)":

```
$sql = "SELECT * FROM comandes LIMIT 10 OFFSET 15";
```

Es pot utilitzar una sintaxi més curta per aconseguir el mateix resultat:

```
$sql = "SELECT * FROM comandes LIMIT 15, 10";
```

Cal anar amb compte, perquè **l'ordre dels números s'inverteix** quan s'utilitza una coma.

## Establir el joc de caràcters

Una altra opció interessant consisteix en **establir el joc de caràcters** que s'utilitzarà en la base de dades, per evitar caràcters erronis en la lectura posterior dels valors inserits. Es passa, com a un dels paràmetres de la funció, la cadena que representa el joc de caràcters que s'utilitzarà en la base de dades MySQL. [Ací](#) hi ha una llista dels jocs suportats en la versió 8.0 de MySQL.

En **procedimental**:

```
mysqli_set_charset($connexio, 'utf8');
```

I **orientat a objectes**:

```
$mysqli->set_charset('utf-8');
```

## Previndre caràcters especials en les cadenes de dades

Una altre aspecte que cal tenir en compte a l'hora d'introduir dades en una taula d'una base de dades és el fet que puguen haver **caràcters especials en les cadenes** (per exemple cometes simples ' o cometes dobles "). Aquests caràcters poden fer que una instrucció sql es considere mal formada i per tant la seua **execució siga errònia**.

Per evitar-ho es pot utilitzar l'opció **real\_escape\_string** (en versió orientada a objectes o procedimental). Aquesta funció s'ha d'utilitzar **una vegada s'ha creat una connexió a una base de dades MySQL**, en cas contrari, el resultat no serà el desitjat.

També és útil per evitar **injecció de SQL** en les consultes.

Per **exemple**, si obtenim un comentari des d'un formulari, abans d'inserir la informació en la base de dades (**INSERT** o **UPDATE**), és recomanable utilitzar aquesta funció per evitar possibles problemes amb caràcters especials.

**Exemple:** L'usuari introdueix el comentari (amb '): Des d'Alzira cap a l'Alcúdia

En procedimental:

```
//S'obté el valor del camp del formulari
if (isset($_POST['comentari'])) {
    $comentari = $_POST['comentari'];
}
// ...
try { // Creem la connexió
    $connexio=mysqli_connect($servidor,$usuari,$contrasenya,$basedades);
    echo "Connectat correctament";
} catch (Exception $e) { // hi ha hagut algun error
    die ("Error en connectar a la base de dades: ".$e);
}
...
//Per si hi ha caràcters especials en la cadena
$comentari = mysqli_real_escape_string($connexio,$comentari);
//Es genera la consulta
$sql = "INSERT INTO taula (comentari) VALUES ('$comentari')";
//S'executa la consulta
if (mysqli_query($connexio, $sql)) {
    //utilitzant mysqli_real_escape_string
    echo "Consulta correcta";
} else { //si no haguérem utilitzat mysqli_real_escape_string
    echo "Error en consulta";
}
//Es tanca la connexió
mysqli_close($connexio);
```

I orientat a objectes:

```
//S'obté el valor del camp del formulari
if (isset($_POST['comentari'])) {
    $comentari = $_POST['comentari'];
}
...
try { //creem connexió
    $connexio = new mysqli($servidor, $usuari, $contrasenya,$basedades);
    echo "Connectat correctament";
} catch (Exception $e) { //hi ha hagut algun error
    die("Error en connectar a la base de dades: ".$e);
}
...
//Per si hi ha caràcters especials en la cadena
$comentari = $connexio->real_escape_string($comentari);
//Es genera la consulta
$sql = "INSERT INTO taula (comentari) VALUES ('$comentari')";
//S'executa la consulta
if ($connexio->query ($sql)) { //utilitzant mysqli_real_escape_string
    echo "Consulta correcta";
} else { //si no haguérem utilitzat mysqli_real_escape_string
    echo "Error en consulta";
}
//Es tanca la connexió
$connexio->close();
```