

# UF3: Tècniques d'accés a dades

## Desenvolupament web en entorn servidor

### 3.1

#### ACCÉS A DADES AMB PHP

## INTRODUCCIÓ



JESUÏTES El Clot  
Escola del Clot



Sergi Grau

# Introducció

Sergi Grau  
sergi.grau@fje.edu

PHP OFEREIX ACCES A BASES DE DADES MITJANÇANT DIVERSOS MECANISMES:

- EXTENSIONS DE BASES DE DADES ESPECIFIQUES DEL PROVEIDOR
- EXTENSIONS DE CAPA D'ABSTRACCIO
- ORM

# Instal·lació

Si la instal·lació es fa directament des de Apache cal configurar quines extensions d'accés a bases de dades volem fer servir. Això difereix significativament segons el sistema operatiu emprat.

Normalment es troba en la secció extensions del fitxer php.ini o bé el directori on es troben `extension_dir=`. En el cas d'Ubuntu es troba en `/usr/lib/php5/`, indicant les biblioteques dinàmiques que es volen carregar.

Si la instal·lació es fa amb LAMP, WAMP, MAMP, o altres, està integrat i configurat.

Si la instal·lació es fa des de repositoris d'una distribució Linux (per exemple en Ubuntu amb `sudo apt-get install php5-mysql`), també ho configura automaticament.

El SGBD que habitualment es fa servir amb PHP és MySQL. La majoria de hostings és el que ofereixen, i les instal·lacions “stack” és el que ofereixen. Tanmateix PHP permet accedir a la majoria de SGBD.

# API

Una Interfície de Programació d'Aplicacions (o API de les seves sigles en anglès), defineix les classes, mètodes, funcions i variables que l'aplicació necessita cridar per realitzar una tasca. En el cas d'aplicacions PHP que necessitin comunicar-se amb bases de dades, les APIs necessàries normalment són exposades mitjançant extensions de PHP.

Les APIs poden ser procedimentals i orientades a objectes. Amb una API procedimental es diuen a funcions per realitzar tasques, amb una API orientada a objectes s'instancien classes i després es llamana mètodes sobre els objectes resultants. De les dues, l'última normalment és la interfície preferida, ja que és més moderna i condueix a un codi millor organitzat.

En escriure aplicacions PHP que necessiten connexió a un servidor de MySQL, hi ha diverses opcions d'APIs disponibles. Aquest document tracta sobre el que està disponible i com triar la millor solució per a l'aplicació.

# Connector

En la documentació de MySQL, el terme connector es refereix a una peça de programari que permet a les aplicacions connectar amb el servidor de base de dades MySQL. MySQL proporciona connectors per a molts llenguatges, inclòs PHP.

Si una aplicació de PHP necessita comunicar-se amb un servidor de bases de dades es necessitarà escriure codi de PHP per dur a terme activitats com connectar al servidor de la base de dades, consultar la base de dades i altres funcions relacionades amb la base de dades . Es requereix programari per proporcionar l'API que l'aplicació de PHP usarà, i també gestionar la comunicació entre l'aplicació i el servidor de la base de dades, possiblement usant altres biblioteques intermediàries quan sigui necessari. Aquest programari es coneix generalment com a connector, ja que permet a l'aplicació connectar-se al servidor de la base de dades.

# Controlador

Un controlador (o driver) és una peça de programari dissenyada per a la comunicació amb un tipus específic de servidor de base de dades. El controlador també crida a una biblioteca, com la Biblioteca Client de MySQL o el Controlador Nadiu de MySQL. Aquestes biblioteques implementen el protocol de baix nivell usat per comunicar-se amb el servidor de base de dades MySQL.

La capa d'abstracció de bases de dades disposa d'objectes de Dades de PHP (PDO) que poden usar un dels diversos controladors específics de bases de dades. Un dels controladors disponibles és el controlador MySQL de PDO, que li permet funcionar juntament amb el servidor MySQL.

De vegades les persones fan servir els termes connector i controlador els usen de manera intercanviable, i això pot ser confús. En la documentació relacionada amb MySQL, el terme "controlador" està reservat per al programari que proporciona la part específica de bases de dades d'un paquet connector.

# Extensió

En la documentació de PHP es trobarà amb un altre terme, l'extensió. El codi de PHP consisteix en un nucli, amb extensions opcionals per a la funcionalitat del nucli. Les extensions relacionades amb MySQL de PHP, com l'extensió `mysqli`, i l'extensió `mysql`, estan implementades usant el framework d'extensions de PHP.

Una extensió típicament exposa una API al programador de PHP, per poder usar les seves facilitats programàticament. No obstant això, algunes extensions que usen el framework d'extensions de PHP no exposen una API al programador de PHP.

L'extensió del controlador MySQL de PDO, per exemple, no exposa una API al programador de PHP, però proporciona una interfície per a la capa de PDO superior.

Els termes API i extensió no s'haurien de prendre com si fossin la mateixa cosa, una extensió pot no exposar necessàriament una API al programador.

# Extensions del fabricant

- CUBRID
- DB++
- dBase
- filePro
- Firebird/InterBase
- FrontBase
- IBM DB2 — IBM DB2, Cloudscape and Apache Derby
- Informix
- Ingres — Ingres DBMS, EDBC, and Enterprise Access Gateways
- MaxDB
- Mongo — MongoDB Native Driver
- mSQL
- Mssql — Microsoft SQL Server
- MySQL — MySQL Drivers and Plugins
- OCI8 — Oracle OCI8
- Ovrimos SQL
- Paradox — Paradox File Access
- PostgreSQL
- SQLite
- SQLite3
- SQLSRV — Microsoft SQL Server Driver for PHP
- Sybase
- tokyo\_tyrant



# Observacions amb respecte MySQL

En PHP 5 les biblioteques de client MySQL no s'incorporen, a causa de problemes de llicències i altres.

Hi ha també una nova extensió, MySQLi (MySQL millorat), de fet en les noves versions de PHP és la versió recomanada o PDO.

A partir de PHP 5, l'extensió SQLite ve incorporada en PHP. SQLite és un sistema de base de dades encastable. No és una biblioteca client que s'usi per connectar a un gran servidor de base de dades (com MySQL o PostgreSQL). La biblioteca SQLite llegeix i escriu directament de la base de dades en el disc.

# Introducció a l'accés al SGBD MySQL

PHP ofereix diversos controladors i complements de MySQL per accedir i manejar MySQL.

PHP ofereix tres APIs diferents per connectar-se a MySQL: `mysql`, `mysqli`, i `PDO`.

Es recomana utilitzar les extensions `mysqli` o `PDO_MySQL`.

No es recomana utilitzar l'extensió `mysql` antiga per a nous desenvolupaments, ja que s'ha declarat obsoleta a partir de PHP 5.5.0 i serà eliminada en el futur. Es proporciona una matriu detallada de comparació de característiques més baix. El rendiment global de les tres extensions es considera aproximadament el mateix. Encara que el rendiment de l'extensió aporta només una fracció del total del temps d'execució d'una consulta web de PHP. Sovint l'impacte és tan baix com 0.1%.

# Triar la biblioteca MySQL

Sergi Grau  
sergi.grau@fje.edu

Les extensions de PHP mysqli, PDO\_MySQL i mysql són embolcalls lleugeres a la part alta d'una biblioteca client en C. Les extensions poden utilitzar la biblioteca mysqlnd o la biblioteca libmysql. Escollir una biblioteca és una decisió en temps de compilació.

La biblioteca mysqlnd és part de la distribució de PHP des de la versió 5.3.0. Ofereix característiques com connexions lentes i emmagatzematge en memòria cau de consultes, característiques que no estan disponibles amb libmysql, per la qual cosa es recomana utilitzar la biblioteca mysqlnd interna

```
// compilar amb mysqlnd
$ ./configure --with-mysqli=mysqlnd --with-pdo-mysql=mysqlnd --with-
mysql=mysqlnd
```

```
// compilar amb libmysql
$ ./configure --with-mysqli=/ruta/a/mysql_config --with-pdo-
mysql=/ruta/a/mysql_config --with-mysql=/ruta/a/mysql_config
```

# Triar la biblioteca MySQL

Sergi Grau  
sergi.grau@fje.edu

	<b>CONTROLADOR Nadiu de MySQL ( <i>mysqlnd</i> )</b>	<b>Biblioteca Client Servidor de MySQL ( <i>libmysql</i> )</b>
Part de la Distribució de PHP	Sí	No
Versió de PHP On sí va introduir	5.3.0	N / A
LLICÈNCIA	LLICÈNCIA PHP 3,01	LLICÈNCIA Dual
Estat de Desenvolupament	Actiu	Actiu
Cicle de Vida	Fi no Anunciat	Fi no Anunciat
Compilació defecte en PHP 5.4 (paràgraf gust Totes les Extensions MySQL)	Sí	No
Compilació defecte en PHP 5.3 (paràgraf gust Totes les Extensions MySQL)	No	Sí
Suport paràgraf protocol sanitari de Compresió	Sí (5.3.1 +)	Sí
Suport paràgraf SSL	Sí (5.3.3 +)	Sí
Suport paràgraf Canonada amb Nom	Sí (5.3.4 +)	Sí
No Bloqueig, les consultes asíncrones	Sí	No
Estadístiques de Rendiment	Sí	No
CÀRREGA LOCAL INFILE respecte a la <u>directiva open_basedir</u>	Sí	No
EUA El Sistema de Gestió de memòria Natiu de PHP (p.ex., compleix els límits de memòria de PHP)	Sí	No
Retorna Columnes numèriques COM Valors doble (COM_QUERY)	Sí	No
Retorna Columnes numèriques COM Valors string (COM_QUERY)	Sí	Sí
API de complements	Sí	Limitada
Separació de Lectura / escriptura paràgraf Replicació MySQL	Sí, amb complement	No
Balace de Càrrega	Sí, amb complement	No
Tolerància a Fallades	Sí, amb complement	No
Connexions lentes	Sí, amb complement	No
Storage quart de memòria cau de consultes	Sí, amb complement	No
Manipulació de Consultes transparent (p.ex., auto-Explain o Monitorització)	Sí, amb complement	No

# Exemples de les 3 APIS

```
<?php
// mysqli
$mysqli = new mysqli("host.com", "usuari", "contrasenya", "basedades");
$resultat = $mysqli->query("SELECT '¡Hola, MySQL!' AS _missatge FROM DUAL");
$fila = $resultat->fetch_assoc();echo htmlentities($fila['_missatge']);

// PDO
$pdo = new PDO('mysql:host=host.com;dbname=basedades', 'usuari', 'contrasenya');

$sentencia = $pdo->query("SELECT 'Hola, MySQL!' AS _missatge FROM DUAL");

$fila = $sentencia->fetch(PDO::FETCH_ASSOC);
echo htmlentities($fila['_missatge']);

// mysql
$c = mysql_connect("host.com", "usuari", "contrasenya");

mysql_select_db("basedades");

$resultat = mysql_query("SELECT 'Hola, MySQL!' AS _missatge FROM DUAL");

$fila = mysql_fetch_assoc($resultat);
echo htmlentities($fila['_missatge']);

?>
```

# Comparació d'APIs de MySQL

Sergi Grau  
 sergi.grau@fje.edu

	ext / mysqli	Pdo_mysql	ext / mysql
Versió de PHP On sí va introduir	5.0	5,1	2,0
Inclusa amb PHP 5.x	Sí	Sí	Sí
Estat de Desenvolupament	Actiu	Actiu	Només amb Invitació Manteniment
Cicle de Vida	Actiu	Actiu	Obsolet
Recomanada paràgraf Nous Projectes	Sí	Sí	No
Interfície POO	Sí	Sí	No
Interfície procedimental	Sí	No	Sí
L'API suporta el Bloqueig no, les consultes asíncrones amb mysqlnd	Sí	No	No
Connexions persistents	Sí	Sí	Sí
L'API suporta conjunt de caràcters	Sí	Sí	Sí
L'API suporta Sentències preparades del Costat del servidor	Sí	Sí	No
L'API suporta Sentències preparades del Costat del Client	No	Sí	No
L'API suporta procediments emmagatzemats	Sí	Sí	No
API suporta Sentències Múltiples	Sí	La Majoria	No
L'API suporta Transaccions	Sí	Sí	No
Les Transaccions ES PODEN Controlar amb SQL	Sí	Sí	Sí
Suporta Tota la funcionalitat de MySQL 5.1 +	Sí	La Majoria	No

# Comprovació

## mysql

MySQL Support	enabled
Active Persistent Links	0
Active Links	0
Client API version	mysqlnd 5.0.8-dev - 20102224 - \$Id: 65fe78e70ce53d27a6cd578597722950e490b0d0 \$

Directive	Local Value	Master Value
mysql.allow_local_infile	On	On
mysql.allow_persistent	On	On
mysql.connect_timeout	60	60
mysql.default_host	no value	no value
mysql.default_password	no value	no value
mysql.default_port	3306	3306
mysql.default_socket	/var/mysql/mysql.sock	/var/mysql/mysql.sock
mysql.default_user	no value	no value
mysql.max_links	Unlimited	Unlimited
mysql.max_persistent	Unlimited	Unlimited
mysql.trace_mode	Off	Off

## mysqli

Mysqli Support	enabled
Client API library version	mysqlnd 5.0.8-dev - 20102224 - \$Id: 65fe78e70ce53d27a6cd578597722950e490b0d0 \$
Active Persistent Links	0
Inactive Persistent Links	0
Active Links	0

Directive	Local Value	Master Value
mysqli.allow_local_infile	On	On
mysqli.allow_persistent	On	On
mysqli.default_host	no value	no value
mysqli.default_port	3306	3306
mysqli.default_pw	no value	no value
mysqli.default_socket	/var/mysql/mysql.sock	/var/mysql/mysql.sock
mysqli.default_user	no value	no value
mysqli.max_links	Unlimited	Unlimited
mysqli.max_persistent	Unlimited	Unlimited
mysqli.reconnect	Off	Off

## mysqlnd

mysqlnd	enabled
Version	mysqlnd 5.0.8-dev - 20102224 - \$Id: 65fe78e70ce53d27a6cd578597722950e490b0d0 \$
Compression	supported
SSL	supported
Command buffer size	4096
Read buffer size	32768
Read timeout	31536000
Collecting statistics	Yes
Collecting memory statistics	No

## PDO

PDO support	enabled
PDO drivers	mysql, pgsql, sqlite, sqlite2

## pdo\_mysql

PDO Driver for MySQL	enabled
Client API version	mysqlnd 5.0.8-dev - 20102224 - \$Id: 65fe78e70ce53d27a6cd578597722950e490b0d0 \$

Directive	Local Value	Master Value
pdo_mysql.default_socket	/var/mysql/mysql.sock	/var/mysql/mysql.sock

## pdo\_pgsql

PDO Driver for PostgreSQL	enabled
PostgreSQL(libpq) Version	9.0.5
Module version	1.0.2
Revision	\$Id\$

## pdo\_sqlite

PDO Driver for SQLite 3.x	enabled
SQLite Library	3.7.7.1

# Consultes emmagatzemades i no emmagatzemades en buffer

Les consultes utilitzen un mode d'emmagatzematge en memòria intermèdia per omissió. Això vol dir que els resultats de les consultes són transferits immediatament del servidor MySQL a PHP on després es mantenen en la memòria del procés de PHP. Això permet operacions addicionals com el compte del nombre de files i moure (buscar) el punter actual del resultat. També permet emetre més consultes a la mateixa connexió mentre es treballa amb el conjunt de resultats. El costat negatiu de la manera d'emmagatzematge en buffer és que els conjunts de resultats grans requereixen bastant més memòria. La memòria es mantindrà ocupada fins que totes les referències al conjunt de resultats siguin desestablertes o s'alliberi explícitament el conjunt de resultats, el que automàticament passarà durant la finalització de l'última petició. La terminologia "resultat emmagatzemat" també s'usa per la manera d'emmagatzematge en memòria intermèdia, ja que el conjunt de resultats complet s'emmagatzema d'una vegada.



# Consultes emmagatzemades i no emmagatzemades en buffer

Quan s'usa libmysql com biblioteca, el límit de memòria de PHP no comptarà la memòria usada pels conjunts de resultats a menys que les dades siguin posades en variables de PHP. Amb mysqlnd, la memòria tinguda en compte inclourà el conjunt de resultats complet.

Les consultes de MySQL no emmagatzemades en memòria intermèdia executen la consulta i després tornen un resource mentre les dades encara segueixen esperat al servidor MySQL que siguin extretes. Això utilitza menys memòria per part de PHP, però pot incrementar la càrrega del servidor. Llevat que s'obtingui el conjunt de resultats complet del servidor no es podran enviar més consultes a la mateixa connexió. Les consultes no emmagatzemades en buffer també poden ser descrites com "resultats d'ús".

# Consultes emmagatzemades i no emmagatzemades en buffer

Seguint aquestes característiques, les consultes emmagatzemades en memòria intermèdia haurien usar-se en casos on s'espera només un conjunt de resultats limitat o es necessiti conèixer la quantitat de files retornades abans de llegir totes les files. La manera de no emmagatzematge en memòria intermèdia hauria d'usar quan s'esperen grans resultats.

Ja que les consultes emmagatzemades en memòria intermèdia són les predeterminades, els exemples demostren com executar consultes no emmagatzemades en buffer amb una API.

Exemple de consulta no emmagatzemada en buffer: mysql

```
<?php
$conn = mysql_connect("localhost", "usuari", "contrasenya");
$db = mysql_select_db("world");

$res_sense_buffer = mysql_unbuffered_query("SELECT nom FROM ciutat");
if ($res_sense_buffer) {
    while ($fila = mysql_fetch_assoc($res_sense_buffer)) {
        echo $fila['nom'] . PHP_EOL;
    }
}
?>
```

# Conjunts de caràcters

De manera ideal, un conjunt de caràcters apropiat serà establert a nivell del servidor, que està descrit dins de la secció "Configuració del conjunt de caràcters del manual del servidor MySQL. De manera alternativa, una API de MySQL ofereix mètodes per establir el conjunt de caràcters en temps d'execució.

El conjunt de caràcters s'hauria d'entendre i ser definit, ja que afecta cada acció, i inclou implicacions de seguretat. Per exemple, el mecanisme d'escapament (p.ex., `mysqli_real_escape_string ()` per `mysqli`, `mysql_real_escape_string ()` per `mysql`, i `PDO :: quote ()` per `PDO_MySQL`) complirà amb aquesta configuració. És important adonar-se que aquestes funcions no usaran el conjunt de caràcters que està definit en una consulta, de manera que, per exemple, el següent codi no tindrà efecte sobre elles:

```
<?php
$conn = mysql_connect("localhost", "usuari", "contrasenya");
$db    = mysql_select_db("world");

if (!mysql_set_charset('utf8', $conn)) {
    echo "Error: no s'ha pogut establir el conjunt de caracters.\n";
    exit;
}
echo 'El conjunt de caracters actual es: ' . mysql_client_encoding($conn);
?>
```