

# L'accés a la base de dades del WordPress de *Mosaic* des del *plug-in* VisDa

Carlos Casado

PID\_00201052



*Els textos i imatges publicats en aquesta obra estan subjectes –llevat que s'indiqui el contrari– a una llicència de Reconeixement-NoComercial-SenseObraDerivada (BY-NC-ND) v.3.0 Espanya de Creative Commons. Podeu copiar-los, distribuir-los i transmetre'ls públicament sempre que en citeu l'autor i la font (FUOC. Fundació per a la Universitat Oberta de Catalunya), no en feu un ús comercial i no en feu obra derivada. La llicència completa es pot consultar a <http://creativecommons.org/licenses/by-nc-nd/3.0/es/legalcode.ca>*

# Índex

<b>1. Introducció</b> .....	5
<b>2. Estructura de la base de dades del WordPress</b> .....	7
<b>3. Les taules una a una</b> .....	9
3.1. wp_posts .....	9
3.2. wp_postmeta .....	11
3.3. wp_comments .....	11
3.4. wp_commentmeta .....	13
3.5. wp_links .....	13
3.6. wp_users .....	14
3.7. wp_usermeta .....	15
3.8. wp_terms .....	15
3.9. wp_term_taxonomy .....	16
3.10. wp_term_relationships .....	17
3.11. wp_options .....	18
<b>4. Relacions entre taules</b> .....	19
4.1. Les taules meta .....	19
4.2. Comentaris i entrades .....	19
4.3. wp_term_taxonomy i wp_terms .....	20
4.4. Entrades, enllaços, categories i etiquetes .....	21
<b>5. Consultar la base de dades</b> .....	24
5.1. Consultes senzilles .....	24
5.2. Accedir a les dades de dues taules .....	26
5.3. Agrupant dades i comptant .....	27
5.4. Categories i etiquetes .....	28



## 1. Introducció

Com hem vist en el capítol anterior, el WordPress és un gestor de continguts especialitzat en la gestió de blogs que s'usa molt per la seva potència, flexibilitat i facilitat d'ús i perquè és GPL. Com la majoria de gestors de continguts actuals, el WordPress es pot ampliar fàcilment afegint *plug-ins* que **qualsevol** pot programar i distribuir recolzant-se en la documentació que els desenvolupadors del programa posen a la disposició de la comunitat.

Aquest material vol ser una ajuda per a les persones que volen fer un *plug-in* per al WordPress i en particular per als qui vulguin visualitzar la informació d'un blog elaborat amb aquest gestor de continguts.

Si volem fer un *plug-in* per al WordPress és possible que vulguem recuperar alguna informació de la base de dades. Si volem fer una llista amb les etiquetes més usades, els usuaris que més escriuen, els *posts* amb més comentaris, haurèm d'accedir a la base de dades per a poder recuperar aquesta informació.

En general, si volem recuperar qualsevol contingut generat per l'usuari i emmagatzemat en el WordPress, haurèm d'accedir a la base de dades i recuperar d'allà la informació. En la base de dades es guarda informació sobre les entrades, usuaris, etiquetes, comentaris, etc. Conèixer la base de dades, com està estructurada –taules, camps, relacions–, quina informació hi podem trobar i de quina manera extreure les dades és imprescindible si volem crear un *plug-in* que usi la informació generada per l'usuari. Aquest apartat vol explicar com es pot extreure informació de la base de dades.

Abans d'entrar en l'estructura de la base de dades del WordPress, cal tenir present que el WordPress usa MySQL i està programat en PHP. Se suposa que el lector té un cert coneixement del MySQL i l'accés a les dades des de PHP.

Per a explicar algunes característiques de la base de dades del WordPress usarem dades de la revista *Mosaic*. *Mosaic* és una revista electrònica que en el moment de redactar aquest material té una mica més de deu anys de vida i noranta-quatre números.

Aquest material es divideix bàsicament en dues parts: la primera és un repàs exhaustiu de la base de dades del WordPress, i la segona explica com es poden obtenir les dades que ens interessin des d'un *plug-in*.

Aquest document està actualitzat a la versió 3.4 del WordPress. Encara que la base de dades del WordPress és bastant estable i no se sol modificar amb els canvis de versió, és convenient tenir en compte que en versions futures la base de dades podria no ser exactament igual que la que es tracta en aquest material.

### **Alguns detalls de nomenclatura**

El WordPress, com a gestor de continguts especialitzat en blogs, basa tota l'estructura de la seva base de dades en el contingut principal, els *posts*, encara que també té un altre tipus de contingut, que anomena *pàgines*. La diferència principal és que les pàgines estan pensades per a contingut que sempre ha d'estar present i que acostuma a ser accessible des d'un menú en la pàgina principal del blog. Per contra, els *posts* apareixen al principi de la primera pàgina quan fa poc que estan escrits i es van enfonsant en l'estructura del blog a mesura que esdevenen més antics i altres més moderns en van ocupant la posició.

En català la paraula *post* se sol traduir per *entrada* o per *article* encara que continua essent bastant comú l'ús de la paraula anglesa. En aquest text s'usen indiferentment, *post* i *entrada*.

Quant a les pàgines, convé diferenciar una pàgina del WordPress d'una pàgina web. En aquest document totes les vegades que s'usi la paraula *pàgina* serà amb referència a les pàgines del WordPress, tret que s'especifiqui *web* a continuació.

Finalment, com el WordPress, usarem la paraula *terme* per a referir-nos de manera genèrica a categories i etiquetes.

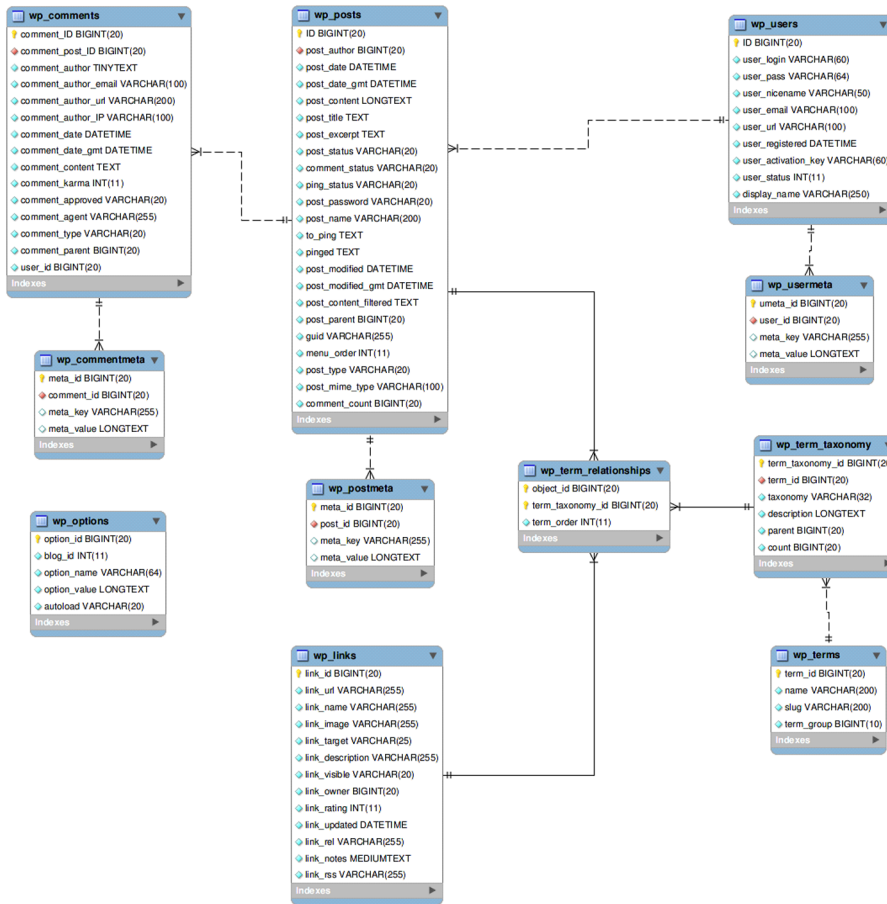
## 2. Estructura de la base de dades del WordPress

La base de dades del WordPress es fonamenta en aquestes onze taules:

Taules WordPress

Nom de la taula	Descripció
wp_post	És la taula més important de la base de dades del WordPress. Guarda informació sobre les entrades, les pàgines, els fitxers adjunts i els menús. Per a cada versió d'una entrada se'n guarda el contingut, el títol, l'autor, la data, el resum, l'estatus, etc. Hi ha un camp (post_type) que indica de quin tipus és.
wp_postmeta	Metadades de les entrades. Entre altres coses guarda informació sobre els adjunts associats.
wp_comments	En aquesta taula es guarden els comentaris amb la informació associada: autor, data, contingut, si està o no aprovat, etc.
wp_commentsmeta	Metadades dels comentaris. Tan sols és usada per alguns <i>plugins</i> com Akismet.
wp_links	Llista d'enllaços.
wp_term_relationships	Per mitjà d'aquesta taula es relacionen les entrades i els enllaços amb les seves etiquetes i categories.
wp_term_taxonomy	Taula d'etiquetes i categories. El WordPress guarda juntes etiquetes i categories tant dels enllaços com dels <i>posts</i> (encara que no són compartides).
wp_terms	Noms de les etiquetes i les categories.
wp_users	Informació dels usuaris. En aquesta taula es guarden el nom d'usuari ( <i>nickname</i> ), el correu electrònic, l'adreça web que aquest especifiqui, la data de registre, la clau d'activació (que s'envia per a comprovar l'adreça de correu electrònic dels nous usuaris) i el nom que s'ha de mostrar.
wp_usersmeta	Opcions dels usuaris. Aquí es guarda la resta d'informació de l'usuari, com el nom real, el tipus d'editor que usa, el nivell, etc.
wp_options	Taula d'opcions del WordPress.

Figura 1. Diagrama de la base de dades del WordPress



Disponible a [http://codex.wordpress.org/Database\\_Description](http://codex.wordpress.org/Database_Description)

En la figura 1 podem veure totes les taules de la base de dades del WordPress i les relacions entre elles.

Com és normal, la base de dades sembla centrada en la taula d'entrades. Tan sols la taula d'opcions no manté cap relació amb wp\_posts.

Un detall important és que malgrat que en la imatge el nom de totes les taules comença per wp\_, això no sempre és així. El WordPress permet canviar el prefix de les taules per a facilitar la convivència de diferents instal·lacions del gestor de continguts en una mateixa base de dades. Hi ha una classe global, anomenada \$wpdb, que ens facilita l'accés a les taules independentment del prefix que s'hagi posat al nom.



## 3. Les taules una a una

### 3.1. wp\_posts

La taula wp\_posts és la que té més camps i la de més pes en la majoria d'instal·lacions del WordPress.

A *Mosaic*, en el moment d'escriure aquest text, wp\_posts tenia 7.924 files que ocupaven un total de 60 Mb. Per fer-nos-en una idea, la taula següent en pes, és la wp\_term\_relationships amb 436 kb i 7.455 files. A wp\_posts es guarden les entrades (*posts*), les pàgines, totes les revisions que es fan de tots dos tipus de continguts i la informació relacionada amb tots els arxius que es pugen.

Els camps que conté aquesta taula són:

Camps taula wp\_posts

Camp	Tipus	Descripció
ID	BIGINT(20)	Identificador.
post_author	BIGINT(20)	Autor. Relaciona la taula wp_posts amb la taula wp_users.
post_date	DATETIME	Data i hora de creació de l'entrada/pàgina en l'horari local.
post_date_gmt	DATETIME	Data i hora de creació de l'entrada /pàgina en horari GMT.
post_content	LONGTEXT	El contingut de l'entrada o pàgina.
post_title	TEXT	El títol de l'entrada o pàgina.
post_excerpt	TEXT	Extracte.
post_status	VARCHAR(20)	Situació del <i>post</i> : publicat, pendent, privat, etc.
comment_status	VARCHAR(20)	Estat dels comentaris: <i>open/closed</i> (s'accepten o no).
ping_status	VARCHAR(20)	Situació dels PING: <i>open/closed</i> .
post_password	VARCHAR(20)	Si s'assigna una contrasenya a una entrada, aquest només es pot llegir si s'introdueix aquesta contrasenya. Si el seu estat és <i>publish</i> apareixerà el títol però sol·licitarà la contrasenya o <i>password</i> per a poder-lo llegir. La contrasenya es desa en la base de dades sense encriptar.
post_name	VARCHAR(200)	Nom de l'entrada. Si està publicat coincideix amb el títol. Si és una revisió o un autodesat rep com a nom l'ID de l'entrada, la paraula <i>revisionautosave</i> i el número de revisió (si n'hi ha més d'una).
to_ping	TEXT	Si des d'aquesta entrada es fa un PING a una altra pàgina, queda reflectit aquí.

#### Opció de mantenir revisions

Si es vol es pot desactivar l'opció de mantenir les revisions afegint la línia següent a l'arxiu wp-config.php:  

```
define('WP_POST_REVISIONS', false);
```

Camp	Tipus	Descripció
pinged	TEXT	Si aquesta entrada ha rebut un PING des d'una altra pàgina, queda reflectit aquí.
post_modified	DATETIME	Data de l'última modificació de l'entrada o pàgina en horari local.
post_modified_gmt	DATETIME	Ídem que l'anterior però en format GMT.
post_content_filtered	TEXT	No usat directament pel WordPress, usat per alguns <i>plug-ins</i> que modifiquen el contingut del <i>post</i> , per a no haver de fer la modificació cada vegada que es mostra l'entrada.
post_parent	BIGINT(20)	En cas que sigui una revisió es refereix a l'ID del <i>post</i> revisat. En cas que sigui un adjunt serà l'ID del <i>post</i> en què es va pujar. En cas que sigui una pàgina o un ítem de menú, la pàgina o l'ítem de menú pare.
guid	VARCHAR(255)	L'adreça del <i>post</i> o de l'arxiu (si es tracta d'un adjunt).
menu_order	INT(11)	S'usa en els fitxers adjunts per a mantenir un ordre entre ells. També s'usa per a fer la llista de les pàgines en un ordre diferent de l'alfabètic, que és el que usa el WordPress per defecte.
post_type	VARCHAR(20)	Classe de contingut emmagatzemat (attachment, page, post, revision, nav_menu_item).
post_mime_type	VARCHAR(100)	Tipus de contingut quan es tracta d'un arxiu adjunt (jpg, gif, pdf, etc.).
comment_count	BIGINT(20)	Nombre de comentaris.

El camp **post\_status** pot tenir els valors següents:

- **publish**. L'entrada o pàgina està publicada i és visible per a tots els visitants.
- **inherit**. Una revisió.
- **pending**. Pendent de publicació.
- **private**. Privat, solament accessible per als usuaris registrats.
- **future**. Amb data de publicació futura.
- **draft**. Esborrany, no desat encara.
- **trash**. Esborrat.

En el camp **post\_type** es desa quina classe de contingut s'emmagatzema en aquest registre. Les diferents classes poden ser:

- **attachment**. Indica que en aquest registre es guarda informació d'un arxiu pujat. L'enllaç a l'arxiu es troba en el camp **guid**.
- **page**. Pàgina.
- **post**. Entrada.
- **revision**. Revisió d'una entrada.
- **nav\_menu\_item**. Ítem de menú.

Un *plug-in* podria afegir noves classes de contingut.

## Índexs

Aquesta taula té els índexs següents:

- post\_name (primari)
- post\_type+post\_status+post\_date+ID
- post\_parent
- post\_author

### 3.2. wp\_postmeta

En la taula wp\_postmeta es guarda informació referent als registres guardats a wp\_post.

Per exemple, es guarden tots els camps personalitzats de les entrades, també s'usa per a "bloquejar" una entrada o pàgina i evitar així que dos usuaris la puguin modificar alhora. Com que les pàgines poden usar diferents plantilles, també es guarda en aquesta taula la plantilla que es fa servir per a una determinada pàgina.

Camps taula wp\_postmeta

Camp	Tipus	Descripció
meta_id	BIGINT(20)	Identificador del registre.
post_id	BIGINT(20)	Identificador del post o entrada a què va referit.
meta_key	VARCHAR(255)	El nom del camp personalitzat.
meta_value	LONGTEXT	El valor del camp personalitzat.

## Índexs

Aquesta taula té els índexs següents:

- meta\_id
- post\_id
- meta\_key

### 3.3. wp\_comments

En la taula wp\_comments es guarden els comentaris tant dels apunts com de les pàgines.

## Camps taula wp\_postmeta

Camp	Tipus	Descripció
comment_id	BIGINT(20)	Identificador del comentari.
comment_post_id	BIGINT(20)	Identificador de l'entrada o pàgina relacionada.
comment_author	TINYTEXT	Nom de l'autor. Se li demana abans d'escriure el comentari. És una dada obligatòria.
comment_author_email	VARCHAR(100)	Correu de l'autor. També se l'obliga a posar-lo abans de publicar el comentari.
comment_author_url	VARCHAR(200)	Adreça de la pàgina web del comentarista. No obligatori.
comment_author_ip	VARCHAR(100)	Adreça IP de la persona que escriu el comentari.
comment_date	DATETIME	Data i hora en horari local.
comment_date_gmt	DATETIME	Data i hora en format GMT.
comment_content	TEXT	El contingut del comentari.
comment_karma	INT(11)	Aquest camp no és usat directament pel WordPress (almenys en aparença), però sí, per alguns <i>plug-ins</i> .
comment_approved	VARCHAR(20)	Especifica si el comentari està aprovat (1), no (0) o està marcat com a <i>spam</i> (spam).
comment_agent	VARCHAR(255)	En cas que el comentari sigui un <i>trackback</i> o un <i>pingback</i> guarda l'adreça des de la qual s'ha fet.
comment_type	VARCHAR(20)	Determina si el comentari és un <i>trackback</i> , un <i>pingback</i> , o un <i>comment</i> .
comment_parent	BIGINT(20)	En cas que el tema ofereixi la possibilitat de niar comentaris, guarda l'id del comentari pare.
user_id	BIGINT(20)	En cas que l'usuari que fa el comentari sigui un usuari registrat, guarda el seu id.

## Índexs

Aquesta taula té els índexs següents:

- comment\_ID
- comment\_approved
- comment\_post\_ID
- comment\_approved + comment\_date\_gmt
- comment\_date\_gmt
- comment\_parent

### 3.4. wp\_commentmeta

La taula wp\_commentmeta permet afegir metadades als comentaris del WordPress.

L'aplicació no la usa directament, sinó per mitjà del *plug-in* Akismet, que el WordPress inclou per defecte.

Camps taula wp\_commentmeta

Camp	Tipus	Descripció
meta_id	BIGINT(20)	Identificador del registre.
comment_id	BIGINT(20)	Identificador del comentari al qual es refereix.
meta_key	VARCHAR(255)	El nom del camp personalitzat.
meta_value	LONGTEXT	El valor del camp personalitzat.

### Índexs

Aquesta taula té els índexs següents:

- meta\_ID
- comment\_id
- meta\_key

### 3.5. wp\_links

En la taula wp\_links es guarda informació sobre els enllaços emmagatzemats en el WordPress.

El WordPress ofereix una eina de gestió d'enllaços bastant potent.

Camps taula wp\_links

Camp	Tipus	Descripció
link_id	BIGINT(20)	Identificador del registre.
link_url	VARCHAR(255)	Adreça de l'enllaç.
link_name	VARCHAR(255)	Nom de la pàgina enllaçada.
link_image	VARCHAR(255)	Imatge representativa de la pàgina enllaçada.
link_target	VARCHAR(25)	Tres opcions: _top, _blank o _none.
link_description	VARCHAR(255)	Descripció de la pàgina enllaçada.

Camp	Tipus	Descripció
link_visible	VARCHAR(20)	Si és visible o no en el blog.
link_owner	BIGINT(20)	Identificador de l'usuari que ha afegit l'enllaç.
link_rating	INT(11)	Puntuació (d'1 a 10) de l'enllaç.
link_updated	DATETIME	El WordPress no usa actualment aquest camp, però hi ha <i>plug-ins</i> que hi posen la data en què s'ha modificat l'enllaç.
link_rel	VARCHAR(255)	Relació de coneixement de l'usuari que posa l'enllaç amb el propietari de la pàgina enllaçada.
link_notes	MEDIUMTEXT	Comentaris sobre l'enllaç.
link_rss	VARCHAR(255)	Adreça RSS de la pàgina enllaçada.

## Índexs

Aquesta taula té els índexs següents:

- link\_ID
- link\_category
- link\_visible

### 3.6. wp\_users

En la taula wp\_users es guarda informació sobre els usuaris registrats.

Camps taula wp\_users

Camp	Tipus	Descripció
ID	BIGINT(20)	Identificador del registre.
user_login	VARCHAR(60)	Nom identificador de l'usuari. És el nom que l'usuari usa per a accedir a l'escriptori del WordPress. Pot tenir majúscules, minúscules i espais, però no accents.
user_pass	VARCHAR(64)	Contrasenya codificada en MD5.
user_nicename	VARCHAR(50)	El <i>login</i> convertit en minúscules i amb els espais substituïts per guions.
user_email	VARCHAR(100)	Adreça de correu electrònic de l'usuari.
user_url	VARCHAR(100)	Adreça de la pàgina web de l'usuari.
user_registered	DATETIME	Data en què l'usuari es va registrar.
user_activation_key	VARCHAR(60)	Codi de comprovació. S'envia a l'usuari si ha perdut la contrasenya per a permetre crear-ne una de nova.

Camp	Tipus	Descripció
user_status	INT(11)	Actualment no usat pel WordPress.
display_name	VARCHAR(250)	Nom que es mostra en les entrades escrites per l'usuari.

## Índexs

Aquesta taula té els índexs següents:

- ID
- user\_login
- user\_nicename

### 3.7. wp\_usermeta

En la taula wp\_usermeta es guarda informació de l'usuari que no s'emmagatzema en la taula wp\_users.

Entre la informació que es guarda aquí, hi ha el nom real de l'usuari, el nivell d'accés, l'àlies (*nickname*), el tipus d'editor que usa, etc.

Camps taula wp\_usermeta

Camp	Tipus	Descripció
umeta_id	BIGINT(20)	Identificador del registre.
user_id	BIGINT(20)	Identificador de l'usuari al qual va referit.
meta_key	VARCHAR(255)	El nom del camp personalitzat.
meta_value	LONGTEXT	El valor del camp personalitzat.

## Índexs

Aquesta taula té els índexs següents:

- umeta\_id
- user\_id
- meta\_key

### 3.8. wp\_terms

En la taula wp\_terms es guarden totes les categories, etiquetes i formats creats.

És important tenir en compte que si una categoria i una etiqueta tenen el mateix nom no estaran duplicades en aquesta taula, sinó que el nom només apareixerà una vegada i serà la informació que hi ha en la taula `wp_term_taxonomy` la que ens servirà per a diferenciar-ne l'un de l'altre.

### Formats predeterminats

Des de la versió 3.1 del WordPress, aquí també s'inclouen els "formats predeterminats" per a les entrades. Aquests formats permeten tenir diferents plantilles per a diferents tipus d'entrades. Aquests formats estan predeterminats i no se n'haurien d'afegir de nous ni per temes ni per *plug-ins*.

Camps taula `wp_terms`

Camp	Tipus	Descripció
<code>term_id</code>	BIGINT(20)	Identificador del registre
<code>name</code>	VARCHAR(200)	Nom
<code>slug</code>	VARCHAR(200)	Nom només amb lletres, dígit i guions. Usat per a facilitar-ne l'ús en una adreça HTTP. Aquest és el nom que s'usa quan fem una llista de totes les entrades d'una determinada categoria o que contenen una determinada etiqueta.
<code>term_group</code>	BIGINT(10)	El valor del camp personalitzat.

### Índexs

Aquesta taula té els índexs següents:

- `term_ID`
- `slug`
- `name`

### 3.9. `wp_term_taxonomy`

En la taula `wp_term_taxonomy` es guarda informació sobre el terme (si és una categoria, una etiqueta, una categoria d'un enllaç o un format) guardat en la taula `wp_terms`. També es guarda informació sobre el nombre de vegades que s'ha usat aquesta categoria o etiqueta i la relació pare-fill si hi és.

Camps taula `wp_term_taxonomy`

Camp	Tipus	Descripció
<code>term_taxonomy_id</code>	BIGINT(20)	Identificador del registre.
<code>term_id</code>	BIGINT(20)	Identificador de la taula <code>wp_terms</code> .



Camp	Tipus	Descripció
taxonomy	VARCHAR(32)	Pot tenir quatre possibles valors: category per a les categories, link_category per a les categories dels enllaços, post_tag per a les etiquetes i post_format per als formats pre-determinats. Actualment, el WordPress permet crear taxonomies, de manera que a aquests valors es podrien afegir els que fossin necessaris per a les noves taxonomies.
description	LONGTEXT	Descripció.
parent	BIGINT(20)	En el cas de les categories hi pot haver una relació de pare-fill. En aquest cas aquesta relació queda especificada aquí.
count	BIGINT(20)	Nombre d'entrades/enllaços que tenen aquesta categoria, etiqueta o format. Hi pot haver registres que tinguin en aquest camp un valor zero si s'han creat però no s'estan usant.

## Índexs

Aquesta taula té els índexs següents:

- term\_taxonomy\_id
- term\_id + taxonomy
- taxonomy

### 3.10. wp\_term\_relationships

La taula wp\_term\_relationships permet relacionar els termes amb les entrades i/o enllaços que les usen.

Camps taula wp\_term\_relationships

Camp	Tipus	Descripció
object_id	BIGINT(20)	Identificador de l'entrada o enllaç. Atès que es pot referir a qualsevol de les dues taules, ens podem trobar amb valors repetits. Així, un 2 es pot referir tant a un enllaç com a una entrada. Per diferenciar-los haurem d'anar a la taula wp_term_taxonomy
term_taxonomy_id	BIGINT(20)	Identificador de la taula wp_term_taxonomy a partir del qual sabrem si és una categoria, una categoria d'enllaços o una etiqueta, i també el seu nom.
term_order	INT(11)	Sense usar actualment.

## Índexs

Aquesta taula té els índexs següents:

- `object_id + term_taxonomy_id`
- `term_taxonomy_id`

### 3.11. `wp_options`

En la taula `wp_options` es guarden les diferents opcions del WordPress. També s'hi guarden les opcions creades pels *plug-ins* instal·lats.

El WordPress ofereix funcions per a afegir, consultar, modificar i esborrar files d'aquesta taula, per la qual cosa no serà necessari accedir-hi directament. No obstant això, a tall informatiu, s'expliquen els camps que la formen.

Camps taula `wp_options`

Camp	Tipus	Descripció
<code>option_id</code>	BIGINT(20)	Identificador del registre.
<code>blog_id</code>	INT(11)	Identificador del blog (en multiblog).
<code>option_name</code>	VARCHAR(64)	Nom de l'opció.
<code>option_value</code>	LONGTEXT	Valor de l'opció.
<code>autoload</code>	VARCHAR(20)	Quan el WordPress arrenca, carrega totes les opcions que estiguin marcades com a <i>yes</i> en aquest camp. <i>Yes</i> és, a més, el valor per defecte.

### Índexs

Aquesta taula té els índexs següents:

- `option_id + blog_id + option_name`
- `option_name`

## 4. Relacions entre taules

Sovint, quan volem obtenir informació de la base de dades del WordPress, les dades que volem obtenir estan repartides entre dues o més taules relacionades entre elles. Ens interessa saber quines taules estan relacionades i quina és aquesta relació per a saber com recuperar les dades.

### 4.1. Les taules meta

Les taules `wp_comments`, `wp_posts` i `wp_users` tenen una taula relacionada anomenada `wp_commentmeta`, `wp_postmeta` i `wp_usermeta`, respectivament, en què es guarda informació opcional.

Taules "meta"

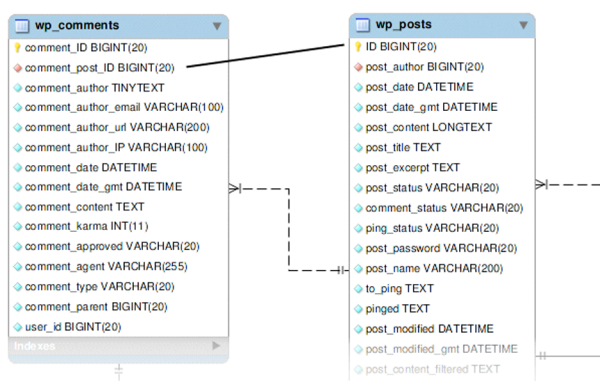
	Taula	Taula d'opcions
Nom Camp de relació	<code>wp_posts</code> ID	<code>wp_postmeta</code> <code>meta_id</code>
Nom Camp de relació	<code>wp_comments</code> <code>comment_ID</code>	<code>wp_commentmeta</code> <code>meta_id</code>
Nom Camp de relació	<code>wp_users</code> ID	<code>wp_usermeta</code> <code>umeta_id</code>

Habitualment ens interessarà recuperar els camps opcionals d'una determinada entrada, d'un comentari o d'un usuari. Si en sabem l'ID podrem accedir fàcilment a les dades opcionals.

### 4.2. Comentaris i entrades

La taula de comentaris té una relació *N* a 1 amb la taula de *posts*. La relació s'estableix per mitjà del camp `comment_post_ID` de la taula `wp_comments` amb el camp `ID` de la taula `wp_posts`.

Figura 2. Les taules `wp_comments` i `wp_posts`



#### Vegeu també

En l'apartat "Estructura de la base de dades del WordPress" trobareu la figura 1, en què es poden veure totes les relacions entre taules que explicarem en aquest apartat.

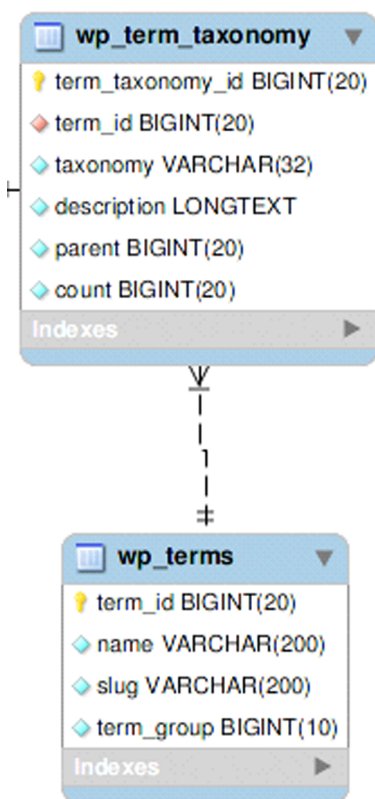
#### Vegeu també

En l'apartat "Consultes senzilles" d'aquest material es poden trobar alguns exemples sobre com recuperar les dades opcionals d'un determinat apunt.

### 4.3. wp\_term\_taxonomy i wp\_terms

Tal com s'ha explicat més amunt, la taula wp\_term\_taxonomy guarda informació sobre el terme (categoria, etiqueta, categoria d'enllaç), mentre que la taula wp\_terms guarda el nom de l'element. Aquestes dues taules tenen poca utilitat per separat, així que serà habitual consultar-les conjuntament.

Figura 3. Etiquetes i categories



La relació d'aquestes dues taules és de N a 1, atès que un terme pot ser dues coses alhora: podem tenir una categoria anomenada **fotos** i també una etiqueta amb el mateix nom. Seguint amb aquest exemple, en la taula wp\_terms tindriem un registre en què en el camp *name* tindriem la paraula **fotos** (i en el camp *term\_id* un valor **n**) i en la taula wp\_term\_taxonomy tindriem dos registres, amb un mateix valor (**n**) en el camp *term\_id* i dos valors diferents (un en cada registre) en el camp *taxonomy*:

Taula wp\_terms

wp_terms	
term_id	5
name	fotos
slug	fotos
term_grup	0

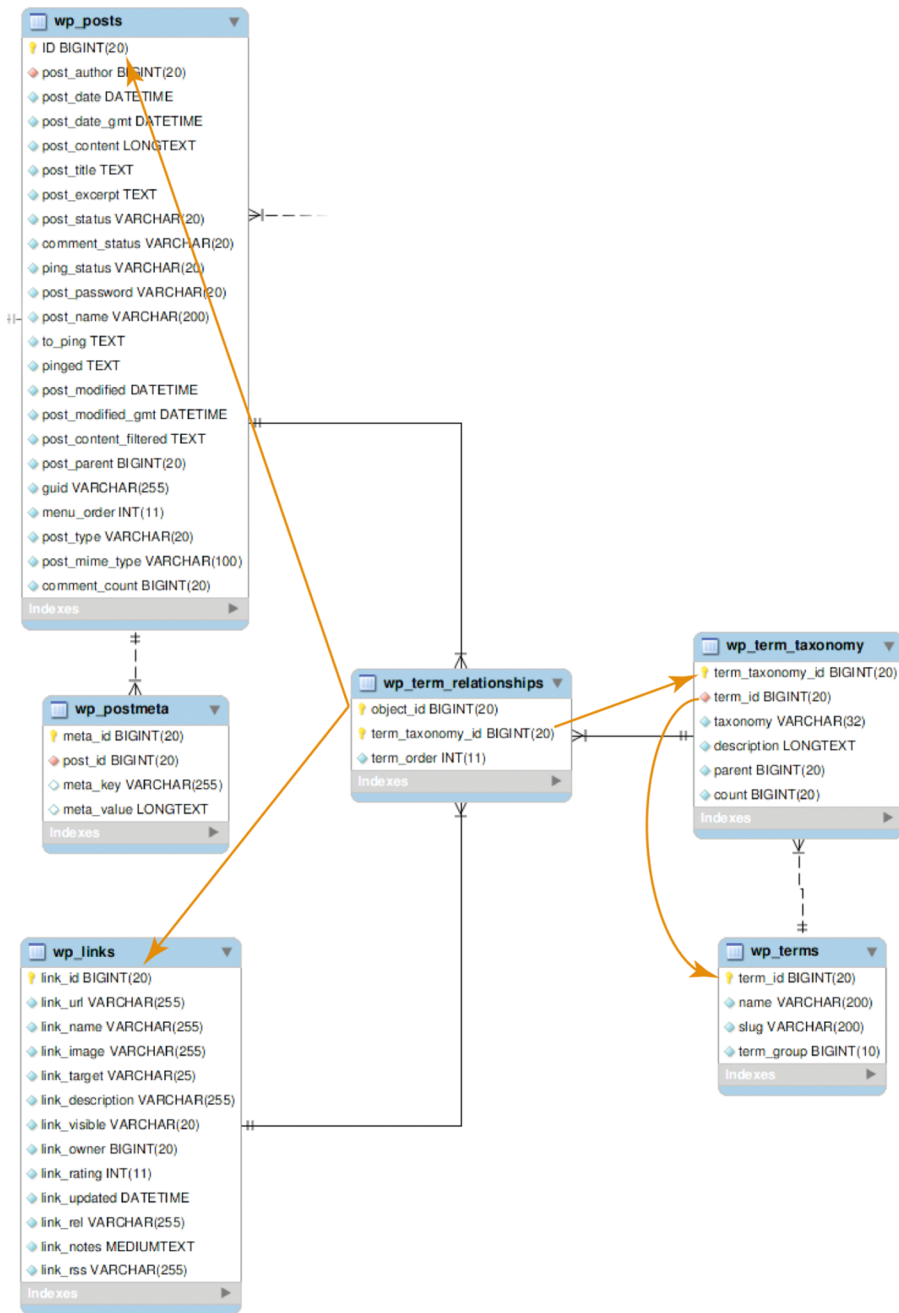
Taula wp\_term\_taxonomy

wp_term_taxonomy		
	Registre 1	Registre 2
term_taxonomy_id	102	253
term_id	5	5
taxonomy	post_tag	category

#### 4.4. Entrades, enllaços, categories i etiquetes

El quadre de relacions següent (figura 4) és el més complex, però també el més útil a l'hora de fer un *plug-in* de visualització de dades del WordPress. Aquestes relacions ens permeten conèixer les entrades d'una determinada categoria o que tenen una determinada etiqueta. Encara que la taula d'enllaços segurament no seria útil a l'hora de visualitzar informació, està inclosa aquí perquè forma part del conjunt.

Figura 4. Les relacions entre les entrades i els enllaços, i les categories i les etiquetes



La relació entre les entrades i les categories i/o les etiquetes s'estableix per mitjà de la taula wp\_term\_relationships, en què es guarden parells de valors compostos per l'identificador de l'entrada i l'identificador de la categoria o etiqueta (representat per l'identificador de la taula wp\_term\_taxonomy).

La "dificultat" d'aquest conjunt d'interrelacions ve de la necessitat, si es volen obtenir les entrades d'una determinada categoria per exemple, d'obtenir totes les entrades relacionades des de la taula wp\_terms, en què hi haurà el nom de la categoria.

El camí serà:

- Buscar la categoria pel seu nom en la taula `wp_terms` i així obtenir l'identificador del terme (camp `term_id`).
- Amb l'identificador de la categoria, buscar l'identificador de la taula `wp_term_taxonomy` (`term_taxonomy_id`). Com que podem tenir un mateix nom associat a la categoria i l'etiqueta (per exemple), dels registres que obtinguem, ens quedarem el que en el camp `taxonomy` tingui `category`. Amb això obtenim l'identificador `term_taxonomy_id`.
- En la taula `wp_term_relationships`, buscar totes les aparicions de l'identificador de `wp_term_taxonomy` (`term_taxonomy_id`). Obtindrem un conjunt de registres corresponent amb totes les entrades d'aquesta categoria.
- Obtenir les dades que volem de les entrades a partir dels registres oposats.

A partir d'aquests senzills passos, es pot crear una consulta SQL que obtingui els títols de totes les entrades d'una determinada categoria (o qualsevol altra dada que ens interessi: data de publicació, contingut, etc.). En l'apartat següent veurem alguns exemples senzills de com es pot fer.

## 5. Consultar la base de dades

Els propers apartats expliquen com es pot accedir a la base de dades per a obtenir la informació que ens interessa. Estan plantejats de manera que siguin fàcils de seguir sense necessitat de tenir coneixements d'SQL, el llenguatge que usarem per a accedir a les dades.

Evidentment, uns bons coneixements d'SQL poden fer innecessari llegir aquesta part del material. No obstant això, si voleu fer un *plug-in* per al WordPress que inclogui accessos de lectura i escriptura a la base de dades, pot ser interessant tenir presents els consells següents:

- No afegiu camps a les taules del WordPress. Les taules poden canviar en versions posteriors, la qual cosa podria fer que el vostre *plug-in* deixés de funcionar.
- No afegiu registres directament a la taula `wp_options`, utilitzeu les funcions que el WordPress ofereix per a això.
- Useu sempre la classe `wpdb` per a referir-vos a les taules del WordPress. Això facilitarà que el vostre *plug-in* funcioni en totes les instal·lacions del WordPress.
- Si heu de crear una nova taula, utilitzeu el mètode `query` de la classe `wpdb`.

En podeu veure un exemple en el web de Hungred Dot Com.

### 5.1. Consultes senzilles

Comencem per un exemple senzill: obtenir els títols de totes les entrades del blog.

La consulta SQL seria:

```
SELECT wp_posts.post_title
FROM wp_posts;
```

Que "traduït" diu: seleccionem el camp títol de la taula `wp_posts`.

#### Prefix

Quan configurem el WordPress se'ns permet escollir el prefix que tindran les taules de la base de dades. Per defecte és `wp_`, però se'n pot escollir qualsevol altre. Per a l'elaboració d'aquest material, en els exemples SQL s'ha utilitzat el prefix per defecte.



No obstant això, com s'explica en els consells inicials d'aquest apartat, en els exemples de PHP s'usa la classe `$wpdb` que s'encarrega d'usar el prefix que correspongui.

Aquesta consulta s'hauria de fer en el WordPress de la manera següent:

```
global $wpdb;
$query = "SELECT $wpdb->posts.post_title
FROM $wpdb->posts";
$resultado = $wpdb->get_results($query, ARRAY_N);
```

Per a poder executar la instrucció SQL en el WordPress, primer hem de guardar-la en una variable per després executar-la.

La variable global `$wpdb` facilita la consulta en qualsevol instal·lació del gestor de blogs, ja que la fa independent del prefix que s'hagi escollit per a les taules en instal·lar el WordPress.

En la variable `$resultado` es guardarà un *array* amb el contingut de tots els títols del blog. No obstant això, si executem la instrucció tal qual, veurem molts títols repetits, títols d'entrades que no existeixen, títols de pàgines, etc. Toca repassar el que hem vist sobre la taula `wp_posts`.

El WordPress guarda en la taula `wp_posts` les dades necessàries de *posts*, pàgines, arxius pujats i menús creats pels usuaris. Per a poder diferenciar els diferents tipus de registres guardats en la taula, hi ha un camp anomenat `post_type` en què es guarda aquesta informació. Així en els *posts*, el valor pres per aquest camp serà 'post'.

Tenint això en compte, podem escriure la instrucció SQL que ens permetrà fer una llista solament dels títols dels *posts*:

```
SELECT wp_posts.post_title
FROM wp_posts
WHERE wp_posts.post_type = 'post'
```

No obstant això, aquesta instrucció inclou els esborranys i els *posts* programats per a veure més endavant.

Què hem de fer per a poder limitar els registres llistats a només als que estan publicats? Tal com s'explica en l'apartat corresponent, els *posts* publicats tenen en el camp `post_status` el valor de 'publish'. Així doncs, triarem els *posts* amb `post_status` 'publish' i `post_type` 'post'. Aquesta seria la consulta SQL:

```
SELECT wp_posts.post_title
FROM wp_posts
WHERE post_status = 'publish' AND post_type = 'post';
```

Per a fer la consulta amb PHP i en el WordPress, hauríem d'escriure el codi següent:

```
global $wpdb;
$query = "SELECT $wpdb->posts.post_title
FROM $wpdb->posts
WHERE $wpdb->posts.post_status = 'publish' AND $wpdb->posts.post_type = 'post'";
$resultado = $wpdb->get_results($query, ARRAY_N);
```

## 5.2. Accedir a les dades de dues taules

Com hem vist, fer una llista amb dades d'una sola taula resulta molt fàcil. Una mica més complicat resulta fer-ho amb dues taules. Però no gaire més.

Per a accedir al contingut de dues taules necessitem saber per quins camps es relacionen i per a això usarem les taules de relacions de l'apartat anterior.

A continuació, farem una llista amb els títols i la data dels *posts* publicats i les dates de tots els comentaris que s'han fet a cada *post*.

Com hem vist, les taules `wp_posts` i `wp_comments` estableixen la seva relació per mitjà dels camps `ID` i `comment_post_ID`, respectivament. Per a fer una llista dels camps que ens interessin, utilitzarem la instrucció SQL següent:

```
SELECT wp_comments.comment_date, wp_posts.post_title, wp_posts.ID
FROM wp_comments , wp_posts
WHERE wp_comments.comment_post_ID = wp_posts.ID;
```

Per a afinar una mica més, podem afegir les condicions que eliminen els *posts* sense publicar i les pàgines amb comentaris (si n'hi ha):

```
SELECT wp_comments.comment_date, wp_posts.post_title, wp_posts.ID
FROM wp_comments , wp_posts
WHERE wp_comments.comment_post_ID = wp_posts.ID
AND wp_posts.post_status = 'publish' AND wp_posts.post_type = 'post';
```

Encara queda eliminar d'aquesta llista els comentaris no desitjats (*spam*) i els no aprovats, amb la qual cosa ens quedarem només amb els aprovats:

```
SELECT wp_comments.comment_date, wp_posts.post_title, wp_posts.ID
FROM wp_comments, wp_posts
WHERE wp_comments.comment_post_ID = wp_posts.ID
AND wp_posts.post_status = 'publish' AND wp_posts.post_type = 'post'
AND wp_comments.comment_approved = 1;
```

Aquesta instrucció SQL fa una llista amb la data de tots els comentaris de cada *post*: per a un *post* que tingui quatre comentaris, tindrem quatre línies, cada una amb la data del comentari que toqui, el títol i l'ID del *post*.

Per exemple:

Exemple *post* amb 4 comentaris

comment_date	post_title	ID
2012-08-10 16:35:01	Fent proves amb el WordPress	2
2012-08-10 20:40:45	Fent proves amb el WordPress	2
2012-08-11 00:17:41	Fent proves amb el WordPress	2
2012-08-11 01:05:07	Fent proves amb el WordPress	2

### 5.3. Agrupant dades i comptant

L'SQL permet operacions molt més elaborades que les que hem vist fins ara. A continuació veurem un parell d'exemples una mica més complexos a partir del que hem fet en l'apartat anterior.

Hem vist la manera de fer una llista amb tots els comentaris de totes les entrades, escollint fins i tot només els que estaven aprovats. Ara comptarem el nombre total de comentaris **aprovats** que hi ha en el blog. Per a això usarem la funció **count (camp)**, que compta el nombre de registres en què el camp no és nul.

```
SELECT count(*) FROM wp_comments
WHERE wp_comments.comment_approved = 1;
```

Usar la funció count() fa que no es faci una llista dels registres, sinó que simplement es presenti el resultat final.

Però saber el nombre total de comentaris que hi ha en el blog no és gaire útil. Segurament és més útil (donaria una mena de rànquing de popularitat dels *posts*) una llista que ens digui quants comentaris té cada entrada.

Aquesta seria la consulta SQL que posaria en una llista, ID títol i nombre de comentaris de tots els *posts*:

```
SELECT ID, post_title, count(*) FROM wp_posts, wp_comments
WHERE comment_approved = 1
AND comment_post_ID = ID
AND post_status = 'publish'
AND post_type = 'post'
GROUP BY ID;
```

#### Funcions en l'MySQL

Podem veure una llista de funcions disponibles en l'MySQL en l'adreça següent: <http://dev.mysql.com/doc/refman/5.0/en/func-op-summary-ref.html>.

Com és fàcil d'imaginar, GROUP BY agrupa pel camp que li diguem, en aquest cas ID, amb la qual cosa es farà una llista de l'ID, títol i nombre de comentaris. En agrupar no rebem un sol valor, com en l'exemple anterior, sinó que obtenim un valor per cada ID.

Seguint amb l'exemple, podríem ordenar els *posts* per nombre de comentaris, i així obtindríem el nostre rànquing de popularitat:

```
SELECT ID, post_title, count(*) FROM wp_posts, wp_comments
WHERE comment_approved = 1
AND comment_post_ID = ID
AND post_status = 'publish'
AND post_type = 'post'
GROUP BY ID
ORDER BY count(*) DESC;
```

ORDER BY ordena pel camp que diguem, en aquest cas count(), i DESC indica que ordenarà de manera descendent, amb el *post* amb més comentaris primer.

Per acabar, podríem limitar a cinc el nombre de registres mostrats afegint LIMIT 5 al final.

#### 5.4. Categories i etiquetes

Possiblement les consultes més complicades que podem fer a la base de dades del WordPress són les relacionades amb categories i etiquetes. Com que per a posar en una llista el títol d'un *post* amb la categoria o categories a què està assignat necessitem passar per dues taules intermèdies, les consultes són més llargues.

Un exemple senzill seria posar en una llista els títols de totes les entrades i les seves categories:

```
SELECT post_title, name
FROM wp_posts, wp_term_taxonomy, wp_term_relationships, wp_terms
WHERE ID = object_id
AND wp_term_relationships.term_taxonomy_id = wp_term_taxonomy.term_taxonomy_id
AND wp_term_taxonomy.term_id = wp_terms.term_id
AND taxonomy = 'category';
```

Això faria una llista amb tots els registres de la taula *posts* amb una categoria assignada. Val la pena tenir en compte alguns detalls d'aquesta consulta:

- taxonomy és un camp de la taula wp\_term\_taxonomy. En comparar-lo amb 'category' automàticament fem una llista dels *posts* i deixem fora els enllaços.

- Si no és possible que hi hagi confusió en especificar el nom d'un camp, no cal indicar a quina taula pertany. No obstant això, en aquesta consulta hi ha diverses taules que comparteixen noms de camp, amb la qual cosa és imprescindible indicar la taula.
- Amb aquesta consulta, si hi ha *posts* amb més d'una categoria, el títol aparixerà repetit tantes vegades com categories tingui.

Podríem explicar quantes entrades hi ha de cada categoria:

```
SELECT name, count(*)
FROM wp_posts, wp_term_taxonomy, wp_term_relationships, wp_terms
WHERE ID = object_id
AND wp_term_relationships.term_taxonomy_id = wp_term_taxonomy.term_taxonomy_id
AND wp_term_taxonomy.term_id = wp_terms.term_id
AND taxonomy = 'category';
GROUP BY name;
```

Aquí faltaria escollir els *posts* publicats (ara mateix fa la llista de tots els *posts*).  
D'altra banda, fer el mateix per a les etiquetes seria molt senzill...

