

ANÁLISIS DE LA BASE DE DATOS DE WORDPRESS

Felipe Robles López – robleslf93@gmail.com

Un CMS necesita almacenar información, como nombres de usuario o configuraciones, dentro de una base de datos. WordPress, en particular, es una plataforma de creación web basada en la combinación de PHP y MySQL o MariaDB. Su funcionamiento básico es el siguiente: alguien visita el sitio web para ver el contenido, lo que genera una petición de información al servidor donde está alojada, entonces el lenguaje de programación PHP recibe esa petición y comprueba en la base de datos si existe la información solicitada, la recupera y la muestra al visitante a través de su navegador de Internet. Este proceso se completa de manera automática en cuestión de segundos, o incluso menos, dependiendo de la cantidad de contenido disponible y de datos almacenados. Esta base de datos tiene la siguiente estructura:

a. Tablas, relaciones, restricciones de la base de datos

wp_commentmeta

La tabla **wp_commentmeta** se utiliza para almacenar metadatos relacionados con comentarios en WordPress. Estos metadatos pueden ser información adicional o personalizada asociada a los comentarios y se almacenan junto con el ID del comentario al que pertenecen. La combinación de *comment_id* y *meta_key* se utiliza para identificar de manera única cada metadato de comentario.

Para ver más claro en qué consisten estos metadatos podemos poner un par de ejemplos:

1. Supongamos que deseamos destacar ciertos comentarios en nuestro sitio web como "Comentarios Destacados".

Podemos utilizar metadatos para marcar estos comentarios. Por ejemplo:

- Comentario 1: Usuario A deja un comentario que consideramos destacado, y marcamos el comentario con una etiqueta "Destacado".
- Comentario 2: Usuario B deja un comentario normal sin etiquetas.

Luego, podemos crear una sección especial en nuestro sitio web donde solo se muestren los "Comentarios Destacados", que nos mostraría solo el comentario 1..

2. Supongamos que tenemos un sitio web de reseñas de películas y se permite que los usuarios dejen comentarios sobre las películas. Podemos utilizar metadatos para capturar la calificación que cada usuario otorga a la película en su comentario. Por ejemplo:

- Comentario 1: Usuario A comenta la película "Rocky IV" y otorga una calificación de 4 estrellas.
- Comentario 2: Usuario B comenta la misma película y otorga una calificación de 5 estrellas.

Luego, podemos usar estos metadatos para calcular y mostrar la calificación promedio de la película en la página de la película.

Campos principales de la tabla wp_commentmeta

- **meta_id:** Este campo es la clave primaria de la tabla y se utiliza para identificar de manera única cada registro de metadato de comentario en WordPress. Es un número entero largo (*bigint*) con un ancho de 20 dígitos. Se genera automáticamente para cada nuevo registro y se incrementa secuencialmente, lo que garantiza que cada metadato

tenga su propio ID único.

- **comment_id:** Este campo almacena el ID del comentario al que está asociado el metadato. Es otro número entero largo (*bigint*) con un ancho de 20 dígitos. Este campo no permite valores nulos y tiene un valor predeterminado de 0. Establece una relación con la tabla **wp_comments**, lo que indica a qué comentario pertenece el metadato.
- **meta_key:** El campo *meta_key* almacena el nombre o clave del metadato del comentario. Es una cadena de texto (*varchar*) con un máximo de 255 caracteres y utiliza el conjunto de caracteres *utf8mb4_unicode_ci*. Puede permitir valores nulos. Este campo se utiliza para identificar y buscar metadatos específicos del comentario.
- **meta_value:** El campo *meta_value* almacena el valor del metadato del comentario. Es de tipo *longtext* y utiliza el conjunto de caracteres *utf8mb4_unicode_ci*. Puede permitir valores nulos. Este campo almacena el valor asociado al metadato del comentario, que puede ser cualquier tipo de información relacionada con el comentario.

Relaciones de la tabla **wp_commentmeta**

- Relación con la tabla **wp_comments** (Campo *comment_id*):
 - El campo *comment_id* en la tabla **commentmeta** se utiliza para relacionar cada entrada de metadatos con un comentario específico.
 - Ejemplo: Si tenemos un comentario en **wp_comments** con *comment_id* igual a 42, los metadatos relacionados con ese comentario tendrán una entrada en **wp_commentmeta** donde el campo *comment_id* sea 42.
 - Uso: Esta relación permite asociar metadatos personalizados a comentarios específicos. Por ejemplo, se podrían utilizar metadatos para almacenar información adicional sobre un comentario, como calificaciones, votos o datos personalizados.

Restricciones de la tabla **wp_commentmeta**

- **meta_id:** Clave primaria (PRIMARY KEY) y autoincremento (AUTO_INCREMENT).
 - **comment_id:** No permite valores nulos (NOT NULL).
 - **meta_key:** Puede permitir valores nulos o estar vacío dependiendo de la configuración.
 - **meta_value:** Puede permitir valores nulos o estar vacío dependiendo de la configuración.
-

wp_comments

La tabla **wp_comments** en WordPress se utiliza para almacenar los comentarios realizados por los usuarios en las entradas o páginas del sitio web. Los comentarios son una parte fundamental de la interacción en sitios web y permiten la participación de los usuarios fomentando la discusión.

Campos principales de la tabla **wp_comments**

- **comment_ID:** Este campo es la clave primaria de la tabla y se utiliza para identificar de manera única cada comentario en WordPress. Es un número entero largo (*bigint*) con un ancho de 20 dígitos. Se genera automáticamente para cada nuevo comentario y se incrementa secuencialmente.
- **comment_post_ID:** Almacena el ID de la entrada o página a la que se refiere el comentario. Es otro número entero largo (*bigint*) con un ancho de 20 dígitos. Este campo establece una relación con la tabla **wp_posts**, indicando a qué entrada o página pertenece el comentario.
- **comment_author:** Contiene el nombre del autor del comentario. Es un campo de tipo texto (*tinytext*) y puede contener hasta 255 caracteres.
- **comment_author_email:** Almacena la dirección de correo electrónico del autor del comentario. Es de tipo texto (*varchar*) con un máximo de 100 caracteres.

- **comment_author_url:** Este campo permite que el autor del comentario proporcione su sitio web, blog o perfil en redes sociales. Es de tipo texto (*varchar*) con un máximo de 200 caracteres.
- **comment_author_IP:** Registra la dirección IP del autor del comentario. Es de tipo texto (*varchar*) con un máximo de 100 caracteres.
- **comment_date:** Indica la fecha y hora en que se realizó el comentario. Se utiliza el formato *datetime* para este campo.
- **comment_date_gmt:** Almacena la fecha y hora del comentario en formato GMT (Tiempo Medio de Greenwich).
- **comment_content:** Contiene el texto del comentario en sí. Es un campo de tipo texto (*text*) y puede almacenar una cantidad significativa de texto.
- **comment_karma:** Este campo se utiliza para mantener un registro del "karma" o calificación del comentario, que puede ser utilizado para sistemas de votación o valoración de comentarios.
- **comment_approved:** Indica si el comentario ha sido aprobado o se encuentra en espera de moderación. Puede tener valores como "approve" (aprobado), "spam" (correo no deseado), "trash" (papelera), entre otros.
- **comment_agent:** Almacena información sobre el agente de usuario o navegador web que se utilizó para enviar el comentario. Es de tipo texto (*varchar*) con un máximo de 255 caracteres.
- **comment_type:** Este campo se utiliza para categorizar el tipo de comentario, por ejemplo, si es un comentario estándar, un pingback o un trackback.
- **comment_parent:** Establece una relación jerárquica entre comentarios, indicando cuál es el comentario padre de un comentario específico.
- **user_id:** Si el comentario ha sido realizado por un usuario registrado, este campo almacena el ID de usuario correspondiente. De lo contrario, se establece en 0.

Relaciones de la tabla wp_comments

- Relación con **wp_posts** (Campo *comment_post_ID*):
 - El campo *comment_post_ID* en la tabla **wp_comments** se utiliza para relacionar cada comentario con la entrada o página a la que hace referencia. Este campo almacena el *post_id* de la entrada o página asociada.
 - Ejemplo: Supongamos que tenemos una entrada con *post_id* 99 titulada "Mi Entrada" y un comentario en el que *comment_post_ID* es igual a 99. Esta relación indica que el comentario pertenece a la entrada con *post_id* 99, es decir, "Mi Entrada".
 - Uso: Esta relación es crucial para mostrar comentarios específicos en las páginas de entradas correspondientes. WordPress utiliza esta relación para recuperar y mostrar los comentarios adecuados bajo cada entrada o página.
- Relación con la tabla wp_commentmeta (Campo *comment_ID*):
 - El campo *comment_ID* en la tabla **wp_commentmeta** se utiliza para relacionar cada entrada de metadatos con un comentario específico en **wp_comments**.
 - Ejemplo: Si tenemos un comentario en **wp_comments** con *comment_ID* igual a 42, los metadatos relacionados con ese comentario tendrán una entrada en **wp_commentmeta** donde el campo *comment_ID* sea 42.
 - Uso: Esta relación permite asociar metadatos personalizados a comentarios específicos. Por ejemplo, se podrían utilizar metadatos para almacenar información adicional sobre un comentario, como calificaciones, votos o datos personalizados, de manera similar a la relación entre **wp_commentmeta** y **wp_comments**.
- Relación Jerárquica (Campo *comment_parent*):

-Campo Relevante: El campo *comment_parent* se utiliza para establecer relaciones jerárquicas entre comentarios. Define si un comentario es una respuesta a otro comentario y establece el comentario padre.

-Ejemplo Práctico: Si tenemos un comentario A con *comment_ID* 22 y un comentario B con *comment_ID* 23 como respuesta a A, el campo *comment_parent* de B se establecerá en 22, lo que indica que B es una respuesta a A.

-Uso: Esta relación jerárquica se utiliza para estructurar las conversaciones y respuestas en los comentarios. Facilita la organización y visualización de los hilos de comentarios.

- Relación con **wp_users** (Campos *comment_author* y *_comment_author_id*):
 - Campo Relevante: Los campos *comment_author* y *_comment_author_id* se relaciona con la tabla **wp_users** mediante el ID del usuario. Este campo indica quién es el autor del comentario.-
 - Ejemplo: Si el campo *comment_author* de un comentario es 5, significa que el usuario de la tabla **wp_users** con ID 5 es el autor de ese comentario.-
 - Uso: Esta relación permite identificar y atribuir comentarios a sus respectivos autores. Facilita la administración y la atribución de comentarios en el sitio web, lo que es especialmente útil en sitios con múltiples autores.

Restricciones de la tabla *wp_comments*

- **comment_ID**: Clave primaria (PRIMARY KEY) y autoincremento (AUTO_INCREMENT). Esta restricción garantiza que cada comentario tenga un ID único y que se genere automáticamente para cada nuevo comentario.
- **comment_post_ID**: No permite valores nulos (NOT NULL). Esto asegura que cada comentario esté asociado a una entrada o página específica en WordPress y no pueda quedar sin asignar.
- **comment_author**: No permite valores nulos (NOT NULL). Asegura que cada comentario tenga un autor especificado.
- **comment_author_email**: Puede permitir valores nulos o estar vacío dependiendo de la configuración. No se impone una restricción de unicidad en este campo, lo que significa que varios comentarios pueden tener la misma dirección de correo electrónico si es necesario.
- **comment_author_url**: Puede permitir valores nulos o estar vacío dependiendo de la configuración. Esto permite que el campo de URL del autor sea opcional en algunos comentarios.
- **comment_author_IP**: Puede permitir valores nulos o estar vacío dependiendo de la configuración. Esto permite que el campo de dirección IP del autor sea opcional en algunos comentarios.
- **comment_date**: No permite valores nulos (NOT NULL). Asegura que cada comentario tenga una fecha y hora asociada.
- **comment_date_gmt**: No permite valores nulos (NOT NULL). Asegura que cada comentario tenga una fecha y hora en formato GMT asociada.
- **comment_content**: No permite valores nulos (NOT NULL). Garantiza que cada comentario tenga contenido de texto.
- **comment_karma**: No permite valores nulos (NOT NULL). Asegura que el campo de karma del comentario tenga un valor especificado.
- **comment_approved**: Puede permitir valores nulos o estar vacío dependiendo de la configuración. Esto permite que el campo de aprobación de comentarios sea opcional en algunos casos.
- **comment_agent**: Puede permitir valores nulos o estar vacío dependiendo de la configuración. Esto permite que el campo del agente de usuario del comentario sea opcional en algunos comentarios.
- **comment_type**: Puede permitir valores nulos o estar vacío dependiendo de la configuración. Esto permite que el campo de tipo de comentario sea opcional en algunos comentarios.
- **comment_parent**: No permite valores nulos (NOT NULL). Asegura que cada comentario tenga un comentario padre especificado.

- **user_id:** No permite valores nulos (NOT NULL). Garantiza que cada comentario esté asociado a un usuario específico de WordPress.
-

wp_links

La tabla **wp_links** almacena información relacionada con los enlaces o hipervínculos en un sitio web de WordPress. Estos enlaces pueden ser utilizados para redirigir a los usuarios a otras páginas web o recursos en línea. A continuación, se describen los campos principales, relaciones y restricciones de esta tabla.

Campos Principales de la tabla wp_links:

- **link_id:** Este campo es la clave primaria de la tabla y se utiliza para identificar de manera única cada enlace. Es un número entero largo (*bigint*) con un ancho de 20 dígitos. Se genera automáticamente para cada nuevo enlace y se incrementa secuencialmente, lo que garantiza que cada enlace tenga su propio ID único.
- **link_url:** Almacena la dirección URL a la que se redirige el enlace. Es de tipo *varchar(255)* y utiliza el conjunto de caracteres *utf8mb4_unicode_ci*. Este campo es esencial para definir la ubicación de destino del enlace.
- **link_name:** Contiene el nombre o título del enlace. Es de tipo *varchar(255)* y utiliza *utf8mb4_unicode_ci*. Este campo proporciona una descripción legible para el enlace.
- **link_image:** Puede almacenar una URL de imagen relacionada con el enlace. Es de tipo *varchar(255)* y utiliza *utf8mb4_unicode_ci*. Esto permite asociar una imagen con el enlace, que puede ser utilizada para representar visualmente el enlace.
- **link_target:** Indica la forma en que se abrirá el enlace. Es de tipo *varchar(25)* y utiliza *utf8mb4_unicode_ci*. Los valores típicos pueden ser "*blank*" para abrir en una nueva ventana o "*self*" para abrir en la misma ventana.
- **link_description:** Permite agregar una breve descripción del enlace. Es de tipo *varchar(255)* y utiliza *utf8mb4_unicode_ci*. Esta descripción puede proporcionar información adicional sobre el enlace.
- **link_visible:** Este campo indica si el enlace es visible en el sitio web. Es de tipo *varchar(20)* y utiliza *utf8mb4_unicode_ci*. Puede tener valores como "Y" para sí o "N" para no. Esto permite controlar la visibilidad de los enlaces en el sitio.
- **link_owner:** Almacena el ID del usuario que es propietario del enlace. Es de tipo *bigint(20)* y se relaciona con la tabla de usuarios (**wp_users**). Esto permite atribuir la propiedad del enlace a un usuario específico.
- **link_rating:** Permite asignar una valoración o puntuación al enlace. Es de tipo *int(11)* y no permite valores nulos. Esto puede utilizarse para clasificar o calificar los enlaces.
- **link_updated:** Almacena la fecha y hora de la última actualización del enlace. Es de tipo *datetime* y se utiliza para rastrear los cambios en el enlace.
- **link_rel:** Puede contener información sobre la relación del enlace con otros recursos. Es de tipo *varchar(255)* y utiliza *utf8mb4_unicode_ci*. Esto permite establecer relaciones específicas con otros enlaces o recursos.
- **link_notes:** Este campo permite agregar notas adicionales o comentarios sobre el enlace. Es de tipo *mediumtext* y utiliza *utf8mb4_unicode_ci*. Las notas pueden proporcionar información detallada sobre el enlace.
- **link_rss:** Almacena la dirección URL del feed RSS asociado con el enlace, si corresponde. Es de tipo *varchar(255)* y utiliza *utf8mb4_unicode_ci*. Esto permite conectar el enlace con un feed RSS relacionado.

Relaciones de la tabla wp_links:

- Relación con **wp_users** (Campo *link_owner*):
 - Campo Relevante: El campo *link_owner* se relaciona con la tabla *wp_users* mediante el ID del usuario propietario del enlace. Esto permite identificar quién es el propietario del enlace.-

- Ejemplo: Si el campo *link_owner* de un enlace tiene el valor 5, significa que el usuario de la tabla *wp_users* con ID 5 es el propietario de ese enlace.-
- Uso: Esta relación facilita la gestión de la propiedad de los enlaces en el sitio web, lo que permite atribuir y administrar enlaces específicos a usuarios individuales.

Restricciones de la tabla *wp_links*:

- **link_id:** Clave primaria (PRIMARY KEY) y autoincremento (AUTO_INCREMENT).
- **link_url:** No permite valores nulos (NOT NULL).
- **link_name:** No permite valores nulos (NOT NULL).
- **link_target:** No permite valores nulos (NOT NULL).
- **link_description:** No permite valores nulos (NOT NULL).
- **link_visible:** No permite valores nulos (NOT NULL).
- **link_owner:** No permite valores nulos (NOT NULL) y tiene un valor predeterminado de 1.
- **link_rating:** No permite valores nulos (NOT NULL) y tiene un valor predeterminado de 0.
- **link_updated:** No permite valores nulos (NOT NULL) y tiene un valor predeterminado de "0000-00-00 00:00:00".
- **link_rel:** No permite valores nulos (NOT NULL).
- **link_notes:** No permite valores nulos (NOT NULL).
- **link_rss:** No permite valores nulos (NOT NULL).

wp_options

La tabla *wp_options* en WordPress es fundamental para almacenar y gestionar configuraciones y opciones del sitio web. Permite a los administradores de WordPress personalizar el comportamiento y la apariencia de su sitio de una manera versátil. Cada fila en esta tabla representa una opción con un nombre único, y el valor asociado a esa opción.

Campos Principales de la tabla *wp_options*

- **option_id:** Este campo es la clave primaria de la tabla y se utiliza para identificar de manera única cada opción en WordPress. Es un número entero largo (*bigint*) con un ancho de 20 dígitos. Se genera automáticamente para cada nueva opción y se incrementa secuencialmente, garantizando que cada opción tenga su propio ID único.
- **option_name:** Almacena el nombre de la opción. Es de tipo *varchar(191)* y utiliza el conjunto de caracteres *utf8mb4_unicode_ci*. Este campo es esencial para identificar y acceder a cada opción de manera única. No permite valores nulos.
- **option_value:** Almacena el valor asociado con la opción. Es de tipo *longtext* y utiliza el conjunto de caracteres *utf8mb4_unicode_ci*. Aquí se guarda la información o configuración específica correspondiente a cada opción. Puede contener datos de diversos tipos, como texto, números o incluso estructuras de datos más complejas. No permite valores nulos.

***autoload:** Indica si la opción debe cargarse automáticamente al iniciar WordPress. Este campo es de tipo *varchar(20)* y utiliza el conjunto de caracteres *utf8mb4_unicode_ci*. Los valores típicos son "yes" para cargar automáticamente y "no" para no cargar automáticamente. Determina si la opción se carga en la memoria cuando WordPress arranca. No permite valores nulos.

Relaciones de la tabla *wp_options*

Esta tabla no está directamente relacionada con otras tablas mediante claves foráneas o campos específicos de relación. La tabla **wp_options** generalmente se utiliza para almacenar configuraciones y opciones generales del sitio, como las opciones de configuración de WordPress, plugins, temas y otras personalizaciones del sitio. Estas opciones se guardan en la tabla **wp_options** bajo el campo *option_name*, pero no establece relaciones directas con otras tablas.

Restricciones de la tabla wp_options:

- **option_id:** Clave primaria (PRIMARY KEY) y autoincremento (AUTO_INCREMENT). Esta restricción asegura que cada registro en la tabla tenga un ID único y que se genere automáticamente para cada nuevo registro.
- **option_name:** No permite valores nulos (NOT NULL) y debe ser único (restricción de unicidad). Esto garantiza que no se puedan ingresar valores nulos y que cada nombre de opción sea único en la tabla.
- **option_value:** No permite valores nulos (NOT NULL). Esta restricción asegura que cada registro tenga un valor de opción asociado y que no se permitan valores nulos en esta columna.
- **autoload:** No permite valores nulos (NOT NULL) y tiene un valor predeterminado de "yes". Esto significa que el campo "autoload" no puede estar vacío y, si no se especifica un valor al insertar un nuevo registro, se utilizará el valor predeterminado "yes".

wp_postmeta

La tabla wp_postmeta se utiliza para almacenar metadatos relacionados con las entradas y páginas en WordPress. Estos metadatos pueden ser información adicional o personalizada asociada a las entradas y páginas, y se almacenan junto con el ID de la entrada o página a la que pertenecen. La combinación de *post_id* y *meta_key* se utiliza para identificar de manera única cada metadato de entrada o página.

Para comprender mejor en qué consisten estos metadatos, podemos poner un ejemplo relacionado con una página de películas y series:

- Supongamos que tenemos un sitio web dedicado a reseñas de películas y series, y queremos enriquecer la información de cada reseña. Podemos utilizar *wp_postmeta* para almacenar información adicional sobre cada película o episodio de serie. Por ejemplo:
 - Entrada 1: Una reseña de la película con metadatos que incluyen la calificación, el director, el año de lanzamiento y un resumen de la trama.
 - Entrada 2: Otra reseña sobre un episodio de una serie con metadatos que indican la temporada y el número de episodio, la calificación y una breve descripción.
 - De esta manera, podemos proporcionar información detallada sobre cada película o episodio de serie, lo que facilita a los usuarios encontrar contenido relevante y tomar decisiones sobre qué ver.

Campos principales de la tabla wp_postmeta

- **meta_id:** Este campo es la clave primaria de la tabla y se utiliza para identificar de manera única cada registro de metadato de entrada o página en WordPress. Es un número entero largo (*bigint*) con un ancho de 20 dígitos. Se genera automáticamente para cada nuevo registro y se incrementa secuencialmente, lo que garantiza que cada metadato tenga su propio ID único.
- **post_id:** El campo *post_id* almacena el ID de la entrada o página a la que está asociado el metadato. Es otro número entero largo (*bigint*) con un ancho de 20 dígitos. Este campo establece una relación con la tabla **wp_posts**, lo que indica a qué entrada o página pertenece el metadato.
- **meta_key:** El campo *meta_key* almacena el nombre o clave del metadato de la entrada o página. Es una cadena de texto (*varchar*) con un máximo de 255 caracteres y utiliza el conjunto de caracteres *utf8mb4_unicode_ci*. Puede permitir valores nulos. Este campo se utiliza para identificar y buscar metadatos específicos de la entrada o página.

- **meta_value:** El campo *meta_value* almacena el valor del metadato de la entrada o página. Es de tipo *longtext* y utiliza el conjunto de caracteres *utf8mb4_unicode_ci*. Puede permitir valores nulos. Este campo almacena el valor asociado al metadato de la entrada o página, que puede ser cualquier tipo de información relacionada con la entrada o página, como calificaciones, etiquetas, resúmenes, o cualquier dato personalizado necesario para el sitio web.

Relaciones de la tabla `wp_postmeta`

- Relación con **wp_posts** (Campo *post_id*):
 - El campo *post_id* en la tabla **wp_postmeta** se utiliza para relacionar cada registro de metadato con una entrada o página específica en WordPress. Este campo almacena el *post_id* de la entrada o página a la que está vinculado el metadato.-
 - Ejemplo: Supongamos que tenemos una entrada con *post_id* 12 titulada "Mi Entrada" y un registro en `wp_postmeta` con *post_id* igual a 12. Esta relación indica que el metadato en cuestión pertenece a la entrada con *post_id* 12, es decir, "Mi Entrada".-
 - Uso: Esta relación permite asociar metadatos personalizados a entradas o páginas específicas. Los metadatos pueden contener información adicional o personalizada que se relaciona con una entrada o página particular.

Restricciones de la tabla `wp_postmeta`:

- **meta_id:** Clave primaria (PRIMARY KEY) y autoincremento (AUTO_INCREMENT). Esta restricción garantiza que cada registro de metadatos tenga un ID único y que se genere automáticamente para cada nuevo registro.
 - **post_id:** No permite valores nulos (NOT NULL). Esto asegura que cada metadato esté asociado a una entrada o página específica en WordPress y no pueda quedar sin asignar.
 - **meta_key:** Puede permitir valores nulos o estar vacío dependiendo de la configuración. No se impone una restricción de unicidad en este campo, lo que permite que múltiples registros tengan la misma clave de metadato si es necesario.
 - **meta_value:** Puede permitir valores nulos o estar vacío dependiendo de la configuración. Esto permite que el valor del metadato pueda ser opcional en algunos casos, pero debe seguir las restricciones de tipo de datos (*longtext*).
-

`wp_posts`

La tabla **wp_posts** almacena información sobre todas las entradas y páginas del sitio web.

Campos Principales de la tabla `wp_posts`:

- **post_id:** Este campo es la **clave primaria** de la tabla y se utiliza para identificar de manera única cada entrada o página en WordPress. Es un número entero largo (*bigint*) con un ancho de 20 dígitos. Se genera automáticamente para cada nueva entrada y se incrementa secuencialmente, por lo que cada entrada tiene su propio ID único.
- **post_author:** Este campo almacena el ID del usuario que creó la entrada o página. Es otro número entero largo (*bigint*) con un ancho de 20 dígitos. La relación con la tabla de `wp_users` se establece mediante este campo, lo que permite saber quién es el autor de cada entrada.
- **post_date:** Almacena la fecha y hora exactas en que se creó la entrada o página. Es de tipo *datetime* y no permite valores nulos. La fecha y hora se registran en el formato "YYYY-MM-DD HH:MM:SS". Esto facilita la ordenación de las entradas y páginas por fecha.
- **post_date_gmt:** Similar al campo *post_date*, almacena la fecha y hora de creación en formato GMT (Hora Media de Greenwich). Al igual que *post_date*, es de tipo *datetime* y no permite valores nulos. Esto asegura que las fechas sean consistentes y no dependan de la zona horaria del servidor.

- **post_content:** Este campo almacena el contenido principal de la entrada o página. Es de tipo *longtext*, lo que significa que puede contener una gran cantidad de texto y otros elementos, como imágenes y etiquetas HTML. Utiliza el conjunto de caracteres *utf8mb4_unicode_ci* para admitir caracteres especiales y emojis. Este campo es esencial para el contenido visible en el sitio web.
- **post_title:** Almacena el título de la entrada o página. Es de tipo *text* y utiliza el conjunto de caracteres *utf8mb4_unicode_ci*. Este campo es importante para la identificación y presentación del contenido en el sitio web.
- **post_excerpt:** Este campo almacena un extracto o resumen de la entrada, si se proporciona. Es de tipo *text* y también utiliza *utf8mb4_unicode_ci*. El extracto se puede utilizar en las páginas de archivo o en resúmenes de entradas.
- **post_status:** Indica el estado de la entrada, como "*publish*" (publicado), "*draft*" (borrador), "*private*" (privado), entre otros. Es de tipo *varchar* con un ancho de 20 caracteres y su valor predeterminado es "*publish*" (publicado).
- **comment_status:** Indica el estado de los comentarios para la entrada, como "*open*" (abierto) o "*closed*" (cerrado). Es de tipo *varchar* con un ancho de 20 caracteres y su valor predeterminado es "*open*" (abierto).
- **ping_status:** Indica el estado de pingbacks y trackbacks para la entrada, como "*open*" (abierto) o "*closed*" (cerrado). Es de tipo *varchar* con un ancho de 20 caracteres y su valor predeterminado es "*open*" (abierto). Los pingbacks y trackbacks son métodos utilizados en sistemas de blogs y sitios web para notificar a otros sitios web cuando se ha enlazado a su contenido.
- **post_password:** Este campo permite establecer una contraseña para proteger el acceso a la entrada. Es de tipo *varchar* con un ancho de 255 caracteres y es opcional. Si se establece una contraseña, se almacena aquí.
- **post_name:** Almacena un slug (URL amigable) generado automáticamente a partir del título de la entrada. Es de tipo *varchar* con un ancho de 200 caracteres y es importante para la construcción de las URL de las entradas.
- **to_ping:** Este campo almacena una lista de URLs a las que se enviarán pingbacks cuando se publique la entrada. Es de tipo *text* y se utiliza en procesos de notificación a otros sitios web cuando se enlaza a ellos desde la entrada.
- **pinged:** Almacena una lista de URLs de sitios web que han enviado pingbacks a esta entrada. Es de tipo *text* y se actualiza automáticamente cuando se reciben pingbacks.
- **post_modified:** Almacena la fecha y hora de la última modificación de la entrada o página. Es de tipo *datetime* y se utiliza para rastrear cambios en el contenido.
- **post_modified_gmt:** Al igual que *post_modified*, almacena la fecha y hora de la última modificación en formato GMT.
- **post_content_filtered:** Almacena una versión filtrada del contenido de la entrada. Puede usarse para almacenar una versión del contenido con ciertos elementos eliminados o modificados. Es de tipo *longtext*. Esta versión filtrada del contenido generalmente se utiliza para optimizar o modificar el contenido original de alguna manera específica para ciertos propósitos o plugins.
- **post_parent:** Este campo establece la relación entre entradas principales y sus hijos o páginas secundarias. Es de tipo *bigint* y se utiliza para establecer la jerarquía de las páginas.
- **guid:** Almacena una URL única globalmente identificable que se utiliza principalmente en procesos de importación y exportación de contenidos. Es de tipo *varchar*.
- **menu_order:** Se utiliza para establecer el orden en que las entradas aparecen en los menús personalizados. Es de tipo *int*.
- **post_type:** Indica el tipo de contenido, como "*post*" (entrada) o "*page*" (página). Es de tipo *varchar* con un ancho de 20 caracteres y su valor predeterminado es "*post*".

- **post_mime_type:** Este campo se utiliza para almacenar el tipo de archivo asociado a la entrada, si corresponde. Por ejemplo, para una entrada que adjunta una imagen, podría contener "*image/jpeg*". Es de tipo *varchar* con un ancho de 100 caracteres.
- **comment_count:** Almacena el número de comentarios asociados a la entrada. Es de tipo *bigint* y se actualiza automáticamente cuando se agregan o eliminan comentarios.

Relaciones de la tabla `wp_posts`

- Relación con **wp_comments** (Campo *comment_post_ID*):
 - El campo *comment_post_ID* en la tabla **wp_comments** se utiliza para relacionar cada comentario con la entrada o página a la que hace referencia. Este campo almacena el *post_id* de la entrada o página asociada.
 - Ejemplo: Supongamos que tenemos una entrada con *post_id* 14 titulada "Mi Entrada" y un comentario en el que *comment_post_ID* es igual a 14. Esta relación indica que el comentario pertenece a la entrada con *post_id* 14, es decir, "Mi Entrada".
 - Uso: Esta relación es crucial para mostrar comentarios específicos en las páginas de entradas correspondientes. WordPress utiliza esta relación para recuperar y mostrar los comentarios adecuados bajo cada entrada o página.
- Relación con la tabla **wp_term_relationships** (Campo *object_id*):
 - El campo *ID* en la tabla **wp_posts** se utiliza para relacionar cada entrada o publicación con una o más relaciones de términos almacenadas en la tabla **wp_term_relationships**.
 - Ejemplo: Si tenemos una entrada en **wp_posts** con *ID* igual a 30, significa que esta entrada está relacionada con una o más relaciones de términos en la tabla **wp_term_relationships** donde el campo *object_id* sea igual a 30.
 - Uso: Esta relación permite asignar términos, que pueden ser categorías o etiquetas, a las publicaciones. Por ejemplo, una entrada puede estar relacionada con las categorías "Tecnología" y "Gadgets", lo que ayuda a organizar y clasificar el contenido en WordPress. La tabla `wp_term_relationships` actúa como un puente entre las publicaciones y los términos que las categorizan.
- Relación Jerárquica (Campo *post_parent*):
 - Campo Relevante: El campo *post_parent* se utiliza para establecer relaciones jerárquicas entre entradas o páginas. Define la entrada principal (padre) a la que pertenece una página secundaria (hija).
 - Ejemplo Práctico: Si tenemos una página principal llamada "Productos" con *post_id* 17 y una página secundaria llamada "Teclados" con *post_id* 131, el campo *post_parent* de la página "Teclados" se establecerá en 17, lo que indica que "Teclados" es una página hija de "Productos".
 - Uso: Esta relación jerárquica se utiliza para estructurar el contenido del sitio web en árboles de navegación. Por ejemplo, en los menús de navegación, se mostrarán las páginas secundarias como subelementos de las páginas principales.
- Relación con **wp_users** (Campo *post_author*):
 - Campo Relevante: El campo *post_author* se relaciona con la tabla **wp_users** mediante el ID del usuario. Este campo indica quién es el autor de una entrada o página.
 - Ejemplo: Si el campo *post_author* de una entrada es 5, significa que el usuario de la tabla **wp_users** con ID 5 es el autor de esa entrada.
 - Uso: Esta relación permite identificar y atribuir entradas y páginas a sus respectivos autores. Facilita la administración y la atribución de contenido en el sitio web, lo que es especialmente útil en sitios con múltiples autores.

Restricciones de la tabla `wp_posts`

- **post_id:** Clave primaria (PRIMARY KEY) y autoincremento (AUTO_INCREMENT).

- **post_author:** No permite valores nulos (NOT NULL).
- **post_date:** No permite valores nulos (NOT NULL).
- **post_date_gmt:** No permite valores nulos (NOT NULL).
- **post_content:** No permite valores nulos (NOT NULL).
- **post_title:** No permite valores nulos (NOT NULL).
- **post_excerpt:** No permite valores nulos (NOT NULL).
- **post_status:** No permite valores nulos (NOT NULL).
- **comment_status:** No permite valores nulos (NOT NULL).
- **ping_status:** No permite valores nulos (NOT NULL).
- **post_password:** Puede permitir valores nulos o estar vacío dependiendo de la configuración.
- **post_name:** No permite valores nulos (NOT NULL).
- **post_parent:** No permite valores nulos (NOT NULL).
- **guid:** No permite valores nulos (NOT NULL).
- **menu_order:** No permite valores nulos (NOT NULL).
- **post_mime_type:** Puede permitir valores nulos o estar vacío dependiendo de la configuración.
- **comment_count:** No permite valores nulos (NOT NULL).

wp_termmeta

La tabla wp_termmeta en WordPress se utiliza para almacenar metadatos personalizados relacionados con los términos de taxonomía. Los términos de taxonomía son utilizados para organizar y categorizar contenido, como publicaciones y páginas, en el sitio web. Los metadatos de términos proporcionan información adicional y personalizada sobre estos términos.

Los metadatos son datos extra que se utilizan para dar más detalles o categorizar elementos, como entradas de blog, imágenes o términos de categoría. Por ejemplo, para una entrada de blog, los metadatos podrían incluir la fecha de publicación, el autor y las etiquetas relacionadas. Ayudan a organizar, buscar y mostrar contenido de manera efectiva en un sitio web.

Un ejemplo real sería el siguiente:

Supongamos que tenemos un sitio web dedicado a reseñas de libros, y utilizamos la taxonomía "Géneros" para clasificar los libros en diferentes categorías. Los metadatos se usarían así:

- Términos de Taxonomía (Géneros):
 - Término 1: "Ficción"
 - Término 2: "No Ficción"
 - Término 3: "Ciencia Ficción"
- Metadatos de Términos (wp_termmeta):
 - Para el término "Ficción":
 - meta_key: "Descripción"
 - meta_value: "Esta categoría incluye novelas y cuentos de ficción."

- Para el término "No Ficción":
-meta_key: "Descripción" -meta_value: "Esta categoría abarca libros basados en hechos reales y no ficción." -meta_key: "Número de Libros" -meta_value: "25" (indicando la cantidad de libros en esta categoría)
- Para el término "Ciencia Ficción":
- meta_key: "Descripción"
- meta_value: "Esta categoría incluye obras de ciencia ficción, viajes en el tiempo y alienígenas."

A continuación, se describen los aspectos clave de esta tabla:

Campos principales de la tabla `wp_termmeta`

- **meta_id:** Este campo es la clave primaria de la tabla y se utiliza para identificar de manera única cada registro de metadato de término en WordPress. Es un número entero largo (*bigint*) con un ancho de 20 dígitos. Se genera automáticamente para cada nuevo registro y se incrementa secuencialmente, lo que garantiza que cada metadato tenga su propio ID único.
- **term_id:** Este campo almacena el ID del término de taxonomía al que está asociado el metadato. Es otro número entero largo (*bigint*) con un ancho de 20 dígitos. Este campo establece una relación con la tabla **wp_terms**, lo que indica a qué término pertenece el metadato.
- **meta_key:** El campo *meta_key* almacena el nombre o clave del metadato del término. Es una cadena de texto (*varchar*) con un máximo de 255 caracteres y utiliza el conjunto de caracteres `utf8mb4_unicode_ci`. Cada clave de metadato es única dentro del contexto de un término específico y se utiliza para identificar y buscar metadatos relacionados con ese término.
- **meta_value:** El campo *meta_value* almacena el valor del metadato del término. Es de tipo *longtext* y utiliza el conjunto de caracteres `utf8mb4_unicode_ci`. Puede contener cualquier tipo de información relacionada con el término de taxonomía.

Relaciones de la tabla `wp_termmeta`

- Relación con **wp_terms** (Campo *term_id*):
 - El campo *term_id* en la tabla **wp_termmeta** se utiliza para relacionar cada registro de metadato con un término de taxonomía específico en WordPress. Este campo almacena el *term_id* del término al que está vinculado el metadato.
 - Ejemplo: Supongamos que tenemos un término de categoría con *term_id* 5 llamado "Tecnología" y un registro en **wp_termmeta** con *term_id* igual a 5. Esta relación indica que el metadato en cuestión pertenece al término "Tecnología".
 - Uso: Esta relación permite asociar metadatos personalizados a términos de taxonomía específicos. Los metadatos pueden contener información adicional o personalizada relacionada con una categoría o etiqueta en particular.

Restricciones de la tabla `wp_termmeta`

- **meta_id:** Clave primaria (PRIMARY KEY) y autoincremento (AUTO_INCREMENT). Esta restricción garantiza que cada registro de metadatos tenga un ID único y que se genere automáticamente para cada nuevo registro.
- **term_id:** No permite valores nulos (NOT NULL). Esto asegura que cada metadato esté asociado a una etiqueta o categoría específica en WordPress y no pueda quedar sin asignar.
- **meta_key:** Puede permitir valores nulos o estar vacío dependiendo de la configuración. No se impone una restricción de unicidad en este campo, lo que permite que múltiples registros tengan la misma clave de metadato si es necesario.

- **meta_value:** Puede permitir valores nulos o estar vacío dependiendo de la configuración. Esto permite que el valor del metadato pueda ser opcional en algunos casos, pero debe seguir las restricciones de tipo de datos (longtext).
-

wp_terms

La tabla `wp_terms` en WordPress almacena información sobre las etiquetas y categorías utilizadas en el sitio web. Cada fila de esta tabla representa una etiqueta o categoría con un nombre único. Estas etiquetas y categorías se utilizan para organizar y clasificar el contenido del sitio, lo que facilita la navegación y la búsqueda de contenido relacionado.

Campos principales de la tabla `wp_terms`:

- **term_id:** Este campo es la clave primaria (PRIMARY KEY) de la tabla y se utiliza para identificar de manera única cada término de taxonomía en WordPress. Es un número entero largo (*bigint*) con un ancho de 20 dígitos. Se genera automáticamente para cada nuevo término y se incrementa secuencialmente, lo que garantiza que cada término tenga su propio ID único.
- **name:** El campo `name` almacena el nombre del término de taxonomía. Es una cadena de texto (*varchar*) con un máximo de 200 caracteres y utiliza el conjunto de caracteres `utf8mb4_unicode_ci`. Este campo almacena el nombre descriptivo del término, como "Noticias", "Deportes", "Tecnología", "Cine", etc.
- **slug:** El campo `slug` almacena la versión amigable para URL del término. Es una cadena de texto (*varchar*) con un máximo de 200 caracteres y utiliza el conjunto de caracteres `utf8mb4_unicode_ci`. El valor en este campo se utiliza en las URLs del sitio web, por lo que suele estar en minúsculas y separado por guiones, por ejemplo, "noticias", "deportes", "tecnologia", "cine", etc.
- **term_group:** Este campo se utiliza para agrupar términos relacionados, pero su uso es menos común en versiones recientes de WordPress. Es un número entero (*bigint*) con un ancho de 10 dígitos y no permite valores nulos. Por defecto, se establece en 0, lo que significa que el término no pertenece a ningún grupo específico.

Relaciones de la tabla `wp_terms`

- Relación con **wp_termmeta** (Campo `term_id`):
 - El campo `term_id` en la tabla **wp_terms** se utiliza para relacionar cada término de taxonomía con metadatos personalizados almacenados en la tabla **wp_termmeta**. Este campo actúa como clave primaria en `wp_terms` y clave foránea en `wp_termmeta`.
 - Ejemplo: Supongamos que tenemos un término de categoría con `term_id` 5 llamado "Tecnología" y un registro en `wp_termmeta` con `term_id` igual a 5. Esta relación indica que el metadato en cuestión pertenece al término "Tecnología".
 - Uso: Esta relación permite asociar metadatos personalizados a términos de taxonomía específicos, lo que puede ser útil para agregar información adicional o personalizada a categorías o etiquetas.
- Relación con **wp_term_taxonomy** (Campo `term_id`):
 - El campo `term_id` en la tabla **wp_terms** se utiliza para relacionar cada término de taxonomía con la tabla `wp_term_taxonomy`. Este campo actúa como clave primaria en **wp_terms** y clave foránea en `wp_term_taxonomy`.
 - Ejemplo: Supongamos que tenemos un término de categoría con `term_id` 5 llamado "Tecnología" y un registro en **wp_term_taxonomy** con `term_id` igual a 5. Esta relación indica que el término "Tecnología" está asociado a una entrada en **wp_term_taxonomy** que contiene información adicional sobre la taxonomía a la que pertenece.
 - Uso: Esta relación permite vincular cada término de taxonomía en **wp_terms** con información más detallada en **wp_term_taxonomy**, lo que facilita la gestión y organización de términos de taxonomía en WordPress.

Restricciones de la tabla `wp_terms`

- **term_id:** Clave primaria (PRIMARY KEY) y autoincremento (AUTO_INCREMENT). Esta restricción garantiza que cada término de taxonomía tenga un ID único y que se genere automáticamente para cada nuevo término.
 - **name:** No permite valores nulos (NOT NULL). Esto asegura que cada término tenga un nombre asociado y que no pueda quedar sin asignar.
 - **slug:** No permite valores nulos (NOT NULL). Esto asegura que cada término tenga un slug (identificador único) asociado y que no pueda quedar sin asignar.
 - **term_group:** Puede permitir valores nulos o estar vacío dependiendo de la configuración. Este campo se utiliza para agrupar términos relacionados, pero su uso puede ser opcional, por lo que no se impone una restricción estricta.
-

wp_term_relationships

La tabla **wp_term_relationships** en WordPress se utiliza para establecer relaciones entre entradas o páginas y términos de taxonomía, como categorías y etiquetas. Cada fila de esta tabla representa una relación entre una entrada o página y un término de taxonomía específico. Estas relaciones permiten organizar y clasificar el contenido del sitio web de manera efectiva.

Campos principales de la tabla wp_term_relationships:

- **object_id:** Este campo almacena el ID del objeto al que se aplica la relación. En la mayoría de los casos, se refiere al ID de una entrada o página en la tabla **wp_posts**. Es un número entero largo (*bigint*) con un ancho de 20 dígitos. Este campo establece una relación con la entrada o página asociada.
- **term_taxonomy_id:** El campo *term_taxonomy_id* almacena el ID de la taxonomía del término relacionado. Es un número entero largo (*bigint*) con un ancho de 20 dígitos. Este campo establece una relación con la tabla **wp_term_taxonomy**, que contiene información detallada sobre la taxonomía a la que pertenece el término.
- **term_order:** El campo *term_order* almacena el orden de la relación entre el objeto y el término de taxonomía. Es un número entero (*int*) que indica la posición relativa del término en relación con otros términos asociados al mismo objeto.

Relaciones de la tabla wp_term_relationships

- Relación con **wp_posts** (Campo *object_id*):
 - El campo *object_id* en la tabla **wp_term_relationships** se utiliza para relacionar cada relación con los contenidos almacenados en la tabla **wp_posts**. Este campo actúa como clave foránea en **wp_term_relationships** y se relaciona con la clave primaria ID en la tabla **wp_posts**.
 - Ejemplo: Supongamos que tenemos una relación en **wp_term_relationships** con *object_id* igual a 42 y *term_taxonomy_id* igual a 5. Esto indica que el contenido con ID igual a 42 está relacionado con la etiqueta o término de taxonomía que tiene *term_taxonomy_id* igual a 5. La tabla **wp_posts** contiene información sobre este contenido específico.
 - Uso: Esta relación permite vincular etiquetas o términos de taxonomía a contenidos específicos en **wp_posts**, lo que facilita la asignación de etiquetas a las entradas y páginas en WordPress y permite a los usuarios organizar y buscar contenidos relacionados con etiquetas específicas.
- Relación con **wp_term_taxonomy** (Campo *term_taxonomy_id*):
 - El campo *term_taxonomy_id* en la tabla **wp_term_relationships** se utiliza para relacionar cada relación con la información detallada de la taxonomía en la tabla **wp_term_taxonomy**. Este campo actúa como clave foránea en **wp_term_relationships** y clave primaria en **wp_term_taxonomy**.

- Ejemplo: Supongamos que tenemos una relación en **wp_term_relationships** con *term_taxonomy_id* igual a 5 y una entrada con *object_id* igual a 42 relacionada con esta relación. Esto indica que la entrada está relacionada con la taxonomía que tiene *term_taxonomy_id* igual a 5, y la tabla **wp_term_taxonomy** contiene información adicional sobre esta taxonomía.
- Uso: Esta relación permite vincular cada relación entre objetos y términos de taxonomía con información detallada sobre las taxonomías en **wp_term_taxonomy**, lo que facilita la gestión y organización de los términos de taxonomía en WordPress.

Restricciones de la tabla wp_term_relationships

- **object_id:** Clave primaria (PRIMARY KEY) junto con *term_taxonomy_id*. Esto garantiza que cada relación entre un objeto y un término de taxonomía sea única y que se genere automáticamente para cada nueva relación.
- **term_taxonomy_id:** Clave primaria (PRIMARY KEY) junto con *object_id*. Esto garantiza que cada relación entre un objeto y un término de taxonomía sea única y que se genere automáticamente para cada nueva relación.
- **term_order:** No permite valores nulos (NOT NULL). Esto asegura que cada relación tenga un orden asociado y que no pueda quedar sin asignar. El orden se utiliza para determinar la posición relativa de un término en relación con otros términos asociados al mismo objeto.

wp_term_taxonomy

La tabla **wp_term_taxonomy** en WordPress almacena información sobre las taxonomías utilizadas en el sitio web, como categorías y etiquetas. Cada fila de esta tabla representa una taxonomía específica con detalles sobre su estructura y uso en el sitio.

Campos principales de la tabla wp_term_taxonomy:

- **term_taxonomy_id:** Este campo es la clave primaria (PRIMARY KEY) de la tabla y se utiliza para identificar de manera única cada entrada de taxonomía en WordPress. Es un número entero largo (*bigint*) con un ancho de 20 dígitos. Se genera automáticamente para cada nueva entrada de taxonomía y se incrementa secuencialmente, lo que garantiza que cada entrada de taxonomía tenga su propio ID único.
- **term_id:** El campo *term_id* almacena el ID del término de taxonomía relacionado. Es un número entero largo (*bigint*) con un ancho de 20 dígitos. Este campo establece una relación con la tabla **wp_terms**, que contiene información sobre los términos de taxonomía.
- **taxonomy:** El campo *taxonomy* almacena el nombre de la taxonomía, que puede ser una categoría, etiqueta u otra taxonomía personalizada. Es una cadena de texto (*varchar*) con un máximo de 32 caracteres y utiliza el conjunto de caracteres utf8mb4_unicode_ci. Este campo identifica el tipo de taxonomía a la que pertenece la entrada.
- **description:** El campo *description* almacena una descripción de la taxonomía. Es de tipo *longtext* y utiliza el conjunto de caracteres utf8mb4_unicode_ci. Este campo proporciona información adicional sobre la taxonomía, como su propósito o contexto.
- **parent:** El campo *parent* almacena el ID del término de taxonomía padre, si existe. Es un número entero largo (*bigint*) con un ancho de 20 dígitos. Este campo establece relaciones jerárquicas entre términos de taxonomía, lo que permite crear estructuras de categorización más complejas.
- **count:** El campo *count* almacena el recuento de objetos (entradas, páginas, etc.) asociados a la taxonomía. Es un número entero largo (*bigint*) que indica cuántas veces se ha utilizado esta taxonomía en objetos del sitio.

Relaciones de la tabla wp_term_taxonomy:

- Relación con **wp_terms** (Campo *term_id*):

- El campo *term_id* en la tabla **wp_term_taxonomy** se utiliza para relacionar cada entrada de taxonomía con el término de taxonomía correspondiente en la tabla **wp_terms**. Este campo actúa como clave primaria en **wp_term_taxonomy** y clave foránea en **wp_terms**.
- Ejemplo: Supongamos que tenemos una entrada de taxonomía con *term_taxonomy_id* igual a 8 y *term_id* igual a 5. Esto indica que la entrada de taxonomía está asociada al término de taxonomía que tiene *term_id* igual a 5 en la tabla **wp_terms**.
- Uso: Esta relación permite vincular cada entrada de taxonomía en **wp_term_taxonomy** con información más detallada sobre los términos en **wp_terms**, lo que facilita la organización y clasificación de taxonomías en WordPress.
- Relación con **wp_term_relationships** (Campo *term_taxonomy_id*):
 - El campo *term_taxonomy_id* en la tabla **wp_term_taxonomy** se utiliza para relacionar cada entrada de taxonomía con los objetos (entradas, páginas, etc.) a los que está asociada. Este campo actúa como clave primaria en **wp_term_taxonomy** y clave foránea en **wp_term_relationships**.
 - Ejemplo: Supongamos que tenemos una entrada de taxonomía con *term_taxonomy_id* igual a 8 y existe un registro en **wp_term_relationships** donde *term_taxonomy_id* también es igual a 8. Esto indica que la entrada de taxonomía está asociada a un objeto específico en el sitio, como una entrada o página.
 - Uso: Esta relación permite vincular taxonomías en **wp_term_taxonomy** con objetos en **wp_term_relationships**, lo que facilita la clasificación y organización de objetos mediante taxonomías en WordPress.
- Relación jerárquica con otros términos (Campo *parent*):
 - El campo *parent* en la tabla **wp_term_taxonomy** se utiliza para establecer relaciones jerárquicas entre términos de taxonomía. Cuando se asigna un valor en este campo, indica que la entrada de taxonomía actual es un término secundario o hijo del término con el ID especificado en *parent*.
 - Ejemplo: Supongamos que tenemos un término de taxonomía con *term_id* igual a 10 y *parent* igual a 5. Esto indica que el término con *term_id* 10 es un término secundario o hijo del término con *term_id* 5.
 - Uso: Esta relación jerárquica se utiliza para crear estructuras de categorización en forma de árbol, donde un término puede ser subordinado a otro.

Restricciones de la tabla **wp_term_taxonomy**:

- **term_taxonomy_id**: Clave primaria (PRIMARY KEY) y autoincremento (AUTO_INCREMENT). Esta restricción garantiza que cada entrada de taxonomía tenga un ID único y que se genere automáticamente para cada nueva entrada.
 - **term_id**: No permite valores nulos (NOT NULL). Esto asegura que cada entrada de taxonomía esté asociada a un término de taxonomía válido en WordPress y no pueda quedar sin asignar.
 - **taxonomy**: No permite valores nulos (NOT NULL). Esto asegura que cada entrada de taxonomía tenga un nombre de taxonomía válido y que no pueda quedar sin asignar.
 - **description**: Puede permitir valores nulos o estar vacío dependiendo de la configuración. Este campo es opcional y se utiliza para proporcionar una descripción adicional de la taxonomía.
 - **parent**: No permite valores nulos (NOT NULL). Esto asegura que las relaciones jerárquicas entre términos de taxonomía estén correctamente definidas y que no pueda quedar sin asignar.
 - **count**: No permite valores nulos (NOT NULL). Esto asegura que el recuento de objetos asociados a la taxonomía esté actualizado y que no pueda quedar sin asignar.
-

wp_usermeta

La tabla **wp_usermeta** en WordPress se utiliza para almacenar metadatos relacionados con los usuarios registrados en el sitio web. Se enfoca en metadatos de usuario. Esta tabla es esencial para almacenar información personalizada sobre los usuarios, como preferencias, configuraciones y datos adicionales.

Campos principales de la tabla wp_usermeta:

- **umeta_id:** Este campo es la clave primaria (PRIMARY KEY) de la tabla y se utiliza para identificar de manera única cada registro de metadatos de usuario en WordPress. Es un número entero largo (*bigint*) con un ancho de 20 dígitos. Se genera automáticamente para cada nuevo registro y se incrementa secuencialmente, lo que garantiza que cada metadato tenga su propio ID único.
- **user_id:** El campo *user_id* almacena el ID del usuario al que está asociado el metadato. Es otro número entero largo (*bigint*) con un ancho de 20 dígitos. Este campo establece una relación con la tabla **wp_users**, lo que indica a qué usuario pertenece el metadato.
- **meta_key:** El campo *meta_key* almacena la clave o nombre del metadato de usuario. Es una cadena de texto (*varchar*) con un máximo de 255 caracteres y utiliza el conjunto de caracteres *utf8mb4_unicode_ci*. Cada clave de metadato es única dentro del contexto de un usuario específico y se utiliza para identificar y buscar metadatos relacionados con ese usuario.
- **meta_value:** El campo *meta_value* almacena el valor del metadato de usuario. Es de tipo *longtext* y utiliza el conjunto de caracteres *utf8mb4_unicode_ci*. Puede contener cualquier tipo de información relacionada con el usuario, como cadenas de texto largas, datos serializados o información estructurada.

Relaciones de la tabla wp_usermeta

- Relación con la tabla **wp_users**
 - El campo *user_id* en la tabla **wp_usermeta** se utiliza para relacionar cada registro de metadatos de usuario con la tabla **wp_users**. Este campo actúa como clave foránea (FOREIGN KEY) que referencia el ID de usuario en la tabla **wp_users**.
 - Ejemplo: Supongamos que tenemos un registro en **wp_usermeta** con *user_id* igual a 5 y *meta_key* igual a "email" y *meta_value* igual a "lazaroperez@correo.com". La relación indica que este metadato pertenece al usuario con ID igual a 5 en la tabla **wp_users**, y que representa su dirección de correo electrónico.
 - Uso: Esta relación permite asociar información personalizada o metadatos adicionales a usuarios específicos registrados en el sitio de WordPress. Por ejemplo, se pueden almacenar datos como direcciones de correo electrónico, nombres de usuario, preferencias de notificaciones, roles personalizados, y más en la tabla **wp_usermeta** para mejorar la funcionalidad del sitio y personalizar la experiencia de usuario.

Restricciones de la tabla wp_usermeta:

- **umeta_id:** Clave primaria (PRIMARY KEY) y autoincremento (AUTO_INCREMENT). Esta restricción garantiza que cada registro de metadatos tenga un ID único y que se genere automáticamente para cada nuevo registro.
 - **user_id:** No permite valores nulos (NOT NULL). Esto asegura que cada metadato esté asociado a un usuario específico en WordPress y no pueda quedar sin asignar.
 - **meta_key:** Puede permitir valores nulos o estar vacío dependiendo de la configuración. No se impone una restricción de unicidad en este campo, lo que permite que múltiples registros tengan la misma clave de metadato si es necesario.
 - **meta_value:** Puede permitir valores nulos o estar vacío dependiendo de la configuración. Esto permite que el valor del metadato pueda ser opcional en algunos casos, pero debe seguir las restricciones de tipo de datos (*longtext*).
-

wp_users

La tabla **wp_users** en WordPress almacena información sobre los usuarios registrados en el sitio web. Cada fila de esta tabla representa un usuario único con un identificador único (ID). Estos usuarios pueden ser administradores, autores, editores o suscriptores, según sus roles en el sitio.

Campos principales de la tabla wp_users:

- **ID:** Este campo es la clave primaria (PRIMARY KEY) de la tabla y se utiliza para identificar de manera única a cada usuario en WordPress. Es un número entero largo (bigint) con un ancho de 20 dígitos. Se genera automáticamente para cada nuevo usuario y se incrementa secuencialmente, lo que garantiza que cada usuario tenga su propio ID único.
- **user_login:** El campo *user_login* almacena el nombre de usuario con el que un usuario inicia sesión en el sitio. Es una cadena de texto (*varchar*) con un máximo de 60 caracteres y utiliza el conjunto de caracteres utf8mb4_unicode_ci. Cada usuario debe tener un nombre de usuario único, y este campo se utiliza para la autenticación.
- **user_pass:** El campo *user_pass* almacena la contraseña del usuario de manera segura. Es una cadena de texto (*varchar*) con un máximo de 255 caracteres y utiliza el conjunto de caracteres utf8mb4_unicode_ci. La contraseña se almacena de forma encriptada o hash, lo que garantiza la seguridad de las contraseñas de los usuarios.
- **user_nicename:** El campo *user_nicename* almacena un nombre único y amigable para URL del usuario. Es una cadena de texto (*varchar*) con un máximo de 50 caracteres y utiliza el conjunto de caracteres utf8mb4_unicode_ci. Este valor se utiliza en las URL del perfil del usuario.
- **user_email:** El campo *user_email* almacena la dirección de correo electrónico del usuario. Es una cadena de texto (*varchar*) con un máximo de 100 caracteres y utiliza el conjunto de caracteres utf8mb4_unicode_ci. La dirección de correo electrónico se utiliza para fines de comunicación y notificaciones.
- **user_url:** El campo *user_url* almacena la URL del sitio web o el enlace del perfil del usuario. Es una cadena de texto (*varchar*) con un máximo de 100 caracteres y utiliza el conjunto de caracteres utf8mb4_unicode_ci. Los usuarios pueden proporcionar un enlace a su sitio web personal si lo desean.
- **user_registered:** El campo *user_registered* registra la fecha y hora en que se registró el usuario en el sitio. Es de tipo datetime y no permite valores nulos. Esto permite llevar un registro de cuándo se unió cada usuario al sitio.
- **user_activation_key:** El campo *user_activation_key* se utiliza para almacenar una clave de activación temporal cuando se requiere que los usuarios activen sus cuentas por correo electrónico. Es una cadena de texto (*varchar*) con un máximo de 255 caracteres y utiliza el conjunto de caracteres utf8mb4_unicode_ci.
- **user_status:** El campo *user_status* almacena el estado del usuario, que puede indicar si la cuenta está activa o inactiva. Es un número entero (*int*) y no permite valores nulos. Los valores comunes son 0 para cuentas inactivas y 1 para cuentas activas.
- **display_name:** El campo *display_name* almacena el nombre que se muestra públicamente para el usuario en el sitio web. Es una cadena de texto (*varchar*) con un máximo de 250 caracteres y utiliza el conjunto de caracteres utf8mb4_unicode_ci. Este nombre se utiliza para mostrar la autoría de las publicaciones y comentarios.

Relaciones de la tabla wp_users:

- Relación con **wp_usermeta** (Campo *user_id*):
 - El campo ID en la tabla **wp_users** se utiliza para relacionar cada usuario con metadatos personalizados almacenados en la tabla **wp_usermeta**. El campo *user_id* en la tabla **wp_usermeta** actúa como una clave foránea que se relaciona con el ID del usuario en **wp_users**.

- Ejemplo: Supongamos que tenemos un usuario con ID 1 llamado "usuarioABC" en **wp_users**. Podemos encontrar registros relacionados en **wp_usermeta** con *user_id* igual a 1, lo que indica que estos registros de metadatos personalizados pertenecen a "usuarioABC".
- Uso: Esta relación permite asociar metadatos personalizados a usuarios específicos, lo que puede ser útil para almacenar información adicional o configuraciones personalizadas para cada usuario.
- Relación con **wp_comments** (Campo *user_id*):
 - El campo ID en la tabla **wp_users** también se utiliza para relacionar cada usuario con los comentarios que han realizado en el sitio. En la tabla **wp_comments**, el campo *user_id* actúa como una clave foránea que se relaciona con el ID del usuario en **wp_users**.
 - Ejemplo: Si un usuario con ID 2 ha realizado un comentario en una publicación, en la tabla **wp_comments** habrá un registro donde el campo *user_id* sea igual a 2, lo que indica que ese comentario fue realizado por el usuario con ID 2.
 - Uso: Esta relación permite asociar comentarios a usuarios específicos, lo que facilita la identificación de quién hizo cada comentario en el sitio.
- Relación con **wp_posts** (Campo *post_author*):
 - El campo ID en la tabla **wp_users** también se utiliza para relacionar cada usuario con las publicaciones que han creado en el sitio. En la tabla **wp_posts**, el campo *post_author* actúa como una clave foránea que se relaciona con el ID del usuario en **wp_users**.
 - Ejemplo: Si un usuario con ID 3 ha creado una publicación, en la tabla **wp_posts** habrá un registro donde el campo *post_author* sea igual a 3, lo que indica que esa publicación fue creada por el usuario con ID 3.
 - Uso: Esta relación permite asociar publicaciones a usuarios específicos, lo que facilita la identificación de quién es el autor de cada publicación en el sitio.
- Relación con la tabla **wp_links** (Campo *link_owner*):
 - El campo ID en la tabla **wp_users** se utiliza para relacionar cada usuario con los enlaces que ha creado. En la tabla **wp_links**, el campo *link_owner* almacena el ID del usuario que es propietario de un enlace.
 - Ejemplo: Si tenemos un usuario en **wp_users** con ID igual a 15, y ese usuario ha creado un enlace en **wp_links**, entonces en la tabla **wp_links**, el campo *link_owner* contendrá el valor 15 para ese enlace, indicando que el usuario con ID 15 es el propietario de ese enlace.
 - Uso: Esta relación permite asociar enlaces a usuarios específicos en WordPress. Los usuarios pueden crear y gestionar sus propios enlaces, lo que puede ser útil para administrar recursos externos o referencias relacionadas con el sitio web.

Restricciones de la tabla **wp_users**

- **term_id**: Clave primaria (PRIMARY KEY) y autoincremento (AUTO_INCREMENT). Esta restricción garantiza que cada término de taxonomía tenga un ID único y que se genere automáticamente para cada nuevo término.
- **name**: No permite valores nulos (NOT NULL). Esto asegura que cada término tenga un nombre asociado y que no pueda quedar sin asignar.
- **slug**: No permite valores nulos (NOT NULL). Esto asegura que cada término tenga un slug (identificador único) asociado y que no pueda quedar sin asignar.
- **term_group**: Puede permitir valores nulos o estar vacío dependiendo de la configuración. Este campo se utiliza para agrupar términos relacionados, pero su uso puede ser opcional, por lo que no se impone una restricción estricta.

La instalación de algunos plugins pueden añadir más tablas a la base de datos, como por ejemplo el plugin WP RSS Aggregator, que añade la tabla `wp_wprss_logs`

The diagram illustrates the WordPress database schema, showing the relationships between various tables. The tables and their fields are as follows:

- wp_usermeta**:
 - umeta_id BIGINT(20)
 - user_id BIGINT(20)
 - meta_key VARCHAR(255)
 - meta_value LONGTEXT
- wp_users**:
 - ID BIGINT(20)
 - user_login VARCHAR(60)
 - user_pass VARCHAR(255)
 - user_nicename VARCHAR(50)
 - user_email VARCHAR(100)
 - user_url VARCHAR(100)
 - user_registered DATETIME
 - user_activation_key VARCHAR(255)
 - user_status INT(11)
 - display_name VARCHAR(250)
- wp_options**:
 - option_id BIGINT(20)
 - option_name VARCHAR(191)
 - option_value LONGTEXT
 - autoload VARCHAR(20)
- wp_comments**:
 - comment_ID BIGINT(20)
 - comment_post_ID BIGINT(20)
 - comment_author TINYTEXT
 - comment_author_email VARCHAR(100)
 - comment_author_url VARCHAR(200)
 - comment_author_IP VARCHAR(100)
 - comment_date DATETIME
- wp_posts**:
 - ID BIGINT(20)
 - post_author BIGINT(20)
 - post_date DATETIME
 - post_date_gmt DATETIME
 - post_content LONGTEXT
 - post_title TEXT
 - post_excerpt TEXT
 - post_status VARCHAR(20)
 - comment_status VARCHAR(20)
- wp_links**:
 - link_id BIGINT(20)
 - link_url VARCHAR(255)
 - link_name VARCHAR(255)
 - link_image VARCHAR(255)
 - link_target VARCHAR(25)
 - link_description VARCHAR(255)
 - link_visible VARCHAR(20)
 - link_owner BIGINT(20)
 - link_rating INT(11)
 - link_updated DATETIME
 - link_rel VARCHAR(255)
 - link_notes MEDIUMTEXT
 - link_rss VARCHAR(255)
- wp_term_relationships**:
 - object_id BIGINT(20)
 - term_taxonomy_id BIGINT(20)
 - term_order INT(11)
- wp_term_taxonomy**:
 - term_taxonomy_id BIGINT(20)
 - term_id BIGINT(20)
 - taxonomy VARCHAR(32)
 - description LONGTEXT
 - parent BIGINT(20)
 - count BIGINT(20)
- wp_terms**:
 - term_id BIGINT(20)
 - name VARCHAR(200)
 - slug VARCHAR(200)
 - term_group BIGINT(10)
- wp_commentmeta**:
 - meta_id BIGINT(20)
 - comment_id BIGINT(20)
 - meta_key VARCHAR(255)
 - meta_value LONGTEXT
- wp_postmeta**:
 - meta_id BIGINT(20)
 - post_id BIGINT(20)
 - meta_key VARCHAR(255)
 - meta_value LONGTEXT
- wp_termmeta**:
 - meta_id BIGINT(20)
 - term_id BIGINT(20)
 - meta_key VARCHAR(255)
 - meta_value LONGTEXT

The diagram shows the following relationships:

- wp_users** is the primary key for **wp_usermeta** (user_id).
- wp_posts** is the primary key for **wp_comments** (comment_post_ID).
- wp_posts** is the primary key for **wp_postmeta** (post_id).
- wp_comments** is the primary key for **wp_commentmeta** (comment_id).
- wp_posts** is the primary key for **wp_term_relationships** (object_id).
- wp_term_taxonomy** is the primary key for **wp_terms** (term_id).
- wp_term_taxonomy** is the primary key for **wp_termmeta** (term_id).
- wp_term_taxonomy** is the primary key for **wp_links** (link_owner).
- wp_term_taxonomy** is the primary key for **wp_term_relationships** (term_taxonomy_id).
- wp_term_relationships** is the primary key for **wp_links** (link_id).

c. Descripción de cada tabla en formato tabla

Table 1. *wp_commentmeta*

Nombre del campo	Tipo	Longitud	Descripción	Uso
meta_id	bigint	20	Clave primaria de la tabla, generada automáticamente, se incrementa secuencialmente.	Identificación única de cada registro de metadato de comentario.
comment_id	bigint	20	Almacena el ID del comentario al que está asociado el metadato. No permite valores nulos y tiene un valor predeterminado de 0.	Establece una relación con la tabla wp_comments para identificar a qué comentario pertenece el metadato.
meta_key	varchar	255	Almacena el nombre o clave del metadato del comentario. Puede permitir valores nulos o estar vacío dependiendo de la configuración.	Permite identificar y buscar metadatos específicos del comentario.
meta_value	longtext	-	Almacena el valor del metadato del comentario. Puede permitir valores nulos o estar vacío dependiendo de la configuración.	Almacena información relacionada con el comentario, que puede ser de cualquier tipo.

Table 2. *wp_comments*

Nombre del campo	Tipo	Longitud	Descripción	Uso
comment_ID	bigint	-	Clave primaria de la tabla, generada automáticamente, se incrementa secuencialmente.	Identifica de manera única cada comentario.
comment_post_ID	bigint	-	ID de la entrada o página a la que se refiere el comentario.	Establece una relación con la tabla wp_posts , indicando a qué entrada o página pertenece el comentario.
comment_author	tinytext	255	Nombre del autor del comentario.	Identifica mediante el nombre al autor del comentario.
comment_author_email	varchar	100	Dirección de correo electrónico del autor del comentario.	Almacena la dirección de correo electrónico del autor del comentario si se proporciona.
comment_author_url	varchar	200	URL del sitio web, blog o perfil en redes sociales del autor del comentario si se proporciona.	Almacena el sitio web del autor del comentario si se proporciona.
comment_author_IP	varchar	100	Dirección IP del autor del comentario.	Registra la dirección IP del autor del comentario.
comment_date	datetime	-	Fecha y hora en que se realizó el comentario.	Indica la fecha y hora en que se realizó el comentario.
comment_date_gmt	datetime	-	Fecha y hora del comentario en formato GMT (Tiempo Medio de Greenwich).	Almacena la fecha y hora del comentario en formato GMT.

Nombre del campo	Tipo	Longitud	Descripción	Uso
comment_content	text	-	Contenido del comentario en sí.	Almacena el contenido del comentario.
comment_karma	int	11	Registro del "karma" o calificación	Se usa para calificar los comentarios. El karma es una característica que permite a los usuarios y administradores del sitio otorgar una calificación o puntuación a los comentarios que se realizan en las publicaciones. Esta función se utiliza para dar a los comentarios una valoración positiva o negativa según su calidad o relevancia.
comment_approved	varchar	20	Indica si el comentario ha sido aprobado o se encuentra en espera de moderación.	Registra el estado del comentario (aprobado, spam, papelera, etc.).
comment_agent	varchar	255	Información sobre el agente de usuario o navegador web que se utilizó para enviar el comentario.	Registra el agente de usuario del navegador.
comment_type	varchar	20	Categoría del tipo de comentario.	Categoriza el tipo de comentario, por ejemplo, si es un comentario estándar, un pingback o un trackback.

Nombre del campo	Tipo	Longitud	Descripción	Uso
comment_parent	bigint	-	Relación jerárquica entre comentarios.	Establece relaciones jerárquicas entre comentarios, indicando cuál es el comentario padre de un comentario específico..
user_id	bigint	-	Si el comentario ha sido realizado por un usuario registrado, este campo almacena el ID de usuario correspondiente. De lo contrario, se establece en 0.	Asocia el comentario a un usuario registrado, si corresponde.

Table 3. *wp_links*

Nombre del campo	Tipo	Longitud	Descripción	Uso
link_id	bigint	20	Clave primaria que identifica de manera única cada enlace en WordPress. Se genera automáticamente para cada nuevo enlace y se incrementa secuencialmente, lo que garantiza que cada enlace tenga su propio ID único.	Identifica de manera única cada enlace.
link_url	varchar	255	URL a la que se redirige el enlace y utiliza el conjunto de caracteres utf8mb4_unicode_ci.	Define la ubicación de destino del enlace.
link_name	varchar	255	Nombre o título del enlace, utiliza utf8mb4_unicode_ci.	Proporciona una descripción legible para el enlace.
link_image	varchar	255	Puede almacenar una URL de imagen relacionada con el enlace y utiliza utf8mb4_unicode_ci.	A
link_target	varchar	25	Indica la forma en que se abrirá el enlace. Los valores típicos pueden ser " <i>blank</i> " para abrir en una nueva ventana o " <i>self</i> " para abrir en la misma ventana.	Define cómo se abrirá el enlace.
link_description	varchar	255	Permite agregar una breve descripción del enlace, utiliza utf8mb4_unicode_ci.	Proporciona información adicional sobre el enlace.

Nombre del campo	Tipo	Longitud	Descripción	Uso
link_visible	varchar	20	Indica si el enlace es visible en el sitio web, utiliza utf8mb4_unicode_ci.	Controla la visibilidad de los enlaces en el sitio.
link_owner	bigint	-	Almacena el ID del usuario que es propietario del enlace.	Atribuye la propiedad del enlace a un usuario específico, relacionándose con la tabla wp_users .
link_rating	int	-	Permite asignar una valoración o puntuación al enlace. No permite valores nulos	Se utiliza para clasificar o calificar los enlaces.
link_updated	datetime	-	Almacena la fecha y hora de la última actualización del enlace.	Rastrea los cambios en el enlace.
link_rel	varchar	255	Puede contener información sobre la relación del enlace con otros recursos, utiliza utf8mb4_unicode_ci.	Establece relaciones específicas con otros enlaces o recursos.
link_notes	mediumtext	-	Permite agregar notas adicionales o comentarios sobre el enlace, utiliza utf8mb4_unicode_ci.	Proporciona información detallada sobre el enlace.
link_rss	varchar	255	Almacena la dirección URL del feed RSS asociado con el enlace, si corresponde.	Conecta el enlace con un feed RSS relacionado.

Table 4. *wp_options*

Nombre del campo	Tipo	Longitud	Descripción	Uso
option_id	bigint	20	Clave primaria que identifica de manera única cada opción en WordPress. Se genera automáticamente para cada nueva opción y se incrementa secuencialmente.	Identifica de manera única cada opción.
option_name	varchar	191	Almacena el nombre de la opción y utiliza el conjunto de caracteres utf8mb4_unicode_ci. No permite valores nulos.	Permite identificar y acceder a cada opción de manera única.
option_value	longtext	-	Almacena el valor asociado con la opción. Puede contener datos de diversos tipos, como texto, números o incluso estructuras de datos más complejas. No permite valores nulos.	Guarda información o configuración específica correspondiente a cada opción.
autoload	varchar	20	Indica si la opción debe cargarse automáticamente al iniciar WordPress. utiliza el conjunto de caracteres utf8mb4_unicode_ci. Los valores típicos son "yes" para cargar automáticamente y "no" para no cargar automáticamente. No permite valores nulos.	Determina si la opción se carga en la memoria cuando WordPress arranca.

Table 5. *wp_postmeta*

NOMBRE del campo	Tipo	Longitud	Descripción	Uso
meta_id	bigint(20)	-	Clave primaria que se genera automáticamente para cada nuevo registro y se incrementa secuencialmente	Identifica de manera única cada metadato de entrada o página en WordPress.
post_id	bigint(20)	-	Almacena el ID de la entrada o página a la que está asociado el metadato.	Establece una relación con la tabla wp_posts , lo que indica a qué entrada o página pertenece el metadato..
meta_key	varchar(255)	255	Almacena el nombre o clave del metadato de la entrada o página. Utiliza el conjunto de caracteres utf8mb4_unicode_ci. Puede permitir valores nulos.	Identifica y busca metadatos específicos de la entrada o página.
meta_value	longtext	-	Almacena el valor del metadato de la entrada o página. Utiliza el conjunto de caracteres utf8mb4_unicode_ci. Puede permitir valores nulos.	Guarda información personalizada relacionada con la entrada o página, como calificaciones, etiquetas, resúmenes, o cualquier dato personalizado necesario para el sitio web..

Table 6. *wp_posts*

Nombre del campo	Tipo	Longitud	Descripción	Uso
ID	bigint	20	Clave primaria que se genera automáticamente para cada nueva entrada y se incrementa secuencialmente	Identifica de manera única cada entrada o página.
post_author	bigint	20	Almacena el ID del usuario que creó la entrada o página.	Establece una relación con la tabla wp_users para identificar al autor.
post_date	datetime	-	Almacena la fecha y hora exactas en que se creó la entrada o página. La fecha y hora se registran en el formato "YYYY-MM-DD HH:MM:SS".	Registra la fecha de creación y facilita la ordenación de las entradas y páginas por fecha.
post_date_gmt	datetime	-	Almacena la fecha y hora de creación en formato GMT (Hora Media de Greenwich).	Registra la fecha de creación en GMT para asegurar que las fechas sean consistentes y no dependan de la zona horaria del servidor..
post_content	longtext	-	Contiene el contenido principal de la entrada o página. Puede contener una gran cantidad de texto y otros elementos, como imágenes y etiquetas HTML. Utiliza el conjunto de caracteres utf8mb4_unicode_ci para admitir caracteres especiales y emojis.	Almacena el contenido visible en el sitio web.

Nombre del campo	Tipo	Longitud	Descripción	Uso
post_title	text	-	Almacena el título de la entrada o página y utiliza el conjunto de caracteres utf8mb4_unicode_ci.	Almacena el título.
post_excerpt	text	-	Almacena un extracto o resumen de la entrada, si se proporciona, y utiliza el conjunto de caracteres utf8mb4_unicode_ci.	Almacena el extracto o resumen de la entrada, si se proporciona.
post_status	varchar	20	Indica el estado de la entrada, como "publish" (publicado), "draft" (borrador), "private" (privado), entre otros. Su valor predeterminado es " <i>publish</i> " (publicado).	Indica el estado de la entrada.
comment_status	varchar	20	Indica el estado de los comentarios para la entrada, como "open" (abierto) o "closed" (cerrado). Su valor predeterminado es " <i>open</i> " (abierto).	Indica el estado de los comentarios.
ping_status	varchar	20	Indica el estado de pings y trackbacks para la entrada, como "open" (abierto) o "closed" (cerrado) y su valor predeterminado es " <i>open</i> " (abierto).	Indica el estado de pings y trackbacks, utilizados en sistemas de blogs y sitios web para notificar a otros sitios web cuando se ha enlazado a su contenido.

Nombre del campo	Tipo	Longitud	Descripción	Uso
post_password	varchar	255	Permite establecer una contraseña para proteger el acceso a la entrada. Es opcional	Almacena la contraseña.
post_name	varchar	200	Almacena un slug (URL amigable) generado automáticamente a partir del título.	Almacena el slug.
to_ping	text	-	Almacena una lista de URLs a las que se enviarán pingbacks cuando se publique la entrada.	Lista de URLs para pingbacks, se utiliza en procesos de notificación a otros sitios web cuando se enlaza a ellos desde la entrada.
pinged	text	-	Almacena una lista de URLs de sitios web que han enviado pingbacks a esta entrada. Se actualiza automáticamente cuando se reciben pingbacks.	Lista de URLs de pingbacks recibidos.
post_modified	datetime	-	Almacena la fecha y hora de la última modificación de la entrada o página.	Registra la fecha de la última modificación. Se utiliza para rastrear cambios en el contenido.
post_modified_gmt	datetime	-	Almacena la fecha y hora de la última modificación en formato GMT (Hora Media de Greenwich).	Registra la fecha de la última modificación en GMT.

Nombre del campo	Tipo	Longitud	Descripción	Uso
post_content_filtered	longtext	-	Almacena una versión filtrada del contenido de la entrada.	Versión filtrada del contenido, generalmente se utiliza para optimizar o modificar el contenido original de alguna manera específica para ciertos propósitos o plugins.
post_parent	bigint	20	Establece la relación entre entradas principales y sus hijos o páginas secundarias.	Define la jerarquía. Se utiliza para establecer la jerarquía de las páginas.
guid	varchar	255	Almacena una URL única globalmente identificable.	Almacena la URL única globalmente identificable que se utiliza principalmente en procesos de importación y exportación de contenidos.
menu_order	int	11	Orden en menús.	Se utiliza para establecer el orden en que las entradas aparecen en los menús personalizados.
post_type	varchar	20	Indica el tipo de contenido, como "post" (entrada) o "page" (página). Su valor predeterminado es "post".	Indica el tipo de contenido.
post_mime_type	varchar	100	Almacena el tipo de archivo asociado a la entrada, si corresponde.	Indica el tipo de archivo asociado.

Nombre del campo	Tipo	Longitud	Descripción	Uso
comment_count	bigint	20	Número de comentarios asociados, se actualiza automáticamente cuando se agregan o eliminan comentarios..	Almacena el número de comentarios asociados a la entrada.

Table 7. *wp_termmeta*

Nombre del campo	Tipo	Longitud	Descripción	Uso
meta_id	bigint	20	Clave primaria que identifica de manera única cada registro de metadato de término en WordPress. Se genera automáticamente para cada nuevo registro y se incrementa secuencialmente.	Identifica de manera única cada metadato de término.
term_id	bigint	20	Almacena el ID del término de taxonomía al que está asociado el metadato.	Establece una relación con la tabla wp_terms para identificar a qué término pertenece el metadato.
meta_key	varchar	255	Almacena el nombre o clave del metadato del término y utiliza el conjunto de caracteres utf8mb4_unicode_ci. Puede permitir valores nulos o estar vacío dependiendo de la configuración.	Permite identificar y buscar metadatos específicos del término.
meta_value	longtext	-	Almacena el valor del metadato del término. Puede permitir valores nulos o estar vacío dependiendo de la configuración.	Almacena información personalizada relacionada con el término de taxonomía.

Table 8. *wp_terms*

Nombre del campo	Tipo	Longitud	Descripción	Uso
term_id	bigint	20	Clave primaria que identifica de manera única cada término de taxonomía en WordPress. Se genera automáticamente para cada nuevo término y se incrementa secuencialmente.	Identifica de manera única cada término de taxonomía.
name	varchar	200	Almacena el nombre del término de taxonomía. Utiliza el conjunto de caracteres utf8mb4_unicode_ci.	Almacena el nombre descriptivo del término.
slug	varchar	200	Almacena la versión amigable para URL del término y utiliza el conjunto de caracteres utf8mb4_unicode_ci. suele estar en minúsculas y separado por guiones, por ejemplo, "noticias", "deportes", "tecnologia", "cine", etc.	Utilizado en las URLs del sitio web.
term_group	bigint	10	Se utiliza para agrupar términos relacionados, pero su uso es menos común en versiones recientes de WordPress. Por defecto, se establece en 0, lo que significa que el término no pertenece a ningún grupo específico.	Puede agrupar términos relacionados, pero su uso es menos común en versiones recientes de WordPress.

Table 9. wp_term_relationships

Nombre del campo	Tipo	Longitud	Descripción	Uso
object_id	bigint	20	Almacena el ID del objeto al que se aplica la relación. En la mayoría de los casos, se refiere al ID de una entrada o página en la tabla wp_posts .	Identifica la entrada o página relacionada.
term_taxonomy_id	bigint	20	Almacena el ID de la taxonomía del término relacionado.	Relaciona la relación con el término de taxonomía correspondiente en wp_terms.
term_order	int	11	Almacena el orden de la relación entre el objeto y el término de taxonomía.	Determina la posición relativa del término en relación con otros términos asociados al mismo objeto.

Table 10. *wp_term_taxonomy*

Nombre del campo	Tipo	Longitud	Descripción	Uso
term_taxonomy_id	bigint	20	Almacena el ID de entrada de taxonomía único en WordPress.	Se utiliza para relacionar cada relación con la información detallada de la taxonomía en la tabla wp_term_taxonomy . Este campo actúa como clave foránea en wp_term_relationships y clave primaria en wp_term_taxonomy .
term_id	bigint	20	Almacena el ID del término de taxonomía relacionado.	Relaciona la entrada de taxonomía con el término correspondiente en wp_terms.
taxonomy	varchar	32	Almacena el nombre de la taxonomía (categoría, etiqueta, etc.).	Identifica el tipo de taxonomía al que pertenece la entrada.
description	longtext	-	Almacena una descripción adicional de la taxonomía.	Proporciona información adicional sobre la taxonomía.
parent	bigint	20	Almacena el ID del término de taxonomía padre (jerarquía).	Establece relaciones jerárquicas entre términos de taxonomía.
count	bigint	20	Almacena el recuento de objetos asociados a la taxonomía.	Indica cuántas veces se ha utilizado esta taxonomía en objetos del sitio.

Table 11. *wp_usermeta*

Nombre del campo	Tipo	Longitud	Descripción	Uso
umeta_id	bigint	20	Almacena el ID de metadato de usuario único en WordPress.	El campo <i>user_id</i> en la tabla wp_usermeta se utiliza para relacionar cada registro de metadatos de usuario con la tabla wp_users . Este campo actúa como clave foránea (FOREIGN KEY) que referencia el ID de usuario en la tabla wp_users .
user_id	bigint	20	Almacena el ID del usuario al que está asociado el metadato.	Relaciona el metadato con el usuario correspondiente en wp_users.
meta_key	varchar	255	Almacena la clave o nombre del metadato de usuario.	Identifica y busca metadatos relacionados con un usuario específico.
meta_value	longtext	-	Almacena el valor del metadato de usuario.	Contiene información personalizada o metadatos adicionales relacionados con el usuario.

Table 12. wp_users

Nombre del campo	Tipo	Longitud	Descripción	Uso
ID	bigint	20	Almacena el ID único de cada usuario en WordPress. Se genera automáticamente para cada nuevo usuario y se incrementa secuencialmente.	Identifica de manera única a cada usuario.
user_login	varchar	60	Almacena el nombre de usuario utilizado para iniciar sesión. Utiliza el conjunto de caracteres utf8mb4_unicode_ci.	Utilizado para autenticación.
user_pass	varchar	255	Almacena la contraseña del usuario de manera segura. Utiliza el conjunto de caracteres utf8mb4_unicode_ci.	Las contraseñas se almacenan en forma encriptada o hash.
user_nicename	varchar	50	Almacena un nombre amigable para URL del usuario. utiliza el conjunto de caracteres utf8mb4_unicode_ci.	Utilizado en las URL del perfil del usuario.
user_email	varchar	100	Almacena la dirección de correo electrónico del usuario. Utiliza el conjunto de caracteres utf8mb4_unicode_ci.	Utilizado para fines de comunicación y notificaciones.

Nombre del campo	Tipo	Longitud	Descripción	Uso
user_url	varchar	100	Almacena la URL del sitio web o enlace del perfil del usuario. Utiliza el conjunto de caracteres utf8mb4_unicode_ci.	Los usuarios pueden proporcionar un enlace a su sitio web personal.
user_registered	datetime	-	Registra la fecha y hora de registro del usuario.	Permite llevar un registro de cuándo se unió cada usuario.
user_activation_key	varchar	255	Almacena una clave de activación temporal para la cuenta. Utiliza el conjunto de caracteres utf8mb4_unicode_ci.	Utilizada cuando se requiere activación por correo electrónico.
user_status	int	-	Almacena el estado del usuario (activo o inactivo). Los valores comunes son <i>0</i> para cuentas inactivas y <i>1</i> para cuentas activas.	Indica si la cuenta está activa o inactiva.
display_name	varchar	250	Almacena el nombre que se muestra públicamente para el usuario. Utiliza el conjunto de caracteres utf8mb4_unicode_ci.	Utilizado para mostrar la autoría en publicaciones y comentarios.