

Aprende SQL

Con MySQL y MariaDB

Pastor Ramos

Aprende SQL

Con MySQL y MariaDB

Styde.net y Pastor Ramos

Este libro está a la venta en <http://leanpub.com/aprende-mysql>

Esta versión se publicó en 2020-02-07



Este es un libro de [Leanpub](#). Leanpub anima a los autores y publicadoras con el proceso de publicación. [Lean Publishing](#) es el acto de publicar un libro en progreso usando herramientas sencillas y muchas iteraciones para obtener retroalimentación del lector hasta conseguir el libro adecuado.

© 2018 - 2020 Styde.net y Pastor Ramos

Índice general

Introducción	i
¿Cómo leer este libro?	i
Convenciones usadas en el libro	ii
Ponerse en contacto	iv
Visión General	1
Arquitectura	1
Componentes de MySQL	1
Convenciones de nombres de bases de datos	3

Introducción

Bienvenidos al maravilloso mundo de las bases de datos, a lo largo de este libro, **Aprende SQL con MySQL y MariaDB**, encontrarás lo que andas buscando para participar en cualquier proyecto que requiera que utilices el lenguaje SQL.

Este libro se conforma de varias partes que tienen un propósito específico.

Parte I: Contiene una visión general del servidor que vas a utilizar y la aplicación de las convenciones de nombres de bases de datos para que uses cualquiera de los frameworks más poderosos que existen, sin problemas.

El libro también incluye temas relacionados con la instalación del software de base de datos MariaDB y la configuración de la seguridad de acceso mínima requerida. Esto es útil si no tienes instalado ningún servidor basado en MySQL.

Luego, se enfoca en la definición del esquema o estructura de una base de datos, sus tablas (el principal elemento) y culmina explicándose cómo puedes modificar dicha estructura.

Parte II: Describe las operaciones de manipulación que puedes aplicar en una base de datos: Como insertar, eliminar, actualizar y consultar datos sobre una tabla. Además tiene un apartado sobre el tema de los índices para que los comprendas sin inconvenientes.

Parte III: Explica los operadores relacionales que existen tales como el producto cartesiano, la intersección, la sustracción y sobre todo ayuda a que comprendas mejor el funcionamiento interno del comando SELECT. Expone ampliamente la importante operación de junta o “JOIN” recomendada por su eficiencia y finaliza con las sub-consultas que ampliará aún más tu lógica.

Parte IV: Iré a fondo con explicaciones sobre unos de los aspectos menos comprendido por la gran mayoría: el uso de los operadores para efectuar cálculos de campos derivados tanto en forma horizontal como verticalmente a través de funciones de agrupamiento y sus variantes. Todo libro completo sobre SQL debe enseñar la amplia variedad de funciones para manipular tipos de datos sobre cadenas, fechas y horas, las cuales incluyo sin duda alguna.

Parte V: Por cuestiones de seguridad informática considero que es importante que conozcas el aspecto de las cuentas de usuario, los privilegios de acceso y los respaldos de bases de datos, así que de esto se trata esta última parte.

¿Cómo leer este libro?

En general, el libro está destinado para ser leído básicamente siguiendo la secuencia en la que está siendo escrito, aunque si lo prefieres, puedes saltarte los últimos capítulos que están destinados a que conozcas aspectos que tienen que ver con la seguridad informática y la sección sobre instalación.

Un plan sugerido para una primera lectura sería:

- Leer el capítulo 1 “ligeramente”
 - Capítulo 1 - Visión General
- Leer muy ligeramente los capítulos 2 y 3 ya que describe la herramienta de trabajo y su configuración.
 - Capítulo 2 - Instalación del software
 - Capítulo 3 - Configuración
- Leer “ligeramente” el capítulo 4
 - Capítulo 4 - Creación de una base de datos
- Leer los capítulos 5 y 6 muy detenidamente
 - Capítulo 5 - Creación de tablas
 - Capítulo 6 - Modificación de la estructura de las tablas
- Leer los capítulos 7, 8, 10, 11, 12, 13 14 y 15 con mucho detenimiento pero saltando el capítulo 9
 - Capítulo 7 - Insertando datos en tablas
 - Capítulo 8 - Consultas sobre tablas
 - Capítulo 9 - Índices
 - Capítulo 10 - Actualizando datos
 - Capítulo 11 - Eliminando datos
 - Capítulo 12 - Operadores relacionales
 - Capítulo 13 - Funcionamiento interno del comando SELECT
 - Capítulo 14 - Juntas
 - Capítulo 15 - Sub-consultas
- Leer ligeramente los capítulos 16 - 18 y prestar mucha atención al capítulo 18
 - Capítulo 16 - Funciones para manejo de cadenas
 - Capítulo 17 - Funciones para manejo de fechas y horas
 - Capítulo 18 - Funciones de agrupamiento y cálculos horizontales
- Leer selectivamente los capítulos 19 y 20
 - Capítulo 19 - Creación de cuentas de usuario y privilegios
 - Capítulo 20 - Respaldando y restaurando una base de datos
- Leer ligeramente los apéndices.
 - Apéndice A - Instalación de MySQL de Oracle
 - Apéndice B - Instalación de PerconaDB

Convenciones usadas en el libro

Las negritas

Se usan para representar términos que no son comandos del lenguaje pero son importantes en el ámbito de bases de datos.

Los comandos, cláusulas, palabras clave de SQL

Están escritos en letras mayúsculas, por ejemplo:

```
CREATE TABLE
```

Los nombres de tablas y de campos

Aparecen en letras minúsculas, tal como: providers, date_of_register.

Referencias en el texto a ejemplos de código

Esto es un párrafo normal, con un trozo de código, `DESCRIBE providers;` insertado en el medio del mismo.

Bloques de código

Por lo general, el código mostrado son comandos SQL que incluyen el resultado de la consulta, separados por una línea. En caso de que el resultado sea muy extenso se colocará en un bloque de código aparte. El siguiente es un ejemplo de párrafo de código con resultado:

```
MariaDB [almacen]> CREATE TABLE providers (
    id INT AUTO_INCREMENT PRIMARY KEY,
    prov_name CHAR(200) UNIQUE,
    category_id INT UNSIGNED,
    total NUMERIC(10,2) UNSIGNED,
    date_of_register DATETIME,
    description TEXT);
```

```
Query OK, 0 rows affected (0.37 sec)
```

Tips, notas, advertencias y ejercicios.



Representa un tip o sugerencia que te puede servir para resolver un problema.



Representa una advertencia o precaución.



Indica alguna información especial.



Indica un ejercicio.

Ponerse en contacto

En caso de que quieras ponerte en contacto con nosotros por algo relacionado con el libro, enviarnos feedback u otros asuntos que te gustaría traer a nuestra atención, no dudes en contactarnos.

Nombre	Correo electrónico	Twitter
Styde	admin+sql@styde.net	@Stydenet
Pastor Ramos	pastorramos2004@gmail.com	@IngPastorRamos

Esta obra se encuentra en progreso, apenas esté disponible una actualización del libro recibirás un aviso en tu correo electrónico para que la descargues. Esto forma parte de las bondades del proceso de publicación “Lean Publishing” con el cual serás el primero en recibir el contenido actualizado y nosotros podemos anticipar el lanzamiento de un libro donde esté garantizada la máxima calidad.

Este libro representa una inversión de varios meses de trabajo de parte de su autor y del equipo de Styde y está disponible para su venta en <https://leanpub.com/aprende-mysql>. Agradecemos tu apoyo y colaboración para la realización exitosa de este proyecto.

Visión General

En este capítulo se estudiarán algunos conceptos básicos sobre las bases de datos que se aplican a MySQL, MariaDB, Percona Server o cualquier producto basado en MySQL. La idea general es que no sientas que debes cambiar el sistema que tengas instalado para seguir los consejos de este libro.

Arquitectura

El uso que le daré en este libro a MySQL es el de servidor de base de datos relacional para aplicaciones cliente-servidor, en el cual la base de datos reside en un computador central y cuya información es compartida por diversos usuarios que ejecutan las aplicaciones en sus computadores locales, o clientes.

Actualmente, MySQL posee versiones orientadas a sistemas relacionales de bases de datos distribuidas, pero esto excede el alcance de este libro.

Componentes de MySQL

Estos son los principales componentes de MySQL.

- **El servidor MySQL** es el servidor de base de datos. Procesa todas las consultas y manipula tablas y bases de datos.
- **Los clientes MySQL** son programas que se comunican con el servidor.
- **El directorio de datos** es el lugar donde MySQL guarda los datos.
- **El motor de almacenamiento** maneja cómo la información es organizada, almacenada y accedida.

El servidor MySQL

Es el encargado de manejar los cambios en el directorio de datos, el cual contiene las bases de datos y tablas. Acepta conexiones de los clientes y maneja su acceso a las bases de datos. Estos son los principales programas del servidor MySQL:

- **mysqld** es el servidor de base de datos.
- **mysqld_safe** es un script para iniciar MySQL que añade unas características extras.

Clientes de MySQL

Son los programas que se conectan a MySQL para acceder a las bases de datos y tablas. Entre los más importantes están:

- `mysql` es un programa interactivo donde puedes escribir las consultas. También puede ejecutar scripts escritos en SQL.
- `mysqladmin` es un programa administrativo de MySQL, puedes usarlo, por ejemplo, para configurar cuentas de usuario.
- `mysqldump` es un programa que puedes usar para hacer respaldos de las bases de datos.

Motores de almacenamiento

MySQL usa arquitectura de motor de almacenamiento “enchufable”. Por ejemplo: InnoDB, Memory, CSV, MyISAM.

Directorio Data

Este directorio contiene toda la información manejada por el servidor MySQL.

En MySQL la información de la base de datos se guarda en un directorio con el mismo nombre.

Las bases de datos contienen una colección de objetos que se usan para representar, almacenar y tener acceso a los datos. Algunos de estos objetos que puede contener son:

Tablas

Son la esencia de las bases de datos, son éstas las que almacenan los datos en una base de datos. Agrupan los datos en forma de filas y columnas como una hoja de cálculo.

Cada fila representa un registro y cada columna un atributo o campo de la tabla.

Los campos de una tabla pueden tener restricciones en cuanto al contenido que van a almacenar.

Índices

Son tipos especiales de archivos que trabajan asociados con tablas. Su finalidad es acelerar el proceso de acceso a un determinado registro o grupo de registros.

Diagramas de bases de datos

Son la representación gráfica de las tablas, vistas e índices almacenados por la base de datos. Puedes interactuar con este diagrama usando para ello tu ratón para la ejecución de varias tareas sin la necesidad de usar comandos por consola.

Vistas

Es una tabla virtual cuyo contenido se ha definido por medio de una consulta de base de datos. Una vista no almacena datos como lo hace una tabla.

Procedimientos almacenados

Generalmente manipula la base de datos usando instrucciones en lenguaje SQL a través de una aplicación que sirva de interfaz con la base de datos.

Puedes crear un programa escrito en lenguaje SQL que se almacene y ejecute directamente en el servidor. Esto es lo que se conoce como procedimiento almacenado.

Puedes activarlo mediante una llamada de procedimiento desde tu aplicación. Tienen la ventaja de que su ejecución es más rápida.

Triggers o disparadores

Un trigger es un tipo de procedimiento almacenado que se ejecuta automáticamente cuando un dato de la tabla es modificado como resultado de la ejecución de un comando SQL de tipo INSERT, UPDATE, DELETE. Puedes usarlos para mantener restricciones más complejas o fuertes que limiten el valor de los datos insertados en una columna.

Índices Full-Text

Es un tipo especial de índices que permiten la ejecución de consultas con base en columnas cuyo contenido sea de tipo texto o caracteres de texto.

Convenciones de nombres de bases de datos

Las convenciones de nombres de bases de datos son muy importantes porque denotan uniformidad de criterio al aplicar estándares entre todas las personas que participen en el desarrollo de cualquier proyecto de base de datos. Lo que a la larga se traduce en un mejor desarrollo.

Típicamente en MySQL los nombres se componen de letras, números y algunos caracteres especiales, pero no impone restricciones a los nombres que uses. Solo estarás limitado por la cantidad de caracteres que acepte el sistema para un nombre de objeto de base de datos: tablas, campos, etc.

Sin embargo, hoy en día existen diversos frameworks de desarrollo de software que usan el paradigma “convención sobre configuración”, el cual permite agilizar y facilitar el trabajo al programador, ya que éste solo necesitará tomar decisiones no convencionales de la aplicación. Dichos frameworks establecen y sugieren convenciones para nombrar los elementos de base de datos que tomaremos en cuenta para definir los ejemplos de este libro. De esta manera, aplicar uniformidad a la base de datos y adicionalmente, facilitarte el trabajo si quieres dar ese “salto de nivel” que andas buscando en el desarrollo de aplicaciones usando frameworks.

Puedes usar las siguientes convenciones de nombres de bases de datos:

1. Es recomendable que uses **sustantivos en idioma inglés** ya que al ser el idioma universal puedes crear bases de datos que sean entendidas por diferentes programadores, diseñadores y empresas de todo el globo.
2. Nombres de tablas:
 - Usa sustantivos en minúsculas y plural. Por ejemplo: notes, students, professors, books, parts, etc.
 - Existen casos donde un solo sustantivo no será suficiente y deberás usar un sustantivo compuesto por dos o más palabras para diferenciarlo de los demás. Por ejemplo: art_books, physic_books, product_types, etc.
 - De lo anterior, debes emplear “guion bajo” o “_” para separar dos sustantivos.
3. Nombres de los campos/columnas/atributos.
 - Usa sustantivos con letras minúsculas pero en singular. Por ejemplo: address, name, description, etc.
 - En los casos que necesites más de un sustantivo sepáralos con un guión bajo. Por ejemplo: date_of_birth, provider_name, registration_date, etc.
4. Las claves primarias.
 - Nombra las claves primarias de tus tablas como id.
 - Debe ser de tipo entero sin signo y con auto-incremento.
 - Una clave primaria jamás puede permitir valores nulos.
5. Las claves foráneas.
 - Usa un sustantivo compuesto por el nombre de la tabla de donde proviene en minúsculas y en singular seguido por el sufijo _id. Por ejemplo user_id, category_id, provider_id, product_id, etc.
 - Dado que son campos que contienen valores que son clave primaria en otra tabla, también debes definirla de tipo entero sin signo.
 - Puedes permitir valores nulos.
6. Añade a la tabla los campos especiales created_at y updated_at.
 - Son dos campos especiales de tipo de dato TIMESTAMP para indicar cuando se creó el registro y cuando fue la última vez que fue actualizado. Nómbralos como created_at y updated_at.
7. Vínculos **muchos a muchos** también conocidos como **tablas pivot**.
 - Usa los nombres de las **tablas participantes** en singular y orden alfabético separados por un guión bajo. Por ejemplo, para un vínculo **muchos a muchos** entre la tabla users y la tabla roles corresponde el nombre role_user.
 - Debe contener una clave foránea para cada **tabla participante**. En el ejemplo anterior se usarían los campos user_id y role_id.
8. Vínculos **uno a uno, uno a muchos, muchos a uno**.
 - No crees una tabla adicional a menos que se justifique.
 - Usa una clave foránea para representar el vínculo en la otra tabla.
9. Vínculos que incluyen al **cero** en su cardinalidad.
 - Haz que tu clave foránea permita valores nulos del lado dependiente.