

# Introducción a MySQL

## Instalación de MySQL

MySQL es un manejador de bases de datos relacionales de acceso libre (sin costo) pero ello no significa que sea una herramienta de baja calidad o plagada de errores. Por el contrario es una herramienta robusta y muy documentada. Existen versiones de MySQL para varias plataformas incluyendo Windows y Linux.

La versión de MySQL para Windows empleada en este tutorial es **MySQL Server 5.1.43** Para instalarla siga el siguiente procedimiento:

1. Haga doble clic en el archivo de instalación, **mysql-5.1.43-win32.msi**. Aparecerá la pantalla inicial del asistente de instalación de **MySQL Server** mostrado en la figura 1:



**Figura 1**

2. Haga clic en el botón **Next**. Aparecerá el cuadro de dialogo de la figura 2. Aquí se establece el tipo de instalación deseada. Como se hará la instalación típica no modifique nada.



Figura 2

3. Haga clic en el botón **Next**. Aparecerá el cuadro de dialogo de la figura 3 para confirmar el tipo de instalación y la ubicación de los archivos.

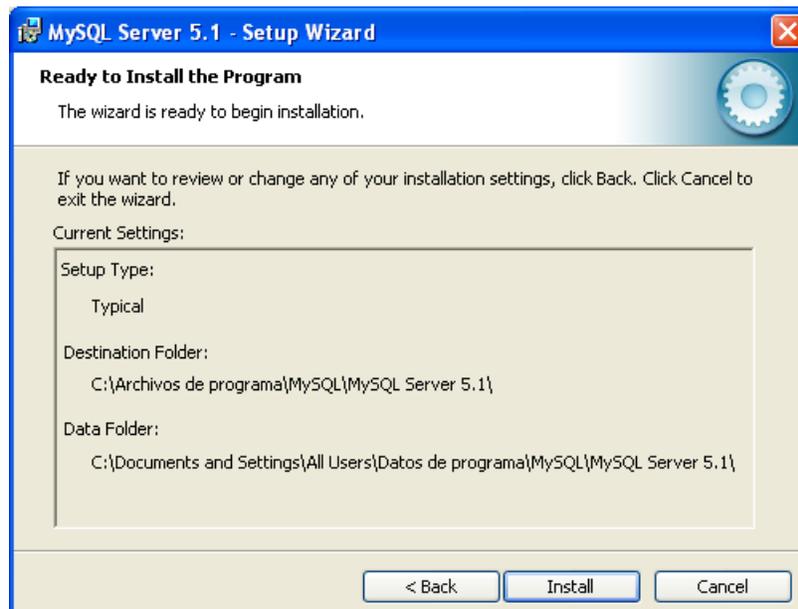


Figura 3

4. Acepte esta configuración, haciendo clic en el botón **Install**. Al hacerlo empezará la instalación de **MySQL Server**, figura 4. Después aparecerá el cuadro de la figura 5 que contiene un comercial de los diferentes productos de la compañía que produce MySQL.

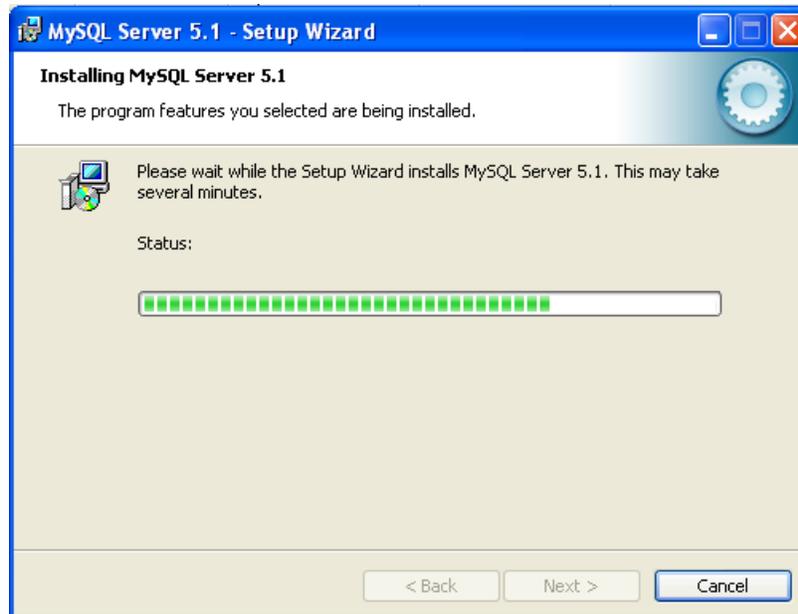


Figura 4



Figura 5

5. Haga clic en el botón **Next**. Aparecerá el cuadro de dialogo de la figura 6 con otro comercial.
6. Haga clic en el botón **Next**. Aparecerá el cuadro de dialogo de la figura 7. Este cuadro de diálogo nos informa que la instalación de MySQL terminó y nos pregunta si deseamos configurar a MySQL y si deseamos registrar al servidor MySQL.



Figura 6



Figura 7

7. Deje seleccionada la primera casilla de verificación para configurar el servidor y desmarque la segunda casilla para no registrar el servidor y haga clic en el botón **Finish**. Aparecerá el cuadro de dialogo de la figura 8.
8. Este cuadro de diálogo es el asistente de configuración de MySQL. Para iniciar la configuración haga clic en el botón **Next**. Aparecerá el cuadro de la figura 9.

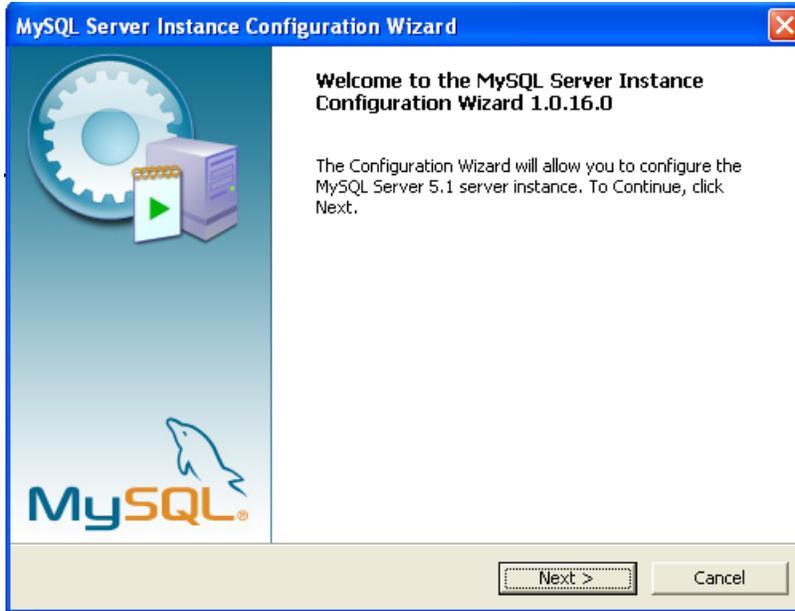


Figura 8



Figura 9

9. Aquí podemos seleccionar el tipo de configuración. Seleccionaremos la configuración estándar seleccionando el botón de radio **Standard Configuration** y haciendo clic en el botón **Next**. Aparecerá el cuadro de la figura 10.



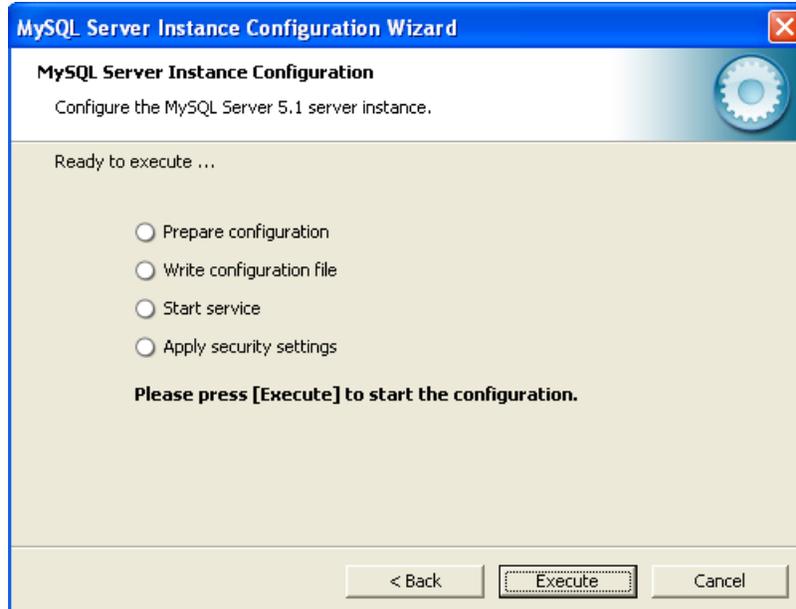
Figura 10

10. En este cuadro de diálogo la primera caja de selección: **Install As Windows Service**, nos permite instalar a **MySQL** como un servicio. Lo aceptaremos. La segunda caja de selección, **Launch the MySQL Server automatically**, nos permite seleccionar si deseamos arrancar a **MySQL** automáticamente. Acéptelo. La tercera caja de selección permite incluir la carpeta **bin** de **MySQL** en la ruta de búsqueda de Windows para que podamos acceder a los programas en esta carpeta desde cualquier carpeta desde una ventana del sistema. Marque la caja de selección y presione el botón **Next**. Aparecerá el cuadro de la figura 11.



Figura 11

11. En este cuadro de diálogo teclee la contraseña de la cuenta de root (administrador). Tecléela de nuevo para confirmar. Deje el resto del cuadro como se muestra en la figura 11 y presione el botón **Next**. Aparecerá el cuadro de la figura 12.



**Figura 12**

12. Aquí **MySQL Server** nos pide confirmación antes de proceder con la configuración. Clic en el botón **Execute** para confirmar. **MySQL Server** empezará la configuración y al terminar desplegará el cuadro de la figura 13. Haga clic en el botón **Finish** para terminar cerrar el asistente.

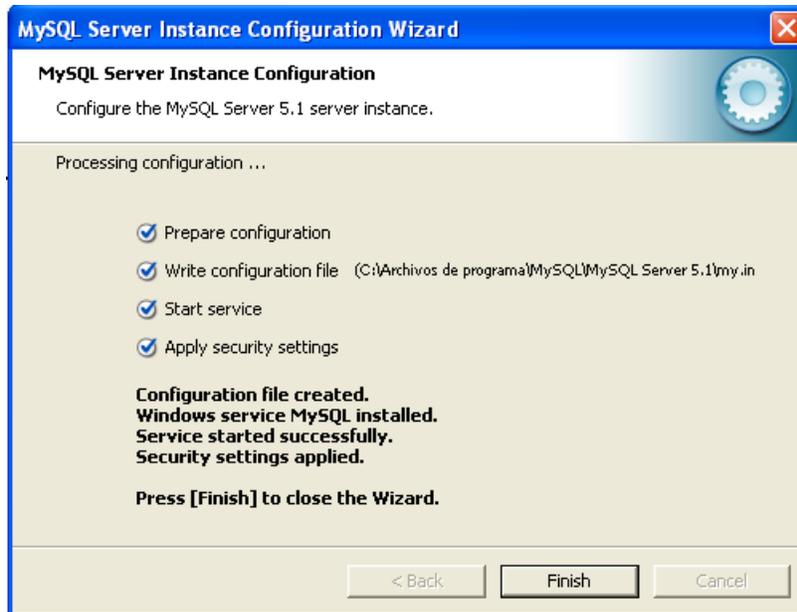


Figura 13

## Ejecución de un Cliente de Consola para MySQL Server

Para administrar cuentas, crear bases de datos, tablas, etc. requerimos abrir una sesión con **MySQL Server**. Esto se logra con un cliente para **MySQL Server**. **MySQL Server** nos proporciona un cliente de consola llamado **mysql**. Para ejecutar **mysql** tenemos que abrir una ventana de consola como se muestra en la figura 14.

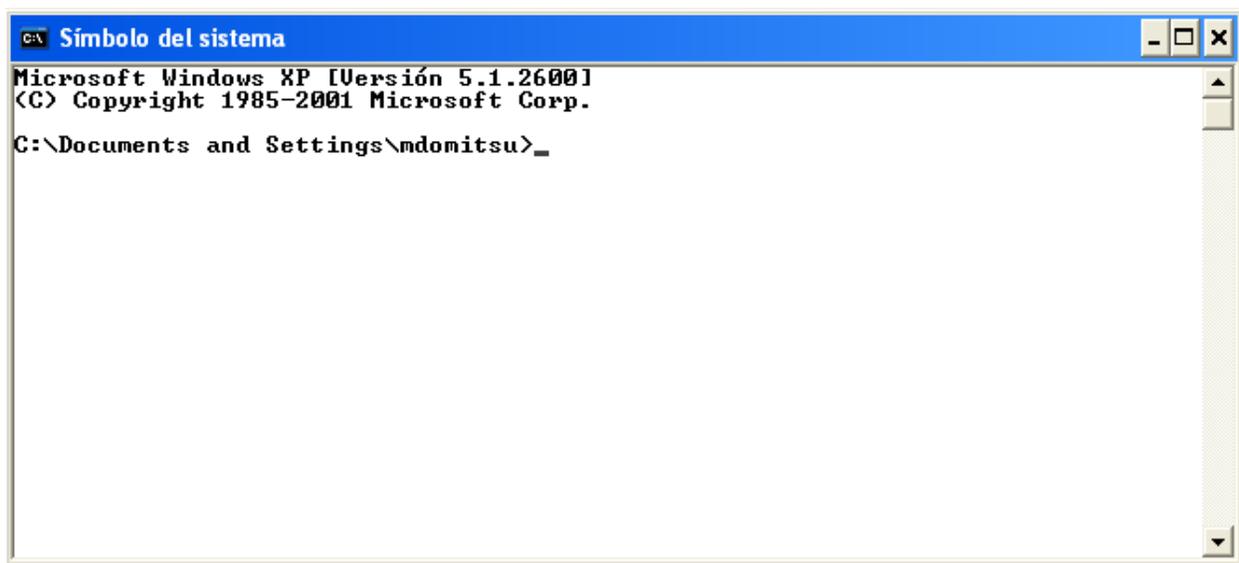


Figura 14

En la línea de comando teclearemos:

```
C:\Documents and Settings\mdomitsu>mysql -u root -p
```

**mysql** nos pide la contraseña de root:

```
Enter password:
```

**mysql** nos responde:

```
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 9 to server version: 4.1.10a-nt

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.
```

indicándonos que tenemos acceso a las bases de datos de **MySQL Server** como el usuario root. El usuario root tiene todos los permisos para acceder a todas las bases de datos la base de datos de **MySQL Server**.

## Administración de Cuentas de Usuario

Al instalarse **MySQL Server** crea una base de datos llamada `mysql` con varias tablas. Para ver las bases de datos existentes en el servidor podemos usar una de las variantes del comando `show`:

```
mysql> show databases;
```

Note que todos los comandos deben terminar en un punto y coma (;). **MySQL Server** nos muestra sus bases de datos, respondiendo:

```
+-----+
| Database |
+-----+
| mysql   |
| test    |
+-----+
2 rows in set (0.00 sec)
```

La base de datos **mysql** contiene varias tablas empleadas para la administración del servidor. Primero se establece que se va a trabajar ella, usando el comando `use`, tecleando:

```
mysql> use mysql;
```

**MySQL Server** responde:

```
Database changed
```

Indicando que la base de datos en uso es mysql. Para ver las tablas de la base de datos mysql, usaremos el comando **show**, tecleando:

```
mysql> show tables;
```

MySQL responde:

```
+-----+
| Tables_in_mysql |
+-----+
| columns_priv    |
| db              |
| func            |
| help_category   |
| help_keyword    |
| help_relation   |
| help_topic      |
| host            |
| tables_priv     |
| time_zone       |
| time_zone_leap_second |
| time_zone_name  |
| time_zone_transition |
| time_zone_transition_type |
| user            |
+-----+
15 rows in set (0.00 sec)
```

Una de las tablas, la tabla **user**, contiene la relación de los usuarios y desde donde tienen acceso a la base de datos. Para visualizar el contenido parcial o total de una o más tablas usaremos el comando **select**. Para listar los campos user, password y host de la tabla teclearemos:

```
mysql> select user,password,host from user;
```

MySQL responde:

```
+-----+-----+-----+
| user | password | host |
+-----+-----+-----+
| root | *E3BD66E01EE21D5E235BAA48A35FEC5D42C87EB2 | localhost |
+-----+-----+-----+
1 row in set (0.00 sec)
```

Mostrando un usuario: root con acceso local, es decir, desde la computadora que tiene **MySQL Server (localhost)**. Note que la contraseña está encriptada.

Ahora crearemos un nuevo usuario que tendrá acceso local. Para ello usaremos el comando **insert**:

```
mysql> insert into user (host,user,password) values ('localhost',
'usuario', password('contraseña'));
```

Donde *usuario* es el nombre del nuevo usuario y *contraseña* es su contraseña.  
**MySQL Server** responde:

```
Query OK, 1 row affected (0.11 sec)
```

Los comandos muy largos pueden segmentarse en varias líneas para que los podamos visualizar más claramente. Para ello podemos presionar la tecla [Intro] cuando queramos partir la línea. Recuerde que el comando sólo termina cuando tecleamos un punto y coma (;) seguido de la tecla [Intro]. Al presionar la tecla [Intro] para partir la línea, **MySQL Server** nos despliega una flecha indicándonos que espera el resto del comando. Por ejemplo, el comando **insert** anterior pudo haberse tecleado como:

```
mysql> insert into user (host,user,password)
-> values ('localhost', 'usuario', password('contraseña'));
```

Debido a los caches internos usados por **MySQL Server**, puede ser que los cambios hechos comando **insert** no se guarden inmediatamente en la tabla user. Para forzar la actualización inmediata de la tabla podemos usar el comando **flush**, tecleando:

```
mysql> flush privileges;
```

**mysql** responde:

```
Query OK, 0 rows affected (0.10 sec)
```

Para verificar que se creó el nuevo usuario, repetimos el comando **select**:

```
mysql> select user, password, host from user;
```

**mysql** responde:

```
+-----+-----+-----+
| user   | password                                     | host   |
+-----+-----+-----+
| root   | *E3BD66E01EE21D5E235BAA48A35FEC5D42C87EB2 | localhost |
| usuario | *65EE8013472A455529AE1173AD495800460BA919 | localhost |
+-----+-----+-----+
2 rows in set (0.00 sec)
```

Para probar que el acceso a la cuenta **usuario** cerramos la sesión actual tecleando el comando **quit**:

```
mysql> quit;
```

**mysql** responde:

```
Bye
```

Probemos una sesión como el usuario **usuario**, tecleando:

```
C:\Documents and Settings\mdomitsu>mysql -u usuario -p
```

**mysql** nos pide la contraseña. Al teclearla se abrirá la sesión como antes, aunque no podremos hacer mucho como podremos ver si queremos utilizar la base de datos **user**, tecleando:

```
mysql> use mysql;
```

**mysql** responde:

```
ERROR 1044 (42000): Access denied for user 'usuario'@'localhost'
to database 'user'
```

Cierre la sesión actual y abra otra como **root**.

## Creación de una Base de Datos

A continuación crearemos una base de datos llamada **musica**, usando el comando **create**. Para ello teclearemos:

```
mysql> create database musica;
```

**mysql** responde:

```
Query OK, 1 row affected (0.00 sec)
```

Para verificar que se creó la base de datos, usamos el comando **show**, tecleando:

```
mysql> show databases;
```

**mysql** responde:

```
+-----+
| Database |
+-----+
| musica  |
| mysql   |
| test    |
+-----+
3 rows in set (0.00 sec)
```

Ahora usamos el comando **use** para usar la base de datos **musica**

```
mysql> use musica
```

**mysql** responde:

```
Database changed
```

Ahora le daremos todos los privilegios de acceso local a las tablas de la base de datos **musica** al usuario creado en la sección anterior. Para ello usamos el comando **grant**:

```
mysql> grant all privileges on musica.*
-> to usuario@localhost identified by 'contraseña'
-> with grant option;
```

Donde *usuario* es el nombre del nuevo usuario y *contraseña* es su contraseña. **mysql** responde:

```
Query OK, 0 rows affected (0.06 sec)
```

Podemos usar el comando **grant** para crear un usuario y darle privilegios de acceso a una base de datos. Por ejemplo, para crear un usuario y darle acceso remoto para consultar, insertar, actualizar y borrar a las tablas de la base de datos **musica** desde una máquina remota teclaremos:

```
mysql> grant select, insert, update, delete on musica.*
-> to usuario@%' identified by 'contraseña';
```

Donde *usuario* es el nombre del nuevo usuario y *contraseña* es su contraseña. **mysql** responde:

```
Query OK, 0 rows affected (0.11 sec)
```

Termine la sesión.

Ahora abriremos una sesión como usuario y usaremos la base de datos **musica**. Podemos abrir una sesión y seleccionar una base de datos al mismo tiempo con la siguiente variante de la llamada a **mysql**:

```
C:\mysql\bin>mysql -u usuario -p musica
```

**mysql** responde:

```
Enter password:
```

Al teclear la contraseña, **mysql** responde:

```
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 5 to server version: 4.0.12-max-debug
```

```
Type 'help;' or '\h' for help. Type '\c' to clear the buffer.
```

## Creación de una Tabla

La base de datos **música** tendrá las tablas **canciones**, **películas** y **generos** empleadas para almacenar los datos de las canciones, películas y géneros del programa Amante Música. Las tablas de la base de datos **música**, sus campos y sus relaciones se muestran en el diagrama entidad – relación de la figura 15.

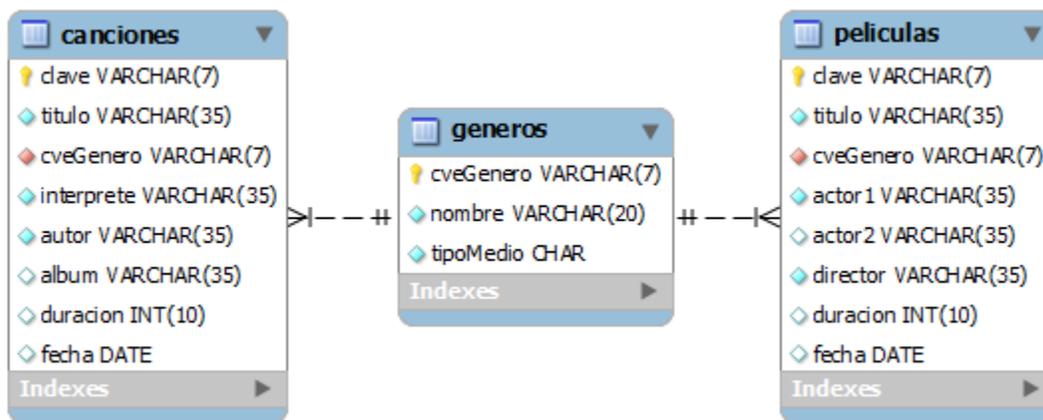


Figura 15

El símbolo en forma de una llave indica que el campo es la llave primaria. El símbolo en forma de un rombo café indica que el campo es una llave foránea. Los rombos en azul indican campos no nulos. Los rombos huecos indican campos que pueden ser nulos.

**VARCHAR(n)** indica que el campo es una cadena de longitud variable de n caracteres, **CHAR** indica que el campo es un caracter, **INT(N)** indica que el campo es un entero, **DATE** es un campo para almacenar fechas.

Para crear la tabla **generos** usaremos el comando **create**:

```
mysql> create table generos (
  -> cveGenero varchar(7) not null,
  -> nombre varchar(20) not null,
  -> tipoMedio char not null,
  -> primary key(cveGenero));
```

Donde la expresión `varchar(n)` indica que el tipo del campo es una cadena variable de n caracteres, `char(n)` indica que el tipo del campo es una cadena fija de n caracteres, `not null` indica que al insertar un renglón en la tabla, ese campo no puede dejarse vacío y `primary key` nos indica que ese campo es la llave primaria (de ordenamiento).

**mysql** responde:

```
Query OK, 0 rows affected (0.11 sec)
```

Para verificar que se creo la tabla canciones, veremos la lista de tablas de la base de datos musica, mediante el comando **show**:

```
mysql> show tables;
```

**mysql** responde:

```
+-----+
| Tables_in_musica |
+-----+
| generos          |
+-----+
1 row in set (0.00 sec)
```

También podemos ver los campos de la tabla **canciones** usando otra variante del comando **show**:

```
mysql> show columns from canciones;
```

**mysql** responde:

```
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| cveGenero  | varchar(7) | NO   | PRI |          |       |
| nombre     | varchar(35) | NO   |     |          |       |
| tipoMedio  | char       | NO   |     |          |       |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

## Inserción de un Renglón en una Tabla

Para insertar un renglón en la tabla **canciones**, usaremos el comando **insert**:

```
mysql> insert into canciones values(
-> 'GC001'
-> Balada'
-> 'C');
```

**mysql** responde:

```
Query OK, 1 row affected (0.00 sec)
```

En el comando **insert** anterior se proporcionaron los valores para todos los campos del renglón a insertar. Si sólo deseamos proporcionar parte de los valores (los campos obligatorios deben proporcionarse), podemos usar esta otra variante del comando **insert**:

```
mysql> insert into generos (
-> cveGenero,
-> nombre,
-> tipoMedio)
```

```
-> values (
-> 'GC002'
-> 'Bossanova');
```

**mysql** responde:

```
Query OK, 1 row affected (0.00 sec)
```

Para comprobar que los renglones se han insertado en la tabla ejecutamos el comando **select**:

```
mysql> select cveGenero, nombre from generos;
```

**mysql** responde:

```
+-----+-----+
| cveGenero | nombre          |
+-----+-----+
| GC001     | Balada          |
| GC002     | Bossanova       |
+-----+-----+
2 rows in set (0.02 sec)
```

Para mostrar todos los campos de la tabla, podemos teclear:

```
mysql> select * from generos;
```

**mysql** responde:

```
+-----+-----+-----+
| cveGenero | nombre          | tipoMedio |
+-----+-----+-----+
| GC001     | Balada          | C         |
| GC002     | Bossanova       | C         |
+-----+-----+-----+
2 rows in set (0.00 sec)
```

## Esquema para Crear Tablas

La creación de una base de datos y sus tablas desde la línea de comando, sobre todo en tablas con muchas columnas, es un proceso en el cual es fácil cometer errores al estar tecleando el comando, lo que nos obliga a empezar de nuevo cada vez que cometemos un error. Una alternativa al uso de este comando es el de escribir las sentencias para la creación de la base de datos y sus tablas en un archivo texto y hacer que **mysql** procese las sentencias contenidas en este archivo, tal y como lo haría si las sentencias se teclearan en la línea de comando. Un archivo con sentencias sql que deseamos que **mysql** ejecute, se conoce como un **esquema**. Por ejemplo, el recuadro siguiente muestra el contenido del archivo `bd_musica.sql`, un esquema con

las sentencias necesarias para crear la base de datos y las tablas **canciones**, **películas** y **generos** que se usarán para almacenar los datos de las canciones y películas del programa Amante Música.

### tablas\_musica.sql

```
-- Esquema para crear la base de datos 'musica' y sus tablas:
-- 'canciones', 'peliculas' y 'generos'
--
-- Sintaxis para crear la base de datos y sus tablas a partir
-- de este esquema:
--
--     mysql -u root -p < musica.sql
--
--
-- Elimina la base de datos
--
DROP DATABASE IF EXISTS musica;

--
-- Crea la base de datos
--
CREATE DATABASE musica;

--
-- Establece los permisos
--
GRANT ALL PRIVILEGES ON musica.* TO mdomitsu@%' IDENTIFIED BY 'sesamo' WITH
GRANT OPTION;
GRANT ALL PRIVILEGES ON musica.* TO mdomitsu@localhost IDENTIFIED BY 'sesamo'
WITH GRANT OPTION;

--
-- Utiliza la base de datos
--
USE musica;

--
-- Crea las tablas
--

CREATE TABLE generos (
  cveGenero varchar(7) NOT NULL default '',
  nombre varchar(20) NOT NULL default '',
  tipoMedio char not null default ' ',
  PRIMARY KEY (cveGenero)
) ENGINE=InnoDB;

CREATE TABLE canciones (
  clave varchar(7) NOT NULL default '',
  titulo varchar(35) NOT NULL default '',
  cveGenero varchar(7) NOT NULL default '',
  interprete varchar(35) NOT NULL default '',
  autor varchar(35) NOT NULL default '',
  album varchar(35) default NULL,
  duracion int(10) default '0',
```

```

    fecha date default NULL,
    PRIMARY KEY (clave),
    INDEX (cveGenero),
    FOREIGN KEY (cveGenero)
      REFERENCES generos(cveGenero)
      ON UPDATE CASCADE ON DELETE RESTRICT
) ENGINE=InnoDB;

CREATE TABLE peliculas (
  clave varchar(7) NOT NULL default '',
  titulo varchar(35) NOT NULL default '',
  cveGenero varchar(7) NOT NULL default '',
  actor1 varchar(35) NOT NULL default '',
  actor2 varchar(35) default NULL,
  director varchar(35) NOT NULL default '',
  duracion int(10) default '0',
  fecha date default NULL,
  PRIMARY KEY (clave),
  INDEX (cveGenero),
  FOREIGN KEY (cveGenero)
    REFERENCES generos(cveGenero)
    ON UPDATE CASCADE ON DELETE RESTRICT
) ENGINE=InnoDB;

```

Al establecer que una columna de una tabla es una llave foránea, estamos indicando que los valores que esa columna puede tomar, deben de existir en otra columna de otra tabla, llamada tabla padre. Para establecer que una columna de una tabla constituye una llave foránea se usa la cláusula **foreign key**. Su sintaxis es:

```

foreign key (nombreLlaveForanea)
  references nombreTabla(nombreLlave)
  on update cascade on delete restrict

```

donde *nombreLlaveForanea* es el nombre de la columna en la tabla hija que se desea que sea una llave foránea, *nombreTabla* es el nombre de la tabla padre y *nombreLlave* es el nombre de la columna de la tabla padre de la que dependen los valores de la columna de la tabla hija. La opción **update cascade** indica que una modificación en un valor de una llave de la tabla padre se propaga en el correspondiente valor de la llave foránea de la tabla hija. La opción **delete restrict** indica que no se puede eliminar un renglón de la tabla padre si hay un renglón o renglones en la tabla hija con la misma llave foránea que la llave padre.

## Esquema para Insertar Renglones en Tablas

Al igual que en el caso de la creación de la base de datos y sus tablas, podemos crear un esquema para insertar renglones en tablas. Por ejemplo, el recuadro siguiente

muestra el contenido del archivo `datos_musica.sql`, un esquema con las sentencias necesarias para insertar seis renglones en la tabla **generos**, tres renglones en la tabla **canciones** y dos renglones en la tabla **películas**.

### **datos\_musica.sql**

```
-- Esquema para insertar renglones en las tablas 'canciones', 'películas'
-- y 'generos' en la base de de datos musica. La base de datos y las
-- tablas deben ya existir.
--
-- Sintaxis para insertar renglones en las tablas a partir de este esquema:
--
--     mysql -u root -p < datos_musica.sql
--
--
-- Utiliza la base de datos
--
USE musica;
--
-- Datos de la tabla `generos`
--
INSERT INTO generos VALUES ('GC001','Balada','C');
INSERT INTO generos VALUES ('GC002','BossaNova','C');
INSERT INTO generos VALUES ('GC003','Rock','C');
INSERT INTO generos VALUES ('GP001','Drama','P');
INSERT INTO generos VALUES ('GP002','Ciencia Ficción','P');
INSERT INTO generos VALUES ('GP003','Comedia','P');
--
--
-- Datos de la tabla `canciones`
--
INSERT INTO canciones VALUES ('C0001','Desafinado','GC002',
'Joao Gilberto','Joao Gilberto', 'Bossanova Jazz Vol 1',
3,'1970-03-24');
INSERT INTO canciones VALUES ('C0002','Garota de Ipanema','GC002',
'Los Indios Tabajaras','Antonio Carlos Jobim','Bossanova Jazz Vol 1',
3,'1980-12-1');
INSERT INTO canciones VALUES ('C0003','The long and winding road','GC001',
'The Beatles','John Lennon', 'Let it be', 3, '1970-03-24');
--
--
-- Datos de la tabla `películas`
--
INSERT INTO peliculas VALUES ('P0001','Casa Blanca','GP001',
'Humphrey Bogart','Ingrid Bergman', 'Michael Curtiz', 102,'1942-01-01');
INSERT INTO peliculas VALUES ('P0002','2001 Space Odyssey','GP002',
'Keir Dullea','Gary Lockwood', 'Stanley Kubrick', 141,'1968-01-01');
```

Los esquemas para la base de datos y sus tablas y para agregarle renglones a tablas pueden combinarse en un solo esquema, siempre y cuando primero se escriba la sentencia para crear la base de datos y sus tablas y después se escriban las sentencias para agregarle renglones.

## Ejecución de un Esquema

Para hacer que **mysql** ejecute un esquema, ya sea para crear la base de datos y sus tablas o insertar renglones en tablas, se sigue el siguiente procedimiento:

1. Abra una ventana de consola como se muestra en la figura 12.
2. Cámbiese a la carpeta en la que está el esquema que desea ejecute (o abra la ventana de consola en esa carpeta).
3. La sintaxis para que mysql ejecute un esquema es:

```
mysql -u root -p < esquema
```

donde *esquema* es el esquema que se ejecurá para hacer esas tareas. Por ejemplo para crear la base de datos *musica* y las tablas *canciones* y *peliculas* en la base de datos *canciones*, a partir del esquema *tablas\_musica.sql*, teclearemos en la línea de comando:

```
C:\Documents and Settings\mdomitsu>mysql -u root -p bd_musica.sql
```

**mysql** nos pide la contraseña de root:

```
Enter password:
```

Al teclear la contraseña, **mysql** ejecutará el esquema creando las tablas. Para agregarle renglones a las tablas previamente creadas, a partir del esquema *datos\_musica.sql*, tecleando en la línea de comando:

```
C:\Documents and Settings\mdomitsu>mysql -u root -p <  
datos_musica.sql
```

## Creación de un Esquema a partir de una Base de Datos

Para respaldar o para transferir una base de datos de un servidor a otro, podemos obtener un vaciado de la información de una base de datos como un esquema. Para ello, MySQL tiene un programa de utilería llamado **mysqldump**. Para generar un esquema a partir de una base de datos, se sigue el siguiente procedimiento:

1. Abra una ventana de consola como se muestra en la figura 12.
2. La sintaxis para que mysqldump genere un esquema es:

```
mysqldump -u root -p [{-d|-t}] baseDatos > esquema
```

donde *baseDatos* es la base de datos de la que se creará el esquema y *esquema* es el nombre del archivo que contendrá el esquema. La opción **-d** hace que **mysqldump** omita las sentencias para agregar los renglones a las tablas, esto es, sólo generará las sentencias para crear las tablas. La opción **-t** hace que **mysqldump** omita las sentencias para crear las tablas, esto es, sólo generará las sentencias para insertar los renglones en las tablas.

Por ejemplo para generar el esquema *musica.sql*, con las sentencias para crear las tablas canciones y películas, así como las sentencias para insertar los renglones de esas las tablas, teclearemos en la línea de comando:

```
C:\Documents and Settings\mdomitsu>mysqldump -u root -p musica >
musica.sql
```

**mysql** nos pide la contraseña de root:

```
Enter password:
```

Al teclear la contraseña, **mysqldump** generará el esquema.

Para generar el esquema *tablas\_musica.sql*, con sólo las sentencias para crear las tablas canciones y películas, teclearemos en la línea de comando:

```
C:\Documents and Settings\mdomitsu>mysqldump -u root -p -d musica
> tablas_musica.sql
```

Para generar el esquema *datos\_musica.sql*, con sólo las sentencias para insertar los renglones a las tablas canciones y películas, teclearemos en la línea de comando:

```
C:\Documents and Settings\mdomitsu>mysqldump -u root -p -t musica
> datos_musica.sql.
```