	UNIVERSIDAD DON BOSCO **FACULTAD DE INGENIERÍA** **ESCUELA DE COMPUTACIÓN**
CICLO I-2021	**Lenguajes Interpretados en el Servidor** **Guía de práctica No. 9** **Creación de bases de datos y manipulación de datos con MySQL**

I. OBJETIVOS

- Utilice el gestor de bases de datos MySQL tanto localmente como remotamente.
- Desarrolle la habilidad para crear bases de datos y tablas MySQL.
- Adquiera la habilidad en la creación de consultas del lenguaje de manipulación de datos para realizar operaciones con los registros de las tablas.
- Haga uso de sentencias para la definición de restricciones de clave foránea.
- Introducir a la creación de scripts para conexión y consulta de datos de MySQL desde PHP.

II. INTRODUCCIÓN TEÓRICA

Bases de datos

Hoy en día, casi todos los sitios web que manejan información hacen uso de algún tipo de base de datos. Las bases de datos que se utilizan con mayor frecuencia son las bases de datos relacionales. Una base de datos es una herramienta utilizada para almacenar datos. Además, permite crear, leer, actualizar y eliminar esos datos de alguna forma. En los Sistemas de Gestión de Bases de Datos Relacionales (RDBMS: Relational Database Management Systems) los datos se organizan de la siguiente forma:

• Cada RDBMS consta de una o más bases de datos.

• Los datos contenidos en cada base de datos se organizan en una o más tablas.

• Las tablas se organizan en filas y columnas.

• Cada columna representa una porción individual de información con un tipo de dato específico para un registro dado.

• Cada fila representa un único registro de la base de datos.

Las bases de datos, como sistemas de almacenamiento, son mucho más eficientes que los archivos de texto, principalmente porque las bases de datos nos permiten un acceso directo al dato que necesitamos, sin que sea preciso recorrer todo el archivo para encontrarlo.

Cómo es una base de datos

En una base de datos la información se almacena en tablas, las cuales están organizadas en filas y columnas. Las filas son llamadas también registros o tuplas para hacer una analogía con la terminología utilizada en archivos. Cada fila o registro se divide en columnas o campos (así llamados en archivos). Los campos contienen un único dato, preferiblemente indivisibles. Todos los campos de una misma columna conservan la misma estructura, mientras que cada fila contiene información de una sola ocurrencia de la entidad. Esto quiere decir que si se tiene una tabla de empleados de una empresa, todos los campos de una misma fila pertenecen a un solo empleado o que si se tiene una tabla de clientes, todos los campos de una fila de clientes pertenecen a un solo cliente.

Si se desea crear una base de datos para una biblioteca, resulta lógico pensar que sería necesario realizar alguna tabla para almacenar la información de los libros. Por ahora, no pondremos mucha atención en aspectos como la normalización. Dicho esto, en una tabla de libros lo que se necesita almacenar es el título del libro, el autor, la editorial, la edición, el ISBN, el número de páginas, la categoría del libro, y, posiblemente, la ubicación, si se trata de una base de datos de alguna biblioteca que intenta implementar una base de datos.

El lenguaje SQL

SQL son las siglas del Lenguaje Estructurado de Consultas que es el lenguaje utilizado para manipular los datos almacenados en una base de datos. El lenguaje SQL posee una serie de sentencias que en virtud de la función que realizan, se pueden clasificar en tres grupos: 1. El Lenguaje de Definición de Datos (DDL) 2. El Lenguaje de Control de Datos (DCL), y 3. El Lenguaje de Manipulación de Datos (DML).

El Lenguaje de Definición de Datos

Esta parte del lenguaje SQL es el que posibilita la creación y la modificación de los objetos presentes en una base de datos. Las cuatro sentencias principales de este sublenguaje son:

• CREATE,

• ALTER,

• DROP y

• TRUNCATE.

El Lenguaje de Control de Datos

Este sublenguaje contiene los elementos que permiten controlar la integridad, atomicidad y los aspectos de seguridad de los datos. Brinda elementos útiles para el trabajo en ambientes multiusuario en el que juegan un papel importante: la protección de los datos, la seguridad de las tablas, el establecimiento de restricciones de acceso y la compartición de datos en ambientes con usuarios concurrentes. El Lenguaje de Control de Datos se utiliza principalmente para establecer o modificar permisos a los usuarios en función de la base de datos. Las sentencias SQL relacionadas con el control de datos son:
• GRANT,

• REVOKE

El Lenguaje de Manipulación de Datos

Este sublenguaje permite a los usuarios de una base de datos llevar a cabo tareas como consultar o recuperar datos de las tablas, añadir registros, actualizarlos o suprimirlos. Las principales instrucciones son: SELECT, INSERT, UPDATE y DELETE.

Con la sentencia SELECT se pueden recuperar datos de la base de datos utilizando diversas cláusulas y criterios. La sintaxis de esta sentencia se muestra a continuación:

```
SELECT [opciones] elementos [INTO detalles_archivo]
FROM tablas
[WHERE condiciones]
[GROUP BY tipo_grupo]
[HAVING defincion_de_donde]
[ORDER BY tipo_orden]
[LIMIT criterios_limite] [opciones_bloqueo];
```

La sentencia INSERT permite ingresar nuevos registros en las tablas de la base de datos. La sentencia INSERT tiene la siguiente sintaxis:

```
INSERT [INTO] tabla[(campo1, campo2, ..., campoN)]  VALUES(valor1, valor2, ..., valorN);
```

Otra sintaxis válida para la sentencia INSERT es la siguiente:

```
INSERT [INTO] tabla SET campo1=valor1, campo2=valor2, ..., campoN=valorN;
```

La sentencia UPDATE se utiliza para modificar y actualizar los registros de las tablas de la base de datos.

```
UPDATE [LOW_PRIORITY] [IGNORE] tabla
SET campo1=valor1, campo2=valor2, ..., campoN=valorN
[WHERE condicion]
[ORDER BY criterio_orden]
[LIMIT numero];
```

La idea de esta sentencia es actualizar los campos de la tabla a los valores especificados en cada campo dado. Con la cláusula WHERE se puede limitar la aplicación de la operación UPDATE a las filas indicadas por la condición o criterio establecido.

Con la sentencia DELETE se pueden eliminar registros de las tablas de la base de datos.

```
DELETE [LOW_PRIORITY] [QUICK] [IGNORE] FROM tabla
[WHERE condicion]
[LIMIT numero]
[ORDER BY orden_columnas]
[LIMIT numero];
```

El Gestor de Bases de Datos MySQL

MySQL es un poderoso gestor de bases de datos relacionales. Un sistema de gestión de bases de datos es básicamente un programa que se encarga de administrar listas de información. Esta información puede tener diversos orígenes. Puede tratarse de datos de investigación, registros de negocios, solicitudes de clientes, estadísticas deportivas,

informes de ventas, notas de un colegio, registros personales, etc. Cuando se utiliza MySQL se están empleando dos programas que son:
1. El servidor MySQL, que puede identificar como mysqld. El servidor se ejecuta en la computadora o equipo donde se almacenan las bases de datos y se encuentra a la espera de las peticiones de los clientes a través de la red para acceder al contenido de las bases de datos y proporcionar la información solicitada.
2. El cliente MySQL, que es el programa que se conecta con el servidor de las bases de datos y que es donde se ingresan las consultas para indicar la información que se requiere recuperar.

Conectarse al servidor MySQL
Para iniciar sesión en MySQL debe utilizar la interfaz de línea de comandos de su equipo e ingresar el siguiente comando:
En un sistema Linux:
[usuario@localhost]# mysql –uroot –p

En un sistema Windows:
C:\wamp\bin\mysql\mysql5.1.36\bin>mysql –uroot -p

En ambos casos la respuesta del servidor MySQL será:
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 1 to server versión: 5.0.27-community-nt
Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

En este punto estará listo para ingresar las sentencias SQL necesarias para trabajar con la base de datos.

Existe una forma alternativa de conectarse a MySQL desde el programa cliente. Se suele decir que esta es la forma completa y la anterior la resumida. La sintaxis es la siguiente: mysql --host=nombre_host --user=nombre_usuario --password

La instrucción anterior debe ser escrita en la línea de comandos del sistema operativo que esté utilizando.

Crear la base de datos
Para crear una base de datos desde cero, se debe utilizar la sentencia CREATE DATABASE, seguida del nombre de la base de datos. Por ejemplo, para crear una base de datos llamada notas. Debería ingresar el siguiente comando o sentencia en la consola: CREATE DATABASE notas;
Si todo ha ido bien, el servidor le debería mostrar un mensaje como el siguiente:
Query OK, 1 row affected (0.00 sec)

Creación de tablas en la base de datos

Para crear tablas se utiliza la sentencia CREATE TABLE. A continuación y entre paréntesis, se especifican los campos de la tabla, indicando como mínimo el tipo de dato y el ancho del mismo cuando sea necesario. La sintaxis de la sentencia CREATE TABLE se muestra a continuación:

CREATE TABLE nombre_tabla (detalle_de_campos,)

Un ejemplo de creación de tabla se muestra a continuación:

```
CREATE TABLE alumno(
carnet CHAR(8) NOT NULL PRIMARY KEY,
nombre VARCHAR(30) NOT NULL,
apellido VARCHAR(25) NOT NULL,
codcarrera INT UNSIGNED NOT NULL
);
```

Ejemplo de creación de otra tabla:

```
CREATE TABLE carrera(
codcarrera INT UNSIGNED NOT NULL PRIMARY KEY,
carrera VARCHAR(50) NOT NULL,
activa SMALLINT UNSIGNED NOT NULL );
```

Definir relaciones entre tablas

A través de las relaciones es posible indicar que un índice de una tabla está relacionado con un índice de otra. También permite definir restricciones sobre cómo deben ser las relaciones entre tablas. La base de datos es la encargada de hacer que las reglas de esta relación mantengan la integridad referencial. En MySQL el motor de almacenamiento InnoDB proporciona soporte para utilización de claves externas o foráneas. Para que esto funcione correctamente es necesario que:

- La tabla padre debe contener una clave primaria con los valores clave originales.
- La tabla hija debe ser la tabla relacionada a la que harán referencia los valores clave de la tabla padre.
- Los valores clave de la tabla padre se utilizan para asociar dos tablas.

Esto significa que los valores del índice en la tabla hija deben coincidir con los de la tabla padre o, bien, ser nulos para indicar que no hay registro asociado en la tabla padre.

La sintaxis para definir una clave externa o foránea en una tabla hija es la siguiente:

```
[CONSTRAINT nombre_restriccion]
FOREIGN KEY [nombre_indice] (columnas_indice)
REFERENCES nombre_tabla (columna_indice)
[ON DELETE {RESTRICT | CASCADE | SET NULL | NO ACTION}]
 [ON UPDATE {RESTRICT | CASCADE | SET NULL | NO ACTION}] [MATCH FULL | MATCH PARTIAL | MATCH SIMPLE]
```

En el caso de que las tablas estén ya definidas, tendrá que usar una sentencia ALTER TABLE para agregar una restricción de clave foránea.

Ejemplo:

```
ALTER TABLE alumno
ADD CONSTRAINT carrera_alumno
FOREIGN KEY carrera_alumno_fk (codcarrera)
REFERENCES carrera (codcarrera)
ON DELETE RESTRICT
ON UPDATE CASCADE;
```

El efecto que tiene esta restricción es que no se podrá establecer una carrera para el alumno que no sea una de las definidas en la tabla carreras. Cualquier intento de asignar una carrera que no exista a un alumno resultará en un mensaje de error de parte del servidor. Si el caso es cambiar el valor de una de las carreras en la tabla carrera, el efecto será que en la tabla alumno, cualquier carrera relacionada con esta terminará el nombre.

III. MATERIALES Y EQUIPO

Para la realización de la guía de práctica se requerirá lo siguiente:

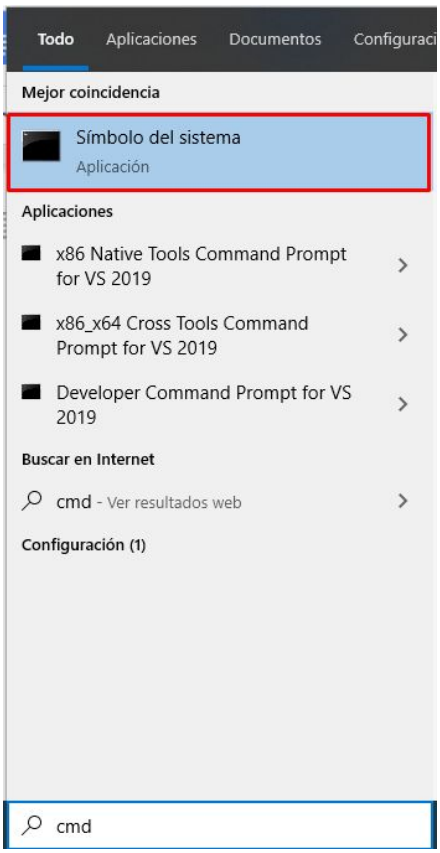
No.	Material	Cantidad
1	Guía de práctica #9: Creación de bases de datos y manipulación de datos con MySQL	1
2	Computadora con WampServer instalado y funcionando correctamente	1
3	Editor PHP sublime Text o Eclipse PHP	1
4	Memoria USB o disco flexible	1

IV. PROCEDIMIENTO

Indicaciones: Asegúrese de digitar el código de los siguientes ejemplos que se presentan a continuación. Tenga en cuenta que el PHP Designer no es compilador solamente un editor. Por lo tanto, los errores de sintaxis los podrá observar únicamente hasta que se ejecute el script cargando la página en el navegador de su preferencia.

Ejercicio 1: Ejecución de sentencias SQL básicas. Iniciará sesión con el usuario root y luego ejecutará algunas sentencias SQL que le darán información del servidor y de las bases de datos administradas en el mismo.

Para arrancar la consola en sistemas operativos con Windows se presiona el botón de inicio y a continuación se digita cmd (o símbolo del sistema) y en el conjunto de resultados se da clic sobre la opción Símbolo del sistema, como se ilustra a continuación:



Una vez cargado el símbolo del sistema de esta forma se debe mover hasta la carpeta donde está el archivo ejecutable de MySQL (mysql.exe). En una instalación con Wamp, MySQL debe estar en la ruta: C:\Wamp\bin\mysql\mysql5.6.17\bin. Deberá moverse hasta esa carpeta o configurar una variable de entorno para que el comando mysql sea reconocido independientemente de la ruta actual en el símbolo del sistema.

```
#Iniciar sesión en el servidor MySQL
C:\wamp\bin\mysql\mysql5.6.17\bin>mysql –hlocalhost –uroot –p
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 1 to server versión: 5.0.27-community-nt

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

#Obtener información de la fecha y hora, el usuario y la versión de MySQL mysql>SELECT now(), user(), version();
+---------------------+--------------------+---------------------+
|now()                |user()              |version()            |
+---------------------+--------------------+---------------------+
|2011-03-27 14:59:15  |root@localhost      |5.0.27 community-nt  |
+---------------------+--------------------+---------------------+

#Usar el mismo comando, pero ahora incorporando un alias para los títulos de las columnas
mysql>SELECT now() AS 'Fecha-Hora', user() AS 'Usuario', version() AS 'Version';
+---------------------+--------------------+---------------------+
|Fecha-Hora           |Usuario             |Version              |
+---------------------+--------------------+---------------------+
|2011-03-27 14:59:15  |root@localhost      |5.0.27 community-nt  |
+---------------------+--------------------+---------------------+
```

#Utilizar funciones matemáticas con MySQL
mysql>SELECT MOD(9,5), POW(5,3), RAND(), SIN(PI()); +-------------+-----------+------------------+--------------------+
|MOD(9,5) |POW(5,3)|RAND() |SIN(PI()) |
+-------------+-----------+------------------+--------------------+
|4 |125 |0.32787095692742 |1.2246063538224e-016 |
+-------------+-----------+------------------+--------------------+

#Escriba ud. la sentencia SQL para realizar la misma operación, pero usando alias para #los nombres de las columnas.
Use Residuo para MOD(), Potencia para POW(), Aleatorio para
#RAND() y Seno para SIN(PI()). Capture la pantalla del resultado que obtenga e incorpórelo
#en un documento que enviará por correo al finalizar la práctica.

#Utilizar funciones de cadena de MySQL
mysql>SELECT CONCAT('Lenguajes ', 'Interpretados ', 'en ', 'el ', 'Servidor') AS 'CONCATENAR', CONV(117,10,16) AS
'CONVERTIR', LOWER('UDB') AS 'MINUSCULAS', REVERSE(UPPER('Don Bosco'))AS 'INVERTIR EN MAYUSCULAS';
+---------------------------------------+----------+----------+-----------------+ |CONCATENAR |CONVERTIR
|MINUSCULAS |INVERTIR_MAYUSC | +---------------------------------------+----------+----------+-----------------+ |Lenguajes
Interpretados en el Servidor |75 |udb |OCSOB NOD |
+---------------------------------------+----------+----------+-----------------+

#Funciones de fecha y hora de MySQL
mysql>SELECT DATE('2013-12-04 10:27:46'), CURTIME(), WEEKDAY('2013-03-27'), MONTH('2013-0325'),
MONTHNAME('2013-01-30'), YEAR('1998-10-12');
#Capture la pantalla de la consola con el resultado e incorpórelo al documento que enviará
#al final de la práctica de laboratorio.

Ejercicio #2: Inicie sesión en la consola de MySQL haciendo uso del usuario root. La contraseña para root está vacía. En este ejercicio del procedimiento crearemos la base de datos almacen y luego verificaremos que la base de datos ha sido creada.

#Iniciar sesión en el servidor MySQL desde la consola abriendo primero símbolo del sistema
#desde Botón de Inicio→Accesorios y no desde el icono del Wamp.
#Debe moverse con comandos DOS hasta el directorio donde está el ejecutable mysql.exe
C:\wamp\bin\mysql\mysql5.6.17\bin>mysql –hlocalhost –uroot –p
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 1 to server versión: 5.0.27-community-nt

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

#Crear una base de datos.
mysql>CREATE DATABASE almacen CHARACTER SET utf8 COLLATE utf8_general_ci;

#Verificar que las base de datos almacén está creada en el servidor
mysql> SHOW databases;
+--------------------------+
| Database |
+--------------------------+
| information_schema |
| almacen |
| mysql |
+--------------------------+

#También es posible utilizar una sintaxis donde se use el operador igual al momento de #crear la base de datos asignando valores a los parámetros CHARACTER SET y COLLATE. #Para probarlo eliminaremos la base de datos antes
mysql> DROP DATABASE almacen;

#Verificamos con SHOW DATABASES que la base de datos almacen ha sido eliminada
mysql> SHOW databases;

```
+------------------------+
| Database               |
+------------------------+
| information_schema     |
| mysql                  |
+------------------------+
```

#A continuación ingresamos el comando CREATE DATABASE como se indicó mysql>

#Volvemos a ejecutar la sentencia SHOW DATABASES para verificar que la bas de datos ha #sido creada nuevamente

```
+------------------------+
| Database               |
+------------------------+
| information_schema     |
| almacen                |
| mysql                  |
+------------------------+
```

#Seleccionar la base de datos de trabajo
mysql> USE almacen;

#Crear tabla bodega en base de datos almacen
mysql> CREATE TABLE bodega (
-> idprodbod int NOT NULL AUTO_INCREMENT,
-> idprodcat int NOT NULL,
-> fecha datetime NOT NULL,
-> precioc decimal(8,2) NOT NULL,
-> preciov decimal(8,2) NOT NULL,
-> unidades int NOT NULL,
-> PRIMARY KEY (idprodbod),
-> KEY idprodcat (idprodcat)
->) ENGINE=InnoDB DEFAULT CHARSET=utf8 COMMENT='Tabla Bodega';
Al presionar ENTER:
Query OK, 0 rows affected (0.34 sec)

#Verificar la estructura de la tabla bodega usando la sentencia DESCRIBE
mysql> DESCRIBE bodega;

```
+-------------+-------------+------+-----+---------+----------------+
| Field       | Type        | Null | Key | Default | Extra          |
+-------------+-------------+------+-----+---------+----------------+
| idprodbod   | int(11)     | NO   | PRI | NULL    | auto_increment |
| idprodcat   | int(11)     | NO   | MUL | NULL    |                |
| fecha       | datetime    | NO   |     | NULL    |                |
| precioc     | decimal(8,2)| NO   |     | NULL    |                |
| preciov     | decimal(8,2)| NO   |     | NULL    |                |
| unidades    | int(11)     | NO   |     | NULL    |                |
+-------------+-------------+------+-----+---------+----------------+
```

#Crear tabla cliente en base de datos almacen
mysql> CREATE TABLE cliente (
-> idcliente int NOT NULL AUTO_INCREMENT,
-> nombre varchar(60) NOT NULL,
-> email varchar(50) NOT NULL,
-> telefono char(8) NOT NULL,
-> PRIMARY KEY (idcliente)
->) ENGINE=InnoDB DEFAULT CHARSET=utf8 COMMENT='Tabla cliente';
Al presionar ENTER:
Query OK, 0 rows affected (0.56 sec)

#Verifique ud. la estructura creada para la tabla cliente. Capture la pantalla de
#la consola y agréguela en el documento donde mostrará el procedimiento de la guía

#Mostrar las tablas de la base de datos que se han creado mysql> SHOW TABLES; +----------------------------+
|Tables_in_almacen |
+----------------------------+
|bodega |
|cliente |
+----------------------------+

#Creación de una tercera tabla que llamaremos categoria para agrupar productos
#de la bodega en esas categorias
mysql> CREATE TABLE categoria (
-> idcategoria int NOT NULL AUTO_INCREMENT PRIMARY KEY,
-> nombrecategoria varchar(36) NOT NULL,
-> descripcion text NULL,
-> INDEX idx_nombrecat (nombrecategoria)
->) ENGINE=InnoDB DEFAULT CHARSET=utf8 COMMENT='Tabla categoria';

#Volver a mostrar las tablas que existen en la base de datos almacen
mysql> SHOW TABLES;
+----------------------------+
|Tables_in_almacen |
+----------------------------+
|bodega |
|categoria |
|cliente |
+----------------------------+

#Agregar un campo a la tabla bodega que contendrá el nombre del producto. El campo
#se agregará justo después del campo idprodcat. Este campo será de tipo varchar con
#un ancho de 50 caracteres
mysql> ALTER TABLE bodega ADD producto varchar(50) NOT NULL AFTER idprodcat;
Query OK, 0 rows affected (0.74 sec)
Records: 0 Duplicates: 0 Warnings 0

#Verficar que el campo ha sido agregado en la tabla producto usando la sentencia DESCRIBE
mysql> DESCRIBE bodega; +-------------+-------------+-------+-------+----------+-------------------+
| Field | Type | Null | Key | Default | Extra |
+-------------+-------------+-------+-------+----------+-------------------+
idprodbod	int(11)	NO	PRI	NULL	auto_increment
idprodcat	int(11)	NO	MUL	NULL	
producto	varchar(50)	NO		NULL	
fecha	datetime	NO		NULL	
precioc	decimal(8,2)	NO		NULL	
preciov	decimal(8,2)	NO		NULL	
unidades	int(11)	NO		NULL	
+-------------+-------------+-------+-------+----------+-------------------+

#Crear una relación entre las tablas categoria y bodega, de modo que la clave o llave #primaria idcategoria de la tabla
categoria esté vinculada como llave foránea en la #tabla bodega y así organizar los productos en categorias
#Primero crear la llave foránea idcategoria en la tabla
mysql> ALTER TABLE bodega ADD CONSTRAINT fk_idproducto
-> FOREIGN KEY (idprodcat) REFERENCES categoria (idcategoria)
-> ON DELETE RESTRICT ON UPDATE CASCADE;
Query OK, 0 rows affected (0.54 sec)
Records: 0 Duplicates: 0 Warnings: 0

#Insertar registros en la tabla cliente usando sintaxis extendida de INSERT
mysql> INSERT INTO cliente (nombre, email, telefono)
-> VALUES ('Karla López', 'karla_lopez05@hotmail.com', '22250694'),
-> ('Nelson Villatoro', 'nvillatoro25@gmail.com', '77255038'),
-> ('Lorena Corado', 'lorencorado@outlook.com', '22211845'),
-> ('José Gutiérrez', 'josegut1@yahoo.es', '22269077'),

-> ('Mario Sorto', 'mariosorto_fcb@fastmail.com', '22604179'),
-> ('Damian Escobar', 'damian_escobar_sv@gmail.com', '78301026'),
-> ('Evelyn Rodezno', 'evelrod@inbox.com', '22158418');
Al presionar ENTER:
Query OK, 7 rows affected (0.12 sec)
Records: 7 Duplicates: 0 Warnings: 0

#Insertar otro registro en la tabla cliente usando sintaxis INSERT INTO ... SET
mysql> INSERT INTO cliente SET nombre='Alejandra Avalos', email='ale_avalosabc@yahoo.com',
-> telefono='22385218';
Al presionar ENTER:
Query OK, 1 row affected (0.05 sec)

#Insertar registros en la tabla cliente usando sintaxis normal del INSERT INTO
mysql> INSERT INTO cliente (nombre, email, telefono) VALUES ('Ricardo Morales', 'richarmor@msn.com', '25906235');
Al presionar ENTER:
Query OK, 1 row affected (0.00 sec)

#Mostrar todos los registros de la tabla cliente que se han insertado
mysql> SELECT * FROM cliente; +----------+----------+----------+----------+
| idcliente | nombre | email | telefono
| +----------+----------+----------+----------+
1	Karla López	karla_lopez05@hotmail.com	22250694
2	Nelson Villatoro	nvillatoro25@gmail.com	77255038
3	Lorena Corado	lorencorado@outlook.com	22211845
4	Jose Gutiérrez	josegut1@yahoo.es	22269077
5	Mario Sorto	mariosorto_fcb@fastmail.com	22604179
6	Damian Escobar	damian_escobar_sv@gmail.com	78301026
7	Evelyn Rodezno	evelrod@inbox.com	22158418
8	Alejandra Avalos	ale_avalosabc@yahoo.com	22385218
9	Ricardo Morales	richarmor@msn.com	25906235
+----------+----------+----------+----------+
9 rows in set (0.00 sec)

#Obtener nuevamente los registros, pero ordenándolos por nombre
mysql> SELECT * FROM cliente ORDER BY nombre; +----------+----------+----------+----------+ |
idcliente | nombre | email | telefono
| +----------+----------+----------+----------+
8	Alejandra Avalos	ale_avalosabc@yahoo.com	23852118
6	Damian Escobar	damian_escobar_sv@gmail.com	78301026
7	Evelyn Rodezno	evelrod@inbox.com	22158418
4	Jose Gutiérrez	josegut1@yahoo.es	22269077
1	Karla López	karla_lopez05@hotmail.com	22250694
3	Lorena Corado	lorencorado@outlook.com	22211845
5	Mario Sorto	mariosorto_fcb@fastmail.com	22604179
2	Nelson Villatoro	nvillatoro25@gmail.com	77255038
9	Ricardo Morales	richarmor@msn.com	25906235
+----------+----------+----------+----------+
9 rows in set (0.00 sec)

#Ingresar 3 registros en la tabla categoria usando INSERT EXTENDIDO
mysql> INSERT INTO categoria (nombrecategoria, descripcion)
-> VALUES ('Cereales', 'Productos de la categoría cereales'),
-> ('Lácteos', 'Productos derivados de la leche'),
-> ('Verduras', 'Productos vegetales');
Query OK, 3 rows affected (0.09 sec)
Records: 3 Duplicates: 0 Warnings: 0

#Agregar un registro más a la tabla categoria usando una sentencia INSERT RESUMIDA mysql> INSERT INTO categoria

-> VALUES (4, 'Frutas', 'Productos de la categoría Frutas');

Query OK, 1 row affected (0.06 sec)

#Ingresar 5 registros de productos en la tabla bodega usando INSERT EXTENDIDO, recordando
#que como hemos establecido una restricción de llave foránea debemos ingresar en el campo
#idprodcat un valor que se corresponda con la llave primaria de la tabla categoria.
#En esa tabla sólo existen cuatro categorias y sólo una de esas cuatro categorías podemos
#ingresar. Los valores que se corresponden con esas categorías son: 1, 2, 3, y 4.
mysql> INSERT INTO bodega (idprodcat, producto, fecha, precioc, preciov, unidades)
-> VALUES (2, 'Leche condensada', '2016-03-20 10:35:08', 2.20, 2.60, 6),
-> (3, 'Apio', '2016-03-18 14:12:25', 0.30, 0.50, 12),
-> (4, 'Mango sazón', '2016-03-26 09:30:08', 0.18, 0.30, 25),
-> (4, 'Piña', '2016-03-28 15:18:23', 0.75, 1.15, 10);
Query OK, 4 rows affected (0.06 sec)
Records: 4 Duplicates: 0 Warnings: 0

#Ingresar un registro de producto más en la tabla bodega
mysql> INSERT INTO bodega (idprodcat, producto, fecha, precioc, preciov, unidades)
-> VALUES (1, 'Frijol rojo', '2016-03-18 12:50:45', 1.00, 1.25, 30);

#Intentar ingresar un nuevo registro de producto en la tabla bodega, pero cuya categoría #no exista actualmente en la tabla categoria. Por ejemplo, un producto con idcategoria=6 mysql> INSERT INTO bodega (idprodcat, producto, fecha, precioc, preciov, unidades)
-> VALUES (6, 'Jugo de naranja', '2016-03-25 13:21:04', 0.75, 0.95, 12);

#Al ejecutar la sentencia anterior el servidor MySQL responderá con un mensaje de error #como el siguiente. Piense ud. la razón y explíquelo en el documento que entregará: ERROR 1452 (23000): Cannot add or update a child row: a foreign key constraint fails ('almacen'.'bodega', CONSTRAINT 'fk_idproducto' FOREIGN KEY ('idprodcat') REFERENCES 'categoria' ('idcategoria')ON UPDATE CASCADE)

#Vaciar la tabla cliente conservando intacta la estructura
mysql> TRUNCATE TABLE cliente;
Al presionar ENTER:
Query OK, 0 rows affected (0.05 sec)

#Intentar consultar filas de tabla cliente nuevamente
mysql> SELECT * FROM cliente;
Al presionar ENTER:
Empty set (0.00 sec)

#Eliminar tabla bodega de la base de datos almacen
mysql> DROP TABLE bodega;
Al presionar ENTER:
Query OK, 0 rows affected (0.00 sec)

#Eliminar tabla cliente de la base de datos almacen
mysql> DROP TABLE cliente;
Al presionar ENTER:
Query OK, 0 rows affected (0.00 sec)

#Eliminar base de datos almacen
mysql> DROP DATABASE almacen;

#Repita todo el procedimiento de este ejercicio 2 para tener de nuevo la base de datos #con sus registros en las tres tablas puesto que se utilizará en el ejercicio 3.

Ejercicio #3: Creación de un usuario.

#Iniciar sesión en el servidor MySQL
#Introduzca las instrucciones necesarias que ya utilizó en los ejercicios 1 y 2 ...

```
mysql>CREATE USER dawsis IDENTIFIED BY 'tecnologico';

#Verificar que el usuario lis ha sido creado accediendo a la base de datos mysql
#donde se crea un registro de los usuarios creado en la tabla user.
mysql>USE mysql;
Database changed
mysql>SELECT * FROM user;

#En los resultados devueltos observará un usuario dawsis. Verifíquelo, aunque costará  #un pocopor la distribución en tabla que se desordena en el modo texto. Note además,  #que la contraseña asignada al usuario LIS aparece encriptada (cifrada).
#Para tener una mejor visualización de los campos que nos interesan de la tabla mysql, #vamos indicar en la sentencia SELECT los campos que nos debe devolver la consulta.
#De la siguiente forma:
mysql>SELECT Host, User, Password FROM user;

#El resultado debe ser una tabla donde se muestren nada más tres campos de la tabla
+-----------------------------------------------------------------------+
| Host    | User      | Password                    | +-------------------------------------------------------------------------+
| localhost | root    |                                  |
| 127.0.0.1 | root    |                                  |
| %       | dawsis    | *3DFCB131FC4171843DB19CF417C72DC2423E6BB8  |
+-----------------------------------------------------------------------+

# Para que podamos autenticar con el usuario de forma local debemos
# crear otros 3 registros que indicarán a MySQL que el usuario tendrá
# acceso local también
mysql> CREATE USER 'dawsis'@'localhost' IDENTIFIED BY 'tecnologico';
Query OK, 0 rows affected (0.00 sec)

# Agregar el usuario 'dawsis' desde el servidor '127.0.0.1'
mysql> CREATE USER 'dawsis'@'127.0.0.1' IDENTIFIED BY 'tecnologico';
Query OK, 0 rows affected (0.00 sec)

# Agregar el usuario 'dawsis' desde el servidor '::1'
mysql> CREATE USER 'dawsis'@'::1' IDENTIFIED BY 'tecnologico';
Query OK, 0 rows affected (0.00 sec)

#Salir de mysql
mysql>quit;

#Iniciar sesión con el usuario recién creado. Recuerde que si accede a MySQL desde
#el icono de Wamp en el área de notificación del sistem sólo podrá hacerlo con
#el usuario root. Así que tendrá que ingresar a la consola de sistema del modo #habitual, desde el icono de inicio de Windows.
C:\wamp\mysql\bin>mysql –udawsis –p
Enter password: ********** ...

#Ejecute alguna sentencia SQL
mysql>SHOW DATABASES;
+----------------------+
|DATABASE              |
+----------------------+
|information_schema              |
|test                  |
+----------------------+
1 row in set (0.00 sec)
#Note que no se muestran todas las bases de datos al usuario dawsis
#Cierre MySQL e inicie sesión nuevamente como usuario root C:\wamp\bin\mysql\mysql5.6.17\bin>mysql –hlocalhost –uroot –p
Enter password:
```

```
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 1 to server versión: 5.0.27-community-nt

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

#Seleccionar como base de datos de trabajo la base de datos almacen
mysql>USE almacen;

#Otorgar permisos al usuario dawsis desde el servidor local
mysql>GRANT ALL ON almacen.* TO 'dawsis'@'localhost' IDENTIFIED BY 'tecnologico'
-> WITH GRANT OPTION;
Query OK, 0 rows affected (0.07 sec)

#Otorgar permisos al usuario dawsis desde cualquier servidor
mysql>GRANT ALL ON almacen.* TO 'dawsis'@'%' IDENTIFIED BY 'tecnologico' WITH GRANT OPTION;
Query OK, 0 rows affected (0.07 sec)

#Volver a salir con el comando quit e iniciar sesión con el usuario dawsis C:\wamp\mysql\bin>mysql –udawsis –p
Enter password: ********** ...

#Ejecute nuevamente la sentencia SHOW DATABASES
mysql>SHOW DATABASES;
+----------------------+
|DATABASE             |
+----------------------+
|information_schema              |
|almacen              |
|test                 |
+----------------------+
2 row in set (0.00 sec)

#Salir de MySQL
mysql>quit Bye

#Ingresar con el usuario root
C:\wamp\mysql\bin>mysql –uroot –p
Enter password:
...

mysql>GRANT ALL PRIVILEGES ON *.* TO 'admin'@'%' IDENTIFIED BY 'admin' WITH GRANT OPTION;
Query OK, 0 row affected (0.00 sec)

mysql>GRANT ALL PRIVILEGES ON *.* TO 'admin'@'localhost' IDENTIFIED BY 'admin' WITH GRANT OPTION;
Query OK, 0 row affected (0.00 sec)

#Salir de MySQL e ingresar con el usuario admin, recién creado C:\wamp\bin\mysql\mysql5.1.36\bin>mysql –uadmin
–p
Enter password: ***** ...

#Ejecutar nuevamente la sentencia SHOW DATABASES
#Notará que ahora le aparecen listadas todas las bases de datos.
mysql> SHOW DATABASES;
+----------------------+
|DATABASE             |
+----------------------+
|information_schema              |
|almacen              |
|clubmembers          |
|datos                |
|mysql                |
|performance_schema              |
```

```
|prueba                |
|test                  |
+----------------------+
8 rows in set (0.00 sec)
```

Ejercicio #4: Creación de una base de datos y todas sus tablas.

```
#Iniciar sesión con el usuario admin en el servidor MySQL
...
#Crear una base de datos llamada peliculas
mysql>CREATE DATABASE peliculas;
Query OK, 1 row affected (0.05 sec)

#Seleccionar la base de datos peliculas
USE peliculas;

#Crear la tabla genero
mysql> CREATE TABLE genero (
-> idgenero int NOT NULL AUTO_INCREMENT,
-> generopelicula varchar(30) NOT NULL,
-> PRIMARY KEY  (idgenero)
-> ) ENGINE=InnoDB;
Query OK, 0 rows affected (0.14 sec)

#Crear la tabla pelicula
mysql> CREATE TABLE pelicula (
-> idpelicula int NOT NULL,
-> titulopelicula varchar(120) NOT NULL,
-> descripcion text NOT NULL,
-> imgpelicula varchar(200) NOT NULL,
-> titulooriginal varchar(60) NOT NULL,
-> duracion varchar(25) NOT NULL,
-> idgenero int NOT NULL,
-> PRIMARY KEY  (idpelicula)
-> ) ENGINE=InnoDB;
Query OK, 0 rows affected (0.06 sec)

mysql> CREATE TABLE director (
-> iddirector int NOT NULL AUTO_INCREMENT,
-> nombre varchar(80) NOT NULL,
-> nacionalidad varchar(30) NOT NULL,
-> PRIMARY KEY (iddirector)
-> ) ENGINE=InnoDB;
Query OK, 0 rows affected (0.09 sec)

mysql> INSERT INTO director (iddirector, nombre, nacionalidad)
-> VALUES
-> (1, 'Chris Columbus', 'Estadounidense'),
-> (2, 'Lee Daniels', 'Estadounidense'),
-> (3, 'Terry Gilliam', 'Estadounidense'),
-> (4, 'Richard LaGravenese', 'Estadounidense'),
-> (5, 'Eric Bress', 'Estadounidense'),
-> (6, 'Barry Sonnenfeld', 'Estadounidense'),
-> (7, 'Anne Fletcher', 'Estadounidense'),
-> (8, 'Frank Darabont', 'Franc'),
-> (9, 'Peter Jackson', 'Neozeland'),
-> (10, 'George Lucas', 'Estadounidense'),
-> (11, 'Manoj Nelliyattu Shyamalan', 'Ind&uacute;'),
-> (12, 'Gabriele Muccino', 'Italiano'),
-> (13, 'Frank Coraci', 'Estadounidense');
```

```
Query OK, 13 rows affected (0.09 sec)
Records: 9 Duplicates: 0 Warning: 0

#Examinar la estructura de las tablas recién creadas
mysql>DESCRIBE genero; +--------------+------------+--------+-------+----------+----------+
|Field         |Type        |Null    |Key    |Default   |Extra     |
+--------------+------------+--------+-------+----------+----------+
|idgenero      |int(11)     |NO      |PRI    |          |          |
|generopelicula|varchar(30) |NO      |       |          |          |
+--------------+------------+--------+-------+----------+----------+
2 rows in set (0.05 sec)

mysql>DESCRIBE pelicula;
#Cargará la estructura de la tabla pelicula

mysql>DESCRIBE director;
#Cargará la estructura de la tabla director

#Crear una restricción de clave foránea después de haber creado las tablas
mysql>ALTER TABLE pelicula
->ADD CONSTRAINT fk_genero_pelicula
->FOREIGN KEY(idgenero)
->REFERENCES genero(idgenero)
->ON DELETE RESTRICT
->ON UPDATE CASCADE;
Query OK, 0 rows affected (0.17 sec)
Records:0 Duplicates:0 Warnings:0

#Insertar registros en la tabla genero
mysql> INSERT INTO genero (idgenero, generopelicula) VALUES (1, 'Acci&oacute;n');
Query OK, 1 row affected (0.66 sec)

mysql> INSERT INTO genero (idgenero, generopelicula) VALUES (2, 'Drama');
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO genero (idgenero, generopelicula)
mysql> VALUES (3, 'Aventuras'),
mysql> (4, 'Comedia Rom&aacute;ntica'),
mysql> (5, 'Suspenso'),
mysql> (6, 'Musical'),
mysql> (7, 'Familiar'),
mysql> (8, 'Infantil');
Query OK, 1 row affected (0.42 sec)

#Insertar registros en la tabla pelicula
mysql> INSERT INTO pelicula (idpelicula, titulopelicula, descripcion, imgpelicula, titulooriginal, duracion, idgenero) VALUES (1, 'Percy Jackson y el Ladr&oacute;n del Rayo', 'La historia narra la vida de un estudiante que descubre ser hijo de Poseid&oacute;n, a ra&iacute;z de esto se ve envuelto en una carrera contra el tiempo para impedir que los dioses griegos inicien una guerra que tiene como campo de batalla el continente americano de hoy en día.', 'img/percy.jpg', 'Percy Jackson & the Olympians: The lightning thief', '119 min', 1);
Query OK, 1 row affected (0.05 sec)

mysql> INSERT INTO pelicula (idpelicula, titulopelicula, descripcion, imgpelicula, titulooriginal, duracion, idgenero) VALUES (2, 'Precious', 'En Harlem, una adolescente analfabeta con sobre peso, quien adem&aacute;s est&aacute; embarazada de su segundo hijo es invitada para inscribirse a una escuela alternativa. Este acontecimiento le da esperanzas de que su vida pueda girar en una nueva direcci&oacute;n.', 'img/precious.jpg', 'Precious: Based on the novel "Push" by Sapphire', '109 min', 2);
Query OK, 1 row affected (0.36 sec)

mysql> INSERT INTO pelicula (idpelicula, titulopelicula, descripcion, imgpelicula, titulooriginal, duracion, idgenero) VALUES (3, 'El Imaginario Mundo Del Doctor Parnassus', 'El Doctor Parnassus tiene una inexplicable capacidad de
```

poder guiar la imaginación de los demás, pero a su vez él guarda un temible secreto. Adicto al juego, muchos años atrás, apostó con el demonio, Mr. Nick, juego que lo convirtió en inmortal. Pero siglos después el doctor conoce a su amor verdadero, y vuelve a realizar otro pacto con el diablo, esta vez intercambiando su inmortalidad por su juventud, con la única condición que cuando su hija llegase a la edad de 16 años, pasaría a ser propiedad del diablo.', 'img/doc.jpg', 'The Imaginarium Of Doctor Parnassus', '122 min', 3);
Query Ok, 1 row affected (0.21 sec)

mysql> INSERT INTO pelicula (idpelicula, titulopelicula, descripcion, imgpelicula, titulooriginal, duracion, idgenero) VALUES (4, 'PD. Te Amo', 'La vida de Holly (Hilary Swank) se ve truncada cuando su marido, Gerry (Gerard Butler), muere. Incapaz de salir adelante por sí misma, su madre y sus amigos intentan animarla. Un día, después de su 30 cumpleaños, Holly recibe una carta de Gerry animándola a salir, a divertirse, a seguir adelante. Cada mes recibirá una carta firmada con un "Posdata: Te amo", que le devolverán las ganas de vivir.', 'img/ps.jpg', 'P.S. I love you', '115 min', 4);
Query OK, 1 row affected (0.09 sec)

mysql> INSERT INTO pelicula (idpelicula, titulopelicula, descripcion, imgpelicula, titulooriginal, duracion, idgenero) VALUES (5, 'Efecto Mariposa', 'Evan Treborn, un joven que se está esforzando por superar unos dolorosos recuerdos de su infancia, descubre una técnica que le permite viajar atrás en el tiempo y ocupar su cuerpo de niño para poder cambiar el curso de su dolorosa historia. Sin embargo también descubre que cualquier mínimo cambio en el pasado altera enormemente su futuro.', 'img/efecto.jpg', 'The Butterfly Effect', '100 min', 5);
Query OK, 1 row affected (0.21 sec)

mysql> INSERT INTO pelicula (idpelicula, titulopelicula, descripcion, imgpelicula, titulooriginal, duracion, idgenero) VALUES (6, 'Vacaciones en familia', 'Un ejecutivo preocupado por no perderse unas vacaciones con su familia decide llevarlos a vacacionar al mismo lugar donde tendrá una importante reunión de trabajo, pero sin decírselos', 'imgvacacionesenfamilia.jpg', 'RV', '98 min', 7);
Query OK, 1 row affected (0.13 sec)

mysql> INSERT INTO pelicula (idpelicula, titulopelicula, descripcion, imgpelicula, titulooriginal, duracion, idgenero) VALUES (7, 'La propuesta', 'Una poderosa editora llamada Margaret (Sandra Bullock) al enfrentarse ante la posibilidad de ser deportada a su país de origen, Canadá, decide comprometerse con su asistente Andrew (Ryan Reynolds) con el propósito de evitarlo', 'imgla-propuesta-poster.jpg', 'The proposal', '108 min', 4);
Query OK, 1 row affected (0.21 sec)

mysql> INSERT INTO pelicula (idpelicula, titulopelicula, descripcion, imgpelicula, titulooriginal, duracion, idgenero) VALUES (8, 'Milagros inesperados', 'La película narra la vida de Paul Edgecomb (Tom Hanks), quien siendo un anciano de 108 años, cuenta su historia como oficial de la Milla Verde, una penitenciaría del estado de Luisiana, durante la década de 1930. Edgecomb cuenta que entre sus presos tuvo un personaje con poderes sobrenaturales, capaz de sanar a personas.', 'img\\greenmille.jpg', 'The Green Mile', '189 min', 2);
Query OK, 1 row affected (0.09 sec)

#Actualizar registros de las tablas
mysql>UPDATE pelicula SET titulopelicula='El efecto mariposa' WHERE pelicula.idpeliculas=5;

#Realizar consultas de más de una tabla
mysql> SELECT titulopelicula AS titulo, generopelicula AS genero, duracion
mysql> FROM pelicula
mysql> JOIN genero
ysql> WHERE pelicula.idgenero = genero.idgenero;

#Realizar consulta de unión en tres tablas
mysql> SELECT titulopelicula, nombre AS director, generopelicula, duracion
mysql> FROM pelicula JOIN director ON pelicula.iddirector = director.iddirector
mysql> JOIN genero ON genero.idgenero = pelicula.idgenero;

#Consulta de unión de varias tablas y filtrado de resultados
mysql> SELECT titulopelicula, nombre AS director, generopelicula, duracion
mysql> FROM pelicula JOIN director ON pelicula.iddirector = director.iddirector
mysql> JOIN genero ON genero.idgenero = pelicula.idgenero

```
mysql> WHERE duracion < 120 ORDER BY titulopelicula;

#Consulta de unión de varias tablas y filtrado de resultados
mysql> SELECT peli.titulopelicula AS Pelicula, dir.nombre AS Director, gen.generopelicula
mysql> AS Genero, peli.duracion AS Duracion
mysql> FROM pelicula AS peli JOIN director AS dir ON peli.iddirector = dir.iddirector  mysql> JOIN genero AS gen ON gen.idgenero = peli.idgenero
mysql> WHERE gen.generopelicula IN ('Familiar', 'Comedia Rom&aacute;ntica')
mysql> ORDER BY peli.duracion

#Realizar consultas en varias tablas haciendo uso de alias para las tablas
msyql> SELECT peli.titulopelicula AS Pelicula, dir.nombre AS Director,
msyql> gen.generopelicula AS Genero, peli.duracion AS Duracion
msyql> FROM pelicula AS peli JOIN director AS dir ON peli.iddirector = dir.iddirector  msyql> JOIN genero AS gen ON gen.idgenero = peli.idgenero
msyql> WHERE peli.duracion < 120 ORDER BY peli.titulopelicula;

#Intentar eliminar uno de los géneros en la tabla genero
mysql>DELETE FROM genero WHERE generopelicula='Familiar';

#Se obtendrá un mensaje de error
ERROR 1451 (23000): Cannot delete or update a parent row: a foreign key constraint fails (peliculas/pelicula', CONSTRAINT 'peliculas_ibfk_1' FOREIGN KEY ('idgenero') REFERENCES 'genero' ('idgenero') ON UPDATE CASCADE.
```

Ejercicio #5: Restaurar una base de datos a partir de un archivo .sql

```
#Iniciar sesión en MySQL con usuario admin
C:\wamp\bin\mysql\mysql5.1.36\bin>mysql –uadmin –p
Enter password: ***** ...

#Crear una base de datos llamada almacen
mysql>CREATE DATABASE almacen;
Query OK, 1 row affected (0.66 sec)

#Salir de MySQL
mysql>quit Bye

#Ejecutar desde la línea de comandos el comando mysql para restaurar la base de datos #la base de datos almacen
C:\wamp\bin\mysql\mysql5.1.36\bin>mysql –uadmin –p almacen < almacen.sql
Enter password: *****
```

Ejercicio #6: Realizar un backup de la base de datos peliculas

```
#Utilizar el comando sqldump para hacer un backup C:\wamp\bin\mysql\mysql5.1.36\bin>mysqldump –uadmin –p peliculas > peliculasbkp.sql Enter password: ***** ...

#Verificar que el archivo ha sido creado. El archivo debe haber sido almacenado en
#la carpeta bin de mysql
```

V. DISCUSIÓN DE RESULTADOS

1. Agregue tres registros más a la tabla película en la base de datos películas. Tome información verdadera de las películas que agregue.

2. Modifique uno de los géneros y observe qué pasa con el campo idgenero en la tabla pelicula que tenía el género modificado.

3. Realice una copia de la base de datos almacen al terminar la práctica de laboratorio.

4. Elimine la base de datos almacen.

5. Restaure de nuevo la base de datos almacen a partir del archivo de copia de seguridad. Edite antes las líneas del archivo de copia de seguridad como se le indicará durante la práctica.

6. Utilice el phpMyAdmin para realizar las tareas que se ejecutaron durante esta práctica pero ahora con una interfaz de usuario orientada a la web.

VI. BIBLIOGRAFIA

- Gutiérrez, Abraham / Bravo, Ginés. PHP 5 a través de ejemplos. Editorial Alfaomega RAMA. 1ra edición. México. Junio 2005.
- Gil Rubio, Francisco Javier/Villaverde, Santiago Alonso/Tejedor Cerbel, Jorge A. Creación de sitios web con PHP 5. Editorial McGraw-Hill. 1ra edición. Madrid, España, 2006.
- John Coggeshall. La Biblia de PHP 5. 1ra Edición. Editorial Anaya Multimedia. Madrid España.
- http://www.php.net/manual/en