



**UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA
FACULTAD DE INGENIERÍA (UNIDAD MEXICALI)
DOCUMENTO DEL SISTEMA DE CALIDAD**

Formato para prácticas de laboratorio

CARRERA	PLAN DE ESTUDIO	CLAVE ASIGNATURA	NOMBRE DE LA ASIGNATURA
IC	2003-1	5046	Bases de Datos

PRÁCTICA No.	LABORATORIO DE	Bases de Datos	DURACIÓN (HORA)
5	NOMBRE DE LA PRÁCTICA	Administración básica de base de datos en Mysql	2

1 INTRODUCCIÓN

Después de conectarnos de manera satisfactoria al servidor de Mysql, el siguiente paso es crear la base de datos que contendrá la o las tablas compuestas por una o mas columnas. En MySql se pueden realizar diferentes operaciones con tablas y columnas así como con la misma base de datos.

2 OBJETIVO (COMPETENCIA)

Practicar las diferentes sentencias que son utilizadas en la creación y manipulación de una base de datos en MySql, mediante la realización de una serie de ejercicios en el laboratorio. Además se efectuarán también diferentes operaciones con tablas y columnas.

Formuló	Revisó M.C. Gloria Etelbina Chavez Valenzuela	Aprobó	Autorizó M.C. Miguel Ángel Martínez Romero
Maestro	Coordinador de la Carrera	Gestión de la Calidad	Director de la Facultad



**UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA
FACULTAD DE INGENIERÍA (UNIDAD MEXICALI)
DOCUMENTO DEL SISTEMA DE CALIDAD**

Formato para prácticas de laboratorio

3 FUNDAMENTO

Creando y usando una base de datos

Supongamos que tenemos diversas mascotas en casa (nuestro pequeño zoológico) y deseamos tener registros de los datos acerca de ellas. Podemos hacer esto al crear tablas que guarden esta información, para que posteriormente la consulta de estos datos sea bastante fácil y de manera muy práctica. Esta sección muestra como crear una base de datos y una tabla.

La base de datos "zoológico" será muy simple (deliveredamente), pero no es difícil pensar en situaciones del mundo real, en las cuales una base de datos similar puede ser usada.

Primeramente usaremos la sentencia SHOW para ver cuáles son las bases de datos existentes en el servidor al que estamos conectados:

```
mysql> SHOW DATABASES;
+-----+
| Database |
+-----+
| mysql   |
| test    |
+-----+
2 rows in set (0.00 sec)
```

Es probable que la lista de bases de datos que veamos sea diferente en nuestro caso, pero seguramente las bases de datos "mysql" y "test" estarán entre ellas. En particular, la base de datos "mysql" es requerida, ya que ésta tiene la información de los privilegios de los usuarios de MySQL. La base de datos "test" es creada durante la instalación de MySQL con el propósito de servir como área de trabajo para los usuarios que inician en el aprendizaje de MySQL.

Se debe anotar también que es posible que no veamos todas las bases de datos si no se tiene el privilegio SHOW DATABASES. Si este es tu caso comentaselo a tu tutor de laboratorio.

Si la base de datos "test" existe, hay que intentar acceder a ella:

```
mysql> USE test
Database changed
mysql>
```

Observar que USE, al igual que QUIT, no requieren el uso del punto y coma, aunque si se usa éste, no hay ningún problema. El comando USE es especial también de otra manera: éste debe ser usado en una sólo línea.



**UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA
FACULTAD DE INGENIERÍA (UNIDAD MEXICALI)
DOCUMENTO DEL SISTEMA DE CALIDAD**

Formato para prácticas de laboratorio

3 FUNDAMENTO

Podríamos usar la base de datos "test" (si tenemos acceso a ella) para los ejemplos que vienen a continuación, pero cualquier cosa que hagamos puede ser eliminada por cualquier otro usuario que tenga acceso a esta base de datos. Por esta razón, es recomendable que preguntemos al administrador MySQL acerca de la base de datos que podemos usar. Supongamos que deseamos tener una base de datos llamada "zoologico" (nótese que no se está acentuando la palabra) a la cual sólo nosotros tengamos acceso, para ello el administrador necesita ejecutar un comando como el siguiente:

```
mysql> GRANT ALL on zoologico.* TO MiNombreUsuario@MiComputadora
-> IDENTIFIED BY 'MiContraseña';
```

en donde MiNombreUsuario es el nombre de usuario asignado dentro del contexto de MySQL, MiComputadora es el nombre o la dirección IP de la computadora desde la que nos conectamos al servidor MySQL, y MiContraseña es la contraseña que se nos ha asignado, igualmente, dentro del ambiente de MySQL exclusivamente. Ambos, nombre de usuario y contraseña no tienen nada que ver con el nombre de usuario y contraseña manejados por el sistema operativo (si es el caso).

Si el administrador creó la base de datos al momento de asignar los permisos, podemos hacer uso de ella. De otro modo, nosotros debemos crearla:

```
mysql> USE zoologico
ERROR 1049: Unknown database 'zoologico'
mysql>
```

El mensaje anterior indica que la base de datos no ha sido creada, por lo tanto necesitamos crearla.

```
mysql> CREATE DATABASE zoologico;
Query OK, 1 row affected (0.00 sec)
```

```
mysql> USE zoologico
Database changed
mysql>
```

Bajo el sistema operativo Unix, los nombres de las bases de datos son sensibles al uso de mayúsculas y minúsculas (no como las palabras clave de SQL), por lo tanto debemos de tener cuidado de escribir correctamente el nombre de la base de datos. Esto es cierto también para los nombres de las tablas.

Al crear una base de datos no se selecciona ésta de manera automática; debemos hacerlo de manera explícita, por ello usamos el comando USE en el ejemplo anterior.



**UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA
FACULTAD DE INGENIERÍA (UNIDAD MEXICALI)
DOCUMENTO DEL SISTEMA DE CALIDAD**

Formato para prácticas de laboratorio

3 FUNDAMENTO

Borrando una base de datos

Para borrar la base de datos utilizamos la sentencia
mysql>Drop database nombre_de_la_base_de_datos;

Creando y visualizando tablas

Crear la base de datos es la parte más fácil, pero en este momento la base de datos está vacía, como lo indica el comando SHOW TABLES FROM:

```
mysql> SHOW TABLES FROM nombre_de_la_base_de_datos;
Empty set (0.00 sec)
```

La parte un tanto complicada es decidir la estructura que debe tener nuestra base de datos: qué tablas se necesitan y qué columnas estarán en cada tabla. En principio, necesitamos una tabla que contenga un registro para cada una de nuestras mascotas. Ésta puede ser una tabla llamada mascotas, y debe contener por lo menos el nombre de cada uno de nuestros animalitos. Ya que el nombre en sí no es muy interesante, la tabla debe contener alguna otra información. Por ejemplo, si más de una persona en nuestra familia tiene una mascota, es probable que tengamos que guardar la información acerca de quien es el dueño de cada mascota. Así mismo, también sería interesante contar con alguna información más descriptiva tal como la especie, y el sexo de cada mascota.

¿Y que sucede con la edad?. Esto puede ser también de interés, pero no es una buena idea almacenar este dato en la base de datos. La edad cambia conforme pasa el tiempo, lo cual significa que debemos de actualizar los registros frecuentemente. En vez de esto, es una mejor idea guardar un valor fijo, tal como la fecha de nacimiento. Entonces, cuando necesitemos la edad, la podemos calcular como la diferencia entre la fecha actual y la fecha de nacimiento. MySQL proporciona funciones para hacer operaciones entre fechas, así que no hay ningún problema por que dichas funciones se estudiarán en la práctica 9 .

Al almacenar la fecha de nacimiento en lugar de la edad tenemos algunas otras ventajas:

Podemos usar la base de datos para tareas tales como generar recordatorios para cada cumpleaños próximo de nuestras mascotas. Podemos calcular la edad en relación a otras fechas que la fecha actual. Por ejemplo, si almacenamos la fecha en que murió nuestra mascota en la base de datos, es fácil calcular que edad tenía nuestro animalito cuando falleció. Es probable que estemos pensando en otro tipo de información que sería igualmente útil en la tabla "mascotas", pero para nosotros será suficiente por ahora contar con información de nombre, propietario, especie, nacimiento y fallecimiento.

Usaremos la sentencia CREATE TABLE para indicar como estarán conformados los registros de nuestras mascotas.

```
mysql> CREATE TABLE mascotas(
-> nombre VARCHAR(20), propietario VARCHAR(20),
-> especie VARCHAR(20), sexo CHAR(1), nacimiento DATE,
-> fallecimiento DATE);
Query OK, 0 rows affected (0.02 sec)
mysql>
```



**UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA
FACULTAD DE INGENIERÍA (UNIDAD MEXICALI)
DOCUMENTO DEL SISTEMA DE CALIDAD**

Formato para prácticas de laboratorio

3 FUNDAMENTO

Para verificar que la tabla fué creada como nosotros esperabamos, usaremos la sentencia DESCRIBE:

```
mysql> DESCRIBE mascotas;
```

Field	Type	Null	Key	Default	Extra
nombre	varchar(20)	YES		NULL	
propietario	varchar(20)	YES		NULL	
especie	varchar(20)	YES		NULL	
sexo	char(1)	YES		NULL	
nacimiento	date	YES		NULL	
fallecimiento	date	YES		NULL	

```
6 rows in set (0.01 sec)
```

```
mysql>
```

Podemos hacer uso de la sentencia DESCRIBE en cualquier momento, por ejemplo, si olvidamos los nombres ó el tipo de las columnas en la tabla.

Alterando tablas

MySQL nos provee del comando ALTER TABLE para realizar modificaciones a la estructura de una tabla. Con este comando nosotros podemos:

- add (agregar) o bien delete (borrar) columnas "ADD y DROP"
- change (cambiar) la definición de tablas existentes "ALTER, CHANGE, MODIFY".
- rename (renombrar) columnas "CHANGE" o aun en si "RENAME AS".

Para agregar una columna a una tabla utilizamos la sentencia

```
ALTER TABLE nombre_de_la_tabla ADD nombre_de_la_columna tipo;
```

Ejemplo 1:

En el sig. ejemplo se agrega a la tabla Tablita un index en la columna d y la columna a se agrega como llave primaria.

```
mysql>ALTER TABLE Tablita ADD index(d), ADD PRIMARY KEY (a);
```

Ejemplo 2:

En el sig. ejemplo se agrega una columna llamada C de tipo entero que se autoincrementa

```
mysql>ALTER TABLE Tablita ADD C INT UNSIGNED NOT NULL AUTO_INCREMENT, ADD INDEX(c);
```

Ejemplo 3:

En el sig. ejemplo agregaremos en la tabla user un campo llamado yahoo de tipo varchar con longitud de 50

```
mysql>ALTER TABLE user ADD yahoo VARCHAR(50) NOT NULL;
```

Se puede utilizar la palabra AFTER para especificar la posición en donde se desea colocar el nuevo campo



**UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA
FACULTAD DE INGENIERÍA (UNIDAD MEXICALI)
DOCUMENTO DEL SISTEMA DE CALIDAD**

Formato para prácticas de laboratorio

3 FUNDAMENTO

Ejemplo 4:

En el sig ejemplo para la tabla user agregamos el campo yahoo exactamente después del campo userid.

```
mysql>ALTER TABLE user ADD yahoo VARCHAR(50) NOT NULL AFTER userid;
```

Para cambiar el tipo de dato de una columna utilizamos la sentencia

```
ALTER TABLE nombre_de_la_tabla MODIFY nombre_de_la_columna nuevo_tipo;
```

Para cambiar una columna de nombre utilizamos la sentencia

```
ALTER TABLE nombre_de_la_tabla CHANGE nombre_actual_de_la_columna nombre_nuevo tipo;
```

Para borrar una columna de una tabla utilizamos la sentencia

```
ALTER TABLE nombre_de_la_tabla DROP nombre_de_la_columna;
```

Para renombrar una tabla utilizamos la sentencia

```
ALTER TABLE nombre_de_la_tabla RENAME nombre_nuevo;
```

o bien

```
RENAME Table nombre_de_la_tabla TO nuevo_nombre;
```

Para borrar una tabla utilizamos la siguiente sentencia

```
DROP TABLE nombre_de_la_tabla;
```

4 PROCEDIMIENTO (DESCRIPCIÓN)

A	EQUIPO NECESARIO	MATERIAL DE APOYO
---	------------------	-------------------



**UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA
FACULTAD DE INGENIERÍA (UNIDAD MEXICALI)
DOCUMENTO DEL SISTEMA DE CALIDAD**

Formato para prácticas de laboratorio

4 PROCEDIMIENTO (DESCRIPCIÓN)

Computadora con acceso al servidor Mysql

Práctica impresa

B DESARROLLO DE LA PRÁCTICA

- 1.-Introduce tu login y password para acceder al servidor de MySQL.
- 2.-Crear una base de datos llamada ZooXXXX donde XXXX es el número de tu matrícula.
- 3.-Nuestra base de datos contendrá las sig. Tablas

Tabla Animales	Tabla Zoologicos	Tabla Veterinarios
Id_Animal (llave)	Id_Zoologico(llave)	Id_Veterinario
Especie	Id_Animal	Id_Zoologico
Hábitat	Nombre_Zoo	Nombre_Vete
Nombre	Tel_Zoo	Cedula
FechaNaci		Dire_Vete
Paísdeorigen		Tel_Vete
Id_Veterinario		

- 4.- Lista las BD existentes
- 5.- Ahora Lista las tablas que contiene la BD ZooXXXX
- 6.- Muestre en pantalla las columnas que contiene la Tabla Animales
- 7.- Liste las columnas que contiene Zoologicos y por último las de Veterinarios
- 8.- Sustituya el nombre de la tabla animales por el de Mascotas
- 9.-Para la tabla Mascotas, cambie el nombre de la columna Paisdeorigen por Origen
- 10.-Agregar una columna nueva a la tabla Mascotas y pongale como nombre Alimentacion
- 11.-Liste las columnas de la Tabla Mascotas
- 12.-Borre la Tabla Veterinarios
- 13.-Liste las tablas existentes en la BD ZooXXXX
- 14.-Cambie la longitud del tipo de dato de la columna Nombre de la tabla Mascotas. Es decir aumentelo tres caracteres
- 15.-Borre la columna Origen de la tabla Mascotas
- 16.-Usted se ha equivocado y realmente no deseaba borrar la columna Origen, por lo tanto agregala de nuevo
- 17.-Para la tabla Mascotas, coloque la columna Id_Veterinario como índice
- 18.-Agregar en la tabla Zoologicos la columna País justo después de la columna Nombre_Zoo
- 19.- Utilizando las sentencias necesarias de Mysql, elabore una BD que tenga mínimo 3 tablas y cada tabla tendrá como mínimo 3 columnas.
- 20.-Liste las bases de datos existemtes
- 21.- Muestre en pantalla las tablas de la BD creada en el punto 19.
- 22.- Borre la base de datos creada en el punto 19
- 23.- Liste las bases de datos existentes
- 24.- Fin



**UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA
FACULTAD DE INGENIERÍA (UNIDAD MEXICALI)
DOCUMENTO DEL SISTEMA DE CALIDAD**

Formato para prácticas de laboratorio

C CÁLCULOS Y REPORTE

5 RESULTADOS Y CONCLUSIONES

Al finalizar la práctica el alumno será capaz de manejar las principales ordenes que se utilizan en MySql para la manipulacion de tablas y columnas en una base de datos sencilla.

6 ANEXOS