



**UAEM**

Universidad Autónoma  
del Estado de México

**CENTRO UNIVERSITARIO VALLE DE CHALCO  
LICENCIATURA EN INGENIERÍA EN COMPUTACIÓN  
UNIDAD DE APRENDIZAJE: FUNDAMENTOS DE BASES DE DATOS**

**Título:**

**DESCRIPCIÓN Y PRÁCTICA DEL LENGUAJE SQL**

**Presenta:**

**CRISTINA JUÁREZ LANDÍN**

**Semestre 2016 A**

# CONTENIDO

1. Identificar los conceptos de Base de Datos
2. Crear modelos lógicos/conceptuales de Base de Datos
3. Aplicar las bases matemáticas que sustentan a los lenguajes para el acceso a una Base de Datos
4. Utilizar las reglas de normalización en un diseño de Base de Datos
5. Analizar el manejo de concurrencia en un Sistema de Base de Datos



# OBJETIVO DE LA UNIDAD DE APRENDIZAJE

- Estudiar los conceptos fundamentales de Base de Datos, para el análisis, el diseño y la implementación de un Sistema de Base de Datos. Conocer la forma en que los sistemas de BD implementan la concurrencia y las formas de implementación para mantener la Base de Datos segura y la integridad de los datos. Todo lo anterior, haciendo uso del lenguaje SQL.



# DESCRIPCIÓN DEL MATERIAL DE LA UA

- Esta presentación esta desarrollada con base a la unidad de aprendizaje (UA) de Fundamentos de Bases de Datos del programa de estudios de Licenciatura en Ingeniería en Computación (ICO) para complementar el curso teórico-practico e ir abordando las prácticas durante el transcurso del semestre.
- Se aborda el 40% del contenido teórico de la UA y se alternara con la práctica al 100% mediante el desarrollo de un proyecto de investigación definido para el curso, se propone una descripción en general del Lenguaje SQL para una mejor comprensión.



## ¿QUÉ ES SQL?

- SQL (Lenguaje de Consulta Estructurada o Structured Query Language por su siglas en inglés) es un lenguaje de programación diseñado para **almacenar, manipular y recuperar** datos almacenados en una base de datos relacionadas.



## SECCIONES COMANDOS.

- **Comandos SQL:** Las instrucciones SQL básicas para almacenar , recuperación y manipulación de datos en una base de datos.
- **Manipulación de tabla:** Cómo se utilizan las SQL para administrar las tablas dentro en una base de datos.



## SECCIONES DE COMANDOS

- **SQL Avanzado:** Comandos SQL avanzados.
- **Sintaxis SQL:** Una página única que enumera la sintaxis para todos los comandos SQL.



# CARACTERÍSTICAS GENERALES DE SQL

- **Lenguaje de definición de datos:**  
Proporciona comandos para la definición de esquemas de relación, borrado de relaciones y modificaciones de los esquemas de relación.





## CARACTERÍSTICAS GENERALES DE SQL

- **Lenguaje interactivo de manipulación de datos:** Incluye lenguajes de consultas basado tanto en álgebra relacional como en cálculo relacional de tuplas (en matemáticas, es una secuencia ordenada de objetos, esto es, una lista con un número limitado de objetos).



## CARACTERÍSTICAS GENERALES DE SQL

- **Integridad:** Incluye comandos para especificar las restricciones de integridad que deben cumplir los datos almacenados en la base de datos.
- **Definición de vistas:** Comandos para definir las vistas.



## CARACTERÍSTICAS GENERALES DE SQL

- **Control de transacciones:** Comandos para especificar el comienzo y el final de una transacción.
- **SQL, incorporado y dinámico:** Esto quiere decir que se pueden incorporar instrucciones de SQL en lenguajes de programación como C++, C, Java, PHP, Cobol, Pascal y Fortran.



# CARACTERÍSTICAS GENERALES DE SQL

- **Autorización:** Comandos para especificar los derechos de acceso a las relaciones y a las vistas.



## TIPO DE DATOS.

- Estos son algunos de los tipos de datos básicos de SQL como numérico, fecha y marca temporal , cadena y espacial , entre las mas usadas son:

**Int:** Es de tipo numérico, se almacena como un entero de 4 bytes y su rango es -2,000,000,000 a 2,000,000,000' aprox.



## TIPO DE DATOS.

**Varchar:** Una cadena de longitud variable en el cual determinamos el máximo de caracteres con un argumento “x” que va entre paréntesis.

Si se omite el argumento coloca 1 por defecto.

Su rango va de 1 a 8000 caracteres.



## TIPO DE DATOS.

**Text:** Guarda datos binarios de longitud variable.

Puede contener hasta 2,000,000,000 caracteres.

No admite un argumento para especificar su longitud.



## TIPO DE DATOS.

**Date:** Se pone la fecha en formato de AAAA-MM-DD, en un intervalo de 0001-01-01 a 9999-12-31. Precisión de un día. Tamaño de almacenamiento de 3 bytes.

No cuenta precisión de fracciones de segundo definido por el usuario y ajuste de zona de horario.





# ¿ QUÉ ES UNA BASE DE DATOS?

- Un conjunto de tablas relacionadas entre si.

**Productos**

Campo 1	Campo 2	Campo 3	Campo 4

**Sucursales**

Campo 1	Campo 2	Campo 3	Campo 4

**Clientes**

Campo 1	Campo 2	Campo 3	Campo 4



## MYSQL: UN GESTOR DE BASE DE DATOS.

- Es un programa que permite crear y mantener una base de datos, asegura su integridad, confidencialidad, seguridad, sencillo, gratuito, rápido y eficiente, ya se para linux y/o windows.



## ¿DONDE PUEDO DESCARGAR MYSQL?

- Hay muchas maneras de descargarlo mysql:
- Sitio web oficial de mysql  
<http://dev.mysql.com/downloads/mysql/>
- Programas que incluye mysql como Xampp, WampServer y ApacheServer que son totalmente gratuitas.



# CUANDO INSTALES MYSQL

- En cualquiera de estos programas no es necesario introducir un usuario y contraseña para mysql (Pero si quieres poner un usuario y/o una contraseña, acuérdate bien de tu usuario y/o contraseña ya que si ella no podrás entrar a mysql).



## ¿COMO PUEDO ENTRAR A MYSQL?

- Desde cmd podrás entra identifica donde esta tu carpeta de mysql (ya se que lo instalaste en el sitio web o programas como xampp, wamp, apacheserver). Ejemplo:
- C:\>mysql
- C:\wamp\bin>mysql
- C:\xampp>mysql
- C:\appsev>mysql



## ¿COMO PUEDO ENTRAR A MYSQL?

- o Nota: Estos son algunos comandos donde se utiliza en cmd:

**cd** : Muestra el nombre o cambia al directorio actual. Ejemplo:

```
C:\> cd mysql
```

```
C:\mysql>
```



## COMANDOS BÁSICOS DE CMD

**cd..** : Esto es para retroceder. Ejemplo:

```
C:\mysql>cd..
```

```
C:\>
```

**dir** : Muestra una lista de archivos y subdirectorios en un directorio. Ejemplo:

```
C:\>dir
```



## COMANDOS BÁSICOS DE CMD

**cls** : Borra los símbolos o el texto en la pantalla de la consola. Se usa como una especie de borrador. Sin opciones, solo es necesario introducir CLS y presionar Enter. Ejemplo:

```
C:\>cls
```





## ¿COMO PUEDO ENTRAR A MYSQL?

- Después encontrar la carpeta bin y dale Enter. Ejemplo:
- C:\mysql\bin>
- Despues escribe la siguiente linea y Enter:  
`mysql -u root -h localhost`
- Ejemplos: C:\mysql\bin\>mysql -u root -p -h localhost



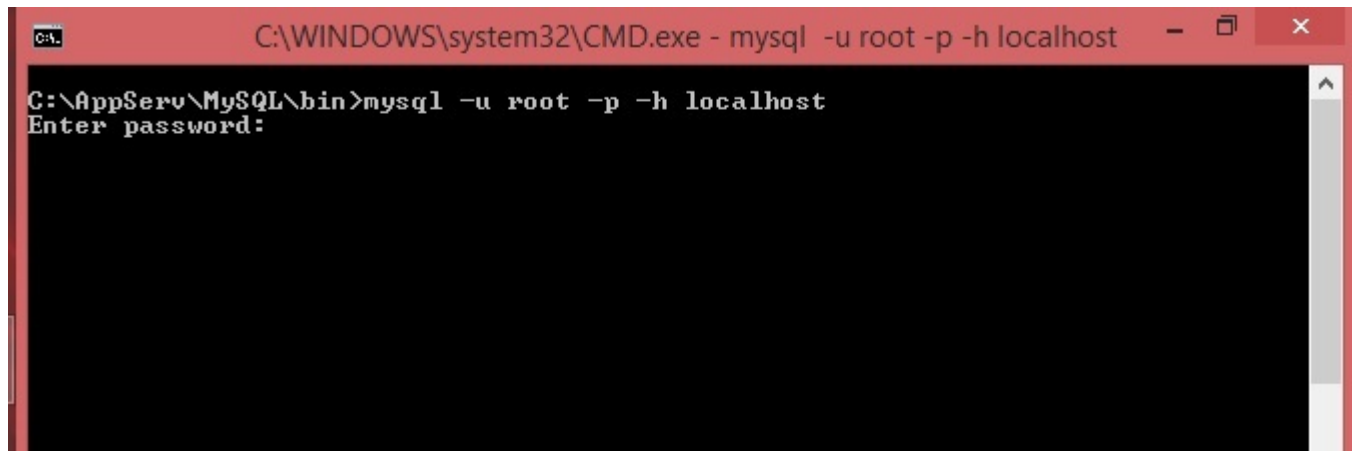
## ¿COMO PUEDO ENTRAR A MYSQL?

- C:\AppServ\MySql\bin>mysql -u root -p -h localhost



## ¿COMO PUEDO ENTRAR MYSQL?

- Después escribe tu contraseña y Enter (Si no tiene contraseña dar Enter).

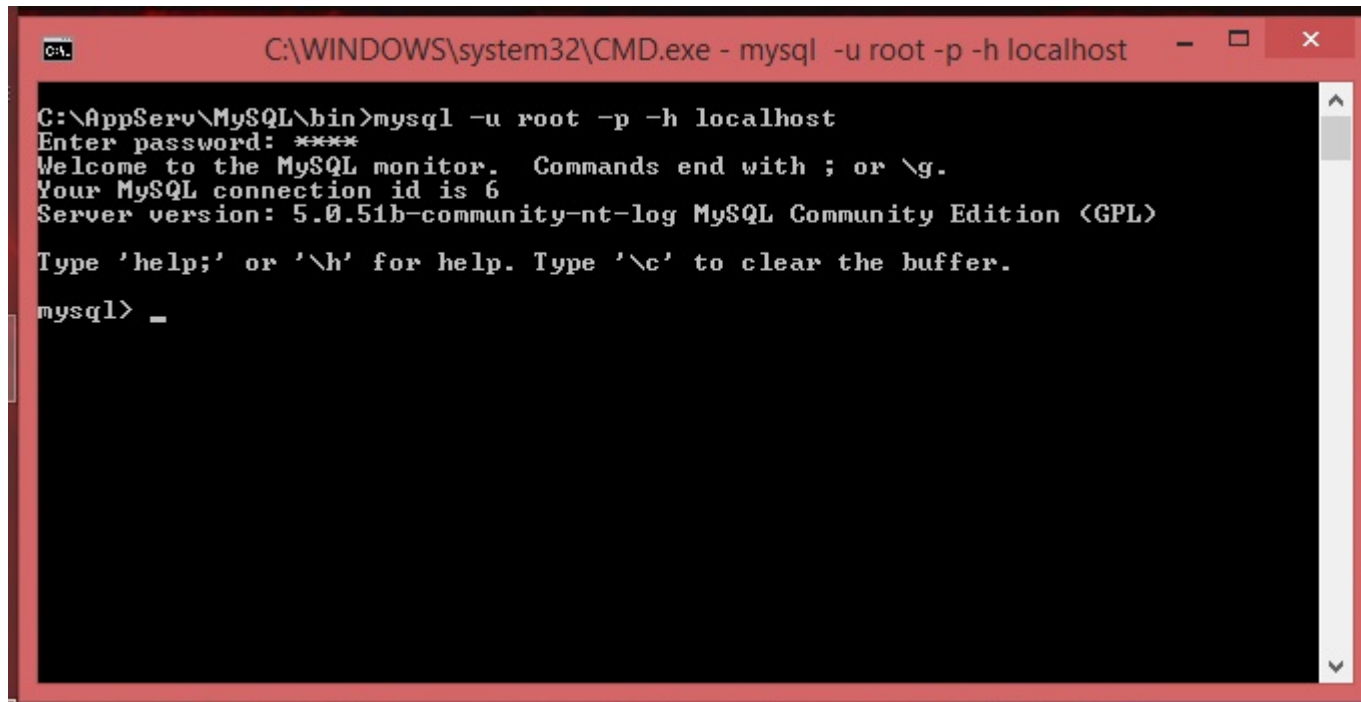


```
C:\WINDOWS\system32\CMD.exe - mysql -u root -p -h localhost
C:\AppServ\MySQL\bin>mysql -u root -p -h localhost
Enter password:
```



# ¿COMO PUEDO ENTRAR A MYSQL?

- o Y listo.



```
C:\WINDOWS\system32\CMD.exe - mysql -u root -p -h localhost
C:\AppServ\MySQL\bin>mysql -u root -p -h localhost
Enter password: ****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 6
Server version: 5.0.51b-community-nt-log MySQL Community Edition <GPL>

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

mysql> _
```



## VER LA ESTRUCTURA DE LA BASE DE DATOS.

- Para ver la(s) estructura(s) de dato(s) (Si es que tiene una) utilizamos el comando de mysql:
- `mysql> show databases;`
- Siempre al final de cada uno de los comandos se pone al final “;”, de lo contrario no saldrá un error o simplemente no podrá realizar el comando o comandos.



# EJEMPLO:

```
C:\Windows\system32\cmd.exe - mysql -u root -p -h localhost
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| hola      |
| mysql    |
| performance_schema |
| prueba   |
| ream2    |
| ream     |
| test     |
+-----+
8 rows in set (0.10 sec)
mysql>
```



## CREACIÓN DE UNA BASE DE DATOS.

- Para crear una base de datos, podremos hacerlo con un comando “**create database**” seguido del nombre de la nueva base de datos. Ejemplo:
  - `mysql>create database miprueba;`



## EJEMPLO:

```
| ream |  
| test |  
+-----+  
8 rows in set (0.10 sec)  
  
mysql> create database miprueba;  
Query OK, 1 row affected (0.03 sec)  
  
mysql> _
```





## CREAR UNA BASE DE DATOS

- Para visualizar la nueva base de datos se escribe el comando anterior . Ejemplo:

```
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| hola        |
| miprueba   |
| mysql      |
| performance_schema |
| prueba     |
| reaem2     |
| ream       |
| test       |
+-----+
9 rows in set (0.00 sec)

mysql>
```



## COMANDO PARA USAR LA BASE DE DATOS.

- Si queremos usar una base de datos que ya fue creada previamente escribimos el siguiente comando “ **use**” seguido del nombre de la base de datos. Ejemplo:
  - `mysql> use miprueba;`



## COMANDO DE VISUALIZACIÓN DE TABLAS.

- Lógicamente, esta base de datos recién creada estará vacía, pero si estuviéramos usando una base de datos ya creada y queremos ver las tablas que tiene escribiéramos el comando “**show tables;**”.

Ejemplo:

- `mysql> show tables;`



## COMANDO DE VISUALIZACIÓN DE TABLAS.

- Si no hay tablas, nos dirá algo como “Empty set”, pero si tenemos varias tablas dadas de alta en la base de datos que estamos usando, nos saldrá una lista de ellas. Ejemplo:

```
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| hola        |
| miprueba   |
| mysql      |
| performance_schema |
| prueba     |
| reaem2     |
| ream       |
| test       |
+-----+
9 rows in set (0.00 sec)

mysql> use miprueba;
Database changed
mysql> show tables;
Empty set (0.00 sec)

mysql> _
```



## COMANDO PARA CREAR TABLAS.

- Para crear una o varias tablas en la base de datos se escribe el siguiente comando “**create table**” seguido del nombre de la tabla, después entre paréntesis se escribe el tipo de dato que se requiera para la tabla (int, double, data, timestamp, varchar, text, etc.), es opcional o si requiere de las opiniones como:
  - [NOT NULL | NULL]
  - [DEFAULT default\_value]



## COMANDO PARA CREAR TABLAS.

- [AUTO\_INCREMENT]
- [UNIQUE[KEY] | [PRIMARY] KEY]
- [COMMENT `string `]



## EJEMPLO:

```
mysql> create table prueba(id_prueba int);  
Query OK, 0 rows affected (0.08 sec)  
  
mysql>
```



## VISUALIZAR LA TABLA

- Para visualizar la tabla “prueba” usaremos el comando anterior. Ejemplo:

```
mysql> create table prueba(id_prueba int);
Query OK, 0 rows affected (0.00 sec)

mysql> show tables;
+-----+
| Tables_in_miprueba |
+-----+
| prueba              |
+-----+
1 row in set (0.000 sec)

mysql>
```





## COMANDO PARA OBTENER INFORMACIÓN DE UNA TABLA

- Ahora, si deseamos obtener información sobre una tabla, para saber que campos tiene y de que tipo, podemos utilizar el comando “**describe**” seguido del nombre de la tabla. Ejemplo:
  - `mysql> describe prueba;`



## EJEMPLO:

```
| prueba |
+-----+
1 row in set (0.00 sec)
mysql> describe prueba;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id_prueba | int(11) | YES | | NULL | |
+-----+-----+-----+-----+-----+-----+
1 row in set (0.01 sec)
mysql>
```



## COMANDO PARA CAMBIAR EL NOMBRE DE UNA TABLA

- Para cambiar el nombre de una tabla en la base de datos se escribe el comando **“rename table”** nombre de la tabla **“to”** el nuevo nombre que quiera poner. Ejemplo:
- `mysql> rename table prueba to nuevo_prueba;`



## EJEMPLO:

```
1 row in set (0.01 sec)

mysql> rename table prueba to nuevo_prueba;
Query OK, 0 rows affected (0.01 sec)

mysql>
```



## VISUALIZACIÓN DEL NUEVO NOMBRE DE LA TABLA

- Para visualizar el nuevo nombre de la tabla escribimos el comando anterior . Ejemplo:

```
mysql> rename table prueba to nuevo_prueba;
Query OK, 0 rows affected (0.01 sec)

mysql> show tables;
+-----+
| Tables_in_miprueba |
+-----+
| nuevo_prueba       |
+-----+
1 row in set (0.00 sec)

mysql>
```



## COMANDO ALTER TABLE EN MYSQL.

- En SQL, “**alter**” es un comando de la categoría DDL(Data Definition Lenguaje) y como su nombre lo indica sirve para alterar objetos ya creados en una base de datos, su uso principal es la modificación de tablas.



## COMANDO ALTER TABLA EN MYSQL

- Comando para añadir otra columna en la tabla “**alter table**” nombre de la tabla “**add(**” nombre de la columna que desea añadir, tipo de dato (int, var, varchar, data, etc.) y cantidad); .
- Ejemplo:
- `mysql> alter table nuevo_prueba add (nombre varchar (30));`



## EJEMPLO:

```
nuevo_prueba add(nombre varchar(30)) at line 1
mysql> alter table nuevo_prueba add(nombre varchar(30));
Query OK, 0 rows affected (0.26 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql>
```





## COMANDO ALTER TABLA EN MYSQL

- Para comprobar si fue añadido utilizamos el comando describe como lo muestra el siguiente ejemplo.

```
mysql> alter table nuevo_prueba add(nombre varchar(30));
Query OK, 0 rows affected (0.03 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> describe nuevo_prueba;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id_prueba  | int(11)       | YES  |     | NULL    |       |
| nombre     | varchar(30)   | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql> █
```



## COMANDO ALTER TABLA EN MYSQL

- Para eliminar columna se utiliza el comando “**alter table**” nombre de la tabla “**drop**” y nombre de la columna a la que va a eliminar ; .
- Ejemplo:
- `mysql> alter table nuevo_prueba drop nombre;`



## EJEMPLO:

```
2 rows in set (0.00 sec)

mysql> alter table nuevo_prueba drop nombre;
Query OK, 0 rows affected (0.03 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> _
```



## COMANDO ALTER TABLA EN MYSQL.

- Para comprobar que se hay eliminado escribimos el comando de describe tabla. Ejemplo:

```
mysql> alter table nuevo_prueba drop nombre;  
Query OK, 0 rows affected (0.03 sec)  
Records: 0 Duplicates: 0 Warnings: 0  
  
mysql> describe nuevo_prueba;  
+-----+-----+-----+-----+-----+-----+  
| Field      | Type      | Null | Key | Default | Extra |  
+-----+-----+-----+-----+-----+-----+  
| id_prueba  | int(11)   | YES  |    | NULL    |      |  
+-----+-----+-----+-----+-----+-----+  
1 row in set (0.00 sec)  
  
mysql> _
```



## COMANDO ALTER TABLA EN MYSQL

- Para modificar la columna de la tabla en mysql se escribe le siguiente comando **“alter table”** nombre de la tabla **“modify”** nombre de la columna y finalmente el tipo de dato al que va a modificar. Ejemplo:
  - 
  - `mysql > alter table nuevo_prueba modify id_prueba int(5) primary key;`



## EJEMPLO:

```
1 row in set (0.26 sec)
mysql> alter table nuevo_prueba modify id_prueba int(5) primary key;
Query OK, 0 rows affected (0.12 sec)
Records: 0 Duplicates: 0 Warnings: 0
mysql> _
```



## COMANDO ALTER TABLA EN MYSQL

- Para visualizar la modificar se escribe el comando anterior (describe ). Ejemplo:

```
mysql> alter table nuevo_prueba modify id_prueba int(5) primary key;
Query OK, 0 rows affected (0.12 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> describe nuevo_prueba;
+-----+-----+-----+-----+-----+-----+
| Field      | Type   | Null | Key  | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id_prueba  | int(5) | NO   | PRI  | NULL    |       |
+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql>
```



## COMANDO ALTER TABLA EN MYSQL

- Para renombrar una columna de la tabla se escribe el siguiente comando “**alter table**” nombre de la tabla “**change**” nombre de la columna y el nuevo nombre que va a cambiar. Ejemplo:
- `mysql > alter table nuevo_prueba change id_prueba id int(5);`





## COMANDO ALTER TABLA EN MYSQL

```
mysql> alter table nuevo_prueba change id_prueba id int(5);  
Query OK, 0 rows affected (0.01 sec)  
Records: 0 Duplicates: 0 Warnings: 0  
mysql> _
```



## COMANDO ALTER TABLA EN MYSQL

- Para visualizar el cambio de la columna de la tabla escribimos el comando anterior( describe). Ejemplo:



## COMANDO TRUNCATE TABLE EN MYSQL

- Este comando es para deshacer de los datos pero no de la tabla en si. Para esto, podemos utilizar el comando “**truncate table**”. Ejemplo:
  - `mysql> truncate table nuevo_prueba;`



## EJEMPLO:

```
mysql> truncate table nuevo_prueba;  
Query OK, 0 rows affected (0.01 sec)  
mysql>
```



## COMANDO DROP TABLE EN MYSQL

- Este comando es para eliminar una tabla en la base de datos y se escribe de la siguiente manera “**drop table**” nombre de la tabla a la cual va a eliminar. Ejemplo:
  - `mysql> drop table nuevo_prueba;`



## COMANDO DROP TABLE EN MYSQL

```
mysql> drop table nuevo_prueba;  
Query OK, 0 rows affected (0.01 sec)  
mysql> _
```



## COMANDO DROP TABLE EN MYSQL

- Para visualizar que ya eliminamos la table escribimos el comando show tables.

Ejemplo:

- `mysql> show tables;`

```
mysql> drop table nuevo_prueba;  
Query OK, 0 rows affected (0.01 sec)  
  
mysql> show tables;  
Empty set (0.00 sec)  
  
mysql> _
```



## COMANDO INSERT INTO EN MYSQL

- Este comando es para inserta una fila por vez y hay otra que es insertar filas múltiples por vez. El comando para insertar datos en una tabla mediante una fila por vez es **“insert into”** nombre de la tabla **“(columna 1, columna 2, ... )”** **“values”** **“(valor 1, valor 2, ... )”**;. Ejemplo:
  - `mysql> insert into(id) values(1);`





## COMANDO INSERT INTO EN MYSQL

```
mysql> insert into nuevo_prueba(id) values(1);  
Query OK, 1 row affected (0.00 sec)  
  
mysql> _
```



## COMANDO INSERT INTO EN MYSQL

- Para visualizar el contenido de la tabla escribimos el comando “**select \* from**” nombre de la tabla;. Ejemplo:
- `mysql> select * from nuevo_prueba;`



## COMANDO INSERT INTO EN MYSQL

```
mysql> select * from nuevo_prueba;
+----+
| id |
+----+
|  1 |
+----+
1 row in set (0.00 sec)

mysql> _
```



## CONEXIONES .

- SQL interactúa con diferentes tipos de lenguajes de programación en esta ocasión se va relacionar MySql con los lenguajes de PHP y HTML.



## CONEXIONES.

- Un ejemplo sencillo es la de guardar(agregar) datos de una tabla de mysql desde html y php, pero antes de realizar necesitamos una base de datos con una tabla en mysql.



## CONEXIONES .

- Debes de crear una base de datos en mysql(como los ejemplos anteriores) con una tabla llamada contacto y en ella se agregaran los campos de id, nombre, ap, am, nu\_cuenta, como se muestra siguiente ejemplo:



## EJEMPLO:

```
mysql> describe contacto;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| id         | int(11)       | NO   | PRI | NULL    | auto_increment |
| nombre    | varchar(30)   | YES  |     | NULL    |                |
| ap        | varchar(30)   | YES  |     | NULL    |                |
| am        | varchar(30)   | YES  |     | NULL    |                |
| nu_cuenta | int(8)        | YES  |     | NULL    |                |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.03 sec)

mysql>
```



## CONEXIONES.

- Ya que esta nuestra tabla vamos agregar un nombre, ap(apellido paterno), am( apellido materno), nu\_cuenta (numero de cuenta), el id no se pone por que es automáticamente incrementado.





## CONEXIONES .

- Luego de hacer nuestra base de datos en mysql vamos a buscar una carpeta de wampserver o appserv o xampp llamada www y agregamos una carpeta. Después agregamos un archivo(con el bloc de notas o un editor su preferencia) y lo guardamos como index.html.



## CONEXIONES.

- En el archivo index.html agregamos el siguiente código:

- `<html>`

- `<body>`

- `</body>`

- `</html>`



## CONEXIONES.

- Dentro de los body escribimos el siguiente código:
- `<form action="nombre del archivo.php" method="get" name="formulario" enctype="multipart/form-data">`



## CONEXIONES.

- Creamos varias cajas de textos llamados:
- Nombre `<input type="text" id="campo1" name="nombre" /> </br>`
- Apellido Paterno `<input type="text" id="campo2" name="ap" /> </br>`



## CONEXIONES.

- Apellido Materno `<input type="text" id="campo3" name="am" /> </br>`
- Numero de cuenta `<input type="text" id="campo4" name="nu_cuenta" /> </br>`



## CONEXIONES.

- Se crea un botón:
- `<input id="campo5" name="guardar" type="submit" value="guardar" />`



## CONEXIONES.

- Se crea otro archivo con .php(Ejemplo: index.php) y se escribe el siguiente código:
- `<html>`
- `<body>`
- `</body>`
- `</html>`



## CONEXIONES.

- Dentro del body se escribe el lenguaje php para conectar a mysql y se escribe el siguiente código:
- `<?php`
- `$conexion =mysql_connect("localhost",`  
“ Aquí se escribe el usuario(Si es que tiene uno)”, “ Aquí se escribe la contraseña(Si es que tiene uno)”);





## CONEXIONES.

- `$bd = mysql_select_db(" Aquí se escribe el nombre de la base de datos", $conexion);`
- `$nombre = $_GET['nombre'];`
- `$ao = $_GET['ap'];`
- `$am = $_GET['am'];`
- `$nu_cuenta = $_GET['nu_cuenta'];`



## CONEXIONES.

- `$boton = $_GET['guardar'];`
- `if (isset($boton)) {`
- `$sql = "INSERT INTO contacto(nombre, ap, am, nu_cuenta) VALUES(" . utf8_encode($nombre) . ", " . utf8_encode($ap) . ", " . utf8_encode($am) . ", " . utf8_encode($nu_cuenta) . ")";`



## CONEXIONES.

- `$query = mysql_query($sql, $conexion);`
- `?>`



## CONEXIONES.

- `<div id= "contenedor">`
- `<div id="nombre"><?php echo $nombre; ?></div>`
- `<div id="ap"><? echo $ap ?></div>`
- `<div id="am"><? echo $am ?></div>`
- `<div id="nu_cuenta"><? echo $nu_cuenta ?></div>`
- `</br></br> <div id="mensaje"> Datos guardados</div>`
- `</div>`



## CONEXIONES.

- Y por ultimo:
- `</div>`
- `<?php`
- `} else {`
- `}`
- `?>`



## CONEXIONES.

- Si esta todo bien tendremos la base de datos de mysql con los archivos index.html y index.php.
- Para compilarlos vamos a nuestro navegador y escribimos localhost/nombre de la carpeta(si es que hay una)/index.html.
- Ejemplo:
- localhost/index.html
- localhost/sql/index.html



# CONEXIONES.

- Ejemplo:



A screenshot of a web browser window. The address bar shows the URL `localhost/SQL/index.html`. The page content includes a form with the following fields and a button:

- Nombre
- Apellido Paterno
- Apellido Materno
- Numero de cuenta
-

## CONEXIONES.

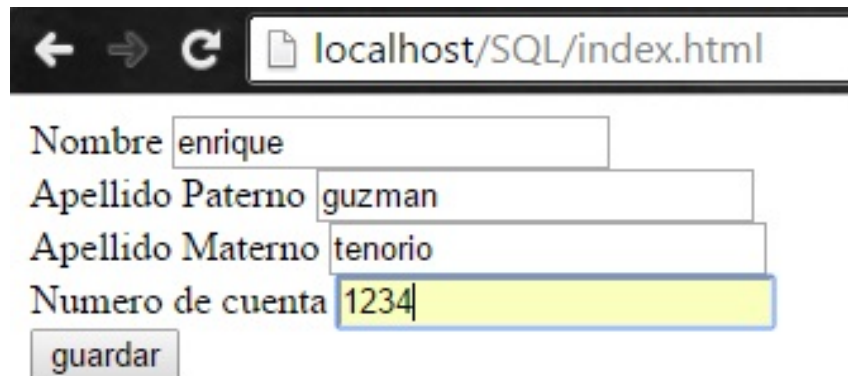
- Agregamos los datos y de damos en el botón de guardar y listo si nos vamos a la base de datos en mysql veremos que efectivamente ahí esta lo que se agrego.





# CONEXIONES.

Ejemplo:



A screenshot of a web browser window showing a form. The address bar displays 'localhost/SQL/index.html'. The form contains four input fields: 'Nombre' with the value 'enrique', 'Apellido Paterno' with 'guzman', 'Apellido Materno' with 'tenorio', and 'Numero de cuenta' with '1234'. The 'Numero de cuenta' field is highlighted in yellow. Below the fields is a 'guardar' button.

```
| 16 | enrique | guzman | tenorio | 1234 |  
+-----+-----+-----+-----+  
5 rows in set (0.00 sec)  
mysql>
```



## REFERENCIAS BIBLIOGRÁFICAS

- Microsoft (2016). Sql. Consultado de: <http://www.microsoft.com/es-xl/server-cloud/products/sql-server/default.aspx> el 5 de agosto de 2016.
- Plan de estudios de ICO. Consultado de: <http://www.uaemex.mx>

