

GUIA N° 7 – COMANDOS MYSQL

I. CREAR, SELECCIONAR, VISUALIZAR

1. CREAR BASE DE DATOS
`CREATE DATABASE Nombre_Base_Datos;`
2. VER LISTADO DE BASES DE DATOS
`SHOW DATABASES;`
3. USAR UNA BASE DE DATOS
`USE NOMBRE_BASE_DATOS;`
4. ELIMINAR UNA BASE DE DATOS
`DROP DATABASE NOMBRE_BASE_DATOS;`
5. VER LISTADO DE TABLAS
`SHOW Tables;`
6. MOSTRAR LA ESTRUCTURA DE UNA TABLA
`DESCRIBE Nombre_Tabla;`



II. CREAR UNA TABLA

```
CREATE TABLE Nombre_Tabla
(
  Campo_1 Tipo1 PRIMARY KEY Not Null AUTO_INCREMENT,
  Campo_2 Tipo2(Tamaño),
  :
  :
  Campo_n Tipon(Tamaño)
);
```

Nota: `AUTO_INCREMENT` se utiliza cuando el campo se necesita definir como `AUTONUMERICO`

III. TIPO DE DATOS BASICOS EN TABLAS - MY SQL

Tipo	Bytes	Descripción
INT o INTEGER	4	Números enteros. Existen otros tipos de mayor o menor longitud
BIGINT	8	Numero entero grande
DOUBLE o REAL	8	Números reales (grandes y con decimales). Permiten almacenar todo tipo de número no entero.
CHAR	1/caracter	Alfanuméricos de longitud fija predefinida
VARCHAR	1/caracter+1	Alfanuméricos de longitud variable
DATE	3	Fechas, existen múltiples formatos
BIT o BOOLEAN	1	Almacenan un bit de información (verdadero o falso)

	GUÍA DE TRABAJO N° 7 – GRADO 11		
	Articulación SENA	Programación de Software	
	Ing. Néstor Raúl Suarez Perpiñan Página 2 de 6		

IV. ALTERACIONES SOBRE UNA TABLA

1. AGREGAR UN CAMPO A UNA TABLA

```
ALTER TABLE Nombre_tabla ADD Nombre_Campo tipo;
```

2. ELIMINAR UN CAMPO DE UNA TABLA

```
ALTER TABLE Nombre_tabla DROP COLUMN Nombre_Campo;
```

3. CAMBIAR NOMBRE DE UN CAMPO

```
ALTER TABLE Nombre_tabla CHANGE Nombre_Campo Nuevo_Nombre tipo;
```

4. CAMBIAR TIPO DE UN CAMPO

```
ALTER TABLE Nombre_tabla MODIFY Nombre_Campo Nuevo_Tipo;
```

5. ELIMINAR UNA TABLA

```
DROP TABLE Nombre_tabla;
```

6. REINICIAR UNA TABLA

```
TRUNCATE TABLE Nombre_tabla;
```

7. CAMBIAR NOMBRE DE UNA TABLA

```
RENAME TABLE Nombre_tabla TO Nuevo_Nombre_Tabla;
```

V. RELACIONES ENTRE TABLAS

1. AGREGAR UNA CLAVE FORANEA

```
ALTER TABLE Nombre_tabla ADD KEY Fk_NombreClave (CampoClaveForanea);
```

2. CREAR UNA RELACIÓN

```
ALTER TABLE Tabla_Foranea ADD
CONSTRAINT FK_TablaForanea_TablaPrimaria
FOREIGN KEY (CampoFK)
REFERENCES Tabla_Primary (CampoPK);
```

3. VER ESPECIFICACIÓN DE UN TABLA (Listado de Relaciones)

```
SHOW CREATE TABLE Nombre_tabla
```



4. ELIMINAR UNA RELACIÓN:

```
ALTER TABLE Nombre_tabla DROP FOREIGN KEY nombre_del_fk;
Donde: nombre_del_fk normalmente es FK_TablaForanea_TablaPrimaria
```

VI. MANIPULACION DE DATOS

1. INSERTAR DATOS

```
INSERT INTO Nombre_tabla (campo1, campo2...)
VALUES ('valor 1', 'valor 2'...);
```

	GUÍA DE TRABAJO N° 7 – GRADO 11		
	Articulación SENA	Programación de Software	
	Ing. Néstor Raúl Suarez Perpiñan Página 3 de 6		

2. MODIFICAR DATOS

```
UPDATE Nombre_tabla SET campo1 = 'valor1', campo2 = 'valor2' ...
WHERE Condición;
```

3. ELIMINAR DATOS

```
DELETE FROM Nombre_tabla WHERE Condición;
```

4. CONSULTAR DATOS

```
SELECT * FROM Nombre_tabla;
SELECT campos FROM Nombre_tabla WHERE Condición;
```

Cuando se tienen varias condiciones, se pueden utilizar los operadores lógicos en conjunto con los operadores de comparación:

Operadores lógicos: AND (&), OR (|) NOT.

Operadores de comparación:

= comparador de equidad
 < Menor que
 > Mayor que
 >= Mayor o igual que
 <= Menor o igual que
 (<>, !=, not) Diferente de

Ejemplo de consulta:

```
SELECT * FROM tabla WHERE (a=1 AND B>5 OR C='Mil');
```



El comando de consulta también se puede utilizar con la cláusula LIKE, que permite manejar patrones, es decir, se especifica un patrón para realizar las comparaciones, y los registros que tengan ese patrón en el atributo especificado se van a obtener independientemente mayúscula o minúscula.

El uso de esta cláusula es de la siguiente manera: ('%patrón%'), esto especifica que no importa en donde aparezca ese patrón, incluso puede no existir, es decir no importa que tenga atrás, ni que tenga delante del patrón, se va a obtener el registro. ('patrón%'), Este uso define que todos los registros que empiecen con el patrón especificado se van a obtener. (%patrón), de la misma manera, que el anterior a diferencia que entre son los que terminan.

Ejemplo:

```
SELECT * FROM persona WHERE (Apellidos LIKE 'Perez%');
```

En este ejemplo se obtienen todos los registros de la tabla persona que empiezan con Perez, no importa que tenga después.

	GUÍA DE TRABAJO N° 7 – GRADO 11		
	Articulación SENA	Programación de Software	
	Ing. Néstor Raúl Suarez Perpiñan Página 4 de 6		

Cuando existen varias condiciones sobre un mismo atributo en una sentencia es tedioso estar especificando cada condición sobre el atributo, para eliminar esta problemática y que las sentencias no se tornen muy grandes, SELECT se combina con la cláusula IN que recibe un conjunto de valores en los cuales se van a evaluar las condiciones

Ejemplo: **SELECT * FROM** tabla **WHERE** (id [NOT] IN (valor1, valor2, valor3,....,valorn))

EJEMPLOS DE CONSULTAS MySQL

- ✓ Para seleccionar sólo los campos que cumplan una determinada condición:

```
SELECT * FROM usuario WHERE edad < 18;
```

- ✓ Para agrupar por un determinado campo:

```
SELECT ciudad, COUNT(ciudad) FROM usuario GROUP BY ciudad;
```

- ✓ La salida de un GROUP BY está ordenada según los valores del campo sobre el que se agrupa. La ordenación se realiza por defecto en orden ascendente (ASC) para hacerlo en orden descendente se añade la palabra clave DESC:

```
SELECT ciudad, COUNT(ciudad) FROM usuario GROUP BY ciudad DESC;
```

- ✓ Para seleccionar solo algunos campos de entre el resultado de un GROUP BY se puede utilizar la cláusula HAVING:

```
SELECT ciudad, COUNT(ciudad) FROM usuario GROUP BY ciudad DESC HAVING COUNT(ciudad)<1
```

- ✓ Para ordenar según los valores de un campo:



```
SELECT * FROM usuario ORDER BY nombre;
```

- ✓ Como GROUP BY, se ordena de forma ascendente, a menos que se utilice la palabra clave DESC:

```
SELECT * FROM usuario ORDER BY nombre DESC;
```

- ✓ Para limitar el número de registros que queremos que devuelva SELECT se utiliza LIMIT:

```
SELECT * FROM usuario LIMIT 2;  
SELECT * FROM usuario LIMIT 2,2;
```

	GUÍA DE TRABAJO N° 7 – GRADO 11		
	Articulación SENA	Programación de Software	
	Ing. Néstor Raúl Suarez Perpiñan Página 5 de 6		

VII. FUNCIONES DE AGREGACIÓN

Funciones de agregación

COUNT(<fila>) {devuelve el total de filas seleccionadas}

SUM(<columna>) {suma los valores de una columna}

MIN(<columna>) {devuelve valor mínimo columna}

MAX(<columna>) {devuelve valor máximo columna}

AVG(<columna>) {devuelve la mediana de la columna}

...

• Ejemplos:

```
SELECT COUNT(*)
FROM alumnos;
```

```
SELECT AVG(edad)
FROM alumnos
WHERE curso_actual = 3;
```

VII. MANUAL ONLINE DE MYSQL

<http://dev.mysql.com/doc/refman/5.0/es/index.html>

<http://mundogeek.net/archivos/2009/02/16/mysql/>

TALLER

1. Usando MYSQL y los comandos mostrados en esta guía implemente una base de datos que corresponda con el Modelo Relacional de una “Biblioteca” propuesto

2. Una vez terminado realice las siguientes tareas sobre las tablas correspondientes:
 - a) Ingresar, consultar, actualizar y eliminar los datos de la institución educativa(Colegio)
 - b) Ingresar, consultar, actualizar y eliminar los datos de la biblioteca
 - c) Ingresar, consultar, actualizar y eliminar materias o asignaturas
 - d) Ingresar, consultar, actualizar y eliminar libros
 - e) Ingresar, consultar, actualizar y eliminar los clientes (estudiantes)
 - f) Ingresar, consultar, actualizar y eliminar préstamos de libros

3. Implemente en MYSQL la Base De Datos Correspondiente a su Proyecto SENA

