

MySQL

El SQL de MySQL

El SQL de MySQL

- ❑ La forma que crearemos y manipularemos la información es por medio de un lenguaje llamado SQL (Structured Query Language), diseñado por IBM en los años setenta.
- ❑ El lenguaje se divide, en términos generales, en los siguientes módulos:

El SQL de MySQL

- ❑ Selección de información (Data Query Language o DQL)
- ❑ SELECT

El SQL de MySQL

- ☐ Modificación de información de tablas
- ☐ (Data Modification Language o DML)
- ☐ DELETE
- ☐ INSERT
- ☐ LOAD DATA
- ☐ OPTIMIZE TABLE
- ☐ REPLACE
- ☐ UPDATE

El SQL de MySQL

- ☐ Creación, vaciado y selección de bases de datos
- ☐ (Data definition language DDL)
- ☐ CREATE DATABASE
- ☐ DROP DATABASE
- ☐ USE

El SQL de MySQL

- ☐ Creación, alteración y vaciado de tablas e índices
- ☐ ALTER TABLE
- ☐ CREATE INDEX
- ☐ CREATE TABLE
- ☐ DROP INDEX
- ☐ DROP TABLE

El SQL de MySQL

- ❑ Sentencia de administración
- ❑ FLUSH
- ❑ GRANT
- ❑ KILL
- ❑ REVOKE

El SQL de MySQL

- ❑ Obtención de información sobre base de datos y consultas
- ❑ DESCRIBE
- ❑ EXPLAIN
- ❑ SHOW

El SQL de MySQL

- ❑ *Mísceláneas de sentencias*
- ❑ CREATE FUNCTION
- ❑ DROP FUNCTION
- ❑ LOCK TABLES
- ❑ SET
- ❑ UNLOCK TABLES

MySQL

Crear una base de datos

Crear una base de datos

- ❑ En MySQL se pueden crear varias bases de datos, y dentro de estas varias tablas. Para crear una base de datos se utiliza la sentencia
- ❑ `CREATE DATABASE nombre
CHARACTER SET utf8 COLLATE
utf8_general_ci`

Eliminar una base de datos

- ❑ En MySQL se puede eliminar una base de datos con el comando:
- ❑ `DROP DATABASE nombre`

Mostrar las bases de datos

- ❑ En MySQL se puede mostrar las bases de datos con el comando:
- ❑ `SHOW DATABASES`

MySQL

Crear una tabla en una base de datos

Crear una tabla

- ❑ CREATE TABLE nombre (definición de columnas)
- ❑ Los tipo de las columnas ya se vieron en el capítulo anterior, pero los principales son:

Crear una tabla

- ❑ Integer: Número entero con rango de -2,147,483,648 y 2,147,483,647
- ❑ VarChar(n): Es una cadena con longitud variable, con un máximo de 255 caracteres. El parámetro n marca el máximo de la longitud esperada.

Crear una tabla

- ❑ Text: Campo de texto que puede almacenar hasta 65,535 caracteres.
- ❑ MediumText: Campo de texto que permite almacenar hasta 16,777,215 caracteres.

Crear una tabla

- ❑ Datetime: Guarda una fecha en el formato YYYYDDMMHHMMSS
- ❑ Timestamp: Tipo de campo que almacena la fecha actual en formato YYYYDDMMHHMMSS

Crear una tabla

- ❑ Cada uno de los campos o columnas podrá tener alguno de los siguientes comportamientos:
- ❑ NOT NULL | NULL: Permite que el campo tenga o no valor vacíos o nulos.
- ❑ DEFAULT: Valor que tomará el campo por default.
- ❑ AUTO INCREMENT: Se le asignará al campo un número a partir de l valor más alto de esa columna más uno.
- ❑ PRIMARY KEY: Es la llave principal de la tabla, por lo que no puede haber valores duplicados o nulos. Es una buena práctica tener un campo con llave principal en cada tabla.

Crear una tabla

```
);  
  
CREATE TABLE alumnos(  
    id INTEGER AUTO_INCREMENT PRIMARY KEY,  
    nombre VARCHAR(30) NOT NULL,  
    apellidos VARCHAR(30) NOT NULL,  
    nacimiento DATE,  
    promedio TINYINT,  
    sexo CHAR(1),  
    idEscuela TINYINT,  
    idSalon TINYINT,  
    idCurso TINYINT  
);
```

Crear una tabla

```
CREATE TABLE escuelas(  
  id INTEGER AUTO_INCREMENT PRIMARY KEY,  
  descripción VARCHAR(60)  
);
```

```
CREATE TABLE salon(  
  id INTEGER AUTO_INCREMENT PRIMARY KEY,  
  descripción VARCHAR(60)  
);
```

Crear una tabla

```
CREATE TABLE cursos (
```

```
    id INTEGER
```

```
    AUTO_INCREMENT PRIMARY KEY,
```

```
    descripción VARCHAR(60)
```

```
);
```

MySQL

Mostrar las tablas de una base de datos

Mostrar las tablas de una base de datos

- ❑ Para que usted pueda ver las tablas existentes, necesita ejecutar el siguiente comando:
- ❑ `SHOW tables;`

MySQL

Borrar una tabla de la base de datos

Borrar las tablas de una base de datos

- ❑ Para borrar una tabla o una base de datos basta con la sentencia
- ❑ `DROP TABLE nombreTabla;`

MySQL

Insertar datos a una tabla

Insertar datos a una tabla

- ❑ Para insertar datos a una tabla, necesitamos la siguiente instrucción:
- ❑ `INSERT INTO tabla (columnas)`
`VALUES (valores);`

Insertar datos a una tabla

```
❑ INSERT INTO alumnos (nombre,  
    apellido, nacimiento, promedio, sexo,  
    idEscuela, idSalon, idCurso)  
VALUES ("Paco", "Arce",  
    "1989-02-10", 8, "M", 1, 1, 1);
```

Insertar datos a una tabla

- ❑ Se pueden omitir los nombres de los campos, pero MySQL los insertará en orden en que aparecen los campos en la tabla. Hay que tener cuidado al respecto:
- ❑ `INSERT INTO alumnos VALUES (0, "Paco", "Arce", "1966-02-10", 8, "M", 1, 1, 1);`

MySQL

Insertar datos por medio de un archivo SQL

Insertar datos por medio de un archivo SQL

- ❑ Otra forma de insertar datos es por medio de un archivo de tipo texto que contiene las instrucciones SQL.
- ❑ Por ejemplo:
- ❑ En un procesador de palabra que no maneje caracteres especiales teclee las siguientes instrucciones:

Insertar datos por medio de un archivo SQL

```
INSERT INTO escuelas VALUES (0,"Escuela 1");
```

```
INSERT INTO escuelas VALUES (0,"Libre");
```

```
INSERT INTO escuelas VALUES (0,"Escuela 2");
```

```
INSERT INTO escuelas VALUES (0,"Abierta de  
economía");
```

Insertar datos por medio de un archivo SQL

Desde el sistema operativo tecle:

```
mysql -u usuario -p password ESCUELA <  
insertEscuelas.sql
```

Obviamente deberá de cambiar el usuario y el password al que le corresponde dentro del sistema.

Ahora en la tabla escuelas habrá insertado 4 filas.

MySQL

Insertar datos desde un archivo externo

Insertar datos desde un archivo externo

- ❑ Podemos insertar una gran cantidad de información desde un archivo externo por medio de la instrucción LOAD DATA. Su sintaxis general es:

```
LOAD DATA [LOW_PRIORITY] [LOCAL] INFILE "nombre_archivo.txt"
```

```
[IGNORE | REPLACE]
```

```
INTO TABLE nombre_tabla
```

```
[FIELDS
```

```
  [TERMINATED BY 'cadena']
```

```
  [OPTIONALLY] ENCLOSED BY 'character']
```

```
  [ESCAPED BY 'character']]
```

```
[LINES TERMINATED BY 'cadena']
```

```
[IGNORE n LINES]
```

```
[(lista_columnas)]
```

Insertar datos desde un archivo externo

- ❑ `LOW_PRIORITY` retrasa la sentencia hasta que ningún cliente esté leyendo de la tabla.
- ❑ `LOCAL` hará que el archivo se lea de la computadora del cliente y se envíe al servidor.
- ❑ `IGNORE` No carga las columnas que tengan la llave principal duplicada
- ❑ `REPLACE` Remplaza los registros con llaves duplicadas

Insertar datos desde un archivo externo

- ❑ `SÍ` se usa `FIELDS` haya que especificar `TERMINATED BY` (describe el carácter o caracteres que delimitan valores dentro de una fila)
- ❑ `ENCLOSED BY` especifica un carácter entrecomillado que se quita al final del campo de valores `SÍ` se usa `OPTIONALLY`.

Insertar datos desde un archivo externo

- ❑ `ESCAPED BY` especifica el carácter de escape de los caracteres especiales.
- ❑ `FIELDS` siempre debe preceder a `LINES`.
- ❑ `LINES TERMINATED BY` especifica un carácter o caracteres que significan el final de las líneas.

Insertar datos desde un archivo externo

- ❑ Si no se especifica ni FIELDS ni LINES, los valores por default serán:
- ❑ FIELDS
- ❑ TERMINATED BY '\t'
- ❑ ENCLOSED BY ''
- ❑ ESCAPED BY '\\'
- ❑ LINES TERMINATED BY '\n'

Insertar datos desde un archivo externo

- ❑ La sentencia `IGNORE n LINES` indica que se descartan las primeras `n` líneas.
- ❑ Supongamos que tenemos un archivo con campos separados por tabuladores y cuyas líneas terminan con una Línea nueva y un retorno de carro.

Insertar datos desde un archivo externo

1 ActionScript

2 PHP

3 MySQL

4 Photoshop

5 Dreamweaver

6 Flash

Insertar datos desde un archivo externo

- ❑ La sentencia deberá ser escrita como sigue:
- ❑ `LOAD DATA LOCAL INFILE "archivo.txt" INTO TABLE cursos LINES TERMINATED BY "\r\n";`
- ❑ MySQL le indicará cuantas líneas han sido añadidas.

MySQL

Actualizar datos

Actualizar datos

- ❑ En muchas ocasiones es necesario modificar los datos que ya se encuentran en nuestras tablas. Para ello contamos con la siguiente sentencia:
- ❑ `UPDATE [LOW_PRIORITY] tabla`
- ❑ `SET columna=valor [,columna2=valor2...]`
- ❑ `[WHERE condición]`
- ❑ `[LIMIT n]`

Actualizar datos

- Donde SET contiene una lista separada por comas de las columnas que deben actualizarse y el nuevo valor de cada columna, con el formato `columna=valor`.

Actualizar datos

- El valor suministrado por las expresiones incluye elementos tales como constantes, valores seleccionados de una columna de otra tabla o vista, o valores calculados por una expresión compleja.

Actualizar datos

- ❑ **WHERE:** Especifica la condición de búsqueda que define las filas de las tablas de origen que están calificadas para proporcionar valores para las expresiones de la cláusula SET.
- ❑ **LOW_PRIORITY** indica que no se ejecuta la actualización hasta que ningún cliente esté leyendo la tabla.
- ❑ **LIMIT n** establece en n el número máximo de filas que se pueden actualizar.

Actualizar datos

- ❑ Si quisiéramos modificar los datos añadidos en el punto anterior, la sentencia sería:
- ❑ `UPDATE contactos SET email="paco@mail.com" WHERE nombre="Paco";`

Actualizar datos

- ❑ La cláusula `WHERE` en la sentencia `UPDATE` es opcional, y si se omite, entonces se actualizan **TODAS** las filas de la tabla destino.
- ❑ Mucho cuidado: Debido a que puede cambiar en forma definitiva la información de sus tablas, siempre utilice el `UPDATE` con la sentencia `WHERE` (a menos que desee cambiar todas las columnas por un mismo valor).

MySQL

Borrar registros

Borrar registros

- ❑ Para borrar un registro, o grupos de registros, se utilizará la sentencia

DELETE:

- ❑ DELETE [LOW_PRIORITY] FROM
tabla

- ❑ [WHERE condición]

- ❑ [LIMIT n]

Borrar registros

- ❑ En esta instrucción también es necesario utilizar la sentencia `WHERE`, a menos que desee borrar todos los datos de la tabla, por ejemplo:
- ❑ `DELETE FROM contactos;`

Borrar registros

- ❑ Para borrar solo un registro, puede utilizar la siguiente sentencia:
- ❑ `DELETE FROM contactos WHERE nombre = "Paco";`