

SQL: Structured Query Language

Lenguaje de consultas estructuradas

Elizabeth León Guzmán, Ph.D.

eleonguz@unal.edu.co

Jonatan Gómez Perdomo, Ph. D.

jgomezpe@unal.edu.co

Arles Rodríguez, Ph.D.

aerodriguezp@unal.edu.co

Camilo Cubides, Ph.D. (c)

eccubidesg@unal.edu.co

Carlos Andres Sierra, M.Sc.

casiterrav@unal.edu.co

Research Group on Data Mining – Grupo de Investigación en Minería de Datos – (Midas)

Research Group on Artificial Life – Grupo de Investigación en Vida Artificial – (Alife)

Computer and System Department

Engineering School

Universidad Nacional de Colombia

Agenda

- 1 Introducción
- 2 Creación Base de Datos
- 3 Creación de tablas - CREATE TABLE
- 4 Borrado de tablas - DROP TABLE
- 5 Inserción de Datos - INSERT
- 6 Actualización de datos - UPDATE
- 7 Borrado de datos - DELETE
- 8 Alteración/Modificación de tablas - ALTER TABLE
- 9 Problemas



Introducción

- SQL lenguaje usado para definir, manipular, y controlar bases de datos relacionales
- Definido por ANSI (American National Standards Institute)
- Comandos SQL se pueden dividir en tres categorías:
 - DDL (Data definition language)
`create, alter, drop`
 - DML (Data manipulation language)
`select, insert, update, delete`
 - DCL (Data control language)
`grant, revoke`



Operaciones básicas sobre la Base de Datos



Crear, Leer, Actualizar, Borrar
Create, Read, Update, Delete
(CRUD)

- Creación/Borrado de tablas y BD
- Alteración de las tablas
- Inserción de datos
- Borrado de datos
- Consulta de datos
- Actualización de datos



Agenda

- 1 Introducción
- 2 Creación Base de Datos
- 3 Creación de tablas - CREATE TABLE
- 4 Borrado de tablas - DROP TABLE
- 5 Inserción de Datos - INSERT
- 6 Actualización de datos - UPDATE
- 7 Borrado de datos - DELETE
- 8 Alteración/Modificación de tablas - ALTER TABLE
- 9 Problemas



Creación de una Base de Datos

Para crear una base de Datos en MySQL, se usa la instrucción **CREATE SCHEMA**.

- Crea una base de datos (schema) llamada libreria (se recomienda no usar tildes en los nombres de los objetos de las bases de datos).

```
CREATE SCHEMA libreria;
```

Luego de creada la Base de Datos es necesario seleccionarla para operar sobre ella. Se usa la instrucción **USE**:

- Selecciona la base de datos libreria

```
USE libreria;
```



Tablas

Objeto de Bases de Datos

Una Tabla es un objeto de base de datos que almacena datos en filas y columnas:

Tabla → relación del modelo relacional

- 1 Nombre de la tabla y de cada columna (modelo físico de la BD)
- 2 Determinar las propiedades de cada columna. Una de las propiedades es el tipo de dato (modelo físico de la BD)

Integer de tamaño 13	Char de tamaño 50	Integer tamaño 4
LIBRO		
libId	libNombre	libPub
1001	Cien años de soledad	1967
1002	La vorágine	1924
1003	María	1867
1004	Cóndores no se entierran todos los días	1971



Agenda

- 1 Introducción
- 2 Creación Base de Datos
- 3 Creación de tablas - CREATE TABLE**
- 4 Borrado de tablas - DROP TABLE
- 5 Inserción de Datos - INSERT
- 6 Actualización de datos - UPDATE
- 7 Borrado de datos - DELETE
- 8 Alteración/Modificación de tablas - ALTER TABLE
- 9 Problemas



Creación de Tablas I

Propiedades de los atributos

Al crear una tabla es necesario especificar las propiedades de los atributos.
Sintaxis básica de crear la tabla:

```
CREATE TABLE nombre_tabla (  
    nombre_columna lista de propiedades ,  
    ...  
    nombre_columna lista de propiedades ,  
    [referencias]  
);
```

Algunas propiedades básicas de un atributo son:

- Tipo de dato
- NULL | NOT NULL
- IDENTITY o AUTOINCREMENTAL
- DEFAULT
- ENUM (Lista de valores del dominio)
- PRIMARY KEY
- FOREIGN KEY



Creación de Tablas II

Tipos de Datos en MySQL

Numéricos

- INT (Integer)
- SMALLINT
- TINYINT
- MEDIUMINT
- BIGINT
- FLOAT(n) donde n es la precisión
- DOUBLE(i,j) o REAL(i,j) donde i es el número total de dígitos y j es el número de dígitos decimales
- DECIMAL (i,j) o NUMERIC (i,j) donde i es la precisión y j la cantidad de dígitos decimales

Cadenas de caracteres

- CHAR(n) donde n es la longitud de la cadena de caracteres
- VARCHAR(n) es una cadena de tamaño variable cuya longitud máxima es n.
- VARBINARY(n) es una cadena que almacena cadenas de binarias de bits

Fecha y Hora

- DATE → YYYY-MM-DD
- TIME → HH:MM:SS
- TIMESTAMP → Incluye fecha y hora más un mínimo de 6 fracciones decimales para los segundos y con TIME ZONE opcional (depende del gestor de BD)

Más tipos de datos y sus descripciones en:

<https://dev.mysql.com/doc/refman/8.0/en/data-types.html>



Creación de Tablas III

Propiedad NULL | NOT NULL

- Un NULL representa un valor desconocido o que no aplica
 - Para valores numéricos, NULL no es igual a 0
 - Para caracteres, NULL no es igual a " " (en blanco)
- NULL no se considera menor que, mayor que, o igual a cualquier otro valor
- Dos NULL no se consideran iguales

Ejemplo

Para una tabla persona, con atributos nombre y email.

- El atributo "email" puede ser NULL, ya que no todas las personas tienen correo electrónico.
- El nombre es NOT NULL, todas las personas tienen nombre



Creación de Tablas IV

Propiedad IDENTITY o AUTOINCREMENTAL

- El servidor automáticamente asigna un número secuencial (único) a la columna que tiene la propiedad de IDENTITY o AUTOINCREMENTAL en el momento de insertar datos en la tabla.
- Propiedades:
 - Sólo puede existir una por tabla
 - Debe de ser de tipo numérico sin posiciones decimales
 - No se puede actualizar
 - No acepta valores tipo NULL
 - Inicia por default en 1.

Ejemplo

Los códigos e identificadores: el Id de una persona, de un empleado, de un estudiante, etc.



1		
2		
3		



Creación de Tablas V

Propiedad DEFAULT

Los atributos pueden tener valores **por defecto**. Es decir, en el momento de la inserción si no hay valor para ese atributo se insertará el valor definido por defecto.

Ejemplo

Para una tabla persona, con atributos nombre, email y país de nacimiento.

- El atributo "pais" puede tener un valor por defecto, en caso de que la mayoría de personas sean de Colombia el valor por defecto puede ser "Colombia"
- El atributo nombre no puede tener un valor por defecto.



Creación de Tablas VI

Propiedad ENUM

Los atributos pueden tener una lista de valores permitidos, usando ENUM. En el momento de la inserción solo podrá tener un valor de esa lista.

Ejemplos

- El "tipo de identificación", valores: CC, TI, NUIP, Pasaporte
- La "forma de pago" de una venta, valores: Efectivo, Tarjeta Crédito, Tarjeta Débito, Transferencia, PSE

Ejemplo

```
CREATE TABLE venta(  
  vta_fatura      integer      AUTO_INCREMENT ,  
  vta_id_cliente  integer      NOT NULL ,  
  vta_fecha       date         NOT NULL ,  
  vta_total       money        NULL ,  
  vta_pago        ENUM('Efectivo ', 'Tarjeta Crédito ', 'Tarjeta Débito ', 'Transferencia '  
                   , 'PSE'),  
  vta_trans       varchar(20)  DEFAULT "Electrónica"  
);
```

Creación de Tablas VII

Llave Primaria - *Primary Key*

Atributo que no repite valor (único), que se utiliza para identificar cada registro como único. Se utiliza la instrucción PRIMARY KEY

Ejemplo

Los códigos, los números de identificación, los números de las facturas.

```
CREATE TABLE libro (  
    libId          integer      NOT NULL PRIMARY KEY,  
    libNombre     char(30)     NOT NULL,  
    libPub        smallint     NULL  
);
```



Creación de Tablas VIII

Llave Primaria - *Primary Key*

Pueden haber llaves primarias compuestas, es decir que se componen de mas de un atributo. En este caso se especifica al final de la definición de los atributos de la tabla.

Ejemplo

— Tabla con llave primaria compuesta de tres atributos

```
CREATE TABLE venta(  
  vta_id_cliente      integer NOT NULL,  
  vta_id_producto     integer NOT NULL,  
  vta_cantidad        integer NULL,  
  vta_fecha           date     NOT NULL,  
  vta_precio          money    NULL,  
  vta_total           money    NULL,  
  PRIMARY KEY (vta_id_cliente, vta_id_producto, vta_fecha)  
);
```



Creación de Tablas VIII

Llave Foraneas - *Foreign Key*

Para definir un atributo como referencia a otro atributo en otra tabla se usa la expresion FOREIGN KEY al final de la definición de todos los atributos de la tabla.

```
FOREIGN KEY (atributo) REFERENCES  
    tabla_referencia(atributo_tabla_referencia)
```

Ejemplo

— vta_id_cliente es un atributo que es llave foránea. Referencia al atributo id_cliente de la tabla cliente

```
CREATE TABLE venta (  
    vta_id_factura INT NOT NULL,  
    vta_id_cliente INT NOT NULL,  
    vta_cantidad INT,  
    PRIMARY KEY(vta_id_factura),  
    FOREIGN KEY (vta_id_cliente) REFERENCES cliente(id_cliente)  
);
```



Creación de Tablas IX

Sintaxis en SQL

Sintaxis SQL simplificada para Crear una Tabla o relación:

```
CREATE TABLE nombre_tabla (  
    nombre_columna tipo_dato [AUTO_INCREMENT] [[PRIMARY] KEY] [NULL | NOT NULL] [  
        DEFAULT {valor}] ],  
    ...  
    nombre_columna tipo_dato [AUTO_INCREMENT] [[PRIMARY] KEY] [NULL | NOT NULL] [  
        DEFAULT {valor}] ],  
    FOREIGN KEY nombre_atributo REFERENCES nombre_tabla_ref(nombre_atributo_ref)  
);
```

Ejemplo

— Crea la tabla persona con cuatro atributos

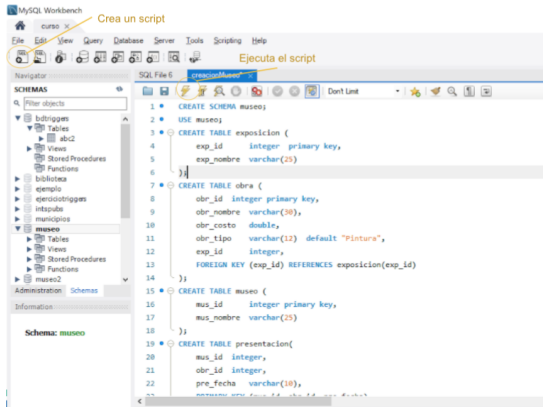
```
CREATE TABLE persona (  
    perID            integer        AUTO_INCREMENT PRIMARY KEY ,  
    perNombre        char(30)      NOT NULL ,  
    perCorreo        char(30)      NULL ,  
    perPais           ENUM('Argentina', 'Alemania', 'Colombia', 'Italia', 'México', 'España', 'EUA')  
        DEFAULT "Colombia"  
);
```



Creación de Tablas X

Recomendaciones para el desarrollo

- Crear todas las tablas a través de un **script**
 - Facilita la recreación de los objetos
 - Sirve como material permanente de referencia



Creación de Tablas XI

En MySQL

Ejemplo

```
CREATE SCHEMA libreria;
USE libreria;
CREATE TABLE libro (
    libId          integer      NOT NULL PRIMARY KEY,
    libNombre      char(30)     NOT NULL,
    libPub         smallint     NULL
);

CREATE TABLE autor (
    autId          integer      AUTO_INCREMENT PRIMARY KEY,
    autApellido    varchar(30)  NOT NULL DEFAULT "",
    autNombre      varchar(30)  NOT NULL,
    autEmail       char(15)     NULL,
    autPais        ENUM('Argentina', 'Alemania', 'Colombia', 'Italia', 'México', 'España', 'EUA')
                    DEFAULT "Colombia"
);

CREATE TABLE venta (
    vtaId          integer      AUTO_INCREMENT PRIMARY KEY,
    vtaFecha       date NOT NULL,
    libId          integer NOT NULL,
    vtaCantidad    smallint NOT NULL,
    FOREIGN KEY (libId) REFERENCES libro(libId),
);
```



Creación de Tablas XII

Verificación de la creación de tablas

- Para verificar la creación de tablas se usa:

```
SHOW TABLES;
```

- Para revisar la estructura de una tabla:

```
SHOW COLUMNS FROM nombre_tabla;
```

- Para ver la descripción de la tabla se usa:

```
DESCRIBE nombre_tabla;
```

Ejemplo

Ejecutar en la bd libreria:

```
SHOW TABLES;
```

```
DESCRIBE LIBRO;
```

```
SHOW COLUMNS FROM LIBRO;
```



Agenda

- 1 Introducción
- 2 Creación Base de Datos
- 3 Creación de tablas - CREATE TABLE
- 4 Borrado de tablas - DROP TABLE**
- 5 Inserción de Datos - INSERT
- 6 Actualización de datos - UPDATE
- 7 Borrado de datos - DELETE
- 8 Alteración/Modificación de tablas - ALTER TABLE
- 9 Problemas



Borrado de Tablas

Sintaxis Simplificada

Comando DROP. Borra el objeto tabla con sus datos. Sintaxis Simplificada para drop:

```
DROP TABLE table_name
```

Ejemplo

— borra la tabla autor

```
DROP TABLE autor
```



Agenda

- 1 Introducción
- 2 Creación Base de Datos
- 3 Creación de tablas - CREATE TABLE
- 4 Borrado de tablas - DROP TABLE
- 5 Inserción de Datos - INSERT**
- 6 Actualización de datos - UPDATE
- 7 Borrado de datos - DELETE
- 8 Alteración/Modificación de tablas - ALTER TABLE
- 9 Problemas



Inserción de Datos I

Sintaxis Simplificada

Sintaxis INSERT Simplificada:

```
INSERT [INTO] nombre_tabla [(lista_columnas)]  
{ VALUES (lista_valores) | sentencia_select }
```

Ejemplo

— Inserta un registro en la tabla libro usando lista
— de columnas. Asume que la llave primary es autoincremental

```
INSERT INTO libro (libId, libNombre, libPub) VALUES ("El olvido que seremos", 2005);
```

— Inserta un registro en la tabla libro. Se inserta la llave

```
INSERT INTO libro VALUES (201, "El renacuajo paseador", 1867);
```



Inserción de Datos II

En MySQL

Ejemplo

— Insertar los siguientes tres registros en la tabla Libro:

```
INSERT INTO libro (libId, libNombre, libPub) VALUES (1001, "Cien años de soledad", 1967);
```

```
INSERT INTO libro (libId, libNombre, libPub) VALUES (1002, "La Vorágine", 1924);
```

```
INSERT INTO libro (libId, libNombre, libPub) VALUES (1003, "María", 1867);
```

— Insertar los libros:

— "Cóndores no se entierran todos los días" publicado en 1971 con ID 1004,

— "El renacuajo paseador" publicado en 1867 con ID 1005);

— Para verificar las actualizaciones, ejecutar antes y después del INSERT

```
SELECT * FROM libro;
```

```
SELECT * FROM autor;
```



Inserción de Datos III

En MySQL

Ejemplo (continuación)

— Insertar los dos registros en la tabla Autor, solo se insertan nombre y apellido.

```
INSERT INTO autor (autApellido, autNombre) VALUES ("García Márquez", "Gabriel");
```

— Insertar un autor con autNombre

```
INSERT INTO autor (autApellido, autNombre) VALUES ("Isaacs", "Jorge");
```

— Insertar un autor con autPais

```
INSERT INTO autor (autApellido, autNombre, autPais) VALUES ("Alighieri", "Dante", "Italia");
```

— Insertar un autor con correo

```
INSERT INTO autor (autApellido, autNombre, autEmail) VALUES ("Trueba", "Fernando", "ftrueba@gmail.com");
```

— Insertar un autor con autPais

```
INSERT INTO autor (autApellido, autNombre, autPais) VALUES ("Rivera", "Jose Eustasio");
```



MySQL



Agenda

- 1 Introducción
- 2 Creación Base de Datos
- 3 Creación de tablas - CREATE TABLE
- 4 Borrado de tablas - DROP TABLE
- 5 Inserción de Datos - INSERT
- 6 Actualización de datos - UPDATE**
- 7 Borrado de datos - DELETE
- 8 Alteración/Modificación de tablas - ALTER TABLE
- 9 Problemas



Actualización de Datos

Sintaxis UPDATE simplificada:

```
UPDATE nombre_tabla  
SET columna1=valor1, columna2=valor2, ...  
WHERE condicion
```

Ejemplo

— Actualiza el Id del libro con Id = 1003

```
UPDATE libro SET libPub=2000 WHERE libId = 1003;
```

— Actualiza el Nombre del libro con Id = 1003

```
UPDATE libro SET libNombre = "La Casa Grande", libPub=1962 WHERE libId = 1003;
```

— Para verificar las actualizaciones, ejecutar antes y después del UPDATE

```
SELECT * FROM libro;
```



Agenda

- 1 Introducción
- 2 Creación Base de Datos
- 3 Creación de tablas - CREATE TABLE
- 4 Borrado de tablas - DROP TABLE
- 5 Inserción de Datos - INSERT
- 6 Actualización de datos - UPDATE
- 7 Borrado de datos - DELETE**
- 8 Alteración/Modificación de tablas - ALTER TABLE
- 9 Problemas



Borrado de Datos

Sintaxis DELETE simplificada:

```
DELETE FROM nombre_tabla WHERE condicion
```

La condición es opcional. Si no se usa borra todos los datos de la tabla.

Ejemplos

Borra los libros con año de publicación menor a 1990

```
DELETE FROM libro where libPub < 1990;
```

Borra todos los datos de la tabla libro.

```
DELETE FROM libro;
```



Actividad. Borrar tres registros de la tabla autor con alguna condición



Agenda

- 1 Introducción
- 2 Creación Base de Datos
- 3 Creación de tablas - CREATE TABLE
- 4 Borrado de tablas - DROP TABLE
- 5 Inserción de Datos - INSERT
- 6 Actualización de datos - UPDATE
- 7 Borrado de datos - DELETE
- 8 Alteración/Modificación de tablas - ALTER TABLE**
- 9 Problemas



Alterar Tablas I

Sintaxis ALTER simplificada:

```
ALTER nombre_tabla  
  ADD nombre_columna definicion_columna ,  
  DROP nombre_columna ,  
  RENAME COLUMN nombre_actual TO nombre_nuevo
```

Ejemplo

— Adiciona un atributo de precio a libro con valor por defecto en 0

```
ALTER libro ADD libPrecio float(7,1) DEFAULT 0;
```

— Borra una columna

```
ALTER autor DROP fechaNto;
```

— Renombra un atributo

```
ALTER autor RENAME COLUMN autEmail TO autCorreo;
```

— Para verificar los cambio, ejecutar antes y después de la instrucción

```
SELECT * FROM libro;
```

```
SELECT * FROM autor;
```



Alterar Tablas II

Ejemplo

Para el ejemplo de la librería, suponer que un libro es escrito por un solo autor. Por lo que se desea incluir el atributo del identificador del autor en la tabla libro como llave foránea a la tabla autor.

```
ALTER libro ADD autID integer FOREIGN KEY (autID) REFERENCES autor (autID);
```

Insertar a los libros el correspondiente identificador del autor:

```
UPDATE libro SET autID=1 WHERE libId=1001
```

¿Qué sucede con la siguiente instrucción?, si no funciona ¿Cuál es la razón?

```
UPDATE libro SET autID=10 WHERE libId=1005
```



Agenda

- 1 Introducción
- 2 Creación Base de Datos
- 3 Creación de tablas - CREATE TABLE
- 4 Borrado de tablas - DROP TABLE
- 5 Inserción de Datos - INSERT
- 6 Actualización de datos - UPDATE
- 7 Borrado de datos - DELETE
- 8 Alteración/Modificación de tablas - ALTER TABLE
- 9 Problemas



Problemas I

Crear BD y cargar script

Problema

- ①
 - Bajar el script [museo.sql](#)
 - Usando la aplicación Workbench, en File del menú principal y la opción open script cargar el script museo.sql
 - Ejecutar el script haciendo click en el botón execute (rayo)
 - Verificar si los datos fueron cargados con el comando SHOW (todas las tablas)

```
SHOW columns FROM obra;  
SHOW columns FROM exposicion;
```

- Verificar si tienen datos

```
SELECT * FROM obra;  
SELECT * FROM exposicion;
```

- Explique las tablas y que relación existen entre ellas.



Problemas II

cargar script - corregir errores

Problema

- 2
 - Bajar el script [veterinariaN.sql](#)
 - Usando la aplicación Workbench, en File del menú principal y la opción open script cargar el script veterinariaN.sql
 - Ejecutar el script
 - Corregir el script con cada uno de los errores que se van generando y ejecutar nuevamente el script



Problemas III

SQL

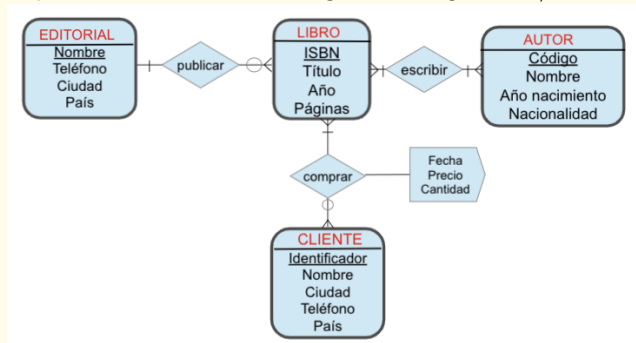
Problema

- 3
- Escribir la sentencia SQL para crear una tabla que guarde la siguiente información: Se desea almacenar información de Películas, como el título que debe ser obligatorio, tema que puede ser opcional, género que debe ser obligatorio, pero si no se conoce el género por defecto se le colocará la cadena "*desconocido*". También se debe incluir el año del estreno que debe ser obligatorio, el id de la película será el identificador único y es un número consecutivo. Deberá tener una llave foránea a una tabla director con el ID del director de la película.
 - Insertar datos a la tabla creada
 - Alterar la tabla para incluir un atributo numérico que permita guardar la duración de la película en minutos. Por defecto colocar la duración en 0.
 - Alterar la tabla para incluir un atributo numérico que permita guardar una calificación de la película.
 - Actualizar la calificación de las películas que tienen año de estreno anterior al 2000, colocar una calificación de 0.0

Problemas IV

Problema

- 4 Implementar las tablas del siguiente diagrama E/R.



- Insertar datos a las tablas
Generar un script (file.sql) con las instrucciones de creación e inserción de datos.

