



UNIVERSIDAD NACIONAL DE COLOMBIA
FACULTAD DE INGENIERÍA

BASES DE DATOS

Taller 2 (Primera Parte: creación de tablas y consultas)

OBJETIVOS

1. Aprender a usar un sistema administrador de bases de datos utilizando lenguaje SQL
2. Consultas (Proyección, selección y Join)

Manipulación básica del Motor MYSQL

1. Comandos Básicos

* Comandos de interprete para consola

`\q` salir
`\h` ayuda

Dentro de MySql pueden existir varias bases de datos. Cada base de datos es un espacio independiente con sus propios objetos de datos. Para ver la lista de bases de datos disponible se usa:

`SHOW DATABASES;`

* Seleccionar Bases de Datos o Esquema

Para seleccionar una base de datos se usa:

`USE nombre_base_de_datos;`

`USE DB1` – Secciona la base de datos DB1

* Mostrar tablas en la base de datos

Para ver la lista de tablas que contiene la base de datos ya seleccionada se usa:

`SHOW TABLES;`

* Mostrar descripción de una tabla

Para ver la descripción de una tabla específica de la base de datos seleccionada se usa:

`DESCRIBE nombre_tabla;`

En el ejemplo:

```
DESCRIBE restaurante;
```

Seleccionar el contenido de una tabla

Para seleccionar toda la información de la tabla:

```
SELECT * FROM restaurante;
```

Para ver la información de un campo específico (Proyección) se usa:

```
SELECT nombre_campo FROM nombre_tabla;
```

Seleccionar otros campos de la tabla.

Cambiar password

Para cambiar la contraseña del usuario se usa:

```
SET password = password ('nuevo password');
```

Para verificar el password: salir de Mysql (EXIT) y volver a entrar.

2. Creación y Borrado de tablas en MYSQL

El comando para crear tablas es **CREATE TABLE** (se recomienda escribirlo siempre en mayúscula).

Ejemplo:

```
CREATE TABLE agendaNumero (  
    nombre TEXT,  
    apellido TEXT,  
    telefono INTEGER,  
    nacimiento INTEGER,  
    estatura FLOAT  
);
```

En este ejemplo TEXT e INTEGER son dos tipos de datos que existen en MySQL.

El comando usado para borrar tablas es **DROP TABLE**. Ejemplo:

```
DROP TABLE agenda
```

Actividad: Crear una tabla llamada agenda_numero donde numero es el numero del grupo al cual pertenecen. Para ver si la tabla fue creada exitosamente usar SHOW TABLES, y para ver todos los datos de la tabla SHOW COLUMNS FROM agendaNumero.

3. Inserción de datos

El comando usado es **INSERT**. Ejemplos:

```
INSERT INTO agenda (nombre, apellido, telefono, nacimiento, estatura) VALUES ('Juan','Rios', 2309876, 1990, 1.70);  
INSERT INTO agenda (nombre, apellido, telefono, nacimiento, estatura) VALUES ('Ana','Suarez', 2456876, 1983, 1.60);
```

Actividad: insertar 10 filas, al menos dos con el mismo nombre pero distinto apellido, y al menos dos con la misma fecha de nacimiento.

4. Borrado de datos

El comando para borrar datos en la tabla es **DELETE TABLE** (se recomienda escribirlo siempre en mayúscula). Para borrar todos los datos de la tabla se usa:

```
DELETE TABLE nombre tabla
```

Para borrar solo algunos registros de la tabla se usa con la cláusula **WHERE**. Ejemplo:

```
DELETE TABLE agenda WHERE nacimiento < 1980
```

Actividad: borrar 3 filas de la tabla agenda con alguna condición.

5. Consultas

Los siguientes comandos ejecutan consultas a la base de datos:

```
SELECT * FROM agenda;  
SELECT nombre, apellido FROM agenda;  
SELECT telefono FROM agenda WHERE apellido='Gonzalez';  
SELECT nombre FROM agenda WHERE nacimiento = 1983;
```

```
SELECT DISTINCT nombre FROM agenda;
```

```
SELECT nombre FROM agenda WHERE apellido='Rios' OR apellido='Gonzalez'  
SELECT nombre FROM agenda WHERE apellido='Rios' AND nacimiento = 1983;
```

```
SELECT apellido FROM agenda WHERE LIKE 'P%';
```

Actividad: Escribir 5 consultas sobre la tabla de agenda creada por el grupo.

Actividad: Crear una tabla con mínimo 5 columnas (tema de libre elección). Insertar 10 filas y luego hacer 5 consultas diferentes.

6. Consultas Complejas

6.1 Usar Base de Datos del Grupo

- a. Crear y seleccionar la base de datos (esquema) de su grupo (usando USE o dando doble click sobre el esquema). GR1_2 indica que el grupo del curso es el 1 y el grupo de trabajo es el 2. Si usted es del grupo2 y su grupo de trabajo es el 5 entonces su base de datos es GR2_5
- b. Cargar el archivo municipios.sql
 - i. Primero bajar el archivo municipios.sql de la página del curso en la carpeta talleres y
 - ii. Usando la aplicación de Query Browser o Workbench, en File del menú principal y la opción open script se carga el script municipios.sql
 - iii. Ejecutar el script haciendo click en el boton execute (verde enquery browser o el rayo en workbench)
- c. Verificar si los datos fueron cargados
 - i. SHOW columns FROM comunidad;
 - ii. SHOW columns FROM municipio;
- d. Verificar si tienen datos
 - i. SELECT * FROM comunidad;
 - ii. SELECT * FROM municipio;

Si las tablas no existen o no contienen datos, repetir el proceso cuidadosamente. Las tablas deben estar creadas y cargadas para los siguientes puntos.

- e. Explique las tablas y que relación existen entre ellas.

6.2 Búsqueda de Strings

El operador más usado en MySql para buscar strings es **LIKE**. Con este comando se pueden realizar las siguientes consultas:

- ◆ Consulta Exacta
LIKE 'Catalunya'
- ◆ Consulta omitiendo una parte
LIKE 'Catal%'
- ◆ Consulta omitiendo un caracter
LIKE 'Ca_t__unya'

Se pueden combinar usando AND, OR y NOT

- e. Cuales son los resultados de:
SELECT nom FROM comunidad WHERE nom LIKE '%Castilla%' AND nom NOT LIKE '%Mancha%';
Explicar.

Funciones que se pueden aplicar sobre los strings:

- ◆ LENGHT('texto') -> 5
- ◆ CONCAT('text', 'to') -> texto
- ◆ LOWER('TextO') -> texto
- ◆ UPPER('TextO') -> TEXTO
- ◆ RTRIM('TextO ') -> TextO //quitar espacios

Todas las funciones pueden ser combinadas, por ejemplo:

```
SELECT nom FROM municipio WHERE LOWER (nom) LIKE '%a%' AND LENGTH(nom) <=3
```

f. Escribir una consulta para:

- i. Municipios que tengan solamente una X en su nombre (una y sólo una) y que su nombre sea mas largo que 20 caracteres
- ii. Municipios que tengan sólo un tipo de vocal en su nombre. Esto incluye: todos los que se escriben solo con A, todos los que se escriben solo con E, ... en una sólo consulta. Ejemplo: "Tolox" y "Jun" están bien, pero "portbou" no. Use paréntesis para agrupar condiciones, y escriba todo en una sola consulta

6.3 Búsqueda con condiciones numéricas

Los operadores numéricos que se usan son: <, >, !=, <= y >=

Ejemplo: `SELECT NOM from municipio WHERE poblacio2003 > 1000000`

Existe la posibilidad de usar operadores de agregación, que usan los valores de una columna completa. Los típicos son:

- ◆ AVG: promedio,
- ◆ MIN y MAX : minimo y máximo
- ◆ SUM: suma
- ◆ Count: contar

Ejemplo usando operadores de agregación:

```
SELECT MAX(poblacio2003) FROM municipio;  
SELECT poblacio2001-poblacio1991 FROM municipio;  
SELECT AVG(superficie) FROM municipio;  
SELECT nom FROM municipio WHERE poblacio2003=(SELECT MAX(poblacio2003) FROM municipio)
```

Las columnas numéricas se pueden usar para ordenar usando ORDER BY. Adicionalmente se puede usar LIMIT para restringir el máximo de elementos a retornar

Ejemplo: `SELECT nom FROM municipio ORDER BY poblacio2003 DESC LIMIT 10`

- a. Ejecutar la sentencia del ejemplo y explicar.

Escribir una consulta para:

- a. Nombre de los 5 municipios más densamente poblados en el 2003. $Densidad = población / superficie$
- b. Nombre de los municipios que tienen una superficie menor a 1/10 del promedio
- c. Nombre y población en el 1991 y en el 2001 de los municipios que tienen una población 10 veces mayor en el 2001 que en 1991, pero que tenían habitantes en 1991 (hay sólo uno)

6.4 Búsqueda en más de una tabla (JOIN)

Para buscar en mas de una tabla, lo que se necesita es unir ambas tablas por una columna que tenga el mismo significado. En este caso, la tabla comunidad tiene una columna ca_id, y la tabla municipio también:

Ejemplo:

- ◆ `SELECT municipio.nom, comunidad.nom FROM municipio, comunidad WHERE municipio.ca=id = comunidad.ca_id AND comunidad.nom = 'Catalunya' ORDER BY municipio.nom;`
- ◆ `SELECT comunidad.nom, SUM(municipio.poblacio2003) FROM municipio, comunidad WHERE municipio.ca_id= comunidad.ca_id GROUP BY comunidad.nom ORDER BY comunidad.nom`

- a. Ejecutar las sentencias del ejemplo
- b. Hacer las mismas consultas pero usando el operador JOIN (visto en clase)

Escribir una sentencia para consultar:

- c. la superficie total de cada comunidad
- d. Variación de la población entre 1991 y 2001 en cada comunidad
- e. Superficie total de la comunidad de Galicia