## **Investigacion**

• JONATAN PALOMARES CASTAÑEDA

• JUAN DIEGO PATIÑO MUÑOZ

### **A. SQL**

#### **Que es SQL:**

SQL es un acrónimo en inglés para Structured Query Language. Un Lenguaje de Consulta Estructurado. Un tipo de lenguaje de programación que te permite manipular y descargar datos de una base de datos. Tiene capacidad de hacer cálculos avanzados y álgebra. Es utilizado en la mayoría de empresas que almacenan datos en una base de datos. Ha sido y sigue siendo el lenguaje de programación más usado para bases de datos relacionales.

https://datademia.es/blog/que-es-sql

#### **Para que sirve SQL**

Con SQL, puedes realizar diversas operaciones en una base de datos:

Consultas: Puedes escribir consultas para obtener información específica de la base de datos, seleccionando y filtrando datos según ciertos criterios.

Inserción: Permite insertar nuevos registros o filas de datos en una tabla existente de la base de datos.

Actualización: Permite modificar los valores de uno o más registros existentes en una tabla.

Eliminación: Permite eliminar registros específicos de una tabla.

Creación y modificación de esquemas: SQL también se utiliza para crear y modificar la estructura de una base de datos, como crear tablas, definir restricciones, establecer relaciones entre tablas, etc.

https://iddigitalschool.com/bootcamps/para-que-se-utiliza-el-lenguaje-de-programacion-sql/

#### **Que es DML**

El Lenguaje de Manipulación de Datos, o DML por sus siglas en inglés, es el grupo de comandos responsables de manipular datos en una base de datos; esto generalmente implica insertar, editar o borrar filas en tablas SQL.

El comando SQL para insertar una nueva fila en una tabla es el comando INSERT. Si queremos añadir una fila a la tabla Accounts para un usuario llamado Evan Johnson con un saldo de 3200 $, haríamos algo como esto:

**sql**

INSERT INTO Accounts (Name, Balance)

VALUES (‘Evan Johnson’, 3200)

https://learnsql.es/blog/que-son-ddl-dml-dql-y-dcl-en-sql/

#### **Que es DDL**

El Lenguaje de Definición de Datos, o DDL, está compuesto por los comandos responsables de crear, editar y borrar tablas SQL. Estos comandos son CREATE TABLE, ALTER TABLE, y DROP TABLE.

En los ejemplos anteriores, trabajábamos con la tabla Accounts que ya tenía una estructura y registros. ¿Pero cómo haríamos para crear esta tabla en primer lugar? Tendríamos que utilizar el comando CREATE TABLE:

**sql**

CREATE TABLE Accounts (

    ID INT,

    Name VARCHAR(255),

    Balance INT

)

https://learnsql.es/blog/que-son-ddl-dml-dql-y-dcl-en-sql/

#### **Que es DCL**

El Lenguaje de Control de Datos, o DCL para abreviar, es responsable de todo tipo de tareas administrativas en torno a la propia base de datos. La más notable es establecer permisos para los usuarios de la base de datos, lo que se hace principalmente a través de los comandos GRANT, REVOKE y DENY.

GRANT es el comando SQL responsable de conceder permisos a un usuario de la base de datos. Si quisiéramos conceder el permiso para consultar la tabla Accounts a un usuario llamado Mark, haríamos algo como esto:

**sql**

GRANT SELECT TO ‘mark’@’localhost’

https://learnsql.es/blog/que-son-ddl-dml-dql-y-dcl-en-sql/

### **Motor de bases de datos**

#### **Que son**

Un motor de base de datos (o motor de almacenamiento) es el componente de software subyacente que un sistema de administración de la base de datos (SGBD) utiliza para crear, leer, actualizar y eliminar (CRUD) datos de una base de datos. La mayoría de sistemas de administración de la base de datos incluyen su interfaz de programación de aplicación propia (API) que permite al usuario interaccionar con su motor subyacente sin pasar por la interfaz de usuario del SGBD.

El término de “motor de base de datos” es frecuentemente llamado “servidor de base de datos” o “sistema de administración de la base de datos”. Un caso de base de datos’ se refiere a los procesos y estructuras de memoria de la base de datos de un motor de base de datos.

https://es.wikipedia.org/wiki/Motor\_de\_base\_de\_datos

#### **Que motores ofrece SQLzoo**

SQLZoo ofrece los motores: SQL Server, Oracle, MySQL, DB2, Mimer, PostgreSQL, SQLite y Access.

#### **Que bases de datos ofrece SQLzoo**

SQLZoo utiliza bases de datos de ejemplo para practicar, como: Nobel, BBC, World, Movie, Scottish Parliament, entre otras. Estas contienen información real o simulada (países, premios, películas, etc.) y sirven para aprender a hacer consultas en SQL.

## **Practica**

### **A. SQL Zoo**

#### **0 SELECT basics**

![](data:None;base64,)

#### **1 SELECT name**

![](data:None;base64,)

![](data:None;base64,)

![](data:None;base64,)

![](data:None;base64,)

![](data:None;base64,)

![](data:None;base64,)

![](data:None;base64,)

![](data:None;base64,)

![](data:None;base64,)

![](data:None;base64,)

![](data:None;base64,)

![](data:None;base64,)

![](data:None;base64,)

![](data:None;base64,)

#### **2 SELECT from WORLD**

![](data:None;base64,)

![](data:None;base64,)

![](data:None;base64,)

![](data:None;base64,)

![](data:None;base64,)

![](data:None;base64,)

![](data:None;base64,)

![](data:None;base64,)

![](data:None;base64,)

![](data:None;base64,)

![](data:None;base64,)

![](data:None;base64,)

![](data:None;base64,)

#### **3 SELECT from NOBEL**

![](data:None;base64,)

![](data:None;base64,)

![](data:None;base64,)

![](data:None;base64,)

![](data:None;base64,)

![](data:None;base64,)

![](data:None;base64,)

![](data:None;base64,)

![](data:None;base64,)

![](data:None;base64,)

![](data:None;base64,)

![](data:None;base64,)

#### **4 SELECT within SELECT**

![](data:None;base64,)

![](data:None;base64,)

![](data:None;base64,)

![](data:None;base64,)

![](data:None;base64,)

![](data:None;base64,)

![](data:None;base64,)

![](data:None;base64,)

![](data:None;base64,)

![](data:None;base64,)

#### **5 SUM and COUNT**

![](data:None;base64,)

![](data:None;base64,)

![](data:None;base64,)

![](data:None;base64,)

![](data:None;base64,)

![](data:None;base64,)

![](data:None;base64,)

![](data:None;base64,)

#### **Quices**

1.

![](data:None;base64,)

1.

![](data:None;base64,)

1.

![](data:None;base64,)

1.

![](data:None;base64,)

1.

![](data:None;base64,)

![](data:None;base64,)

### **B. Re-escritura de consultas**

1.

**sql**

SELECT yr, subject, winner

  FROM nobel

 WHERE yr = 1950

$$  
{t.yr, t.subject, t.winner | t \in nobel \land t.yr = 1950}  
$$

1.

**sql**

SELECT winner

  FROM nobel

 WHERE yr = 1962

   AND subject = 'literature'

$$  
{t.winner | t \in nobel \land t.yr = 1962 \land t.subject = ‘literature’}  
$$

1.

**sql**

SELECT yr,subject

FROM nobel

WHERE winner='Albert Einstein'

$$  
{t.yr | t \in nobel \land t.winner = ‘Albert Einstein’}  
$$

1.

**sql**

SELECT winner

FROM nobel

WHERE subject='peace' and yr>=2000

$$  
{t.winner | t \in nobel \land t.subjec=‘peace’ \land t.yr \ge 2000}  
$$

1.

**sql**

SELECT yr,subject,winner

FROM nobel

WHERE yr>=1980 and yr<=1989 and subject='literature'

$$  
{t.yr.t.subject,t.winner | t \in nobel \land t.subjec=‘literature’ \land  
t.yr \in [1980, 1989]}  
$$

### **Funciones**

1.

Contar el numero de bandas que empizan con la letra ‘S’

**sql**

SELECT count(name) AS numero

FROM band

WHERE band\_name LIKE 'S%'

1.

Promedio del contacto de todas las bandas

**sql**

SELECT AVG(band\_contact) AS promedio\_contacto

FROM band;

1.

Banda con mayor contactos

**sql**

SELECT MAX(band\_contact)

FROM band

### **Esquemas**

1.

Saber cuantas bandas por tipo hay siempre y cuando haya mas de una

**sql**

SELECT band\_type, COUNT(\*) AS total

FROM band

GROUP BY band\_type

HAVING COUNT(\*) > 1;

1.

Ordendar las bandas por numero de contacto

**sql**

SELECT band\_name, band\_contact

FROM band

ORDER BY band\_contact DESC;

1.

Obtener los tipos de bandas sin repetir

**sql**

SELECT DISTINCT band\_type

FROM band;