

Unidad 4.

SQL

Prof: Alberto Rivera.

Índice

- ❑ Introducción a SQL
- ❑ Elementos Básicos SQL
- ❑ Comandos DDL
- ❑ Comandos DML
- ❑ Comandos DCL
- ❑ Comandos TCL
- ❑ Modificadores: LIKE
- ❑ JOIN
- ❑ Agregadores: COUNT
- ❑ Manejo de Cadenas STRING
- ❑ Filtros: HAVING

SQL

- ❑ **Structured Query Language** (Lenguaje de Consulta Estructurado), fue desarrollado por IBM en los años 70 como una respuesta a la necesidad de un sistema de gestión de bases de datos relacionales más eficiente y accesible.
- ❑ Se ha convertido en el estándar para gestionar y manipular datos en bases de datos relacionales, actuando como un intermediario que permite a los usuarios interactuar con los datos sin necesidad de conocer la complejidad técnica subyacente.
- ❑ Características principales de SQL:
 - ❑ **Facilidad de uso:** Su sintaxis declarativa permite a los usuarios especificar qué resultados desean obtener, sin preocuparse por cómo se obtienen.
 - ❑ **Flexibilidad:** Facilita desde consultas simples hasta operaciones complejas que involucran múltiples tablas.
 - ❑ **Estandarización:** Es un estándar reconocido, lo que permite a los desarrolladores transferir habilidades entre diferentes sistemas de bases de datos.

SQL

❑ ¿Por qué es importante?

- **Amplia adopción:** Es el lenguaje principal de las bases de datos relacionales, lo que lo convierte en una habilidad fundamental en áreas como análisis de datos, desarrollo de software y gestión empresarial.
- **Integridad de datos:** Permite definir restricciones y reglas para garantizar la consistencia y precisión de los datos almacenados.
- **Alta demanda laboral:** SQL es una de las habilidades más buscadas en el mercado tecnológico actual:
 - **Gestión empresarial:** Para almacenar información de clientes, productos, ventas y finanzas.
 - **Análisis de datos:** Los analistas extraen datos relevantes y generan informes utilizando consultas avanzadas.
 - **Desarrollo web:** Proporciona la base para aplicaciones dinámicas como comercio electrónico y redes sociales.
 - **Sistemas integrados:** En dispositivos móviles, sistemas de IoT y aplicaciones embebidas.

SQL: Elementos Básicos

- ❑ El lenguaje SQL está compuesto por:
 - **COMANDOS.**
 - **CLÁUSULAS.**
 - **LITERALES.**
 - **OPERADORES.**
 - **FUNCIONES.**
 - **IDENTIFICADORES.**
- ❑ Todos estos elementos se combinan en las instrucciones y se utilizan para crear, actualizar y manipular bases de datos.

SQL: Elementos Básicos

❑ COMANDOS

- Van a ser las **instrucciones** que se pueden crear en SQL. Se pueden distinguir en tres grupos:
 - De definición de datos (**DDL**, Data Definition Language), que permiten crear y definir nuevas bases de datos, tablas, campos, etc.
 - De manipulación de datos (**DML**, Data Manipulation Language), que permiten generar consultas para ordenar, filtrar y extraer datos de la base de datos.
 - De control y seguridad de datos (**DCL**, Data Control Language), que administran los derechos y restricciones de los usuarios.

❑ CLÁUSULAS:

- También llamadas **condiciones** o **criterios**, son palabras especiales que permiten modificar el funcionamiento de un comando.

❑ LITERALES

- Les podemos llamar también **constantes** y serán valores concretos, como por ejemplo un número, una fecha, un conjunto de caracteres, etc.

SQL: Elementos Básicos

❑ **OPERADORES:** permiten crear expresiones complejas.

- Tipos principales:
 - Aritméticos: Realizan cálculos matemáticos.
 - $+$, $-$, $/$, $\%$ (módulo), $+=$, $-=$, $/=$, ...
 - De comparación: Comparan valores.
 - $=$, $!=$ ($<>$), $<$, $>$, $<=$, $>=$.
 - Lógicos: Combinan condiciones.
 - **AND**: Ambas condiciones deben ser verdaderas.
 - **OR**: Al menos una condición debe ser verdadera.
 - **NOT**: Invierte el resultado de una condición.
- Ejemplo:
 - **SELECT * FROM** empleados **WHERE** salario **>** 50000 **AND** departamento = 'Ventas';

SQL: Elementos Básicos

❑ Funciones: FUNCIONES:

- son operaciones predefinidas que realizan tareas comunes sobre los datos.
- Funciones comunes en SQL:
 - Agregación:
 - **SUM()**: Suma de valores.
 - **AVG()**: Promedio de valores.
 - **COUNT()**: Número de filas o valores.
 - **MAX()**: Valor máximo.
 - **MIN()**: Valor mínimo.
 - De fecha y hora:
 - **NOW()**: Fecha y hora actuales.
 - **DATE()**: Extrae la fecha de un valor de fecha y hora.

SQL: Elementos Básicos

❑ Funciones: (cont.)

- Funciones comunes en SQL:
 - De texto:
 - **CONCAT()**: Une cadenas de texto.
 - **UPPER()**: Convierte texto a mayúsculas.
 - **LOWER()**: Convierte texto a minúsculas.
- Ejemplo:
 - **SELECT** departamento, **SUM**(salario) **FROM** empleados **GROUP BY** departamento;

SQL: Elementos Básicos

❑ IDENTIFICADORES:

- Son nombres que se asignan a los objetos de una base de datos, como tablas, columnas, índices, vistas y esquemas.
- **Reglas para crear identificadores:**
 - Deben comenzar con una **letra**.
 - No pueden coincidir con **palabras clave reservadas** del lenguaje SQL (como **SELECT** o **WHERE**).
 - Pueden contener letras, números y algunos caracteres especiales como guiones bajos (**_**).
 - Es recomendable que sigan un **formato coherente** y descriptivo.
- Ejemplo:
 - Una tabla llamada **empleados** puede tener columnas como **nombre**, **apellido** y **salario**.

SQL: Elementos Básicos

❑ Para la escritura del código SQL tendremos que seguir unas normas sencillas pero primordiales:

- Todas las instrucciones terminan con un signo de **punto y coma**.
- No se distingue entre **mayúsculas y minúsculas**.
- Cualquier comando puede separarse con **saltos de línea o espacios** para facilitar su lectura y comprensión.
- Los **comentarios**:
 - De una sola línea: Utilizando -- al inicio de la línea.
 - De bloque (multilinea): Se usan los delimitadores /* y */ para comentar varias líneas.
- Se pueden **tabular líneas** para facilitar la lectura si fuera necesario.

SQL: Tipos de Datos

❑ **Tipos de Datos:** dominio al que pertenece cada columna de una tabla, y que determina qué tipo de información puede almacenarse.

❑ Categorías principales:

- Numéricos:
 - **INT:** Números enteros.
 - **FLOAT:** Números con decimales.
 - **DECIMAL(a,b):** Números decimales con precisión definida (antes y después de la coma).
- Cadenas de texto:
 - **VARCHAR(n):** Texto de longitud variable (hasta *n* caracteres).
 - **CHAR(n):** Texto de longitud fija (*n* caracteres; los no utilizados se rellenan).
 - **TEXT:** Texto de gran longitud sin límite específico.
- Booleanos:
 - **BOOLEAN:** Representa valores verdadero/falso.

SQL: Tipos de Datos

❑ Categorías principales:

- Fechas y tiempos:
 - **DATE**: Solo fecha.
 - **TIME**: Solo hora.
 - **DATETIME**: Fecha y hora combinadas.
 - **TIMESTAMP**: Fecha y hora con zona horaria.

▪ Listados:

- **ENUM**: conjunto limitado de valores predefinidos. Es útil para almacenar datos que solo pueden tomar uno de varios valores posibles, mejorando la integridad y consistencia de los datos.

❑ Ejemplo:

```
CREATE TABLE Empleados ( ..., genero ENUM('Masculino', 'Femenino', 'No  
especificado') NOT NULL DEFAULT 'Masculino');
```


DDL

- ❑ El Lenguaje de Definición de Datos (DDL) es parte del SQL utilizado para definir y modificar la estructura de las bases de datos. Los comandos DDL permiten crear, modificar y eliminar objetos de bases de datos como tablas, índices y esquemas.
- ❑ Principales comandos DDL:
 - - CREATE
 - - ALTER
 - - DROP
 - - RENAME
 - - TRUNCATE
 - - COMMENT
 - - SHOW
 - - DESCRIBE

DDL: CREATE

- ❑ El comando **CREATE** se utiliza para crear nuevas bases de datos y sus objetos, como tablas, índices y vistas.
- ❑ Antes de crear la tabla es conveniente planificar algunos detalles:
 - Qué nombre le vamos a dar a la tabla.
 - Qué nombre le vamos a dar a cada una de las columnas.
 - Qué tipo y tamaño de datos vamos a almacenar en cada columna.
 - Qué restricciones tenemos sobre los datos.
 - Alguna otra información adicional que necesitemos.
- ❑ Reglas que se deben cumplir para los nombres de las tablas:
 - No podemos tener nombres de tablas duplicados en un mismo esquema (usuario).
 - Deben comenzar por un carácter alfabético.
 - Su longitud máxima es de 30 caracteres.
 - Solo se permiten letras del alfabeto inglés, dígitos o el signo de guion bajo.
 - No puede coincidir con las palabras reservadas de SQL (comandos como: WHERE, SELECT, ...).
 - No se distingue entre mayúsculas y minúsculas.

DDL: CREATE

- ❑ La sintaxis básica del comando que permite crear una base de datos y una tabla es la siguiente:
- ❑ Sintaxis para crear una base de datos:
CREATE DATABASE [IF NOT EXISTS] NombredeMiBaseDatos
- ❑ Sintaxis básica para crear una tabla:
CREATE TABLE nombre_tabla (
columna1 tipo_dato [restricción de columna],
columna2 tipo_dato [restricción de columna],
...
[restricción de tabla],
[restricción de tabla],
...
);

Ej: Crear una tabla de Departamentos.

```
CREATE TABLE Departamentos (  
  id INT PRIMARY KEY,  
  nombre VARCHAR(50) NOT NULL,  
  fecha_creacion DATE);
```