**Integrantes:**

Jose David Daza Camacho

Jose David Zabaleta Calderon

Maria Alejandra Villazon Montero

**Instructora**:

Isaura Suarez

**Ficha:**

2925960

**fecha**:

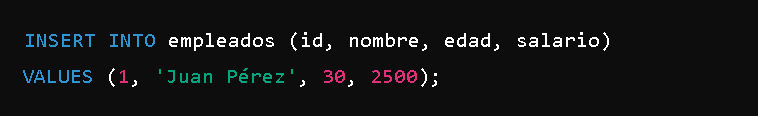
05/03/2025

**Qué es, sentencias DML Y DDL, CONSULTAS, JOINS).**

El DML (lenguaje de Manipulación de Datos) se usa para gestionar los datos dentro de la base de datos. Incluye operaciones como insertar, actualizar, eliminar y seleccionar datos.

**INSERT** → Agregar datos a una tabla.

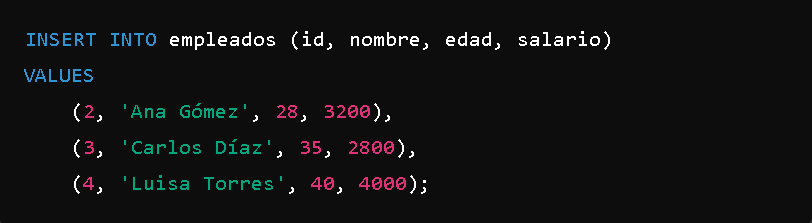
Ejemplo 1→ Insertar un solo registro.



Estamos insertando un empleado con id=1, nombre= ´juan perez´, edad=30 y salario=2500.

Solo insertamos datos en una columna que mencionamos en INSERT INTO.

Ejemplo 2→ Insertar múltiples registros.

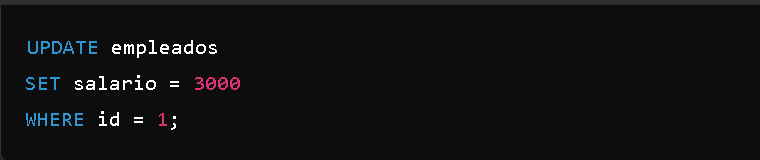


Insertamos varios registros en una sola sentencia.

Cada conjunto de valores está separado por una coma.

**UPDATE** → Modificar datos existentes.

Ejemplo 1→ Modificar el salario de un empleado.



Cambia el salario a 3500 solo para el empleado cuyo id es 1.

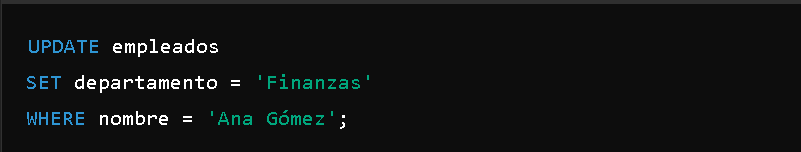
en la cláusula WHERE es necesaria para evitar actualizar todos los registros de la tabla.

Ejemplo 2 → Aumentar el salario en 10% para todos los empleados.



modifica el salario de todos los empleados multiplicando por 1.10 (10% más).

Ejemplo 3 → Cambiar el departamento de un departamento de un empleado especifico

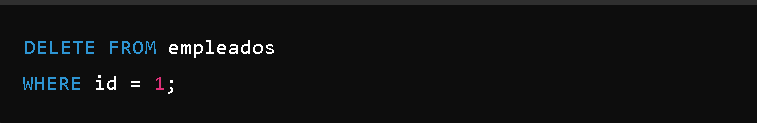


Cambia el departamento de “Ana Gómez” a “Finanzas”.

y si hay más de una persona con el mismo nombre, todas serán afectadas.

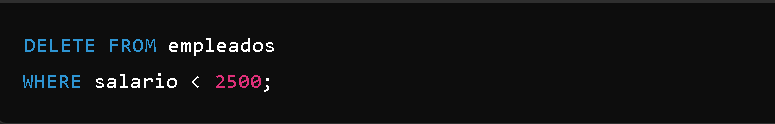
**DELETE** → Eliminar datos de una tabla.

Ejemplo 1→ Eliminar un empleado específico.



Se elimina el empleado con el id=3. y sin WHERE, se eliminan todos los empleados de la tabla.

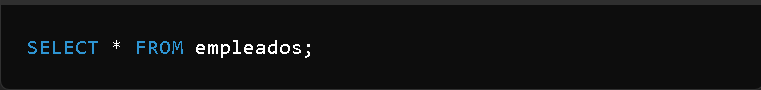
Ejemplo 2→ Eliminar empleados con salario menor a 2500.



borra a todos los empleados que tengan un salario menor a 2500.

**SELECT** → Recuperar información de una tabla.

Ejemplo 1→ Obtener todos los empleados



el \* significa que seleccionamos todas las columnas de la tabla empleados.

Ejemplo 2→ Obtener nombres y salarios de empleados.



solo se selecciona las columnas nombre y salario de la tabla empleados.

Ejemplo 3→ Filtrar empleados con salario mayor a 3000.



esto hace que solo devuelve los empleados que tienen un salario mayor a 3000.

Ejemplo 4→ Ordenar empleados por edad.



esto hace que se ordene de mayor a menor en la columna de edad.

Las sentencias DDL (Data Definition Language) son un tipo de comandos utilizados en bases de datos para definir y modificar la estructura de los datos y los objetos de la base de datos, como tablas, índices, vistas, esquemas, entre otros. A diferencia de las sentencias DML (Data Manipulation Language), que se utilizan para manipular los datos dentro de las tablas (como INSERT, UPDATE, DELETE), las sentencias DDL se enfocan en la creación, modificación y eliminación de la estructura de la base de datos.

Comando create: se utiliza para nuevos objetos en la base de datos, como tablas, indices, vistas etc

**CREATE TABLE empleados (**

**id INT PRIMARY KEY,**

**nombre VARCHAR(50),**

**salario DECIMAL(10, 2)**

**);**

Alter: se utiliza para modificar la estructura de un objeto existente como agregar o eliminar columnas en una tabla

**ALTER TABLE empleados ADD COLUMN departamento VARCHAR(50);**

**Drop: se utiliza para eliminar objetos de la base de datos, tablas, indices, vistas, etc.**

**DROP TABLE empleados;**

Truncate:

Se utiliza para eliminar todos los datos de una tabla, pero manteniendo la estructura de la tabla

**TRUNCATE TABLE empleados;**

**Rename: se utiliza para cambiar el nombre de un objeto en la base de datos**

**ALTER TABLE empleados RENAME TO empleados nuevo;**

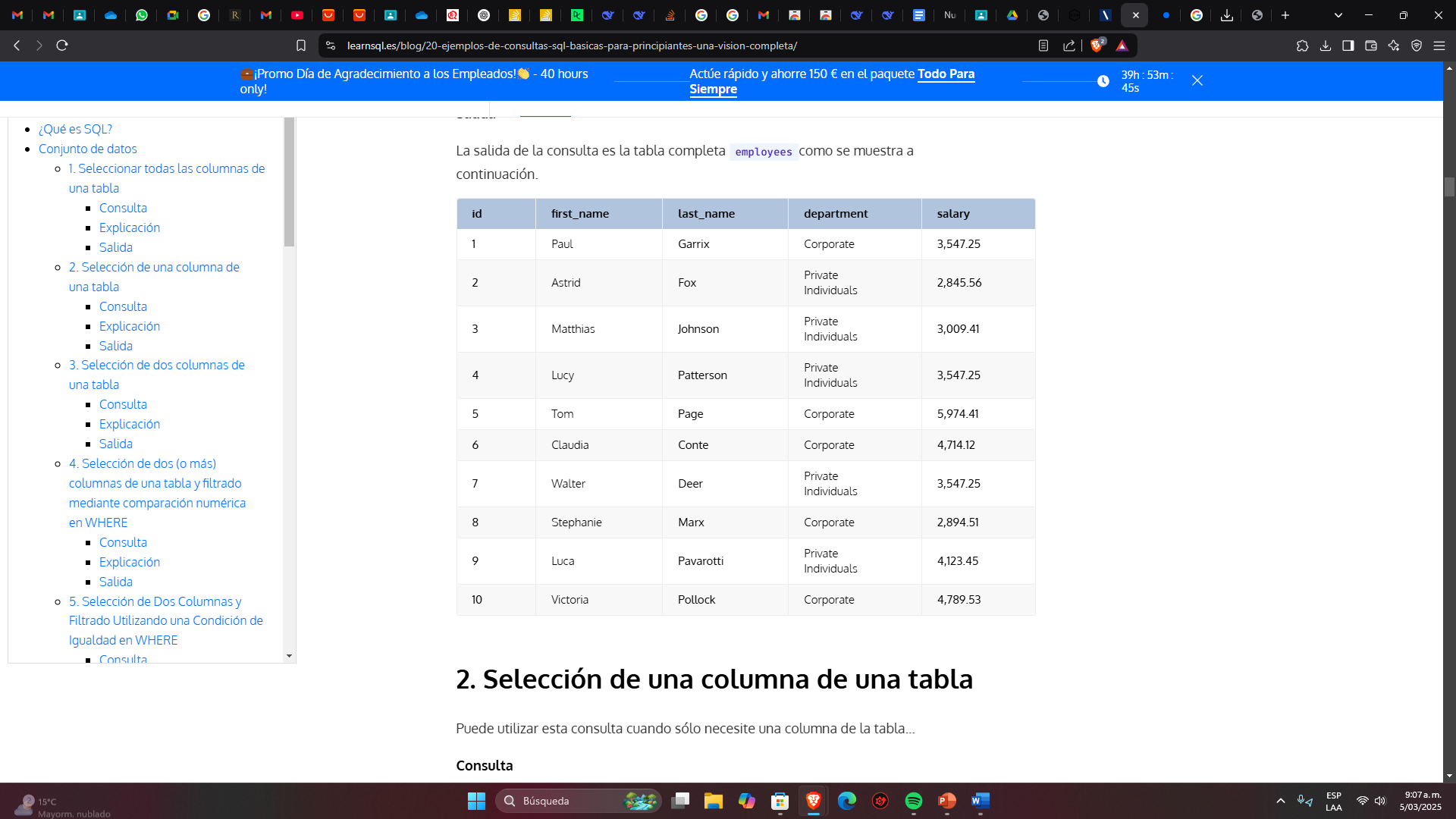
**Comment: se utiliza para agregar comentarios a las tablas o columnas**

**Consultas**

Es una solicitud de información a una operación realizada sobre los datos almacenados en una base de datos, estas consultas se utilizan para recuperar, filtrar, ordenar, agrupar o manipular datos , estas operaciones se realizan mediante comandos específicos, principalmente utilizando el lenguaje sql

Algunos ejemplos de comandos de consulta anteriormente mencionados

**Select \*:** Se usa para seleccionar todas las columnas de una tabla Ejemplo: SELECT \* FROM employees; La salida de la consulta seria toda la tabla “employees” Selección de una columna de una tabla SELECT first\_name FROM employees; En este caso selecciona el nombre de la columna que se desea consultar y la tabla “employees “ Esta nos devuelve la columna completa



**Group by:** se utiliza para agrupar resultados de una o mas columnas

ejemplo:

Consulta:

**SELECT**

**department,**

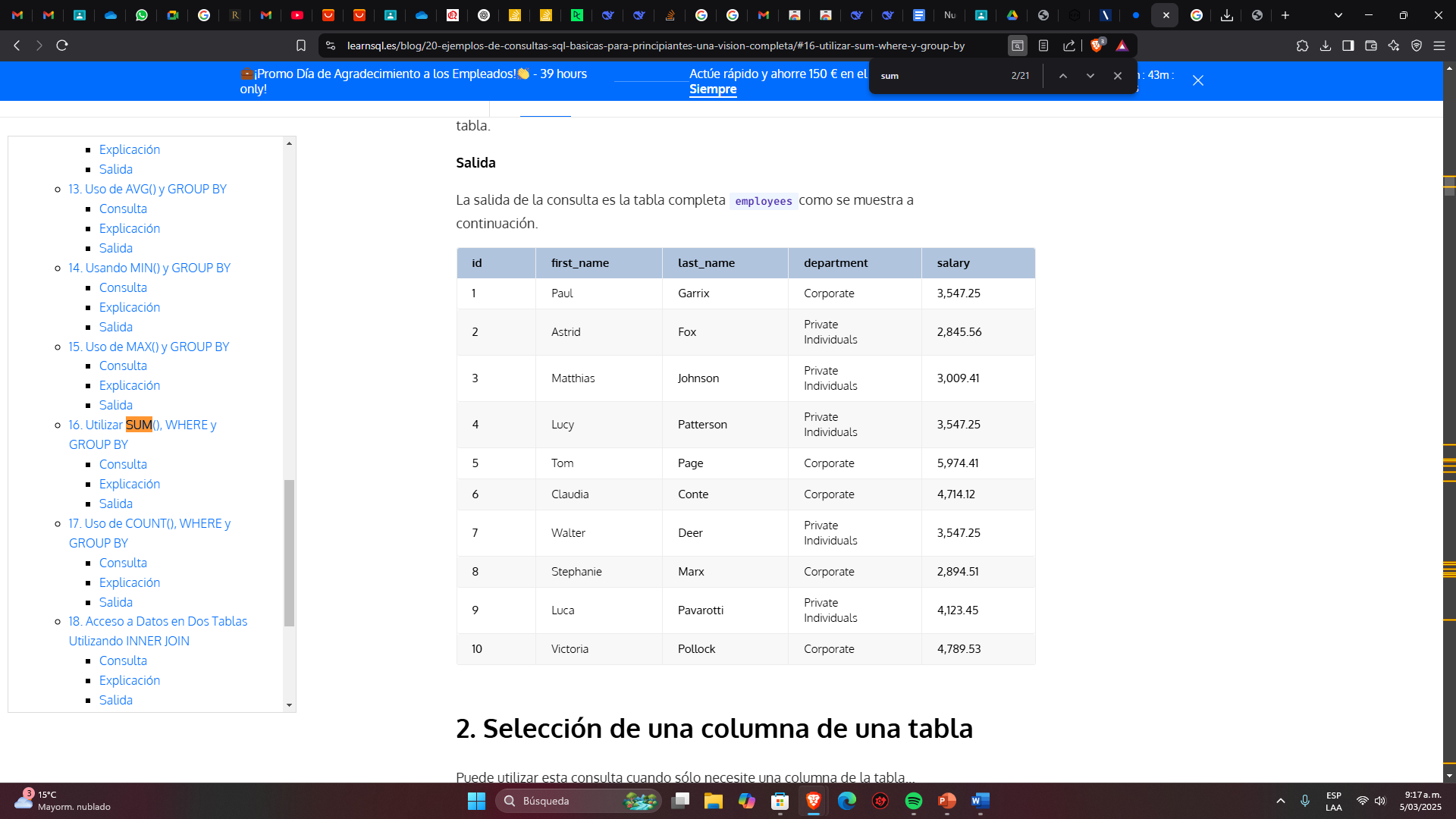
**SUM(salary) AS total\_salary**

**FROM employees**

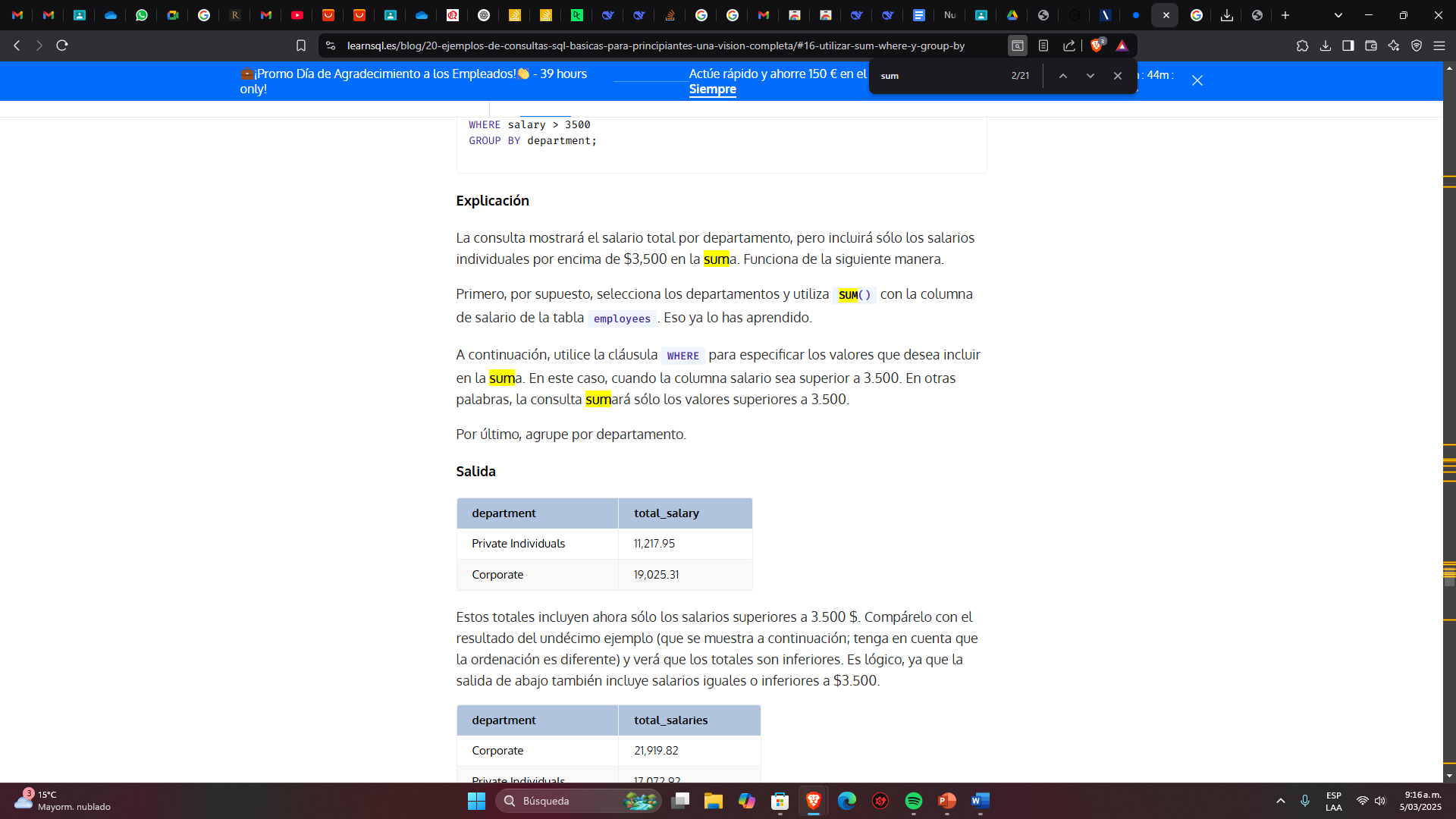
**WHERE salary > 3500**

**GROUP BY department;**

**tabla principal:**



**resultado final**



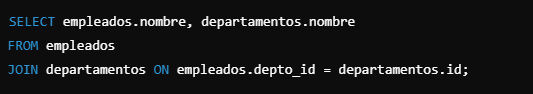
La sentencia JOIN en base de datos es una instrucción SQL que permite combinar registros de dos o más tablas basándose en una columna en común. se usa para relacionar datos que están almacenados en distintas tablas, optimizando la organización y reduciendo la redundancia

En bases de datos relacionales, los datos suelen dividirse en varias tablas para mantener la normalización. **JOIN** permite recuperar información combinada de esas tablas en una sola consulta.

Por ejemplo, en una base de datos de empleados:

* Una tabla empleados tiene la información de cada trabajador.
* Una tabla departamentos almacena las áreas de la empresa.
* Ambas están relacionadas por depto\_id.

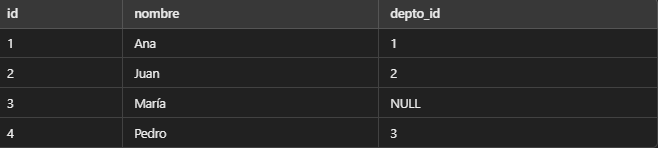
Para obtener los nombres de los empleados junto con el nombre de su departamento, usamos:

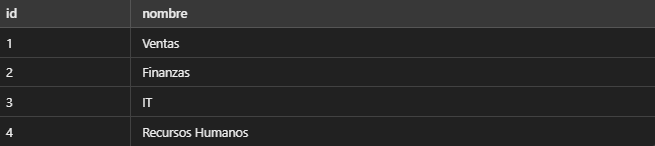


**TIPOS DE JOIN**

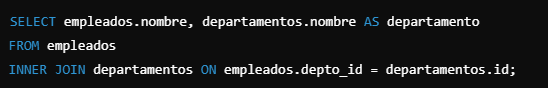
**Ejemplo:**

**Tabla empleados:**



**Tabla departamentos:**  


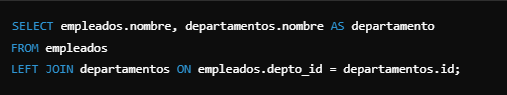
**INNER JOIN:** Devuelve solo los empleados que tiene un departamento asignado



resultado:



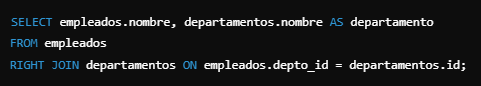
**LEFT JOIN:** Devuelve todos los empleados y sus departamentos, incluyendo los que no tienen departamento



resultado:



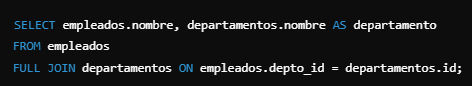
**RIGHT JOIN:** devuelve todos los departamentos y los empleados asignados a ellos(NULL si no hay empleados)



resultado:



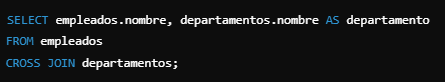
**FULL JOIN**: Devuelve todos los empleados y todos los departamentos, combinando coincidencias y colocando NULL en los datos faltantes.



resultado:



**CROSS JOIN:** Devuelve todas las combinaciones posibles entre empleados y departamentos.



resultado:

