Contenido

[¿Qué es SQL y para qué se utiliza en ciencia de datos?✔️ 2](#_Toc204674165)

[¿Cuál es la diferencia entre WHERE y HAVING? ✔️ 3](#_Toc204674166)

[Ejercicios con where 6](#_Toc204674167)

[Ejercicios con having 7](#_Toc204674168)

[¿Qué diferencia hay entre INNER JOIN, LEFT JOIN, RIGHT JOIN y FULL OUTER JOIN? ✔️ 8](#_Toc204674169)

[¿Cuál es la diferencia entre COUNT(\*) y COUNT(columna)? ✔️ 9](#_Toc204674170)

[¿Qué significa NULL en SQL y cómo se maneja? ✔️ 10](#_Toc204674171)

[¿Qué es una clave primaria y una clave foránea? ✔️ 11](#_Toc204674172)

[¿Qué hace la cláusula GROUP BY? ✔️ 11](#_Toc204674173)

[¿Cuál es la diferencia entre DISTINCT y GROUP BY? ✔️ 13](#_Toc204674174)

[¿Qué es una subconsulta y cuándo se usaría? 14](#_Toc204674175)

[¿Qué diferencia hay entre UNION y UNION ALL?✔️ 17](#_Toc204674176)

[**🟢 Básico (1-10)** 19](#_Toc204674177)

[¿Qué tipos de JOIN existen?✔️ 19](#_Toc204674178)

[¿Qué hace el comando GROUP BY?✔️ 20](#_Toc204674179)

[¿Cuál es la diferencia entre NULL, 0 y una cadena vacía ('')?✔️ 20](#_Toc204674180)

[¿Cómo eliminarías registros duplicados de una tabla?\* 20](#_Toc204674181)

[¿Qué función cumple DISTINCT?✔️ 21](#_Toc204674182)

[¿Qué es una subconsulta (subquery) y cómo se diferencia de una JOIN?✔️ 21](#_Toc204674183)

[**🟡 Intermedio (11–20)** 21](#_Toc204674184)

[¿Cuál es la diferencia entre INNER JOIN y LEFT JOIN en la práctica?✔️ 21](#_Toc204674185)

[¿Qué son las funciones de agregación? Menciona al menos cinco ejemplos.✔️ 22](#_Toc204674186)

[¿Cuál es la diferencia entre UNION y UNION ALL?✔️ 23](#_Toc204674187)

[¿Cómo actualizarías múltiples filas en una tabla con una sola sentencia UPDATE?❕ 23](#_Toc204674188)

[¿Qué es una tabla derivada y cómo se usa?✔️ 23](#_Toc204674189)

[¿Cómo contarías el número de empleados por departamento, pero solo de los que ganan más de $1000?✔️ 24](#_Toc204674190)

[¿Qué significa normalización de datos y por qué es importante?✔️ 25](#_Toc204674191)

[¿Qué es un índice y cómo afecta el rendimiento de las consultas?✔️ 25](#_Toc204674192)

[¿Cuál es el orden de ejecución de una consulta SQL?✔️ 26](#_Toc204674193)

[**🔵 Avanzado (21–30)❕** 27](#_Toc204674194)

[¿Qué es una CTE (Common Table Expression) y cuándo usarla? 27](#_Toc204674195)

[¿Cómo calcularías un promedio móvil con SQL? 27](#_Toc204674196)

[¿Qué diferencias hay entre EXISTS, IN y NOT IN? 27](#_Toc204674197)

[¿Cómo manejarías valores NULL en funciones agregadas? 27](#_Toc204674198)

[¿Qué son las funciones analíticas y cómo se diferencian de las agregadas? 27](#_Toc204674199)

[¿Cómo detectarías outliers usando SQL? 27](#_Toc204674200)

[¿Qué es una transacción y cuáles son sus propiedades (ACID)? 27](#_Toc204674201)

[¿Qué es una vista (view) y cuáles son sus ventajas y desventajas? 27](#_Toc204674202)

[¿Cómo optimizarías una consulta lenta? 28](#_Toc204674203)

[¿Cómo harías una consulta para encontrar la segunda fila más alta en una tabla de ventas? 28](#_Toc204674204)

[Enfoque en ciencia de datos 28](#_Toc204674205)

[¿Cómo identificarías a los clientes más valiosos para un negocio usando SQL o Python? ✔️ 28](#_Toc204674206)

[Si las ventas de un producto disminuyen, ¿cómo investigarías las causas usando datos? ✔️ 29](#_Toc204674207)

[¿Cómo construirías un análisis de cohortes en SQL? ¿Qué información te aporta?❕ 29](#_Toc204674208)

[Qué son las KPIs ✔️ 29](#_Toc204674209)

[¿Cómo calcularías la tasa de retención de usuarios mes a mes con SQL?✔️ 30](#_Toc204674210)

[¿Cómo agruparías datos por semana, mes o trimestre en SQL? ¿Y en Pandas? 30](#_Toc204674211)

[¿Cómo calcularías el crecimiento porcentual mes a mes de una métrica? 30](#_Toc204674212)

[¿Qué es un “lag” en series de tiempo y cómo lo aplicarías en una base de datos? 30](#_Toc204674213)

[¿Cómo identificarías una tendencia estacional en una serie temporal? 30](#_Toc204674214)

[¿Qué harías si te encuentras con timestamps en diferentes zonas horarias? 30](#_Toc204674215)

[¿Cómo detectarías outliers en una variable numérica usando SQL o Python? 30](#_Toc204674216)

[¿Qué pasos seguirías si una tabla tiene muchas columnas con más del 50% de valores nulos? 30](#_Toc204674217)

[¿Cómo tratarías datos categóricos con errores tipográficos o formatos inconsistentes (como “Male”, “M”, “male”)? 30](#_Toc204674218)

[Te entregan un dataset de usuarios y eventos. ¿Cómo calcularías la conversión desde el registro hasta una compra? 31](#_Toc204674219)

[Tienes un dataset con 100 columnas. ¿Cómo identificarías las variables más relevantes para predecir una salida (target)? 31](#_Toc204674220)

Promt: Genera una tabla con 10 observaciones y dame un ejercicio nivel Y, K, L para practicar Z en SQL. Dame también la salida esperada

# ¿Qué es SQL y para qué se utiliza en ciencia de datos?✔️

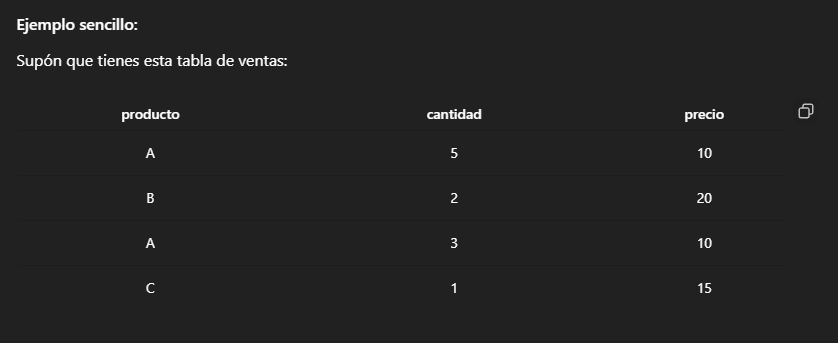
SQL es un lenguaje de consulta estructurado, **no un lenguaje de programación general**, especializado en el almacenamiento y consulta de bases de datos relacionales. Existen diferentes motores como MySQL, PostgreSQL, SQL server, etc. En la ciencia de datos lo utilizamos para almacenar datos, estructurarlos, limpiarlos y poder acceder a ellos de forma eficiente, escalable y segura

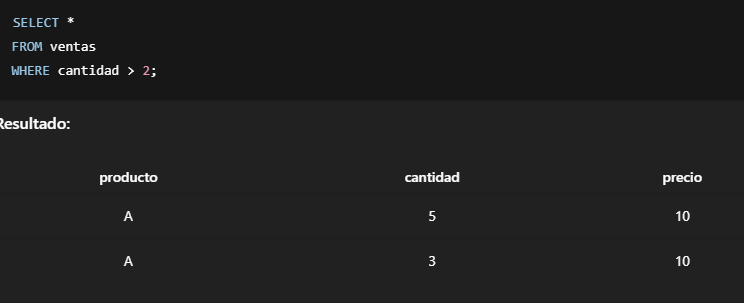
# ¿Cuál es la diferencia entre WHERE y HAVING? ✔️

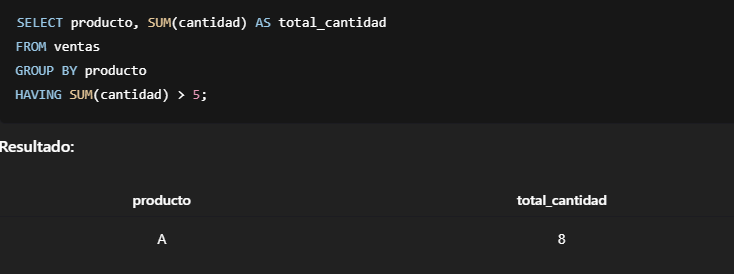
 **WHERE** se usa para **filtrar filas** *antes* de que se agrupen los datos. Es decir, filtra directamente sobre las filas de la tabla.

 **HAVING** se usa para **filtrar grupos** *después* de hacer una agregación con GROUP BY. Es decir, filtra sobre el resultado de las agrupaciones.

Resumen: where se aplica sobre la tabla bruta having se aplica tras una agrupacion

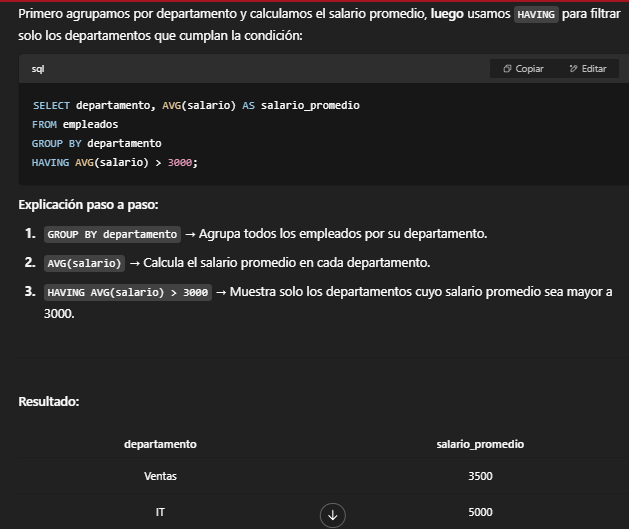






Veamos otro ejemplo





En conclusión podemos decir que el having Es una filtración de la filtración agrupada previamente

## Ejercicios con where



🟢 Ejercicio Fácil

**Pregunta:**  
Selecciona todos los clientes que vivan en Guadalajara.

Select c.nombre, c.ciudad

From clientes c

Where c.ciudad = “Guadalajara”;

🟡 Ejercicio Intermedio

**Pregunta:**  
Selecciona los clientes que tengan un saldo mayor a 2000 **y** sean menores de 40 años.

Select c.nombre

From clients c

Where c.saldo > 2000 and c.edad < 40;

🔴 Ejercicio Difícil

**Pregunta:**  
Selecciona los clientes registrados después del 1 de enero de 2022 que **no** sean de Ciudad de México **y** cuyo saldo sea inferior a 1000.

SELECT \* FROM clientes

WHERE fecha\_registro > '2022-01-01'

AND ciudad <> 'Ciudad de México'

AND saldo < 1000;

## Ejercicios con having

🟢 Ejercicio Fácil

**Pregunta:**  
Agrupa por ciudad y muestra solo aquellas donde haya **más de 1 cliente**.

Select c.ciudad, count(\*) as total\_clientes

From clientes c

Group by ciudad

having count(\*) > 1;



“Selecciona la columna ciudad de la tabla clientes y cuenta cuántas veces aparece cada ciudad, es decir, cuántos clientes hay por ciudad. Esta cuenta se almacena en una columna temporal llamada total\_clientes. Luego, se agrupan las filas según los valores únicos de ciudad usando GROUP BY. Finalmente, el filtro HAVING COUNT(\*) > 1 asegura que solo se incluyan en el resultado aquellas ciudades que tienen **más de un cliente** asociado.”

🟡 Ejercicio Intermedio

**Pregunta:**  
Agrupa por ciudad y muestra aquellas donde el **saldo promedio de los clientes supere los 2000**.

Select c.ciudad, avg(saldo) as salario\_promedio

From clientes c

Group by ciudad

Having avg(salario) > 2000;

🔴 Ejercicio Difícil❕

**Pregunta:**  
Agrupa por ciudad y muestra aquellas donde el **número de clientes menores de 30 años sea al menos 2**.

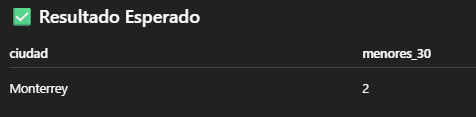
SELECT ciudad,

COUNT(CASE WHEN edad < 30 THEN 1 END) AS menores\_30

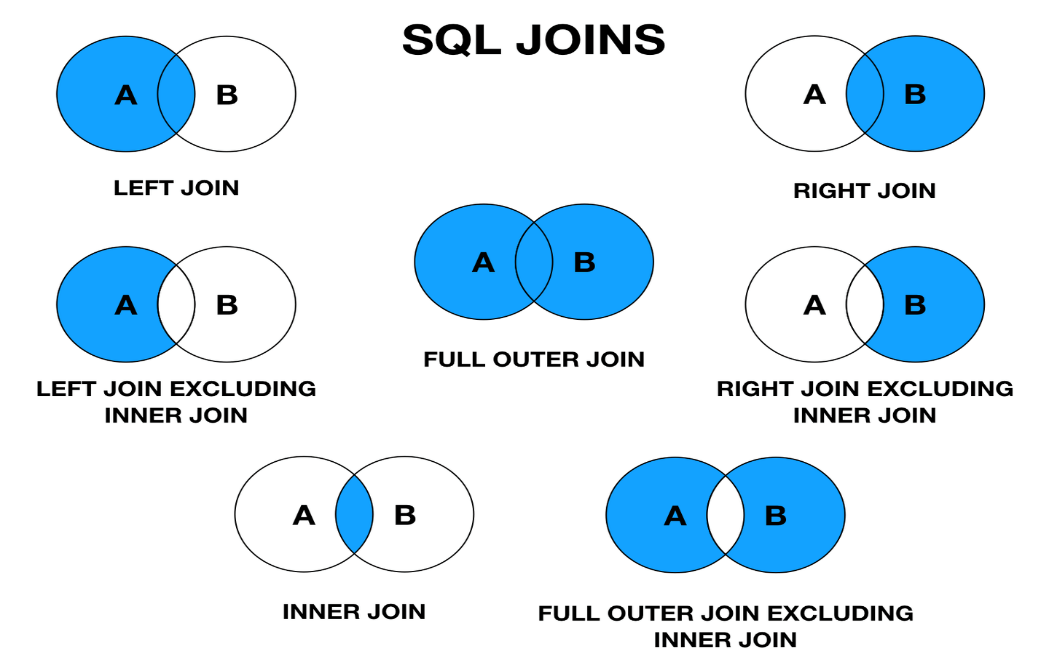
FROM clientes

GROUP BY ciudad

HAVING COUNT(CASE WHEN edad < 30 THEN 1 END) >= 2;



# ¿Qué diferencia hay entre INNER JOIN, LEFT JOIN, RIGHT JOIN y FULL OUTER JOIN? ✔️



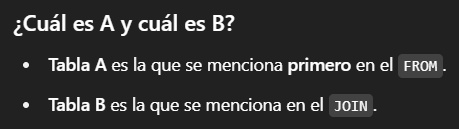
Este tipo de referencias solo existen cuando tenemos tablas relacionadas (Puede ser dos o más tablas)

Inner join: Sólo se trae la información que comparte la tabla A y la tabla B. Es decir, trae la intersección de la tabla A con la tabla B

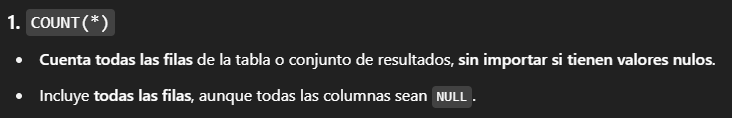
Left join: Trae a todas las observaciones de la tabla A y además la intersección con la tabla B

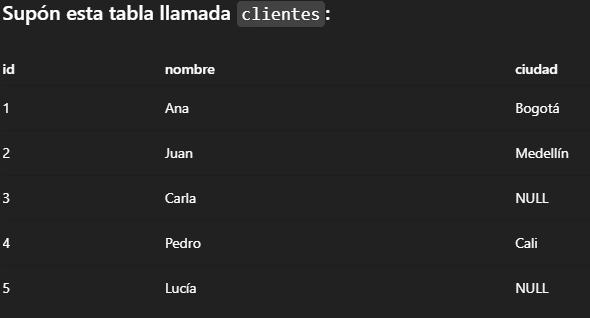
Right join: Trae todas las observaciones de la tabla B además de la intersección con la tabla A

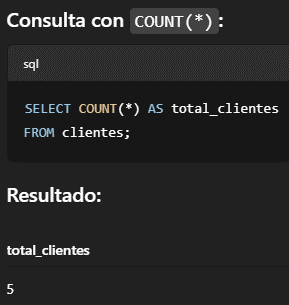
Full outer join: Trae todos los registros de la tabla A y la tabla B

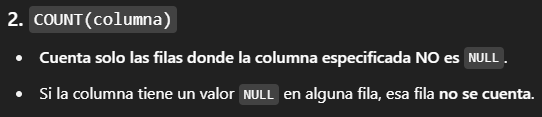


# ¿Cuál es la diferencia entre COUNT(\*) y COUNT(columna)? ✔️

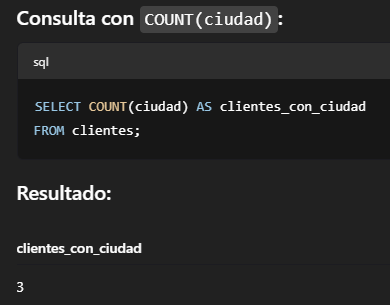








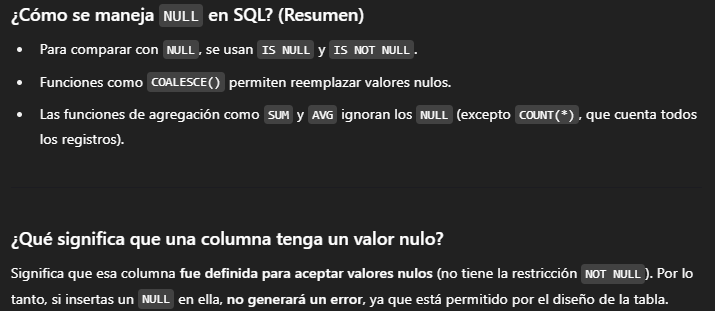
Consideremos la misma tabla que anteriormente



Es decir el count(columna) Ignora los valores nulos no los contabiliza y sin embargo el count (\*) si los contabiliza

# ¿Qué significa NULL en SQL y cómo se maneja? ✔️

Null significa que la tabla en cuastion si acepta valores nulos y por ende, la observación de esa columna no contiene valor alguno. Es decir, representa un **valor desconocido, ausente o no aplicable**. No significa "cero" ni una cadena vacía — simplemente indica que **no hay ningún valor asignado** en esa celda.



# ¿Qué es una clave primaria y una clave foránea? ✔️

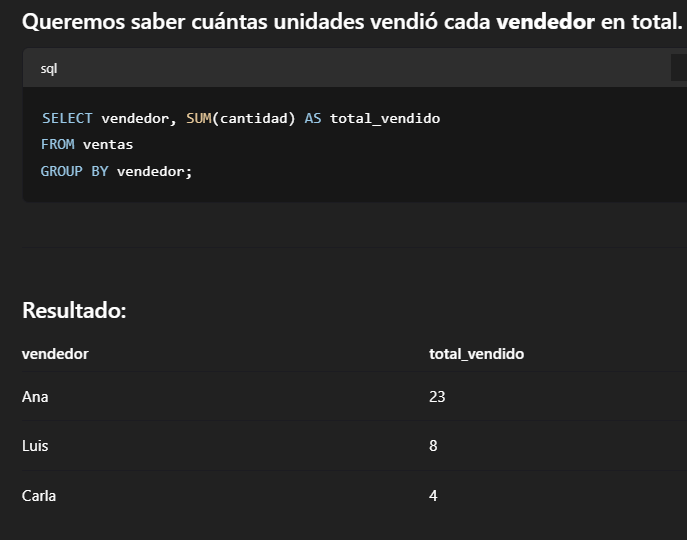
La clave primaria es un campo (o conjunto de campos) en una tabla que identifica de forma única a cada fila o registro. No puede contener valores nulos ni duplicados. Cada tabla debe tener una única clave primaria.

La clave foránea es un campo (o conjunto de campos) en una tabla que hace referencia a la clave primaria de otra tabla. Sirve para establecer relaciones entre tablas y mantener la integridad referencial de los datos.

# ¿Qué hace la cláusula GROUP BY? ✔️

La cláusula GROUP BY en SQL **agrupa las filas que tienen los mismos valores en una o más columnas** en conjuntos, para que se pueda aplicar funciones de agregación (como SUM, COUNT, AVG, etc.) a cada grupo.



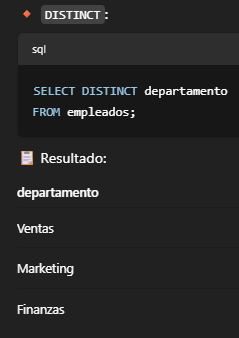


# ¿Cuál es la diferencia entre DISTINCT y GROUP BY? ✔️

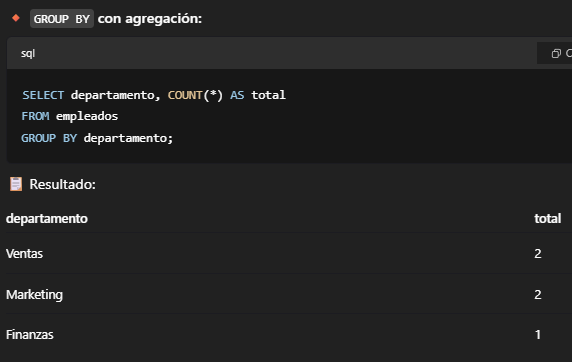
* DISTINCT Se usa para **eliminar filas duplicadas** en los resultados y mostrar solo combinaciones únicas de columnas.
* Group by Se usa para **agrupar filas** que tienen los mismos valores en columnas específicas, **para aplicar funciones de agregación** como SUM(), COUNT(), AVG(), etc.

Para comprenderlos veamos un ejemplo práctico





es decir, el DISTINCT Sirve para saber las categorías de una columna (No ignoran los valores nulos los cuenta como una categoría más) y además elimina los duplicados en la consulta



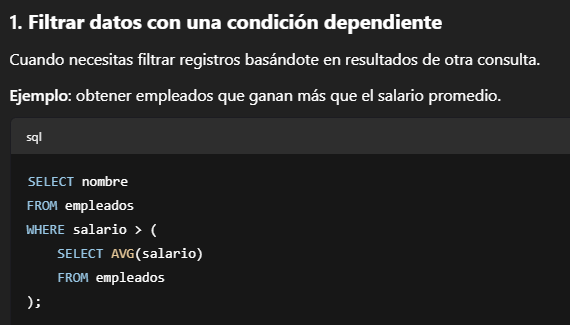
# ¿Qué es una subconsulta y cuándo se usaría?

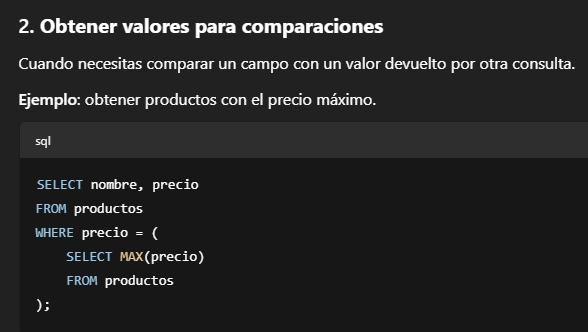
En SQL, un subquery (subconsulta) es una consulta anidada dentro de otra consulta. La subconsulta se ejecuta primero y su resultado se utiliza por la consulta externa. Los subqueries son útiles para realizar consultas más complejas y obtener datos específicos que dependen de resultados intermedios.

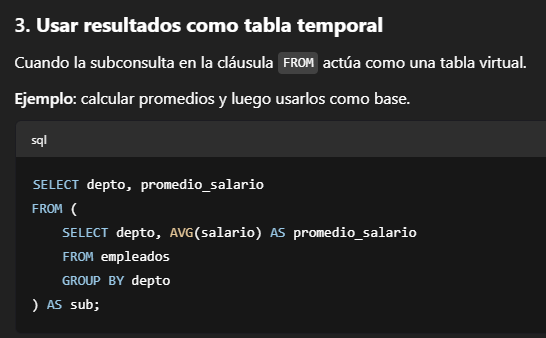
Propósito y Ventajas de los Subqueries

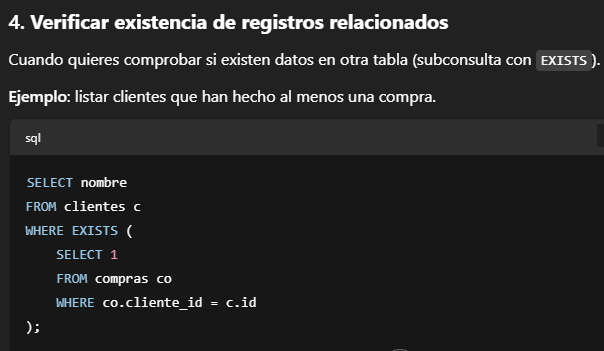
1. **Modularidad**: Facilitan la descomposición de una consulta compleja en partes más manejables.
2. **Reutilización**: Permiten el uso de resultados intermedios en varias partes de la consulta principal.
3. **Flexibilidad**: Ayudan a realizar consultas que serían difíciles o menos eficientes de realizar de otra manera.
4. **Claridad**: Pueden mejorar la legibilidad de una consulta al aislar partes específicas de la lógica

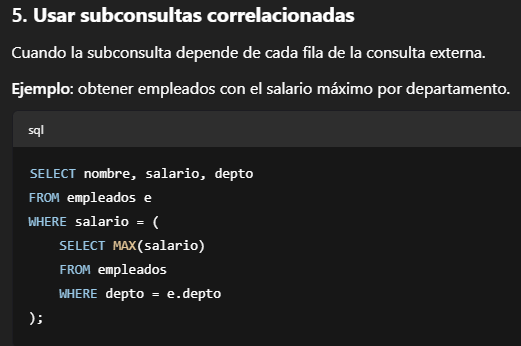
Las subconsultas se usan en SQL cuando necesitas resolver una parte de la consulta dependiendo de otro conjunto de datos





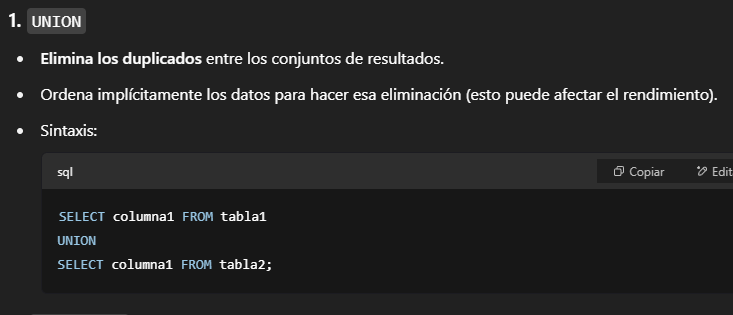


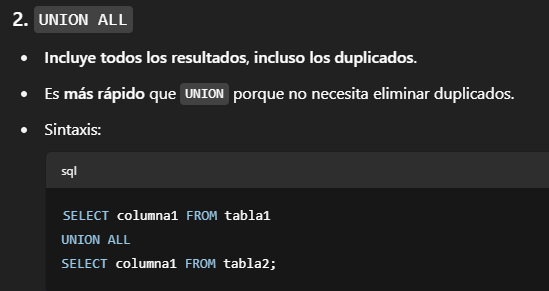




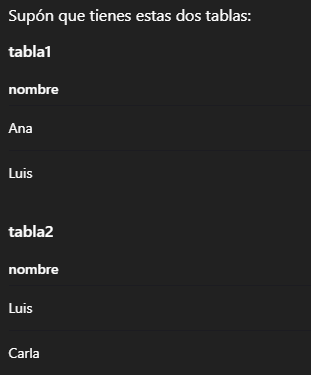
# ¿Qué diferencia hay entre UNION y UNION ALL?✔️

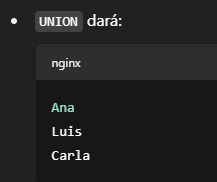
En SQL, tanto UNION como UNION ALL se utilizan para **combinar los resultados de dos o más consultas SELECT**, pero hay una diferencia clave entre ellos:

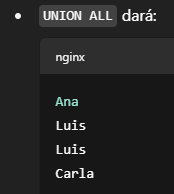




Ahora veamos un ejemplo práctico



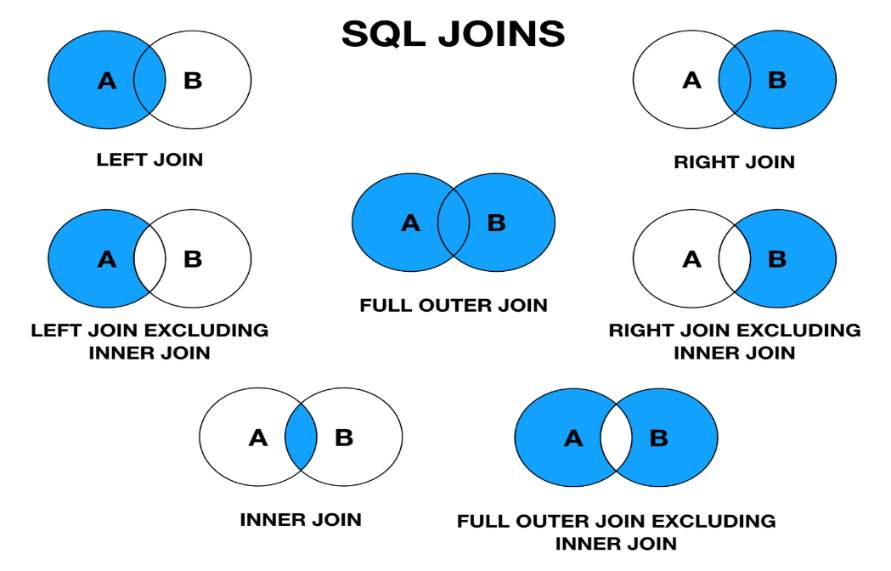




**🟢 Básico (1-10)**

# ¿Qué tipos de JOIN existen?✔️

Existen multiples tipos de join, pero los mas usuales son:



Supongamos que tenemos una tabla A y una tabla B, donde la tabla a es aquella que se pone antes del join y la tabla B es la que se pone junto al Join (en el codigo)

* Inner join: Trae la intersección de la tabla A con la tabla B
* Ful outer join excluiding inner join: Trae la table A y la table B excluyendo la intersección de éstas
* Left join: Trae la table A y la intersección con la table B
* Left join excluiding inner join: Trae la table A y excluye la intersección con la table B
* Right join: Devuelve **todas las filas de la tabla B** (derecha) y las filas coincidentes de la tabla A. Si no hay coincidencia, A tendrá valores NULL.
* Right joing excluiding inner join: Devuelve las filas de la tabla B **que no tienen coincidencia** en la tabla A.
* Full outer join: Trae la table A y la table B así como sus intersecciones

# ¿Qué hace el comando GROUP BY?✔️

El comando group by agrupa categorías de una columna según una especificacion

# ¿Cuál es la diferencia entre NULL, 0 y una cadena vacía ('')?✔️

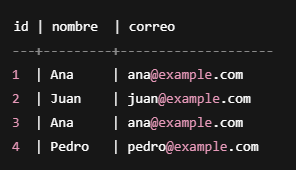
Null significa que no hay valor asignado en la celda

Cero significa que existe un valor en la celda y es el cero (ya sea como int o varchar)

Una cadena vacia significa que hay un string existente en la columna (Es decir el tipo de dato es de texto ya sea varchar o char o cualquier otro compatible)

# ¿Cómo eliminarías registros duplicados de una tabla?\*

Alterando la tabla e introduciendo una consulta con distinct



# ¿Qué función cumple DISTINCT?✔️

**DISTINCT** es una cláusula en SQL que **elimina filas duplicadas** del resultado de una consulta, mostrando **solo valores únicos** de una o más columnas. Viéndolo de una Forma distinta también sirve para saber las categorías de una columna incluyendo los valores nulos es decir los valores nulos los toma como una categoría

# ¿Qué es una subconsulta (subquery) y cómo se diferencia de una JOIN?✔️

Una subconsulta es una consulta anidada y un join es una consulta combinada donde se involucra dos o más tablas relacionadas

**🟡 Intermedio (11–20)**

# ¿Cuál es la diferencia entre INNER JOIN y LEFT JOIN en la práctica?✔️

La diferencia en el código es simplemente el nombre, porque previo a la unión está la tabla A y después la tabla B.

Select COLUMNA

From TABLA A

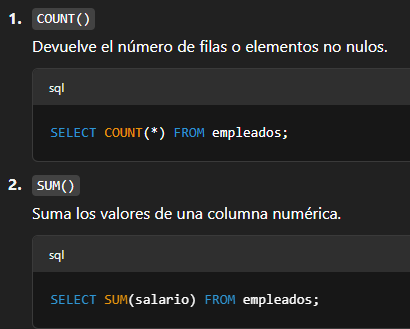
Inner or left join TABLA B

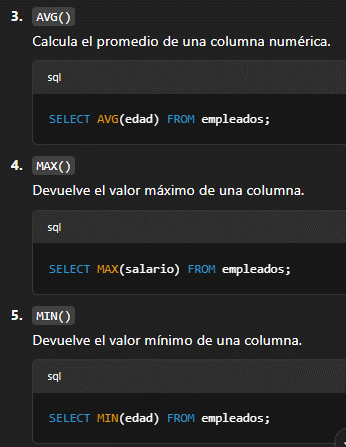
On TABLA A.id = TABLA B.id;

Y la diferencia conceptual es que el inner join trae los registros coincidentes entre ambas tablas y el left join Trae todos los registros de la tabla A y los que coinciden con la tabla B

# ¿Qué son las funciones de agregación? Menciona al menos cinco ejemplos.✔️

Las **funciones de agregación en SQL** son funciones que **operan sobre un conjunto de valores** y **devuelven un único valor de resumen**. Se utilizan comúnmente en combinación con GROUP BY para agrupar datos y obtener estadísticas o resúmenes.

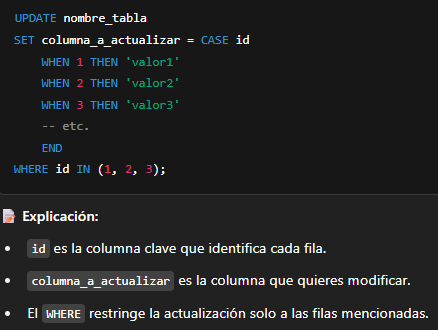




# ¿Cuál es la diferencia entre UNION y UNION ALL?✔️

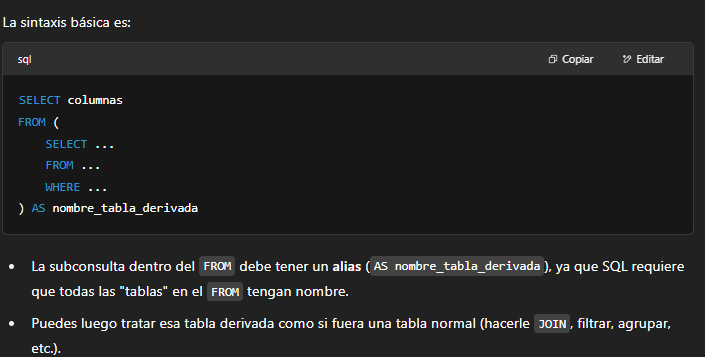
Son cláusulas que permiten unir dos o más consultas. Union permite unir dos consultas ignorando valores duplicados (los elimina) y unión all no ignora duplicados (trae todo)

# ¿Cómo actualizarías múltiples filas en una tabla con una sola sentencia UPDATE?❕

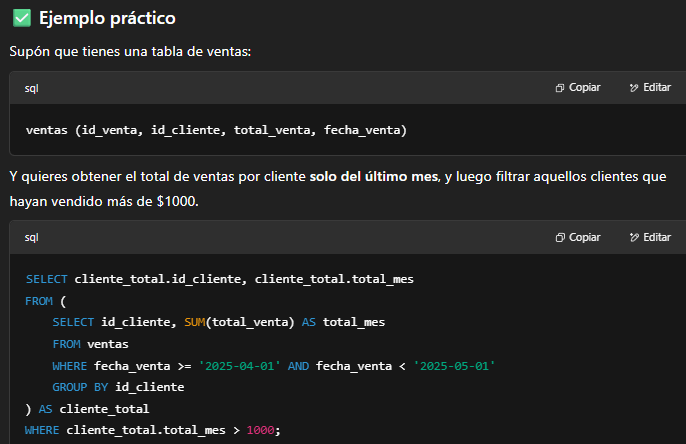


# ¿Qué es una tabla derivada y cómo se usa?✔️

Una **tabla derivada** en SQL es una **subconsulta que se coloca en la cláusula FROM** de una instrucción SELECT. Actúa como una tabla temporal que solo existe durante la ejecución de esa consulta.



Veamos un ejemplo para saber cómo funciona las tablas derivadas en una subconsulta:



# ¿Cómo contarías el número de empleados por departamento, pero solo de los que ganan más de $1000?✔️

SELECT departamento, COUNT(\*) AS num\_empleados

FROM TABLA

WHERE salario > 1000

GROUP BY departamento;

# ¿Qué significa normalización de datos y por qué es importante?✔️

Normalizar una base de datos significa estructurar de forma lógica y correcta todas las tablas para que no haya errores lógicos al momento de hacer consultas o cualquier otro fin. Es decir, significa ordenar las tablas a nivel lógico o relacional.

Las ventajas que tiene la normalización:

 Eliminar redundancias (datos repetidos).

 Evitar inconsistencias.

 Mejorar la integridad de los datos.

 Facilitar el mantenimiento a largo plazo.

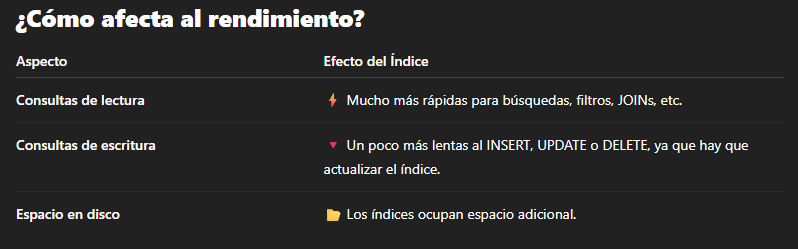
Esto se hace mediante las 3 formas normales (o a veces 5 formas normales). Se debe seguir un proceso metodológico para poder normalizar correctamente una base de datos

Cabe señalar que en ciencia de datos cuando usamos el término normalización nos referimos a un concepto no tan técnico. Nos referimos al hecho de escalar o dejar la base de datos limpia para poder hacer manipulaciones estadísticas correctamente

# ¿Qué es un índice y cómo afecta el rendimiento de las consultas?✔️

Un **índice** en una base de datos es una estructura especial (parecida a un índice en un libro) que permite **acceder a los datos de forma más rápida**, sin necesidad de revisar toda la tabla.

➡ Técnicamente, es una estructura que guarda referencias a las filas de una tabla basándose en uno o varios campos (columnas) clave.



# ¿Cuál es el orden de ejecución de una consulta SQL?✔️

El **orden de ejecución real de una consulta SQL** no siempre coincide con el orden en que escribimos los comandos



**🔵 Avanzado (21–30)❕**

# ¿Qué es una CTE (Common Table Expression) y cuándo usarla?

# ¿Cómo calcularías un promedio móvil con SQL?

# ¿Qué diferencias hay entre EXISTS, IN y NOT IN?

# ¿Cómo manejarías valores NULL en funciones agregadas?

# ¿Qué son las funciones analíticas y cómo se diferencian de las agregadas?

# ¿Cómo detectarías outliers usando SQL?

# ¿Qué es una transacción y cuáles son sus propiedades (ACID)?

# ¿Qué es una vista (view) y cuáles son sus ventajas y desventajas?

# ¿Cómo optimizarías una consulta lenta?

# ¿Cómo harías una consulta para encontrar la segunda fila más alta en una tabla de ventas?

# Enfoque en ciencia de datos

**🔶 Análisis de Datos y Negocio (1–5)**

¿Cómo identificarías a los clientes más valiosos para un negocio usando SQL o Python? ✔️  
(Pista: piensa en RFM – Recency, Frequency, Monetary)

Definiría una consulta con where para filtrar, a partir de un umbral e indicador disponible, a aquellos que más se destaquen en dicho indicador

✅ **Recency (Recencia):** Cuánto tiempo ha pasado desde la última compra → Clientes recientes son más valiosos.  
✅ **Frequency (Frecuencia):** Cuántas veces ha comprado el cliente → Más compras, más valioso.  
✅ **Monetary (Monetario):** Cuánto dinero ha gastado → Más gasto, más valioso.

Es decir, en sql haría una consulta analizando cuidadosamente las relaciones de tablas y utilizaría Python para analizar gráficamente esa consulta.

## Si las ventas de un producto disminuyen, ¿cómo investigarías las causas usando datos? ✔️

Analizaría algún, o preferentemente, algunos indicadores disponibles, priorizando la fecha y encontrar relaciones temporales con el aumento o disminución de estas. Pero también algunos otros indicadores disponibles para ver e interpretar cómo se mueve el valor de la venta en correspondencia con dicho indicador o indicadores correspondientes

¿Cómo construirías un análisis de cohortes en SQL? ¿Qué información te aporta?❕

Qué son las KPIs ✔️

En **ciencia de datos**, las **KPIs** (*Key Performance Indicators* o **Indicadores Clave de Desempeño**) son métricas cuantitativas que se utilizan para evaluar el rendimiento de modelos, procesos, proyectos o iniciativas de análisis de datos. Su función es **medir el progreso hacia objetivos específicos** y ayudar en la toma de decisiones informadas.

 **KPIs de modelos de machine learning**

* Accuracy, Precision, Recall, F1-score, AUC-ROC, RMSE, MAE, Log-loss.

 **KPIs de impacto en negocio**

* ROI, aumento de ventas, reducción de costos, mejora en conversión, disminución de churn.

 **KPIs del proceso de ciencia de datos**

* Tiempo de desarrollo, tiempo de entrenamiento/predicción, número de iteraciones, éxito en despliegue.

 **KPIs de calidad de datos**

* % de valores faltantes/duplicados, nivel de completitud, frecuencia de actualización, consistencia.

 **KPIs de uso del modelo en producción**

* Tasa de adopción, número de predicciones por día, tiempo de respuesta, tasa de errores.
* Permiten **medir objetivamente el valor** que genera un proyecto de ciencia de datos.
* Ayudan a **alinear los objetivos técnicos con los de negocio**.
* Facilitan la **comunicación con stakeholders no técnicos**.

¿Cómo calcularías la tasa de retención de usuarios mes a mes con SQL?✔️

Agruparía por mes y numero de usuarios de ese mes para observar cómo varia la cantidad de usuarios en función del mes (la gráfica la haría con Python o Power BI)

**🔷 Series Temporales y Fechas (6–10)**

¿Cómo agruparías datos por semana, mes o trimestre en SQL? ¿Y en Pandas?

¿Cómo calcularías el crecimiento porcentual mes a mes de una métrica?

¿Qué es un “lag” en series de tiempo y cómo lo aplicarías en una base de datos?

¿Cómo identificarías una tendencia estacional en una serie temporal?

¿Qué harías si te encuentras con timestamps en diferentes zonas horarias?

**🔸 Calidad de Datos y Preprocesamiento (11–13)**

¿Cómo detectarías outliers en una variable numérica usando SQL o Python?

¿Qué pasos seguirías si una tabla tiene muchas columnas con más del 50% de valores nulos?

¿Cómo tratarías datos categóricos con errores tipográficos o formatos inconsistentes (como “Male”, “M”, “male”)?

**🔺 Aplicación Práctica y Pensamiento Analítico (14–15)**

Te entregan un dataset de usuarios y eventos. ¿Cómo calcularías la conversión desde el registro hasta una compra?

Tienes un dataset con 100 columnas. ¿Cómo identificarías las variables más relevantes para predecir una salida (target)?