### Consultas (PostgreSQL)

**Crea las siguientes consultas**

**1.- Lista de todos los pedidos junto con nombre del cliente y el producto en un rango de fechas.**

Para crear una consulta que liste todos los pedidos junto con el nombre del cliente y el nombre del producto en un rango de fechas, necesitarás asegurarte de que las tablas relevantes tengan las relaciones necesarias. Supongamos que tienes las siguientes tablas:

clientes (id\_cliente, nombre\_cliente)

productos (id\_producto, nombre\_producto)

pedidos (id\_pedido, id\_cliente, id\_producto, fecha\_pedido)

A continuación, te muestro una consulta SQL que realiza lo que solicitas:

sql

SELECT

p.id\_pedido,

c.nombre\_cliente,

pr.nombre\_producto,

p.fecha\_pedido

FROM

pedidos p

JOIN

clientes c ON p.id\_cliente = c.id\_cliente

JOIN

productos pr ON p.id\_producto = pr.id\_producto

WHERE

p.fecha\_pedido BETWEEN '2023-01-01' AND '2023-12-31' -- Rango de fechas que quieras filtrar

ORDER BY

p.fecha\_pedido;

Desglose de la consulta:

SELECT : Aquí especificamos las columnas que queremos mostrar: el ID del pedido, el nombre del cliente, el nombre del producto y la fecha del pedido.

FROM : Usamos la tabla de pedidos como tabla principal.

JOIN : Unimos la tabla clientes y productos con las tablas correspondientes utilizando sus IDs. Esto permite acceder a los nombres del cliente y del producto.

WHERE : Filtramos los resultados para que solo se muestren los pedidos que se hicieron en un rango específico de fechas. Debes ajustar las fechas según lo que necesites.

ORDER BY : Opcionalmente, ordenamos los resultados por la fecha de pedido para facilitar la lectura.

**2. Calcula el total de ventas por cliente.**

Para calcular el total de ventas por cliente en PostgreSQL, primero necesitas asegurarte de que tienes las tablas adecuadas y que estas tienen la información necesaria. Supongamos que tienes las siguientes tablas:

clientes (id\_cliente, nombre\_cliente)

pedidos (id\_pedido, id\_cliente, total\_venta)

En este escenario, total\_venta es el monto total de cada pedido, y cada pedido está asociado a un cliente a través de id\_cliente.

Consulta SQL

Calcula el total de ventas por cliente:

sql

SELECT

c.id\_cliente,

c.nombre\_cliente,

SUM(p.total\_venta) AS total\_ventas

FROM

clientes c

LEFT JOIN

pedidos p ON c.id\_cliente = p.id\_cliente

GROUP BY

c.id\_cliente, c.nombre\_cliente

ORDER BY

total\_ventas DESC; -- Opcional: ordenar por el total de ventas

Desglose de la consulta:

SELECT : Seleccionamos el ID del cliente, su nombre y la suma total de las ventas.

FROM : Usamos la tabla clientes como la principal.

LEFT JOIN : Hacemos una unión con la tabla pedidos para obtener todos los pedidos asociados a cada cliente.

SUM : Calculamos la suma total de total\_venta por cliente.

GROUP BY : Agrupamos los resultados por el ID y el nombre del cliente para que la suma funcione correctamente.

ORDER BY : Ordenamos los resultados por el monto total de ventas en orden descendente, lo que te permitirá ver primero al cliente con mayor venta.

**3. Encuentra a los 3 mejores clientes, que tengan el mayor gasto.**

Para encontrar a los 3 mejores clientes que tengan el mayor gasto en PostgreSQL, puedes usar una consulta que calcule el total de gastos por cliente y luego limite el resultado a los 3 mejores. Usando la misma estructura de tablas que mencionamos antes:

clientes (id\_cliente, nombre\_cliente)

pedidos (id\_pedido, id\_cliente, total\_venta)

Aquí tienes la consulta SQL:

sql

SELECT

c.id\_cliente,

c.nombre\_cliente,

SUM(p.total\_venta) AS total\_ventas

FROM

clientes c

JOIN

pedidos p ON c.id\_cliente = p.id\_cliente

GROUP BY

c.id\_cliente, c.nombre\_cliente

ORDER BY

total\_ventas DESC

LIMIT 3; -- Limita el resultado a los 3 mejores clientes

**Desglose de la consulta:**

SELECT : Seleccionamos el ID del cliente, su nombre y la suma total de las ventas (total\_venta) para cada cliente.

FROM : Usamos la tabla clientes como la principal.

JOIN : Realizamos una unión con la tabla pedidos para obtener todos los pedidos asociados a cada cliente.

SUM : Calculamos la suma total de total\_venta por cliente.

GROUP BY : Agrupamos los resultados por el ID y el nombre del cliente.

ORDER BY : Ordenamos los resultados por el total de ventas en orden descendente para que los clientes con mayor gasto aparezcan primero.

LIMIT 3 : Limita el resultado a los 3 clientes que tienen el mayor gasto.

**4.Veriﬁca que los índices de tus tablas estén siendo utilizados.**

Para verificar que los índices de tus tablas están siendo utilizados en PostgreSQL, puedes utilizar la vista pg\_stat\_user\_indexes y la función pg\_stat\_get\_index\_usage.

Aquí te dejo algunas consultas que te pueden ayudar:

**1. Consulta para ver el uso de índices:**

sql

SELECT

schemaname,

relname AS table\_name,

indexrelname AS index\_name,

idx\_scan AS times\_used

FROM

pg\_stat\_user\_indexes

WHERE

idx\_scan > 0

ORDER BY

times\_used DESC;

Esta consulta te mostrará los índices que han sido utilizados, junto con el número de veces que han sido escaneados.

**2. Consulta para ver índices no utilizados:**

sql

SELECT

schemaname,

relname AS table\_name,

indexrelname AS index\_name

FROM

pg\_stat\_user\_indexes

WHERE

idx\_scan = 0

ORDER BY

table\_name;

Esta consulta te mostrará los índices que no han sido utilizados en absoluto.

**3. Consulta para obtener información detallada sobre los índices:**

sql

SELECT

t.relname AS table\_name,

i.relname AS index\_name,

pg\_catalog.pg\_get\_indexdef(i.oid) AS index\_definition,

s.idx\_scan AS times\_used

FROM

pg\_catalog.pg\_index x

JOIN

pg\_catalog.pg\_class t ON t.oid = x.indrelid

JOIN

pg\_catalog.pg\_class i ON i.oid = x.indexrelid

JOIN

pg\_stat\_user\_indexes s ON s.indexrelname = i.relname

WHERE

t.relkind = 'r' -- solo tablas

ORDER BY

times\_used DESC;

**5. Muestra los nombres de los clientes que no han realizado ningún pedido.**

# Consulta utilizando un LEFT JOIN entre las tablas de clientes y pedidos. Al hacer un LEFT JOIN, obtendrás todos los clientes, incluidos aquellos que no tienen correspondencia en la tabla de pedidos. Luego puedes filtrar aquellos registros donde no hay pedidos.

# Suponiendo que tus tablas son las siguientes:

# clientes (id\_cliente, nombre\_cliente)

# pedidos (id\_pedido, id\_cliente, total\_venta)

# Aquí está la consulta SQL que puedes usar:

# sql

# SELECT

# c.nombre\_cliente

# FROM

# clientes c

# LEFT JOIN

# pedidos p ON c.id\_cliente = p.id\_cliente

# WHERE

# p.id\_pedido IS NULL; -- Filtra para obtener solo los clientes sin pedidos

# Desglose de la consulta:

# SELECT : Seleccionamos el nombre del cliente (nombre\_cliente) de la tabla clientes.

# FROM : Usamos la tabla clientes como la tabla principal.

# LEFT JOIN : Hacemos un LEFT JOIN con la tabla pedidos sobre la columna id\_cliente. Esto asegura que todos los clientes se muestren, incluso aquellos que no tienen pedidos.

# WHERE : Filtramos los resultados para incluir solo aquellos donde p.id\_pedido IS NULL, es decir, clientes que no tienen ningún pedido registrado.

# Resultado

# Esta consulta devolverá una lista de nombres de clientes que no han realizado ningún pedido en la base de datos