

PRÁCTICA 1

Sistemas de Gestión de Bases de Datos



19 OCTUBRE

PRÁCTICA 1 – EJERCICIO 11

Mario Ventura Burgos 43223476-J

Grado en Ingeniería Informática (GIN 3)

CURSO 2023-2024

1. CÓDIGO Y RESULTADO

Abriremos la herramienta de Query Tool en pgAdmin (aunque también se puede hacer en sql shell) y añadimos “EXPLAIN” al inicio de la consulta 5 del examen cuya solución se encuentra en el archivo mvb135_6.sql y obtenemos lo siguiente:

INPUT:

```
SELECT VEN.codi, COUNT(*) AS NUM_COD  
FROM Venda VEN  
GROUP BY VEN.codi  
HAVING COUNT(*) > 1;
```

OUTPUT:

HashAggregate (cost=22.45..32.83 rows=277 width=22)

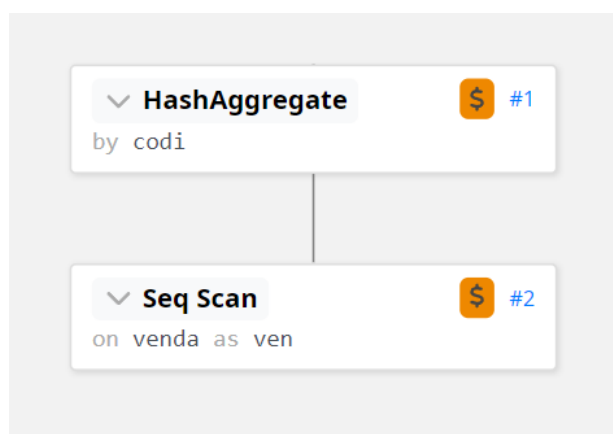
Group Key: codi

Filter: (count(*) > 1)

-> Seq Scan on venda ven (cost=0.00..18.30 rows=830 width=14)

2. ANÁLISIS DEL QUERY PLAN

Al igual que en el caso anterior, el plan de ejecución está organizado en una estructura jerárquica que se muestra mediante tabulaciones en el output. En este caso, el output obtenido es notablemente más corto que el resto de outputs obtenidos hasta hora. Aun así, al igual que hemos hecho en los casos anteriores, representaremos en forma de árbol el output obtenido por simple que sea:



Como se puede apreciar, el árbol que representa el plan de ejecución cuenta con tan solo 2 nodos, representados por pgAdmin (o, en su defecto, sql shell) mediante las 4 líneas obtenidas como output. Analicemos el resultado nodo por nodo:

1. **HASH AGGREGATE** → Las líneas 1, 2 y 3 del plan de ejecución están representadas en el primer nodo de este árbol. Analicemos línea a línea el plan de ejecución:
 - 1.1 La primera línea del plan de ejecución indica que se hace una operación de agregación de tipo “Hash Aggregate”, que se utiliza para hacer agregación y agrupación de datos. En este caso, se hace con la finalidad de contar sobre la tabla Venda (VEN), donde se busca, además, agrupar todas las filas que tienen el mismo valor en la columna codi.
Esto tiene un coste estimado de **22.45..32.83**, y se espera una cantidad de **277** filas de anchura **22** bytes.
 - 1.2 La segunda línea hace referencia al criterio de agrupación ya descrito. Tal y como hemos mencionado, “Hash Aggregate” se utiliza con la finalidad de agregar y agrupar, pero para agrupar necesitamos un criterio. Este es el criterio que describe la línea 2, que nos indica que, sobre la tabla Venda, la operación de agregación agrupará en función del valor de la columna codi (GROUP BY VEN.codi)
 - 1.3 La tercera línea especifica un filtro adicional expresado en la consulta mediante el uso de la cláusula HAVING. Es una condición sobre el agrupamiento anteriormente descrito, que indica que, de todas las filas, solo se mostrarán en el

resultado final de la consulta aquellas tales que COUNT(*) sea estrictamente mayor a 1, es decir, los grupos que tienen más de 1 fila.

2. **SEQ SCAN** → Este segundo nodo indica que antes de la agregación del nodo 1, se hará un escaneo secuencial sobre la tabla Venda (VEN) con el objetivo de buscar las filas que coincidan con las condiciones establecidas. Esto implicará un coste de **0.00..18.30** unidades de coste y del cual esperamos obtener **830** filas con una anchura de **14** bytes.

En resumen, se hace un escaneo secuencial de la tabla Venda y, de todas las filas leídas, se escogerán solo las que cumplan las restricciones establecidas (Filter) en la función de agregación de nivel superior, que, a su vez, se encargará de agrupar en función del valor de la columna codi. El resultado de la consulta será que se muestren las filas que tienen el mismo valor en la columna codi y de esas, solamente mostrar las que tienen más de una fila.