

Indices

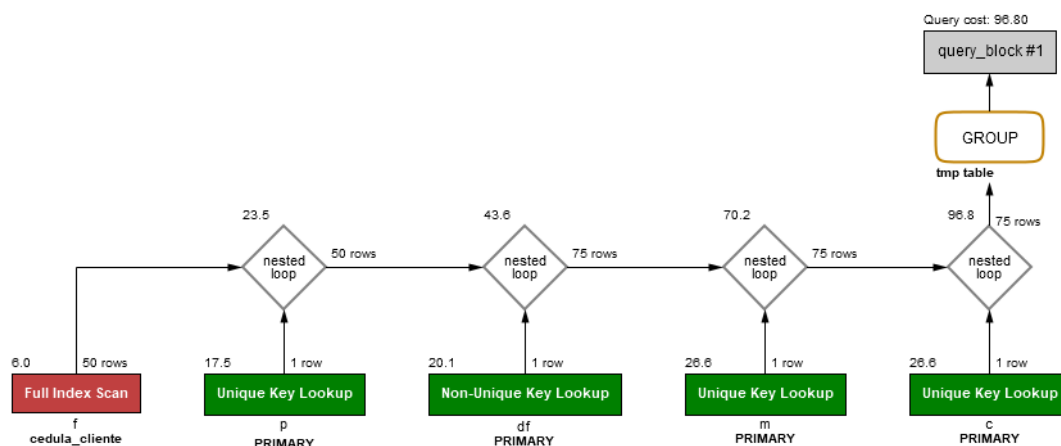
Para el análisis, tomaremos en cuenta la siguiente consulta:

- **Frecuencia de compras de clientes por categoría.** Debe presentar el nombre de la categoría, el nombre del cliente, la cantidad de veces que ha comprado productos de la categoría, y el total de dinero que ha comprado productos de la categoría.

De acuerdo con los registros de nuestra base de datos, el resultado de la consulta es lo siguiente:

nombreCategoria	cliente	cantidadCompras	totalCompras
Antialérgicos	Jose Garzon Molina	2	5
Analgésicos	Jose Garzon Molina	1	0.8999999761581421
Antiinflamatorios	Jose Garzon Molina	1	1
Antiinfecciosos	paolo hernandez rivera	1	1.8000000715255737
mucolíticos	daniel rivera rivera	2	4
Antialérgicos	mercy miranda rivera	1	1.5
Antialérgicos	patricia zambrano suarez	2	5.5
mucolíticos	patricia zambrano suarez	1	10
Antipiréticos	pepe lopez garcia	1	4
Antialérgicos	maria biassini guerra	1	4
Antialérgicos	anderson alvez cajamarca	1	4
Antiinfecciosos	anderson alvez cajamarca	2	7.800000309944153
Antialérgicos	nicolas nelson ramirez	3	13.5
Analgésicos	nicolas nelson ramirez	1	1.3499999642372131

Y tiene el siguiente costo:

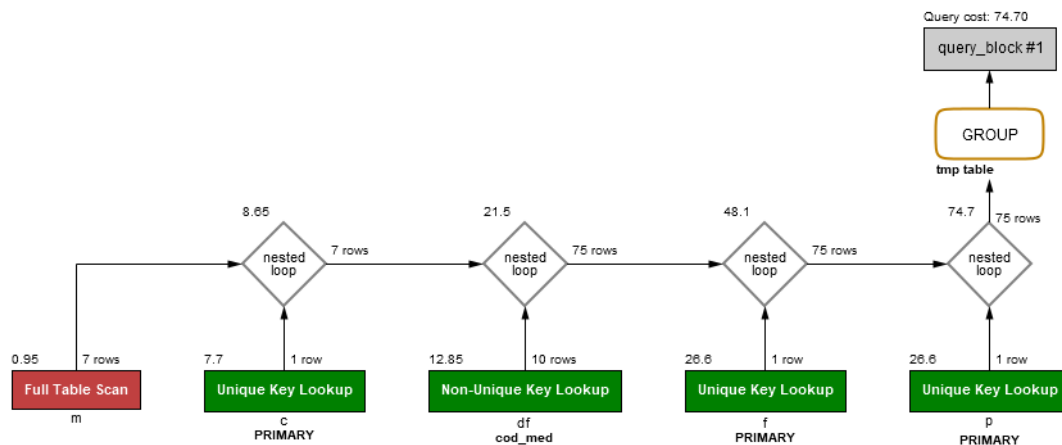


Notemos que la consulta prioriza mostrar el nombre de los clientes que adquirieron medicamentos para una categoría específica, por lo que consideramos evidente manejar índices para los nombres de los clientes. Adicionalmente, en ciertas consultas es vital conocer el nombre de las personas, tanto de trabajadores como de clientes como es en este caso, por lo que es notorio que se trata de una consulta recurrente, lo que avala aún más nuestra decisión.

Creando índices de forma independiente

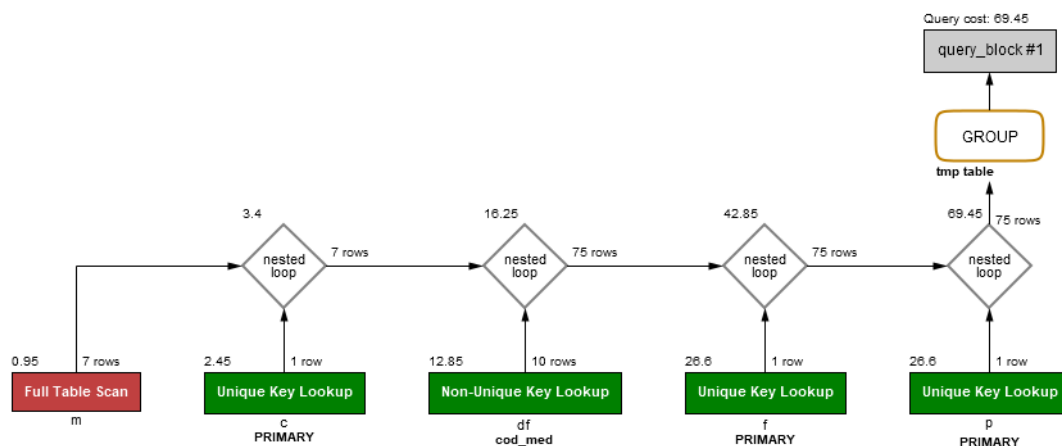
- create index idx_nombre on Persona(nombre);
- create index idx_apell_pat on Persona(apell_pat);
- create index idx_apell_mat on Persona(apell_mat);

El costo de la consulta ahora se ve reducido y se muestra en la siguiente imagen:



No obstante, notamos que con un índice combinado, el costo de la consulta se reduce aún más:

- create index idx_nombre_apellido_pat_mat on Persona(nombre, apell_pat, apell_mat);



Por lo que concluimos que el índice combinado es una mejor opción y es el que aplicaremos en nuestra base de datos.

Por otra parte, otras consultas recurrentes se basan en el nombre y categoría de la medicina, por lo que también se crearán los siguientes índices:

- create index idx_med_nombre on Medicina(nombre);
- create index idx_nomCategory on Categoria(nombre);

Consideraciones Adicionales con respecto al proyecto:

- La interfaz gráfica fue realizada con el IDE Apache Netbeans 11. Esta versión ya no cuenta con el entorno de desarrollo de JavaFX, por lo que se procedió a instalarlo manualmente.
- Se intentó realizar el proyecto en Maven, pero por problemas e inconsistencias durante la instalación se decidió finalmente no usarlo, y se trabajó con un proyecto Ant.
- Como registro del trabajo realizado se adjunta el siguiente link:
<https://github.com/tomdbelt/SBDD-Proyecto>