# Iteración 3: SuperAndes

Andrés Felipe Hernández León, Jenifer Paola Rodríguez Villamizar
Reporte técnico Iteración 3
Universidad de los Andes, Bogotá, Colombia
{af.hernandezl, jp.rodriguezv}@uniandes.edu.co
Fecha de presentación: Noviembre 4 de 2018

## Tabla de contenido

1 Introducción 2
2 Diseño de la aplicación 3
2.1 Modelo conceptual SuperAndes 3
3 Diseño físico 4
3.1 Índices en todos los requerimientos funcionales de consulta: 4
3.1.1 Requerimiento funcional de consulta 1 4
3.1.2 Requerimiento funcional de consulta 2 4
3.1.3 Requerimiento funcional de consulta 3 4
3.1.4 Requerimiento funcional de consulta 4 5
3.1.5 Requerimiento funcional de consulta 5 6
3.1.6 Requerimiento funcional de consulta 6 6
3.1.7 Requerimiento funcional de consulta 7 7
3.1.8 Requerimiento funcional de consulta 8 8
3.1.9 Requerimiento funcional de consulta 9 8
3.2 Análisis realizado para los cuatro nuevos requerimientos funcionales de consulta: 8
3.2.1 Requerimiento funcional de consulta 10 8
3.2.2 Requerimiento funcional de consulta 11 9
3.2.3 Requerimiento funcional de consulta 12 10
3.2.4 Requerimiento funcional de consulta 13 11
4 Construcción de la aplicación, pruebas y análisis 13
4.1 Diseño de la aplicación, diseño y carga de datos: 13
4.2 Ajustes al programa para los nuevos requerimientos: 13
4.3 Optimización y ejecución de las consultas: 13

# 1 Introducción

En el siguiente documento se evidenciará el desarrollo de la tercera iteración del curso. Para cada parte de la actividad, se mostrará el trabajo logrado según lo requerido en el documento del proyecto. Adicionalmente, se podrán encontrar anexos a este archivo archivos *.sql* con las sentencias de los requerimientos en texto plano y el proyecto java.

# 2 Diseño de la aplicación

## 2.1 Modelo conceptual SuperAndes

En la figura 1 se muestra el modelo conceptual completo correspondiente al caso de estudio “SuperAndes”. Para esta iteración el modelo no sufrió ningún cambio.

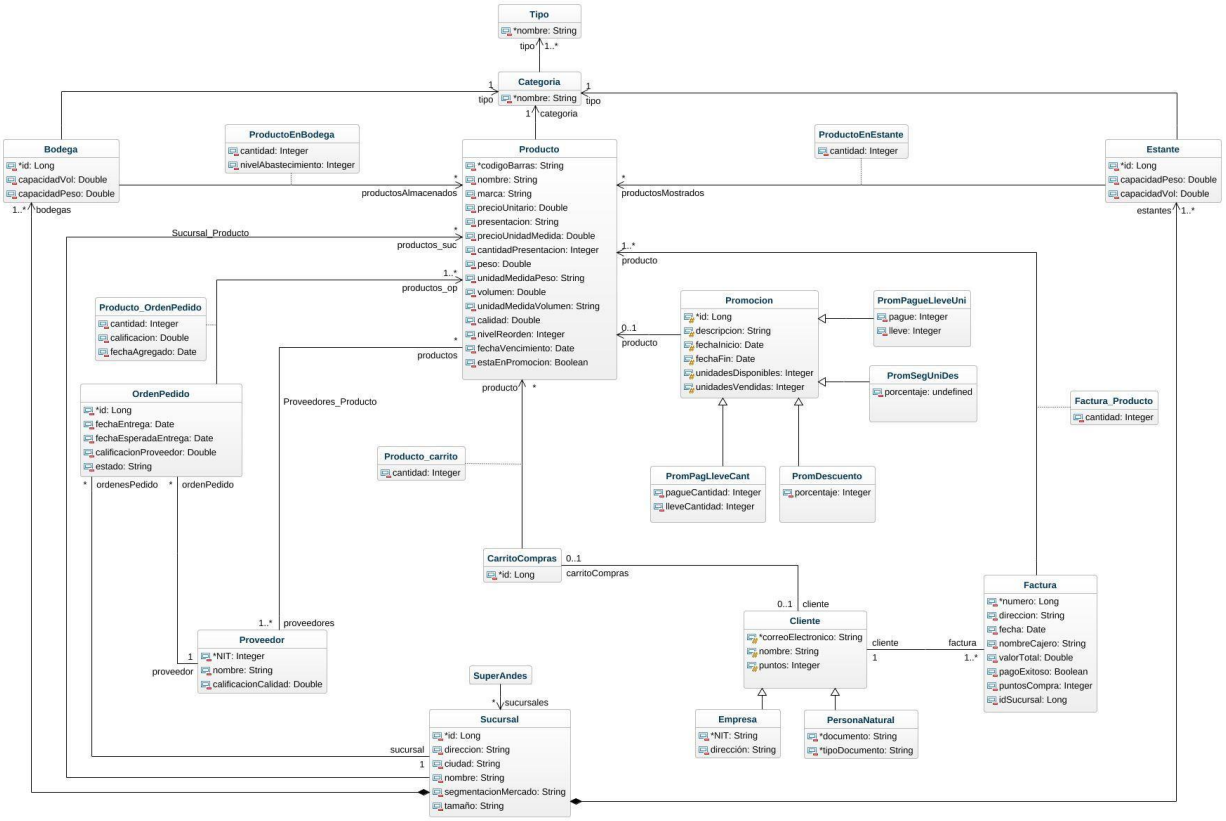


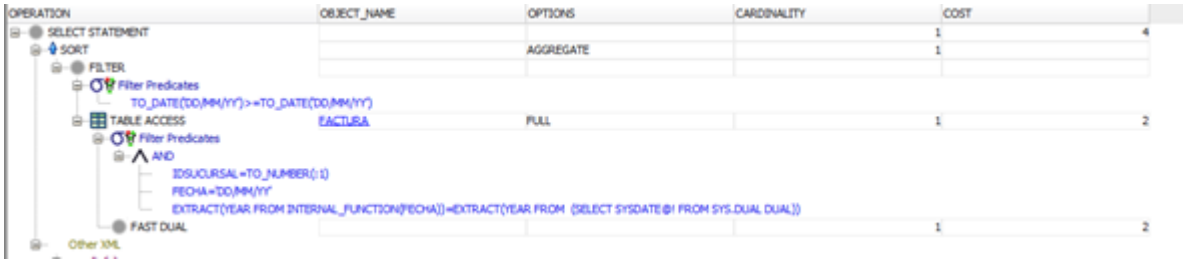

Figura 1. Modelo UML SuperAndes

# 3 Diseño físico

## 3.1 Índices en todos los requerimientos funcionales de consulta:

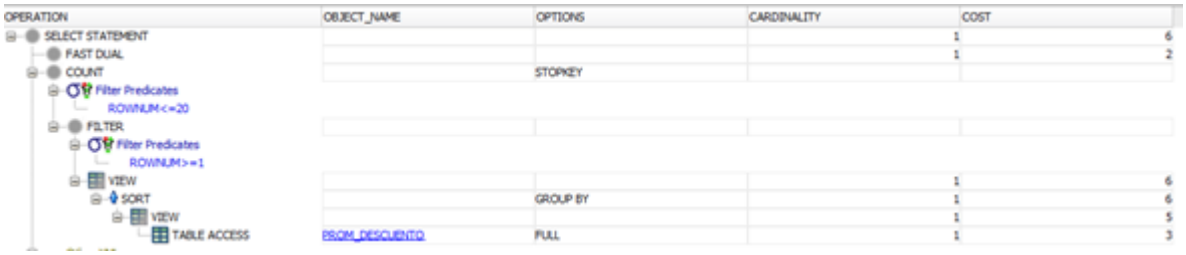
### 3.1.1 Requerimiento funcional de consulta 1

SELECT SUM (VALORTOTAL) AS VALORTOTAL FROM ( SELECT VALORTOTAL, FECHA, IDSUCURSAL FROM FACTURA WHERE EXTRACT (YEAR FROM FECHA) = EXTRACT (YEAR FROM (SELECT SYSDATE FROM DUAL)) AND FECHA BETWEEN 'DD/MM/YY' AND 'DD/MM/YY' AND IDSUCURSAL =? );




### 3.1.2 Requerimiento funcional de consulta 2

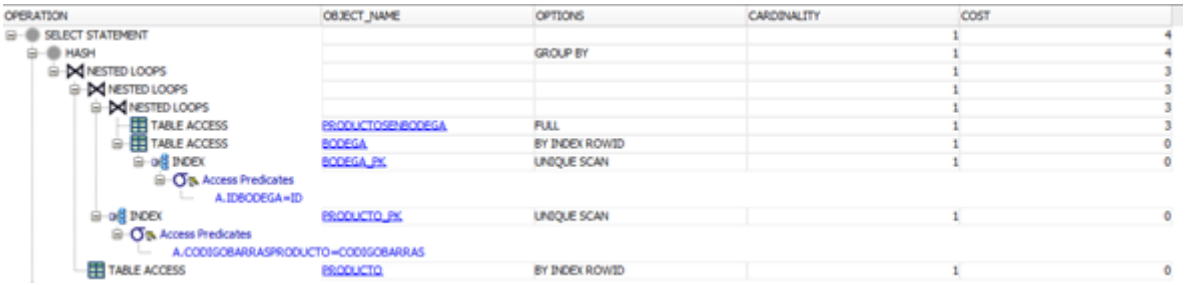
SELECT * FROM (SELECT RELACION, ID FROM (SELECT TRUNC ((SELECT SYSDATE FROM DUAL) - FECHAINICIAL) AS RELACION, ID FROM prom_descuento) GROUP BY RELACION, ID ORDER BY RELACION DESC) WHERE rownum between 1 and 20;




### 3.1.3 Requerimiento funcional de consulta 3

SELECT IDESTANTE,SUM(INDICEPESO1) AS INDICEPESO,SUM(INDICEVOL1) INDICEVOL,IDSUCURSAL FROM(SELECT IDESTANTE, PESOTOTAL/CAPACIDADPESO AS INDICEPESO1, VOLUMENTOTAL/CAPACIDADVOL AS INDICEVOL1, IDSUCURSAL FROM(SELECT A.IDESTANTE AS ESTANTE, A.CANTIDAD, A.CODIGOBARRASPRODUCTO, (a.cantidad*B.PESO) AS PESOTOTAL, (A.CANTIDAD*B.VOLUMEN ) AS VOLUMENTOTAL FROM PRODUCTOSENESTANTE A JOIN (SELECT CODIGOBARRAS, PESO, VOLUMEN FROM PRODUCTO) B ON A.CODIGOBARRASPRODUCTO = b.codigobarras) K JOIN (SELECT CAPACIDADVOL, CAPACIDADPESO, IDSUCURSAL, ID AS IDESTANTE FROM ESTANTE) L ON k.ESTANTE= l.idestante) GROUP BY IDESTANTE, IDSUCURSAL;

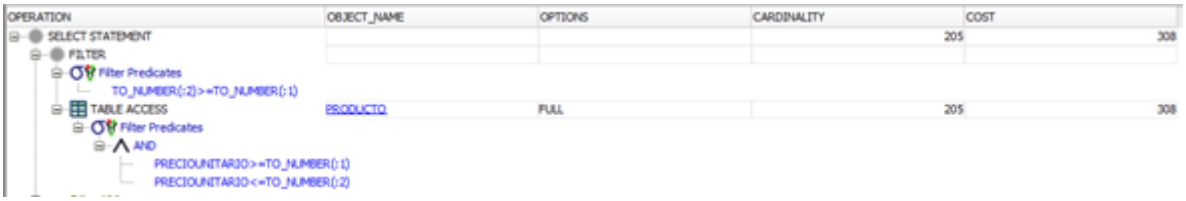
SELECT IDBODEGA, SUM(INDICEPESO1) AS INDICEPESO,SUM(INDICEVOL1) INDICEVOL,IDSUCURSAL FROM( SELECT IDBODEGA, PESOTOTAL/CAPACIDADPESO AS INDICEPESO1, VOLUMENTOTAL/CAPACIDADVOL AS INDICEVOL1, IDSUCURSAL FROM( SELECT A.IDBODEGA AS BODEGA, A.CANTIDAD, A.CODIGOBARRASPRODUCTO, (a.cantidad*B.PESO) AS PESOTOTAL, (A.CANTIDAD*B.VOLUMEN ) AS VOLUMENTOTAL FROM PRODUCTOSENBODEGA A JOIN(SELECT CODIGOBARRAS, PESO, VOLUMEN FROM PRODUCTO)B ON A.CODIGOBARRASPRODUCTO = b.codigobarras ) K JOIN (SELECT CAPACIDADVOL, CAPACIDADPESO, IDSUCURSAL,ID AS IDBODEGA FROM BODEGA) L ON k.BODEGA= l.idBODEGA) GROUP BY IDBODEGA, IDSUCURSAL;




### 3.1.4 Requerimiento funcional de consulta 4

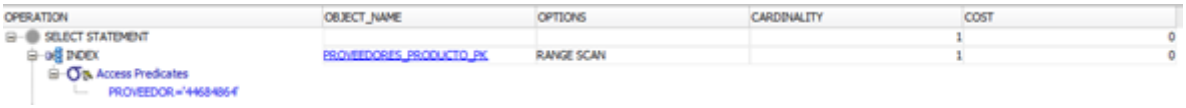
--con un precio en un rango dado.

SELECT * FROM PRODUCTOS WHERE PRECIO BETWEEN ? AND ?;



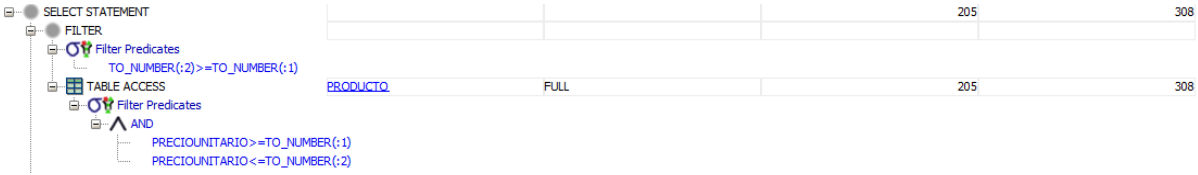

--productos ofrecidos por un proveedor.

SELECT * FROM PROVEEDORES_PRODUCTO WHERE PROVEEDOR = '?';



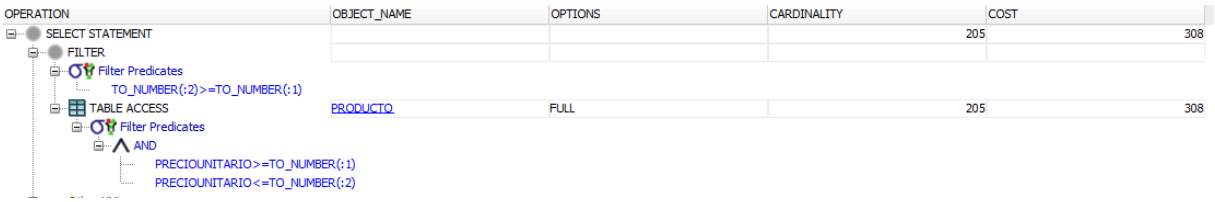

--productos con fecha de vencimiento después de cierta fecha.

SELECT * FROM PRODUCTO WHERE FECHAVENCIMIENTO > ?;



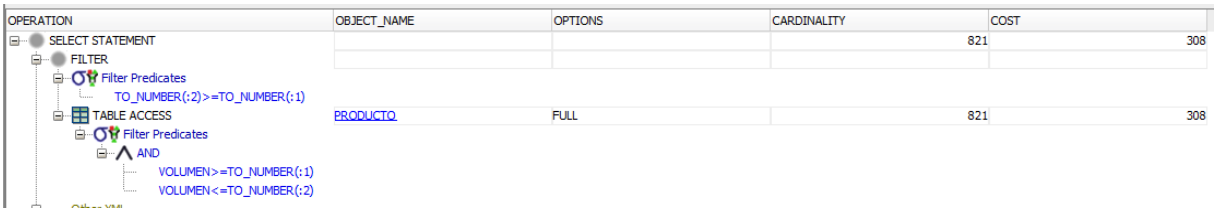

--productos con un peso en un rango dado.

SELECT * FROM PRODUCTO WHERE PRECIOUNITARIO BETWEEN ? AND ?;



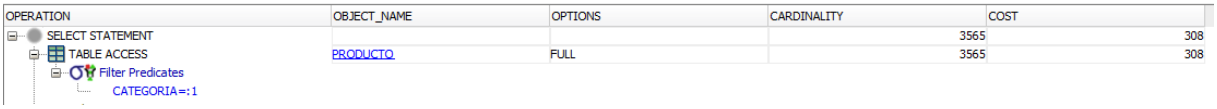

--productos con un volumen en un rango dado.

SELECT * FROM PRODUCTO WHERE VOLUMEN BETWEEN ? AND ?;



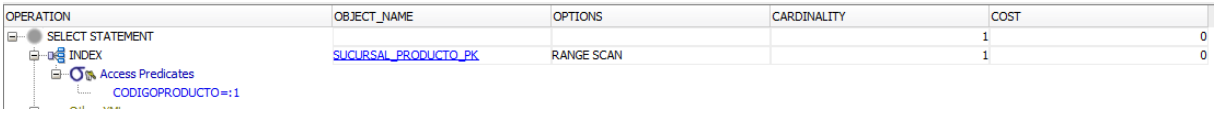

--productos con una categoria dado.

SELECT * FROM PRODUCTO WHERE CATEGORIA= ?;



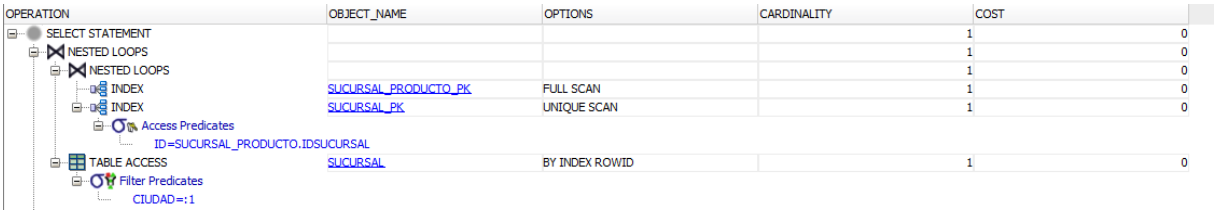

--productos ofrecidos por cierta sucursal.

SELECT * FROM SUCURSAL_PRODUCTO WHERE PRODUCTO= ?;



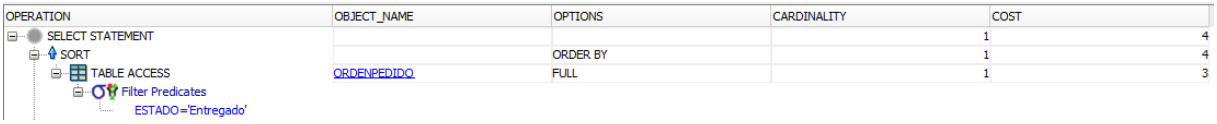

--productos ofrecidos en cierta ciudad.

SELECT ID, CIUDAD, CODIGOPRODUCTO FROM (SELECT ID, CIUDAD FROM SUCURSAL WHERE CIUDAD=?)A JOIN (SELECT * FROM SUCURSAL_PRODUCTO) B ON A.ID= B.IDSUCURSAL;




### 3.1.5 Requerimiento funcional de consulta 5

SELECT * FROM ORDENPEDIDO WHERE ESTADO = 'Entregado' ORDER BY PROVEEDOR;

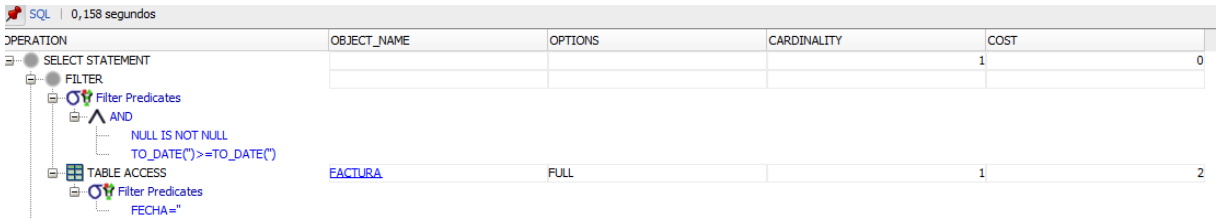



### 3.1.6 Requerimiento funcional de consulta 6

MOSTRAR LAS VENTAS DE SUPERANDES A UN USUARIO DADO, EN UN RANGO DE FECHAS INDICADO

select * from factura

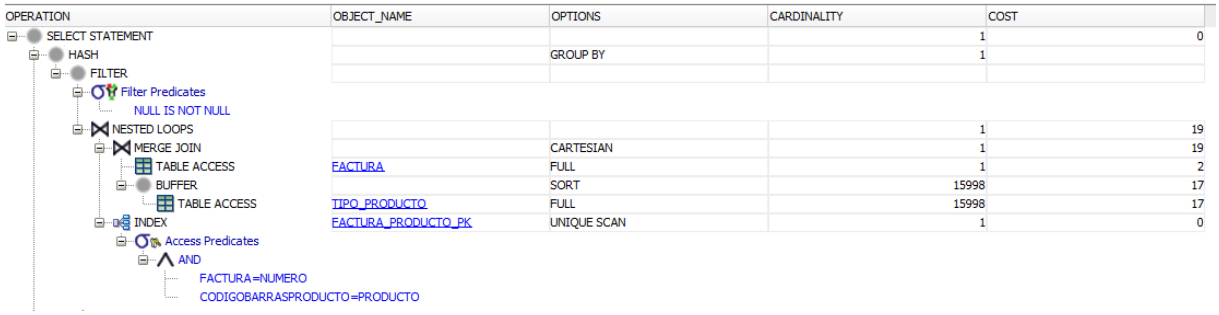
where cliente =? and fecha between ? and ?;




### 3.1.7 Requerimiento funcional de consulta 7

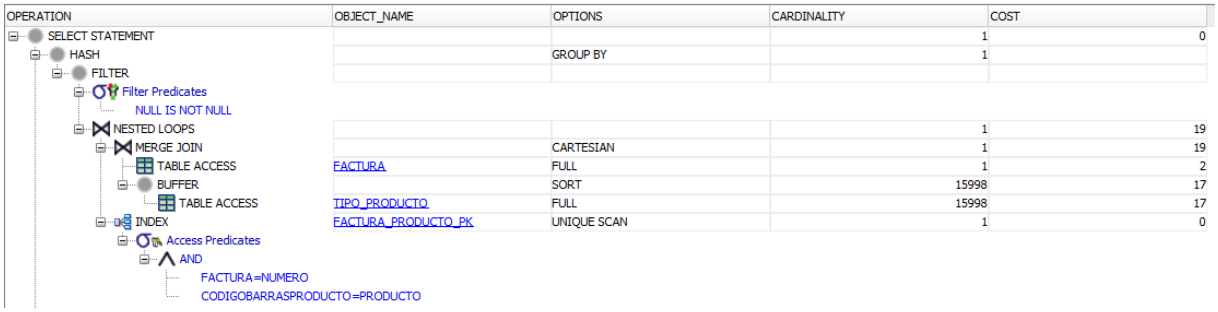
producto más solicitado

SELECT MAX(CANTIDAD), SEMANA FROM (SELECT COUNT(A.PRODUCTO) AS CANTIDAD, TO_CHAR(fecha, '??') AS SEMANA FROM ( SELECT CODIGOBARRASPRODUCTO AS PRODUCTO FROM TIPO_PRODUCTO WHERE NOMBRETIPO =") A JOIN (SELECT * FROM(SELECT FACTURA , PRODUCTO FROM FACTURA_PRODUCTO) P JOIN (SELECT NUMERO, FECHA FROM FACTURA )K ON P.FACTURA= K.NUMERO)B ON A.PRODUCTO = B.PRODUCTO GROUP BY TO_CHAR(fecha, '??') ) GROUP BY SEMANA;




producto menos solicitado

SELECT Min(CANTIDAD), SEMANA FROM (SELECT COUNT(A.PRODUCTO) AS CANTIDAD, TO_CHAR(fecha, '??') AS SEMANA FROM ( SELECT CODIGOBARRASPRODUCTO AS PRODUCTO FROM TIPO_PRODUCTO WHERE NOMBRETIPO =") A JOIN ( SELECT * FROM ( SELECT FACTURA , PRODUCTO FROM FACTURA_PRODUCTO)P JOIN (SELECT NUMERO, FECHA FROM FACTURA )K ON P.FACTURA= K.NUMERO)B ON A.PRODUCTO = B.PRODUCTO GROUP BY TO_CHAR(fecha, '??')) GROUP BY SEMANA;



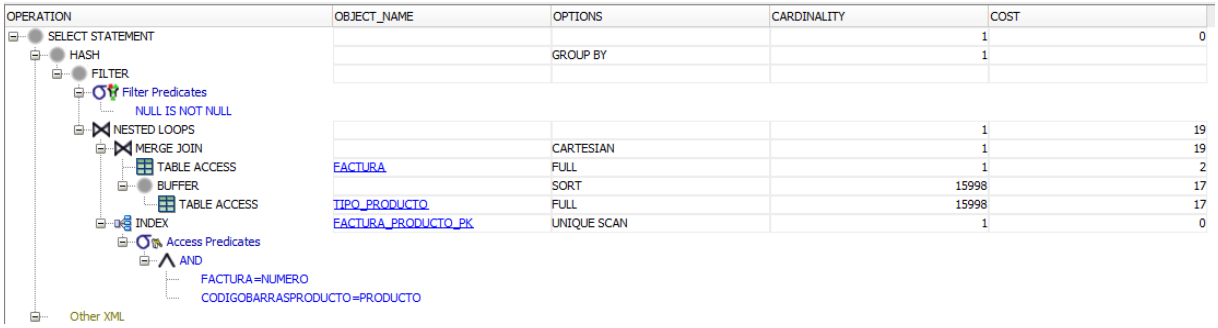

mayores ingresos

SELECT MAX(CANTIDAD) as valorTotal, SEMANA FROM (SELECT COUNT(valortotal) AS cantidad, TO_CHAR(fecha, '??') AS SEMAN FROM

( SELECT CODIGOBARRASPRODUCTO AS PRODUCTO FROM TIPO_PRODUCTO

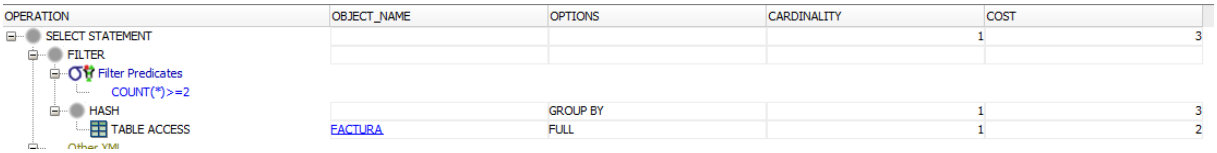
WHERE NOMBRETIPO =") A JOIN (SELECT * FROM(SELECT FACTURA , PRODUCTO FROM FACTURA_PRODUCTO)P JOIN (SELECT NUMERO, FECHA,valortotal FROM FACTURA )K ON P.FACTURA= K.NUMERO)B ON A.PRODUCTO = B.PRODUCTO GROUP BY TO_CHAR(fecha, '??'))

GROUP BY SEMANA;




### 3.1.8 Requerimiento funcional de consulta 8

SELECT MESES , COUNT(ID_FACTURA), CLIENTE FROM(SELECT FECHA, EXTRACT(MONTH FROM FECHA) AS MESES, NUMERO AS ID_FACTURA, CLIENTE FROM FACTURA) GROUP BY MESES, CLIENTE, ID_FACTURA HAVING COUNT (ID_FACTURA) >=2;




### 3.1.9 Requerimiento funcional de consulta 9

## 3.2 Análisis realizado para los cuatro nuevos requerimientos funcionales de consulta:

### 3.2.1 Requerimiento funcional de consulta 10

select *

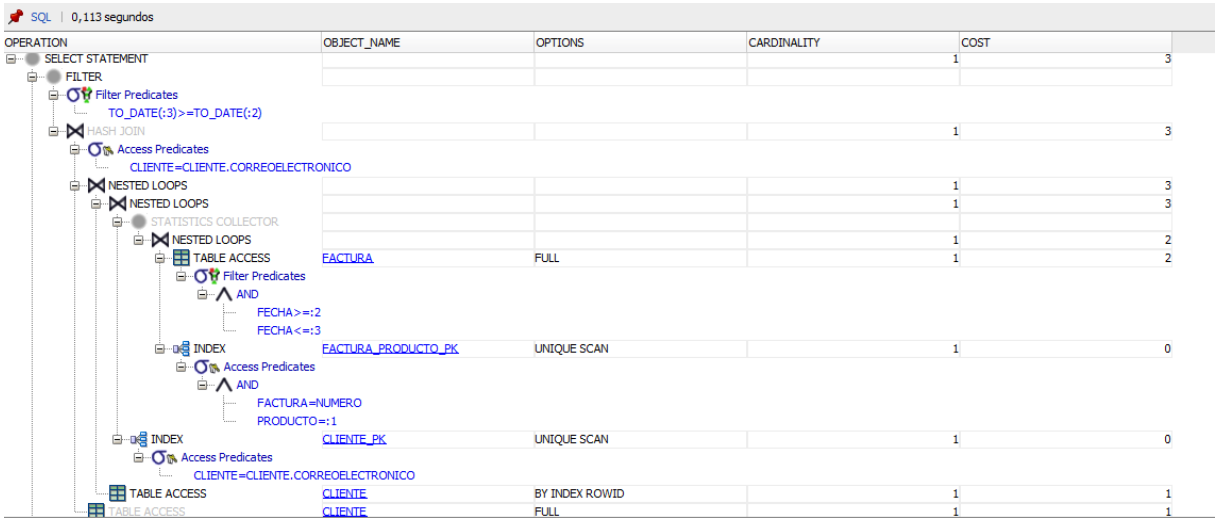
from( select factura from factura_producto

where producto = ? ) a join (select *

from (( select numero , cliente from factura

where fecha BETWEEN ? AND ?) c join(select * from cliente)d on c.cliente= d.correoelectronico )) b on a.factura=b.numero;

Sin indices

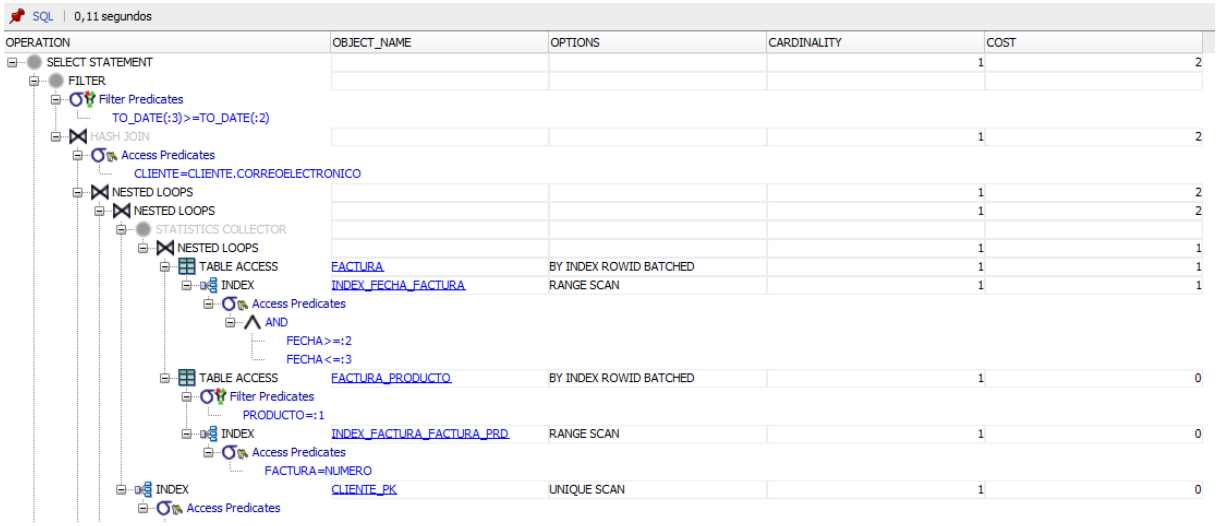


| OPERATION | OBJECT_NAME | OPTIONS | CARDINALITY | COST |
|---|---|---|---|---|
| SELECT STATEMENT | | | 1 | 3 |
| FILTER | | | | |
| Filter Predicates | | | | |
| TO_DATE(:3)>=TO_DATE(:2) | | | | |
| HASH JOIN | | | 1 | 3 |
| Access Predicates | | | | |
| CLIENTE=CLIENTE.CORREOELECTRONICO | | | | |
| NESTED LOOPS | | | 1 | 3 |
| NESTED LOOPS | | | 1 | 3 |
| STATISTICS COLLECTOR | | | | |
| NESTED LOOPS | | | 1 | 2 |
| TABLE ACCESS | FACTURA | FULL | 1 | 2 |
| Filter Predicates | | | | |
| AND | | | | |
| FECHA>=:2 | | | | |
| FECHA<=:3 | | | | |
| INDEX | FACTURA_PRODUCTO_PK | UNIQUE SCAN | 1 | 0 |
| Access Predicates | | | | |
| AND | | | | |
| FACTURA=NUMERO | | | | |
| PRODUCTO=:1 | | | | |
| INDEX | CLIENTE_PK | UNIQUE SCAN | 1 | 0 |
| Access Predicates | | | | |
| CLIENTE=CLIENTE.CORREOELECTRONICO | | | | |
| TABLE ACCESS | CLIENTE | BY INDEX ROWID | 1 | 1 |
| TABLE ACCESS | CLIENTE | FULL | 1 | 1 |

Con  indices sobre fecha en factura y factura en factura producto

CREATE INDEX index_fecha_factura ON factura(fecha) ;

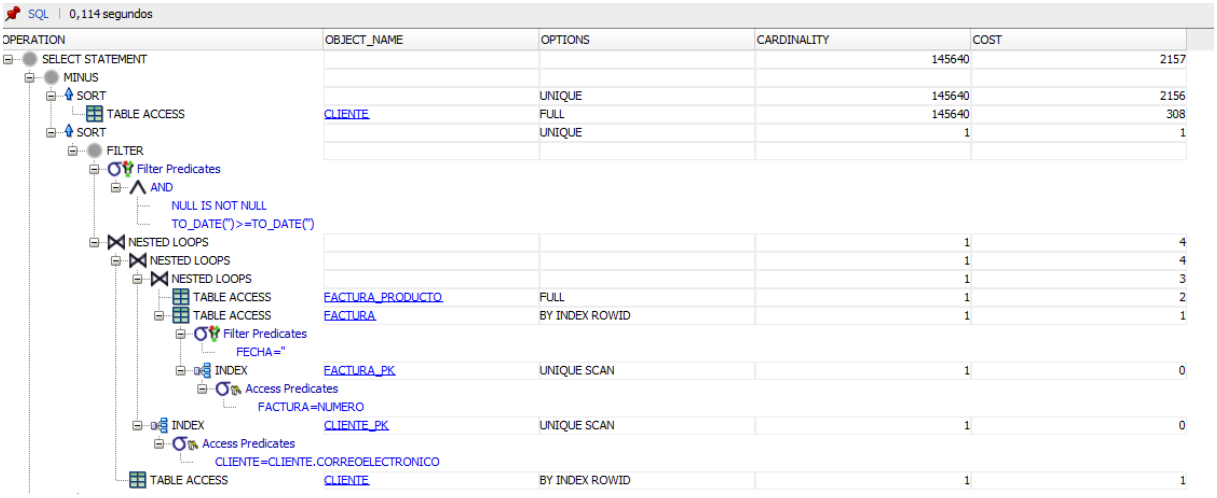
CREATE INDEX index_fACTURA_FACTURA_PRD ON factura_producto(factura) ;



| OPERATION | OBJECT_NAME | OPTIONS | CARDINALITY | COST |
|---|---|---|---|---|
| SELECT STATEMENT | | | 1 | 2 |
| FILTER | | | | |
| Filter Predicates | | | | |
| TO_DATE(:3)>=TO_DATE(:2) | | | | |
| HASH JOIN | | | 1 | 2 |
| Access Predicates | | | | |
| CLIENTE=CLIENTE.CORREOELECTRONICO | | | | |
| NESTED LOOPS | | | 1 | 2 |
| NESTED LOOPS | | | 1 | 2 |
| STATISTICS COLLECTOR | | | | |
| NESTED LOOPS | | | 1 | 1 |
| TABLE ACCESS | FACTURA | BY INDEX ROWID BATCHED | 1 | 1 |
| INDEX | INDEX_FECHA_FACTURA | RANGE SCAN | 1 | 1 |
| Access Predicates | | | | |
| AND | | | | |
| FECHA>=:2 | | | | |
| FECHA<=:3 | | | | |
| TABLE ACCESS | FACTURA_PRODUCTO | BY INDEX ROWID BATCHED | 1 | 0 |
| Filter Predicates | | | | |
| PRODUCTO=:1 | | | | |
| INDEX | INDEX_FACTURA_FACTURA_PRD | RANGE SCAN | 1 | 0 |
| Access Predicates | | | | |
| FACTURA=NUMERO | | | | |
| INDEX | CLIENTE_PK | UNIQUE SCAN | 1 | 0 |
| Access Predicates | | | | |

### 3.2.2 Requerimiento funcional de consulta 11

SELECT * FROM (select * from cliente) minus ( select correoelectronico, nombre, puntos,empresa, documentopn from( select factura from factura_producto where producto = '' ) a join (select * from (( select numero , cliente from factura where fecha BETWEEN '' AND '')  c join(select * from cliente)d on c.cliente= d.correoelectronico )) b on a.factura=b.numero);

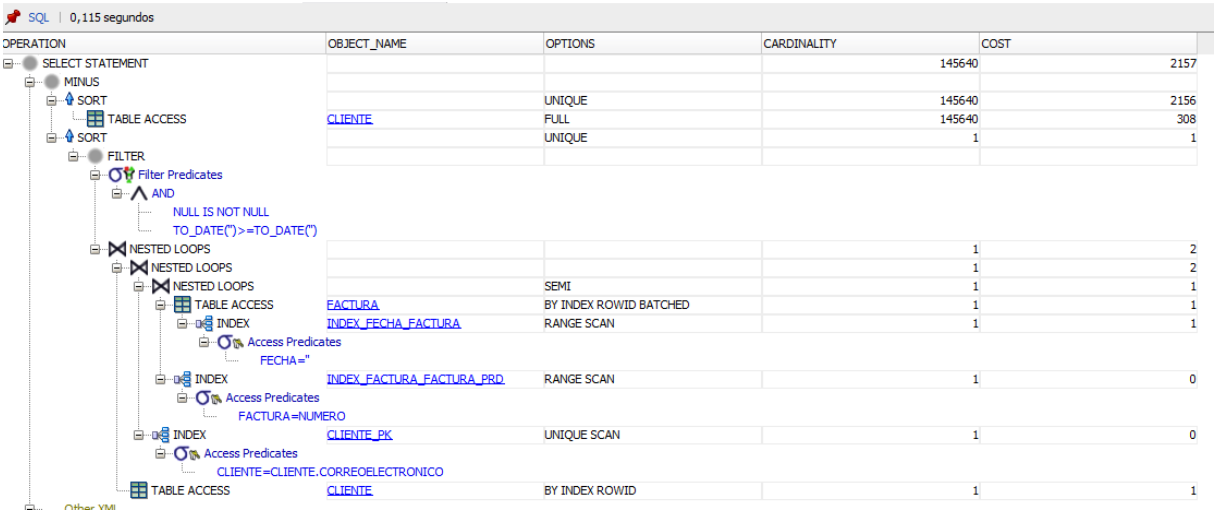

SQL | 0,114 segundos

| OPERATION | OBJECT_NAME | OPTIONS | CARDINALITY | COST |
|---|---|---|---|---|
| SELECT STATEMENT | | | 145640 | 2157 |
| MINUS | | | | |
| SORT | | UNIQUE | 145640 | 2156 |
| TABLE ACCESS | CLIENTE | FULL | 145640 | 308 |
| SORT | | UNIQUE | 1 | 1 |
| FILTER | | | | |
| Filter Predicates | | | | |
| AND | | | | |
| NULL IS NOT NULL | | | | |
| TO_DATE(')>=TO_DATE(') | | | | |
| NESTED LOOPS | | | 1 | 4 |
| NESTED LOOPS | | | 1 | 4 |
| NESTED LOOPS | | | 1 | 3 |
| TABLE ACCESS | FACTURA_PRODUCTO | FULL | 1 | 2 |
| TABLE ACCESS | FACTURA | BY INDEX ROWID | 1 | 1 |
| Filter Predicates | | | | |
| FECHA=' | | | | |
| INDEX | FACTURA_PK | UNIQUE SCAN | 1 | 0 |
| Access Predicates | | | | |
| FACTURA=NUMERO | | | | |
| INDEX | CLIENTE_PK | UNIQUE SCAN | 1 | 0 |
| Access Predicates | | | | |
| CLIENTE=CLIENTE.CORREOELECTRONICO | | | | |
| TABLE ACCESS | CLIENTE | BY INDEX ROWID | 1 | 1 |

Con  indices sobre fecha en factura y factura en factura producto

CREATE INDEX index_fecha_factura ON factura(fecha) ;

CREATE INDEX index_fACTURA_FACTURA_PRD ON factura_producto(factura) ;


SQL | 0,115 segundos

| OPERATION | OBJECT_NAME | OPTIONS | CARDINALITY | COST |
|---|---|---|---|---|
| SELECT STATEMENT | | | 145640 | 2157 |
| MINUS | | | | |
| SORT | | UNIQUE | 145640 | 2156 |
| TABLE ACCESS | CLIENTE | FULL | 145640 | 308 |
| SORT | | UNIQUE | 1 | 1 |
| FILTER | | | | |
| Filter Predicates | | | | |
| AND | | | | |
| NULL IS NOT NULL | | | | |
| TO_DATE(')>=TO_DATE(') | | | | |
| NESTED LOOPS | | | 1 | 2 |
| NESTED LOOPS | | | 1 | 2 |
| NESTED LOOPS | | SEMI | 1 | 1 |
| TABLE ACCESS | FACTURA | BY INDEX ROWID BATCHED | 1 | 1 |
| INDEX | INDEX_FECHA_FACTURA | RANGE SCAN | 1 | 1 |
| Access Predicates | | | | |
| FECHA=' | | | | |
| INDEX | INDEX_FACTURA_FACTURA_PRD | RANGE SCAN | 1 | 0 |
| Access Predicates | | | | |
| FACTURA=NUMERO | | | | |
| INDEX | CLIENTE_PK | UNIQUE SCAN | 1 | 0 |
| Access Predicates | | | | |
| CLIENTE=CLIENTE.CORREOELECTRONICO | | | | |
| TABLE ACCESS | CLIENTE | BY INDEX ROWID | 1 | 1 |

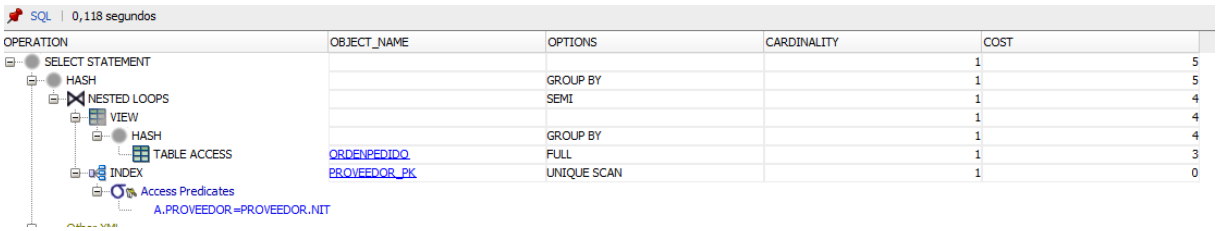
### 3.2.3 Requerimiento funcional de consulta 12

SELECT MAX(NUMERO_PEDIDOS) , MIN(NUMERO_PEDIDOS), SEMANA

FROM ((SELECT COUNT(PROVEEDOR) AS NUMERO_PEDIDOS, proveedor, TO_CHAR(fechaesperadaentrega, 'WW')AS SEMANA FROM ORDENPEDIDO

GROUP BY PROVEEDOR, TO_CHAR(fechaesperadaentrega, 'WW')) A join (SELECT * FROM PROVEEDOR) B ON A.PROVEEDOR= B.NIT)

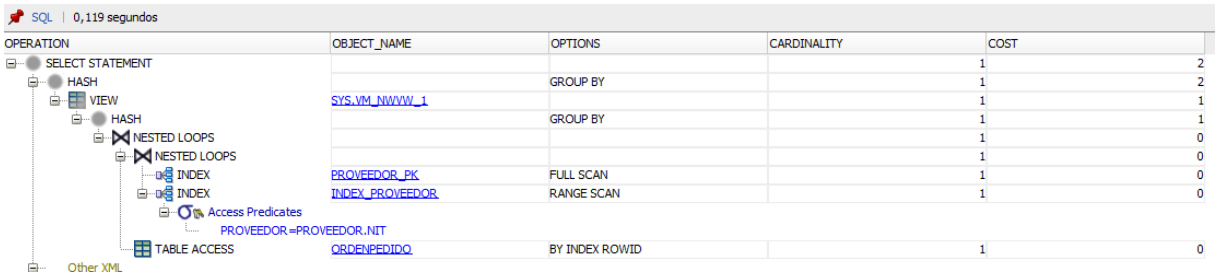
GROUP BY SEMANA ;


SQL | 0,118 segundos

| OPERATION | OBJECT_NAME | OPTIONS | CARDINALITY | COST |
|---|---|---|---|---|
| SELECT STATEMENT | | | 1 | 5 |
| HASH | | GROUP BY | 1 | 5 |
| NESTED LOOPS | | SEMI | 1 | 4 |
| VIEW | | | 1 | 4 |
| HASH | | GROUP BY | 1 | 4 |
| TABLE ACCESS | ORDENPEDIDO | FULL | 1 | 3 |
| INDEX | PROVEEDOR_PK | UNIQUE SCAN | 1 | 0 |
| Access Predicates | | | | |
| A.PROVEEDOR=PROVEEDOR.NIT | | | | |

Con indice sobre proveedor en orden pedido

CREATE INDEX index_PROVEEDOR ON ORDENPEDIDO(PROVEEDOR) ;



| OPERATION | OBJECT_NAME | OPTIONS | CARDINALITY | COST |
|---|---|---|---|---|
| SELECT STATEMENT | | | 1 | 2 |
| HASH | | GROUP BY | 1 | 2 |
| VIEW | SYS.VM_NWVW_1 | | 1 | 1 |
| HASH | | GROUP BY | 1 | 1 |
| NESTED LOOPS | | | 1 | 0 |
| NESTED LOOPS | | | 1 | 0 |
| INDEX | PROVEEDOR_PK | FULL SCAN | 1 | 0 |
| INDEX | INDEX_PROVEEDOR | RANGE SCAN | 1 | 0 |
| Access Predicates | | | | |
| PROVEEDOR=PROVEEDOR.NIT | | | | |
| TABLE ACCESS | ORDENPEDIDO | BY INDEX ROWID | 1 | 0 |
| Other XML | | | | |

SELECT MAX (CANTIDAD) as p_cantidad_maxima , MIN (CANTIDAD) as p_cantidad_minima , semana

FROM(

SELECT sum(a.CANTIDAD) AS CANTIDAD, TO_CHAR(b.fecha, 'WW') AS SEMANA FROM

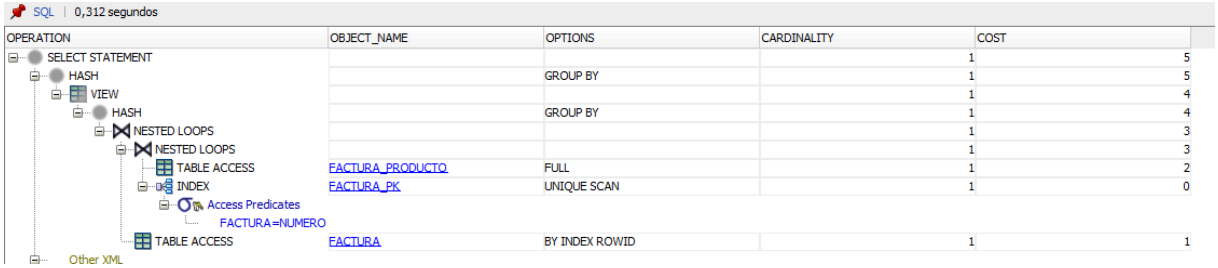
(sELECT CANTIDAD, FACTURA, PRODUCTO FROM FACTURA_PRODUCTO) A

JOIN (SELECT NUMERO, fecha FROM FACTURA) B ON a.FACTURA= B.NUMERO

GROUP BY TO_CHAR(b.FECHA, 'WW'), a.producto

)

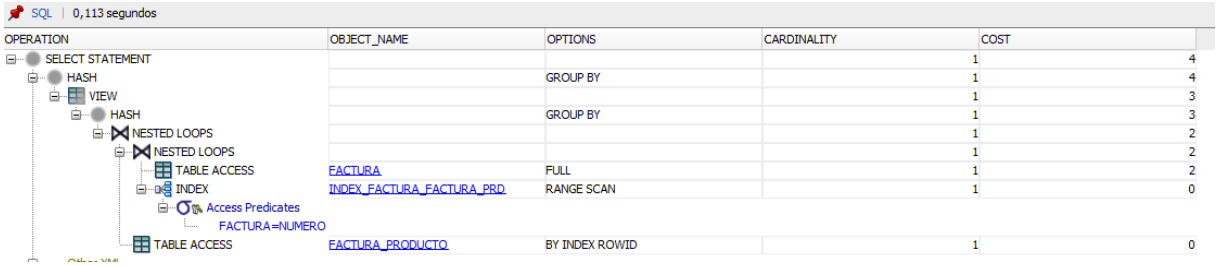
GROUP BY SEMANA;



| OPERATION | OBJECT_NAME | OPTIONS | CARDINALITY | COST |
|---|---|---|---|---|
| SELECT STATEMENT | | | 1 | 5 |
| HASH | | GROUP BY | 1 | 5 |
| VIEW | | | 1 | 4 |
| HASH | | GROUP BY | 1 | 4 |
| NESTED LOOPS | | | 1 | 3 |
| NESTED LOOPS | | | 1 | 3 |
| TABLE ACCESS | FACTURA_PRODUCTO | FULL | 1 | 2 |
| INDEX | FACTURA_PK | UNIQUE SCAN | 1 | 0 |
| Access Predicates | | | | |
| FACTURA=NUMERO | | | | |
| TABLE ACCESS | FACTURA | BY INDEX ROWID | 1 | 1 |
| Other XML | | | | |

con indices sobre factura en factura producto

CREATE INDEX index_fACTURA_FACTURA_PRD ON factura_producto(factura) ;

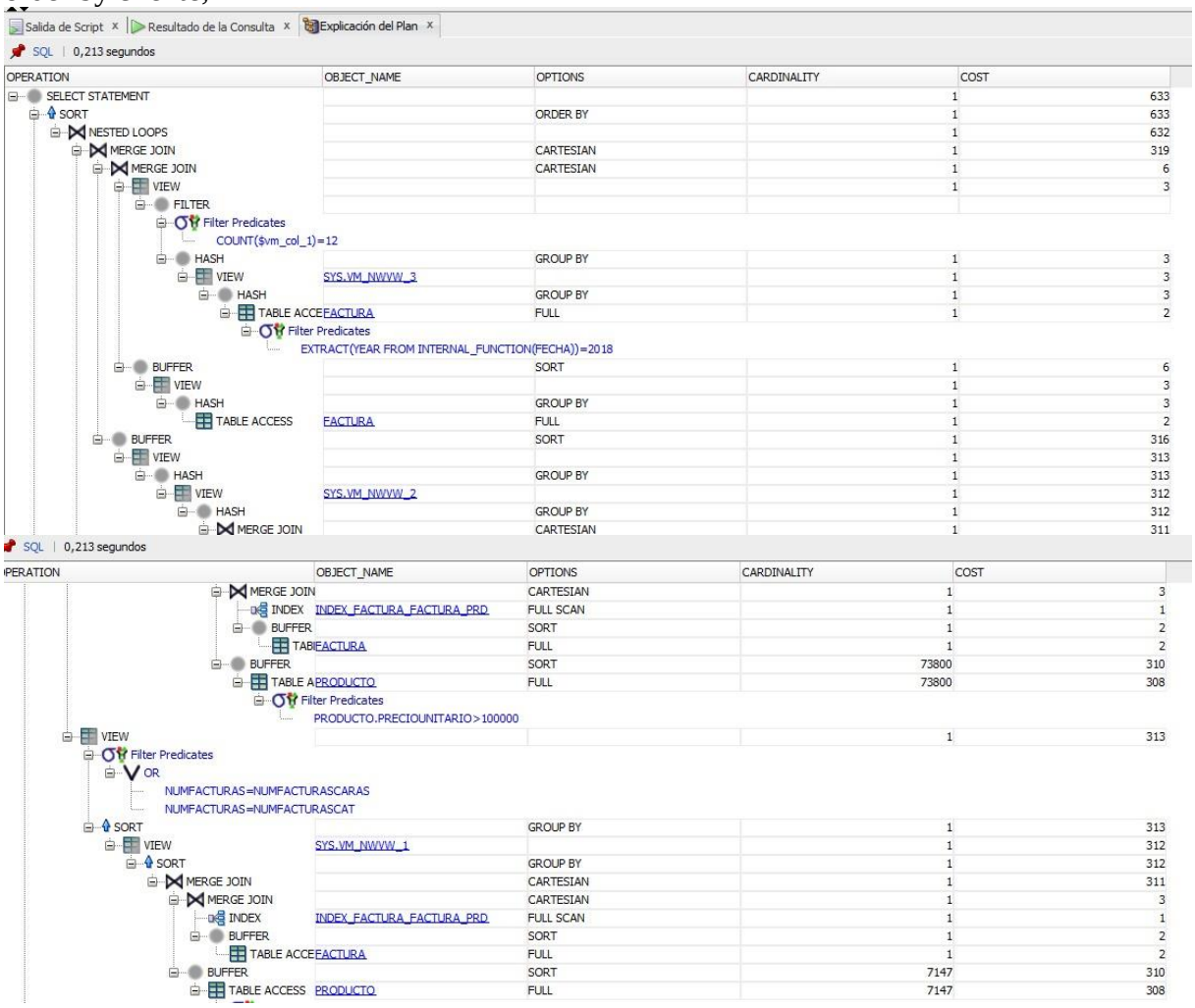


| OPERATION | OBJECT_NAME | OPTIONS | CARDINALITY | COST |
|---|---|---|---|---|
| SELECT STATEMENT | | | 1 | 4 |
| HASH | | GROUP BY | 1 | 4 |
| VIEW | | | 1 | 3 |
| HASH | | GROUP BY | 1 | 3 |
| NESTED LOOPS | | | 1 | 2 |
| NESTED LOOPS | | | 1 | 2 |
| TABLE ACCESS | FACTURA | FULL | 1 | 2 |
| INDEX | INDEX_FACTURA_FACTURA_PRD | RANGE SCAN | 1 | 0 |
| Access Predicates | | | | |
| FACTURA=NUMERO | | | | |
| TABLE ACCESS | FACTURA_PRODUCTO | BY INDEX ROWID | 1 | 0 |

### 3.2.4 Requerimiento funcional de consulta 13

--Los clientes que realizan una compra por lo menos una vez al mes.
Select z.cliente, NumMesesCompra, productoCaro, productoCategorias
from(Select cliente, NumMesesCompra
from( select cliente, count(distinct(month)) as NumMesesCompra
from( select cliente, EXTRACT(MONTH FROM fecha) as month from factura where EXTRACT(YEAR FROM FECHA) = 2018
) group by cliente

```
      ) where NumMesesCompra = 12
   )z,
   (SELECT a.cliente, productoCaro, productoCategorias
    from   ( SELECT cliente, count(numero) as numFacturas from factura group by cliente
          )a,
          (SELECT cliente, count(distinct(factura))as numFacturasCaras, nombre as
productoCaro
           FROM FACTURA, FACTURA_PRODUCTO, PRODUCTO
           WHERE PRODUCTO.PRECIOUNITARIO > 100000 group by cliente,nombre
          ),
          (SELECT cliente, count(distinct(factura))as numFacturasCat, nombre as
productoCategorias
           FROM FACTURA, FACTURA_PRODUCTO , PRODUCTO
           WHERE PRODUCTO.CATEGORIA ='Technology' OR PRODUCTO.CATEGORIA
= 'Tools' group by cliente, nombre
          )
   where numFacturas = numFacturasCaras OR numFacturas = numFacturasCat)
order by cliente;
```



# 4 Construcción de la aplicación, pruebas y análisis

## 4.1 Diseño de la aplicación, diseño y carga de datos:

Se hicieron los cambios necesarios en la creación de las tablas que se encuentra en el proyecto en la ruta "./superAndes/data/Crear y poblar tablas/esquemaSuperAndes.sql". En cuanto a la carga de datos, en principio la hicimos usando insert generados por Mockaroo y concatenados por nosotros. Luego se cargaron otras por csv en sqlLoader. Por tanto, se encuentran tanto los archivos .sql para llenar cada una de las tablas, como los .csv para llenar otras de las tablas con el orden respectivo en que deberían cargarse. Los datos fueron generados de forma uniforme (gracias a expresiones regulares generadas por nosotros) en el generador de datos Mockaroo. Finalmente, cabe resaltar que los datos ya fueron cargados a la base de datos que utiliza la aplicación.

## 4.2 Ajustes al programa para los nuevos requerimientos:

Para esta iteración, no fue necesario realizar cambios mayores en la aplicación puesto que ya habíamos modelado para la iteración dos los roles de administrador y usuarios (persona natural - empresa) y no fue necesario cambiar nada del modelo relacional ni conceptual, por lo que solamente se implementaron las cosas necesarias para las consultas (por ejemplo, en las clases SQL de los debidos resultados agregar los nuevos métodos y conectarlos con interfaz, etc).

## 4.3 Optimización y ejecución de las consultas:

Para esta sección se realizó la implementación de los índices que se puede observar en la sección 3.2 donde se explica para cada tabla como fue tomada la decisión de utilizar un índice o no. Para la ejecución de las consultas hay dos modalidades, una como administrador desde nuestra aplicación (primera opción para ingresar) o directamente corriendo el script en los archivos .sql encontrados en nuestro proyecto en la dirección "./data/Iteracion3/RFC" desde sqlDeveloper.

En lo que se refiere a los requerimientos funcionales de consulta 10 y 11, desde la aplicación se puede elegir los posibles group by u order by deseados. En sqlDeveloper, se adjuntan después del requerimiento básico las consultas para los posibles group by u order by para facilidad de la ejecución de estos.