**Iteración No.3 - SuperAndes**

Santiago Restrepo Sánchez 201714611, Ixtli Barbosa Rincón CODIGO

Grupo A-02

Sistemas Transaccionales Universidad de los Andes, Bogotá, Colombia

{s.restrepos1, [iy.barbosa} @uniandes.edu.co](mailto:jr.pacheco%7d@uniandes.edu.co)

**Fecha de presentación**: Diciembre 02 de 2018

Tabla de contenido

1. Análisis

2. Diseño y construcción de la aplicación

3. Diseño físico de la aplicación

3.1 Índices generados por Oracle

3.2 RFC 10

3.3 RFC 11

3.4 RFC 12

3.5 RFC 13

4. Diseño de datos

5. Conclusiones

**1 Análisis**

**2 Diseño y construcción de la aplicación**

**3 Diseño físico de la aplicación**

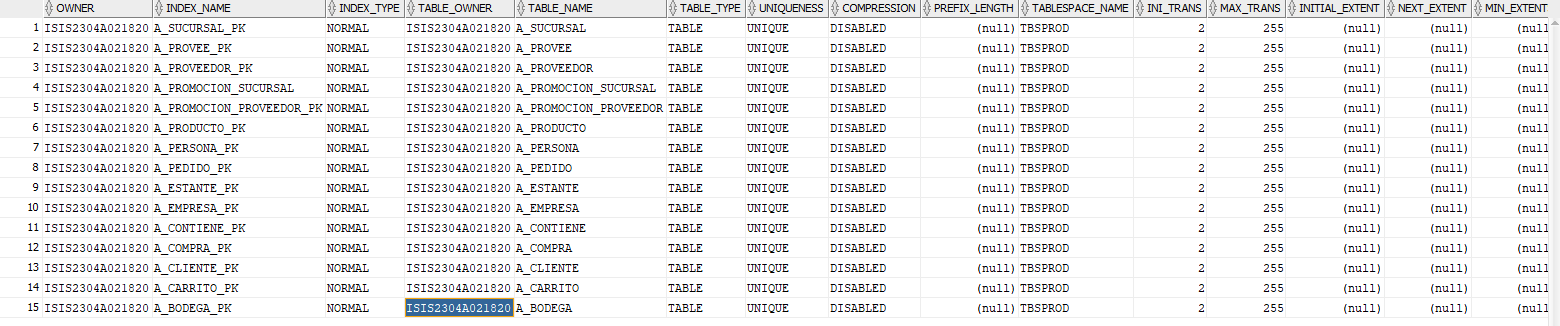
**3.1 Índices generados por Oracle**

Todos los índices encontrados son creados de forma automática por Oracle debido a que son o hacen parte de una llave primaria, esto pasa porque la llave primaria es única y la selectividad es alta. La consulta para obtener los índices es la siguiente:

SELECT \* FROM SYS.ALL\_INDEXES

WHERE OWNER = 'ISIS2304A021820';

La tabla resultante es:



Donde se pueden observar 15 índices de tipo unique sobre toda la base de datos.

**3.2 Requerimiento Funcional de Consulta 10**

Consultar consumo en SuperAndes

Requerimiento: *Se quiere conocer la información de los usuarios que realizaron al menos una compra de un determinado producto en un rango de fechas. Los resultados deben ser clasificados según un criterio deseado por quien realiza la consulta. En la clasificación debe ofrecerse la posibilidad de agrupamiento y ordenamiento de las respuestas según los intereses del usuario que consulta como, por ejemplo, por los datos del cliente, por fecha y número de unidades compradas del producto.*

**Selección de índices:** Se implementó un índice sobre FECHA llamado INDEX1, ya que este atributo muestra para un rango de fechas dado si el cliente realizo una compra. Aunque la selectividad de IDCLIENTE es más alta que FECHA unitariamente, al ser una consulta en rangos FECHA puede tener una selectividad incremental. No se crea un índice combinado debido al peso del árbol resultante, si se crea un índice (IDCLIENTE, FECHA) cada nodo del árbol B+ resultante tendrá 400.000 hijos.

**Sentencia para crear el índice**: CREATE UNIQUE INDEX INDEX1 ON A\_COMPRA (FECHA);

**Sentencia SQL Utilizada:**

SELECT \*

FROM A\_COMPRA, A\_CLIENTE

WHERE PRODUCTOCODIGO = 1

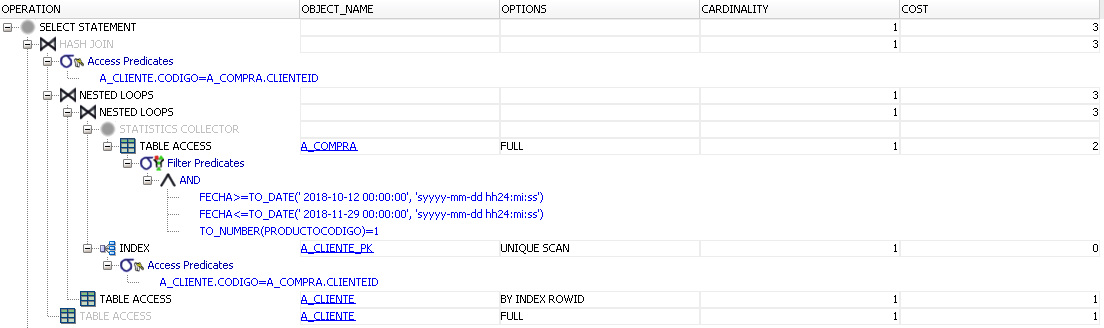
AND FECHA BETWEEN '12/10/2018'

AND '29/11/2018'

AND A\_CLIENTE.CODIGO= A\_COMPRA.CLIENTEID;

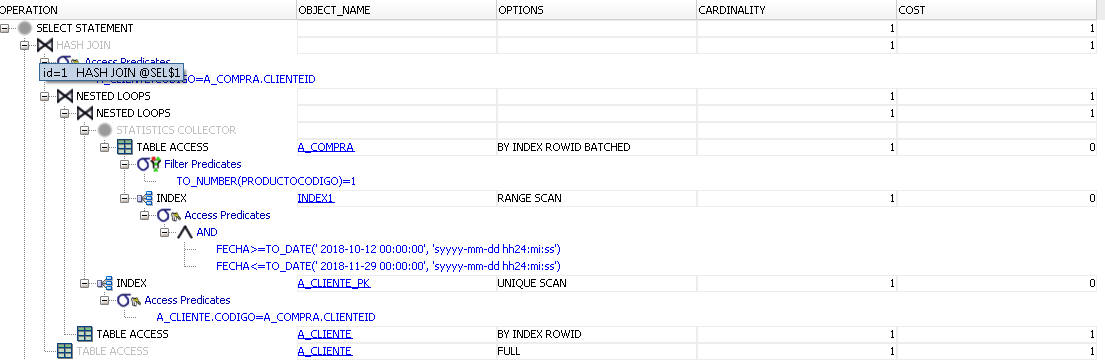
**Valores de parámetros utilizado: COMPLETAR**

**Plan de consulta sin índices y tabla de datos vacía:**



En el plan de consulta se puede observar que Oracle pasa por todas las fechas buscando si están dentro del rango, hace un nested loop join que tiene un costo de 3

**Plan de consulta con índices y tabla de datos vacía:**



En el plan de consulta se puede observar que Oracle utiliza el índice para encontrar las fechas en el rango dado, logrando así reducir el costo del nested loop join de 3 a 1.

**Plan de consulta sin índices y tabla de datos llena:**

**Plan de consulta con índices y tabla de datos llena:**

**3.3 Requerimiento Funcional de Consulta 11**

CONSULTAR CONSUMO EN SUPERANDES – RFC10-V2

Requerimiento: *Se quiere conocer la información de los usuarios que NO realizaron al menos una compra de un determinado producto en un rango de fechas. Los resultados deben ser clasificados según un criterio deseado por quien realiza la consulta. En la clasificación debe ofrecerse la posibilidad de agrupamiento y ordenamiento de las respuestas según los intereses del usuario que consulta como, por ejemplo, por los datos del cliente, por fecha y número de unidades compradas del producto.*

**Selección de índices:** El índice que se utilizó para optimizar la consulta es el mismo del requerimiento 10 debido a que la consulta se realiza sobre los mismos atributos. Las razones de la selección son las mismas.

**Sentencia SQL utilizada:**

SELECT \*

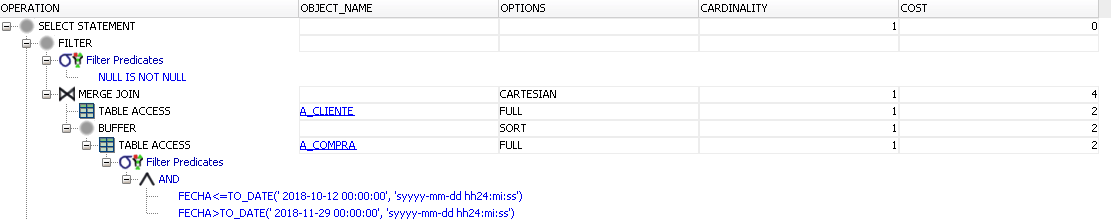
FROM A\_CLIENTE, A\_COMPRA

WHERE FECHA <= '12/10/2018' AND FECHA > '29/11/2018'

AND A\_CLIENTE.CODIGO= A\_COMPRA.CLIENTEID;

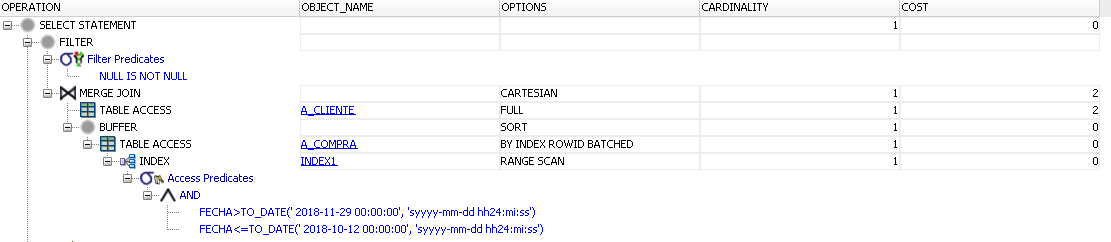
**Valores de parámetros utilizado: COMPLETAR**

**Plan de consulta sin índices y tabla de datos vacía:**



En el plan de consulta se puede apreciar que Oracle recorre todas las fechas y busca las que están dentro del rango. Realiza un merge join que tiene un costo de 4.

**Plan de consulta con índices y tabla de datos vacía:**



En el plan de consulta se observa que Oracle utiliza el índice para encontrar las tuplas que están dentro del rango. En costo del merge join baja de 4 a 2.

**Plan de consulta sin índices y tabla de datos llena:**

**Plan de consulta con índices y tabla de datos llena:**

**3.4 Requerimiento Funcional de Consulta 12**

CONSULTAR FUNCIONAMIENTO

Requerimiento: *Muestra, para cada semana del año (domingo a domingo), el producto más vendido, el producto menos vendido, los proveedores más solicitados y los proveedores menos solicitados. Las respuestas deben ser sustentadas por el detalle de las ventas y pedidos correspondientes. Esta operación es realizada el gerente general de SuperAndes.*

**Selección de índice:** Se necesitaba un índice sobre productoCodigo, pero al intentar los distintos tipos de índice (unique, no-unique, bitmap) el que nos dio una mejor complejidad en general fue el bitmap. Esto se debe a que al agrupar los productos por su código resulta un grupo de pocos únicos, en donde bitmap, este contexto es en el cual bitmap optimiza las consultas.

**Consideración**: Debido a que son distintas respuestas que pueden organizarse a criterio del usuario consideramos que debemos establecer un plan de consulta para cada una de las 4 distintas consultas.

**Sentencia SQL utilizada para el producto más vendido:**

SELECT \* FROM(

(SELECT PRODUCTOCODIGO, COUNT(PRODUCTOCODIGO)

FROM A\_COMPRA

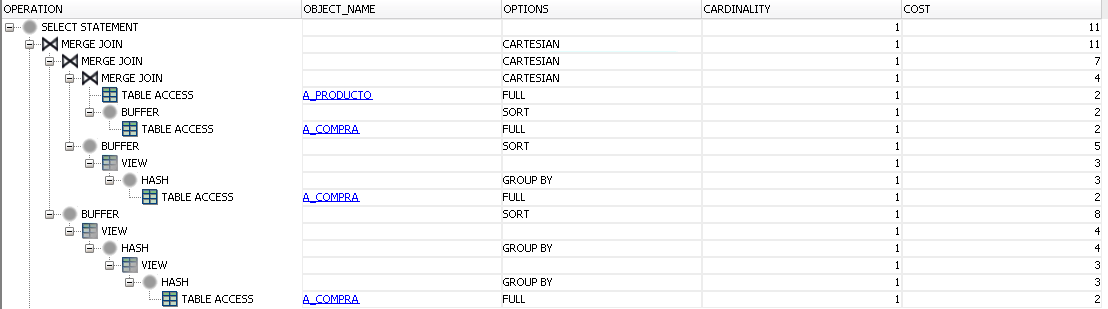
GROUP BY PRODUCTOCODIGO)

NATURAL INNER JOIN (SELECT MAX(CONT)

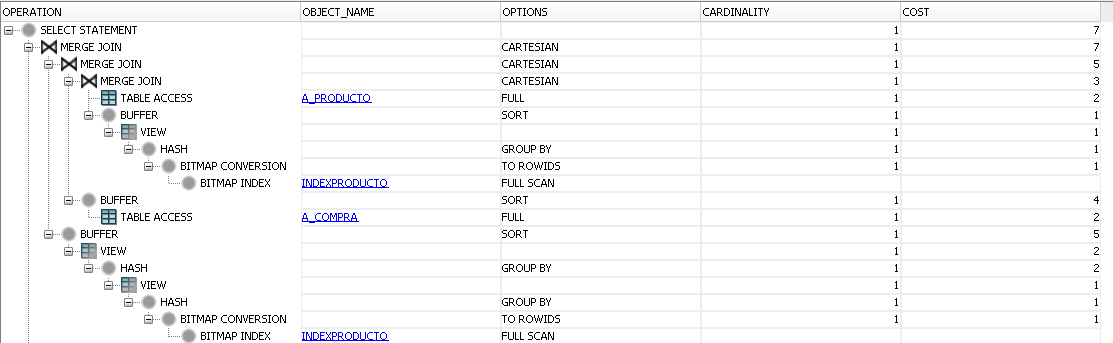
FROM (SELECT PRODUCTOCODIGO, COUNT (PRODUCTOCODIGO) AS CONT FROM A\_COMPRA GROUP BY PRODUCTOCODIGO)

GROUP BY(CONT))),A\_PRODUCTO, A\_COMPRA

**Plan de consulta sin índices y tabla de datos vacía:**



**Plan de consulta con índices y tabla de datos vacía:**



**Plan de consulta sin índices y tabla de datos llena:**

**Plan de consulta con índices y tabla de datos llena:**

**Sentencia SQL utilizada para el producto menos vendido:**

SELECT \* FROM(

(SELECT PRODUCTOCODIGO, COUNT(PRODUCTOCODIGO)

FROM A\_COMPRA

GROUP BY PRODUCTOCODIGO)

NATURAL INNER JOIN (SELECT MIN(CONT)

FROM (SELECT PRODUCTOCODIGO, COUNT (PRODUCTOCODIGO) AS CONT FROM A\_COMPRA GROUP BY PRODUCTOCODIGO)

GROUP BY(CONT))),A\_PRODUCTO, A\_COMPRA

**Plan de consulta sin índices y tabla de datos vacía:**

**Plan de consulta con índices y tabla de datos vacía:**

**Plan de consulta sin índices y tabla de datos llena:**

**Plan de consulta con índices y tabla de datos llena:**

**Sentencia SQL utilizada para los proveedores menos solicitados:**

**Plan de consulta sin índices y tabla de datos vacía:**

**Plan de consulta con índices y tabla de datos vacía:**

**Plan de consulta sin índices y tabla de datos llena:**

**Plan de consulta con índices y tabla de datos llena:**

**Sentencia SQL utilizada para los proveedores más solicitados:**

**Plan de consulta sin índices y tabla de datos vacía:**

**Plan de consulta con índices y tabla de datos vacía:**

**Plan de consulta sin índices y tabla de datos llena:**

**Plan de consulta con índices y tabla de datos llena:**

**3.5 Requerimiento Funcional de Consulta 13**

**CONSULTAR LOS BUENOS CLIENTES**

Requerimiento*: Los buenos clientes son de tres tipos: aquellos que compran en SuperAndes al menos una vez al mes, aquellos que siempre compran por lo menos un producto costoso (Entiéndase como costoso, por ejemplo, con un precio mayor a $100.000.oo) y aquellos que siempre compran productos de tecnología o herramientas. Esta consulta retorna toda la información de dichos clientes, incluyendo la que justifica su calificación como buenos clientes. Esta operación es realizada únicamente por el gerente general de SuperAndes.*

Selección del índice:

**Consideración**: Debido a que son distintas respuestas que pueden organizarse a criterio del usuario consideramos que debemos establecer un plan de consulta para cada una de las 3 distintas consultas.

**Sentencia SQL para una compra mínimo una vez al mes:**

**Plan de consulta sin índices y tabla de datos vacía:**

**Plan de consulta con índices y tabla de datos vacía:**

**Plan de consulta sin índices y tabla de datos llena:**

**Plan de consulta con índices y tabla de datos llena:**

**Sentencia SQL compra un producto con precio alto:**

SELECT \* FROM A\_CLIENTE, A\_COMPRA

WHERE CLIENTEID=CODIGO AND PRODUCTOCODIGO IN (SELECT PRODUCTOCODIGO FROM A\_COMPRA, A\_PRODUCTO WHERE PRODUCTOCODIGO=CODIGODEBARRAS AND PRECIOVENTA >100000)

**Plan de consulta sin índices y tabla de datos vacía:**



**Plan de consulta con índices y tabla de datos vacía:**

**Plan de consulta sin índices y tabla de datos llena:**

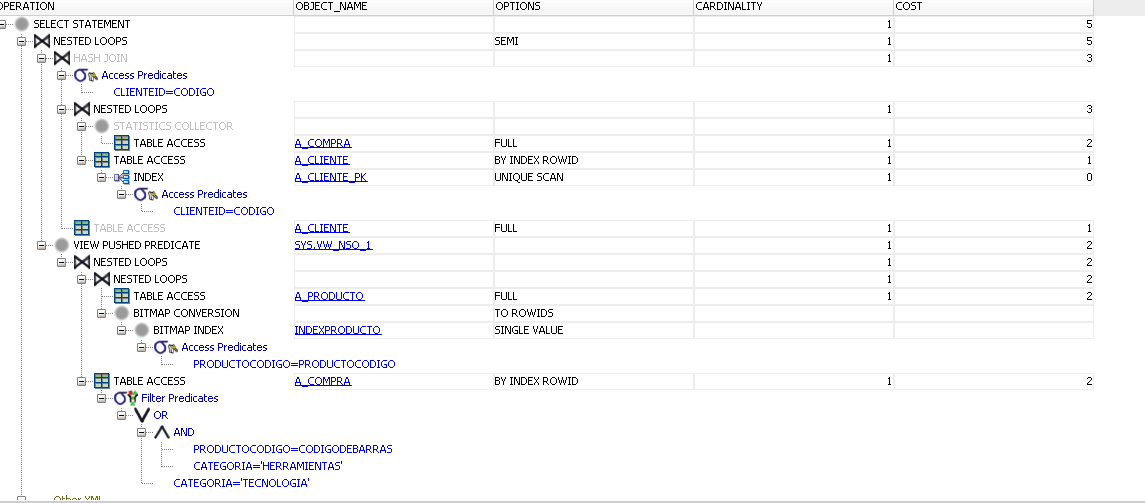
**Plan de consulta con índices y tabla de datos llena:**

**Sentencia SQL compra herramienta o tecnologia:**

SELECT \* FROM A\_CLIENTE, A\_COMPRA

WHERE CLIENTEID=CODIGO AND PRODUCTOCODIGO IN (SELECT PRODUCTOCODIGO FROM A\_COMPRA, A\_PRODUCTO WHERE PRODUCTOCODIGO=CODIGODEBARRAS AND CATEGORIA ='HERRAMIENTAS' OR CATEGORIA='TECNOLOGIA')

**Plan de consulta sin índices y tabla de datos vacía:**



**Plan de consulta con índices y tabla de datos vacía:**

**Plan de consulta sin índices y tabla de datos llena:**

**Plan de consulta con índices y tabla de datos llena:**