## 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, <u>iy.barbosa</u>} @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:

OWNER	NDEX_NAME		TABLE_OWNER	↑ TABLE_NAME	↑ TABLE_TYPE	UNIQUENESS	⊕ COMPRESSION	PREFIX_LENGTH	TABLESPACE_NAME		() MAX_TRANS	NITIAL_EXTENT	NEXT_EXTENT	MIN_EXTENT
1 ISIS2304A021820 A_S	SUCURSAL_PK	NORMAL	ISIS2304A021820	A_SUCURSAL	TABLE	UNIQUE	DISABLED	(null)	TBSPROD	2	255	(null)	(null)	(null^
2 ISIS2304A021820 A_P	PROVEE_PK	NORMAL	ISIS2304A021820	A_PROVEE	TABLE	UNIQUE	DISABLED	(null)	TBSPROD	2	255	(null)	(null)	(null
3 ISIS2304A021820 A_P	PROVEEDOR_PK	NORMAL	ISIS2304A021820	A_PROVEEDOR	TABLE	UNIQUE	DISABLED	(null)	TBSPROD	2	255	(null)	(null)	(null
4 ISIS2304A021820 A_P	PROMOCION_SUCURSAL	NORMAL	ISIS2304A021820	A_PROMOCION_SUCURSAL	TABLE	UNIQUE	DISABLED	(null)	TBSPROD	2	255	(null)	(null)	(null
5 ISIS2304A021820 A_P	PROMOCION_PROVEEDOR_PK	NORMAL	ISIS2304A021820	A_PROMOCION_PROVEEDOR	TABLE	UNIQUE	DISABLED	(null)	TBSPROD	2	255	(null)	(null)	(null
6 ISIS2304A021820 A_P	PRODUCTO_PK	NORMAL	ISIS2304A021820	A_PRODUCTO	TABLE	UNIQUE	DISABLED	(null)	TBSPROD	2	255	(null)	(null)	(null
7 ISIS2304A021820 A_P	PERSONA_PK	NORMAL	ISIS2304A021820	A_PERSONA	TABLE	UNIQUE	DISABLED	(null)	TBSPROD	2	255	(null)	(null)	(null
8 ISIS2304A021820 A_P	PEDIDO_PK	NORMAL	ISIS2304A021820	A_PEDIDO	TABLE	UNIQUE	DISABLED	(null)	TBSPROD	2	255	(null)	(null)	(null
9 ISIS2304A021820 A_E	ESTANTE_PK	NORMAL	ISIS2304A021820	A_ESTANTE	TABLE	UNIQUE	DISABLED	(null)	TBSPROD	2	255	(null)	(null)	(null
10 ISIS2304A021820 A_E	EMPRESA_PK	NORMAL	ISIS2304A021820	A_EMPRESA	TABLE	UNIQUE	DISABLED	(null)	TBSPROD	2	255	(null)	(null)	(null
11 ISIS2304A021820 A_C	CONTIENE_PK	NORMAL	ISIS2304A021820	A_CONTIENE	TABLE	UNIQUE	DISABLED	(null)	TBSPROD	2	255	(null)	(null)	(null
12 ISIS2304A021820 A_C	COMPRA_PK	NORMAL	ISIS2304A021820	A_COMPRA	TABLE	UNIQUE	DISABLED	(null)	TBSPROD	2	255	(null)	(null)	(null
13 ISIS2304A021820 A_C	CLIENTE_PK	NORMAL	ISIS2304A021820	A_CLIENTE	TABLE	UNIQUE	DISABLED	(null)	TBSPROD	2	255	(null)	(null)	(null
14 ISIS2304A021820 A_C	CARRITO_PK	NORMAL	ISIS2304A021820	A_CARRITO	TABLE	UNIQUE	DISABLED	(null)	TBSPROD	2	255	(null)	(null)	(null
15 ISIS2304A021820 A_B	BODEGA_PK	NORMAL	ISIS2304A021820	A_BODEGA	TABLE	UNIQUE	DISABLED	(null)	TBSPROD	2	255	(null)	(null)	(null

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 = I

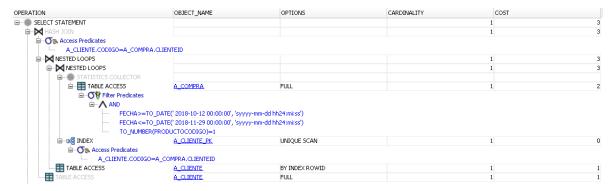
AND FECHA BETWEEN '12/10/2018'

AND '29/11/2018'

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

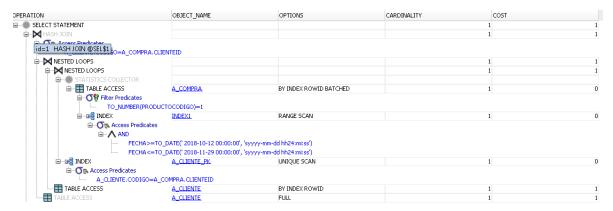
Valores de parámetros utilizado: COMPLETAR

### 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:

PERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST	
SELECT STATEMENT				1	7
		CARTESIAN		1	7
		CARTESIAN		1	5
		CARTESIAN		1	3
TABLE ACCESS	A_PRODUCTO	FULL		1	2
		SORT		1	1
i⊒ VIEW				1	1
⊟ MASH		GROUP BY		1	1
		TO ROWIDS		1	1
BITMAP INDEX	INDEXPRODUCTO	FULL SCAN			
BUFFER BUFFER		SORT		1	4
TABLE ACCESS	A_COMPRA	FULL		1	7
BUFFER		SORT		1	5
Ė ₩ VIEW				1	2
⊟ MASH		GROUP BY		1	2
i VIEW				1	1
⊟ HASH		GROUP BY		1	1
		TO ROWIDS		1	1
BITMAP INDEX	INDEXPRODUCTO	FULL SCAN			

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: