

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](mailto:iy.barbosa@uniandes.edu.co)} @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	INDEX_NAME	INDEX_TYPE	TABLE_OWNER	TABLE_NAME	TABLE_TYPE	UNIQUENESS	COMPRESSION	PREFIX_LENGTH	TABLESPACE_NAME	INL_TRANS	MAX_TRANS	INITIAL_EXTENT	NEXT_EXTENT	MIN_EXTENT
1	ISIS2304A021820	A_SUCURSAL_FK	NORMAL	ISIS2304A021820	A_SUCURSAL	TABLE	UNIQUE	DISABLED	(null)	TBSPROD	2	255	(null)	(null)	(null)
2	ISIS2304A021820	A_PROVEE_FK	NORMAL	ISIS2304A021820	A_PROVEE	TABLE	UNIQUE	DISABLED	(null)	TBSPROD	2	255	(null)	(null)	(null)
3	ISIS2304A021820	A_PROVEEDOR_FK	NORMAL	ISIS2304A021820	A_PROVEEDOR	TABLE	UNIQUE	DISABLED	(null)	TBSPROD	2	255	(null)	(null)	(null)
4	ISIS2304A021820	A_PROMOCION_SUCURSAL	NORMAL	ISIS2304A021820	A_PROMOCION_SUCURSAL	TABLE	UNIQUE	DISABLED	(null)	TBSPROD	2	255	(null)	(null)	(null)
5	ISIS2304A021820	A_PROMOCION_PROVEEDOR_FK	NORMAL	ISIS2304A021820	A_PROMOCION_PROVEEDOR	TABLE	UNIQUE	DISABLED	(null)	TBSPROD	2	255	(null)	(null)	(null)
6	ISIS2304A021820	A_PRODUCTO_FK	NORMAL	ISIS2304A021820	A_PRODUCTO	TABLE	UNIQUE	DISABLED	(null)	TBSPROD	2	255	(null)	(null)	(null)
7	ISIS2304A021820	A_PERSONA_FK	NORMAL	ISIS2304A021820	A_PERSONA	TABLE	UNIQUE	DISABLED	(null)	TBSPROD	2	255	(null)	(null)	(null)
8	ISIS2304A021820	A_PEDIDO_FK	NORMAL	ISIS2304A021820	A_PEDIDO	TABLE	UNIQUE	DISABLED	(null)	TBSPROD	2	255	(null)	(null)	(null)
9	ISIS2304A021820	A_ESTANTE_FK	NORMAL	ISIS2304A021820	A_ESTANTE	TABLE	UNIQUE	DISABLED	(null)	TBSPROD	2	255	(null)	(null)	(null)
10	ISIS2304A021820	A_EMPRESA_FK	NORMAL	ISIS2304A021820	A_EMPRESA	TABLE	UNIQUE	DISABLED	(null)	TBSPROD	2	255	(null)	(null)	(null)
11	ISIS2304A021820	A_CONTIENE_FK	NORMAL	ISIS2304A021820	A_CONTIENE	TABLE	UNIQUE	DISABLED	(null)	TBSPROD	2	255	(null)	(null)	(null)
12	ISIS2304A021820	A_COMPRA_FK	NORMAL	ISIS2304A021820	A_COMPRA	TABLE	UNIQUE	DISABLED	(null)	TBSPROD	2	255	(null)	(null)	(null)
13	ISIS2304A021820	A_CLIENTE_FK	NORMAL	ISIS2304A021820	A_CLIENTE	TABLE	UNIQUE	DISABLED	(null)	TBSPROD	2	255	(null)	(null)	(null)
14	ISIS2304A021820	A_CARRITO_FK	NORMAL	ISIS2304A021820	A_CARRITO	TABLE	UNIQUE	DISABLED	(null)	TBSPROD	2	255	(null)	(null)	(null)
15	ISIS2304A021820	A_BODEGA_FK	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 realizó 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:

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT				3
HASH JOIN			1	3
Access Predicates			1	3
A_CLIENTE.CODIGO=A_COMPRA.CLIENTEID				
NESTED LOOPS			1	3
NESTED LOOPS			1	3
STATISTICS COLLECTOR			1	3
TABLE ACCESS	A_COMPRA	FULL	1	2
Filter Predicates				
AND				
FECHA>=TO_DATE(' 2018-10-12 00:00:00', 'yyyy-mm-dd hh24:mi:ss')				
FECHA<=TO_DATE(' 2018-11-29 00:00:00', 'yyyy-mm-dd hh24:mi:ss')				
TO_NUMBER(PRODUCTOCODIGO)=1				
INDEX	A_CLIENTE_PK	UNIQUE SCAN	1	0
Access Predicates				
A_CLIENTE.CODIGO=A_COMPRA.CLIENTEID				
TABLE ACCESS	A_CLIENTE	BY INDEX ROWID	1	1
TABLE ACCESS	A_CLIENTE	FULL	1	1

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:

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			1	1
HASH JOIN			1	1
Access Predicates				
id=1 HASH JOIN @SEL\$1				
A_CLIENTE.CODIGO=A_COMPRA.CLIENTEID				
NESTED LOOPS			1	1
NESTED LOOPS			1	1
STATISTICS COLLECTOR			1	1
TABLE ACCESS	A_COMPRA	BY INDEX ROWID BATCHED	1	0
Filter Predicates				
TO_NUMBER(PRODUCTOCODIGO)=1				
INDEX	INDEX1	RANGE SCAN	1	0
Access Predicates				
AND				
FECHA>=TO_DATE(' 2018-10-12 00:00:00', 'yyyy-mm-dd hh24:mi:ss')				
FECHA<=TO_DATE(' 2018-11-29 00:00:00', 'yyyy-mm-dd hh24:mi:ss')				
INDEX	A_CLIENTE_PK	UNIQUE SCAN	1	0
Access Predicates				
A_CLIENTE.CODIGO=A_COMPRA.CLIENTEID				
TABLE ACCESS	A_CLIENTE	BY INDEX ROWID	1	1
TABLE ACCESS	A_CLIENTE	FULL	1	1

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:

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT				1 0
FILTER				
Filter Predicates				
NULL IS NOT NULL				
MERGE JOIN		CARTESIAN		1 4
TABLE ACCESS	A_CLIENTE	FULL		1 2
BUFFER		SORT		1 2
TABLE ACCESS	A_COMPRA	FULL		1 2
Filter Predicates				
AND				
FECHA<=TO_DATE(' 2018-10-12 00:00:00', 'yyyy-mm-dd hh24:mi:ss')				
FECHA>TO_DATE(' 2018-11-29 00:00:00', 'yyyy-mm-dd hh24:mi:ss')				

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:

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT				1 0
FILTER				
Filter Predicates				
NULL IS NOT NULL				
MERGE JOIN		CARTESIAN		1 2
TABLE ACCESS	A_CLIENTE	FULL		1 2
BUFFER		SORT		1 0
TABLE ACCESS	A_COMPRA	BY INDEX ROWID BATCHED		1 0
INDEX	INDEX1	RANGE SCAN		1 0
Access Predicates				
AND				
FECHA>TO_DATE(' 2018-11-29 00:00:00', 'yyyy-mm-dd hh24:mi:ss')				
FECHA<=TO_DATE(' 2018-10-12 00:00:00', 'yyyy-mm-dd hh24:mi:ss')				

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:

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			1	11
MERGE JOIN		CARTESIAN	1	11
MERGE JOIN		CARTESIAN	1	11
MERGE JOIN		CARTESIAN	1	7
TABLE ACCESS	A_PRODUCTO	FULL	1	4
BUFFER		SORT	1	2
TABLE ACCESS	A_COMPRA	FULL	1	2
BUFFER		SORT	1	5
VIEW			1	3
HASH		GROUP BY	1	3
TABLE ACCESS	A_COMPRA	FULL	1	2
BUFFER		SORT	1	8
VIEW			1	4
HASH		GROUP BY	1	4
VIEW			1	3
HASH		GROUP BY	1	3
TABLE ACCESS	A_COMPRA	FULL	1	2

Plan de consulta con índices y tabla de datos vacía:

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			1	7
MERGE JOIN		CARTESIAN	1	7
MERGE JOIN		CARTESIAN	1	5
MERGE JOIN		CARTESIAN	1	3
TABLE ACCESS	A_PRODUCTO	FULL	1	2
BUFFER		SORT	1	1
VIEW			1	1
HASH		GROUP BY	1	1
BITMAP CONVERSION		TO ROWIDS	1	1
BITMAP INDEX	INDEXPRODUCTO	FULL SCAN		
BUFFER		SORT	1	4
TABLE ACCESS	A_COMPRA	FULL	1	2
BUFFER		SORT	1	5
VIEW			1	2
HASH		GROUP BY	1	2
VIEW			1	1
HASH		GROUP BY	1	1
BITMAP CONVERSION		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: