# Diseño de la aplicación

Ramón Alejandro Arias Rivera, Andrés Felipe Rojas Pinzón Universidad de los Andes, Bogotá, Colombia {ra.ariasr, af.rojasp}@uniandes.edu.co
Fecha de presentación: Diciembre 2 de 2018

#### Tabla de contenido

1 Int	roducción	1
2 Jus	stificación de índices para los requerimientos	1
2.1	Requerimiento Funcional de consulta 10: Consultar consumo en SuperAndes	1
2.2	Requerimiento Funcional de Consulta 11:Consultar consumo en SuperAndes	2
2.3	Requerimiento Funcional de Consulta 12:Consultar Funcionamiento	2
2.4	Requerimiento Funcional de Consulta 13: Consultar los buenos Clientes	2
2.5	Información generada por Oracle	3
3 An	nálisis realizado	
3.1	Sentencias SQL	3
3.2	Distribución de los datos	
3.5	Planes de ejecución de los requerimientos sql y tiempo Requerimiento 10	4
3.6	Planes de ejecución de los requerimientos sql y tiempo Requerimiento 11	
3.7	Planes de ejecución de los requerimientos sql y tiempo Requerimiento 12	5
3.8	Planes de ejecución de los requerimientos sql y tiempo Requerimiento 13	7
4 Co	onstrucción de la aplicación	7
4.1	Proceso de carga	7
4.2	Análisis del proceso de optimización	

#### 1 Introducción

En este documento se expondrá los cambios que se llevaron a cabo durante la tercera iteración de la materia SISTEMAS TRANSACCIONALES con el fin de mejorar el rendimiento de nuestra aplicación SUPERANDES.

Estos cambios se implementan con el fin de disminuir el tiempo de consulta requerido por la aplicación al momento de analizar los datos pertenecientes al registro de la aplicación.

## 2 Justificación de índices para los requerimientos

## 2.1 Requerimiento Funcional de consulta 10: Consultar consumo en SuperAndes

Este requerimiento nos pide encontrar los clientes que realizaron al menos una compra en un rango de fechas determinado. Además de esto se requiere ordenar según los criterios deseados por el usuario.

Para solucionar este requerimiento nos valimos de índices B+ ya que este tipo de índices es permite seleccionar rangos con mayor facilidad gracias a su manejo de apuntadores. Por este motivo seria mucho más eficiente seleccionar las fechas indicadas por estos apuntadores para seleccionar los datos requeridos.

## 2.2 Requerimiento Funcional de Consulta 11:Consultar consumo en SuperAndes

En este requerimiento nos piden la negación del requerimiento de consulta anterior motivo por el cual se ejecutarían el mismo proceso de anterior pero con un variación importante. Esta variación consta de la resta lógica del conjunto resultante en el requerimiento anterior con el conjunto de datos que están situados en la tabla de compra.

#### 2.3 Requerimiento Funcional de Consulta 12:Consultar Funcionamiento

Para ejecutar este requerimiento se ejecutó una búsqueda parecida a las anteriores en la que se utilizaron índices B+ en las fechas de la tabla compra pero esta vez se contó con la función toChar (fechaDeCompra , 'ww') la cual tenia como objetivo obtener la semana de la compra y agrupar cada semana por el numero resultante de la *query* anteriormente mencionada. Así, podíamos obtener las compras de cada semana gracias a un apuntador al numero de la semana, al agrupar cada semana por producto y sumar la cantidad comprada se pudo obtener los productos más y menos vendidos cada semana gracias a las funciones max(cantidad) y min(cantidad) donde cantidad es la suma de las ventas de ese producto.

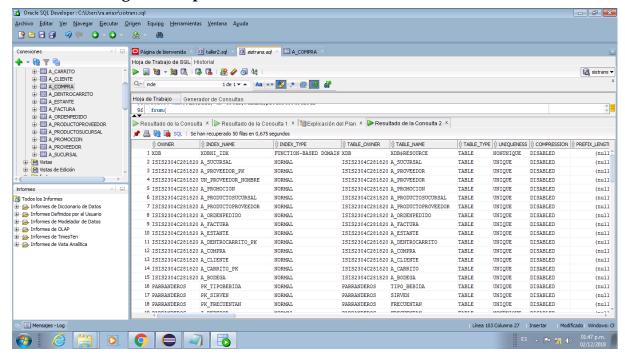
Para la segunda parte de este requerimiento se solicitaba obtener por cada semana los proveedores mas solicitados y menos solicitados por semana. Motivo por el cual tendría la misma lógica del párrafo anterior pero efectuando la consulta sobre la tabla de ordenPedido y sacando la cantidad máxima de ordenes realizadas a un mismo proveedor.

## 2.4 Requerimiento Funcional de Consulta 13: Consultar los buenos Clientes

En este requerimiento se emplearon indicies B+ en los cuales apuntaban hacia el id de cada grupo de compras mayores a 400 que es el nuestro precio para distinguir los clientes que pagan un total de 400 o más. Con este resultado sabemos el id de los clientes que efectuaban dichas compras y obtenemos la fechas realizadas por mes lo cual sería el ultimo criterio de agrupación.

Para las segunda parte de este requerimiento se empleó una unión entre los productos que son de tipo albarrotes y la compras efectuadas unianedo por el id de los productos comprados. Seguido a esto se agrupo por el id del cliente sumando la cantidad comprada de estos productos y finalmente retornando el id de aquellos que los compran.

## 2.5 Información generada por Oracle



Esto fueron creados por Oracle ya que al ser llaves primarias tienen la certeza de que son únicos y no se van a repetir.

#### 3 Análisis realizado

#### 3.1 Sentencias SQL

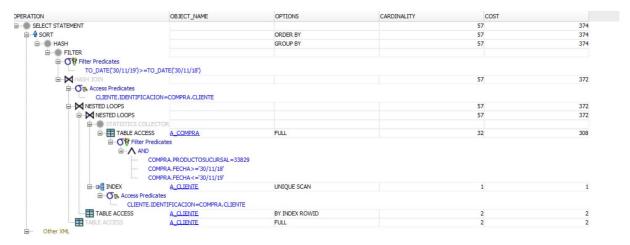
La Sentencias se encuentran en el documento .sql en el cual están contenidas y fueron analizadas en el requerimiento anterior.

#### 3.2 Distribución de los datos

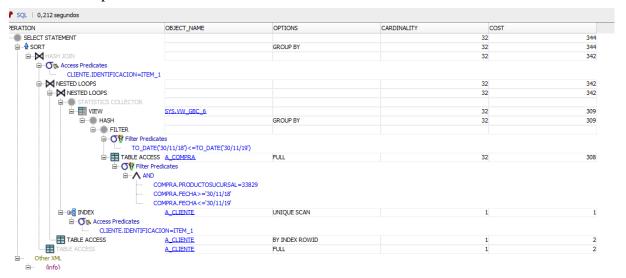
La distribución esta dada de la siguiente manera:

Bodega	1500
Carrito	10000
compra	90464
cliente	550000
DentroCarrito	500
Estante	3000
Factura	200000
OrdenPedido	500
productoProveedor	50000
productoSucursal	50000
promocion	200
proveedor	1000
sucursal	500
total	957664

## 3.5 Planes de ejecución de los requerimientos sql y tiempo Requerimiento 10



## ordenamiento por cantidad



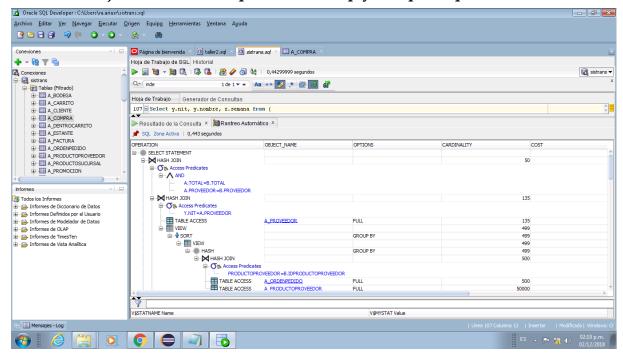
Ordenamiento por fecha

## 3.6 Planes de ejecución de los requerimientos sql y tiempo Requerimiento 11

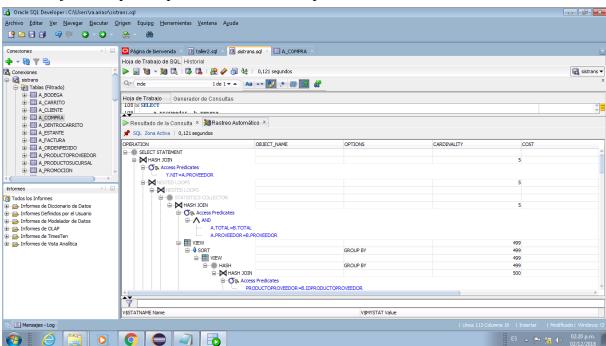


Clientes que no compraron

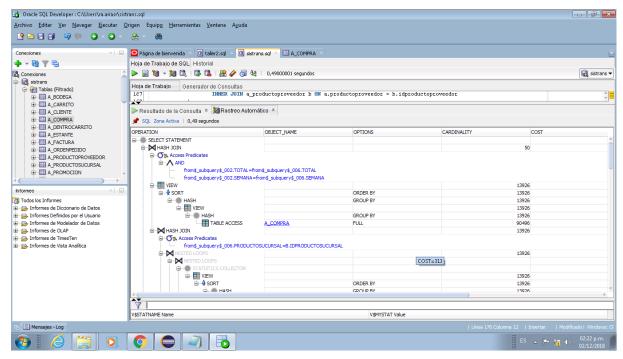
## 3.7 Planes de ejecución de los requerimientos sql y tiempo Requerimiento 12



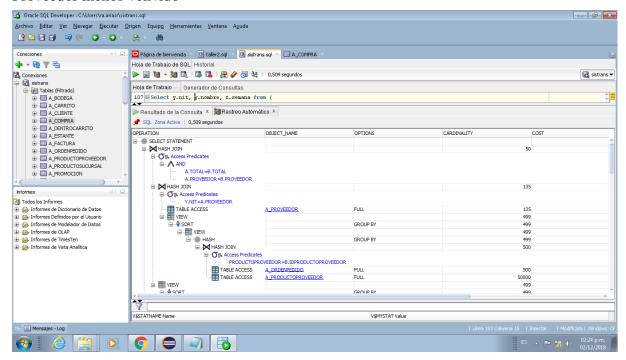
Plan de ejecución para el producto más vendido por semana



Plan de ejecución al producto menos vendido

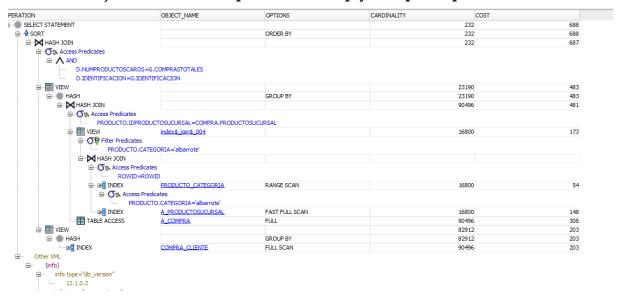


#### Proveedor menos venvido

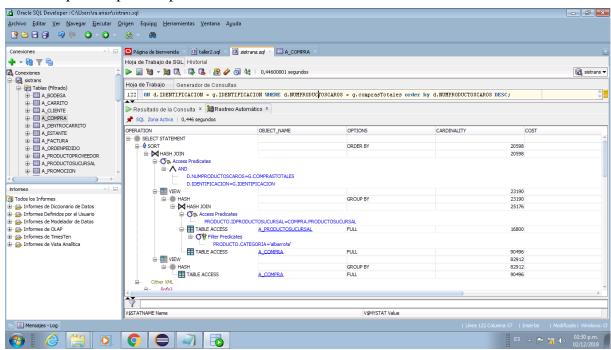


Proveedor menos vendido

## 3.8 Planes de ejecución de los requerimientos sql y tiempo Requerimiento 13



#### Clientes que compran un producto costoso



Clientes que compran albarrotes

## 4 Construcción de la aplicación

#### 4.1 Proceso de carga

Para llevar el proceso de carga se utilizó 2 herramientas. La primera de ellas un generador de datos llamado Mockaroo la cual nos generaba archivos .xml de 1000 datos máximo y la segunda fue excel para poder clonar datos generados cambiando las llaves primarias, fechas y llaves foráneas, esto ultimo se hacia con el fin de logran una gran cantidad de dato.

## 4.2 Análisis del proceso de optimización

La principal diferencia entre un programa el cual efectua los filtos de datos en la misma aplicación el lugar de la base de datos es el proceso de carga de estos datos es el tiempo de carga. Ya que el tiempo requerido por una aplicación la cual efectua las consultas con filtros es menor puesto que la base de datos respondera con una densidad de datos mucho menor y por lo tanto son menos datos los que hay que convertir a VO para que la interfaz los pueda leer adecuadamente, por otro lado las aplicaciones que efectúan filtros en la misma aplicación necesitaran mucho mas tiempo para poder convertir todos los datos a VO y agregarle el tiempo requerido por los filtro que se efectuaran durante la aplicación que dependiendo del algoritmo que se utilice para estos filtros demoraran más tiempo que el requerido por los filtros directamente en la consulta.