

INFORME DESARROLLO SUPERANDES

Juan Sebastián Bravo, Juan Esteban Méndez  
Universidad de los Andes, Bogotá, Colombia  
{js.bravo, je.mendez}@uniandes.edu.co  
Fecha de presentación: Diciembre 2 del 2018

Tabla de contenido

Introducción ..... 1

Modelos solicitados..... 1

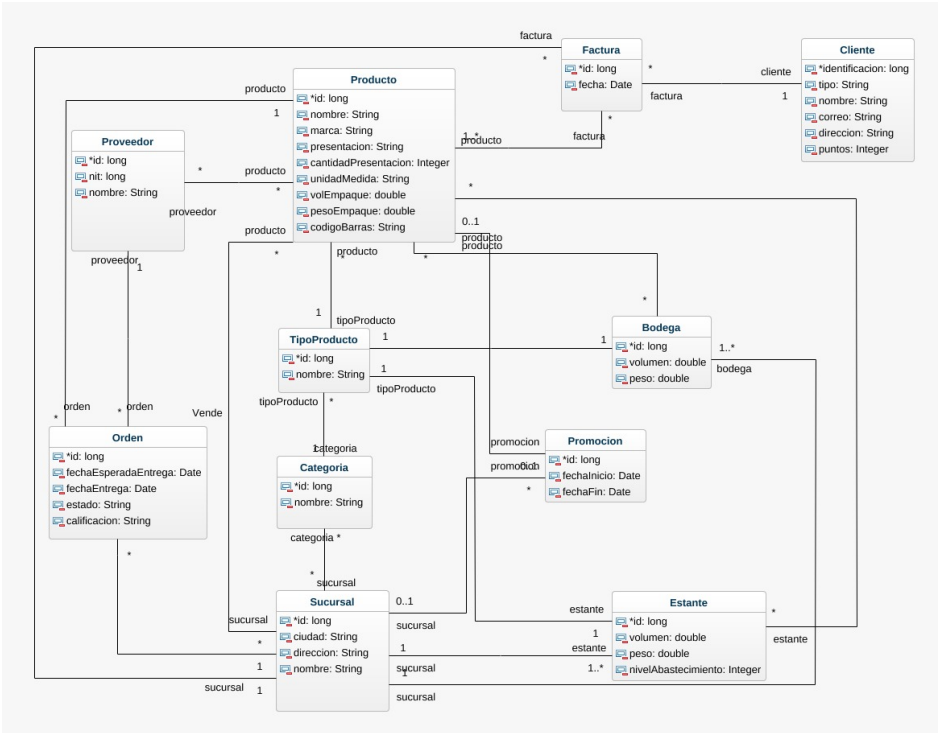
Requerimientos funcionales de consulta iteración 3 ..... 3

Introducción

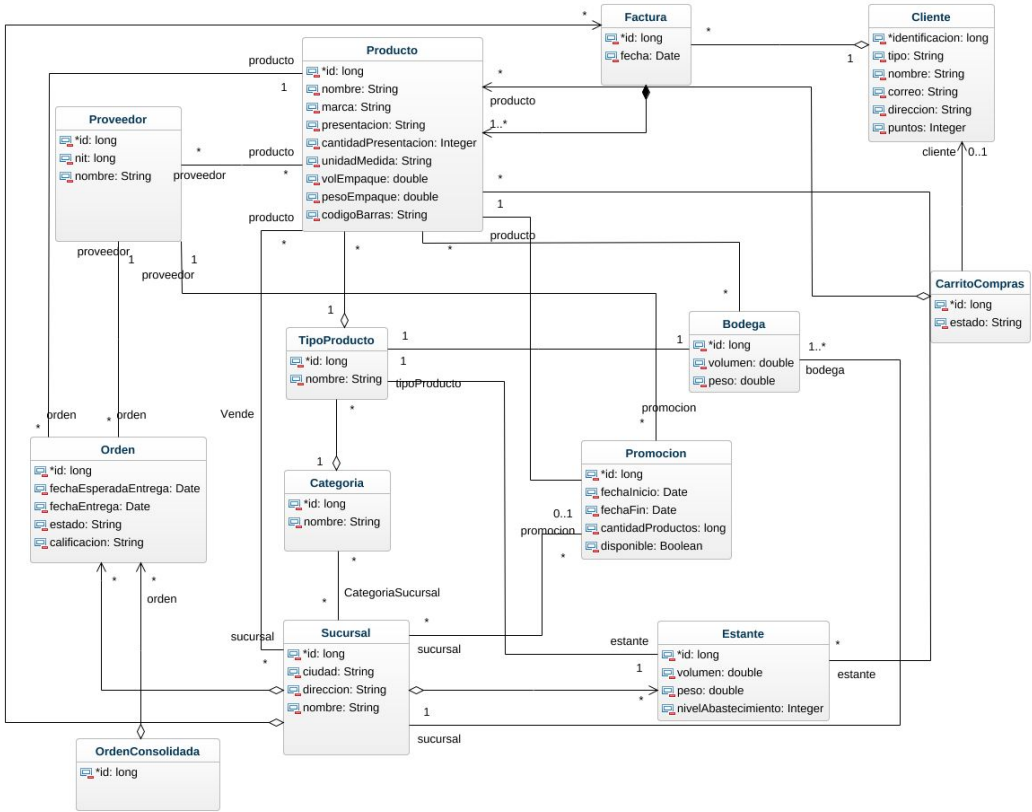
Este documento contiene la documentación sobre el caso de estudio de SuperAndes de la materia ISIS2304 – Sistemas Transaccionales tratado por el grupo B-05 de la materia en la tercera iteración del segundo semestre de 2018. En el siguiente documento se describe los modelos UML y relacionales necesarios y utilizados para el desarrollo de la aplicación que busca suplir los requerimientos de SuperAndes. Durante el documento se presentan los modelos en imágenes que permiten comprender la solución dada para los requerimientos funcionales de consulta propuestos, analizando su costo y el beneficio que trae el uso de índices en sistema manejador de bases de datos.

Modelos solicitados

a. UML (De la primera iteración)



b. UML actualizado



2. Modelo relacional (De la primera iteración)

SUCURSAL				ORDEN				ESTADO				ENTREGA			
id	Ciudad	Direccion	Nombre	id	idProveedor	idSucursal	idProducto	Cantidad	Precio	Estado	fechaEsperadaEntrega	fechaEntrega	calificacion		
Number	Varchar	Varchar	Varchar	Number	Number	Number	Number	Number	Number	Varchar	Date	Date	Varchar		
PK	NN	NN	NN	PK	FK (Proveedor.id)	FK (Sucursal.id)	FK (Producto.id)	CK(>0)	CK(>0)	NN(Entregado, No entregado)	NN	NN	FK (Entrega, No, bueno)		
CATEGORIA				CATEGORIASUCURSAL				PROVEEDOR							
id	Nombre	idSucursal		idCategoria	idSucursal										
Number	Varchar	Number		Number	Number										
PK	NN	FK (Sucursal.id)		FK (Categoria.id)	FK (Sucursal.id)										
TIPOPRODUCTO				PROVEE				CLIENTE							
id	Nombre	idCategoria		idProveedor	idProducto					id	Tipo	Nombre	Correo	Direccion	Puntos
Number	Varchar	Number		Number	Number					Number	Varchar	Varchar	Varchar	Varchar	Number
PK	NN	FK (Categoria.id)		PK	FK (Producto.id)					PK	(NATURAL, EMPRESA)	NN	NN		
PRODUCTO				FACTURA				FACTURAPRODUCTO							
id	Nombre	Marca	idTipoProducto	Presentacion	IntPresentacion	UnidMedida	VolEmpaque	PesoEmpaque	CodBarras	id	idCliente	idSucursal	Fecha	Total	
Number	Varchar	Varchar	Number	Varchar	Number	Varchar	Number	Number	Varchar	Number	Number	Number	Date	Number	
PK	NN	NN	FK (TipoProducto.id)	NN	CK (>0)	NN (g, ml)	CK (>0)	CK (>0)	NN	PK	FK (Cliente.id)	FK (Sucursal.id)	NN	CK (>0)	
VENDE				FACTURA				FACTURAPRODUCTO							
idSucursal	idProducto	Precio	PrecioUnidMedida	NivelOrden	CantRecompra					id	idCliente	idSucursal	Fecha	Total	
Number	Number	Number	Number	Number	Number					Number	Number	Number	Date	Number	
FK (Sucursal.id)	FK	CK (>0)	CK (>0)	CK (>0)	CK (>0)					PK	(Cliente.id, calificacion)	FK (Sucursal.id)	NN	CK (>0)	
ESTANTE				FACTURAPRODUCTO				FACTURAPRODUCTO							
id	idSucursal	idTipoProducto	Volumen	Peso	Abastecimiento					idFactura	idProducto	unidades			
Number	Number	Number	Number	Number	Number					Number	Number	Number			
PK	FK (Sucursal.id)	FK (TipoProducto.id)	CK (>0)	CK (>0)	CK (>0)					PK	FK (Factura.id)	FK (Producto.id)	CK (>0)		
BODEGA				FACTURAPRODUCTO				FACTURAPRODUCTO							
id	idSucursal	idTipoProducto	Volumen	Peso						idFactura	idProducto	unidades			
Number	Number	Number	Number	Number						Number	Number	Number			
PK	FK (Sucursal.id)	FK (TipoProducto.id)	CK (>0)	CK (>0)						PK	FK (Factura.id)	FK (Producto.id)	CK (>0)		
PRODUCTOESTANTE				PRODUCTOBODEGA				PRODUCTOBODEGA							
idProducto	idEstante	Cantidad		idProducto	idBodega	Cantidad				idFactura	idProducto	unidades			
Number	Number	Number		Number	Number	Number				Number	Number	Number			
FK (Producto.id)	FK	CK (>0)		PK	FK (Bodega.id)	CK (>0)				PK	FK (Factura.id)	FK (Producto.id)	CK (>0)		
PROMOCION				PRODUCTOBODEGA				PRODUCTOBODEGA							
id	idProducto	fechaInicio	fechaFin	idSucursal						idFactura	idProducto	unidades			
Number	Number	Date	Date	Number						Number	Number	Number			
PK	FK (Producto.id)	NN	NN	FK (Sucursal.id)						PK	FK (Factura.id)	FK (Producto.id)	CK (>0)		

Modelo relacional actualizado

SUCURSAL				ORDEN											
id	Ciudad	Direccion	Nombre	id	idProveedor	idSucursal	idProducto	Cantidad	Precio	Estado	FechaEsperada	FechaEntrega	Calificacion	idOrdenConsolidada	
Number	Varchar	Varchar	Varchar	Number	Number	Number	Number	Number	Number	Varchar	Date	Date	Varchar	Number	
PK	NN	NN	NN	PK	FK (Proveedor.id)	FK (Sucursal.id)	FK (Producto.id)	CK (0)	CK (0)	FK (Entregado, No entregado)	NN		FK (OrdenConsolidada.id)		
CATEGORIA				CATEGORIASUCURSAL				PROVEEDOR							
id	Nombre			idCategorias	idSucursal		id	Nit	Nombre						
Number	Varchar			Number	Number		Number	Number	Varchar						
PK	NN			FK (Categorias.id)	FK (Sucursal.id)		PK	NN	NN						
TIPOPRODUCTO				PROVEE				CLIENTE							
id	Nombre	idCategorias		idProveedor	idProducto			id	Tipo	Nombre	Correo	Direccion	Postos		
Number	Varchar	Number		Number	Number			Number	Varchar	Varchar	Varchar	Varchar	Number		
PK	NN	FK (Categorias.id)		FK (Proveedor.id)	FK (Producto.id)			PK	(NATURAL, EMPRESA)	NN	NN				
PRODUCTO															
id	Nombre	Marca	idTipoProducto	Presentacion	lastPresentacion	UnidadMedida	VolEmpaque	PesoEmpaque	CodBarras						
Number	Varchar	Varchar	Number	Varchar	Number	Varchar	Number	Number	Varchar						
PK	NN	NN	FK (TipoProducto.id)	NN	CK (0)	IN (gr, ml)	CK (0)	CK (0)	NN						
VENDE				FACTURA											
idSucursal	idProducto	Precio	PrecioUnidadMedida	NivelOrden	CantFlacomp		id	idCliente	idSucursal	Fecha	Total				
Number	Number	Number	Number	Number	Number		Number	Number	Number	Number	Number				
FK (Sucursal.id)	FK (Producto.id)	CK (0)	CK (0)	PK	CK (0)		PK	FK (Cliente.id)	FK (Sucursal.id)	NN	CK (0)				
ESTANTE				FACTURAPRODUCTO											
id	idSucursal	idTipoProducto	Volume	Peso	idAbacoInventario		idFactura	idProducto	idFactura						
Number	Number	Number	Number	Number	Number		Number	Number	Number						
PK	FK (Sucursal.id)	FK (TipoProducto.id)	CK (0)	CK (0)	CK (0)		FK (Factura.id)	FK (Producto.id)	FK (Factura.id)						
BODEGA															
id	idSucursal	idTipoProducto	Volume	Peso											
Number	Number	Number	Number	Number											
PK	FK (Sucursal.id)	FK (TipoProducto.id)	CK (0)	CK (0)											
PRODUCTOESTANTE				PRODUCTOBODEGA											
idProducto	idEstante	Cantidad		idProducto	idBodega	Cantidad									
Number	Number	Number		Number	Number	Number									
FK (Producto.id)	FK (Estante.id)	CK (0)		FK (Producto.id)	FK (Bodega.id)	CK (0)									
PROMOCION															
id	idProducto	fechaInicio	cantidadProducto	fechaFin	idSucursal	idProveedor	ididProductosVendidos								
Number	Number	Date	Number	Date	Number	Number	Number								
PK	FK (Producto.id)	NN			FK (Sucursal.id)	FK (Proveedor.id)									
CARRITOCOMPRAS				ORDENCONSOLIDADA											
id	estado	idCliente	idSucursal	id											
Number	Number	Number	Number	Number											
PK	NN		FK (Vendedor.id)	PK											
VENDECARRITO															
idCarrito	idProducto	cantidadCarrito													
Number	Number	Number													
FK (CarritoCompra.id)	FK (Vendedor.id)														

3. Índices creados

OWNER	INDEX_NAME	INDEX_TYPE	TABLE_OWNER	TABLE_NAME	TABLE_TYPE	UNIQUENESS	COMPRESSION	PREFIX_LENGTH	TABLESPACE_NAME	IN_TRANS	MAX_TRANS	INITIAL_EXTENT	NEXT_EXTENT
XDB	XDB\$I_IDX	FUNCTION-BASED DOMAIN	XDB	RESOURCE	TABLE	NONUNIQUE	DISABLED	(null)	(null)	0	0	(null)	(null)
ISIS23048211820	RESERVAS_FK	NORMAL	ISIS23048211820	RESERVAS	TABLE	UNIQUE	DISABLED	(null)	TBSPROD	2	255	65536	1048576
ISIS23048211820	FK_PROVEEDOR	NORMAL	ISIS23048211820	PROVEEDOR	TABLE	UNIQUE	DISABLED	(null)	TBSPROD	2	255	65536	1048576
ISIS23048211820	UNI_NIT	NORMAL	ISIS23048211820	PROVEEDOR	TABLE	UNIQUE	DISABLED	(null)	TBSPROD	2	255	65536	1048576
ISIS23048211820	PROVEE_FK	NORMAL	ISIS23048211820	PROVEE	TABLE	UNIQUE	DISABLED	(null)	TBSPROD	2	255	65536	1048576
ISIS23048211820	PROMOCIONCONSUCURSAL_FK	NORMAL	ISIS23048211820	PROMOCIONCONSUCURSAL	TABLE	UNIQUE	DISABLED	(null)	TBSPROD	2	255	(null)	(null)
ISIS23048211820	PROMOCION_FK	NORMAL	ISIS23048211820	PROMOCION	TABLE	UNIQUE	DISABLED	(null)	TBSPROD	2	255	65536	1048576
ISIS23048211820	PRODUCTOESTANTE_FK	NORMAL	ISIS23048211820	PRODUCTOESTANTE	TABLE	UNIQUE	DISABLED	(null)	TBSPROD	2	255	65536	1048576
ISIS23048211820	PRODUCTOBODEGA_FK	NORMAL	ISIS23048211820	PRODUCTOBODEGA	TABLE	UNIQUE	DISABLED	(null)	TBSPROD	2	255	65536	1048576
ISIS23048211820	PRODUCTO_FK	NORMAL	ISIS23048211820	PRODUCTO	TABLE	UNIQUE	DISABLED	(null)	TBSPROD	2	255	65536	1048576
ISIS23048211820	SYS_C00501875	NORMAL	ISIS23048211820	ORDENCONSOLIDADA	TABLE	UNIQUE	DISABLED	(null)	TBSPROD	2	255	(null)	(null)
ISIS23048211820	ORDEN_FK	NORMAL	ISIS23048211820	ORDEN	TABLE	UNIQUE	DISABLED	(null)	TBSPROD	2	255	65536	1048576
ISIS23048211820	FECHAVISITANI	NORMAL	ISIS23048211820	FRECUENTAN	TABLE	NONUNIQUE	DISABLED	(null)	TBSPROD	2	255	65536	1048576
ISIS23048211820	FACTURAPRODUCTO_FK	NORMAL	ISIS23048211820	FACTURAPRODUCTO	TABLE	UNIQUE	DISABLED	(null)	TBSPROD	2	255	65536	1048576
ISIS23048211820	FACTURA_FK	NORMAL	ISIS23048211820	FACTURA	TABLE	UNIQUE	DISABLED	(null)	TBSPROD	2	255	65536	1048576
ISIS23048211820	FACTURAINDEX	NORMAL	ISIS23048211820	FACTURA	TABLE	NONUNIQUE	DISABLED	(null)	TBSPROD	2	255	65536	1048576
ISIS23048211820	ESTANTE_FK	NORMAL	ISIS23048211820	ESTANTE	TABLE	UNIQUE	DISABLED	(null)	TBSPROD	2	255	65536	1048576
ISIS23048211820	CLIENTE_FK	NORMAL	ISIS23048211820	CLIENTE	TABLE	UNIQUE	DISABLED	(null)	TBSPROD	2	255	65536	1048576
ISIS23048211820	CLIENTEINDEX	NORMAL	ISIS23048211820	CLIENTE	TABLE	NONUNIQUE	DISABLED	(null)	TBSPROD	2	255	65536	1048576
ISIS23048211820	CATEGORIASUCURSAL_FK	NORMAL	ISIS23048211820	CATEGORIASUCURSAL	TABLE	UNIQUE	DISABLED	(null)	TBSPROD	2	255	65536	1048576
ISIS23048211820	CATEGORIA_FK	NORMAL	ISIS23048211820	CATEGORIA	TABLE	UNIQUE	DISABLED	(null)	TBSPROD	2	255	65536	1048576
ISIS23048211820	CARRITOCOMPRAS_FK	NORMAL	ISIS23048211820	CARRITOCOMPRAS	TABLE	UNIQUE	DISABLED	(null)	TBSPROD	2	255	65536	1048576
ISIS23048211820	BODEGA_FK	NORMAL	ISIS23048211820	BODEGA	TABLE	UNIQUE	DISABLED	(null)	TBSPROD	2	255	65536	1048576
ISIS23048211820	BEBEDORES_FK	NORMAL	ISIS23048211820	BEBEDORES	TABLE	UNIQUE	DISABLED	(null)	TBSPROD	2	255	65536	1048576
ISIS23048211820	A_VISITAN_FK	NORMAL	ISIS23048211820	A_VISITAN	TABLE	UNIQUE	DISABLED	(null)	TBSPROD	2	255	65536	1048576
ISIS23048211820	A_TIPOBEBIDA_FK	NORMAL	ISIS23048211820	A_TIPOBEBIDA	TABLE	UNIQUE	DISABLED	(null)	TBSPROD	2	255	65536	1048576
ISIS23048211820	UN_TIPOBEB_NOMBRE	NORMAL	ISIS23048211820	A_TIPOBEBIDA	TABLE	UNIQUE	DISABLED	(null)	TBSPROD	2	255	65536	1048576
ISIS23048211820	A_SIRVEN_FK	NORMAL	ISIS23048211820	A_SIRVEN	TABLE	UNIQUE	DISABLED	(null)	TBSPROD	2	255	65536	1048576
ISIS23048211820	A_GUSTAN_FK	NORMAL	ISIS23048211820	A_GUSTAN	TABLE	UNIQUE	DISABLED	(null)	TBSPROD	2	255	65536	1048576
ISIS23048211820	A_BEBIDA_FK	NORMAL	ISIS23048211820	A_BEBIDA	TABLE	UNIQUE	DISABLED	(null)	TBSPROD	2	255	65536	1048576
ISIS23048211820	A_BEDEDOR_FK	NORMAL	ISIS23048211820	A_BEDEDOR	TABLE	UNIQUE	DISABLED	(null)	TBSPROD	2	255	65536	1048576
ISIS23048211820	A_BAR_FK	NORMAL	ISIS23048211820	A_BAR	TABLE	UNIQUE	DISABLED	(null)	TBSPROD	2	255	65536	1048576
ISIS23048211820	VENDE_FK	NORMAL	ISIS23048211820	VENDE	TABLE	UNIQUE	DISABLED	(null)	TBSPROD	2	255	65536	1048576
ISIS23048211820	TIPOPRODUCTOINDEX	NORMAL	ISIS23048211820	TIPOPRODUCTO	TABLE	NONUNIQUE	DISABLED	(null)	TBSPROD	2	255	65536	1048576
ISIS23048211820	TIPOPRODUCTO_FK	NORMAL	ISIS23048211820	TIPOPRODUCTO	TABLE	UNIQUE	DISABLED	(null)	TBSPROD	2	255	65536	1048576
ISIS23048211820	SUCURSAL_FK	NORMAL	ISIS23048211820	SUCURSAL	TABLE	UNIQUE	DISABLED	(null)	TBSPROD	2	255	65536	1048576

En la imagen se encuentran listados los índices que se encuentran en la base de datos de SuperAndes. En detalle, Oracle creó automáticamente todos los índices relacionados con las llaves primarias de las tablas. También se encuentran los índices CLIENTEINDEX y TIPOPRODUCTOINDEX, los cuales son explicados más adelante.

Requerimientos funcionales de consulta iteración 3

- Consultar consumo en SuperAndes

```
SELECT CLIENTE.ID, CLIENTE.NOMBRE, FACTURA.ID, FACTURA.IDSUCURSAL, FACTURA.FECHA, FACTURA.TOTAL, FACTURAPRODUCTO.IDPRODUCTO,
```

```
FACTURAPRODUCTO.UNIVENDIDAS

FROM CLIENTE

INNER JOIN FACTURA ON CLIENTE.ID = FACTURA.IDCLIENTE

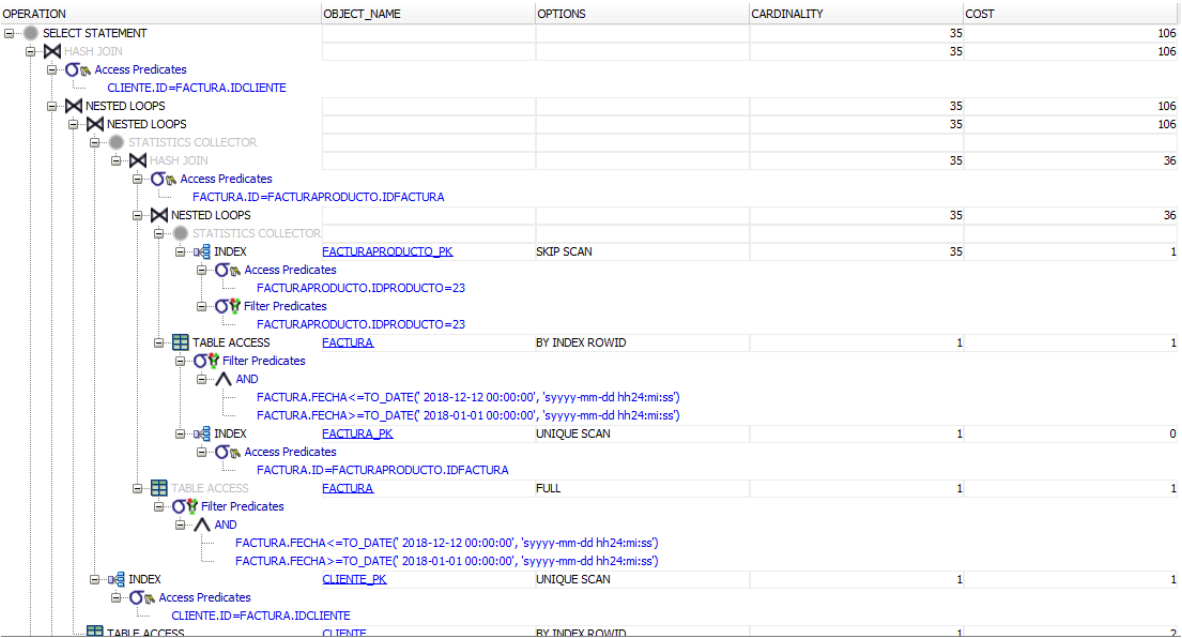
INNER JOIN FACTURAPRODUCTO ON FACTURA.ID = FACTURAPRODUCTO.IDFACTURA

WHERE FACTURAPRODUCTO.IDPRODUCTO = 23 AND FACTURA.FECHA BETWEEN '01/01/2018'
AND '12/12/2018';

-- TABLAS ACCEDIDAS

-- CARDINALIDAD CLIENTE: 1'023.057
-- CARDINALIDAD FACTURA: 270.163
-- CARDINALIDAD FACTURAPRODUCTO: 257.559

-- TIEMPO DE RESPUESTA: 0.056 segundos.
```



Viendo la explicación del plan de Oracle, notamos que tanto el costo como la cardinalidad de la operación es mucho más bajo de lo que esperábamos, reduciéndose por el índice compuesto FACTURAPRODUCTO.PK, puesto que la búsqueda es por llave primaria. La creación de un índice en este requerimiento no es necesario, puesto que el índice automáticamente generado por Oracle FACTURAPRODUCTO.PK es el único necesario, utilizando todos los campos que posee FACTURAPRODUCTO.

Cambiando los parámetros

```
SELECT CLIENTE.ID, CLIENTE.NOMBRE, FACTURA.ID, FACTURA.IDSUCURSAL,
FACTURA.FECHA, FACTURA.TOTAL, FACTURAPRODUCTO.IDPRODUCTO,
FACTURAPRODUCTO.UNIVENDIDAS

FROM CLIENTE

INNER JOIN FACTURA ON CLIENTE.ID = FACTURA.IDCLIENTE

INNER JOIN FACTURAPRODUCTO ON FACTURA.ID = FACTURAPRODUCTO.IDFACTURA

WHERE FACTURAPRODUCTO.IDPRODUCTO = 24 AND FACTURA.FECHA BETWEEN '01/10/2018'
AND '12/12/2018';
```

```
-- TABLAS ACCEDIDAS

-- CARDINALIDAD CLIENTE: 1'023.057
-- CARDINALIDAD FACTURA: 270.163
-- CARDINALIDAD FACTURAPRODUCTO: 257.559

-- TIEMPO DE RESPUESTA: 0.016 segundos.
```

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST	
SELECT STATEMENT				28	85
HASH JOIN				28	85
Access Predicates					
CLIENTE.ID=FACTURA.IDCLIENTE					
NESTED LOOPS				28	85
STATISTICS COLLECTOR					
HASH JOIN				28	29
Access Predicates					
FACTURA.ID=FACTURAPRODUCTO.IDFACTURA					
NESTED LOOPS				28	29
STATISTICS COLLECTOR					
INDEX	FACTURAPRODUCTO_PK	SKIP SCAN		28	1
Access Predicates					
FACTURAPRODUCTO.IDPRODUCTO=24					
Filter Predicates					
FACTURAPRODUCTO.IDPRODUCTO=24					
TABLE ACCESS	FACTURA	BY INDEX ROWID		1	1
Filter Predicates					
AND					
FACTURA.FECHA>=TO_DATE(' 2018-10-01 00:00:00', 'yyyy-mm-dd hh24:mi:ss')					
FACTURA.FECHA<=TO_DATE(' 2018-12-12 00:00:00', 'yyyy-mm-dd hh24:mi:ss')					
INDEX	FACTURA_PK	UNIQUE SCAN		1	0
Access Predicates					
FACTURA.ID=FACTURAPRODUCTO.IDFACTURA					
TABLE ACCESS	FACTURA	FULL		1	1
Filter Predicates					
AND					
FACTURA.FECHA>=TO_DATE(' 2018-10-01 00:00:00', 'yyyy-mm-dd hh24:mi:ss')					
FACTURA.FECHA<=TO_DATE(' 2018-12-12 00:00:00', 'yyyy-mm-dd hh24:mi:ss')					
INDEX	CLIENTEINDEX	RANGE SCAN		1	2
Access Predicates					
CLIENTE.ID=FACTURA.IDCLIENTE					
INDEX	CLIENTEINDEX	FAST FULL SCAN		1	2

Cambiando los parámetros de *fecha final* de la operación y el *id del producto*, el costo y la cardinalidad de la operación se redujeron de manera significativa, así como su tiempo de respuesta.

- Consultar NO consumo en SuperAndes

Notamos que este requerimiento funcional de consulta es el complemento de requerimiento funcional de consulta explicando en el literal anterior, por lo cual decidimos usar MINUS del conjunto total.

```
SELECT * FROM (SELECT CLIENTE.ID, CLIENTE.NOMBRE, TRUNC(FACTURA.FECHA) AS
FECHA, SUM (FACTURAPRODUCTO.UNIVENDIDAS) AS SUMA_TOT
FROM CLIENTE
FULL OUTER JOIN FACTURA ON CLIENTE.ID = FACTURA.IDCLIENTE
FULL OUTER JOIN FACTURAPRODUCTO ON FACTURA.ID = FACTURAPRODUCTO.IDFACTURA
GROUP BY CLIENTE.ID, CLIENTE.NOMBRE, TRUNC(FACTURA.FECHA)
MINUS (SELECT CLIENTE.ID, CLIENTE.NOMBRE, TRUNC(FACTURA.FECHA), SUM
(FACTURAPRODUCTO.UNIVENDIDAS) AS SUMA_TOT
FROM CLIENTE
FULL OUTER JOIN FACTURA ON CLIENTE.ID = FACTURA.IDCLIENTE
FULL OUTER JOIN FACTURAPRODUCTO ON FACTURA.ID = FACTURAPRODUCTO.IDFACTURA
WHERE FACTURA.FECHA BETWEEN '01/01/2018' AND '31/12/2018'
GROUP BY CLIENTE.ID, CLIENTE.NOMBRE, TRUNC(FACTURA.FECHA)))

-- TABLAS ACCEDIDAS

-- CARDINALIDAD CLIENTE: 1'023.057
-- CARDINALIDAD FACTURA: 270.163
-- CARDINALIDAD FACTURAPRODUCTO: 257.559

-- TIEMPO DE RESPUESTA: 5 segundos.
```

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT				28647
VIEW				1026878
MINUS				28647
SORT				
HASH		UNIQUE		1026878
VIEW		GROUP BY		1026878
SYS.VW_F01_0				1026878
HASH JOIN		FULL OUTER		1026878
Access Predicates				
FACTURA.ID=FACTURAPRODUCTO.IDFACTURA				
INDEX	FACTURAPRODUCTO_PK	FAST FULL SCAN		92
VIEW	SYS.VW_F01_1			1023251
HASH JOIN		FULL OUTER		1023251
Access Predicates				
CLIENTE.ID=FACTURA.IDCLIENTE				
TABLE ACCESS	FACTURA	FULL		12695
TABLE ACCESS	CLIENTE	FULL		1023057
SORT		UNIQUE		12740
HASH		GROUP BY		12740
HASH JOIN		OUTER		12740
Access Predicates				
CLIENTE.ID(+) = FACTURA.IDCLIENTE				
MERGE JOIN		OUTER		12740
TABLE ACCESS	FACTURA	BY INDEX ROWID		12695
Filter Predicates				
AND				
FACTURA.FECHA >= TO_DATE('2018-01-01 00:00:00', 'yyyy-mm-dd hh24:mi:ss')				
FACTURA.FECHA <= TO_DATE('2018-12-31 00:00:00', 'yyyy-mm-dd hh24:mi:ss')				
INDEX	FACTURA_PK	FULL SCAN		12695
SORT		JOIN		92
Access Predicates				
FACTURA.ID=FACTURAPRODUCTO.IDFACTURA(+)				
Filter Predicates				
FACTURA.ID=FACTURAPRODUCTO.IDFACTURA(+)				
INDEX	FACTURAPRODUCTO_PK	FULL SCAN		92

Analizando la operación de consulta, decidimos crear un índice compuesto denominado CLIENTEINDEX en los campos ID y NOMBRE de CLIENTE. Lo anterior debido a que la consulta utiliza continuamente estos campos de la tabla CLIENTE, por lo cual un índice podría filtrar más rápidamente.

```
CREATE INDEX CLIENTEINDEX ON CLIENTE (ID, NOMBRE)
```

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT				1026878
VIEW				1026878
MINUS				24985
SORT				
HASH		UNIQUE		1026878
VIEW		GROUP BY		1026878
SYS.VW_F01_0				1026878
HASH JOIN		FULL OUTER		1026878
Access Predicates				
FACTURA.ID=FACTURAPRODUCTO.IDFACTURA				
INDEX	FACTURAPRODUCTO_PK	FAST FULL SCAN		92
VIEW	SYS.VW_F01_1			1023251
HASH JOIN		FULL OUTER		1023251
Access Predicates				
CLIENTE.ID=FACTURA.IDCLIENTE				
TABLE ACCESS	FACTURA	FULL		12695
INDEX	CLIENTEINDEX	FAST FULL SCAN		1023057
SORT		UNIQUE		12740
HASH		GROUP BY		12740
HASH JOIN		OUTER		12740
Access Predicates				
CLIENTE.ID(+) = FACTURA.IDCLIENTE				
MERGE JOIN		OUTER		12740
TABLE ACCESS	FACTURA	BY INDEX ROWID		12695
Filter Predicates				
AND				
FACTURA.FECHA >= TO_DATE('2018-01-01 00:00:00', 'yyyy-mm-dd hh24:mi:ss')				
FACTURA.FECHA <= TO_DATE('2018-12-31 00:00:00', 'yyyy-mm-dd hh24:mi:ss')				
INDEX	FACTURA_PK	FULL SCAN		12695
SORT		JOIN		92
Access Predicates				
FACTURA.ID=FACTURAPRODUCTO.IDFACTURA(+)				
Filter Predicates				
FACTURA.ID=FACTURAPRODUCTO.IDFACTURA(+)				
INDEX	FACTURAPRODUCTO_PK	FULL SCAN		92

Revisando de nuevo el plan de consulta después de haber creado el índice compuesto CLIENTEINDEX, vemos que el costo disminuyó notablemente.

```
NUEVO TIEMPO DE RESPUESTA: 3.8 segundos
```



Cambiando los parámetros

```
SELECT * FROM (SELECT CLIENTE.ID, CLIENTE.NOMBRE, TRUNC(FACTURA.FECHA) AS
FECHA
FROM CLIENTE
FULL OUTER JOIN FACTURA ON CLIENTE.ID = FACTURA.IDCLIENTE
FULL OUTER JOIN FACTURAPRODUCTO ON FACTURA.ID = FACTURAPRODUCTO.IDFACTURA
GROUP BY CLIENTE.ID, CLIENTE.NOMBRE, TRUNC(FACTURA.FECHA)
MINUS (SELECT CLIENTE.ID, CLIENTE.NOMBRE, TRUNC(FACTURA.FECHA), SUM
(FACTURAPRODUCTO.UNIVENDIDAS) AS SUMA_TOT
FROM CLIENTE
FULL OUTER JOIN FACTURA ON CLIENTE.ID = FACTURA.IDCLIENTE
FULL OUTER JOIN FACTURAPRODUCTO ON FACTURA.ID = FACTURAPRODUCTO.IDFACTURA
WHERE FACTURA.FECHA BETWEEN '01/10/2018' AND '31/12/2018'
GROUP BY CLIENTE.ID, CLIENTE.NOMBRE, TRUNC(FACTURA.FECHA)))

-- TABLAS ACCEDIDAS

-- CARDINALIDAD CLIENTE: 1'023.057
-- CARDINALIDAD FACTURA: 270.163
-- CARDINALIDAD FACTURAPRODUCTO: 257.559

-- TIEMPO DE RESPUESTA: 4.8 segundos.
```

Cambiando los parámetros de *fecha final* de la operación y el *id de la sucursal* en la vista gerente, el costo y la cardinalidad de la operación se redujeron de manera significativa, así como su tiempo de respuesta.

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			1026878	15156
VIEW			1026878	15156
MINUS				
SORT		UNIQUE	1026878	15024
HASH		GROUP BY	1026878	15024
VIEW	SYS.VW_FQ1_0		1026878	1396
HASH JOIN		FULL OUTER	1026878	1396
Access Predicates	FACTURA.ID=FACTURAPRODUCTO.IDFACTURA			
INDEX	FACTURAPRODUCTO_PK	FAST FULL SCAN	92	2
VIEW	SYS.VW_FQ1_1		1023251	1392
HASH JOIN		FULL OUTER	1023251	1392
Access Predicates	CLIENTE.ID=FACTURA.IDCLIENTE			
TABLE ACCESS	FACTURA	FULL	12695	477
INDEX	CLIENTEINDEX	FAST FULL SCAN	1023057	912
SORT		UNIQUE	1	132
HASH		GROUP BY	1	132
NESTED LOOPS		OUTER	2	130
MERGE JOIN		OUTER	2	126
TABLE ACCESS	FACTURA	BY INDEX ROWID	1	124
Filter Predicates	AND FACTURA.IDSUCURSAL=38 FACTURA.FECHA>=TO_DATE(' 2018-12-01 00:00:00', 'yyyy-mm-dd hh24:miss') FACTURA.FECHA<=TO_DATE(' 2018-12-31 00:00:00', 'yyyy-mm-dd hh24:miss')			
INDEX	FACTURA_PK	FULL SCAN	12695	25
SORT		JOIN	92	2
Access Predicates	FACTURA.ID=FACTURAPRODUCTO.IDFACTURA(+)			
Filter Predicates	FACTURA.ID=FACTURAPRODUCTO.IDFACTURA(+)			
INDEX	FACTURAPRODUCTO_PK	FULL SCAN	92	1
INDEX	CLIENTEINDEX	RANGE SCAN	1	2

- Consultar funcionamiento en SuperAndes
- Notamos que este requerimiento funcional de consulta puede facilitarse con el uso de vistas, por lo cual creamos las siguientes:

```
CREATE VIEW PRODUCTOSVENDIDOSXSEMANA AS(
SELECT TO_NUMBER(TO_CHAR(TO_DATE(FACTURA.FECHA, 'DD/MM/YYYY'), 'WW')) AS
SEMANA, FACTURAPRODUCTO.IDPRODUCTO, SUM(FACTURAPRODUCTO.UNIVENDIDAS) AS SUMA
```

```
FROM FACTURA
INNER JOIN FACTURAPRODUCTO ON FACTURA.ID = FACTURAPRODUCTO.IDFACTURA
GROUP BY
TO_NUMBER(to_char(to_date(FACTURA.FECHA, 'DD/MM/YYYY'), 'WW')),FACTURAPRODUCTO
.IDPRODUCTO
);

CREATE VIEW PEDIDOSPROVEEDORXSEMANA AS(
SELECT
TO_NUMBER(to_char(to_date(ORDEN.FECHAESPERADAENTREGA, 'DD/MM/YYYY'), 'WW')) AS
SEMANA, IDPROVEEDOR, COUNT(IDPROVEEDOR) AS NUM_PEDIDOS
FROM ORDEN
GROUP BY
TO_NUMBER(to_char(to_date(ORDEN.FECHAESPERADAENTREGA, 'DD/MM/YYYY'), 'WW')),
IDPROVEEDOR);
```

- Caso 1: Producto más vendido en cada semana del año

```
SELECT PRODUCTOSVENDIDOSXSEMANA.SEMANA, PRODUCTOSVENDIDOSXSEMANA.IDPRODUCTO,
PRODUCTO.NOMBRE, SUBQUERY.MAXIMO
FROM
(SELECT SEMANA, MAX(SUMA) AS MAXIMO
FROM PRODUCTOSVENDIDOSXSEMANA
GROUP BY SEMANA) SUBQUERY
INNER JOIN PRODUCTOSVENDIDOSXSEMANA ON PRODUCTOSVENDIDOSXSEMANA.SUMA =
SUBQUERY.MAXIMO AND PRODUCTOSVENDIDOSXSEMANA.SEMANA = SUBQUERY.SEMANA
INNER JOIN PRODUCTO ON PRODUCTO.ID = PRODUCTOSVENDIDOSXSEMANA.IDPRODUCTO
ORDER BY PRODUCTOSVENDIDOSXSEMANA.SEMANA;

-- TABLAS ACCEDIDAS

-- CARDINALIDAD PRODUCTO: 2.563
-- CARDINALIDAD FACTURA: 270.163
-- CARDINALIDAD FACTURAPRODUCTO: 257.559

-- TIEMPO DE RESPUESTA: 3.1 segundos.
```

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			1	191
SORT		ORDER BY	1	191
HASH JOIN			1	190
Access Predicates	PRODUCTO.ID=PRODUCTOSVENDIDOSXSEMANA.IDPRODUCTO			
NESTED LOOPS			1	190
NESTED LOOPS			1	190
STATISTICS COLLECTOR				
HASH JOIN			1	189
Access Predicates				
AND	PRODUCTOSVENDIDOSXSEMANA.SUMA=SUBQUERY.MAXIMO PRODUCTOSVENDIDOSXSEMANA.SEMANA=SUBQUERY.SEMANA			
VIEW			8	95
HASH		GROUP BY	8	95
VIEW	PRODUCTOSVENDIDOSXSEMANA		8	94
HASH		GROUP BY	8	94
NESTED			92	93
NESTED			92	93
INDEX	FACTURAPRODUCTO_PK	FULL SCAN	92	1
INDEX	FACTURA_PK	UNIQUE SCAN	92	1
Access Predicates	FACTURA.ID=FACTURAPRODUCTO.IDFACTURA		1	0
TABLE	FACTURA		1	1
VIEW	PRODUCTOSVENDIDOSXSEMANA	BY INDEX ROWID	8	94
HASH		GROUP BY	8	94
NESTED LOOPS			92	93
NESTED LOC			92	93
INDEX	FACTURAPRODUCTO_PK	FULL SCAN	92	1
INDEX	FACTURA_PK	UNIQUE SCAN	92	1
Access Predicates	FACTURA.ID=FACTURAPRODUCTO.IDFACTURA		1	0
TABLE	FACTURA	BY INDEX ROWID	1	1
INDEX	PRODUCTO_PK	UNIQUE SCAN	1	0



Notamos que con la vista creada llamada PRODUCTOSVENDIDOSXSEMANA se redujo el tiempo de respuesta de gran manera.

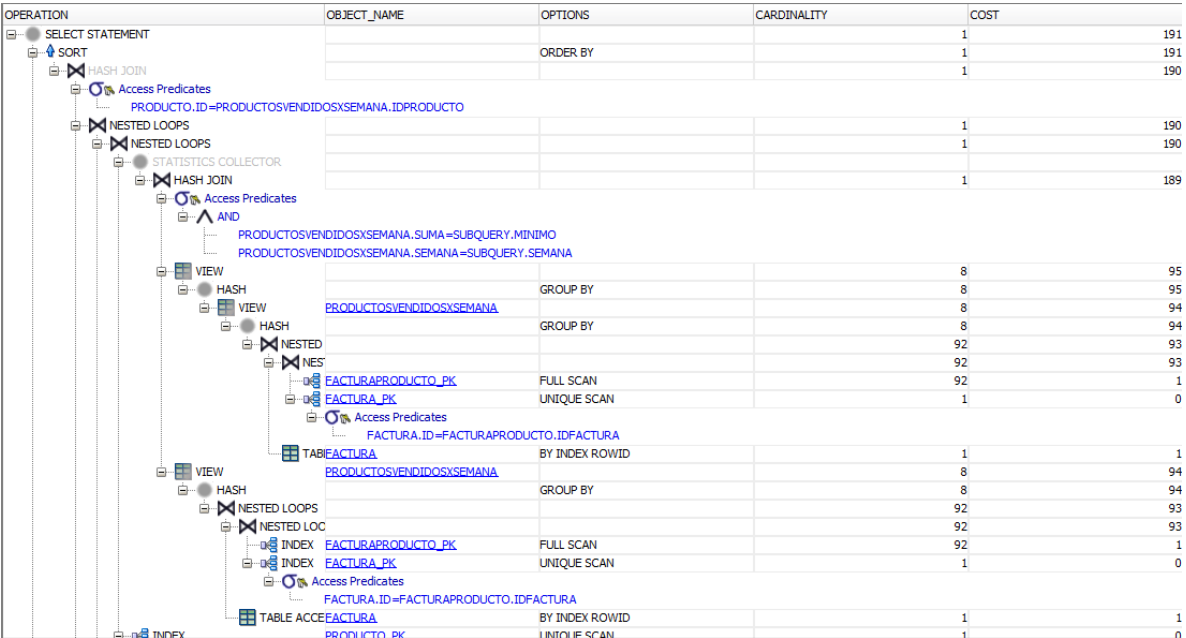
- Caso 2: Producto menos vendido en cada semana del año

```
SELECT PRODUCTOSVENDIDOSXSEMANA.SEMANA, PRODUCTOSVENDIDOSXSEMANA.IDPRODUCTO,
PRODUCTO.NOMBRE, SUBQUERY.MINIMO
FROM
(SELECT SEMANA, MIN(SUMA) AS MINIMO
FROM PRODUCTOSVENDIDOSXSEMANA
GROUP BY SEMANA) SUBQUERY
INNER JOIN PRODUCTOSVENDIDOSXSEMANA ON PRODUCTOSVENDIDOSXSEMANA.SUMA =
SUBQUERY.MINIMO AND PRODUCTOSVENDIDOSXSEMANA.SEMANA = SUBQUERY.SEMANA
INNER JOIN PRODUCTO ON PRODUCTO.ID = PRODUCTOSVENDIDOSXSEMANA.IDPRODUCTO
ORDER BY PRODUCTOSVENDIDOSXSEMANA.SEMANA;

-- TABLAS ACCEDIDAS

-- CARDINALIDAD PRODUCTO: 2.563
-- CARDINALIDAD FACTURA: 270.163
-- CARDINALIDAD FACTURAPRODUCTO: 257.559

-- TIEMPO DE RESPUESTA: 1.5 segundos.
```



Notamos que con la vista creada llamada PRODUCTOSVENDIDOSXSEMANA se redujo el tiempo de respuesta de gran manera.

- Caso 3: Proveedor más solicitado en cada semana del año

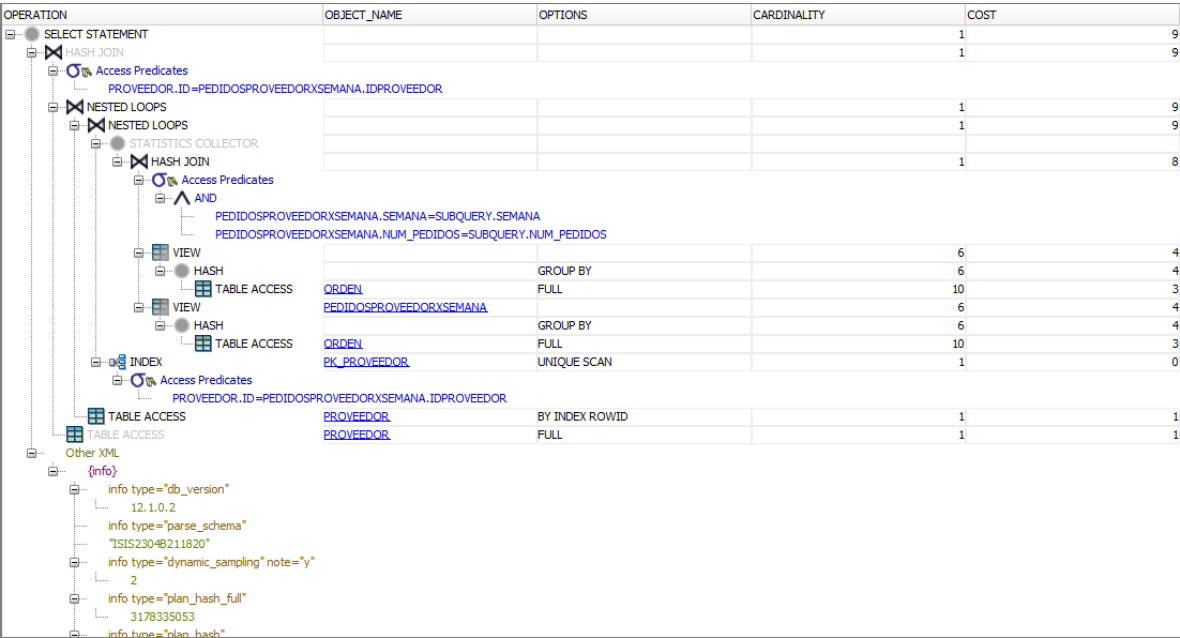
```
SELECT PEDIDOSPROVEEDORXSEMANA.SEMANA, PEDIDOSPROVEEDORXSEMANA.IDPROVEEDOR ,
PROVEEDOR.NOMBRE, PEDIDOSPROVEEDORXSEMANA.NUM_PEDIDOS
FROM
(SELECT SEMANA, MAX(NUM_PEDIDOS) AS NUM_PEDIDOS
FROM PEDIDOSPROVEEDORXSEMANA
GROUP BY SEMANA, IDPROVEEDOR) SUBQUERY
```

```
INNER JOIN PEDIDOSPROVEEDORXSEMANA ON PEDIDOSPROVEEDORXSEMANA.SEMANA =
SUBQUERY.SEMANA AND PEDIDOSPROVEEDORXSEMANA.NUM_PEDIDOS =
SUBQUERY.NUM_PEDIDOS
INNER JOIN PROVEEDOR ON PROVEEDOR.ID = PEDIDOSPROVEEDORXSEMANA.IDPROVEEDOR;

-- TABLAS ACCEDIDAS

-- CARDINALIDAD PROVEEDOR: 1.005
-- CARDINALIDAD PROVEE: 1
-- CARDINALIDAD ORDEN: 11

-- TIEMPO DE RESPUESTA: 0,032 segundos.
```



Notamos que con la vista creada llamada PEDIDOSPROVEEDORXSEMANA se redujo el tiempo de respuesta de gran manera.

- Caso 4: Proveedor menos solicitado en cada semana del año

```
SELECT PEDIDOSPROVEEDORXSEMANA.SEMANA, PEDIDOSPROVEEDORXSEMANA.IDPROVEEDOR ,
PROVEEDOR.NOMBRE, PEDIDOSPROVEEDORXSEMANA.NUM_PEDIDOS
FROM
    (SELECT SEMANA, MIN(NUM_PEDIDOS) AS NUM_PEDIDOS
    FROM PEDIDOSPROVEEDORXSEMANA
    GROUP BY SEMANA, IDPROVEEDOR) SUBQUERY
INNER JOIN PEDIDOSPROVEEDORXSEMANA ON PEDIDOSPROVEEDORXSEMANA.SEMANA =
SUBQUERY.SEMANA AND PEDIDOSPROVEEDORXSEMANA.NUM_PEDIDOS =
SUBQUERY.NUM_PEDIDOS
INNER JOIN PROVEEDOR ON PROVEEDOR.ID = PEDIDOSPROVEEDORXSEMANA.IDPROVEEDOR;

-- TABLAS ACCEDIDAS

-- CARDINALIDAD PROVEEDOR: 1.005
-- CARDINALIDAD PROVEE: 1
-- CARDINALIDAD ORDEN: 11

-- TIEMPO DE RESPUESTA: 0,032 segundos.
```



OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT				9
HASH JOIN			1	9
Access Predicates				9
PROVEEDOR.ID=PEDIDOSPROVEEDORXSEMANA.IDPROVEEDOR				
NESTED LOOPS			1	9
NESTED LOOPS			1	9
STATISTICS COLLECTOR				
HASH JOIN			1	8
Access Predicates				
AND				
PEDIDOSPROVEEDORXSEMANA.SEMANA=SUBQUERY.SEMANA				
PEDIDOSPROVEEDORXSEMANA.NUM_PEDIDOS=SUBQUERY.NUM_PEDIDOS				
VIEW				6
HASH		GROUP BY		4
TABLE ACCESS	ORDEN	FULL		6
VIEW			10	3
HASH	PEDIDOSPROVEEDORXSEMANA		6	4
TABLE ACCESS	ORDEN	GROUP BY		6
HASH		FULL		4
TABLE ACCESS	ORDEN		10	3
INDEX	PK_PROVEEDOR	UNIQUE SCAN	1	0
Access Predicates				
PROVEEDOR.ID=PEDIDOSPROVEEDORXSEMANA.IDPROVEEDOR				
TABLE ACCESS	PROVEEDOR	BY INDEX ROWID		1
TABLE ACCESS	PROVEEDOR	FULL	1	1
Other XML				
{info}				
info type="db_version"				
12.1.0.2				
info type="parse_schema"				
"ISIS2304B211820"				
info type="dynamic_sampling" note="y"				
2				
info type="plan_hash_full"				
3178335053				
info type="plan_hash"				

Notamos que con la vista creada llamada PEDIDOSPROVEEDORXSEMANA se redujo el tiempo de respuesta de gran manera.

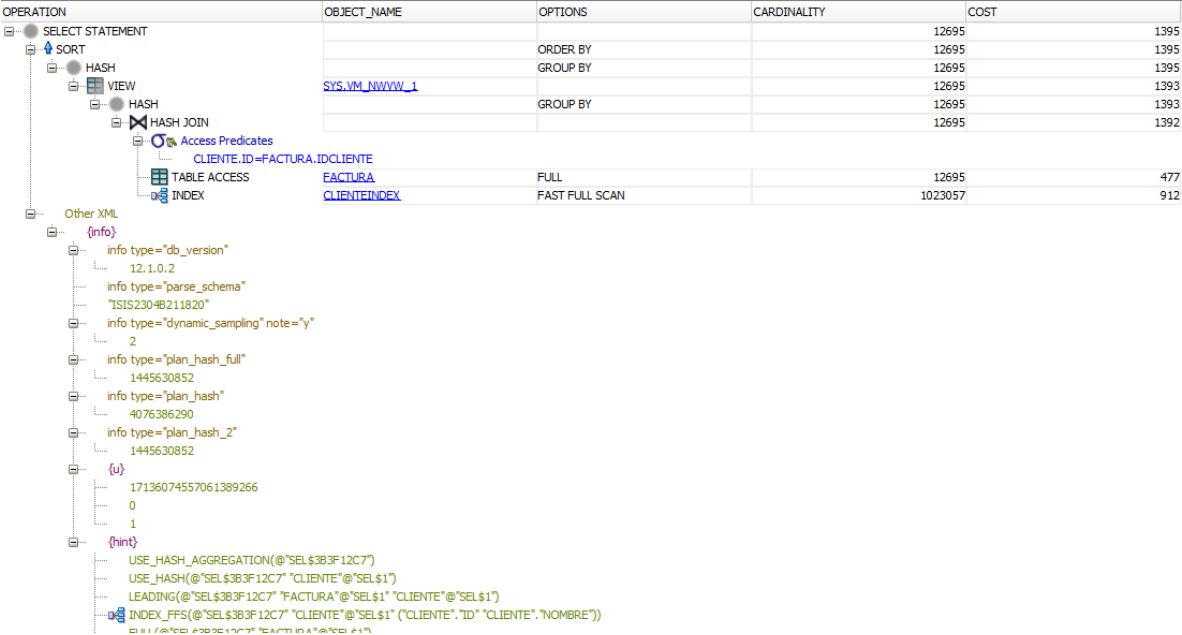
- Consultar los buenos clientes
- Caso 1: Clientes que hicieron al menos una compra por mes

```
SELECT CLIENTE.ID, CLIENTE.NOMBRE, EXTRACT(MONTH FROM FACTURA.FECHA) AS MES, COUNT(DISTINCT FACTURA.ID) AS COMPRAS
FROM CLIENTE
INNER JOIN FACTURA ON CLIENTE.ID = FACTURA.IDCLIENTE
GROUP BY CLIENTE.ID, CLIENTE.NOMBRE,EXTRACT(MONTH FROM FACTURA.FECHA)
ORDER BY COUNT(DISTINCT FACTURA.ID) DESC,CLIENTE.NOMBRE ASC;

-- TABLAS ACCEDIDAS

-- CARDINALIDAD CLIENTE: 1'023.057
-- CARDINALIDAD FACTURA: 270.163

-- TIEMPO DE RESPUESTA: 2 segundos.
```



Revisando el plan de Oracle, notamos que la creación del índice CLIENTEINDEX redujo el costo de la operación y el tiempo de respuesta.

NUEVO TIEMPO DE RESPUESTA: 1,7 segundos

- Caso 2: Clientes que hicieron al menos una compra por mes

```
SELECT DISTINCT CLIENTE.ID, CLIENTE.NOMBRE
FROM CLIENTE
INNER JOIN FACTURA ON CLIENTE.ID = FACTURA.IDCLIENTE
INNER JOIN FACTURAPRODUCTO ON FACTURA.ID = FACTURAPRODUCTO.IDFACTURA
INNER JOIN VENDE ON VENDE.IDPRODUCTO = FACTURAPRODUCTO.IDPRODUCTO
WHERE VENDE.PRECIO > 100000;

-- TABLAS ACCEDIDAS

-- CARDINALIDAD CLIENTE: 1'023.057
-- CARDINALIDAD FACTURA: 270.163
-- CARDINALIDAD FACTURAPRODUCTO: 257.559

-- TIEMPO DE RESPUESTA: 0,064 segundos.
```

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT				207
HASH		UNIQUE		207
HASH JOIN				206
Access Predicates	CLIENTE.ID=FACTURA.IDCLIENTE			
NESTED LOOPS				206
NESTED LOOPS				206
STATISTICS COLLECTOR				
HASH JOIN				70
Access Predicates	FACTURA.ID=FACTURAPRODUCTO.IDFACTURA			
NESTED LOOPS				70
STATISTICS COLLECTOR				
HASH JOIN				2
Access Predicates	VENDE.IDPRODUCTO=FACTURAPRODUCTO.IDPRODUCTO			
NESTED LOOPS				2
STATISTICS COLLECTOR				
INDEX	VENDE_PK	SKIP SCAN		1
Access Predicates	VENDE.PRECIO>3500			
Filter Predicates	VENDE.PRECIO>3500			
INDEX	FACTURAPRODUCTO_PK	FULL SCAN		1
Access Predicates	VENDE.IDPRODUCTO=FACTURAPRODUCTO.IDPRODUCTO			
Filter Predicates	VENDE.IDPRODUCTO=FACTURAPRODUCTO.IDPRODUCTO			
INDEX	FACTURAPRODUCTO_PK	FULL SCAN		1
TABLE ACCESS	FACTURA	BY INDEX ROWID		1
INDEX	FACTURA_PK	UNIQUE SCAN		0
Access Predicates	FACTURA.ID=FACTURAPRODUCTO.IDFACTURA			
TABLE ACCESS	FACTURA	FULL		1

▪ Caso 3: Clientes que han realizado una compra de un producto de ‘Electronics’ y ‘Tools’

```
SELECT DISTINCT CLIENTE.ID, CLIENTE.NOMBRE
FROM CLIENTE
INNER JOIN FACTURA ON CLIENTE.ID = FACTURA.IDCLIENTE
INNER JOIN FACTURAPRODUCTO ON FACTURA.ID = FACTURAPRODUCTO.IDFACTURA
INNER JOIN PRODUCTO ON PRODUCTO.ID = FACTURAPRODUCTO.IDPRODUCTO
INNER JOIN TIPOPRODUCTO ON TIPOPRODUCTO.ID = PRODUCTO.IDTIPOPRODUCTO
WHERE TIPOPRODUCTO.NOMBRE = 'Electronics' or TIPOPRODUCTO.NOMBRE = 'Tools';

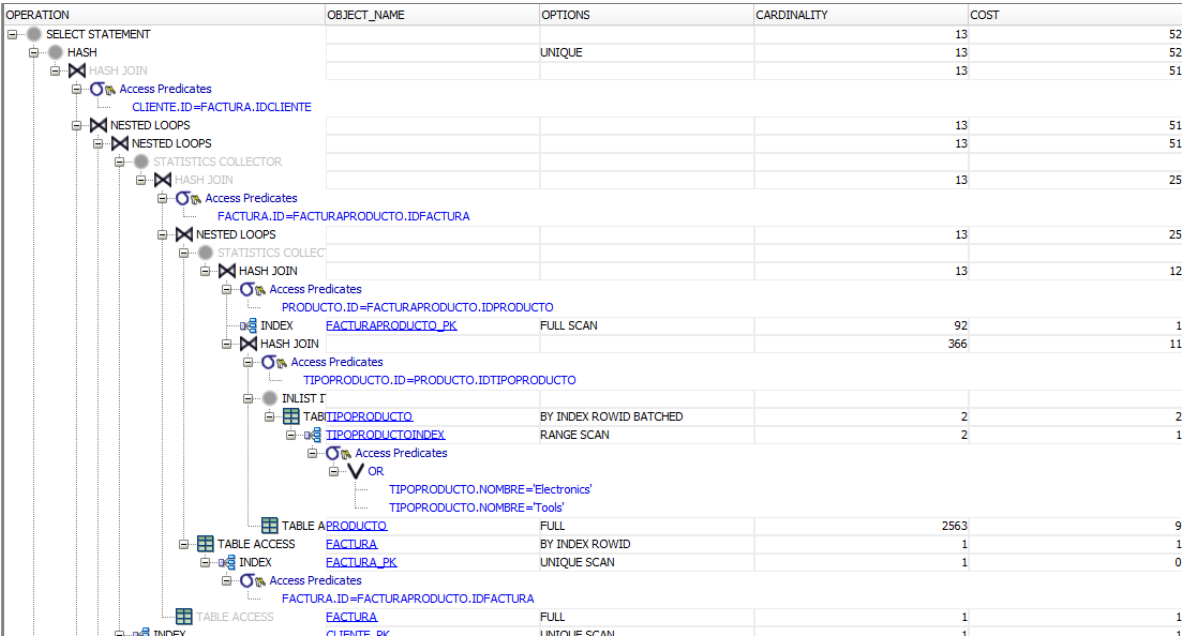
-- CARDINALIDAD CLIENTE: 1'023.057
-- CARDINALIDAD PRODUCTO: 2.563
-- CARDINALIDAD FACTURA: 270.163
-- CARDINALIDAD FACTURAPRODUCTO: 257.559

-- TIEMPO DE RESPUESTA: 1 segundo.
```

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT				53
HASH		UNIQUE		53
HASH JOIN				52
Access Predicates	CLIENTE.ID=FACTURA.IDCLIENTE			
NESTED LOOPS				52
NESTED LOOPS				52
STATISTICS COLLECTOR				
HASH JOIN				26
Access Predicates	FACTURA.ID=FACTURAPRODUCTO.IDFACTURA			
NESTED LOOPS				26
STATISTICS COLLECTOR				
HASH JOIN				13
Access Predicates	PRODUCTO.ID=FACTURAPRODUCTO.IDPRODUCTO			
INDEX	FACTURAPRODUCTO_PK	FULL SCAN		1
HASH JOIN				12
Access Predicates	TIPOPRODUCTO.ID=PRODUCTO.IDTIPOPRODUCTO			
TABLE ACCESS	TIPOPRODUCTO	FULL		3
Filter Predicates	TIPOPRODUCTO.NOMBRE='Electronics' TIPOPRODUCTO.NOMBRE='Tools'			
TABLE ACCESS	PRODUCTO	FULL		9
TABLE ACCESS	FACTURA	BY INDEX ROWID		1
INDEX	FACTURA_PK	UNIQUE SCAN		0
Access Predicates	FACTURA.ID=FACTURAPRODUCTO.IDFACTURA			
TABLE ACCESS	FACTURA	FULL		1
INDEX	CLIENTE_PK	UNIQUE SCAN		1
Access Predicates	CLIENTE.ID=FACTURA.IDCLIENTE			

Analizando la operación de consulta, decidimos crear un índice simple denominado TIPOPRODUCTOINDEX en el campo NOMBRE de TIPOPRODUCTO. Lo anterior debido a que la consulta filtra este campo, por lo cual un índice podría filtrar más rápidamente.

```
CREATE INDEX CLIENTEINDEX ON CLIENTE (ID, NOMBRE)
```



Revisando de nuevo el plan de consulta después de haber creado el índice simple TIPOPRODUCTOINDEX, vemos que el costo disminuyó.

```
NUEVO TIEMPO DE RESPUESTA: 0,044 segundos
```