



Gestión de Datos

Estrategias

GRUPO N°: 33

Nombre del grupo: NAPOLITANA_CON_FRITAS

INTEGRANTES

Carolina Muscari - 1526509

Nahuel Pascual - 1638968

Pedro Bruno - 1380862

Santiago Camargo - 1639079

Año 2020

Modelo de Datos	2
Modelo actual	2
Otras opciones	3
Denormalización	3
Parametrización	3
Migración	4
Tabla Cliente	4
Tabla Sucursal	4
Tabla Tipo_Auto	4
Tabla Tipo_Transmisión	4
Tabla Tipo_Caja	4
Tabla Fabricante	4
Tabla Modelo	4
Tabla Auto	4
Tabla Autoparte	4
Compras	5
Ventas	5

Modelo de Datos

Modelo actual

Las entidades centrales del modelos son *Auto* y *Autoparte*, por ser los productos que comercializa el sistema en cuestión. Ambas tienen relación con un *Modelo*, el cual a su vez tendrá referencias hacia *Fabricante*, *Tipo_Transmision*, *Tipo_Auto* y *Tipo_Caja*.

Aclaración: "tipo_motor" queda simplemente como un atributo de Modelo, ya que al no tener descripción en la "tabla Maestra" no tenía sentido llevar el dato a una tabla paramétrica.

Luego, para las operaciones de compra-venta existen las tablas *Compra* y *Factura* respectivamente. Cada una de ella es referenciada por sus respectivas tablas de items, permitiendo modelar la relación muchos a uno desde los productos (auto y autoparte) hacia las operaciones (compra y factura).

Tanto los "item factura" como los "item compra" se splittean en dos tablas diferentes, para referencia a auto y autoparte según corresponda, ya que al ser entidades diferentes necesitamos un campo único para poder generar la **Foreign Key** desde item hacia el producto.

En resumen, tendríamos un total de 4 tablas "item":

- Compra
 - Item_Compra_Auto
 - Auto
 - Item_Compra_Autoparte
 - Autoparte
- Factura
 - Item_Factura_Auto
 - Auto
 - Item_Factura_Autoparte
 - Autoparte

Tanto *Factura* como *Compra* tienen referencias hacia *Cliente* y *Sucursal*, indicando el cliente (vendedor o comprador, respectivamente) y la sucursal donde se efectuó la operación.

El DER correspondiente al modelo explicado se encuentra adjunto junto con el presente documento.

Otras opciones

Denormalización

En la búsqueda de evitar “duplicar” las tablas *Compra* y *Factura* (que son exactamente iguales en su estructura) y las tablas “ítem” anteriormente explicadas, estudiamos la posibilidad de denormalizarlas en una entidad *Operación* que, contando con un discriminador de tipo, pudiera ser indistintamente una Factura o una Compra.

A su vez, también existía la posibilidad de aplicar la misma estrategia en *Item* para así tener una única tabla que pudiera, indistintamente, referenciar Autos o Autopartes. En este caso, la complicación adicional se debía al ya mencionado problema con la FK; en caso de optar por esta estrategia, la solución hubiera sido quitar la FK y crear un trigger que pudiera garantizar la consistencia en su lugar.

Parametrización

Otra cosa que podríamos haber hecho para atacar el problema de las compras y ventas hubiera sido tener una tabla *Producto* que tuviera los datos comunes de los autos y autopartes, un discriminador de tipo y una tabla paramétrica con los *Atributos*.

Ejemplo:

Producto_Atributos			
Producto_codigo	Producto_Tipo	Atributo_Key	Atributo_Value
123	1	motor	23JK123KJ4G1
321	1	patente	AA2342FF

Esto nos hubiera permitido tener sólo la tabla *Compras*, con su relación 1 a N hacia *Item_Compra*, y la tabla *Factura* con la misma relación hacia *Item_Factura*.

Estas 3 estrategias opcionales podrían haberse aplicado en cualquier combinación posible, logrando así diferentes modelados en cada combinación posible.

Migración

Tabla Cliente

Hicimos el **INSERT** a partir de un **UNION** sobre dos **SELECT**, cada uno buscando clientes de las Facturas y Compras. Filtramos los registros con cliente nulo y, a su vez, el **UNION** nos garantiza no tener repetidos.

Aclaración: debido a la mala organización previa del sistema y la nula normalización del modelo de datos, resulta probable que tuviéramos datos incorrectos o clientes duplicados con datos mal cargados o distinta dirección o mail.

Tabla Sucursal

Nuevamente realizamos el **INSERT** a partir de un **SELECT**, filtrando las rows con sucursal nula.

En este caso debimos procesar el mail para obtener un *ID de sucursal* ya que, evidentemente, el negocio ya tenía estos códigos asignados a pesar de que no estuvieran plasmados en nuestro modelo de datos.

Tabla Tipo_Auto

Realizamos el **INSERT** a partir de un **SELECT**, filtrando las rows con dato nulo.

Tabla Tipo_Transmisión

Realizamos el **INSERT** a partir de un **SELECT**, filtrando las rows con dato nulo.

Tabla Tipo_Caja

Realizamos el **INSERT** a partir de un **SELECT**, filtrando las rows con dato nulo.

Tabla Fabricante

Realizamos el **INSERT** a partir de un **SELECT**, filtrando las rows con dato nulo.

Tabla Modelo

Realizamos el **INSERT** a partir de un **SELECT**, filtrando las rows con dato nulo.

También tuvimos que realizar un **JOIN** contra la anteriormente poblada *Fabricante* para, a partir del “nombre_fabricante” de la tabla Maestra, obtener el ID.

Tabla Auto

Realizamos el **INSERT** a partir de un **SELECT**, filtrando las rows con dato nulo.

Tabla Autoparte

Realizamos el **INSERT** a partir de un **SELECT**, filtrando las rows con dato nulo.

Compras

En este caso generamos una *tabla temporal* con la información de clientes, sucursal y productos (auto y autoparte) relacionados a compras y, obviamente, también la propia información de la compra. La información se obtiene filtrando sólo las rows en que los datos de la factura sean nulos (eso significa que no se trata de una venta entonces, por descarte, se trata de una compra); la condición opuesta no resultaba útil ya que el dato de la compra se duplicaba al registrar la venta.

Esta *tabla temporal* debería reducir el total de rows sobre las que haremos los subsiguientes **SELECT** para poblar las tablas *Compra*, *Item_Compra_Auto* y *Item_Compra_Autoparte*.

El **INSERT** en *Compra* se hace a partir de un **SELECT** en el que agrupamos por cliente, factura y sucursal. Usamos un **SUM** para calcular el precio total de la compra y **JOIN** contra *Cliente* para obtener el ID a partir de los datos de cliente cargados en el modelo viejo.

El **INSERT** en *Compra_Item_Auto* se hace a partir de un **SELECT** que procura obtener sólo las rows en que los datos de auto no sean nulos. Se hace un **SUM** de la cantidad vendida por si se diera el caso de que en una misma compra se hubiera cargado dos veces el mismo producto y un **ISNULL** ya que en el caso particular de los autos, la cantidad no era un dato con el que contáramos en la mayoría de los casos.

El **INSERT** en *Compra_Item_Autoparte* se hace a partir de un **SELECT** que cumple que los datos de autoparte no sean nulos y, al igual que con los autos, se hace un **SUM** de la cantidad por eventuales errores de carga en los datos existentes.

Ventas

Para poblar la tabla *Factura* se hace un **INSERT** a partir de un **SELECT** sobre “tabla Maestra”, que trae datos propios de la factura, la sucursal y el ID de cliente **JOINEANDO** contra la tabla *Cliente*.

Se deja en **NULL** el precio de la factura, para actualizarlo luego.

El **INSERT** en *Item_Factura_Auto* se realiza a partir de un **SELECT** que obtiene la información de la tabla Maestra, **JOINEANDO** contra *Auto* para obtener el ID.

El **INSERT** en *Item_Factura_Autoparte* se hace con un **SELECT** sobre tabla Maestra que trae los datos requeridos.

Por último, para calcular y actualizar los valores totales de las facturas, se crea una *tabla temporal* a partir de **SELECTS** contra *Item_Factura_Auto* y *Item_Factura_Autoparte*.

Luego se hace un **UPDATE** sobre *Factura*, calculando los totales a partir de la información almacenada la *tabla* recién construída.

MODELO BI

El modelo se compone por 4 (cuatro) tablas de hecho:

- BI_VENTA_AUTO
- BI_COMPRA_AUTO
- BI_VENTA_AUTOPARTE
- BI_COMPRA_AUTOPARTE

A su vez existen tablas referentes a las dimensiones en base a las cuales nos interesa organizar los datos.

Migración desde el modelo transaccional

Algunas tablas de dimensión responden a la lógica de rangos o grupos, por lo cual se convierten en tablas paramétricas con segmentaciones. Cada uno de estos rangos tendrá un ID para poder ser referenciado desde las tablas de hechos.

Estas son:

- BI_POTENCIA_RANGO (rangos de potencia del automóvil)
- BI_CLIENTE (rangos de edad)
- BI_CANTIDAD_CAMBIOS (cantidad de cambios de automóvil)
- BI_RUBRO_AUTOPARTE (este dato no existe actualmente en el modelo, por lo cual el único valor posible será “sin datos”)

Por otro lado, hay también varias tablas dimensionales cuyos datos se desprenden directamente de las entidades del modelo transaccional. Su migración consiste simplemente en insertar datos seleccionados desde las otras.

Estas son:

- BI_TIPO_MOTOR
- BI_FABRICANTE
- BI_TIPO_AUTO
- BI_TIPO_CAJA
- BI_TIPO_TRANSMISION
- BI_MODELO
- BI_SUCURSAL
- BI_AUTOPARTE
- BI_MES¹

¹ En el caso particular de BI_MES vale mencionar que se le asigna un ID autoincremental a cada mes en el que tengamos datos. Así, por ejemplo, Enero 2018 será el MES 1, Febrero 2018 será el MES 2, Enero 2019 será el MES 13, etc.

Por último, en cuanto a las tablas de hechos:

En este caso el procesamiento de la información ya es un poco más complejo y se aplican diferentes estrategias.

BI_COMPRA_AUTOPARTE: se crean las tablas temporales #ClienteRangoTemp² y #StockSucursalTemp. La primera determina a qué rango de edad pertenece el cliente, luego de calcular su edad a partir de la fecha de nacimiento y la fecha actual. La segunda determina la variación de stock para cada autoparte, en cada sucursal, por cada mes; este dato se utiliza para calcular el máximo stock que tuvo una sucursal.

Los datos se agrupan por las dimensiones, para acumular las cantidades compradas y el precio total de venta. Luego podrá seguir sumariéndose en función de las dimensiones por las que se pretenda filtrar.

BI_VENTA_AUTOPARTE: igual que en el caso anterior, se usa #ClienteRangoTemp y se suman los valores para reducir la cantidad de registros y facilitar las consultas posteriores.

BI_COMPRA_AUTO: nuevamente la inserción de datos es bastante directa, a excepción de la determinación del rango de potencia que fue resuelto mediante una sentencia CASE. El tipo de motor se obtiene a partir de una subquery. También se usa #ClienteRangoTemp.

BI_VENTA_AUTO: nuevamente usamos #ClienteRangoTemp. Al igual que en el caso anterior, se usa el CASE para la potencia y una subconsulta para el motor. También en esta oportunidad hacemos uso de la tabla temporal #TiempoStockTemp que previamente calculó para cada automóvil la diferencia entre la fecha de venta y la fecha de compra, lo que nos dice el tiempo que estuvo en stock; teniendo esto ya precalculado, ahora sólo necesitamos agrupar por modelo y calcular el promedio de tiempo en stock.

Vistas

VW_AUTO_CANTIDAD_GANANCIA: esta vista muestra la cantidad de automóviles comprados y vendidos, así como las ganancias provenientes de dichas transacciones, por cada sucursal en el lapso de un mes.

VW_AUTOPARTE_GANANCIA: esta vista muestra, para cada sucursal en cada mes, las ganancias por venta de autopartes; se calcula como precios de venta menos precios de compra.

VW_AUTOPARTE_PRECIO_PROMEDIO: esta vista muestra los precios promedio de compra y de venta, para cada autoparte.

VW_AUTOPARTE_MAXIMO_STOCK: haciendo uso de la columna "stock" que se agregó en la tabla de hechos BI_COMPRA_AUTOPARTE, podemos ver qué mes hubo mayor stock. El filtro se aplica a nivel anual, seleccionando el mes de mayor stock dentro del año.

² Esta misma Temporal se usa en todas las demás tablas de hechos, por lo que no volveremos a mencionarla.