

Trabajo Práctico de Gestión de Datos

Estrategia

Grupo LOS_GEDDES						
Cátedra	Ing. en Sistemas de Información	Nombre	Apellido	Legajo	Mail	Curso
Integrantes del Grupo:		Jean Pierre	Chero Pomaleque	152.233-4	pierchero@gmail.com	K3671
		Matías Leandro	Anzorandía	168.063-8	matias.l.anzorandia@gmail.com	K3572
		Emiliano	González	135.246-5	gonzalemi42@gmail.com	K3572

Fecha de Presentación: 25/10/2020

Calificación: _____

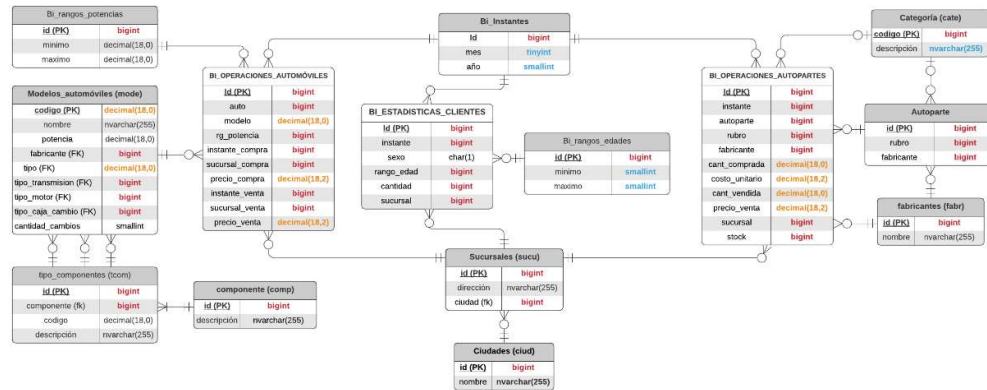
Fecha Devolución: _____

Firma Docente: _____

Índice

- Diagrama de Entidad-Relación Página 02
- Entrega 2 - Estrategia – Modelo de datos Página 03
- Entrega 2 - Estrategia – Performance Página 04
- Entrega 3 - Business Intelligence - Estructuras Página 05
- Entrega 3 - Business Intelligence – Decisiones de Diseño Página 06
- Entrega 3 - Business Intelligence – Vistas Página 07

Diagrama de Entidad-Relación



ESTRATEGIA

ACERCA DEL MODELO DE DATOS

1. **Tablas 'Tipo_componentes' y 'Componentes':** considerando que existe un mismo esquema de definición para los tipos de transmisión, caja de cambios y motor dado por su tipo y descripción decidimos por un lado crear la tabla paramétrica 'Componentes' para identificar cada uno estos tipos (por ej.: 'TRANSMISION', 'CAJA' y 'MOTOR') y por otro lado almacenar los códigos y descripciones en una segunda tabla llamada 'Tipo_componentes' la cual a través de una FK a la tabla 'Componentes' nos permite discriminarlos. El objetivo que perseguimos con esta decisión fue evitar crear distintas tablas con estructura y semántica repetida.
Finalmente, la tabla 'Modelo_automoviles' que representa la información asociada a cada modelo de automóvil es la que hace uso de la información de los distintos tipos y para ello posee distintas relaciones con la tabla 'Tipo_componentes'.
2. **Restricciones de Datos:** durante la migración buscamos asegurar la integridad de los datos, para ello priorizamos crear campos NOT NULL en todos los casos donde vimos que fue posible. De esta manera evitamos incurrir en un posible error de inconsistencia de datos frente a un eventual error en el script de migración de datos
3. **Tablas Items:** buscando un modelo normalizado decidimos crear las tablas Item_por_factura e Item_por_compra que modelan los elementos asociados, siguiendo una estrategia cabecera-detalle.
4. **Relación Cliente – Compra:** luego de validar el modelo de datos con nuestros ayudantes decidimos agregar una relación entre Cliente y Compra para tener registro del Cliente que hizo la compra a través de la sucursal.
5. **Uso de Stored Procedures:** buscando una alta cohesión en nuestro script de migración utilizamos distintos Stored Procedures para agrupar y encapsular la lógica migración de las distintas tablas. Esto también nos permite asegurar la consistencia de los datos.
Por último, realizamos el dropeo de los mismos luego de finalizar la migración ya que no van a necesitar utilizarse en futuras ejecuciones.

PERFORMANCE Y OPTIMIZACIÓN

1. **Uso de Union All:** para mejorar el rendimiento de la migración de las tablas Clientes, Facturas, Tipo_componentes las cuales utilizan subselects, utilizamos la instrucción UNION ALL para la unión de dichos subselects la cuál es más rápida que la instrucción UNION dado que no discrimina datos repetidos. Esta decisión la tomamos luego de validar que el conjunto de datos asociado a cada subselect es excluyente.
2. **Acerca de las Constraints:** realizamos la declaración de las constraints al final de las creaciones de las tablas a través de la instrucción ALTER TABLE para poder agilizar el insert de los datos.
3. **JOINS:** Identificamos los campos únicos de las tablas para poder realizar la menor comparación en los JOINS. Esto nos evita estar matcheando muchos campos para poder identificar el mismo dato.
4. **Claves subrogadas:** para simplificar la relación entre las distintas tablas decidimos utilizar claves subrogadas de tipo BIGINT, esto nos permitió por ejemplo en la tabla Clientes evitar usar claves compuestas teniendo en cuenta que el DNI se repetía. Otras tablas con este tipo de claves son Automóvil, Cliente, Fabricantes, Tipo_Componentes, Sucursales, Ciudades y Categoría.
5. **Tablas Temporales:** Utilizamos tablas temporales para mejorar el rendimiento, esto puede observarse en la migración de los datos asociados a las facturas e ítems donde realizamos una única consulta a la tabla Maestra para luego operar con el resultado de esa búsqueda.
6. **Execution Plan:** en diferentes oportunidades revisamos el plan de ejecución para poder medir y mejorar el costo de las queries.

BUSINES INTELLIGENCE

BUSINESS INTELLIGENCE - ESTRUCTURAS

Tablas de hechos:

BI_Estadisticas_Clientes: contiene información sobre la cantidad de clientes existentes según rango de edad, sucursal donde realizaron operaciones

BI_Operaciones_automoviles: contiene información de la cantidad de operaciones de compra y venta de automóviles, modelo del auto, sucursal y precios de compra y venta respectivamente. Se utiliza para calcular la ganancias, promedios de precio y promedio de tiempo en stock.

BI_Operaciones_autopartes: contiene información de la cantidad de distintas operaciones de autopartes similar a automóviles con el agregado del calculo de stock, fabricante y rubro. Se utiliza para calcular la ganancias.

BI_Stock_Maximo: contiene información sobre la cantidad stock por sucursal, autoparte e instante de tiempo.

Dimensiones:

BI_Instantes: un instante es un conjunto formado por un mes y un año. Esta tabla nos permite manejar la dimensión del tiempo.

BI_rangos_edades: esta tabla contiene los distintos rangos de edades que nos interesan analizar

BI_rangos_potencias: esta tabla contiene los distintos rangos de motores que nos interesan analizar

BUSINESS INTELLIGENCE - DECISIONES DE DISEÑO

1. Comenzamos utilizando la información almacenada en las tablas de modelo creadas durante la entrega dos, teniendo modelo, cantidad comprada, vendida y precios por instante y sucursal, lo cual nos dimos cuenta de que no fue lo mejor para las vistas, sumado a que nos percatamos que la sucursal que compra no siempre es la misma que vende.
En un principio optamos por la estrategia de herencia single table, juntando compras y ventas en una misma tabla con un campo discriminador, de modo que la tabla de operaciones automóviles paso a contener las operaciones individuales para guardar el registro de la entrada y la salida de cada uno.
Finalmente decidimos almacenar la información de compra y venta del automóvil en la misma fila, lo cual si bien dificulto un poco la migración (debiendo hacer updates), simplifico considerablemente las vistas, reduciendo las decisiones y varios joins.
2. Para las autopartes, guardamos algunos datos individuales tales como el fabricante (si bien vimos que es único, consideramos que no necesariamente será así en el futuro)
3. Tomamos la decisión de registrar ganancias negativas para aquellas sucursales que realizaron compras pero no registraron ventas.

BUSINESS INTELLIGENCE - VISTAS

Compraventa_mensual_automoviles: muestra la cantidad comprada y vendida de automóviles por cada instante de tiempo (año y mes) y una sucursal

Precio_promedio_automoviles: muestra el precio promedio de compra y venta de un automóvil por cada instante de tiempo (año y mes) y una sucursal

Ganancias_mensuales_automoviles: muestra la ganancia (Calculada como la diferencia entre ventas y compras) de cada modelo de automóvil por cada instante de tiempo (año y mes) y una sucursal

Tiempo_promedio_en_stock_automoviles: muestra el promedio en stock de un modelo de automóvil

ganancias_mensuales_autopartes: muestra la ganancia (Calculada como la diferencia entre ventas y compras) de cada modelo de automóvil por cada instante de tiempo (año y mes) y una sucursal

Maxima_cantidad_stock_por_sucursal: muestra el stock máximo de una sucursal por año.

Precio_promedio_autopartes: muestra el precio promedio de compra y venta de un autopartes por cada instante de tiempo (año y mes) y una sucursal

Promedio de tiempo en stock de autopartes: esta vista la excluimos del alcance del TP como nos indicaron los ayudantes por el grupo de e-mail.