



Universidad Tecnológica Nacional
Facultad Regional Buenos Aires

Gestión de Datos

Trabajo Práctico
2° Cuatrimestre 2023

FRBA - Inmobiliaria

Grupo SYSTEAM

Apellido, Nombre	Legajo
Benito, Ariel	1633168
Delgado, Marco	159120
Harutyunyan, Elena	1715800
Vautier, Germán	1535018

Modelo Relacional:

VENTA: Almacena información sobre las ventas de los inmuebles. Le añadimos todos los atributos que indicaba el enunciado teniendo VENTA_CÓDIGO como PK. El enunciado también indicaba la existencia de un pago de la venta y la moneda del precio de venta que optamos por detallarlas en tablas separadas. En cuanto al procedimiento almacenado, insertamos los datos de la tabla maestra utilizando joins hacia las tablas “Moneda” y “Anuncio” para referenciar los datos y al final se añadió la fila que asegura seleccionar solo los registros con valores no nulos de la columna VENTA_CODIGO de la tabla maestra.

PAGO_VENTA: Registra los pagos asociados a las ventas. Le añadimos todos los atributos que indicaba el enunciado teniendo PAGO_VENTA_CODIGO como PK que es un atributo que no existía en la tabla maestra y que añadimos para identificar cada pago. En este caso también detallamos la moneda en una tabla separada ya que es un atributo que se repite en tres tablas distintas. Existe el atributo Moneda en la tabla “Pago Venta” y también en la de “Venta” porque el precio de la venta puede estar definido en una moneda distinta a la que se hace el pago; si no existiera esta posibilidad, la Moneda estaría solamente en la tabla Venta. También decidimos normalizar el medio de pago en una tabla separada. En cuanto al procedimiento almacenado, insertamos los datos de la tabla maestra utilizando joins hacia las tablas “Moneda”, “Medio_Pago” y “Venta” para referenciar los datos y al final se añadió la fila que asegura seleccionar solo registros con valores no nulos de la columna PAGO_VENTA_IMPORTE de la tabla maestra.

MEDIO_PAGO: Describe los diferentes métodos de pago disponibles. Como son varios y también se utilizan en, por lo menos, dos tablas distintas (PAGO_ALQUILER y PAGO_VENTA), decidimos normalizarlos en una tabla separada y le añadimos una PK autoincremental que es MEDIO_PAGO_CODIGO para identificar cada medio de pago. Para insertar los datos, como en la tabla maestra existen las columnas PAGO_VENTA_MEDIO_PAGO y PAGO_ALQUILER_MEDIO_PAGO, optamos por utilizar la consulta UNION para traer los registros de ambas columnas y a eso hacerle un SELECT DISTINCT para que no recupere datos repetidos y que cada ID tenga su propio medio de pago.

MONEDA: Creamos una tabla para los tipos de moneda (USD, ARS) con una PK autoincremental de manera tal que al cargar los datos de cada tipo de moneda, cada uno tenga su ID y cada nuevo tipo de moneda ingresado automáticamente reciba una ID única para esta tabla. Para la migra, pudimos haber utilizado la misma estrategia que MEDIO_PAGO (uniendo los registros que existen en las tres columnas de la tabla maestra en las que se utilizan los valores correspondientes a la tabla MONEDA) pero como hay solo dos registros existentes para este atributo, decidimos insertarlos manualmente. Sin embargo, reconocemos que esta práctica puede volverse ineficiente a largo plazo y propensa a errores al insertar múltiples tipos de moneda.

ANUNCIO: Guarda información sobre los anuncios de las ventas y alquileres de los inmuebles. Le añadimos todos los atributos que indicaba el enunciado teniendo ANUNCIO_CÓDIGO como PK. La moneda del anuncio, como detallamos en el punto anterior, se normaliza en la tabla MONEDA. En cuanto al procedimiento almacenado,

insertamos los datos de la tabla maestra utilizando joins hacia las tablas “Moneda”, “Agente”, “Inmueble” y “Tipo Anuncio” para referenciar los datos y al final se añadió la fila que asegura seleccionar solo registros con valores no nulos de la columna ANUNCIO_CODIGO de la tabla maestra.

TIPO_ANUNCIO: Define los tipos de anuncios (alquiler, venta, etc.). La decisión de separar TIPO_ANUNCIO en una tabla aparte de ANUNCIO es para evitar datos repetidos, ya que varios anuncios comparten el mismo tipo. Para insertar los valores en la tabla optamos por seguir la misma estrategia que utilizamos para la tabla MONEDA detallada anteriormente e ingresamos los valores manualmente. TIPO_ANUNCIO también contiene la PK autoincremental para identificar cada tipo de anuncio.

AGENTE: Almacena detalles sobre los agentes inmobiliarios. Le añadimos todos los atributos que indicaba el enunciado teniendo AGENTE_CÓDIGO como PK y SUCURSAL como FK ya que un agente pertenece a una sucursal. En cuanto al procedimiento almacenado, insertamos los datos de la tabla maestra utilizando un join hacia la tabla “Sucursal” para referenciar los datos y al final se añadió la fila que asegura seleccionar solo los registros con valores no nulos de la columna AGENTE_DNI de la tabla maestra, ya que AGENTE_CODIGO es un atributo que definimos como autoincremental y no existe en la tabla maestra.

SUCURSAL: Migramos a esta tabla todos los registros relacionados a la sucursal que existían en la tabla maestra teniendo SUCURSAL_CÓDIGO como PK y LOCALIDAD como FK ya que una sucursal está definida dentro de una localidad y, a su vez, para insertar los datos utilizamos un JOIN hacia la tabla LOCALIDAD para referenciar los datos. Al final del procedure se añadió la fila que asegura seleccionar solo los registros con valores no nulos de la columna SUCURSAL_CÓDIGO de la tabla maestra.

PROVINCIA, LOCALIDAD, BARRIO: Estas tres tablas las relacionamos entre sí ya que un barrio está dentro de una localidad, que a su vez está dentro de una provincia. A cada una de las tres tablas se le asignó una PK autoincremental para facilitar la identificación de cada atributo ya que en la tabla maestra no existen estos IDs. Para migrar los datos se rescatan con un UNION las columnas de inmueble y sucursal de la tabla maestra correspondientes a las provincias, localidades y barrios.

INMUEBLE: Almacena detalles sobre los inmuebles disponibles para venta o alquiler. Le añadimos todos los atributos que indicaba el enunciado teniendo INMUEBLE_CÓDIGO como PK. En cuanto a los atributos INMUEBLE_CANT_AMBIENTES, INMUEBLE_DISPOSICION, INMUEBLE_TIPO, INMUEBLE_ORIENTACION, INMUEBLE_ESTADO decidimos normalizar cada uno de estos en una tabla aparte (teniendo cada uno una PK autoincremental para identificar cada uno de estos atributos) para evitar la repetición de estos datos en múltiples registros de la tabla principal de inmuebles. Para migrar la columna BARRIO de la tabla maestra a la tabla INMUEBLE, fue necesario hacer un JOIN para SUCURSAL y LOCALIDAD ya que hay muchos barrios con el mismo nombre y que son de diferentes sucursales y localidades.

INMUEBLE_CARACTERISTICA: Como en el enunciado se detalla que las características del inmueble están tipificadas, revisamos la tabla maestra y notamos que cada

característica tiene un valor de 1 para indicar que el inmueble posee esa característica. Entonces en este caso, tuvimos que crear esta tabla con PK autoincremental y asignarle a cada ID las características que encontramos en la tabla maestra como son wifi, cable, calefacción y gas, teniendo así cuatro identificadores en total. Después si se quiere añadir otra característica se va a tener que ingresar manualmente en la tabla INMUEBLE_CARACTERÍSTICA. Como los inmuebles pueden tener muchas características y a su vez una característica puede tener muchos inmuebles, optamos por crear una tabla intermedia CARACTERÍSTICA_POR_INMUEBLE que para la migra rescata solamente las características que en la tabla maestra tienen valor 1 y utiliza la cláusula UNION para cargar cada una de estas características, en este caso son cuatro pero si se quiere añadir otra característica se va a tener que añadir otra cláusula UNION y seleccionar la característica adicional.

PROPIETARIO, COMPRADOR, INQUILINO: Estas tablas se crearon de manera similar respetando los atributos marcados en el enunciado y en la tabla maestra. Añadimos la PK autoincremental correspondiente para cada una de estas tablas ya que decidimos no tomar el DNI de cada uno como identificador único porque el DNI es un valor que, con bajas probabilidades, se puede repetir. Sí lo utilizamos al final de cada procedure para asegurarnos de seleccionar solo los registros con valores no nulos de la tabla maestra.

ALQUILER: Le añadimos todos los atributos que indicaba el enunciado teniendo ALQUILER_CÓDIGO como PK. El enunciado también indicaba la existencia de un pago del alquiler que optamos por detallar en una tabla separada. En cuanto al procedimiento almacenado, insertamos los datos de la tabla maestra utilizando joins hacia las tablas “Inquilino” y “Anuncio” para referenciar los datos y al final se añadió la fila que asegura seleccionar solo los registros con valores no nulos de la columna ALQUILER_CODIGO de la tabla maestra.

DETALLE_ALQUILER: Podríamos haber mantenido estos atributos en la tabla ALQUILER pero decidimos separar los atributos de los detalles de la tabla Alquiler para mantener una estructura más normalizada y facilitar la gestión de la base de datos. Si en un futuro se quiere añadir un detalle se puede hacer en esta tabla.

PAGO_ALQUILER: Registra los pagos asociados a los alquileres. Se utilizó la misma estrategia que para PAGO_VENTA.

NOTA: La secuencia de ejecución del script fue cuidadosamente seleccionada por las relaciones entre las tablas. Para garantizar la integridad de los datos, es necesario que haya información presente en una tabla antes de que pueda haber datos en otra que esté relacionada con ella.

Modelo de BI:

En primera instancia decidimos migrar las dimensiones para después en la migración de las tablas de hecho poder asignar las FKs correspondientes.

A la hora de migrar las tablas de hechos se hace según las dimensiones elegidas , por eso en los procedimientos de migración se agrupan siempre según las dimensiones.

Tomando en cuenta que la granularidad mínima es por meses, en un ejemplo si hay dos o más anuncios el mismo mes y que tengan las mismas dimensiones , estos mismos formaran un solo registro en la tabla de hechos cuyo valor en las métricas sea el resultado de los calculos propios de cada tabla de hechos entre las métricas de los anuncios/ventas/alquileres del mismo mes.

También optamos por:

- 1) El uso de funciones escalares para que no se dupliquen filas en los procedimientos de migración de las tablas de hecho.
- 2) El uso de funciones de motor para sumar/contar los valores y/o la cantidad de registros que tengan las mismas dimensiones.

Tener esta granularidad en meses permite un análisis a más alto nivel para facilitar una mejor toma de decisiones.

Calculo de Promedios: para las vistas que ocupen promedios en las tablas de hecho está todo lo necesario para obtener un promedio más preciso :

Teniendo los campos a promediar, que a veces puede ser la suma de valores de diferentes métricas de anuncios/ alquileres / ventas. Para luego ser divididos por el campo cantidad(sumadas en la vistas si es que lo requiere), que representa la cantidad de registros de anuncios/pago venta / pago alquileres cuyas métricas fueron sumadas.

Esta métrica adicional de cantidad es necesaria para obtener un promedio más preciso.

Requerimientos que requieran una granularidad menor al mes:

Para las vistas donde se requiera una granularidad que no se alcance en las tablas de hecho , como puede ser la vista uno , que pide la duración promedio en días. Se dispondrá de agregar una nueva métrica en la tabla de hechos obteniendo su valor en la propia migración , donde todavía nos permite manejar una granularidad en días.

