

Estrategia

TP Gestión de Datos - 2c2023



GRUPO: DeLorean_V2

CURSO: 3522/3622

INTEGRANTES:

- González Peñaflor, José Gerónimo - 2037038
- González Seijas, Bautista - 2045928
- Nardi, Juan Ignacio - 2038924
- Vinciguerra, Camila Lara - 2027963

Índice

Estrategia - DER.....	2
Duración y períodos de alquiler:.....	2
Pago de alquiler:.....	2
Tablas VS ENUM:.....	2
Moneda:.....	3
Estrategia - Migración de Datos.....	4
Cambios en el DER:.....	4
Disposición del script:.....	4
Uso de funciones:.....	4
Uso de cursor:.....	5
Estrategia - BI.....	6
Dimensiones.....	6
Vistas creadas.....	7

Estrategia - DER

Duración y períodos de alquiler:

Colocamos el tipo de periodo en el alquiler y no en el anuncio pues puede ocurrir que el anuncio está dirigido a una venta y debería quedar como NULL en este último caso.

Pago de alquiler:

Lo desligamos de la tabla inquilino por lo mencionado anteriormente. A través del pago de alquiler se puede conocer al inquilino mediante el código de alquiler.

Tablas VS ENUM:

Hubo muchos casos en los que nos encontrábamos con datos ya tipificados, teniendo dos posibles soluciones: modelarlos mediante ENUMS o mediante Tablas Aparte, ambas opciones aportan la solución del problema que podría presentarnos la consistencia de datos al dejar estos datos como simples textos en atributos dentro de las tablas. La mayoría de los casos los solucionamos con ENUMS, que dan más performance y simplicidad a la hora de usarlos, pero con la principal desventaja de que no tienen una gran mantenibilidad como tener los datos en tablas aparte, que permitiría agregar más opciones, como vamos a ver en moneda, más adelante. Algunos de los casos en los que pensamos estas situaciones que convendría mencionar:

- 1) Medios de Pago: Fuera de los más conocidos como Tarjetas, Efectivos o Transferencias, pensamos que podría haber más tipos de medios de pago en un futuro, pero en nuestro acotado entendimiento de la realidad estos siempre se terminan catalogando como transferencias, por lo que terminó quedando como un ENUM.
- 2) Ambientes de un Inmueble: Vimos que hay varias posibilidades para estos, desde monoambientes hasta 5 ambientes con disposición, vemos muy improbable que se deba agregar un nuevo tipo, por lo que decidimos dejarlo como ENUM.
- 3) Orientación y Disposición: Ambos datos son muy acotados y nunca cambiarán (No se agregarán nuevos puntos cardinales, por ejemplo), por lo que los dejamos como ENUMS.
- 4) Moneda, será tratado más adelante.

Moneda:

Fue uno de los principales debates en cuanto a modelado, tuvimos tres alternativas, las cuales fueron:

- 1) Colocarla como un atributo en las tablas correspondientes con el tipo de dato VARCHAR. Rápidamente descartamos elegir este camino, tendríamos problemas con la consistencia de datos y con la mantenibilidad, además no estaríamos respetando la normalización para la tabla.
- 2) Colocarla como un ENUM, solucionando los problemas de consistencia de datos, pero vimos una gran desventaja en cuanto a la mantenibilidad y extensibilidad, es una alternativa que sería muy simple y performante, pero sus principales desventajas nos hicieron decantar por la última alternativa.
- 3) Colocarla como una tabla aparte a la que hacemos referencia a través de una Foreign Key, donde tenemos un ID numérico, junto a su ISO y una breve descripción. Esta opción no es tan performante como lo podría ser la opción anterior, pero nos da una gran mantenibilidad ya que sería más fácil agregar nuevas monedas a la base de datos, que creemos bastante probable que deba hacerse en un futuro no tan lejano, y creemos que esto pondera más que la performance de los ENUM.

Estrategia - Migración de Datos

Cambios en el DER:

- Sacamos los ENUMS innecesarios como tipo_dni en comprador o en propietario, ya que eran datos que no estaban en la tabla maestra.
- Normalizamos aún más la tabla de alquiler, creando una nueva tabla estado_alquiler donde guardamos las posibles tipificaciones de un estado de un alquiler.
- Eliminamos la tabla dirección, ya que en la tabla maestra no había información acerca de un piso o un departamento como habíamos propuesto y directamente relacionamos inmueble a barrio y sucursal a localidad, para evitar inconsistencias de datos.
- Agregamos la tabla medio_de_pago, donde dejamos guardados los posibles medios de pago que podrían aparecer.
- Eliminamos las tablas intermedias inquilino_por_alquiler y propietario_por_inmueble, ya que en base a nuestras consideraciones y a los datos de la tabla maestra pudimos ver que la relación entre inquilino y alquiler, o entre propietario e inmueble no eran many to many, si no one to many.

Disposición del script:

En primer lugar, se crean todas las tablas que consideramos acorde a nuestro DER realizado, **sin especificar aún las constraints de foreign key entre ellas (se realizan al final del script mediante sentencias ALTER TABLE).**

Luego, se crea un procedimiento por cada una de las tablas creadas, mediante el cual se realiza el procesamiento correspondiente a la migración de los datos que a dicha tabla corresponda, obtenidos desde la tabla maestra provista. Para ello dentro de estos procedimientos se recurre al uso de funciones auxiliares, específicamente para la obtención de los valores de las foreign keys.

Luego, se encapsula la ejecución de todos los procedimientos correspondientes a la migración en una única transacción para evitar problemas de consistencia ante potenciales fallas.

Finalmente se eliminan los recursos empleados, tanto procedimientos como funciones auxiliares.

Uso de funciones:

Para las tablas en las cuales requerimos una foreign key que apunte a otra entidad con una clave autogenerada, decidimos usar funciones de usuario. Estas funciones de usuario reciben los datos que son necesarios para devolver un id.

Por ejemplo, cuando migramos una venta que tiene una moneda específica, tipificada anteriormente en otra tabla, usamos la función a la que le damos el nombre, por ejemplo 'Moneda Pesos' y nos devuelve su id correspondiente de la tabla moneda.

Uso de cursor:

Un caso particular de migración se da con las características del inmueble, las cuales en la tabla Maestra se encuentran modeladas como cuatro columnas distintas representando cuatro características y con un valor numérico de 0 (falso) o 1 (verdadero) para indicar si el inmueble correspondiente a ese mismo registro las posee o no. En nuestro modelo, planteamos una relación “muchos a muchos” entre inmueble y característica. Los datos de la tabla característica los insertamos a mano en función de esas cuatro columnas identificadas en la tabla Maestra. Luego, para indicar correctamente qué características corresponden a cada inmueble (es decir, para migrar a la tabla intermedia `caracteristicas_del_inmueble`), utilizamos un cursor que selecciona las columnas de las cuatro características junto con la del código de inmueble de ese registro. Mediante sentencias condicionales, si el valor de dichas columnas era 0 o 1, insertamos en la tabla intermedia un registro con los valores del código del inmueble y la clave foránea que referencia a dicha característica en la tabla característica.

Estrategia - BI

COMENTARIO DE LA ENTREGA

El script de BI tarda un poco más de lo que esperábamos en ejecutarse, demorando en nuestra computadora un total de aproximadamente 6 minutos y 20 segundos.

Dimensiones

En base al requerimiento de generar un nuevo modelo de datos, de Inteligencia de Negocios, que permita unificar la información necesaria para facilitar la creación de los tableros de control a nivel gerencial, se han creado las siguientes dimensiones:

- BI_rango_m2: incluye los posibles valores del rango de metros cuadrado por inmueble
- BI_tipo_de_operacion: se guardan los posibles tipos de operación
- BI_tipo_moneda: se guardan los posibles tipos de moneda
- BI_ambientes: posibles codificaciones de la cantidad de ambientes de un inmueble
- BI_tipo_inmueble: se guardan los posibles tipos de inmueble
- BI_ubicacion: se guardarán aquellos barrios de la tabla maestra que previamente habíamos guardado en tablas normalizadas de la migración de datos
- BI_rango_etario: para los agentes e inquilinos
- BI_estado_alquiler: donde guardamos todos los posibles estados que tienen los alquileres.
- BI_tiempo: guardamos todas las fechas pertenecientes a nuestra base de datos, incluyendo año, cuatrimestre, mes y día, junto a su id. Se crea una función que, dada una fecha, devuelve el id correspondiente dentro de la tabla BI_tiempo

Entre las funciones creadas para facilitar la obtención de la fecha, incluimos `funcion_cuatrimstre` y `funcion_mes`. Estas logran una mejor presentación visual de las vistas.

Ejemplo

Previamente -> '1' indicando primer cuatrimestre

Mediante funciones creadas -> '1er cuatrimestre'

Entre las funciones que creamos se encuentran, por ejemplo, ciertas funciones como la de `funcion_aumento_alquiler`, que mientras hacíamos las correcciones nos dimos cuenta de que eran muy poco performantes, al punto de que al abstraer la lógica -en el caso particular de esta función- nos demoraba aproximadamente 1 hora en insertar los datos dentro de `BI_pago_alquiler`, pero al sacar la función y dejar la lógica directamente en la consulta optimizamos el tiempo a 6 minutos nada más.

Decidimos entonces, por temas de performance, no delegar en funciones. Aun así las dejamos pero sin usarlas.

Vistas creadas

En función a las dimensiones creadas previamente, creamos una serie de vistas que cumplen con el requisito de proveer, en forma simple desde consultas directas, la siguiente información:

1. Duración promedio (en días) que se encuentran publicados los anuncios según el tipo de operación (alquiler, venta, etc), barrio y ambientes para cada cuatrimestre de cada año. Se consideran todos los anuncios que se dieron de alta en ese cuatrimestre. La duración se calcula teniendo en cuenta la fecha de alta y de finalización.
2. Precio promedio de los anuncios de inmuebles según el tipo de operación (alquiler, venta, etc), tipo de inmueble y rango de m² para cada cuatrimestre/año. El precio se debe expresar en el tipo de moneda correspondiente, identificando de cuál se trata.

Para cumplir con los requerimientos 1 y 2 creamos una tabla de hechos BI_anuncio. En ella guardamos lo necesario para poder crear las vistas que cumplan con estos requerimientos y para eso, además, agrupamos los datos de los hechos en base a dos métricas, duracion_anuncio_promedio y precio_publicado_promedio.

3. Los 5 barrios más elegidos para alquilar en función del rango etario de los inquilinos para cada cuatrimestre/año. Se calcula en función de los alquileres dados de alta en dicho período.
4. Porcentaje de incumplimiento de pagos de alquileres en término por cada mes/año. Calculado en función de las fechas de pago y de vencimiento del mismo. El porcentaje se calcula en función del total de pagos en dicho período.
5. Porcentaje promedio de incremento del valor de los alquileres para los contratos en curso por mes/año. Se debe tomar en cuenta el último pago con respecto al del mes en curso, únicamente de aquellos alquileres que hayan tenido un aumento y están activos.

A fin de satisfacer estas condiciones, creamos una tabla de hechos BI_alquiler. Donde guardamos, agrupando en base a las dimensiones, las métricas de la cantidad_pagos, la cantidad_alquileres y el porcentaje_incumplimiento_pagos.

Para la vista de los 5 barrios más elegidos, decidimos cambiarlo a que en lugar de un top 5 de cada tipo de agrupación, sea un top 5 en general de entre todos los barrios, aun así dejamos las dos formas de hacerlo, aunque la primera está comentada, ya que la hicimos antes de las correcciones .

En base a las correcciones realizadas, decidimos crear una nueva tabla de hechos, BI_pago_alquiler, con el objetivo de satisfacer el requisito de la vista 5, debido a que ahora nos quedaba muy complejo el agrupamiento con los demás datos. Solamente guarda la métrica del porcentaje promedio de aumento de alquileres y la cantidad de estos, para sacar el promedio.

Para calcular el aumento a través de los meses implementamos una función que tome el período anterior al pago del alquiler para luego manipularlo aritméticamente y así obtener un

porcentaje. Y esta era la función que usábamos antes y nos demoraba mucho la ejecución, por lo que decidimos cambiarla por poner la lógica directamente en la consulta, como mencionamos al principio de este apartado.

A la hora de ejecutar esta consulta, dado que no había alquileres activos con incremento, resultaba en una tabla vacía por lo que la contemplación de estas condiciones quedó comentada.

6. Precio promedio de m² de la venta de inmuebles según el tipo de inmueble y la localidad para cada cuatrimestre/año. Se calcula en función de las ventas concretadas.

Creamos una tabla de hechos BI_venta. Los agrupamientos los realizamos en base a la ubicación, al tipo de inmueble que se vendió y a la fecha en que se realizó la operación. La única métrica que calculamos para esta agrupación es el promedio de metro cuadrado.

7. Valor promedio de la comisión según el tipo de operación (alquiler, venta, etc) y sucursal para cada cuatrimestre/año. Se calcula en función de los alquileres y ventas concretadas dentro del periodo.
8. Porcentaje de operaciones concretadas (tanto de alquileres como ventas) por cada sucursal, según el rango etario de los empleados por año, en función de la cantidad de anuncios publicados en ese mismo año.
9. Monto total de cierre de contratos por tipo de operación (tanto de alquileres como ventas) por cada cuatrimestre y sucursal, diferenciando el tipo de moneda.

Con el propósito de cumplir con estos criterios creamos una tabla de hechos BI_operacion.

Las agrupaciones fueron realizadas en base al tiempo (fecha en que se realizó la operación), el tipo de operación (si fue alquiler o venta), el rango etario de los agentes que publicaron esta operación, la sucursal a la que pertenecen y la moneda en la que se realiza la operación (dólares, pesos, etc.). Para el monto total de cierre de cada operación utilizamos el monto que aparece en el anuncio con el precio publicado del anuncio.

Las métricas usadas fueron, el cálculo de la comisión promedio, la cantidad de operaciones realizadas en total y también la cantidad de operaciones cerradas, así como también el monto total de cierre de las operaciones.