

Curso K3572

Grupo N°2

“CUATRO HISPANOS”

Integrantes:

- Franco Smuraglia
- Ailín Denoya
- Santiago Pérez
- Gonzalo Carrizo

ENTREGA 1:

Desarrollo del DER:

Para el desarrollo del DER hicimos mucho énfasis en los casos de uso que el sistema debe contemplar, para detectar cuáles serían las entidades más importantes

- Para la **gestión de anuncios** decidimos partir en base a una entidad ANUNCIO, la cual contendrá todos los datos de cada anuncio en la inmobiliaria.

Además, en los agentes su PK es un número único de agente y tiene vinculado una FK con su sucursal. Las sucursales están listadas en otra entidad SUCURSAL para evitar la redundancia de datos.

Decidimos crear una entidad para MONEDA y normalizar este dato debido a su aparición en diferentes entidades (si no se normalizara habría redundancia de datos). Todas las entidades que se relacionan con ella tienen una relación opcional debido a que puede haber monedas que no se hayan asociado a ninguna de esas entidades aún.

El costo es un atributo de la entidad INMUEBLE que, como alternativa, consideramos normalizar haciéndolo entidad para el caso de que tal dato sufra un proceso inflacionario. Pero lo consideramos fijo e igual para todos los anuncios sin tener en cuenta esto, por lo que decidimos mantenerlo como atributo.

En la entidad INMUEBLE tenemos una relación conformada de la siguiente manera: un inmueble tiene uno y un único propietario y un propietario puede tener ninguno o varios inmuebles. Decidimos crear esta entidad para agrupar a todos los tipos de persona que maneja la inmobiliaria, propietarios, inquilinos y comprador, excluyendo agentes ya que estos últimos tienen un rol un poco más diferenciado con los anteriores, y además están asociados con las sucursales.

Por otro lado, en UBICACION decidimos manejar la PK con una clave subrogada autoincremental para facilitar la utilización de la misma como clave foránea en otras entidades y también para que sea de mayor facilidad su búsqueda.

Como la ubicación no consiste de una calle, o una dirección específica, la relación la diseñamos como que un inmueble debe tener una y una única ubicación y una ubicación puede estar en varios inmuebles

Las características al ser una relación muchos a muchos creamos la entidad Característica_X_Inmueble intermedia entre inmueble y

característica, y en la cual también agregamos el atributo cantidad, por si la característica lo requiriese. Por ejemplo, si un inmueble tuviese 2 piscinas, en este atributo tendríamos esa cantidad reflejada, pero si se tratara de wifi, esta cantidad estaría en NULL o tendría 0. Por decisión análoga, modelamos la entidad AMBIENTE de la misma manera.

- Para la **gestión de alquileres** creamos la entidad ALQUILER, que es la que tiene todos los datos referentes a los alquileres en la inmobiliaria. Un alquiler tiene una fecha de inicio de contrato y una fecha de finalización del mismo. Además esta entidad está formada por muchos detalles de importe que manejamos con una entidad DETALLEIMPORTE, debido a que un alquiler puede tener muchos pagos por periodo, y cada uno de ellos puede estar conformado de manera diferente. Cada período de esta entidad va a estar correspondido al atributo tipoPeriodo que tiene la entidad ANUNCIO ya que la entidad ALQUILER está relacionada con ésta. En cada detalle de importe se reflejan fechas para cada periodo y precio que varían según el mismo.

El inquilino se relaciona con la misma entidad anteriormente nombrada, “persona”. En este caso, la persona será quien alquile el inmueble. Y dentro del inmueble tendremos a la persona propietaria, o sea, la persona que puso en alquiler el inmueble.

La relación es similar a la del propietario con el inmueble, un alquiler debe tener uno y un único inquilino (la persona que alquila el inmueble formalmente) puede tener ninguno o varios alquileres.

- Para la **gestión de pago de alquileres**, creamos una entidad llamada PAGOALQUILER, que contendrá todos los datos de cada pago de alquileres. Decidimos normalizar el medio de pago en una entidad medioPago ya que se utiliza en PAGOALQUILER y PAGOVENTA, las cuales tienen una relación opcional debido a que puede haber medios de pago que no se hayan asociado a ninguna de esas entidades aún. Esta normalización se hizo para evitar la redundancia de datos.
- Para **gestión de venta de inmuebles**, creamos la entidad venta, que tiene los datos correspondientes a las ventas de inmuebles La relación con la entidad anuncio, es que una venta debe tener un único anuncio y un anuncio puede o no tener asociado una venta, ya que puede existir anuncios que no sean del tipo venta (atributo de anuncio)

Luego el atributo comprador es una relación con la entidad persona, de características similares a las relaciones de “persona” con las otras entidades inmueble y alquiler

Y la moneda también es una relación de tipo similar a las anteriores. Por otro lado, modelamos el “importe de pago” con una entidad llamada

PAGOVENTA que mantiene una relación de uno a muchos con respecto a la entidad VENTA, ya que en el enunciado se especifica que “El comprador puede dividir el pago del total en partes, permitiendo pagar cada parte con distinta moneda y a través de distintos medios de pago” lo que implica que una venta consta de diferentes pagos con sus respectivos tipos de moneda, y demás atributos que colocamos en PAGOVENTA. En esta entidad, además tenemos un atributo cotización, el cual consideramos como alternativa normalizar y colocarlo como un atributo más de la entidad MONEDA, pero debido a que este dato sólo se utiliza al pagar una venta, decidimos mantenerlo en la entidad PAGOVENTA.

ENTREGA 2

La tabla Ambientes decidimos implementarla con un campo que indique la cantidad y otro extra que diga si son con o sin dependencia

Decidimos implementar PKs autoincrementales en todas las tablas ya que nos facilitaba la migración, en el caso de “Venta”, “Alquiler”, “Anuncio”, “Inmueble” dejamos el código que venía de la maestra igualmente, ya que lo usábamos para generar la ID autoincremental correspondiente y después las necesitábamos también para hacer funciones para obtener estas IDs

La tabla ubicación tiene a barrio, localidad y provincia porque barrio, localidad y provincia tienen muchas ubicaciones, entonces para hacer más fácil la búsqueda por provincia, barrio o localidad lo unimos en ubicación y le implementamos una PK autoincremental

En base al Feedback recibido de la entrega 1, decidimos crear las entidades diferenciadas, orientación, disposición, y estado

Y diferenciamos “persona” como agente, inquilino, propietario, comprador para diferenciar sus características en caso de haberlas o que se agreguen en el futuro

Y relacionamos barrio con localidad y localidad con provincia

Sucursal se relaciona con localidad

ENTREGA 3

Para la tabla “BI_HECHO_ANUNCIO”, al hacer la migración obtuvimos la duración de publicación de un anuncio restando las fechas iniciales y finales. Esto nos permite obtener este dato sin mayores inconvenientes al momento de crear la vista. Para los rangos etario y M2, utilizamos una función auxiliar que busca los agentes y las superficies respectivamente, para poder reutilizarla en futuras consultas y/o comparaciones. Para ambos rangos decidimos crear tablas donde se guardan los registros mencionados en el enunciado, para así facilitar el desarrollo de estas funciones.

Es importante remarcar que creamos una tabla “BI_TIEMPO” en la que juntamos todas las fechas de todas las tablas y le agregamos un ID único para poder identificar cada fecha con facilidad. Esta tabla también tiene sus FKs en cada una de las tablas de tipo “hecho”. Además, agregamos una columna “cuatrimestre”, para que su cálculo no sea repetido cada vez que nos lo pidieran en una vista.

Una función quizás relevante de mencionar es “CUATRO_HISPANOS0.obtenerSiAumentoOno”, en la que fuimos calculando el porcentaje de cada incremento. En el caso de que no hubiese incremento, decidimos retornar NULL, para que en la vista 5, dónde debíamos mostrar el porcentaje promedio de aumento, el COUNT no contara aquellos valores. Además, viendo que esta vista no retornaba ningún registro que cumpliera con la condición de aumento, al hacer el porcentaje nos resultaba en una división por cero.

Debíamos considerar este error para que la query funcionara, por lo que utilizamos la función NULLIF para que, en el caso de que el valor del denominador para obtener el porcentaje fuera cero, nos diera un denominador con valor NULL. Como sabemos que cualquier operación que se realice con el valor de NULL, nos da otro valor NULL (o sea, que algo desconocido comparado u operado contra otro valor desconocido nos da otra cosa desconocida), utilizamos este recurso para aplicarlo a nuestra función y así manejar ese error. Como el enunciado pedía mostrar únicamente los registros cuyos alquileres hayan tenido aumento, implementamos algo similar en el filtro del HAVING, para que la tabla de por si mostrara que no existieron aumentos (según los datos migrados de la tabla maestra original).

ENTREGA 3.1 REENTREGA

Habiendo entregado mal los datos en la versión anterior, corregimos las bases de hechos las cuales habíamos llenado con todos los datos del script inicial, de esta manera ahora cargamos datos ordenados por volúmenes de todas las entidades y agrupados por las dimensiones antes creadas, quedando información relevante para análisis de datos.

Cada vista entonces solamente agrupa por lo que se pide, y además hace algún cálculo de promedio o porcentaje