

Entrega 1: DER

Personas (INQUILINO, AGENTE, PROPIETARIO y COMPRADOR):

A pesar de que estas entidades compartan todos sus atributos, menos el AGENTE que se le suma la sucursal a la que pertenece, se decidió implementarlas como entidades distintas ya que cumplen funciones diferentes en el sistema.

Tipificados:

Todas aquellas entidades que se detallan en el enunciado como “tipificados”, se decidió implementarlas en sus tablas correspondientes con su id (PK) y su detalle, para mantener consistencia en los datos.

ALQUILER:

La entidad IMPORTE_PERIODO guarda el importe por cada rango de períodos del alquiler. Por ejemplo, si el TIPO_PERIODO es mensual y el ANUNCIO del ALQUILER tiene la MONEDA de pesos, entonces una posible entrada de IMPORTE_PERIODO podría ser periodo_inicio=1, periodo_fin=6, precio=100.000. Esa entrada hace referencia a que desde el período 1 hasta el período 6 (en este caso, meses) el precio del alquiler es 100.000 (en este caso, pesos).

La entidad PAGO_ALQUILER guarda los pagos de los períodos de cada alquiler.

DIRECCIÓN:

Se decidió normalizar la dirección, utilizada por el INMUEBLE y la SUCURSAL. Además, se decidió desnormalizar la LOCALIDAD y la PROVINCIA, para evitar JOINS excesivos, siendo que se agregan sólo dos campos a la DIRECCIÓN. La otra posibilidad era simplemente hacer referencia al BARRIO desde la DIRECCIÓN y desde ella obtener la LOCALIDAD, y de forma análoga obtener la PROVINCIA. Esos son los JOINS que se evitan desnormalizándolas.

INMUEBLE:

Se decidió normalizar las características posibles en la entidad CARACTERISTICA, y se implementó una tabla intermedia CARACTERISTICA_X_INMUEBLE para tratar el muchos a muchos con el INMUEBLE.

VENTA:

La entidad PAGO_VENTA guarda los pagos que se realizan en la venta de un inmueble. Como una VENTA puede tener muchos PAGO_VENTA, se referencia a la VENTA desde el PAGO_VENTA.

Entrega 2: Modelo de Datos Relacional y Migración

El script para la migración de datos se decidió organizarlo de la siguiente manera:

1. **Creación de tablas**: se crean las tablas del modelo de datos relacional, siguiendo el DER presentado en la entrega anterior. Las únicas modificaciones realizadas al DER, respecto de la anterior entrega, fueron:
 - a. Agregar entidad AMBIENTE.
 - b. Cambiar PKs y FKs para respetar el tipo de datos de la base de datos maestra: *decimal(18,0)*.
 - c. Eliminar FK de *id_anuncio* en la tabla INMUEBLE, ya que un inmueble puede tener muchos anuncios. Con el fin de no suponer condiciones como “tomar el anuncio más nuevo para guardar en el inmueble”, se decidió eliminar esa referencia desde el inmueble y mantenerla sólo desde el anuncio.
 - d. Cambiar tipo de datos de los *numero_doc* a *numeric(18,0)* para respetar el tipo de datos de la base de datos maestra.
2. **Creación de funciones**: se crean funciones útiles para la migración de los datos.
3. **Declaración de procedures**: se crean los procedimientos para realizar las migraciones de los datos. Se crea un procedimiento por cada una de las nuevas tablas generadas en el modelo de datos relacional, manteniendo el mismo formato para la identificación de los mismos: *GESTIONATE.migrar_<ENTIDAD>*.
4. **Declaración de índices**: se crean ciertos índices sobre las tablas generadas, con el fin de mejorar la performance de la migración.
5. **Ejecución de procedures**: se ejecutan los procedures declarados en la tercera sección en el orden correspondiente para migrar todos los datos al nuevo modelo de datos relacional.

Entrega 3: Business Intelligence

El script para la migración de datos se decidió organizarlo de la siguiente manera:

1. **Eliminación de tablas y vistas**: se eliminan las tablas de hechos y las vistas que pudieron ser creadas en una iteración anterior.
2. **Creación de dimensiones**: se crean las tablas de dimensiones especificadas en el enunciado, sumando la dimensión del Estado del Anuncio que se utilizará para facilitar cálculos y aumentar performance en las últimas dos vistas.
3. **Creación de hechos**: se decidió crear 3 tipos de hechos (Alquiler, Venta y Anuncio) ya que consideramos que son las entidades principales de nuestro sistema y sobre las cuales se van a realizar las consultas en las vistas.
4. **Creación de funciones útiles en llenado**: se crean funciones que realizan cálculos útiles para el llenado de las tablas de dimensiones.
5. **Cargado de tablas de dimensiones**: se llenan las dimensiones con los valores que se obtienen del modelo transaccional.
6. **Creación de funciones para obtener ids**: se crean funciones que, dado un valor, retorna el id de la entrada que corresponde en su tabla de dimensiones. Estas funciones serán de utilidad para el llenado de hechos y las consultas en las vistas.
7. **Llenado de hechos**: se llenan las tres tablas de hechos creadas previamente, con la información obtenida del modelo transaccional.
8. **Creación de vistas**: finalmente, se crean las vistas que cumplen con los 9 requerimientos especificados en el enunciado.