

Corecciones reentrega Creacion Inicial

Modificacion del der

Nos dimos cuenta que en realidad el alquiler no debe tener muchos detalles importe dado que nos sucedia que teniamos muchos alquileres con todos los mismos atributos a excepcion del detalle importe

Por ende, a partir de ahora un detalle importe tiene el codigo del alquiler al que hacer referencia. Esto nos mejora la normalizacion de datos al evitar tener todos estos datos repetidos

```
entity alquiler {
  idAlquiler: numeric PK
  --
  anuncio: Anuncio FK
  inquilino: inquilino FK
  duracion: numeric(18, 2)
  fechaInicio: DateTime
  fechaFin: DateTime
  deposito: numeric (18, 2)
  comision: numeric (18, 2)
  gastosAveriguaciones: numeric (18, 2)
  estadoAlquiler: Estado FK
}
```

```
entity detalleImporte {
  idDetalleImporte: numeric
  alquiler_id : Alquiler FK
  nroPeriodoInicio: numeric
  nroPeriodoFin: numeric
  precio: numeric
}
```

Modificacion del procedure

Ahora a la hora de mirar alquiler no se hace un left join con la tabla detalle_importe, sino que detalle importe hace join con la tabla alquiler y guarda su id. Ademàs, ahora la migracion de pago_alquiler no hace un join con detalle importe, ni un between con las fechas debido a que el detalle importe posee el alquiler.

Modelo de BI

Decidimos modelar cuatro tablas de hechos La primera para los anuncios en que para poder realizar las vistas tuvimos que guardar algunas metricas como el

precio promedio, la cantidad de anuncios, la cantidad de operaciones concretadas y el promedio de comision.

Otra tabla necesaria fue el hecho alquiler en la que tuvimos que guardar algunas metricas relacionadas para poder realizar la vista de incumplimiento de pagos como el porcentaje de incumplimiento de pagos, la cantidad de pagos, y la cantidad de alquileres

Caso similar con la tabla de ventas, que para poder realizar la vista de precio promedio de m2 de la venta, utilizamos la cantidad de ventas y el precio promedio

En todas las tablas de hechos, la PK es siempre el conjunto de las dimensiones necesarias

Además de Alquileres, Ventas y Anuncios, decidimos hacer una tabla de hechos de Pagos de Alquileres, porque la lógica de buscar el pago del mes anterior hacía que la tabla de hechos de alquileres quedara muy complicada.

Un problema con el que nos encontramos es que en la base de datos inicial no existía ningún alquiler activo que tenga más de un registro, por lo que sacar el aumento para ellos no era posible. Para la view que pedía el aumento de los alquileres activos, esto resultaba en que dé vacío, por lo que dejamos comentado el chequeo de si son activos.

Fue necesario también usar una operación de promedio ponderado para calcular las views. Cuando para cada combinación de valores de cada dimensión se calculaba una ya que usar la función `AVG()` no hubiese contemplado la cantidad