Universidad Tecnológica Nacional
Facultad Regional Buenos Aires

Gestión de Datos

Trabajo Práctico

FRBA – Inmobiliaria

Integrantes:

Ascorti, Federico

Medina, Piero

Bravo, Nahuel

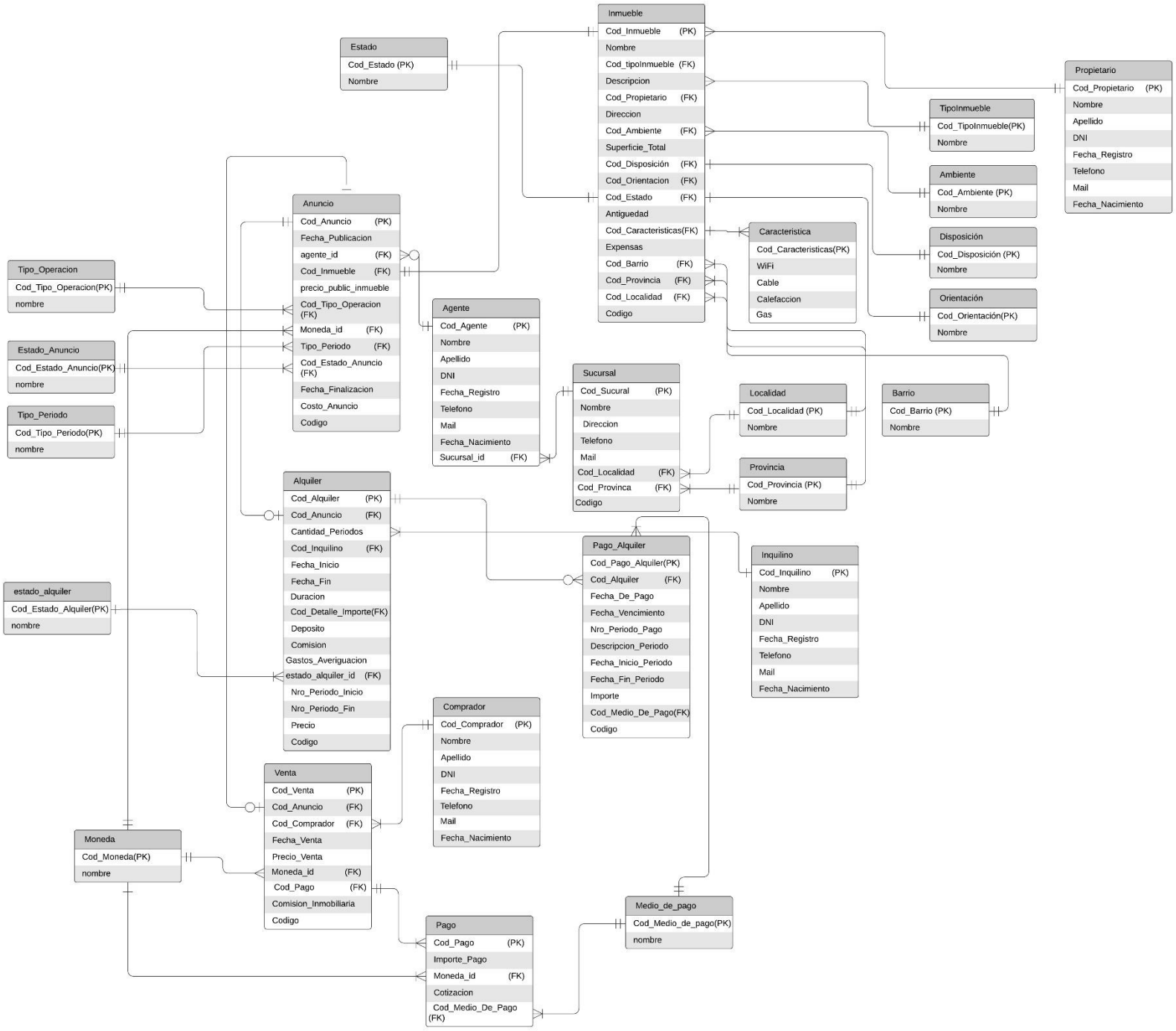
Ibarra, Laureano

Curso: K3622

Número de curso: 5

Índice:

DER


Inmueble
Cod_Inmueble (PK)
Nombre
Cod_tipoInmueble (FK)
Descripcion
Cod_Propietario (FK)
Direccion
Cod_Ambiente (FK)
Superficie_Total
Cod_Disposición (FK)
Cod_Orientacion (FK)
Cod_Estado (FK)
Antiguedad
Cod_Caracteristicas(FK)
Expensas
Cod_Barrio (FK)
Cod_Provincia (FK)
Cod_Localidad (FK)
Codigo
Estado
Cod_Estado (PK)
Nombre
Propietario
Cod_Propietario (PK)
Nombre
Apellido
DNI
Fecha_Registro
Telefono
Mail
Fecha_Nacimiento
TipoInmueble
Cod_TipoInmueble(PK)
Nombre
Ambiente
Cod_Ambiente (PK)
Nombre
Disposición
Cod_Disposición (PK)
Nombre
Orientación
Cod_Orientación(PK)
Nombre
Caracteristica
Cod_Caracteristicas(PK)
WiFi
Cable
Calefaccion
Gas
Anuncio
Cod_Anuncio (PK)
Fecha_Publicacion
agente_id (FK)
Cod_Inmueble (FK)
precio_public_inmueble
Cod_Tipo_Operacion (FK)
Moneda_id (FK)
Tipo_Periodo (FK)
Cod_Estado_Anuncio (FK)
Fecha_Finalizacion
Costo_Anuncio
Codigo
Tipo_Operacion
Cod_Tipo_Operacion(PK)
nombre
Estado_Anuncio
Cod_Estado_Anuncio(PK)
nombre
Tipo_Periodo
Cod_Tipo_Periodo(PK)
nombre
Agente
Cod_Agente (PK)
Nombre
Apellido
DNI
Fecha_Registro
Telefono
Mail
Fecha_Nacimiento
Sucursal_id (FK)
Sucursal
Cod_Sucural (PK)
Nombre
Direccion
Telefono
Mail
Cod_Localidad (FK)
Cod_Provinca (FK)
Codigo
Localidad
Cod_Localidad (PK)
Nombre
Barrio
Cod_Barrio (PK)
Nombre
Provincia
Cod_Provincia (PK)
Nombre
Alquiler
Cod_Alquiler (PK)
Cod_Anuncio (FK)
Cantidad_Periodos
Cod_Inquilino (FK)
Fecha_Inicio
Fecha_Fin
Duracion
Cod_Detalle_Importe(FK)
Deposito
Comision
Gastos_Averiguacion
estado_alquiler_id (FK)
Nro_Periodo_Inicio
Nro_Periodo_Fin
Precio
Codigo
Pago_Alquiler
Cod_Pago_Alquiler(PK)
Cod_Alquiler (FK)
Fecha_De_Pago
Fecha_Vencimiento
Nro_Periodo_Pago
Descripcion_Periodo
Fecha_Inicio_Periodo
Fecha_Fin_Periodo
Importe
Cod_Medio_De_Pago(FK)
Codigo
Inquilino
Cod_Inquilino (PK)
Nombre
Apellido
DNI
Fecha_Registro
Telefono
Mail
Fecha_Nacimiento
estado_alquiler
Cod_Estado_Alquiler(PK)
nombre
Comprador
Cod_Comprador (PK)
Nombre
Apellido
DNI
Fecha_Registro
Telefono
Mail
Fecha_Nacimiento
Venta
Cod_Venta (PK)
Cod_Anuncio (FK)
Cod_Comprador (FK)
Fecha_Venta
Precio_Venta
Moneda_id (FK)
Cod_Pago (FK)
Comision_Inmobiliaria
Codigo
Moneda
Cod_Moneda(PK)
nombre
Pago
Cod_Pago (PK)
Importe_Pago
Moneda_id (FK)
Cotizacion
Cod_Medio_De_Pago (FK)
Medio_de_pago
Cod_Medio_de_pago(PK)
nombre

Aclaraciones DER

Para todas las entidades graficadas, sus atributos los fuimos seleccionando tanto del enunciado, como de la tabla maestra proporcionada por la cátedra. Además, agregamos algunos detalles que nos parecieron convenientes.

También decidimos separar las características en una tabla separada para que estas estén almacenadas de forma conjunta. Además, seleccionamos esos cuatro atributos nada más, ya que son los campos que están en la tabla maestra. Pero se podrían agregar más. Al igual que la dirección, esto podemos considerar modificarlo en un futuro.

La forma de conseguir el inmueble para las ventas y el alquiler es a través del anuncio de estos mismos. Lo decidimos de esta forma porque así esta estipulado en enunciado del trabajo práctico.

Para todas las tablas, decidimos seguir una nomenclatura de “Cod_NombreTabla” para decidirnos por sus claves primarias y lo mismo para representar las claves foráneas. En el script decidimos que a cada PRIMARY KEY llamarla “id” y para las FOREIGN KEY llamamos “nombre_tabla_id”, lo decidimos así para que sea consistente para todas las tablas y agilizar un poco el codeo. Lo aclaramos porque están de formas distintas en el DER y el script.

Aclaraciones Script de creación inicial

Lo primero que hacemos es eliminar de forma dinámica todas las FKs, luego las tablas y por ultimo el esquema. Esto lo realizamos para que cada vez que ejecutemos el script se realice todo de cero, no haya inconsistencia con datos repetidos cuando hacemos la migración y cuando creamos las tablas en caso de tener que cambiar algo no ir borrando todo por separado.

```
-- DROPS FOREIGN KEY
DECLARE @DropFKs NVARCHAR(max) = ''
SELECT @DropFKs += 'ALTER TABLE ' + QUOTENAME(OBJECT_SCHEMA_NAME(parent_object_id)) + '.'
                        +  QUOTENAME(OBJECT_NAME(parent_object_id)) + ' ' + 'DROP CONSTRAINT' + QUOTENAME(name)
FROM sys.foreign_keys

EXEC sp_executesql @DropFKs;
PRINT 'Constraints eliminadas correctamente';
GO

-- DROPS TABLAS
DECLARE @DropTables NVARCHAR(max) = ''
SELECT @DropTables += 'DROP TABLE Los_Magios. ' + QUOTENAME(TABLE_NAME)
FROM INFORMATION_SCHEMA.TABLES
WHERE TABLE_SCHEMA = 'LOS_MAGIOS' and TABLE_TYPE = 'BASE TABLE'

EXEC sp_executesql @DropTables;
PRINT 'Tablas eliminadas';
GO

-- DROP SCHEMA
IF EXISTS(SELECT NAME FROM sys.schemas WHERE name = 'LOS_MAGIOS')
    DROP SCHEMA LOS_MAGIOS;
GO
```

Luego creamos el esquema y procedemos con el creado de las tablas. Como aclaramos anteriormente, decidimos llamar a todas las PKs como id y todas las FKs utilizamos la notación de “nombre_tabla_id”

```
-- CREACION SCHEMA
CREATE SCHEMA LOS_MAGIOS;
GO


-- CREACION DE TABLAS

-- Modulo 1
CREATE TABLE LOS_MAGIOS.anuncio(
    id INT IDENTITY NOT NULL PRIMARY KEY,
    codigo NUMERIC(19, 0) NOT NULL,
    fecha_publicacion DATETIME NOT NULL,
    agente_id INT NOT NULL,
    tipo_operacion_id INT NOT NULL,
    inmueble_id INT NULL,
    precio_publicado NUMERIC(18,2) NOT NULL,
    moneda_id INT NOT NULL,
    tipo_periodo_id INT NOT NULL,
    estado_anuncio_id INT NOT NULL,
    fecha_finalizacion_anuncio DATETIME NOT NULL,
    costo_publicacion NUMERIC(18,2) NOT NULL
);
```

Seguimos con la creación de las FKs

```
-- FOREIGN KEY
ALTER TABLE LOS_MAGIOS.anuncio
ADD CONSTRAINT FK_agente_id FOREIGN KEY (agente_id) REFERENCES LOS_MAGIOS.agente(id),
    CONSTRAINT FK_inmueble_id FOREIGN KEY (inmueble_id) REFERENCES LOS_MAGIOS.inmueble(id),
    CONSTRAINT FK_moneda_id FOREIGN KEY (moneda_id) REFERENCES LOS_MAGIOS.moneda(id),
    CONSTRAINT FK_estado_anuncio_id FOREIGN KEY (estado_anuncio_id) REFERENCES LOS_MAGIOS.estado_anuncio(id),
    CONSTRAINT FK_tipo_periodo_id FOREIGN KEY (tipo_periodo_id) REFERENCES LOS_MAGIOS.tipo_periodo(id),
    CONSTRAINT FK_tipo_operacion_id FOREIGN KEY (tipo_operacion_id) REFERENCES LOS_MAGIOS.tipo_operacion(id)
```

Después realizamos la migración de los datos, antes de crear el procedure buscamos si ya existe y en caso de que lo haga le realizamos un DROP. Esto lo hacemos para que no haya inconsistencia o datos repetidos cuando corremos el script.

```
-- CREACION DE STORED PROCEDURES PARA MIGRACION
IF EXISTS (SELECT [name] FROM sys.procedures WHERE [name] = 'migrar_tipo_operacion_anuncio')
    DROP PROCEDURE migrar_tipo_operacion_anuncio
GO

CREATE PROCEDURE migrar_tipo_operacion_anuncio
 AS
  BEGIN
     INSERT INTO LOS_MAGIOS.tipo_operacion
     (nombre)
     SELECT DISTINCT m.ANUNCIO_TIPO_OPERACION
     FROM gd_esquema.Maestra AS m
     WHERE m.ANUNCIO_TIPO_OPERACION IS NOT NULL
  END
GO
```

Por último, realizamos la migración final ejecutando todos los procedures. Lo realizamos en una transacción para que en caso de que una falle, no se migre nada.

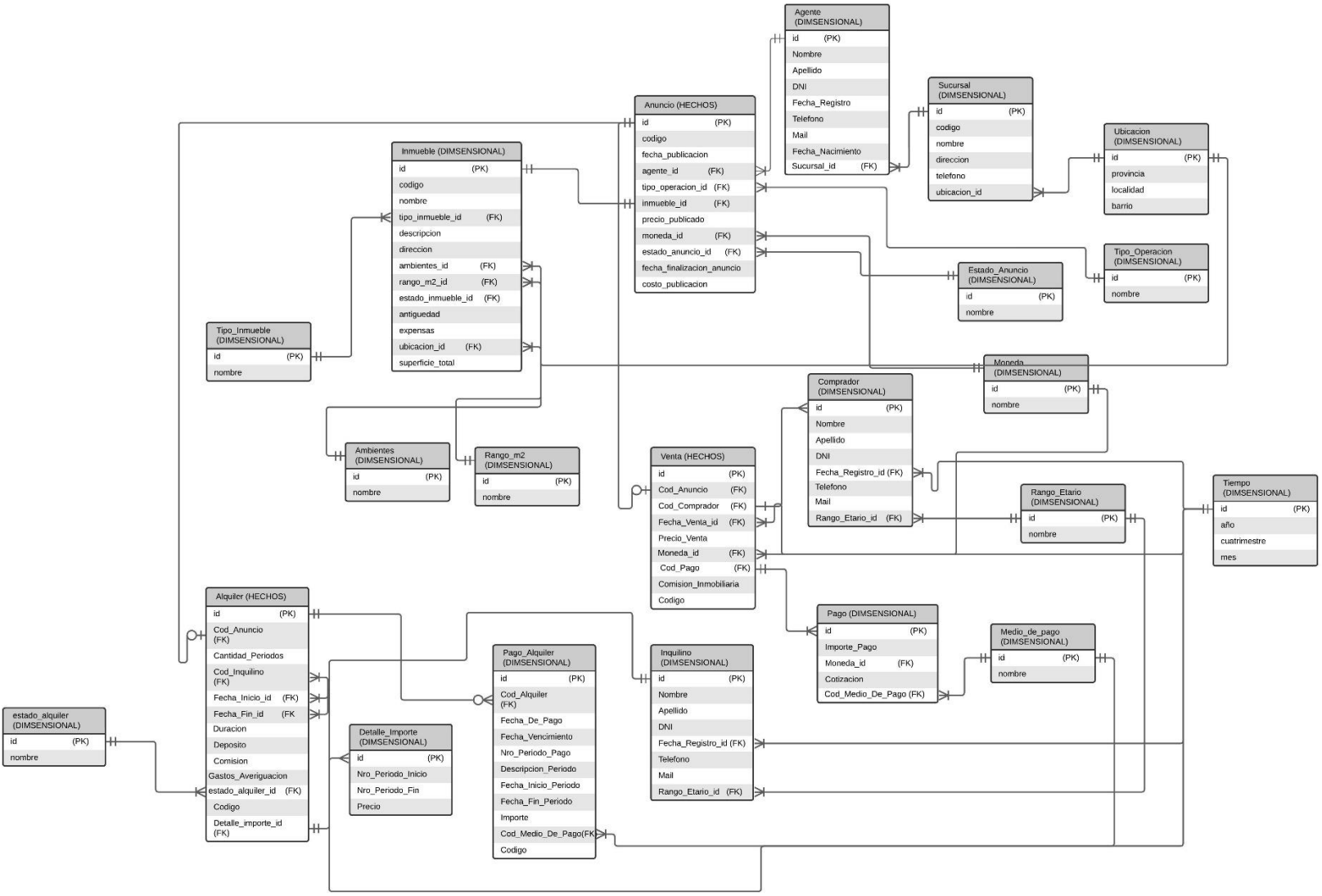
```
-- EJECUCION DE STORED PROCEDURES PARA LA MIGRACION
BEGIN TRANSACTION

BEGIN TRY
     EXECUTE migrar_tipo_operacion_anuncio
     EXECUTE migrar_tipo_periodo_anuncio
     EXECUTE migrar_estado_anuncio
     EXECUTE migrar_provincia
     EXECUTE migrar_estado_inmueble
     EXECUTE migrar_estado_alquiler
     EXECUTE migrar_localidad
     EXECUTE migrar_barrio
     EXECUTE migrar_moneda
     EXECUTE migrar_medio_de_pago
     EXECUTE migrar_tipo_inmueble
     EXECUTE migrar_ambientes
     EXECUTE migrar_disposicion
     EXECUTE migrar_orientacion
     EXECUTE migrar_sucursal
     EXECUTE migrar_agente
     EXECUTE migrar_propietario
     EXECUTE migrar_inquilino
     EXECUTE migrar_comprador
     EXECUTE migrar_pago_venta
     EXECUTE migrar_caracteristicas
     EXECUTE migrar_inmueble
     EXECUTE migrar_anuncio
     EXECUTE migrar_alquiler
     EXECUTE migrar_pago_alquiler
     EXECUTE migrar_venta
END TRY

BEGIN CATCH
     ROLLBACK TRANSACTION;
     THROW 50000, 'Error al migrar', 1;
END CATCH

COMMIT TRANSACTION
```

DER del BI


Agente (DIMSENSIONAL)
id (PK)
Nombre
Apellido
DNI
Fecha_Registro
Telefono
Mail
Fecha_Nacimiento
Sucursal_id (FK)
Sucursal (DIMSENSIONAL)
id (PK)
codigo
nombre
direccion
telefono
ubicacion_id
Ubicacion (DIMSENSIONAL)
id (PK)
provincia
localidad
barrio
Anuncio (HECHOS)
id (PK)
codigo
fecha_publicacion
agente_id (FK)
tipo_operacion_id (FK)
inmueble_id (FK)
precio_publicado
moneda_id (FK)
estado_anuncio_id (FK)
fecha_finalizacion_anuncio
costo_publicacion
Inmueble (DIMSENSIONAL)
id (PK)
codigo
nombre
tipo_inmueble_id (FK)
descripcion
direccion
ambientes_id (FK)
rango_m2_id (FK)
estado_inmueble_id (FK)
antiguedad
expensas
ubicacion_id (FK)
superficie_total
Estado_Anuncio (DIMSENSIONAL)
id (PK)
nombre
Tipo_Operacion (DIMSENSIONAL)
id (PK)
nombre
Tipo_Inmueble (DIMSENSIONAL)
id (PK)
nombre
Moneda (DIMSENSIONAL)
id (PK)
nombre
Comprador (DIMSENSIONAL)
id (PK)
Nombre
Apellido
DNI
Fecha_Registro_id (FK)
Telefono
Mail
Rango_Etario_id (FK)
Ambientes (DIMSENSIONAL)
id (PK)
nombre
Rango_m2 (DIMSENSIONAL)
id (PK)
nombre
Venta (HECHOS)
id (PK)
Cod_Anuncio (FK)
Cod_Comprador (FK)
Fecha_Venta_id (FK)
Precio_Venta
Moneda_id (FK)
Cod_Pago (FK)
Comision_Inmobiliaria
Codigo
Rango_Etario (DIMSENSIONAL)
id (PK)
nombre
Tiempo (DIMSENSIONAL)
id (PK)
año
cuatrimestre
mes
Alquiler (HECHOS)
id (PK)
Cod_Anuncio (FK)
Cantidad_Periodos
Cod_Inquilino (FK)
Fecha_Inicio_id (FK)
Fecha_Fin_id (FK
Duracion
Deposito
Comision
Gastos_Averiguacion
estado_alquiler_id (FK)
Codigo
Detalle_importe_id (FK)
Pago (DIMSENSIONAL)
id (PK)
Importe_Pago
Moneda_id (FK)
Cotizacion
Cod_Medio_De_Pago (FK)
Medio_de_pago (DIMSENSIONAL)
id (PK)
nombre
estado_alquiler (DIMSENSIONAL)
id (PK)
nombre
Detalle_Importe (DIMSENSIONAL)
id (PK)
Nro_Periodo_Inicio
Nro_Periodo_Fin
Precio
Pago_Alquiler (DIMSENSIONAL)
id (PK)
Cod_Alquiler (FK)
Fecha_De_Pago
Fecha_Vencimiento
Nro_Periodo_Pago
Descripcion_Periodo
Fecha_Inicio_Periodo
Fecha_Fin_Periodo
Importe
Cod_Medio_De_Pago(FK)
Codigo
Inquilino (DIMSENSIONAL)
id (PK)
Nombre
Apellido
DNI
Fecha_Registro_id (FK)
Telefono
Mail
Rango_Etario_id (FK)

Aclaraciones DER del BI

Nosotros tenemos 3 tablas de hechos, estas son alquiler, anuncio y venta. El resto son tablas dimensionales. En las tablas decidimos nombrar a todas las PK's como id, para que de esta forma seguir teniendo esa consistencia y agilizar un poco el trabajo.

Para las FK's decidimos nombrar a todas de la forma <nombre_tabla_referenciada_id>, por la misma razón que el id para las PK's. También por esto mismo, a todas las tablas con datos tipificados decidimos que solo almacenen su respectivo id y un nvarchar con el nombre a los que se esté referenciando.

Aclaraciones sobre script de creación del BI

En todo lo que realizamos, tanto para tablas, procedures, funciones, views antes de que se creen o se ejecuten analizamos que si existe les haga un drop. Para que al ejecutar varias veces el codigo luego no haya inconsistencia de datos. Para los único que no lo realizamos así es en las FK's. Estas están al inicio del código y no junto con su creación.

```
-- DROPS DE FKS

-- FKS SUCURSAL
IF EXISTS(SELECT 1
          FROM INFORMATION_SCHEMA.REFERENTIAL_CONSTRAINTS
          WHERE CONSTRAINT_NAME = 'FK_BI_ubicacion_id')
BEGIN    -- Eliminar la clave foránea si existe
    ALTER TABLE LOS_MAGIOS.BI_sucursal
    DROP CONSTRAINT FK_BI_ubicacion_id;
END
```

Luego tenemos 3 funciones auxiliares, una es para queda dada una fecha conseguir el cuatrimestre al que pertenece, otra que dada una fecha de nacimiento obtener el rango etario al que pertenece y por ultima, una que mediante un valor numerico, el cual es la superficie total, nos da como resultado el rango de m^2.

Proseguimos con la creación de todas las tablas del modelo, luego añadimos los contraints correspondientes a cada tabla. Las tablas tiene un nombre del tipo LOS_MAGIOS.BI_<nombre_tabla>.

Seguimos con la creación y ejecución de procedures para migrar los datos. Estos tienen un nombre de migrar_BI_<nombre_tabla>. Para la ejecución decidimos utilizar una transacción, ya que en caso de que una falle, no migra ningún dato.

Después están las creaciones de todas las vistas pedidas. Las vistas tiene un nombre de view_<funcionalidad> y por último están todos los select's comentados de todas las vistas y las tablas.