

# Migración de tipo de datos tipificados

---

Los datos tipificados fueron lo primero que decidimos migrar, ya que no tiene Foreign Keys y son muy simples.

La primera complicación que surgió fue migrar a la misma tabla datos tipificados que se podían encontrar en muchas columnas de la tabla maestra. La solución fue hacer el `INSERT` desde más de un `SELECT` unidos por `UNION`. Con esto el código ya funcionaba, pero teníamos muchas repeticiones de la forma

```
INSERT INTO table
    (nombre)
SELECT DISTINCT column
FROM gd_esquema.Maestra
WHERE column IS NOT NULL;
```

Para solucionarlo, decidimos hacer un procedure que reciba dos parámetros, tabla de destino y columna de origen, y que implementara este formato. Mientras que esto no contempla los casos con un `UNION`, estos se puede emular llamando al nuevo procedure más de una vez con columnas de origen diferentes. `INSERT`

## Migración de personas

---

La principal complicación de esto es que en la bases de datos original no existen las personas, sino que estas pueden ser Inquilinos, Propietarios, Agentes, o Compradores, cada cual tiene sus propias columnas. Si se hubieran migrado todos los datos a la entidad Persona sin tener en cuenta el resto de las Entidades, hubiera sido difícil recuperar la información de si la persona migrada era inquilino, propietario, comprador, o agente. Además, para migrar las tablas como Inquilino, se requiere el id de la Persona, por lo que sólo se la puede migrar después de migrar la persona.

Para encarar esto, por lo tanto, usamos tablas temporales y el operador `OUTPUT`. Habiendo un procedure diferente para inquilinos, propietarios, compradores, y agentes, creamos una tabla temporal cuando se empieza el procedure. Después, se insertan los datos en Personas, y usando

```
OUTPUT inserted.id_persona INTO #Temp (id_persona)
```

insertamos todos los ids nuevamente creados en la tabla temporal, que después insertamos en la tabla que corresponda.

## Migración de tablas normales

Para las tablas normales, el proceso es bastante simple. Se crea la tabla, se insertan los datos, y se crean las Foreign Keys.

Luego, para migrarlas, simplemente se seleccionan los datos de la tabla original y se insertan en la nueva. Y los datos que son Foreign Keys se reemplazan por los ids de las tablas nuevas mediante JOINS.

Un problema que surgió fue para buscar una persona, debido a que el dni no es una primary key, se debía buscar dato por dato y comprobar que todos los datos sean iguales. Para no repetir código, se creó una función que recibe los datos de la persona y devuelve el id de la misma.

```
CREATE FUNCTION GetPersonaID
(
    @nombre NVARCHAR(100),
    @apellido NVARCHAR(100),
    @dni NVARCHAR(100),
    @telefono NVARCHAR(100),
    @mail NVARCHAR(100),
    @fecha_registro DATETIME,
    @fecha_nacimiento DATETIME
)
RETURNS INT
AS
BEGIN
    DECLARE @id_persona INT;

    SELECT @id_persona = id_persona
    FROM ANDY_Y_SUS_SEMINARAS.Persona
    WHERE
        nombre = @nombre
        AND apellido = @apellido
        AND dni = @dni
        AND telefono = @telefono
        AND mail = @mail
        AND fecha_registro = @fecha_registro
        AND fecha_nacimiento = @fecha_nacimiento;

    RETURN @id_persona;
END
GO
```