

ESTRATEGIA - GRUPO_1

Curso: K 3622

Número de grupo: 30

Alumno	Legajo
Tomas Martinez	1758603
Federico Cassina Carbia	2026715
Luciano Nicolás Arrieta	1729573
Martina Abril Dominguez	2035716

Índice

IDs.....	3
Funciones.....	3
Procedures.....	3

IDs

En la entrega anterior, habíamos utilizado como Primary Key el código ya cargado en la tabla maestra. Por ejemplo: Para el alquiler su PK era codigoAlquiler (que venía de la columna ALQUILER_CODIGO de la tabla maestra). Luego al migrar notamos que nos traía menos filas ya que esta columna no cumplía con las condiciones para ser PK, pues había varias filas con un mismo codigoAlquiler pero con distinto inquilino. Entonces, para solucionar este problema, tuvimos que modificar nuestro modelo, y ahora las entidades poseen una columna ID el cual es generado automáticamente de manera incremental.

Funciones

Decidimos crear 2 funciones, la primera tiene como parámetro de entrada el nombre de la moneda que es un VARCHAR(100) y lo convierte en su abreviatura que es la que se almacenará en nuestro modelo, siendo un VARCHAR(3).

```
CREATE FUNCTION GRUPO_1.ConvertirMoneda(@moneda VARCHAR(100))
RETURNS VARCHAR(3)
AS
BEGIN
    DECLARE @resultado VARCHAR(3);

    IF @moneda = 'Moneda Pesos'
    BEGIN
        SET @resultado = 'ARG';
    END
    ELSE IF @moneda = 'Moneda Dolares'
    BEGIN
        SET @resultado = 'USD';
    END
    ELSE
    BEGIN
        SET @resultado = NULL;
    END

    RETURN @resultado;
END
GO
```

La segunda función utilizada recibe como parámetro de entrada el nombre de una característica y retorna el valor de su id, nos fue útil en la migración de la tabla intermedia Caracteristicas_Inmuebles.

```
CREATE FUNCTION GRUPO_1.obtener_id_caracteristica(@caracteristica VARCHAR(100))
RETURNS INT
AS
BEGIN
    DECLARE @caracteristicaId INT;

    SELECT @caracteristicaId = idCaracteristica FROM GRUPO_1.caracteristicas
    WHERE nombre = @caracteristica;

    RETURN @caracteristicaId;
END
GO
```

Procedures

Optamos por crear procedimientos para cada tabla que necesitaba migrarse, lo que nos permitió organizar y simplificar el proceso. Cada procedimiento incluyó medidas para evitar duplicados y garantizar que no se perdieran datos durante la migración. Además, colocamos todo el proceso dentro de una transacción para mantener la integridad de los datos, asegurándonos de que todos los procedimientos se ejecuten correctamente o ninguno lo haga, evitando así problemas de consistencia. Este enfoque nos permitió llevar a cabo la migración de manera eficiente y segura.

```
BEGIN TRANSACTION;
EXECUTE GRUPO_1.migrar_Disposiciones;
EXECUTE GRUPO_1.migrar_Estados;
EXECUTE GRUPO_1.migrar_Orientaciones;
EXECUTE GRUPO_1.migrar_Caracteristicas;
EXECUTE GRUPO_1.migrar_TiposPeriodos;
EXECUTE GRUPO_1.migrar_EstadosAlquileres;
EXECUTE GRUPO_1.migrar_MediosDePago;
EXECUTE GRUPO_1.migrar_TiposInmuebles;
EXECUTE GRUPO_1.migrar_TipoOperaciones;
EXECUTE GRUPO_1.migrar_Barrrios;
EXECUTE GRUPO_1.migrar_Localidades;
EXECUTE GRUPO_1.migrar_Provincias;
EXECUTE GRUPO_1.migrar_CantAmbientes;
EXECUTE GRUPO_1.migrar_Monedas;
EXECUTE GRUPO_1.migrar_Propietarios;
EXECUTE GRUPO_1.migrar_Compradores;
EXECUTE GRUPO_1.migrar_Periodos;
EXECUTE GRUPO_1.migrar_Direcciones;
EXECUTE GRUPO_1.migrar_Sucursales;
EXECUTE GRUPO_1.migrar_Agentes;
EXECUTE GRUPO_1.migrar_Inmuebles;
EXECUTE GRUPO_1.migrar_Inquilinos;
EXECUTE GRUPO_1.migrar_Anuncios;
EXECUTE GRUPO_1.migrar_Alquileres;
EXECUTE GRUPO_1.migrar_Importes;
EXECUTE GRUPO_1.migrar_Ventas;
EXECUTE GRUPO_1.migrar_PagosVentas;
EXECUTE GRUPO_1.migrar_Caracteristicas_Inmuebles;
EXECUTE GRUPO_1.migrar_PagosAlquileres;
COMMIT TRANSACTION;
```