

UNIVERSIDAD TECNOLÓGICA NACIONAL

Facultad Regional Buenos Aires

# Gestión de Datos



## Trabajo Práctico Cuatrimestral

Docentes:

Ing. Edgardo Lacquaniti

Ing. Marcelo Adrian Moscuzza

### Grupo N°33: REYES DE DATOS

---

<b>BRU</b> , Nicolas	203.434-7
<b>BURSZTYN</b> , Alan	203.436-0
<b>SINTAS DE SOUSA BISPO</b> , Martín	204.075-0
<b>VILLALTA CARDENAS</b> , Carmen del Rosario	147.652-0

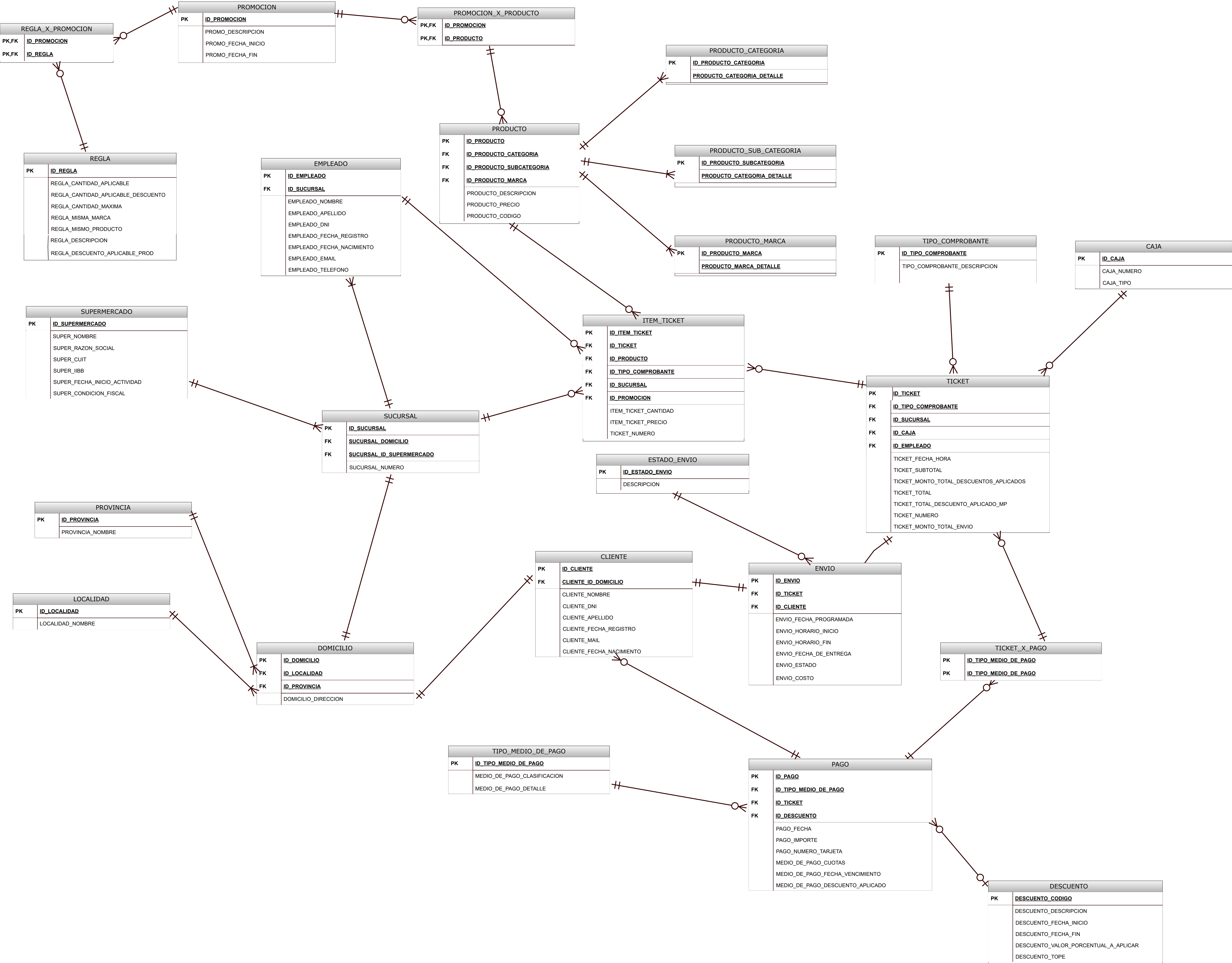
FECHA DE PRESENTACIÓN: 30 / 06 / 2024

## Índices

Modelo Transaccional del Sistema.....	
Diagrama Entidad Relación.....	
Producto Categoría.....	
Producto SubCategoría.....	
Producto Marca.....	
Cliente Contacto.....	
Tipo Medio De Pago.....	
Descuento.....	
Regla.....	
Supermercado.....	
Tipo Caja.....	
Tipo Comprobante.....	
Provincia.....	
Localidad.....	
Promoción.....	
Caja.....	
Domicilio.....	
Sucursal.....	
Cliente.....	
Empleado.....	
Producto.....	
Ticket.....	
Envío.....	
Item Ticket.....	
Pago.....	
Regla X Promoción .....	
Ticket X Pago.....	
Promoción X Item Ticket.....	
Promoción X Producto.....	

## Modelo Transaccional del Sistema

### Diagrama Entidad Relación



## Producto Categoría

Serán nuestros identificadores que tendrán nuestros productos informando a que categoría pertenece. Guardaremos:

```
CREATE TABLE REYES_DE_DATOS.Producto_categoria (  
    id_producto_categoria INT PRIMARY KEY IDENTITY(1,1),  
    producto_categoria_detalle VARCHAR(50) NOT NULL,  
);
```

Para hacer la migración definimos un insert que se encargará primero de obtener de la tabla maestra las distintas categorías de los productos los cuales encontraremos en la columna "PRODUCTO\_CATEGORIA". Como será la que usaremos para identificar cada categoría nueva, buscaremos aquellas que sean distintas de NULL.

```
INSERT INTO REYES_DE_DATOS.Producto_categoria(producto_categoria_detalle)  
    SELECT DISTINCT PRODUCTO_CATEGORIA  
    FROM gd_esquema.Maestra  
    WHERE PRODUCTO_CATEGORIA IS NOT NULL  
PRINT 'Migración de categoria terminada';
```

## Producto SubCategoría

Serán nuestros identificadores que tendrán nuestros productos informando a que subcategoría pertenece. Guardaremos:

```
CREATE TABLE REYES_DE_DATOS.Producto_subcategoria (  
    id_producto_subcategoria INT PRIMARY KEY IDENTITY(1,1),  
    producto_subcategoria_detalle VARCHAR(50) NOT NULL,  
);
```

Para hacer la migración definimos un insert que se encargará primero de obtener de la tabla maestra las distintas sub categorías de los productos los cuales encontraremos en la columna "PRODUCTO\_SUB\_CATEGORIA". Como será la que usaremos para identificar cada sub categoría nueva, buscaremos aquellas que sean distintas de NULL.

```
INSERT INTO REYES_DE_DATOS.Producto_subcategoria(producto_subcategoria_detalle)  
SELECT DISTINCT m.PRODUCTO_SUB_CATEGORIA  
FROM gd_esquema.Maestra m  
WHERE m.PRODUCTO_SUB_CATEGORIA IS NOT NULL  
PRINT 'Migración de subcategoria terminada';
```

## Producto Marca

Serán nuestros identificadores que tendrán nuestros productos informando a que marca pertenece. Guardaremos:

```
CREATE TABLE REYES_DE_DATOS.Producto_marca (  
    id_producto_marca INT PRIMARY KEY IDENTITY(1,1),  
    producto_marca_detalle VARCHAR(30) NOT NULL,  
);
```

Para hacer la migración definimos un insert que se encargará primero de obtener de la tabla maestra las distintas marcas de los productos los cuales encontraremos en la columna "PRODUCTO\_MARCA". Como será la que usaremos para identificar cada marca nueva, buscaremos aquellas que sean distintas de NULL.

```
INSERT INTO REYES_DE_DATOS.Producto_marca(producto_marca_detalle)  
    SELECT DISTINCT m.PRODUCTO_MARCA  
    FROM gd_esquema.Maestra m  
    WHERE m.PRODUCTO_MARCA IS NOT NULL  
PRINT 'Migración de marca terminada';
```

## Tipo Medio de Pago

Será el medio por el cual nuestro cliente pagará su compra. Guardaremos:

```
CREATE TABLE FRBA_SUPERMERCADO.Tipo_medio_de_pago (  
    id_tipo_medio_pago INT PRIMARY KEY IDENTITY(1,1),  
    medio_de_pago_clasificacion VARCHAR(100) UNIQUE NOT NULL, -- credito / debito / efectivo / etc  
    medio_de_pago_detalle VARCHAR(100) UNIQUE NOT NULL, -- visa / mastercard / efectivo / etc  
);
```

Para confeccionar la migración nos va a interesar de la tabla maestra obtener ciertos campos los cuales encontraremos en las columnas “PAGO\_TIPO\_MEDIO\_PAGO” y “PAGO\_MEDIO\_PAGO”.

Finalmente, nos aseguramos que “PAGO\_TIPO\_MEDIO\_PAGO” y “PAGO\_MEDIO\_PAGO” sean distintas de NULL

```
INSERT INTO REYES_DE_DATOS.Tipo_medio_de_pago(medio_de_pago_clasificacion, medio_de_pago_detalle)  
    SELECT DISTINCT m.PAGO_TIPO_MEDIO_PAGO, m.PAGO_MEDIO_PAGO  
    FROM gd_esquema.Maestra m  
    WHERE m.PAGO_TIPO_MEDIO_PAGO IS NOT NULL  
        AND m.PAGO_MEDIO_PAGO IS NOT NULL  
PRINT 'Migración de medio de pago terminada';
```

## Descuento

Será/n el/los descuento/s que se aplicará/n a nuestros productos. Guardaremos:

```
CREATE TABLE FRBA_SUPERMERCADO.Descuento (  
    descuento_codigo INT PRIMARY KEY NOT NULL,  
    descuento_descripcion VARCHAR(100) NOT NULL,  
    descuento_fecha_inicio DATE NOT NULL,  
    descuento_fecha_fin DATE NOT NULL,  
    descuento_valor_porcentual_a_aplicar DECIMAL(5, 2) NOT NULL,  
    descuento_tope DECIMAL(10, 2) NOT NULL,  
);
```

Para hacer la migración definimos un insert que se encargará primero de obtener de la tabla maestra los distintos descuentos de los productos los cuales encontraremos en las columnas “DESCUENTO\_CODIGO”, “DESCUENTO\_DESCRIPCION”, “DESCUENTO\_FECHA\_INICIO”, “DESCUENTO\_FECHA\_FIN”, “DESCUENTO\_PORCENTAJE\_DESC” y “DESCUENTO\_TOPE”. Teniendo esto en cuenta, insertamos los distintos descuentos de la tabla maestra en nuestra tabla. Por último, hay algunos casos en la tabla maestra donde el “DESCUENTO\_CODIGO” es NULL, por lo que nos encargamos de obviar esos casos.

```
INSERT INTO REYES_DE_DATOS.Descuento(  
    descuento_codigo,  
    descuento_descripcion,  
    descuento_fecha_inicio,  
    descuento_fecha_fin,  
    descuento_valor_porcentual_a_aplicar,  
    descuento_tope  
)  
SELECT DISTINCT m.DESCUENTO_CODIGO,  
    m.DESCUENTO_DESCRIPCION,  
    m.DESCUENTO_FECHA_INICIO,  
    m.DESCUENTO_FECHA_FIN,  
    m.DESCUENTO_PORCENTAJE_DESC,  
    m.DESCUENTO_TOPE  
FROM gd_esquema.Maestra m  
WHERE m.DESCUENTO_CODIGO IS NOT NULL  
PRINT 'Migración de descuento terminada';
```



## Regla

serán las reglas que aplicaremos sobre nuestros productos. Guardaremos:

```
CREATE TABLE FRBA_SUPERMERCADO.Regla (  
    id_regla INT PRIMARY KEY IDENTITY(1,1),  
    regla_cantidad_aplicable INT NOT NULL,  
    regla_descripcion VARCHAR(50) UNIQUE NOT NULL,  
    regla_cantidad_aplicable_descuento INT NOT NULL,  
    regla_cantidad_maxima INT NOT NULL,  
    regla_misma_marca BIT NOT NULL,  
    regla_mismo_producto BIT NOT NULL,  
    regla_descuento_aplicable_prod DECIMAL(3, 2) NOT NULL,  
);
```

Para hacer la migración definimos un insert que se encargará primero de obtener de la tabla maestra los distintos descuentos de los productos los cuales encontraremos en las columnas “REGLA\_DESCRIPCION”, “REGLA\_CANT\_APLICABLE\_REGLA”, “REGLA\_CANT\_APLICA\_DESCUENTO”, “REGLA\_CANT\_MAX\_PROD”, “REGLA\_APLICA\_MISMA\_MARCA”, “REGLA\_APLICA\_MISMO\_PROD” y “REGLA\_DESCUENTO\_APLICABLE\_PROD”. Teniendo esto en cuenta, insertamos las distintas reglas de la tabla maestra en nuestra tabla. Por último, hay algunos casos en la tabla maestra donde el “REGLA\_DESCRIPCION” es NULL, por lo que nos encargamos de obviar esos casos.

```
INSERT INTO REYES_DE_DATOS.Regla(  
    regla_descripcion,  
    regla_cantidad_aplicable,  
    regla_cantidad_aplicable_descuento,  
    regla_cantidad_maxima,  
    regla_misma_marca,  
    regla_mismo_producto,  
    regla_descuento_aplicable_prod  
)  
SELECT DISTINCT  
    m.REGLA_DESCRIPCION,  
    m.REGLA_CANT_APLICABLE_REGLA,  
    m.REGLA_CANT_APLICA_DESCUENTO,  
    m.REGLA_CANT_MAX_PROD,  
    m.REGLA_APLICA_MISMA_MARCA,  
    m.REGLA_APLICA_MISMO_PROD,  
    m.REGLA_DESCUENTO_APLICABLE_PROD  
FROM gd_esquema.Maestra m  
WHERE m.REGLA_DESCRIPCION IS NOT NULL  
PRINT 'Migración de regla terminada';
```

## Supermercado

Serán nuestros supermercados. Guardaremos:

```
CREATE TABLE REYES_DE_DATOS.Supermercado (  
    id_supermercado INT PRIMARY KEY IDENTITY(1,1),  
    super_nombre VARCHAR(50) UNIQUE NOT NULL,  
    super_razon_social VARCHAR(100) NOT NULL,  
    super_cuit VARCHAR(30) NOT NULL,  
    super_iibb VARCHAR(30) NOT NULL, --Ingr. Brut. N°: 133452135  
    super_fecha_inicio_actividad DATE NOT NULL,  
    super_condicion_fiscal VARCHAR(50) NOT NULL  
);
```

Para hacer la migración definimos un insert que se encargará primero de obtener de la tabla maestra los distintos supermercados los cuales encontraremos en las columnas “SUPER\_NOMBRE”, “SUPER\_RAZON\_SOC”, “SUPER\_CUIT”, “SUPER\_IIBB”, “SUPER\_FECHA\_INI\_ACTIVIDAD” y “SUPER\_CONDICION\_FISCAL”

. Teniendo esto en cuenta, insertamos los distintos supermercados de la tabla maestra en nuestra tabla. Por último, hay algunos casos en la tabla maestra donde el “SUPER\_CUIT” es NULL, por lo que nos encargamos de obviar esos casos.

```
INSERT INTO REYES_DE_DATOS.Supermercado(  
    super_nombre,  
    super_razon_social,  
    super_cuit,  
    super_iibb,  
    super_fecha_inicio_actividad,  
    super_condicion_fiscal  
)  
SELECT DISTINCT m.SUPER_NOMBRE,  
    m.SUPER_RAZON_SOC,  
    m.SUPER_CUIT,  
    m.SUPER_IIBB,  
    m.SUPER_FECHA_INI_ACTIVIDAD,  
    m.SUPER_CONDICION_FISCAL  
FROM gd_esquema.Maestra m  
WHERE m.SUPER_CUIT IS NOT NULL  
PRINT 'Migración de supermercado terminada';
```

## Tipo Comprobante

Será el tipo de comprobante de nuestro ticket al momento de efectuar la compra por parte del cliente. Guardaremos:

```
CREATE TABLE REYES_DE_DATOS.Tipo_Comprobante(  
    id_tipo_comprobante INT PRIMARY KEY IDENTITY(1,1),  
    tipo_comprobante_nombre VARCHAR (1) NOT NULL, --caracter  
);
```

Para hacer la migración definimos un insert que se encargará de obtener de la tabla maestra los distintos tipos de comprobantes los cuales encontraremos en la columna "TICKET\_TIPO\_COMPROBANTE".

Teniendo esto en cuenta, insertamos los distintos tipos de comprobante de la tabla maestra en nuestra tabla. Por último, hay algunos casos en la tabla maestra donde el "TICKET\_TIPO\_COMPROBANTE" es NULL, por lo que nos encargamos de obviar esos casos.

```
INSERT INTO REYES_DE_DATOS.Tipo_Comprobante(tipo_comprobante_nombre)  
    SELECT DISTINCT TICKET_TIPO_COMPROBANTE  
    FROM gd_esquema.Maestra  
    WHERE TICKET_TIPO_COMPROBANTE IS NOT NULL  
PRINT 'Migración de tipo de comprobante terminada';
```

## Provincia

Serán las provincias donde pertenezcan tanto los supermercados como los clientes y los empleados. Guardaremos:

```
CREATE TABLE FRBA_SUPERMERCADO.Provincia(  
    id_provincia INT PRIMARY KEY IDENTITY(1,1),  
    provincia_nombre VARCHAR(50) UNIQUE NOT NULL,  
);
```

El objetivo principal de este insert es migrar datos de provincias desde la tabla maestra.

El procedimiento inicia seleccionando los nombres de provincias desde distintas columnas de la tabla maestra. Específicamente, se extraen los valores de las columnas "CLIENTE\_PROVINCIA", "SUCURSAL\_PROVINCIA" y "SUPER\_PROVINCIA".

A su vez, verificamos que el "CLIENTE\_PROVINCIA", "SUCURSAL\_PROVINCIA" y "SUPER\_PROVINCIA" sean distintos de NULL.

```
INSERT INTO REYES_DE_DATOS.Provincia(provincia_nombre)  
    SELECT DISTINCT CLIENTE_PROVINCIA  
    FROM gd_esquema.Maestra  
    WHERE CLIENTE_PROVINCIA IS NOT NULL  
    UNION SELECT SUCURSAL_PROVINCIA AS provincia  
    FROM gd_esquema.Maestra  
    WHERE SUCURSAL_PROVINCIA IS NOT NULL  
    UNION SELECT SUPER_PROVINCIA AS provincia  
    FROM gd_esquema.Maestra  
    WHERE SUPER_PROVINCIA IS NOT NULL;  
PRINT 'Migración de Provincia terminada';
```

## Localidad

Serán las localidades donde pertenezcan tanto los supermercados como los clientes y los empleados. Guardaremos:

```
CREATE TABLE FRBA_SUPERMERCADO.Localidad(  
    id_localidad INT PRIMARY KEY IDENTITY(1,1),  
    localidad_nombre VARCHAR(50) UNIQUE NOT NULL,  
);
```

El objetivo principal de este insert es migrar datos de localidades desde la tabla maestra.

El procedimiento inicia seleccionando los nombres de localidades desde distintas columnas de la tabla maestra. Específicamente, se extraen los valores de las columnas "CLIENTE\_LOCALIDAD", "SUCURSAL\_LOCALIDAD" y "SUPER\_LOCALIDAD".

A su vez, verificamos que el "CLIENTE\_LOCALIDAD", "SUCURSAL\_LOCALIDAD" y "SUPER\_LOCALIDAD" sean distintos de NULL.

```
INSERT INTO REYES_DE_DATOS.Localidad(localidad_nombre)  
    SELECT DISTINCT CLIENTE_LOCALIDAD  
    FROM gd_esquema.Maestra  
    WHERE CLIENTE_LOCALIDAD IS NOT NULL  
    UNION SELECT SUCURSAL_LOCALIDAD  
    FROM gd_esquema.Maestra  
    WHERE SUCURSAL_LOCALIDAD IS NOT NULL  
    UNION SELECT SUPER_LOCALIDAD  
    FROM gd_esquema.Maestra  
    WHERE SUPER_LOCALIDAD IS NOT NULL;  
PRINT 'Migración de Localidad terminada';
```

## Promoción

Serán las promociones que se aplicarán a los diversos productos. Guardaremos:

```
CREATE TABLE REYES_DE_DATOS.Promocion(  
    id_promo INT PRIMARY KEY NOT NULL,  
    promo_descripcion VARCHAR(50) NOT NULL,  
    promo_fecha_inicio DATETIME NOT NULL,  
    promo_fecha_fin DATETIME NOT NULL,  
);
```

Para hacer la migración definimos un insert que se encargará de obtener de la tabla maestra las distintas promociones de las cuales encontraremos en las columnas “PROMO\_CODIGO”, “PROMOCION\_DESCRIPCION”, “PROMOCION\_FECHA\_INICIO”, “PROMOCION\_FECHA\_FIN” y “PROMO\_APLICADA\_DESCUENTO”.

Teniendo esto en cuenta, insertamos las distintas promociones de la tabla maestra en nuestra tabla. Por último, hay algunos casos en la tabla maestra donde el “PROMO\_CODIGO” es NULL, por lo que nos encargamos de obviar esos casos.

```
INSERT INTO REYES_DE_DATOS.Promocion(  
    id_promo,  
    promo_descripcion,  
    promo_fecha_inicio,  
    promo_fecha_fin  
)  
SELECT DISTINCT  
    PROMO_CODIGO,  
    PROMOCION_DESCRIPCION,  
    PROMOCION_FECHA_INICIO,  
    PROMOCION_FECHA_FIN  
FROM gd_esquema.Maestra  
WHERE PROMO_CODIGO IS NOT NULL  
PRINT 'Migración de promocion terminada';
```

## Caja

será la caja que se encargue de gestionar el ticket de la compra dentro del Supermercado. Guardaremos:

```
CREATE TABLE REYES_DE_DATOS.Caja(  
    id_caja INT PRIMARY KEY IDENTITY(1,1),  
    caja_numero INT NOT NULL,  
    caja_tipo VARCHAR(30) NOT NULL,  
);
```

Para hacer la migración definimos un insert que se encargará de obtener de la tabla maestra las cajas de las cuales encontraremos en las columnas “CAJA\_NUMERO”.

Teniendo esto en cuenta, insertamos las distintas cajas de la tabla maestra en nuestra tabla. Por último, hay algunos casos en la tabla maestra donde el “CAJA\_NUMERO” es NULL, por lo que nos encargamos de obviar esos casos.

```
INSERT INTO REYES_DE_DATOS.Caja(  
    caja_numero,  
    caja_tipo  
)  
SELECT DISTINCT  
    m.CAJA_NUMERO,  
    m.CAJA_TIPO  
FROM gd_esquema.Maestra m  
WHERE CAJA_NUMERO IS NOT NULL  
PRINT 'Migración de caja terminada';
```

## Domicilio

Será el domicilio tanto del cliente como de la sucursal. Guardaremos:

```
CREATE TABLE REYES_DE_DATOS.Domicilio (  
    id_domicilio INT PRIMARY KEY IDENTITY(1,1),  
    id_localidad INT NOT NULL,  
    id_provincia INT NOT NULL,  
    domicilio_direccion VARCHAR(100) NOT NULL  
);
```

Para confeccionar la migración nos va a interesar de la tabla maestra obtener los domicilios los cuales encontraremos en las columnas “SUPER\_DOMICILIO” y “SUPER\_PROVINCIA”.

```
INSERT INTO REYES_DE_DATOS.Domicilio (  
    domicilio_direccion,  
    id_localidad,  
    id_provincia  
)  
SELECT DISTINCT  
    m.SUPER_DOMICILIO,  
    l.id_localidad,  
    p.id_provincia  
FROM gd_esquema.Maestra m  
    JOIN REYES_DE_DATOS.Localidad l ON l.localidad_nombre = m.SUPER_LOCALIDAD  
    JOIN REYES_DE_DATOS.Provincia p ON p.provincia_nombre = m.SUPER_PROVINCIA  
WHERE SUPER_DOMICILIO IS NOT NULL;  
PRINT 'Migracion de Domicilio de supermercado terminada';
```

```
INSERT INTO REYES_DE_DATOS.Domicilio (  
    domicilio_direccion,  
    id_localidad,  
    id_provincia  
)  
SELECT DISTINCT  
    m.CLIENTE_DOMICILIO,  
    l.id_localidad,  
    p.id_provincia  
FROM gd_esquema.Maestra m  
    JOIN REYES_DE_DATOS.Localidad l ON l.localidad_nombre = m.CLIENTE_LOCALIDAD  
    JOIN REYES_DE_DATOS.Provincia p ON p.provincia_nombre = m.CLIENTE_PROVINCIA  
WHERE CLIENTE_DOMICILIO IS NOT NULL;  
PRINT 'Migracion de Domicilio de cliente terminada';
```



## Sucursal

Serán las distintas sucursales de supermercados que tendremos en nuestro sistema que también figurarán tanto para los empleados como para los tickets.

Guardaremos:

```
CREATE TABLE REYES_DE_DATOS.Sucursal (  
    id_sucursal INT PRIMARY KEY IDENTITY(1,1),  
    sucursal_id_supermercado INT NOT NULL,  
    sucursal_domicilio VARCHAR(100) NOT NULL,  
    sucursal_numero VARCHAR(50) NOT NULL -- solo el numero  
);
```

Para confeccionar la migración nos va a interesar de la tabla maestra obtener las sucursales las cuales encontraremos en las columnas “SUPER\_DIRECCION” “SUCURSAL\_NOMBRE” y haremos un JOIN para verificar que el supermercado (la sucursal en cuestión) tenga mismo CUIT. Básicamente nos aseguramos que sea la misma.

```
INSERT INTO REYES_DE_DATOS.Sucursal (  
    sucursal_domicilio,  
    sucursal_id_supermercado,  
    sucursal_numero -- solo el numero  
)  
SELECT DISTINCT  
    m.SUCURSAL_DIRECCION,  
    s.id_supermercado,  
    m.SUCURSAL_NOMBRE  
FROM gd_esquema.Maestra m  
    JOIN REYES_DE_DATOS.Supermercado s ON m.SUPER_NOMBRE = s.super_nombre  
PRINT 'Migración de sucursal terminada';
```

## Cliente

Serán nuestros clientes que compren productos en los supermercados.

```
CREATE TABLE REYES_DE_DATOS.Cliente (  
    id_cliente INT PRIMARY KEY IDENTITY(1,1),  
    cliente_dni INT NOT NULL,  
    cliente_id_domicilio INT NOT NULL,  
    cliente_nombre VARCHAR(100) NOT NULL,  
    cliente_apellido VARCHAR(100) NOT NULL,  
    cliente_fecha_registro DATE NOT NULL,  
    cliente_mail VARCHAR(100) NOT NULL,  
    cliente_fecha_nacimiento DATE NOT NULL  
);
```

Para hacer la migración definimos un insert que se encargará de obtener de la tabla maestra los distintos clientes de los cuales encontraremos en las columnas

“CLIENTE\_DNI”, “CLIENTE\_DOMICILIO”,  
“CLIENTE\_NOMBRE”, “CLIENTE\_APELLIDO”, “CLIENTE\_FECHA\_REGISTRO”,  
“CLIENTE\_MAIL” y “CLIENTE\_FECHA\_NACIMIENTO”.

A su vez, realizaremos un JOIN con “CLIENTE\_DOMICILIO” para verificar que la dirección de nuestro cliente coincide con el valor del campo “domicilio\_direccion” de nuestra tabla domicilio.

Teniendo esto en cuenta, insertamos los distintos clientes de la tabla maestra en nuestra tabla. Por último, hay algunos casos en la tabla maestra donde el “CLIENTE\_DNI” es NULL, por lo que nos encargamos de obviar esos casos.

```
INSERT INTO REYES_DE_DATOS.Cliente(  
    cliente_dni,  
    cliente_id_domicilio,  
    cliente_nombre,  
    cliente_apellido,  
    cliente_fecha_registro,  
    cliente_mail,  
    cliente_fecha_nacimiento  
)  
SELECT DISTINCT  
    m.CLIENTE_DNI,  
    d.id_domicilio,  
    m.CLIENTE_NOMBRE,  
    m.CLIENTE_APELLIDO,  
    m.CLIENTE_FECHA_REGISTRO,  
    m.CLIENTE_MAIL,  
    m.CLIENTE_FECHA_NACIMIENTO  
FROM gd_esquema.Maestra m  
    JOIN REYES_DE_DATOS.Domicilio d ON m.CLIENTE_DOMICILIO = d.domicilio_direccion  
WHERE m.CLIENTE_DNI IS NOT NULL  
PRINT 'Migración de cliente terminada';
```

## Empleado

serán los empleados que trabajen en los supermercados. Guardaremos:

```
CREATE TABLE REYES_DE_DATOS.Empleado (  
    id_empleado INT PRIMARY KEY IDENTITY(1,1),  
    id_sucursal INT NOT NULL,  
    empleado_nombre VARCHAR(15) NOT NULL,  
    empleado_apellido VARCHAR(30) NOT NULL,  
    empleado_dni INT NOT NULL,  
    empleado_fecha_registro DATE,  
    empleado_fecha_nacimiento DATE,  
    empleado_email VARCHAR(30) NOT NULL,  
    empleado_telefono DECIMAL(8, 0) NOT NULL  
);
```

Para hacer la migración definimos un insert que se encargará de obtener de la tabla maestra los distintos empleados de los cuales encontraremos en las columnas:

“SUCURSAL\_DIRECCION”,  
“EMPLEADO\_NOMBRE”, “EMPLEADO\_APELLIDO”, “EMPLEADO\_DNI”,  
“EMPLEADO\_FECHA\_REGISTRO”, “CLIENTE\_FECHA\_REGISTRO”,  
“EMPLEADO\_MAIL”y “EMPLEADO\_TELEFONO”

Teniendo esto en cuenta, insertamos los distintos clientes de la tabla maestra en nuestra tabla. Por último, hay algunos casos en la tabla maestra donde“EMPLEADO\_DNI” es NULL.

```
INSERT INTO REYES_DE_DATOS.Empleado(  
    id_sucursal,  
    empleado_nombre,  
    empleado_apellido,  
    empleado_dni,  
    empleado_fecha_registro,  
    empleado_fecha_nacimiento,  
    empleado_email,  
    empleado_telefono  
)  
SELECT DISTINCT  
    s.id_sucursal,  
    m.EMPLEADO_NOMBRE,  
    m.EMPLEADO_APELLIDO,  
    m.EMPLEADO_DNI,  
    m.EMPLEADO_FECHA_REGISTRO,  
    m.CLIENTE_FECHA_REGISTRO,  
    m.EMPLEADO_MAIL,  
    m.EMPLEADO_TELEFONO  
FROM gd_esquema.Maestra m  
    JOIN REYES_DE_DATOS.Sucursal s ON m.SUCURSAL_NOMBRE = s.sucursal_numero  
WHERE m.EMPLEADO_DNI IS NOT NULL  
PRINT 'Migración de empleado terminada';
```

## Producto

Conformado por los productos que vende el supermercado los cuales tendrán asociados su categoría, su subcategoría y su marca mediante una foreign key para cada una de ellas.

```
CREATE TABLE REYES_DE_DATOS.Producto (  
    id_producto INT PRIMARY KEY IDENTITY(1,1),  
    producto_codigo NVARCHAR(100) NOT NULL, -- PRODUCTO_NOMBRE  
    id_producto_categoria INT NOT NULL,  
    id_producto_subcategoria INT NOT NULL,  
    id_marca INT NOT NULL,  
    producto_descripcion NVARCHAR(100) NOT NULL,  
    producto_precio DECIMAL(10, 2) NOT NULL  
);
```

Para hacer la migración definimos un insert que se encargará de obtener de la tabla maestra los distintos productos de los cuales encontraremos en las columnas:

“PRODUCTO\_NOMBRE”, “PRODUCTO\_CATEGORIA”,  
“PRODUCTO\_SUB\_CATEGORIA”, “PRODUCTO\_MARCA”,  
“PRODUCTO\_DESCRIPCION” y “PRODUCTO\_PRECIO”.

También tendremos 3 JOINS los cuales nos servirán para relacionar de nuestras tablas Producto\_categoria(con el campo PRODUCTO\_CATEGORIA),  
Producto\_subcategoria (con el campo de PRODUCTO\_SUB\_CATEGORIA) y  
Producto\_marca (con el campo de PRODUCTO\_MARCA)

Teniendo esto en cuenta, insertamos los distintos productos de la tabla maestra en nuestra tabla. Por último, hay algunos casos en la tabla maestra donde el “PRODUCTO\_NOMBRE” es NULL, por lo que nos encargamos de obviar esos casos.

```
INSERT INTO REYES_DE_DATOS.Producto(
    producto_codigo, -- PRODUCTO_NOMBRE
    id_producto_categoria,
    id_producto_subcategoria,
    id_marca,
    producto_descripcion,
    producto_precio
)
SELECT DISTINCT
    m.PRODUCTO_NOMBRE,
    c.id_producto_categoria,
    sc.id_producto_subcategoria,
    mc.id_producto_marca,
    m.PRODUCTO_DESCRIPCION,
    m.PRODUCTO_PRECIO
FROM gd_esquema.Maestra m
    JOIN REYES_DE_DATOS.Producto_categoria c ON m.PRODUCTO_CATEGORIA = c.producto_categoria_detalle
    JOIN REYES_DE_DATOS.Producto_subcategoria sc ON m.PRODUCTO_SUB_CATEGORIA = sc.producto_subcategoria_detalle
    JOIN REYES_DE_DATOS.Producto_marca mc ON m.PRODUCTO_MARCA = mc.producto_marca_detalle
WHERE m.PRODUCTO_NOMBRE IS NOT NULL
PRINT 'Migración de producto terminada';
```

Ticket

Será el ticket que se generará al realizar una compra por parte del cliente los cuales tendrán asociados su tipo de comprobante, sucursal, caja y empleado mediante una foreign key para cada una de ellas.

```
CREATE TABLE REYES_DE_DATOS.Ticket(  
    id_ticket INT PRIMARY KEY IDENTITY(1,1),  
    ticket_numero INT NOT NULL, -- TICKET_NUMERO  
    id_tipo_comprobante INT NOT NULL,  
    id_sucursal INT NOT NULL,  
    id_caja INT NOT NULL,  
    id_empleado INT NOT NULL,  
    ticket_fecha_hora DATE NOT NULL,  
    ticket_subtotal DECIMAL(10, 2) NOT NULL,  
    ticket_total DECIMAL(10, 2) NOT NULL,  
    ticket_total_descuento_aplicado INT NOT NULL,  
    ticket_total_descuento_aplicado_mp INT NOT NULL,  
    ticket_monto_total_envio INT NOT NULL  
);
```

Para hacer la migración definimos un insert que se encargará de obtener de la tabla maestra los distintos tickets de los cuales encontraremos en las columnas:

"TICKET\_NUMERO", "SUCURSAL\_DIRECCION", "EMPLEADO\_MAIL",  
"TICKET\_FECHA\_HORA", "TICKET\_SUBTOTAL\_PRODUCTOS",  
"TICKET\_TOTAL\_DESCUENTO\_APLICADO"  
"TICKET\_TOTAL\_DESCUENTO\_APLICADO\_MP" y  
"TICKET\_TOTAL\_ENVIO".

A su vez, haremos cuatro JOINS para verificar que el ticket pertenezca a la Caja y al Tipo\_Comprobante, Empleado y Sucursal; que son tablas ya creadas más arriba. Básicamente nos aseguramos que sea la misma.

Teniendo esto en cuenta, insertamos los distintos tickets de la tabla maestra en nuestra tabla. Por último, hay algunos casos en la tabla maestra donde el "TICKET\_NUMERO" es NULL, por lo que nos encargamos de obviar esos casos.

```

INSERT INTO REYES_DE_DATOS.Ticket(
    ticket_numero,
    id_tipo_comprobante,
    id_sucursal,
    id_caja,
    id_empleado,
    ticket_fecha_hora,
    ticket_subtotal,
    ticket_total,
    ticket_total_descuento_aplicado,
    ticket_total_descuento_aplicado_mp,
    ticket_monto_total_envio
)
SELECT DISTINCT
    m.TICKET_NUMERO,
    tc.id_tipo_comprobante,
    sc.id_sucursal,
    c.id_caja,
    e.id_empleado,
    m.TICKET_FECHA_HORA,
    m.TICKET_SUBTOTAL_PRODUCTOS,
    m.TICKET_TOTAL_TICKET,
    m.TICKET_TOTAL_DESCUENTO_APLICADO,
    m.TICKET_TOTAL_DESCUENTO_APLICADO_MP,
    m.TICKET_TOTAL_ENVIO
FROM gd_esquema.Maestra m
    JOIN REYES_DE_DATOS.Tipo_Comprobante tc ON m.TICKET_TIPO_COMPROBANTE = tc.tipo_comprobante_nombre
    JOIN REYES_DE_DATOS.Empleado e ON m.EMPLEADO_DNI = e.empleado_dni
    JOIN REYES_DE_DATOS.Sucursal sc ON m.SUCURSAL_NOMBRE = sc.sucursal_numero
    JOIN REYES_DE_DATOS.Caja c ON m.CAJA_NUMERO = c.caja_numero AND m.CAJA_TIPO = c.caja_tipo
WHERE m.TICKET_NUMERO IS NOT NULL
ORDER BY m.TICKET_NUMERO
PRINT 'Migración de Ticket terminada';

```

## Envío

Será el envío que se va a realizar, tendremos asociadas el ticket y el cliente respectivo asociadas mediante foreign key.

```
CREATE TABLE REYES_DE_DATOS.Envío (  
    id_envio INT PRIMARY KEY IDENTITY(1,1),  
    id_ticket INT NOT NULL,  
    id_cliente INT NOT NULL,  
    envio_fecha_programada DATETIME NULL,  
    envio_horario_inicio INT NOT NULL,  
    envio_horario_fin INT NOT NULL,  
    envio_fecha_entrega DATETIME NOT NULL,  
    envio_estado VARCHAR(50) NOT NULL,  
    envio_costo DECIMAL(10, 2) NOT NULL,  
);
```

Para hacer la migración definimos un insert que se encargará de obtener de la tabla maestra los distintos envíos de los cuales encontraremos en las columnas: "TICKET\_NUMERO", "CLIENTE\_DNI", "ENVIO\_FECHA\_PROGRAMADA", "ENVIO\_HORA\_INICIO", "ENVIO\_HORA\_FIN", "ENVIO\_FECHA\_ENTREGA", "ENVIO\_ESTADO" y "ENVIO\_COSTO".

Teniendo esto en cuenta, insertamos los distintos envíos de la tabla maestra en nuestra tabla. Por último, al ser el envío algo optativo, hay casos en los que en la tabla maestra "ENVIO\_HORA\_INICIO", "ENVIO\_HORA\_FIN" y "ENVIO\_FECHA\_ENTREGA" tiene valores NULL, por lo que nos encargamos de obviar esos casos.



```
INSERT INTO REYES_DE_DATOS.Envio(  
    id_ticket,  
    id_cliente,  
    envio_fecha_programada,  
    envio_horario_inicio,  
    envio_horario_fin,  
    envio_fecha_entrega,  
    envio_estado,  
    envio_costo  
)  
SELECT  
    m.TICKET_NUMERO,  
    c.id_cliente,  
    m.ENVIO_FECHA_PROGRAMADA,  
    m.ENVIO_HORA_INICIO,  
    m.ENVIO_HORA_FIN,  
    m.ENVIO_FECHA_ENTREGA,  
    m.ENVIO_ESTADO,  
    m.ENVIO_COSTO  
FROM gd_esquema.Maestra m  
    JOIN REYES_DE_DATOS.Cliente c ON m.CLIENTE_MAIL = c.cliente_mail  
WHERE m.TICKET_NUMERO IS NOT NULL  
    AND m.ENVIO_HORA_INICIO IS NOT NULL  
PRINT 'Migración de envio terminada';
```

## Item Ticket

Serán los ítems provenientes del ticket de nuestra venta, tendremos asociados el ticket, el producto, el tipo comprobante, la sucursal y la promoción asociadas mediante foreign key

```
CREATE TABLE REYES_DE_DATOS.Item_Ticket (  
    id_item_ticket INT PRIMARY KEY IDENTITY(1,1),  
    ticket_numero VARCHAR(50) NOT NULL,  
    id_producto INT NOT NULL,  
    id_tipo_comprobante INT NOT NULL,  
    id_sucursal INT NOT NULL,  
    id_promocion INT,  
    item_ticket_cantidad INT NOT NULL,  
    item_ticket_precio INT NOT NULL  
);
```

Para hacer la migración definimos un insert que se encargará de obtener de la tabla maestra los distintos ítem tickets provenientes de cada ticket de los cuales encontraremos en las columnas:

“TICKET\_NUMERO”, “PRODUCTO\_NOMBRE”,  
“TICKET\_TIPO\_COMPROBANTE”, “SUCURSAL\_DIRECCION”,  
“PROMO\_CODIGO”, “TICKET\_DET\_CANTIDAD” y “TICKET\_DET\_PRECIO”

A su vez, haremos dos JOIN para verificar que el ítem del ticket pertenezca al Tipo\_Comprobante, que es una tabla ya creada más arriba. Básicamente nos aseguramos que sea la misma.

Teniendo esto en cuenta, insertamos los distintos envíos de la tabla maestra en nuestra tabla. Finalmente, verificamos que “TICKET\_NUMERO” sea distinto de NULL.

```

INSERT INTO REYES_DE_DATOS.Item_Ticket(
    ticket_numero,
    id_producto,
    id_tipo_comprobante,
    id_sucursal,
    id_promocion,
    item_ticket_cantidad,
    item_ticket_precio
)
SELECT DISTINCT
    m.TICKET_NUMERO,
    p.id_producto,
    tc.id_tipo_comprobante,
    s.id_sucursal,
    m.PROMO_CODIGO,
    m.TICKET_DET_CANTIDAD,
    m.TICKET_DET_PRECIO
FROM gd_esquema.Maestra m
    JOIN REYES_DE_DATOS.Sucursal s ON m.SUCURSAL_DIRECCION = s.sucursal_domicilio
    JOIN REYES_DE_DATOS.Tipo_Comprobante tc ON m.TICKET_TIPO_COMPROBANTE = tc.tipo_comprobante_nombre
    JOIN REYES_DE_DATOS.Producto p ON m.PRODUCTO_PRECIO = p.producto_precio AND m.PRODUCTO_NOMBRE = p.producto_codigo
WHERE m.TICKET_NUMERO IS NOT NULL
    AND m.PRODUCTO_NOMBRE IS NOT NULL
    AND m.TICKET_DET_CANTIDAD IS NOT NULL
    AND m.TICKET_DET_PRECIO IS NOT NULL
PRINT 'Migración de ítem ticket terminada';

```

## Pago

Serán las transacciones efectuadas por el cliente, tendremos asociados el tipo de medio de pago, el cliente y el descuento; con una foreign key asociada a cada uno. Hay que tener en cuenta que el pago\_numero\_tarjeta puede admitir valores nulos ya que se puede efectuar el pago con efectivo.

```
CREATE TABLE REYES_DE_DATOS.Pago(  
    id_pago INT PRIMARY KEY IDENTITY(1,1),  
    id_tipo_medio_de_pago INT NOT NULL,  
    id_descuento INT NOT NULL,  
    pago_fecha DATETIME NOT NULL,  
    pago_importe DECIMAL(15,2) NOT NULL,  
    pago_numero_tarjeta VARCHAR(20), -- SON OPCIONALES PORQUE PUEDE PAGAR EN EFECTIVO  
    medio_de_pago_cuotas INT,  
    medio_de_pago_fecha_vencimiento DATETIME,  
    medio_de_pago_descuento_aplicado DECIMAL(10, 2),  
);
```

Para hacer la migración definimos un insert que se encargará de obtener de la tabla maestra los distintos pagos los cuales encontraremos en las columnas:

“CLIENTE\_DNI”, “DESCUENTO\_CODIGO”, “PAGO\_FECHA”,  
“PAGO\_IMPORTE”, “PAGO\_TARJETA\_NRO”, “PAGO\_TARJETA\_CUOTAS”,  
“PAGO\_TARJETA\_FECHA\_VENC”, “PAGO\_DESCUENTO\_APLICADO” y  
“PAGO\_MEDIO\_PAGO”

A su vez, haremos un JOIN para verificar que el pago pertenezca al Tipo\_medio\_de\_pago, que es una tabla ya creada más arriba. Básicamente nos aseguramos que sea la misma.

Teniendo esto en cuenta, insertamos los distintos pagos de la tabla maestra en nuestra tabla. Finalmente, verificamos que “TICKET\_NUMERO” sea distinto de NULL.

```
INSERT INTO REYES_DE_DATOS.Pago(  
    id_tipo_medio_de_pago,  
    id_descuento,  
    pago_fecha,  
    pago_importe,  
    pago_numero_tarjeta,  
    medio_de_pago_cuotas,  
    medio_de_pago_fecha_vencimiento,  
    medio_de_pago_descuento_aplicado  
)  
SELECT DISTINCT  
    mp.id_tipo_medio_pago,  
    m.DESCUENTO_CODIGO,  
    m.PAGO_FECHA,  
    m.PAGO_IMPORTE,  
    m.PAGO_TARJETA_NRO,  
    m.PAGO_TARJETA_CUOTAS,  
    m.PAGO_TARJETA_FECHA_VENC,  
    m.PAGO_DESCUENTO_APLICADO  
FROM gd_esquema.Maestra m  
    JOIN REYES_DE_DATOS.Tipo_medio_de_pago mp ON m.PAGO_MEDIO_PAGO = mp.medio_de_pago_detalle  
PRINT 'Migración de Pago terminada';
```

## Regla X Promoción

Corresponderá a las distintas combinaciones que se pueden dar entre reglas y promociones, dado que una regla puede tener distintas promociones. Tendrá asociada tanto la promoción como la regla mediante una foreign key para cada una.

```
CREATE TABLE REYES_DE_DATOS.Regla_x_Promocion (  
    id_promocion INT NOT NULL,  
    id_regla INT NOT NULL,  
);
```

Para hacer la migración definimos un insert que se encargará de obtener de la tabla maestra las distintas reglas para cada promoción las cuales encontraremos en las columnas: “PROMO\_CODIGO” y “REGLA\_DESCRIPCION”

Haremos uso de una función que nos facilitará obtener la “REGLA\_DESCRIPCION”

Teniendo esto en cuenta, insertamos las distintas reglas por promociones de la tabla maestra en nuestra tabla. Finalmente, verificamos que “PROMO\_CODIGO” sea distinto de NULL.

```
INSERT INTO REYES_DE_DATOS.Regla_x_Promocion(  
    id_promocion,  
    id_regla  
)  
SELECT DISTINCT  
    m.PROMO_CODIGO,  
    r.id_regla  
FROM gd_esquema.Maestra m  
    JOIN REYES_DE_DATOS.Regla r ON m.REGLA_DESCRIPCION = r.regla_descripcion  
WHERE m.PROMO_CODIGO IS NOT NULL AND 2 IS NOT NULL  
PRINT 'Migración de regla x promocion terminada';
```

## Ticket X Pago

Corresponderá a las distintas combinaciones que se pueden dar entre los tickets y los pagos. Tendrá asociado tanto el ticket como el pago mediante una foreign key para cada una.

```
CREATE TABLE REYES_DE_DATOS.Ticket_X_Pago(  
    id_ticket INT NOT NULL,  
    id_pago INT NOT NULL,  
);
```

Para hacer la migración definimos un insert que se encargará de obtener de la tabla maestra los distintos tickets por cada pago realizado los cuales encontraremos en la columna: "TICKET\_NUMERO"

A su vez, haremos un JOIN para verificar que el ticket de cada pago pertenezca al Pago, y estableciendo que el "CLIENTE\_DNI" coincida con el id\_cliente de nuestro Pago. También que coincida el día en el que realiza.

Teniendo esto en cuenta, insertamos los distintos envíos tickets por los medios de pago de la tabla maestra en nuestra tabla. Finalmente, verificamos que "TICKET\_NUMERO" sea distinto de NULL.

```
INSERT INTO REYES_DE_DATOS.Ticket_X_Pago(  
    id_ticket,  
    id_pago  
)  
SELECT DISTINCT  
    t.id_ticket,  
    p.id_pago  
FROM gd_esquema.Maestra m  
    JOIN REYES_DE_DATOS.Ticket t ON m.TICKET_NUMERO = t.ticket_numero  
    JOIN REYES_DE_DATOS.Pago p ON t.ticket_total = p.pago_importe AND t.ticket_fecha_hora = p.pago_fecha  
WHERE t.id_ticket IS NOT NULL  
    and p.id_pago IS NOT NULL  
PRINT 'Migración de ticket_x_pago terminada';
```

## Promoción X Producto

Corresponderá a las distintas combinaciones que se pueden dar entre las promociones y los productos. Tendrá asociado tanto la promoción como como el producto mediante una foreign key para cada una.

```
CREATE TABLE REYES_DE_DATOS.Promocion_X_Producto (  
    id_promocion INT NOT NULL,  
    id_producto INT NOT NULL,  
);
```

Para hacer la migración definimos un insert que se encargará de obtener de la tabla maestra las distintas promociones para los distintos productos los cuales encontraremos en la columna: “PROMO\_CODIGO” y “PRODUCTO\_NOMBRE”.

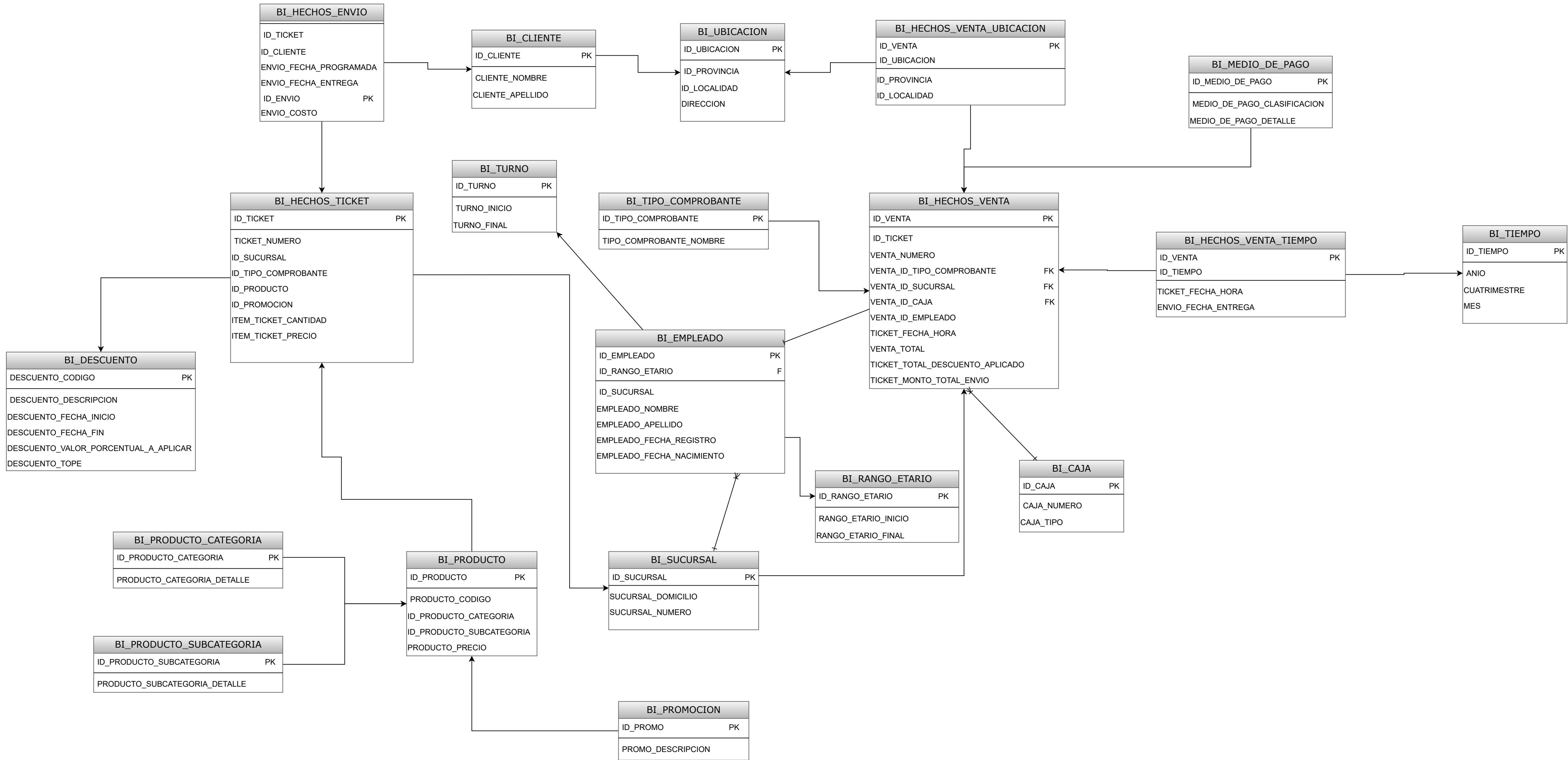
Teniendo esto en cuenta, insertamos las distintas promociones por los distintos producto de la tabla maestra en nuestra tabla. Finalmente, verificamos que “PROMO\_CODIGO” y “PRODUCTO\_NOMBRE” sean distintos de NULL.

```
INSERT INTO REYES_DE_DATOS.Promocion_X_Producto(  
    id_promocion,  
    id_producto  
)  
SELECT DISTINCT  
    m.PROMO_CODIGO,  
    p.id_promo  
FROM gd_esquema.Maestra m  
    JOIN REYES_DE_DATOS.Promocion p ON  
        m.PROMOCION_DESCRIPCION = p.promo_descripcion  
        AND m.PROMOCION_FECHA_INICIO = p.promo_fecha_inicio  
        AND m.PROMOCION_FECHA_FIN = p.promo_fecha_fin  
WHERE  
    m.TICKET_NUMERO IS NOT NULL  
    AND m.PROMO_CODIGO IS NOT NULL  
PRINT 'Migración de promocion_x_producto terminada';  
GO
```

## Modelo de Inteligencia de Negocios

### Diagrama Entidad Relación – BI





## Aclaraciones – BI

Para el modelo de negocios creamos 2 tablas de hechos (mmm me parece que vamos a tener que agregar más para el diagrama):

- Una para todo lo relacionado con las ventas en el tiempo, llamada “BI\_HECHOS\_VENTA\_TIEMPO”. Dicha tabla nos va a servir para agrupar las ventas con en el tiempo determinado en el que se efectuaron.
- Otra para todo lo relacionado con las ventas en las distintas ubicaciones, llamada “BI\_HECHOS\_VENTA\_UBICACION”. Dicha tabla nos va a servir para agrupar las ventas con las distintas ubicaciones y donde se efectuaron.

Otra consideraciones:

- No vimos necesario migrar todos los datos de nuestra migración inicial, solo nos preocupamos por aquellos campos necesarios para resolver nuestros requerimientos.