

PRESENTACIÓN carilla

Índices

ModeloTransaccional 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

Foto del Diagrama

Producto Categoría

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

```
CREATE TABLE FRBA_SUPERMERCADO.Producto_categoria (id_producto_categoria INT PRIMARY KEY);
```

Para hacer la migración definimos un store procedure 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.

```
CREATE PROCEDURE FRBA_SUPERMERCADO.migrar_categoria
AS
BEGIN
    INSERT INTO FRBA_SUPERMERCADO.Producto_categoria(id_producto_categoria)
        SELECT DISTINCT CAST(SUBSTRING(m.PRODUCTO_CATEGORIA, 13,20) AS INT)
        FROM gd_esquema.Maestra m
        WHERE SUBSTRING(m.PRODUCTO_CATEGORIA, 13,20) IS NOT NULL
    PRINT 'Migración de categoria terminada';
END
GO
```

Producto SubCategoría

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

```
CREATE TABLE FRBA_SUPERMERCADO.Producto_subcategoria (id_producto_subcategoria INT PRIMARY KEY);
```

Para hacer la migración definimos un store procedure 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.

```
CREATE PROCEDURE FRBA_SUPERMERCADO.migrar_subcategoria
AS
BEGIN
    INSERT INTO FRBA_SUPERMERCADO.Producto_subcategoria(id_producto_subcategoria)
        SELECT DISTINCT CAST(SUBSTRING(m.PRODUCTO_SUB_CATEGORIA, 13,20) AS INT)
        FROM gd_esquema.Maestra m
        WHERE SUBSTRING(m.PRODUCTO_SUB_CATEGORIA, 13,7) IS NOT NULL
    PRINT 'Migración de subcategoria terminada';
END
GO
```

Producto Marca

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

```
CREATE TABLE FRBA_SUPERMERCADO.Producto_marca (id_producto_marca INT PRIMARY KEY);
```

Para hacer la migración definimos un store procedure 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.

```
CREATE PROCEDURE FRBA_SUPERMERCADO.migrar_marca
AS
BEGIN
    INSERT INTO FRBA_SUPERMERCADO.Producto_marca(id_producto_marca)
        SELECT DISTINCT CAST(SUBSTRING(m.PRODUCTO_MARCA, 9,10) AS DECIMAL(10,0))
        FROM gd_esquema.Maestra m
        WHERE m.PRODUCTO_MARCA IS NOT NULL
    PRINT 'Migración de marca terminada';
END
GO
```

Cliente Contacto

Será el contacto perteneciente a nuestro cliente. Guardaremos:

```
CREATE TABLE FRBA_SUPERMERCADO.Cliente_Contacto (  
    id_cliente_contacto INT PRIMARY KEY IDENTITY(1,1),  
    cliente_contacto_numero VARCHAR(20) UNIQUE NOT NULL,  
);
```

Hablar sobre store procedure

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 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

```
CREATE PROCEDURE FRBA_SUPERMERCADO.migrar_tipo_medio_pago  
AS  
BEGIN  
    INSERT INTO FRBA_SUPERMERCADO.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';  
END  
GO
```


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 store procedure 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.

```
CREATE PROCEDURE FRBA_SUPERMERCADO.migrar_descuento  
AS  
BEGIN  
    INSERT INTO FRBA_SUPERMERCADO.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';  
END  
GO
```

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 store procedure 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.

```
CREATE PROCEDURE FRBA_SUPERMERCADO.migrar_regla
AS
BEGIN
    INSERT INTO FRBA_SUPERMERCADO.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 terminada';
END
GO
```

Supermercado

Serán nuestros supermercados. Guardaremos:

```
CREATE TABLE FRBA_SUPERMERCADO.Supermercado (  
    super_id 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,  
    super_fecha_inicio_actividad DATE NOT NULL,  
    super_condicion_fiscal VARCHAR(50) NOT NULL  
);
```

Para hacer la migración definimos un store procedure 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_IIBB" es NULL, por lo que nos encargamos de obviar esos casos.

```
CREATE PROCEDURE FRBA_SUPERMERCADO.migrar_supermercado
AS
BEGIN
    INSERT INTO FRBA_SUPERMERCADO.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,
        (CAST(SUBSTRING(m.SUPER_IIBB, 16,10) AS DECIMAL(9,0))),
        m.SUPER_FECHA_INI_ACTIVIDAD,
        m.SUPER_CONDICION_FISCAL
    FROM gd_esquema.Maestra m
    WHERE m.SUPER_IIBB IS NOT NULL
    PRINT 'Migración de supermercado terminada';
END
GO
```

Tipo Caja

Serán los distintos tipos de caja que habrá en nuestros supermercados. Guardaremos.

```
CREATE TABLE FRBA_SUPERMERCADO.Tipo_Caja(  
    id_tipo_caja INT PRIMARY KEY IDENTITY(1,1),  
    tipo_caja_descripcion VARCHAR(50) UNIQUE NOT NULL,  
);
```

Para hacer la migración definimos un store procedure que se encargará de obtener de la tabla maestra los distintos tipos de caja los cuales encontraremos en la columna "CAJA_TIPO".

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

```
CREATE PROCEDURE FRBA_SUPERMERCADO.migrar_tipo_caja  
AS  
BEGIN  
    INSERT INTO FRBA_SUPERMERCADO.Tipo_Caja(tipo_caja_descripcion)  
    SELECT DISTINCT(SUBSTRING(CAJA_TIPO,11,20))  
    FROM gd_esquema.Maestra  
    WHERE CAJA_TIPO IS NOT NULL  
    PRINT 'Migración de tipo de caja terminada';  
END  
GO
```

Tipo Comprobante

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

```
CREATE TABLE FRBA_SUPERMERCADO.Tipo_Comprobante(  
    id_tipo_comprobante INT PRIMARY KEY IDENTITY(1,1),  
    tipo_comprobante_nombre VARCHAR (1),  
);
```

Para hacer la migración definimos un store procedure 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.

```
CREATE PROCEDURE FRBA_SUPERMERCADO.migrar_tipo_comprobante  
AS  
BEGIN  
    INSERT INTO FRBA_SUPERMERCADO.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';  
END  
GO
```

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 store procedure 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.

```
CREATE PROCEDURE FRBA_SUPERMERCADO.migrar_Provincia  
AS  
BEGIN  
    INSERT INTO FRBA_SUPERMERCADO.Provincia(provincia_nombre)  
    SELECT 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';  
END  
GO
```


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 store procedure 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.

```
CREATE PROCEDURE FRBA_SUPERMERCADO.migrar_localidad  
AS  
BEGIN  
    INSERT INTO FRBA_SUPERMERCADO.Localidad(localidad_nombre)  
    SELECT 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';  
END  
GO
```

Promoción

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

```
CREATE TABLE FRBA_SUPERMERCADO.Promocion(  
    id_promo INT PRIMARY KEY IDENTITY(1,1),  
    promo_codigo INT NOT NULL,  
    promo_descripcion VARCHAR(50) UNIQUE NOT NULL,  
    promo_fecha_inicio DATE NOT NULL,  
    promo_fecha_fin DATE NOT NULL,  
    promo_valor_descuento DECIMAL(6,2) NOT NULL,  
);
```

Para hacer la migración definimos un store procedure 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.

```
CREATE PROCEDURE FRBA_SUPERMERCADO.migrar_promocion  
AS  
BEGIN  
    INSERT INTO FRBA_SUPERMERCADO.Promocion(  
        promo_codigo,  
        promo_descripcion,  
        promo_fecha_inicio,  
        promo_fecha_fin,  
        promo_valor_descuento  
    )  
    SELECT  
        PROMO_CODIGO,  
        PROMOCION_DESCRIPCION,  
        PROMOCION_FECHA_INICIO,  
        PROMOCION_FECHA_FIN,  
        PROMO_APLICADA_DESCUENTO  
    FROM gd_esquema.Maestra  
    WHERE PROMO_CODIGO IS NOT NULL  
    PRINT 'Migración de promocion terminada';  
END  
GO
```

Caja

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

```
CREATE TABLE FRBA_SUPERMERCADO.Caja(  
    id_caja INT PRIMARY KEY IDENTITY(1,1),  
    caja_numero INT,  
    id_tipo_caja INT FOREIGN KEY REFERENCES FRBA_SUPERMERCADO.Tipo_Caja (id_tipo_caja)  
);
```

Para hacer la migración definimos un store procedure que se encargará de obtener de la tabla maestra las cajas de las cuales encontraremos en las columnas “CAJA_NUMERO”. También deberemos hacer un JOIN a nuestra tabla de Tipo_Caja.

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.

```
CREATE PROCEDURE FRBA_SUPERMERCADO.migrar_caja  
AS  
BEGIN  
    INSERT INTO FRBA_SUPERMERCADO.Caja(  
        caja_numero,  
        id_tipo_caja  
    )  
    SELECT  
        m.CAJA_NUMERO,  
        c.id_tipo_caja  
    FROM gd_esquema.Maestra m  
        JOIN FRBA_SUPERMERCADO.Tipo_Caja c ON  
            c.tipo_caja_descripcion = (SELECT DISTINCT(SUBSTRING(m.CAJA_TIPO,11,20)))  
    WHERE CAJA_NUMERO IS NOT NULL  
    PRINT 'Migración de caja terminada';  
END  
GO
```

Domicilio

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

```
CREATE TABLE FRBA_SUPERMERCADO.Domicilio (  
    id_domicilio INT PRIMARY KEY IDENTITY(1,1),  
    id_localidad INT FOREIGN KEY REFERENCES FRBA_SUPERMERCADO.Localidad(id_localidad) NOT NULL,  
    id_provincia INT FOREIGN KEY REFERENCES FRBA_SUPERMERCADO.Provincia(id_provincia) 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”.

Cabe destacar que tendremos 2 funciones que nos ayuden a encontrar tanto la localidad como la provincia correspondiente.

```
CREATE PROCEDURE FRBA_SUPERMERCADO.migrar_domicilio  
AS  
BEGIN  
    INSERT INTO FRBA_SUPERMERCADO.Domicilio (  
        domicilio_direccion,  
        id_localidad,  
        id_provincia  
    )  
    SELECT  
        m.SUPER_DOMICILIO,  
        FRBA_SUPERMERCADO.get_localidad_id(m.SUPER_DOMICILIO) AS id_localidad,  
        FRBA_SUPERMERCADO.get_provincia_id(m.SUPER_PROVINCIA) AS id_provincia  
    FROM gd_esquema.Maestra m  
    WHERE CLIENTE_DOMICILIO IS NOT NULL OR SUPER_DOMICILIO IS NOT NULL;  
    INSERT INTO FRBA_SUPERMERCADO.Domicilio (  
        domicilio_direccion,  
        id_localidad,  
        id_provincia  
    )
```

```
    SELECT  
        m.CLIENTE_DOMICILIO,  
        FRBA_SUPERMERCADO.get_localidad_id(m.CLIENTE_LOCALIDAD),  
        FRBA_SUPERMERCADO.get_provincia_id(m.CLIENTE_PROVINCIA)  
    FROM gd_esquema.Maestra m  
    WHERE CLIENTE_DOMICILIO IS NOT NULL;  
    PRINT 'Migracion de Domicilio terminada';  
END;  
GO
```

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 FRBA_SUPERMERCADO.Sucursal (  
    id_sucursal INT PRIMARY KEY IDENTITY(1,1),  
    id_domicilio INT FOREIGN KEY REFERENCES FRBA_SUPERMERCADO.Domicilio(id_domicilio) NOT NULL,  
    id_supermercado INT FOREIGN KEY REFERENCES FRBA_SUPERMERCADO.Supermercado(super_id) NOT NULL,  
    sucursal_numero INT NOT NULL,  
);
```

Para confeccionar la migración nos va a interesar de la tabla maestra obtener las sucursales las cuales encontraremos en las columnas "SUPER_DOMICILIO" "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.

Cabe resaltar que tendremos una función que nos ayude a hallar el id del domicilio de la sucursal

```
CREATE PROCEDURE FRBA_SUPERMERCADO.migrar_Sucursal  
AS  
BEGIN  
    INSERT INTO FRBA_SUPERMERCADO.Sucursal (  
        id_domicilio,  
        id_supermercado,  
        sucursal_numero  
    )  
    SELECT  
        FRBA_SUPERMERCADO.get_domicilio_id(m.SUPER_DOMICILIO),  
        s.id_supermercado,  
        (SELECT DISTINCT(SUBSTRING(m.SUCURSAL_NOMBRE,11,20)))  
    FROM gd_esquema.Maestra m  
    JOIN FRBA_SUPERMERCADO.Supermercado s ON m.SUPER_CUIT = s.super_cuit  
    WHERE 1 IS NOT NULL  
    PRINT 'Migración de sucursal terminada';  
END  
GO
```

Cliente

Serán nuestros clientes que compren productos en los supermercados.

```
CREATE TABLE FRBA_SUPERMERCADO.Cliente(  
    id_cliente INT PRIMARY KEY IDENTITY(1,1),  
    cliente_dni INT NOT NULL,  
    cliente_id_domicilio INT FOREIGN KEY REFERENCES FRBA_SUPERMERCADO.Domicilio(id_domicilio) NOT NULL,  
    cliente_nombre VARCHAR(100) NOT NULL,  
    cliente_apellido VARCHAR(100) NOT NULL,  
    cliente_fecha_registro DATE NOT NULL,  
    cliente_mail VARCHAR(100) UNIQUE NOT NULL,  
    cliente_fecha_nacimiento DATE NOT NULL,  
);
```

Para hacer la migración definimos un store procedure 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".

También contamos con una función que nos facilitará la obtención del domicilio del cliente.

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.

```
CREATE PROCEDURE FRBA_SUPERMERCADO.migrar_cliente
AS
BEGIN
    INSERT INTO FRBA_SUPERMERCADO.Cliente(
        cliente_dni,
        cliente_id_domicilio,
        cliente_nombre,
        cliente_apellido,
        cliente_fecha_registro,
        cliente_mail,
        cliente_fecha_nacimiento
    )
    SELECT DISTINCT
        m.CLIENTE_DNI,
        FRBA_SUPERMERCADO.get_domicilio_id(m.CLIENTE_DOMICILIO),
        m.CLIENTE_NOMBRE,
        m.CLIENTE_APELLIDO,
        m.CLIENTE_FECHA_REGISTRO,
        m.CLIENTE_MAIL,
        m.CLIENTE_FECHA_NACIMIENTO
    FROM gd_esquema.Maestra m
    WHERE m.CLIENTE_DNI IS NOT NULL
    ORDER BY m.CLIENTE_DNI
    PRINT 'Migración de cliente terminada';
END
GO
```

Empleado

serán los empleados que trabajen en los supermercados. Guardaremos:

```
CREATE TABLE FRBA_SUPERMERCADO.Empleado(  
    id_empleado INT PRIMARY KEY IDENTITY(1,1),  
    id_sucursal INT FOREIGN KEY REFERENCES FRBA_SUPERMERCADO.Sucursal (id_sucursal) NOT NULL,  
    empleado_nombre VARCHAR(15) NOT NULL,  
    empleado_apellido VARCHAR(30) NOT NULL,  
    empleado_dni INT NOT NULL,  
    empleado_fecha_registro DATE NOT NULL,  
    empleado_fecha_nacimiento DATE NOT NULL,  
    empleado_email VARCHAR (30) UNIQUE NOT NULL,  
);
```

Para hacer la migración definimos un store procedure 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"

También contamos con una función que nos facilitará la obtención de la sucursal en la cual trabaja el empleado.

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 "EMPLEADO_DNI" es NULL, por lo que nos encargamos de obviar esos casos.


```
CREATE PROCEDURE FRBA_SUPERMERCADO.migrar_Empleado
AS
BEGIN
    INSERT INTO FRBA_SUPERMERCADO.Empleado(
        id_sucursal,
        empleado_nombre,
        empleado_apellido,
        empleado_dni,
        empleado_fecha_registro,
        empleado_fecha_nacimiento,
        empleado_email,
        empleado_telefono
    )
    SELECT
        FRBA_SUPERMERCADO.get_sucursal_id(m.SUCURSAL_DIRECCION),
        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
    WHERE m.EMPLEADO_DNI IS NOT NULL
    PRINT 'Migración de empleado terminada';
END
GO
```

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 FRBA_SUPERMERCADO.Producto (  
    id_producto INT PRIMARY KEY,  
    id_producto_categoria INT FOREIGN KEY REFERENCES FRBA_SUPERMERCADO.Producto_categoria(id_producto_categoria) NOT NULL,  
    id_producto_subcategoria INT FOREIGN KEY REFERENCES FRBA_SUPERMERCADO.Producto_subcategoria(id_producto_subcategoria) NOT NULL,  
    id_marca INT FOREIGN KEY REFERENCES FRBA_SUPERMERCADO.Producto_marca(id_producto_marca) NOT NULL,  
    producto_descripcion VARCHAR(100) NOT NULL,  
    producto_precio DECIMAL(10,2) NOT NULL,  
);
```

Para hacer la migración definimos un store procedure 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”

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.

```
CREATE PROCEDURE FRBA_SUPERMERCADO.migrar_Producto
AS
BEGIN
    INSERT INTO FRBA_SUPERMERCADO.Producto(
        id_producto, -- PRODUCTO_NOMBRE
        id_producto_categoria,
        id_producto_subcategoria,
        id_marca,
        producto_descripcion,
        producto_precio
    )
    SELECT
        (CAST(SUBSTRING(m.PRODUCTO_NOMBRE,8,20) AS DECIMAL(11,0))),
        (CAST(SUBSTRING(m.PRODUCTO_CATEGORIA, 13,20) AS INT)),
        (CAST(SUBSTRING(m.PRODUCTO_SUB_CATEGORIA, 16,20) AS INT)),
        (CAST(SUBSTRING(m.PRODUCTO_MARCA,9,10) AS DECIMAL(10,0))),
        (CAST(SUBSTRING(m.PRODUCTO_DESCRIPCION,29,70) AS DECIMAL(11,0))),
        m.PRODUCTO_PRECIO
    FROM gd_esquema.Maestra m
    WHERE PRODUCTO_NOMBRE IS NOT NULL
    PRINT 'Migración de producto terminada';
END
GO
```

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 FRBA_SUPERMERCADO.Ticket(  
    id_ticket INT PRIMARY KEY NOT NULL,  
    id_tipo_comprobante INT FOREIGN KEY REFERENCES FRBA_SUPERMERCADO.Tipo_Comprobante (id_tipo_comprobante) NOT NULL,  
    id_sucursal INT FOREIGN KEY REFERENCES FRBA_SUPERMERCADO.Sucursal (id_sucursal) NOT NULL,  
    id_caja INT FOREIGN KEY REFERENCES FRBA_SUPERMERCADO.Caja (id_caja) NOT NULL,  
    id_empleado INT FOREIGN KEY REFERENCES FRBA_SUPERMERCADO.Empleado (id_empleado) NOT NULL,  
    ticket_fecha_hora DATE NOT NULL,  
    ticket_subtotal DECIMAL(10, 2) NOT NULL,  
    ticket_total DECIMAL(10, 2) NOT NULL,  
    ticket_monto_total_promociones_aplicadas INT NOT NULL,  
    ticket_monto_total_descuentos_aplicados INT NOT NULL,  
);
```

Para hacer la migración definimos un store procedure 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 dos JOIN para verificar que el ticket pertenezca a la Caja y al Tipo_Comprobante, 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.

```

CREATE PROCEDURE FRBA_SUPERMERCADO.migrar_ticket
AS
BEGIN
    INSERT INTO FRBA_SUPERMERCADO.Ticket(
        id_ticket,
        id_tipo_comprobante,
        id_sucursal,
        id_caja,
        id_empleado,
        ticket_fecha_hora,
        ticket_subtotal,
        ticket_total,
        ticket_monto_total_promociones_aplicadas,
        ticket_monto_total_descuentos_aplicados
    )
    SELECT DISTINCT
        m.TICKET_NUMERO,
        tc.id_tipo_comprobante,
        FRBA_SUPERMERCADO.get_sucursal_id(m.SUCURSAL_DIRECCION),
        caja.id_caja,
        FRBA_SUPERMERCADO.get_empleado_id(m.EMPLEADO_MAIL),
        m.TICKET_FECHA_HORA,
        m.TICKET_SUBTOTAL_PRODUCTOS,
        (m.TICKET_TOTAL_DESCUENTO_APLICADO + m.TICKET_TOTAL_DESCUENTO_APLICADO_MP),
        m.TICKET_TOTAL_ENVIO
    FROM gd_esquema.Maestra m
    JOIN FRBA_SUPERMERCADO.Tipo_Caja caja ON m.CAJA_NUMERO = caja.caja_numero
    JOIN FRBA_SUPERMERCADO.Tipo_Comprobante tc ON m.TICKET_TIPO_COMPROBANTE = tc.tipo_comprobante_nombre
    WHERE m.TICKET_NUMERO IS NOT NULL
    PRINT 'Migración de Ticket terminada';
END
GO

```

Envío

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

```
CREATE TABLE FRBA_SUPERMERCADO.Envio(  
    id_envio INT PRIMARY KEY IDENTITY(1,1),  
    id_ticket INT FOREIGN KEY REFERENCES FRBA_SUPERMERCADO.Ticket (id_ticket) NOT NULL,  
    id_cliente INT FOREIGN KEY REFERENCES FRBA_SUPERMERCADO.Cliente (id_cliente) NOT NULL,  
    envio_fecha_programada DATE NOT NULL,  
    envio_horario_inicio TIME NOT NULL,  
    envio_horario_fin TIME NOT NULL,  
    envio_fecha_entrega DATE NOT NULL,  
    envio_estado VARCHAR(50) NOT NULL,  
    envio_costo DECIMAL(10, 2) NOT NULL,  
    envio_estado_envio VARCHAR(50) NOT NULL,  
);
```

Para hacer la migración definimos un store procedure 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.

```
CREATE PROCEDURE FRBA_SUPERMERCADO.migrar_envio
AS
BEGIN
    INSERT INTO FRBA_SUPERMERCADO.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,
        FRBA_SUPERMERCADO.get_cliente_id(m.CLIENTE_DNI),
        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
    WHERE
        m.ENVIO_HORA_INICIO IS NOT NULL
        AND m.ENVIO_HORA_FIN IS NOT NULL
        AND m.ENVIO_FECHA_ENTREGA IS NOT NULL

    PRINT 'Migración de envio terminada';
END
GO
```

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 FRBA_SUPERMERCADO.Item_Ticket(  
    id_item_ticket INT PRIMARY KEY IDENTITY(1,1),  
    id_ticket INT FOREIGN KEY (id_ticket) REFERENCES FRBA_SUPERMERCADO.Ticket (id_ticket) NOT NULL,  
    id_producto INT FOREIGN KEY (id_producto) REFERENCES FRBA_SUPERMERCADO.Producto (id_producto) NOT NULL,  
    id_tipo_comprobante INT FOREIGN KEY (id_tipo_comprobante) REFERENCES FRBA_SUPERMERCADO.Tipo_Comprobante (id_tipo_comprobante) NOT NULL,  
    id_sucursal INT FOREIGN KEY (id_sucursal) REFERENCES FRBA_SUPERMERCADO.Sucursal (id_sucursal) NOT NULL,  
    id_promocion INT FOREIGN KEY (id_promocion) REFERENCES FRBA_SUPERMERCADO.Promocion (id_promocion) NOT NULL,  
    item_ticket_cantidad INT NOT NULL,  
    item_ticket_precio INT NOT NULL,  
);
```

Para hacer la migración definimos un store procedure 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.


```

CREATE PROCEDURE FRBA_SUPERMERCADO.migrar_item_ticket
AS
BEGIN
    INSERT INTO FRBA_SUPERMERCADO.Item_Ticket(
        id_ticket, --ticketunmero
        id_producto,
        id_tipo_comprobante,
        id_sucursal,
        id_promocion,
        item_ticket_cantidad,
        item_ticket_precio
    )
    SELECT
        m.TICKET_NUMERO,
        m.PRODUCTO_NOMBRE, --tiene letras
        tc.id_tipo_comprobante,
        FRBA_SUPERMERCADO.get_sucursal_id(m.SUCURSAL_DIRECCION),
        m.PROMO_CODIGO,
        m. TICKET_DET_CANTIDAD,
        m.TICKET_DET_PRECIO
    FROM gd_esquema.Maestra m
    JOIN FRBA_SUPERMERCADO.Tipo_Comprobante tc ON m.TICKET_TIPO_COMPROBANTE = tc.tipo_comprobante_nombre
    WHERE m.TICKET_NUMERO IS NOT NULL
    PRINT 'Migración de item ticket terminada';
END
GO

```

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 FRBA_SUPERMERCADO.Pago(  
    id_pago INT PRIMARY KEY IDENTITY(1,1),  
    id_tipo_medio_de_pago INT FOREIGN KEY REFERENCES FRBA_SUPERMERCADO.Tipo_medio_de_pago(id_tipo_medio_pago) NOT NULL,  
    id_cliente INT FOREIGN KEY REFERENCES FRBA_SUPERMERCADO.Cliente(id_cliente) NOT NULL,  
    id_descuento INT FOREIGN KEY (id_descuento) REFERENCES FRBA_SUPERMERCADO.Descuento(descuento_codigo),  
    pago_fecha DATE NOT NULL,  
    pago_importe DECIMAL(10,2) NOT NULL,  
    pago_numero_tarjeta VARCHAR(20),  
    medio_de_pago_cuotas INT,  
    medio_de_pago_fecha_vencimiento DATE,  
    medio_de_pago_descuento_aplicado DECIMAL(10, 2),  
);
```

Para hacer la migración definimos un store procedure 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.

```

CREATE PROCEDURE FRBA_SUPERMERCADO.migrar_pago
AS
BEGIN
    INSERT INTO FRBA_SUPERMERCADO.pago(
        id_tipo_medio_de_pago,
        id_cliente,
        id_descuento,
        pago_fecha,
        pago_importe,
        pago_numero_tarjeta,
        medio_de_pago_cuotas,
        medio_de_pago_fecha_vencimiento,
        medio_de_pago_descuento_aplicado
    )
    SELECT
        mp.medio_de_pago_detalle,
        FRBA_SUPERMERCADO.get_cliente_id(m.CLIENTE_DNI),
        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 FRBA_SUPERMERCADO.Tipo_medio_de_pago mp ON m.PAGO_MEDIO_PAGO = mp.medio_de_pago_detalle
    WHERE m.TICKET_NUMERO IS NOT NULL
    PRINT 'Migración de Pago terminada';
END
GO

```

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 FRBA_SUPERMERCADO.Regla_x_Promocion (  
    id_promocion INT,  
    id_regla INT,  
    CONSTRAINT PK_Regla_X_Promocion PRIMARY KEY (id_promocion, id_regla),  
    CONSTRAINT FK_Regla_X_Promocion_Promocion FOREIGN KEY (id_promocion) REFERENCES FRBA_SUPERMERCADO.Promocion(id_promocion),  
    CONSTRAINT FK_Regla_X_Promocion_Regla FOREIGN KEY (id_regla) REFERENCES FRBA_SUPERMERCADO.Regla(id_regla)  
);
```

Para hacer la migración definimos un store procedure 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.

```
CREATE PROCEDURE FRBA_SUPERMERCADO.migrar_regla_x_promocion  
AS  
BEGIN  
    INSERT INTO FRBA_SUPERMERCADO.Regla_x_Promocion(  
        id_promocion,  
        id_regla  
    )  
    SELECT  
        m.PROMO_CODIGO,  
        FRBA_SUPERMERCADO.get_regla_id(m.REGLA_DESCRIPCION)  
    FROM gd_esquema.Maestra m  
    WHERE  
        m.PROMO_CODIGO IS NOT NULL  
        AND 2 IS NOT NULL  
    PRINT 'Migración de regla x promocion terminada';  
END  
GO
```

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 FRBA_SUPERMERCADO.Ticket_X_Pago(  
    id_ticket INT,  
    id_pago INT,  
    CONSTRAINT PK_Ticket_X_Pago PRIMARY KEY (id_ticket, id_pago),  
    CONSTRAINT FK_Ticket_X_Pago_id_ticket FOREIGN KEY (id_ticket) REFERENCES FRBA_SUPERMERCADO.Ticket (id_ticket),  
    CONSTRAINT FK_Ticket_X_Pago_id_pago FOREIGN KEY (id_pago) REFERENCES FRBA_SUPERMERCADO.Sucursal (id_sucursal)  
);
```

Para hacer la migración definimos un store procedure 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.

```
CREATE PROCEDURE FRBA_SUPERMERCADO.migrar_ticket_x_pago  
AS  
BEGIN  
    INSERT INTO FRBA_SUPERMERCADO.Ticket_X_Pago(  
        id_ticket,  
        id_pago  
    )  
    SELECT  
        m.TICKET_NUMERO,  
        p.id_pago  
    FROM gd_esquema.Maestra m  
        JOIN FRBA_SUPERMERCADO.Pago p ON  
            m.FRBA_SUPERMERCADO.get_cliente_id(m.CLIENTE_DNI) = p.id_cliente  
            AND m.PAGO_FECHA = p.pago_fecha  
    WHERE m.TICKET_NUMERO IS NOT NULL  
    PRINT 'Migración de ticket_x_pago terminada';  
END  
GO
```

Promoción X Item Ticket

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

```
CREATE TABLE FRBA_SUPERMERCADO.Promocion_X_Item_Ticket (--
    id_promocion INT,
    id_item_ticket INT,
    CONSTRAINT PK_Promocion_X_ItemTicket PRIMARY KEY (id_promocion, id_item_ticket),
    CONSTRAINT FK_Promocion_X_ItemTicket_Promocion FOREIGN KEY (id_promocion) REFERENCES FRBA_SUPERMERCADO.Promocion(id_promocion),
    CONSTRAINT FK_Promocion_X_ItemTicket_Item_Ticket FOREIGN KEY (id_item_ticket) REFERENCES FRBA_SUPERMERCADO.Item_Ticket(id_item_ticket)
);
```

Para hacer la migración definimos un store procedure que se encargará de obtener de la tabla maestra las distintas promociones por cada item del ticket las cuales encontraremos en la columna: "PROMO_CODIGO".

A su vez, haremos un JOIN para verificar que el item del ticket de cada promoción pertenezca al Item_ticket, y estableciendo que el "TICKET_NUMERO" coincida con el id_ticket de nuestro Item_ticket, a su vez, que coincide con el id_producto y el id_promoción.

Teniendo esto en cuenta, insertamos los distintos envios tickets por los medios de pago de la tabla maestra en nuestra tabla. Finalmente, verificamos que "TICKET_NUMERO" y "PROMO_CODIGO" sean distintos de NULL.

```
CREATE PROCEDURE FRBA_SUPERMERCADO.migrar_promocion_x_item_ticket
AS
BEGIN
    INSERT INTO FRBA_SUPERMERCADO.Promocion_X_Item_Ticket(
        id_promocion,
        id_item_ticket
    )
    SELECT
        m.PROMO_CODIGO,
        id.id_item_ticket
    FROM gd_esquema.Maestra m
        JOIN FRBA_SUPERMERCADO.Item_Ticket it ON
            m.TICKET_NUMERO = it.id_ticket
            AND m.PRODUCTO_NOMBRE = it.id_producto
            AND m.PROMO_CODIGO = it.id_promocion
    WHERE
        m.TICKET_NUMERO IS NOT NULL
        AND m.PROMO_CODIGO IS NOT NULL
    PRINT 'Migración de promo_x_item_ticket terminada';
END
GO
```

Promoción X Producto

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

```
CREATE TABLE FRBA_SUPERMERCADO.Promocion_X_Producto (  
    id_promocion INT,  
    id_producto INT,  
    CONSTRAINT PK_Promocion_X_ItemTicket PRIMARY KEY (id_promocion, id_producto),  
    CONSTRAINT FK_Promocion_X_Producto_id_promocion FOREIGN KEY (id_promocion) REFERENCES FRBA_SUPERMERCADO.Promocion(id_promocion),  
    CONSTRAINT FK_Promocion_X_Producto_id_producto FOREIGN KEY (id_producto) REFERENCES FRBA_SUPERMERCADO.Producto(id_producto)  
);  
--  
GO
```

Para hacer la migración definimos un store procedure 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.

```
CREATE PROCEDURE FRBA_SUPERMERCADO.migrar_promocion_x_producto  
AS  
BEGIN  
    INSERT INTO FRBA_SUPERMERCADO.Promocion_X_Producto(  
        id_promocion,  
        id_producto  
    )  
    SELECT  
        m.PROMO_CODIGO,  
        m.PRODUCTO_NOMBRE  
    FROM gd_esquema.Maestra m  
    WHERE  
        m.TICKET_NUMERO IS NOT NULL  
        AND m.PROMO_CODIGO IS NOT NULL  
    PRINT 'Migración de promocion_x_producto terminada';  
END  
GO
```