

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”

## 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,  
);
```

Hablar sobre Store procedure

## 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,
);
```

hablar sobre Store procedure

## 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
);
```

hablar sobre store procedure

## 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,
);
```

hablar del store procedure

## 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),  
);
```

## 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,  
);
```

Hablar sobre Store Procedure

## 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,  
);
```

hablar store procedures

## 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,  
);
```

hablar sobre store procedures



## 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)  
);
```

hablar del store procedure

## 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,  
);
```

Hablar del procedure

## 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,  
);
```

procedures.....

## 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,  
);
```

store procedure

## 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,  
);
```

Hablar del store procedure

## 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,  
);
```

store procedure....

## 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,  
);
```

store procedures

## Envio

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,  
);
```

store procedure

## 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,  
);
```

store procedure.....

## 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),  
);
```

store procedure

## 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)  
);
```

procedure

## 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)  
);
```

store procedure...

## 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)  
);
```

procedure

## 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 FRBA_SUPERMERCADO.Promocion_X_Producto (  
    id_promocion INT,  
    id_producto INT,  
    CONSTRAINT PK_Promocion_X_ItemTicket PRIMARY KEY (id_promocion, id_producto),  
    CONSTRAINT FK_Promoxion_X_Producto_id_promocion FOREIGN KEY (id_promocion) REFERENCES FRBA_SUPERMERCADO.Promocion(id_promocion),  
    CONSTRAINT FK_Promoxion_X_Producto_id_producto FOREIGN KEY (id_producto) REFERENCES FRBA_SUPERMERCADO.Producto(id_producto)  
);  
--  
GO
```

store procedure