

## DDL

```
/* Create Tables */
```

```
CREATE TABLE INFO (  
    llave varchar(50) PRIMARY KEY NOT NULL,  
    valor text  
);
```

```
/* 0x01 = Domingo, 0x02 = Lunes, 0x04 = Martes, 0x08 = JMiercoles, 0x010 = Jueves, 0x020 = Viernes, 0x40 = Sabado */
```

```
/* 1 = Domingo, 2 = Lunes, 4 = Martes, 8 = Miercoles, 16 = Jueves, 32 = Viernes, 64 = Sabado */
```

```
/* Todos los dias = 127, Lunes a Viernes = 62, Sabados y Domingos = 65 */
```

```
CREATE TABLE Turno (  
    turno_id INT identity(1,1) PRIMARY KEY NOT NULL,  
    nombre VARCHAR(30) NOT NULL,  
    dias INT NOT NULL,  
    hora_comienzo TIME NOT NULL,  
    hora_termina TIME NOT NULL  
);
```

```
CREATE TABLE Empleado(  
    empleado_id INT identity(1,1) PRIMARY KEY NOT NULL,  
    no_identificacion varchar(15),  
    tipo_identificacion varchar(30) CHECK(tipo_identificacion IN('cedula', 'pasaporte')),  
    correo varchar(50) NOT NULL,  
    password varchar(128) NOT NULL,  
    nombre VARCHAR(30) NOT NULL,  
    apellido VARCHAR(30),  
    telefono VARCHAR(12),  
    calle VARCHAR(50),  
    no_casa VARCHAR(50),  
    sector VARCHAR(50),  
    ciudad VARCHAR(50),  
    provincia VARCHAR(50),  
    turno_id INT,  
    cargo varchar(20) CHECK(cargo IN('vendedor', 'supervisor', 'delivery')),  
    constraint empleado_turno_fk FOREIGN KEY (turno_id) REFERENCES Turno(turno_id)  
);
```

```

CREATE TABLE Caja (
    caja_id INT identity(1,1) PRIMARY KEY NOT NULL,
    empleado_id INT NOT NULL,
    cash_entrada money NOT NULL,
    fecha_abre DATETIME NOT NULL DEFAULT GETDATE(),
    fecha_cierra DATETIME,
    autorizador_cierre INT,
    constraint caja_empleado_fk FOREIGN KEY (empleado_id) REFERENCES Empleado(empleado_id),
    constraint caja_autorizador_fk FOREIGN KEY (autorizador_cierre) REFERENCES Empleado(empleado_id)
);

```

```

CREATE TABLE Cliente(
    cliente_id INT identity(1,1) PRIMARY KEY NOT NULL,
    nombre VARCHAR(30) NOT NULL,
    apellido VARCHAR(30),
    telefono VARCHAR(12) NOT NULL,
    correo VARCHAR(100),
    RNC varchar(15),
    calle VARCHAR(100),
    no_casa VARCHAR(100),
    sector VARCHAR(50),
    ciudad VARCHAR(50),
    provincia VARCHAR(50),
);

```

```

CREATE TABLE Suplidor (
    suplidor_id INT NOT NULL PRIMARY KEY identity(1,1),
    nombre varchar(30) NOT NULL,
    descripcion varchar(30),
    calle varchar(50),
    calle_no varchar(30),
    ciudad varchar(50),
    provincia varchar(50),
    telefono varchar(16)
)

```

```

CREATE TABLE Producto(
    producto_id INT identity(1,1) PRIMARY KEY NOT NULL,
    nombre VARCHAR(50) NOT NULL,
    descripcion VARCHAR(100),
    etiqueta_negra BIT DEFAULT 0,
    precio_venta money,
    precio_compra money,
    punto_reorden DECIMAL(10, 2),
    suplidor INT,
    CONSTRAINT producto_suplidor_fk FOREIGN KEY (suplidor) REFERENCES Suplidor(suplidor_id)
);

```

```

);

CREATE TABLE SaborHelado(
    producto_id INT PRIMARY KEY NOT NULL,
    temporal BIT DEFAULT 0,
    principio_temporada DATE,
    fin_temporada DATE,
    CONSTRAINT sabor_helado_producto_fk FOREIGN KEY (producto_id) REFERENCES Producto(producto_id)
);

CREATE TABLE RegistroInventario(
    inventario_id INT PRIMARY KEY identity(1,1) NOT NULL,
    empleado_id INT NOT NULL,
    notas text,
    fecha DATE NOT NULL
    CONSTRAINT inventario_empleado_fk FOREIGN KEY (empleado_id) REFERENCES Empleado(empleado_id)
);

CREATE TABLE RegistroInventario_Productos(
    producto_id INT NOT NULL,
    inventario_id INT NOT NULL,
    cantidad DECIMAL(10, 2) NOT NULL,
    CONSTRAINT Producto_RegistroInventario_pk PRIMARY KEY (producto_id, inventario_id),
    CONSTRAINT producto_fk FOREIGN KEY (producto_id) REFERENCES Producto(producto_id),
    CONSTRAINT inventario_fk FOREIGN KEY (inventario_id) REFERENCES RegistroInventario(inventario_id)
);

CREATE TABLE Orden(
    orden_id INT identity(1,1) PRIMARY KEY NOT NULL,
    empleado_id INT NOT NULL,
    suplidor_id INT NOT NULL,
    NOTAS TEXT,
    fecha_ordenada DATETIME NOT NULL DEFAULT GETDATE(),
    aceptada BIT DEFAULT 0,
    fecha_aceptada DATETIME,
    despachada BIT DEFAULT 0,
    fecha_despachada DATETIME,
    recibida BIT DEFAULT 0,
    fecha_recibida DATETIME,
    constraint orden_empleado_fk FOREIGN KEY (empleado_id) REFERENCES Empleado(empleado_id),
    constraint orden_suplidor_fk FOREIGN KEY (suplidor_id) REFERENCES Suplidor(suplidor_id)
);

```

```

CREATE TABLE Orden_Productos (
    orden_id INT NOT NULL,
    producto_id INT NOT NULL,
    cantidad INT NOT NULL,
    CONSTRAINT orden_producto_pk PRIMARY KEY (orden_id, producto_id),
    CONSTRAINT orden_producto_orden_fk FOREIGN KEY (orden_id) REFERENCES Orden(orden_id),
    CONSTRAINT orden_producto_producto_fk FOREIGN KEY (producto_id) REFERENCES Producto(producto_id)
);

```

```

CREATE TABLE Venta(
    venta_id BIGINT identity(1,1) PRIMARY KEY NOT NULL,
    cliente_id INT,
    caja_id INT NOT NULL,
    fecha DATETIME NOT NULL,
    fecha_entrega DATETIME,
    forma_pago VARCHAR(50) CHECK (forma_pago IN('efectivo', 'tarjeta')) NOT NULL,
    total_money NOT NULL DEFAULT 0,
    entregado_por INT,
    no_tarjeta varchar(4), -- ultimos 4 digitos
    CONSTRAINT venta_cliente_fk FOREIGN KEY (cliente_id) REFERENCES Cliente(cliente_id),
    CONSTRAINT venta_caja_fk FOREIGN KEY (caja_id) REFERENCES Caja(caja_id),
    CONSTRAINT venta_empleado_fk FOREIGN KEY (entregado_por) REFERENCES Empleado(empleado_id)
);

```

```

CREATE TABLE Venta_Productos(
    venta_id BIGINT NOT NULL,
    producto_id INT NOT NULL,
    cantidad INT NOT NULL,
    CONSTRAINT venta_productos_pk PRIMARY KEY (venta_id, producto_id),
    CONSTRAINT venta_id_fk FOREIGN KEY (venta_id) REFERENCES Venta(venta_id),
    CONSTRAINT producto_id_fk FOREIGN KEY (producto_id) REFERENCES Producto(producto_id)
);

```

```

CREATE TABLE NCF(
    no_NCF VARCHAR (25) PRIMARY KEY NOT NULL,
    venta_id BIGINT,
    CONSTRAINT NCF_venta FOREIGN KEY (venta_id) REFERENCES Venta (venta_id) ON DELETE CASCADE
);

```

```

CREATE TABLE Oferta(
    oferta_id INT PRIMARY KEY identity(1,1) NOT NULL,
    nombre VARCHAR(50) NOT NULL,
    descripcion VARCHAR(200),
    fecha_empieza DATE NOT NULL,
    fecha_termina DATE NOT NULL,
    dias_disponible INT NOT NULL, -- BIT MASK
    hora_disponible_empieza TIME NOT NULL,
    hora_disponible_termina TIME NOT NULL,
    cantidad_maxima INT,
    cantidad_minima INT,
    producto_id INT NOT NULL,
    tipo varchar(20) NOT NULL CHECK (tipo IN ('2x1', 'porciento')),
    codigo_cupon varchar(10),
    constraint oferta_producto_pk FOREIGN KEY (producto_id) REFERENCES Producto(producto_id)
);

```

```

CREATE TABLE OfertaPorciento(
    oferta_id INT PRIMARY KEY NOT NULL,
    porciento DECIMAL(10,2) NOT NULL,
    constraint porciento_oferta_fk FOREIGN KEY (oferta_id) REFERENCES Oferta(oferta_id)
);

```

```

CREATE TABLE Venta_Ofertas (
    venta_id BIGINT NOT NULL,
    oferta_id INT NOT NULL,
    rebaja money NOT NULL,
    constraint venta_ofertas_pk PRIMARY KEY (venta_id, oferta_id),
    constraint venta_ofertas_oferta_fk FOREIGN KEY (oferta_id) REFERENCES Oferta(oferta_id),
    constraint venta_ofertas_venta_fk FOREIGN KEY (venta_id) REFERENCES Venta(venta_id)
)

```

```
drop table INFO;  
drop table Venta_Productos  
drop table Venta_Ofertas;  
drop table RegistroInventario_Productos;  
drop table Orden_Productos;  
drop table SaborHelado;  
drop table OfertaPorciento;  
drop table NCF;  
drop table Oferta;  
drop table RegistroInventario;  
drop table Venta;  
drop table Orden;  
drop table Producto;  
drop table Suplidor;  
drop table Caja;  
drop table Cliente;  
drop table Empleado;  
drop table Turno;
```

# DML

```
CREATE PROCEDURE sp_CajaCerrada
(
    @empleado_id INT,
    @fecha DATE,
    @cash_salida AS money
)
AS
BEGIN
(
    SELECT Caja.caja_id, Caja.fecha_abre, Caja.cash_entrada , Empleado.nombre, Empleado.apellido,
    (
        SELECT SUM (venta.total)
        FROM Venta
        WHERE Venta.caja_id = Caja.caja_id AND Venta.forma_pago = 'efectivo'
    ) as TotalVentas, (@cash_salida-(Caja.cash_entrada+
    (SELECT SUM (venta.total) FROM Venta WHERE Venta.caja_id = Caja.caja_id AND Venta.forma_pago = 'efectivo')
    )) AS Balance
    FROM Caja INNER JOIN Empleado
    ON Caja.empleado_id=Empleado.empleado_id
    WHERE Empleado.empleado_id = @empleado_id AND CONVERT(DATE, Caja.fecha_abre) = @fecha
)
END

CREATE PROCEDURE sp_ComprobantesRegistrados
(
    @fc_desde DATETIME,
    @fc_hasta DATETIME
)
AS
BEGIN
(
    SELECT NCF.no_NCF, (Cliente.nombre + ' ' + Cliente.apellido) as Cliente, Cliente.RNC, Venta.total, CONVERT(DECIMAL(10,2),
    (Venta.total * 0.18)) as MontoITBIS
    FROM NCF INNER JOIN Venta
    ON NCF.venta_id=Venta.venta_id INNER JOIN Cliente
    ON Cliente.cliente_id=Venta.cliente_id
    WHERE Venta.fecha BETWEEN @fc_desde AND @fc_hasta
)
END
```

```

CREATE TRIGGER trg_CalcularTotalVenta
ON Venta_Productos
AFTER INSERT
AS
BEGIN

    declare @total_actual AS INT, @precio_producto AS INT, @cantidad_producto AS INT, @venta_id AS INT, @rebaja AS MONEY, @venta_fecha
AS DATETIME;

    select @venta_id = venta_id, @cantidad_producto = cantidad FROM inserted;
    select @total_actual = total, @venta_fecha = fecha FROM Venta WHERE venta_id = @venta_id;
    select @precio_producto = precio_venta FROM Producto WHERE producto_id = (SELECT producto_id FROM inserted);

    -- Aplicar precio producto a venta totals
    SET @total_actual = @total_actual + (@precio_producto * @cantidad_producto);

    -- Revisar si hay ofertas que se apliquen a este producto
    declare @oferta_id as INT, @oferta_tipo as varchar(20);
    SELECT TOP 1 @oferta_id = oferta_id, @oferta_tipo = tipo from Oferta WHERE producto_id = (select producto_id from inserted) AND
    (Oferta.fecha_empieza AND Oferta.fecha_termina) AND
    (@venta_fecha BETWEEN
    (CONVERT(TIME, @venta_fecha) BETWEEN
    hora_disponible_empieza AND hora_disponible_termina) AND
    (dias_disponible & POWER(2, DATEPART(DW,
    @venta_fecha-1)) > 0);

    IF (@oferta_tipo = '2x1')
    BEGIN
        SET @rebaja = @precio_producto * (@cantidad_producto / 2);
        IF @rebaja > 0
            INSERT INTO Venta_Ofertas (venta_id, oferta_id, rebaja) VALUES (@venta_id, @oferta_id, @rebaja);
        ELSE
            SET @rebaja = NULL;
    END
    ELSE IF (@oferta_tipo = 'porciento')
    BEGIN
        declare @porciento AS INT;
        set @porciento = (SELECT porciento FROM OfertaPorciento WHERE oferta_id = @oferta_id) / 100;
        SET @rebaja = @precio_producto * @porciento * @cantidad_producto;
        INSERT INTO Venta_Ofertas (venta_id, oferta_id, rebaja) VALUES (@venta_id, @oferta_id, @rebaja);
    END

    IF (@rebaja IS NOT NULL)
        UPDATE Venta SET total = (@total_actual - @rebaja) WHERE venta_id = @venta_id;
    ELSE
        UPDATE Venta SET total = @total_actual WHERE venta_id = @venta_id;

END

```



```

CREATE PROCEDURE sp_InsertSaborHelado @nombre varchar(50), @descripcion varchar(100) = null, @etiqueta_negra bit = 0,
                                     @precio_compra money, @temporal bit = 0, @principio_temporada date = null, @fin_temporada date = null
AS
BEGIN
    INSERT INTO Producto(nombre, descripcion, etiqueta_negra, precio_venta, precio_compra)
    VALUES (@nombre, @descripcion, @etiqueta_negra, null, @precio_compra);

    INSERT INTO SaborHelado(producto_id, temporal, principio_temporada, fin_temporada)
    VALUES(@@IDENTITY, @temporal, @principio_temporada, @fin_temporada);
END

CREATE PROCEDURE sp_ObetenerNFCParaVenta @venta_id INT
AS
BEGIN
    DECLARE @no_nfc AS INT
    SELECT @no_nfc = no_nfc FROM NFC WHERE venta_id IS NULL
    IF (@no_nfc IS NOT NULL)
        UPDATE NFC SET venta_id = @venta_id WHERE no_nfc = @no_nfc
    ELSE
        RAISERROR('No Quedan Numeros NFCs disponibles', 1, 1);
    RETURN @no_nfc
END

```

```

CREATE TRIGGER trg_CalcularTotalVenta
ON Venta_Productos
AFTER INSERT
AS
BEGIN

    declare @total_actual AS INT, @precio_producto AS INT, @cantidad_producto AS INT, @venta_id AS INT, @rebaja AS MONEY, @venta_fecha
AS DATETIME;

    select @venta_id = venta_id, @cantidad_producto = cantidad FROM inserted;
    select @total_actual = total, @venta_fecha = fecha FROM Venta WHERE venta_id = @venta_id;
    select @precio_producto = precio_venta FROM Producto WHERE producto_id = (SELECT producto_id FROM inserted);

    -- Aplicar precio producto a venta totals
    SET @total_actual = @total_actual + (@precio_producto * @cantidad_producto);

    -- Revisar si hay ofertas que se apliquen a este producto
    declare @oferta_id as INT, @oferta_tipo as varchar(20);
    SELECT TOP 1 @oferta_id = oferta_id, @oferta_tipo = tipo from Oferta WHERE producto_id = (select producto_id from inserted) AND
    (Oferta.fecha_empieza AND Oferta.fecha_termina) AND
    (@venta_fecha BETWEEN
    (CONVERT(TIME, @venta_fecha) BETWEEN
    hora_disponible_empieza AND hora_disponible_termina) AND
    (dias_disponible & POWER(2, DATEPART(DW,
    @venta_fecha-1)) > 0);

    IF (@oferta_tipo = '2x1')
    BEGIN
        SET @rebaja = @precio_producto * (@cantidad_producto / 2);
        IF @rebaja > 0
            INSERT INTO Venta_Ofertas (venta_id, oferta_id, rebaja) VALUES (@venta_id, @oferta_id, @rebaja);
        ELSE
            SET @rebaja = NULL;
    END
    ELSE IF (@oferta_tipo = 'porciento')
    BEGIN
        declare @porciento AS INT;
        set @porciento = (SELECT porciento FROM OfertaPorciento WHERE oferta_id = @oferta_id) / 100;
        SET @rebaja = @precio_producto * @porciento * @cantidad_producto;
        INSERT INTO Venta_Ofertas (venta_id, oferta_id, rebaja) VALUES (@venta_id, @oferta_id, @rebaja);
    END

    IF (@rebaja IS NOT NULL)
        UPDATE Venta SET total = (@total_actual - @rebaja) WHERE venta_id = @venta_id;
    ELSE
        UPDATE Venta SET total = @total_actual WHERE venta_id = @venta_id;

END

```

```

/* Turnos */
INSERT INTO Turno(nombre, dias, hora_comienzo, hora_termina) VALUES ('Semana mañana', 31, '09:00:00', '13:00:00');
INSERT INTO Turno(nombre, dias, hora_comienzo, hora_termina) VALUES ('Semana Tarde', 31, '13:00:00', '18:00:00');

/* Empleados */
-- Password is pass123
INSERT INTO Empleado(no_identificacion, tipo_identificacion, correo, password, nombre, apellido, telefono, calle, no_casa, sector,
ciudad, provincia, turno_id, cargo) VALUES
('001-1493849-1', 'cedula', 'aalvarez@bon.com.do', '32250170a0dca92d53ec9624f336ca24', 'Alan', 'Alvarez', '809-482-5924', 'calle 5', '#4
Residencial Las Flores APTO 5A', 'Los Jardines', 'D.N.', 'Santo Domingo', 1, 'supervisor'),
('001-1494939-1', 'cedula', 'rmartinez@bon.com.do', '32250170a0dca92d53ec9624f336ca24', 'Ramon', 'Martinez', '809-820-9857', 'calle 8',
'#25', 'Los Alcarrazos', 'D.N.', 'Santo Domingo', 2, 'vendedor'),
('001-4311939-7', 'cedula', 'lpeguero@bon.com.do', '32250170a0dca92d53ec9624f336ca24', 'Laide', 'Peguero', '809-394-2119', 'San Francisco
de Macoris', 'Gazcue', 'D. N.', 'Santo Domingo', 'Manz. CQ Edif. 2', 1, 'vendedor'),
('223-4896545-1', 'cedula', 'mpena@bon.com.do', '32250170a0dca92d53ec9624f336ca24', 'Manuela', 'Pena', '809-213-2213', 'Rep. de
Paraguay', 'Ens. La Fe', 'D. N.', 'Santo Domingo', 'No. 123', 2, 'supervisor'),
('223-5211298-1', 'cedula', 'llugo@bon.com.do', '32250170a0dca92d53ec9624f336ca24', 'Lissanna', 'Lugo', '809-424-3923', 'Av. Bolivar',
'Ens. Evaristo Morales', 'D. N.', 'Santo Domingo', 'Edif. Rojo Apart. 21', 2, 'vendedor'),
('001-3850284-4', 'cedula', 'ralcantara@bon.com.do', '32250170a0dca92d53ec9624f336ca24', 'Ramon', 'Alcantara', '809-583-2849', 'Calle
Estrella', '#85', 'Villa Mella', 'D.N.', 'Santo Domingo', 1, 'delivery');

/* Clientes */
INSERT INTO Cliente(nombre, apellido, telefono, correo, RNC, calle, no_casa, sector, ciudad, provincia) VALUES
('Alex', 'Figueroa', '829-458-2948', 'afigueroa@correo.com', '130156964', 'Calle F', '#65', 'Bella Vista', 'Santo Domingo', 'D.N.'),
('Ramon', 'Alcantara', '829-432-9481', 'ra@correo.com', '184930485', 'Calle C', '#89', 'Naco', 'Santo Domingo', 'D.N.');
```

```

INSERT INTO Cliente(nombre, apellido, telefono, correo, RNC, calle, no_casa, sector, ciudad, provincia) VALUES
('Ana', 'Gonzalez', '829-123-3489', 'agonz23@yahoo.com', '341-2524731-2', 'Los Robles', 'Edif. Jose, Apart. 21', 'La Julia', 'Santo
Domingo', 'D.N.'),
('Yunik', 'Rafavelo', '809-482-1381', 'rsayunik@outlook.com', '313-3541652-4', 'Atalaya', 'Casa No. 13', 'La Julia', 'Santo Domingo',
'D.N.'),
('Iris', 'Portobelo', '849-342-3489', 'bonito@msn.com', '223-9493423-3', 'Boy Scouts', 'Casa No. 30 ', 'La Yuca', 'Santo Domingo',
'D.N.'),
('John', 'Lugo', '809-348-5729', 'gentebuena@gmai.com', '402-4523745-7', 'Jose A. Aybar', 'Torre Casandra', 'La Julia', 'Santo Domingo',
'D.N.'),
('Luis', 'Suero', '829-988-5398', 'lioq193@hotmail.es', '001-84882952-0', 'Pedro Henriquez U.', 'Bacno Ademi', 'La Julia', 'Santo
Domingo', 'D.N.'),
('Ilucion', 'Cuello', '809-889-6323', 'ilu0900@yahoo.com', '402-8609238-5', 'Desiderio Arias', 'Edificio La Luz', 'La Julia', 'Santo
Domingo', 'D.N.'),
('Keila', 'Montero', '849-219-8429', 'km1505@hotmail.com', '001-5493784-2', 'F. Geraldino esq. Fco. Pratts', '', 'Piantini', 'Santo
Domingo', 'D. N.'),
('Alberto', 'Baldera', '829-399-8984', 'ab_boy10@hotmail.com', '402-3884902-7', 'Pedro Henriquez', 'Edif. Aurora', 'La Julia', 'Santo
Domingo', 'D. N.'),
('Antonio', 'Alfau', '809-482-3872', 'aa.negocios@outlook.com', '001-8487582-9', 'Atalaya', 'Casa No. 23', 'La Julia', 'Santo Domingo',
'D. N')

```

```

/*-----> Insercion de Registros a la Tabla de Caja*/
INSERT INTO Caja (empleado_id, cash_entrada, fecha_abre, fecha_cierra) VALUES
(1, 10000.00, '3/23/2013 09:00:00', '3/23/2013 13:00:00'),
(2, 10000.00, '3/23/2013 13:00:00', '3/23/2013 18:00:00')

INSERT INTO Caja (empleado_id, cash_entrada, fecha_abre, fecha_cierra)
VALUES
(4, 9386, '2/1/2013 08:30:00', '2/1/2013 12:30:00'),
(3, 7556, '2/1/2013 13:00:00', '2/1/2013 19:30:00'),
(3, 7562, '2/2/2013 08:30:00', '2/2/2013 12:30:00'),
(1, 4093, '2/2/2013 13:00:00', '2/2/2013 19:30:00'),
(5, 10026, '2/3/2013 08:30:00', '2/3/2013 12:30:00'),
(3, 6920, '2/3/2013 13:00:00', '2/3/2013 19:30:00'),
(2, 6258, '2/4/2013 08:30:00', '2/4/2013 12:30:00'),

/* Productos */
INSERT INTO Producto(nombre, descripcion, etiqueta_negra, precio_venta, precio_compra) VALUES
('Barquito', NULL, 0, 100.00, 50.00),
('Barquito', NULL, 1, 150.00, 50.00),
('Barquito Abordado', NULL, 0, 120.00, 60.00),
('Barquito Abordado', NULL, 1, 175.00, 60.00),
('Barquilla Danesa', NULL, 0, 50.00, 25.00),
('Barquilla Danesa', NULL, 1, 100.00, 25.00),
('Bonito', NULL, 0, 50.00, 25.00),

/*Insert RegistroInventario*/
INSERT INTO RegistroInventario (empleado_id, notas, fecha) VALUES
(5, NULL, '1/13/2013'),
(3, NULL, '1/14/2013'),
(3, NULL, '1/15/2013'),
(4, NULL, '1/16/2013'),
(1, NULL, '1/17/2013'),
(5, NULL, '1/18/2013'),
(4, NULL, '1/19/2013'),

```

```
/*RegistroInventario_Productos*/
```

```
INSERT INTO RegistroInventario_Productos (producto_id, inventario_id, cantidad) VALUES  
(12, 1, 30),  
(14, 1, 40),  
(2, 1, 60),  
(3, 1, 12),  
(20, 1, 15),  
(22, 1, 12),  
(9, 1, 14),  
(10, 1, 25),  
(13, 1, 10),  
(15, 1, 14),
```

```
/* Ventas */
```

```
INSERT INTO Venta(cliente_id, caja_id, fecha, forma_pago) VALUES  
(1, 1, '3/23/2013 09:15:00', 'efectivo'),  
(NULL, 1, '3/23/2013 09:20:23', 'tarjeta'),  
(NULL, 1, '3/23/2013 10:01:42', 'tarjeta'),  
(NULL, 1, '3/23/2013 10:03:22', 'tarjeta'),  
(2, 1, '3/23/2013 10:20:43', 'efectivo'),  
(NULL, 1, '3/23/2013 10:30:10', 'efectivo'),  
(NULL, 1, '3/23/2013 10:35:22', 'efectivo'),
```

```
INSERT INTO Venta(cliente_id, caja_id, fecha, total, forma_pago) VALUES  
(NULL, 1, '2/1/2013 8:40:22', 300.00, 'Efectivo'),  
(7, 1, '2/1/2013 8:50:24', 700.00, 'Efectivo'),  
(5, 1, '2/1/2013 9:10:36', 650.00, 'Efectivo'),  
(4, 1, '2/1/2013 9:23:52', 200.00, 'Efectivo'),  
(9, 1, '2/1/2013 9:30:48', 425.00, 'Efectivo'),  
(4, 1, '2/1/2013 9:35:10', 350.00, 'Efectivo'),  
(3, 1, '2/1/2013 9:44:58', 50.00, 'Efectivo'),
```

```
INSERT INTO Venta_Productos(venta_id, producto_id, cantidad) VALUES (1, 4, 1);  
INSERT INTO Venta_Productos(venta_id, producto_id, cantidad) VALUES (2, 1, 4);  
INSERT INTO Venta_Productos(venta_id, producto_id, cantidad) VALUES (3, 6, 1);  
INSERT INTO Venta_Productos(venta_id, producto_id, cantidad) VALUES (3, 2, 1);
```

```
/*NCF*/
```

```
INSERT INTO NCF(no_NCF, venta_id) VALUES  
( 'AC000100724101', 1 ),  
( 'AC000100724102', 2 ),  
( 'AC000100724103', 3 ),  
( 'AC000100724104', 4 ),  
( 'AC000100724105', 5 ),
```

```

/*Tabla Suplidor*/
INSERT INTO Suplidor (nombre, descripcion, calle, calle_no, ciudad, provincia, telefono)
VALUES ('Don Confinteria', 'Conos, Chispas y demás', 'Juana Saltitopa', 'No. 42', 'D. N.', 'Santo Domingo', '809-234-5235'),
('Kids Party', 'Decoraciones Festivas', 'Plaza Naco', 'Local 23', 'D. N.', 'Santo Domingo', '809-125-2352'),
('Central Bon', 'Helados, Mermeladas, Otros', 'Zona Industrial de Herrera', 'No. 45', 'D. N.', 'Santo Domingo', '809-523-5123')

/*Orden*/
INSERT INTO Orden (empleado_id, suplidor_id, NOTAS, aceptada, fecha_aceptada, fecha_despachada, recibida, fecha_recibida)
VALUES (6, 3, 'Pedido Semanal de Helados', 1, '2013/03/17', '2013/03/22', 1, '2013/03/22'),
(4, 2, 'Pedido Materiales para Decorar', 1, '2013/03/25', '2013/03/26', 1, '2013/03/26'),
(5, 3, 'Pedido Semanal de Helados', 1, '2013/03/25', '2013/03/29', 1, '2013/03/29'),
(2, 1, 'Pedido Conos', 1, '2013/03/26', '2013/03/27', 1, '2013/03/27'),

CREATE FUNCTION maskToDias (@mask AS INT)
RETURNS VARCHAR(60)
BEGIN
    DECLARE @ret AS VARCHAR(60)
    SET @ret = ''
    IF (@mask & 1 = 1)
        SET @ret = @ret + 'Domingo, '
    IF (@mask & 2 = 2)
        SET @ret = @ret + 'Lunes, '
    IF (@mask & 4 = 4)
        SET @ret = @ret + 'Martes, '
    IF (@mask & 8 = 8)
        SET @ret = @ret + 'Miercoles, '
    IF (@mask & 16 = 16)
        SET @ret = @ret + 'Jueves, '
    IF (@mask & 32 = 32)
        SET @ret = @ret + 'Viernes, '
    IF (@mask & 64 = 64)
        SET @ret = @ret + 'Sabado, '

    IF (LEN(@ret) > 0)
        RETURN SUBSTRING(@ret, 1, LEN(@ret) - 1);

    RETURN @ret
END

CREATE FUNCTION bitToBool (@bit as BIT)
RETURNS nvarchar(2)
BEGIN
    IF (@bit = 1)
        return 'Si'
    return 'No'
END

```