

Ejercicios Resueltos (SQL SERVER)

PARTE 1 : Consultas

```
CREATE DATABASE BANCO
GO
USE BANCO
GO
CREATE TABLE CLIENTE
(IDCLIENTE VARCHAR(5) PRIMARY KEY,
APELLIDOS VARCHAR(30),
NOMBRES VARCHAR(30))
GO
CREATE TABLE CUENTA
(IDCLIENTE VARCHAR(5),
IDCUENTA VARCHAR(5) PRIMARY KEY,
SALDO FLOAT,
DIVISA VARCHAR(20))
GO
CREATE TABLE MOVIMIENTO
(IDCUENTA VARCHAR(5),
IDMOVIMIENTO VARCHAR(5) PRIMARY KEY,
TIPO VARCHAR(2),
MONTO FLOAT,
FECHA DATETIME)
GO
ALTER TABLE CUENTA
ADD FOREIGN KEY(IDCLIENTE) REFERENCES CLIENTE
GO
ALTER TABLE MOVIMIENTO
ADD FOREIGN KEY(IDCUENTA) REFERENCES CUENTA
GO
```

--Según el Modelo Realice las Siguietes Preguntas

--1A --Consultar las Cuentas de los clientes cuyo Apellido empiece por A

```
SELECT B.APELLIDOS + ',' + B.NOMBRES AS CLIENTE, A.IDCUENTA, A.SALDO
FROM CUENTA A INNER JOIN CLIENTE B
ON A.IDCLIENTE = B.IDCLIENTE
WHERE B.APELLIDOS LIKE 'A%'
```

--1B-Consultar todo los Movimientos de las Cuentas en Divisa EURO del Cliente cuyos Apellidos es PEREZ

```
SELECT A.IDCUENTA, C.FECHA, C.MONTO, TIPO = CASE C.TIPO WHEN 'D' THEN 'DEPOSITO' WHEN 'R' THEN 'RETIRO' WHEN 'T' THEN 'TRANSFERENCIA' END
FROM CUENTA A INNER JOIN CLIENTE B ON A.IDCLIENTE = B.IDCLIENTE
INNER JOIN MOVIMIENTO C ON A.IDCUENTA = C.IDCUENTA
WHERE A.DIVISA = 'EUROS' AND B.APELLIDOS = 'PEREZ'
ORDER BY C.FECHA DESC
```

--1C- Consultar todo los movimientos de las Cuentas del Cliente PEREZ en EUROS (Aunque tenga cuentas en otras divisas) Ordenados por el numero de Cuenta

```
SELECT C.NOMBRES, C.APELLIDOS, CU.IDCUENTA, CU.SALDO, CU.DIVISA, M.IDMOVIMIENTO, M.MONTO, M.TIPO
```

```

FROM CLIENTE C INNER JOIN CUENTA CU
ON C.IDCLIENTE = CU.IDCLIENTE
INNER JOIN MOVIMIENTO M
ON CU.IDCUENTA = M.IDCUENTA
WHERE C.APELLIDOS = 'PEREZ'
ORDER BY CU.IDCUENTA

```

--1D -Consultar las Divisas que no tengan ninguna Cuenta Asociada

```

SELECT B.IDCLIENTE, B.APELLIDOS, B.NOMBRES, A.IDCUENTA, A.DIVISA
FROM CUENTA A INNER JOIN CLIENTE B ON A.IDCLIENTE = B.IDCLIENTE
INNER JOIN MOVIMIENTO C ON A.IDCUENTA = C.IDCUENTA
WHERE A.DIVISA = 'SIN CUENTA'

```

--1E -Consultar los Movimientos de Ingreso de PEREZ y los Movimientos de Traspaso de RODRIGUEZ

```

SELECT A.IDCUENTA, C.FECHA, C.MONTO, C.TIPO
FROM CUENTA A INNER JOIN CLIENTE B ON A.IDCLIENTE = B.IDCLIENTE
INNER JOIN MOVIMIENTO C ON A.IDCUENTA = C.IDCUENTA
WHERE C.TIPO = 'DEPOSITO' AND B.APELLIDOS = 'PEREZ' OR
      C.TIPO = 'TRANSACCIONES' AND B.APELLIDOS = 'RODRIGUEZ'
ORDER BY C.FECHA DESC

```

--1F -CREAR UNA FUNCION QUE RETORNE SI UN CLIENTE ES MOROSO

```

CREATE FUNCTION MOROSO(@FECHAPAGO DATETIME)
RETURNS VARCHAR(20)
AS
BEGIN
DECLARE @FECHACLIENTE DATETIME
DECLARE @MENSAJE VARCHAR(20)
SELECT @FECHACLIENTE = FECHA FROM MOVIMIENTO WHERE TIPO = 'DEPOSITO'
IF MONTH(@FECHACLIENTE) < MONTH(@FECHAPAGO)
SET @MENSAJE = 'EL CLIENTE ES MOROSO'
RETURN @MENSAJE
END
--DEMO
SELECT dbo.MOROSO ('15/02/2011')

```

PARTE 2 : Creacion de Objetos - Tablas

--

2A--CREACION DE TABLAS

```

GO
CREATE DATABASE EMPRESA
GO
USE EMPRESA
GO
CREATE TABLE EMPLEADOS
(ID INT NOT NULL IDENTITY,
DOCUMENTO VARCHAR(30) NOT NULL,
APELLIDO VARCHAR(30) NOT NULL,
NOMBRE VARCHAR(30) NOT NULL,
SECCION VARCHAR(20) NOT NULL,
SUELDO FLOAT NULL,
FECHAINGRESO DATETIME NULL)

```

```

GO
CREATE TABLE CONTROLES
(USUARIO VARCHAR(5),
FECHA DATETIME)
GO
-----
CREATE TABLE COPIAEMPLEADOS
(ID INT NOT NULL,
DOCUMENTO VARCHAR(30) NOT NULL,
APELLIDO VARCHAR(30) NOT NULL,
NOMBRE VARCHAR(30) NOT NULL,
SECCION VARCHAR(20) NOT NULL,
SUELDO FLOAT NULL,
FECHAINGRESO DATETIME NULL)
GO
CREATE TRIGGER TRG_COPIAEMPLEADOS
ON EMPLEADOS
FOR INSERT
AS
BEGIN
INSERT INTO COPIAEMPLEADOS
SELECT * FROM INSERTED
END
GO
--2B--INGRESAR REGISTROS
INSERT
INTO EMPLEADOS VALUES ('22999000','CORDOVA','WLIMER','SISTEMAS',680,'24/1
0/2005')
INSERT
INTO EMPLEADOS VALUES ('22333444','ACOSTA','ANA','SECRETARIA',500,'05/05/
2000')
INSERT
INTO EMPLEADOS VALUES ('22777888','CORDOVA','DANIELA','SECRETARIA',560,'2
5/08/2006')
GO
--DEMO
SELECT * FROM EMPLEADOS
SELECT * FROM COPIAEMPLEADOS
--PARTE 3 -- Creacion de Procedimientos Almacenados

--3A--Incrementar Sueldo de los Empleados con mas de 3 Años de Servicio a
un 20% mas
GO
CREATE PROCEDURE USP_EMPLEADOS_AUMENTARSUELDO
AS
DECLARE @FECHA DATETIME
DECLARE @DIF TINYINT
SELECT @FECHA = FECHAINGRESO FROM EMPLEADOS
SET @DIF = YEAR(GETDATE()) - YEAR(@FECHA)
IF MONTH(GETDATE()) < MONTH(@FECHA)
SET @DIF = @DIF - 1
IF MONTH(GETDATE()) = MONTH(@FECHA) AND
DAY(GETDATE()) < DAY(@FECHA)
SET @DIF = @DIF - 1
IF (@DIF >= 3)
UPDATE EMPLEADOS SET SUELDO = (SUELDO + (SUELDO * 0.20))
GO

```

```

--EJECUTAR SOLO UNA VEZ
EXECUTE USP_EMPLEADOS_AUMENTARSUELDO
GO
--VERIFICANDO
SELECT * FROM EMPLEADOS
-----

--3B-Procedimiento que Permite Almacenar valores nulos en los Campos
SUELDO Y FECHADEINGRESO
GO
CREATE PROCEDURE USP_EMPLEADOSINGRESAR
@DOCUMENTO VARCHAR(30),
@APELLIDO VARCHAR(30),
@NOMBRE VARCHAR(30),
@SECCION VARCHAR(20)
AS
BEGIN
INSERT
INTO EMPLEADOS VALUES (@DOCUMENTO,@APELLIDO,@NOMBRE,@SECCION,NULL,NULL)
END
GO
--DEMO
EXECUTE USP_EMPLEADOSINGRESAR '22777889','DURAND','DANY','SECRETARIA'
GO
--VERIFICANDO
SELECT * FROM EMPLEADOS
-----

--3C-Procedimiento que Permite Realizar BACKUP a la Base de Datos EMPRESA
GO
CREATE PROCEDURE COPIADATABASE
AS
BACKUP DATABASE EMPRESA TO DISK = 'F:\\\\BACKUP.bak'
WITH INIT
GO
--DEMO
EXECUTE COPIADATABASE
GO
--PARTE 4-CREACION DE DISPARADORES
-----

--4A -- DISPARADOR QUE REGISTRA EL USUARIO Y LA FECHA EN LA QUE SE
REALIZA UN INSERT EN LA TABLA EMPLEADOS Y LOS ALMACENA EN LA TABLA
CONTROLES
GO
CREATE TRIGGER TRG_CONTROLES
ON EMPLEADOS
FOR INSERT
AS
INSERT INTO CONTROLES VALUES (Suser_Name(),GETDATE())
GO
--DEMO
INSERT
INTO EMPLEADOS VALUES ('22777899','GONZALES','LUIS','SECRETARIA',560,'25/
08/2008')
GO
SELECT * FROM CONTROLES
-----

--4B - DESENCADENADOR QUE DESCUENTA LA EXISTENCIA DE LA TABLA PRODUCTOS
SEGUN EL PEDIDO

```

```

GO
CREATE DATABASE ALMACEN
GO
USE ALMACEN
GO
CREATE TABLE PRODUCTOS
  (ID_PRODUCTO CHAR(8) PRIMARY KEY NOT NULL,
  NOMBREPRODUCTO VARCHAR(25) NOT NULL,
  EXISTENCIA INT NULL,
  PRECISO DECIMAL(10,2) NOT NULL,
  PRECIOVENTA DECIMAL (10,2))
GO
CREATE TABLE PEDIDO
  (ID_PEDIDO INT IDENTITY,
  ID_PRODUCTO CHAR(8) NOT NULL,
  CANTIDAD_PEDIDO INT
  CONSTRAINT PK_ID_PRODUCTO FOREIGNKEY (ID_PRODUCTO) REFERENCES PRODUCTOS (ID
  _PRODUCTO))
GO
--DESENCADENADOR
CREATE TRIGGER TRG_PEDIDO_ARTICULOS
ON PEDIDO
FOR INSERT
AS
UPDATE PRODUCTOS SET EXISTENCIA = EXISTENCIA -
  (SELECT CANTIDAD_PEDIDO FROM INSERTED)
WHERE ID_PRODUCTO = (SELECT ID_PRODUCTO FROM INSERTED)
GO
INSERT INTO PRODUCTOS VALUES ('P001', 'FILTROS PANTALLA', 5, 10, 12.5)
INSERT INTO PRODUCTOS VALUES ('P002', 'PARLATES', 7, 10, 11.5)
INSERT INTO PRODUCTOS VALUES ('P003', 'MAUSE', 8, 4.5, 6)
GO
--DEMO
INSERT INTO PEDIDO VALUES ('P003', 5)
--VERIFICANDO
SELECT * FROM PRODUCTOS
SELECT * FROM PEDIDO

```