Actividad de replicar los ejercicios propuestos

-- ejercicios en clases

CREATE DATABASE dbcuentas

GO

USE dbcuentas

GO

CREATE TABLE clientes(

id varchar(50),

nombre varchar(50),

apellido1 varchar(50),

apellido2 varchar(50),

ci varchar(20),

f\_nac datetime,

primary key(id)

);

CREATE TABLE cuentas(

idcuenta int,

idcli varchar(50),

numcuenta varchar(50),

saldo float,

primary key (idcuenta),

foreign key (idcli) REFERENCES clientes(id)

);

CREATE TABLE hco\_saldos(

idcuenta int,

saldo float,

f\_saldo date,

foreign key (idcuenta) references cuentas(idcuenta)

);

CREATE TABLE movimientos(

idcuenta int,

saldo\_anterior float,

saldo\_posterior float,

importe float,

f\_movimiento date,

foreign key (idcuenta) references cuentas(idcuenta)

);

--datos

insert into clientes values('c1','juan','perez','quispe','123456','2000/10/10');

insert into clientes values('c2','ana','torrez','lima','234567','1990/01/01');

insert into cuentas values(1,'c1','1-00011',500);

insert into cuentas values(2,'c2','1-00012',350);

insert into cuentas values(3,'c1','1-00013',1000);

CREATE PROCEDURE SPU\_FACTORIAL @NUM INT

AS

DECLARE @C INT ,@F INT

SET @C=1

SET @F=1

BEGIN

WHILE (@C<=@NUM)

BEGIN

SET @F=@F\*@C

SET @C=@C+1

END

PRINT @F

END

-- INVOCACION

DECLARE @NUM INT

SET @NUM =5

EXECUTE SPU\_FACTORIAL @NUM

ALTER PROCEDURE SPU\_FACTORIAL1 @NUM INT,@F INT OUTPUT

AS

DECLARE @C INT

SET @C=1

BEGIN

WHILE (@C<=@NUM)

BEGIN

SET @F=@F\*@C

SET @C=@C+1

END

END

GO

--INVOCAION

DECLARE @NUM INT,@F INT

SET @NUM =5

SET @F =1

EXECUTE SPU\_FACTORIAL1 @NUM ,@F OUTPUT

GO

ALTER PROCEDURE SPU\_CONTVOCALES1 @CAD VARCHAR(50),@VC INT OUTPUT

AS

DECLARE @C INT,@N INT ,@CA CHAR(1)

SET @C =1

SET @N = LEN(@CAD)

SET @VC = 0

BEGIN

WHILE(@C<=@N)

BEGIN

SET @C=SUBSTRING(@CAD,@C,1)

IF(@CA= 'a' or @CA= 'e' or @CA= 'i' or @CA= 'o' or @CA= 'u')

set @VC =@VC + 1

SET @C=@C+1

END

END

GO

-- INVOCACION

DECLARE @CAD VARCHAR(50),@VC INT

SET @CAD ='HOLA MUNDO TRANSACT'

SET @VC=0

EXECUTE SPU\_CONTVOCALES1 @CAD , @VC OUTPUT

PRINT @CAD

PRINT CAST(@VC AS VARCHAR) + ' VOCALES'

CREATE PROCEDURE SPU\_CANTIDAD\_DEPALABRAS @CAD VARCHAR(50),@CP INT OUTPUT

AS

DECLARE @C INT,@N INT ,@CA CHAR(1)

SET @C =1

SET @CAD=LTRIM(@CAD)

SET @CAD=RTRIM(@CAD)

SET @N=LEN(@CAD)

BEGIN

WHILE(@C<=@N)

BEGIN

SET @C=SUBSTRING(@CAD,@C,1)

IF (@CA = ' ')

set @CP =@CP + 1

SET @C=@C+1

END

IF @N!=0

SET @CP = @CP+1

END

GO

--INVOCAION

DECLARE @CAD VARCHAR(50),@CP INT

SET @CAD='HOLA MUNDO TRANSACT'

SET @CP = 0

EXECUTE SPU\_CANTIDAD\_DEPALABRAS @CAD ,@CP OUTPUT

PRINT @CAD

PRINT CAST(@CP AS VARCHAR) + 'PALABRAS'

GO

CREATE PROCEDURE SPU\_ADDCLIENTE @ID VARCHAR(50),@NOMBRE VARCHAR(50),@APELLIDO1 VARCHAR(50),@APELLIDO2 VARCHAR(50),

@CI VARCHAR(50),@F\_NAC DATETIME

AS

INSERT INTO clientes(id,nombre,apellido1,apellido2,ci,f\_nac)

VALUES(@ID,@NOMBRE,@APELLIDO1,@APELLIDO2,@CI,@F\_NAC)

GO

--INVOCAION

DECLARE @ID VARCHAR(50),@NOMBRE VARCHAR(50),@APELLIDO1 VARCHAR(50),@APELLIDO2 VARCHAR(50),

@CI VARCHAR(50),@F\_NAC DATETIME

SET @ID = 'C12'

SET @NOMBRE = 'HUGO'

SET @APELLIDO1 = 'MARTINEZ'

SET @APELLIDO2 = 'PEÑA'

SET @CI = '8465'

SET @F\_NAC = '01/02/1990'

EXEC SPU\_ADDCLIENTE @ID ,@NOMBRE ,@APELLIDO1 ,@APELLIDO2 ,@CI ,@F\_NAC

SELECT \* FROM clientes;

CREATE PROCEDURE SPU\_OBTENERSALDODECUENTA @NUMCUENTA VARCHAR(20),@SALDO DECIMAL(10,2) OUTPUT

AS

BEGIN

SELECT @SALDO = saldo

FROM cuentas

WHERE numcuenta =@NUMCUENTA

END

--INVOCACION

DECLARE @NUMCUENTA VARCHAR(20),@SALDO DECIMAL(10,2)

SET @NUMCUENTA = '1-00012'

SET @SALDO=0

EXEC SPU\_OBTENERSALDODECUENTA @NUMCUENTA,@SALDO OUTPUT

PRINT @SALDO

GO

CREATE PROCEDURE SPU\_LISTADECLIENTESCUENTAS @NUMCLIE VARCHAR (20)

AS

BEGIN

SELECT C.id,C.nombre,C.apellido1,C.apellido2,CU.numcuenta

from clientes c, cuentas cu

where cu.idcli=@NUMCLIE and cu.idcli = c.id

order by cu.numcuenta desc

end

-- invocaion

exec SPU\_LISTADECLIENTESCUENTAS 'c1'

go

create function fn\_clienteApellidoX(@ape1 varchar(50))

returns table

as

return

(

select \* from clientes where apellido1 = @ape1

)

--invocacion declare