-- determinar ek factorial de un numero entero

-- SAUL ESCOBAR SERRANO

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

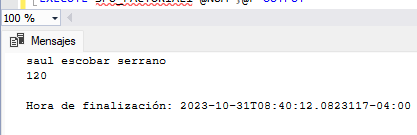
-- INVOCACION

DECLARE @NUM INT

SET @NUM =5

print 'saul escobar serrano'

EXECUTE SPU\_FACTORIAL @NUM



--2) determine el factorial con un numero enetero

create PROCEDURE SPU\_FACTORIAL\_2 @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

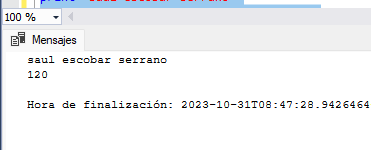
SET @F =1

print 'saul escobar serrano'

EXECUTE SPU\_FACTORIAL\_2 @NUM ,@F OUTPUT

print @F

GO

+

-- ejercicio extra programa que calcule el coeficiente}

-- y restar de dos numeros enteros

CREATE PROCEDURE SPU\_DIVISION @a INT ,@b int ,@c int output , @r int output

AS

begin

if(@b!=0)

begin

set @c = @a / @b

set @r = @a % @b

end

else

begin

print 'error de division por cero'

set @c = -99999999

set @r = -9999999

end

end

-- invocacion

declare @a int ,@b int, @c int ,@r int

set @a = 8

set @b = 5

set @c = 0

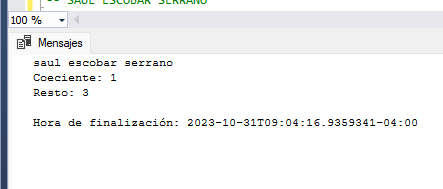
set @r = 0

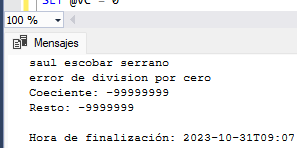
print 'saul escobar serrano'

execute SPU\_DIVISION @a ,@b ,@c output , @r output

print 'Coeciente: ' + CONVERT(VARCHAR , @c)

print 'Resto: ' + CONVERT(VARCHAR , @r)





-- 4) eaizar un sp para mustrara a todos los clientes

create procedure sp\_listar\_cientes

as

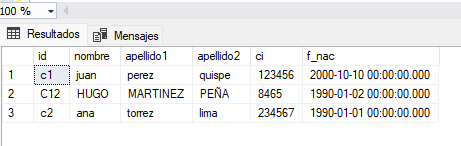
begin

select \* from clientes

end

print 'saul escobar serrrano'

execute sp\_listar\_cientes



-- 5) crear un sp para listar los datos del cliente spm id=x

create procedure sp\_mostrar\_datos\_por\_id @id varchar(50)

as

begin

select\*

from clientes

where id = @id

end

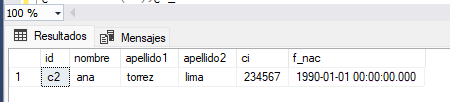
-- invocacion

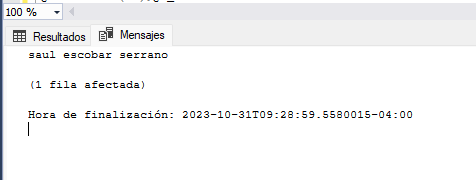
declare @id varchar(50)

set @id = 'c2'

print 'saul escobar serrano'

execute sp\_mostrar\_datos\_por\_id @id





-- 6)crear un sp para registrar un cliente

CREATE PROCEDURE SPU\_insertar\_al\_cliente @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 = 'C3'

SET @NOMBRE = 'Rosa'

SET @APELLIDO1 = 'Perez'

SET @APELLIDO2 = 'Quiroz'

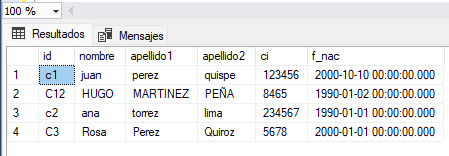
SET @CI = '5678'

SET @F\_NAC = '01/01/2000'

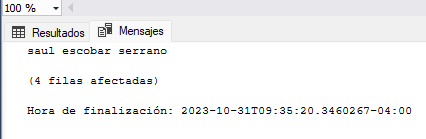
EXEC SPU\_insertar\_al\_cliente @ID ,@NOMBRE ,@APELLIDO1 ,@APELLIDO2 ,@CI ,@F\_NAC

print 'saul escobar serrano'

SELECT \* FROM clientes;



El registro de c12 hugo martinez era de la practca ya planteada



--7) crear un sp para modificar los datos de un cliente = x

alter PROCEDURE SPU\_modificar\_al\_cliente @ID VARCHAR(50),@NOMBRE VARCHAR(50),@APELLIDO1

VARCHAR(50),@APELLIDO2 VARCHAR(50),

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

AS

update clientes set nombre = @NOMBRE , apellido1 = @APELLIDO1 ,

apellido2 = @APELLIDO2 , ci = @CI ,f\_nac = @F\_NAC

where id = @ID

GO

--INVOCAION

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

VARCHAR(50),

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

SET @ID = 'C3'

SET @NOMBRE = 'Mariel'

SET @APELLIDO1 = 'Tola'

SET @APELLIDO2 = 'Ramos'

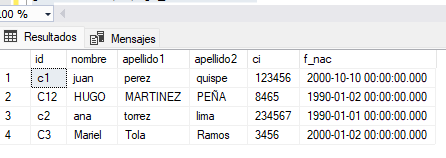
SET @CI = '3456'

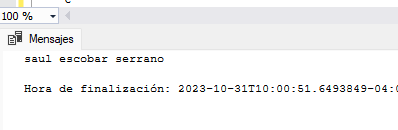
SET @F\_NAC = '01-02-2000'

EXEC SPU\_modificar\_al\_cliente @ID ,@NOMBRE ,@APELLIDO1 ,@APELLIDO2 ,@CI ,@F\_NAC

print 'saul escobar serrano'

SELECT \* FROM clientes;





-- eliminar datos del clientes en sp

create PROCEDURE SPU\_eliminar\_al\_cliente @ID VARCHAR(50)

AS

delete from clientes where id = @ID

GO

--INVOCAION

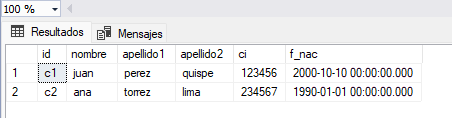
DECLARE @ID VARCHAR(50)

set @ID = 'c3'

exec SPU\_eliminar\_al\_cliente @ID

SELECT \* FROM clientes;

--saul escobar



--9) crear un sp para que devuelva los datos del cliente de una determinada cuenta

create procedure ps\_mostrar\_datos\_del\_cliente @nroCuenta varchar(50)

as

begin

select a.nombre,a.apellido1,a.apellido2,a.ci,b.numcuenta

from clientes as a , cuentas as b

where (a.id = b.idcli) and b.numcuenta = @nroCuenta

end

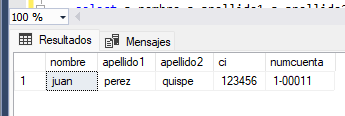
--invococaion

declare @nroCuenta varchar (50)

set @nroCuenta = '1-00011'

execute ps\_mostrar\_datos\_del\_cliente @nroCuenta

* Saul escobar



--10) crear un sp para mostrar las cuentas que tiene un cliente con su ci = x

create procedure ps\_mostrar\_cuentas\_del\_cliente\_con\_ci @CI VARCHAR (20)

as

begin

select a.nombre,a.apellido1,a.apellido2,a.ci,b.saldo,b.idcuenta

from clientes as a , cuentas as b

where (a.id = b.idcli) AND a.ci = @CI

end

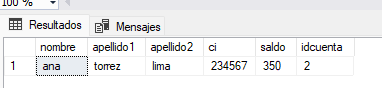
--INCOCACION

declare @CI VARCHAR(20)

set @CI = '234567'

execute ps\_mostrar\_cuentas\_del\_cliente\_con\_ci @CI

--saul escobar



--11) crear una funcion de usuario que recibe el primer apellido de un clienete y devuelve todos sus datos

create function fn\_muestra\_datoscliente\_por\_apellido (@apellido1 varchar(50))

returns table

as

return( select\*

from clientes

where apellido1 = @apellido1

)

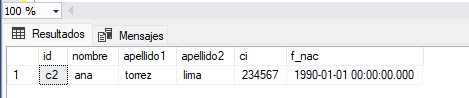
-- invocacion

declare @apellido1 varchar(50)

set @apellido1 = 'torrez'

select \* from fn\_muestra\_datoscliente\_por\_apellido(@apellido1)

* saul escobar



-- 12) crear una funcion de usuario que recibe el numero de cuenta de un cliente

-- y devuelve el ci y saldo de la cuenta

alter function fn\_muestra\_datoscliente\_ci\_y\_saldo\_deCuenta1(@nrocuenta varchar(50))

returns table

as

return(

select a.ci,b.numcuenta,b.saldo

from clientes as a , cuentas as b

where (a.id = b.idcli) and b.numcuenta = @nrocuenta

)

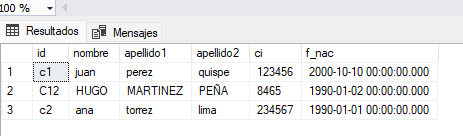
--invocaion

declare @nrocuenta varchar(50)

set @nrocuenta = '1-00011'

select \* from fn\_muestra\_datoscliente\_ci\_y\_saldo\_deCuenta1 (@nrocuenta)

* Saul escobar



--13) crear una funcion de usuario que devuelva eel nombre y apellidos

-- fecha de nac norcuenta y saldo del cliente a traves de su id

create function fn\_mostrar\_datos\_cliente\_cuenta\_por\_su\_id (@id varchar(50))

returns @tabla\_cliente\_cuenta table(

id varchar (50),

nombre varchar (50),

apellido1 varchar (50),

apellido2 varchar (50),

f\_nac date,

nro\_cuenta varchar (50),

saldo float

)

as

begin

insert into @tabla\_cliente\_cuenta

select a.id,a.nombre,a.apellido1,a.apellido2,a.f\_nac,b.numcuenta,b.saldo

from clientes as a , cuentas as b

where (a.id = b.idcli) and a.id = @id

return

end

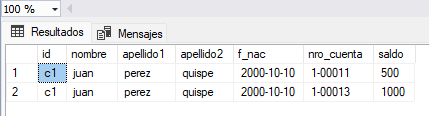
-- invovar

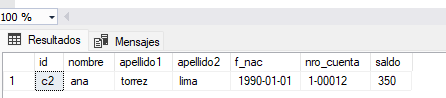
declare @id varchar(50)

set @id = 'c1'

select \* from fn\_mostrar\_datos\_cliente\_cuenta\_por\_su\_id (@id)

* saul escobar





--14) crear una funcion de usuario que devuelva ci,nombre,apellido1,nrocuenta y saldo

-- de un numero de cuenta x

alter function fn\_mostrar\_los\_datos\_Del\_usuario\_que\_devuelva\_el\_ci1 (@nrocuenta varchar(50))

returns @tabla\_cliente\_cuenta1 table(

ci varchar (20),

nombre varchar (50),

apellido1 varchar (50),

nro\_cuenta varchar (50),

saldo float

)

as

begin

insert into @tabla\_cliente\_cuenta1

select a.ci,a.nombre,a.apellido1,a.ci,b.numcuenta,b.saldo

from clientes as a , cuentas as b

where (a.id = b.idcli) and b.numcuenta = @nrocuenta

return

end

-- invocaion

declare @nrocuenta varchar(50)

set @nrocuenta = '1-00012'

select \* from fn\_mostrar\_los\_datos\_Del\_usuario\_que\_devuelva\_el\_ci1(@nrocuenta)