-- Declare Variable de type table

Begin

Declare @agence\_copy table

(numag varchar(15) primary key not null,

nomag varchar(15),

adresseag varchar(15),

telag numeric)

insert into @agence\_copy select \* from agence where adresseag = 'Fes'

select \* from @agence\_copy

End

----------

-- Condition if

Declare @a int

set @a= 5

if(@a>4)

begin

print 'HI'

print 'Hi again'

End

----------

-- Condition case 1

Select \*, 'Categories' =

case

when pu < 40 then 'cat1'

when pu < 1000 then 'cat2'

when pu < 2000 then 'cat3'

Else 'cat4'

End

From produit

Order by categories

-- Condition case 2

Select \*, 'Validte' =

case adresseag

when 'casa' then 'Oui'

when 'ksar' then 'Oui'

Else 'Non'

End

From Agence

-- Condition case 3

select client.numclt, client.nomclt, sum(facture.ttc) as total\_achat, 'remise' =

case

when sum(facture.ttc) < 4000 then 0

when sum(facture.ttc) between 4000 and 5000 then sum(facture.ttc) \* 0.3

when sum(facture.ttc) < 10000 then sum(facture.ttc) \* 0.5

else sum(facture.ttc) \* 0.7

end,'Type\_Client' =

case

when sum(facture.ttc) < 1000 then 'Passager'

when sum(facture.ttc) between 1000 and 5000 then 'Potentielle'

else 'Fideil'

end

from client, facture

where client.numclt = facture.numclt

group by client.numclt, client.nomclt

----------

-- LOOP While 1

Declare @moy float, @val int, @i int,@count int

set @moy = (select \* from facture)

set @val = 5000

set @i = 1

set @count = (select count(\*) from facture)

while (@i<@count)

begin

if (@moy < @val)

begin

update facture set tht = tht + tht \* 0.5, tva= tva + tva \* 0.5, ttc = ttc + ttc \* 0.5

end

set @i=@i+1

end

print 'Les facture est bien modifier ' + ltrim(str(@i)) + ' fois'

----------

-- procédures stockées simple

Create proc ps\_Fact as

begin

select \* from facture

end

-- procédures stockées avec des parametre

create proc infoClient(@adr varchar(20)) as

begin

select \* from client where adresseclt = @adr

end

exec infoClient 'ksar'

-- fonction

create function nbrClt() returns float as

begin

Return (select count(\*) from client)

end

select dbo.nbrClt()

-- fonction avec parametre d'entree 1

create function Multip(@nbr float) returns float as

begin

return @nbr \* 2

end

select dbo.Multip(10)

-- fonction avec parametre d'entree 5 de type table

create function Total\_Achat(@numclt varchar(30))

returns @tab Table(

numclt varchar(15) primary key,

nomclt varchar(15),

prnclt varchar(15),

adresseclt varchar(15),

Total numeric) as

begin

insert into @tab

select client.numclt, client.nomclt, client.prnclt, client.adresseclt, sum(facture.ttc)

from client, facture where client.numclt = facture.numclt and facture.numclt = @numclt

group by client.numclt, client.nomclt,client.prnclt, client.adresseclt

return

end

select \* from dbo.Total\_Achat('cl001')

----------

-- create a cursor

DECLARE CurNumClt CURSOR FOR select numclt from client

-- open cursor

OPEN CurNumClt0

-- insert cursor values in a variable

DECLARE @Num\_Client varchar(30)

FETCH NEXT from CurNumClt INTO @Num\_Client

-- start executing cursor

WHILE @@FETCH\_STATUS = 0

BEGIN

print @Num\_Client

FETCH NEXT from CurNumClt Into @Num\_Client

END

-- close cursor

CLOSE CurNumClt

-- delete cursor

DEALLOCATE CurNumClt

/\*Simple cursor\*/

-- Last

DECLARE Cur1 Scroll CURSOR FOR select numclt from client

OPEN Cur1

DECLARE @Id\_Client varchar(30)

FETCH Last from Cur1 INTO @Id\_Client

print @Id\_Client

CLOSE Cur1

DEALLOCATE Cur1

-- First

DECLARE Cur2 Scroll CURSOR FOR select numclt from client

OPEN Cur2

DECLARE @Id\_Client varchar(30)

FETCH First from Cur2 INTO @Id\_Client

print @Id\_Client

CLOSE Cur2

DEALLOCATE Cur2

-- Absolute

DECLARE Cur3 Scroll CURSOR FOR select numclt from client

OPEN Cur3

DECLARE @Id\_Client varchar(30)

FETCH Absolute 2 from Cur3 INTO @Id\_Client

print @Id\_Client

CLOSE Cur3

DEALLOCATE Cur3

-- Prior

DECLARE Cur4 Scroll CURSOR FOR select numclt from client

OPEN Cur4

DECLARE @Id\_Client varchar(30)

FETCH Prior from Cur4 INTO @Id\_Client

print @Id\_Client

CLOSE Cur4

DEALLOCATE Cur4

----------

-- trigger

-- INSTEAD OF

-- ex1

create table Coureur

(IdCr int primary key,

NomCr varchar(20),

Age int)

insert into Coureur Values

('1','mohamed',20)

insert into Coureur Values

('2','said',21)

insert into Coureur Values

('3','Laaguili',22)

CREATE TABLE Competition

(IdComp int primary key,

DateComp DATETIME,

Ville varchar(20),

IdCour int references Coureur(IdCr))

INSERT INTO Competition values

('1','22/04/2023','ksar','2')

select \* from Coureur

select \* from Competition

Delete from Coureur where IdCr=1

CREATE TRIGGER Tg\_Sup\_Cour ON Coureur

INSTEAD OF DELETE

AS

BEGIN

DELETE FROM Competition WHERE IdCour = (SELECT IdCr FROM deleted)

DELETE FROM Coureur WHERE IdCr = (SELECT IdCr FROM deleted)

END

-- INSTEAD OF

-- ex2

CREATE TABLE Competition\_Copy

(IdComp int,

DateComp DATETIME,

Ville varchar(20),

IdCour int)

Delete from Competition where IdComp = 4

CREATE TRIGGER Tg\_Ins\_Comp ON Competition

INSTEAD OF DELETE

AS

BEGIN

insert into Competition\_Copy select \* from Competition where IdComp = (SELECT IdComp FROM deleted)

DELETE FROM Competition WHERE IdComp = (SELECT IdComp FROM deleted)

END

select \* from Competition

select \* from Competition\_Copy

-- After

insert into Competition values

('6','26/04/2023','ksar','2')

CREATE TRIGGER Message\_Deleted\_Competetion ON Competition

AFTER Insert

AS

BEGIN

print 'Competition Added Successfully'

END