TEST DATABASE + creation d’une base de donnée

use master

go

if exists (select \* from master.sys.databases where name = ‘nomdetabdd’ )

drop database nomdetabdd

go

create database nomdetabdd

go

TEST TABLE + creation d’une table

use nomdetabdd

go

if exists (select \* from nomdetabdd.sys.tables where name = ‘nomdetatable’ )

drop table nomdetatable

go

create table nomdetatable

(identifiant int, nom varchar(255), prenom varchar(255), position int)

go

INSERER des données dans une table

insert into nomdetatable (identifiant, nom, prenom, position) values (‘1’,’Chazal’,’Claire’,’0’)

insert into nomdetatable (identifiant, nom, prenom, position) values (‘2’,Morgane,’Clara’,’69’)

insert into nomdetatable (identifiant, nom, prenom, position) values (‘3’,’Bande’,’Jamas’,’007’)

insert into nomdetatable (identifiant, nom, prenom, position) values (‘4’,’Soumise’,’Kat’,’71’)

insert into nomdetatable (identifiant, nom, prenom, position) values (‘5’,’Bandera’,’Antonos’,’99’)

go

AJOUTER une colonne dans une table

alter table nomdetatable

add nomdetacolonne varchar(255)

go

METTRE à jour des données pour la nouvelle colonne

update nomdetatable

set nomdetacolonne = ‘texteàajouter’

go

TEST TABLE temporaire + creation d’une table temporaire

use nomdetabdd

go

if object\_id (‘tempdb.dbo.#temporaire’) is not null

drop table #temporaire

go

select \* into #temporaire from nomdetatable

go

TEST FONCTION

use nomdetabdd

go

if object\_id (’dbo.nomdetafonction’) is not null

drop function dbo.nomdetafonction

go

CREATION d’une fonction

use nomdetabdd

go

create function dbo.nomdetafonction (@date1 datetime, @date2 datetime)

returns int

as begin

return abs(datediff(day,@date1,@date2))

end

go

UTILISATION d’une fonction (exemple)

use nomdetabdd

go

select dbo.nomdetafonction ( ‘2019-6-4’, getdate() ) as difference

🡺 uilisation comme les fonctions déjà existante (AVG ; MAX ; MIN…)

DECLARATION de variable

declare @var1 as int

set @var1 = 6

go

🡺 ou bien en une ligne

declare @var2 int = 12

go

AFFICHER une variable

declare @var3 int = 59

select @var3 nomdereferencecolonne

go

TRAVAILLER avec des variables

use nomdetabdd

go

declare @firstname as varchar(25), @lastname as varchar(25)

set @firstname = (select prenom from nomdetatable where identifiant = 3)

set @lastname = (select nom from nomdetatable where identifiant = 3)

select @firstname as un\_prénom, @lastname as un\_nom

go

🡺 ou autrement

declare @firstname as varchar(25), @lastname as varchar(25)

select @firstname = prenom, @lastname = nom from nomdetatable where identifiant = 3

select @firstname as un\_prénom, @lastname as un\_nom

go

DECLARATION d’un curseur

declare nomdetoncurseur cursor for (select \* from nomdetatable where position = 69)

open nomdetoncurseur

fetch nomdetonpointeur into @id, @nom, @prenom, @position

go

* attention, ne pas oublier de déclarer les variables (@id, @nom, @prenom, @position) avant la déclaration du curseur
* le nombre de variable dans le fetch, correspond au nombre de colonnes que l’on a sélectionné dans la déclaration du curseur

UTILISATION du curseur

while @@fetch\_status = 0

begin

select @id, @nom, @prenom, @position

fetch nomdetoncurseur

end

close nomdetoncurseur

deallocate nomdetoncurseur

go

JOINTURE de table (utilisation de tables présentes dans la bdd AdventureWorks2017)

select PP.ProductID, PP.name, PP.ProductNumber from Production.Product as PP

inner join Production.ProductInventory as PPI

on PP.ProductID = PPI.ProductID

go

🡺 en plus complexe avec des tables temporaires (sans tester si elles existent…) 🡺

select ProductID, Name, ProductNumber, Color, ListPrice into #temp1 from Production.Product

select ProductID, LocationId, Quantity into #temp2 from Production.ProductInventory

select PP.ProductID, PP.Name, PP.ProductNumber, PP.ListPrice, PPI.Quantity into #temp3 from #temp1

inner join #temp2 on PP.ProductID = PPI.ProductID

go

PRINT et CAST (par l’exemple)

declare @id int,

@nom varchar(50),

@prenom varchar(50),

@position int

select @id = identification, @nom = nom, @prenom = prenom, @position = position from nomdetatable where identifiant = 2

print ‘l actrice ‘ + cast(@prenom as varchar(50)) + ‘ ‘ + cast(@nom as varchar(50)) + ‘ aime la position ‘ + cast(@position as int)

🡺 à venir 🡺 bientôt 🡺 ☺

case