Correction SGBDR Mai2013

```
--1.a
CREATE DATABASE BD_Inscription
ON (
NAME = 'Inscription_Data',
FILENAME = 'C:\Inscription_Data.mdf' ,
SIZE = 5MB ,
MAXSIZE = 1GB ,
FILEGROWTH = 5%)
LOG ON (
NAME = 'Inscription_log',
FILENAME = 'C:\Inscription_log.ldf' ,
SIZE = 5MB ,
MAXSIZE = UNLIMITED,
FILEGROWTH = 10MB);
use BD_Inscription
go

--1.b
create type designation from varchar(100) not null

--1.c
use master
go
create login adminConnexion
    with password = '4444';
use BD_Inscription;
go
CREATE USER admin1 FOR LOGIN adminConnexion;
grant create table to admin1
grant select to admin1
grant insert to admin1
grant update to admin1
grant delete to admin1

--2
create table departement (codeD int primary key, designationD
designation )
create table Professeur (codeP int primary key  , nom
nvarchar(23),prenom nvarchar(23),
                    sexe char, dateNaissance datetime,grade
nvarchar(23),specialite nvarchar(23),diplome nvarchar(23),departement
int,
              Constraint fk_departementP foreign key(departement)
references departement( codeD))
```

```
create table Filiere (codeF int primary key, designationF
designation,departement int,
            Constraint fk_departementF foreign key(departement)
references departement( codeD))

alter table Professeur
add constraint cont_age check((year(getDate())-
year(dateNaissance))>20)
alter table departement
add constraint cont_design unique(designationD)
alter table Professeur
add constraint cont_sexe check (sexe IN ('M','F'))


insert into departement values(11,'informatique')
insert into departement values(12,'mathématique')

 insert into Professeur values (1,'rmiki','noha','F','12/09/1977'
,'première','informatique','licence',11)
insert into Professeur values (2  , 'El Faiz','adil' , 'M',
'08/07/1973','première','informatique','master',11)
insert into Professeur values (3  , 'Alami', 'hassan' ,'M',
'1982/11/03','deuxième','informatique','licence',11)
insert into Professeur values (4 , 'Nasri', 'said','M',
'1962/2/2','première','mathématique','licence',12)
insert into Professeur values (5   ,'Bakari','brahim'  ,'M',
'1976/09/06','deuxième','mathématique','master',12)

insert into Filiere values(31,'BTS DSI',11)
insert into Filiere values(32,'BTS RI',11)
insert into Filiere values(33,'ANALYSE',12)
insert into Filiere values(34,'ALGEBRE',12)

--3.a
create function liste_profs(@designation nvarchar(23))
returns table
as
return (select nom,prenom, grade from professeur p ,departement as d
    where p.departement = d.codeD
    and designationD=@designation )

select * from dbo.liste_profs('informatique') order by grade

--3.b
create procedure ps_age
as begin
declare @n nvarchar(23), @p nvarchar(23),@d datetime
declare @age int
declare cr cursor for select nom,prenom , dateNaissance from
professeur
open cr
```

```
fetch next from cr into @n,@p,@d
while @@fetch_status=0
begin
set @age=year(getDate()) - year(@d)
print @n+' '+@p +' '+ convert(nvarchar(23),@age)
fetch next from cr into @n,@p,@d
end
close cr
deallocate cr
end
exec ps_age

--3.c
create trigger tr_c on departement
instead of delete
as
begin
 delete from Filiere where departement IN (select codeD from deleted)
 delete from professeur where departement IN (select codeD from
deleted)
 delete from departement where codeD IN (select codeD from deleted)
end

--3.d
create trigger tr_d on departement
instead of update
as
begin
if update(codeD)
begin
 declare @nouveau_codeD int,@ancien_codeD int,@design varchar(50)
 declare cr cursor for select deleted.codeD,inserted.codeD ,
             deleted.designationD from deleted,inserted
               where inserted.designationD=deleted.designationD
 open cr
 fetch next from cr into @ancien_codeD,@nouveau_codeD,@design
 while @@fetch_status=0
  begin
  insert into departement values(@nouveau_codeD,'temporaire')
  update Filiere set departement=@nouveau_codeD
         where departement=@ancien_codeD
  update professeur set departement=@nouveau_codeD
         where departement=@ancien_codeD
  delete from departement where codeD=@ancien_codeD
  update departement set designationD=@design
         where codeD=@nouveau_codeD
  fetch next from cr into @ancien_codeD,@nouveau_codeD,@design
 end
 close cr
 deallocate cr
 end   --end if
```

```
 if update(designationD)
 begin
 update departement set designationD=inserted.designationD
    from departement,inserted
     where departement.codeD=inserted.codeD
end
end
```

Remarque :

Ce trigger prend en considération :

- Les contraintes de foreign key du champ codeD
- La contrainte unique du champ designationD
- La mise à jour de plusieurs lignes pour ça nous avons utilisé le curseur
- La mise à jour de tous les champs