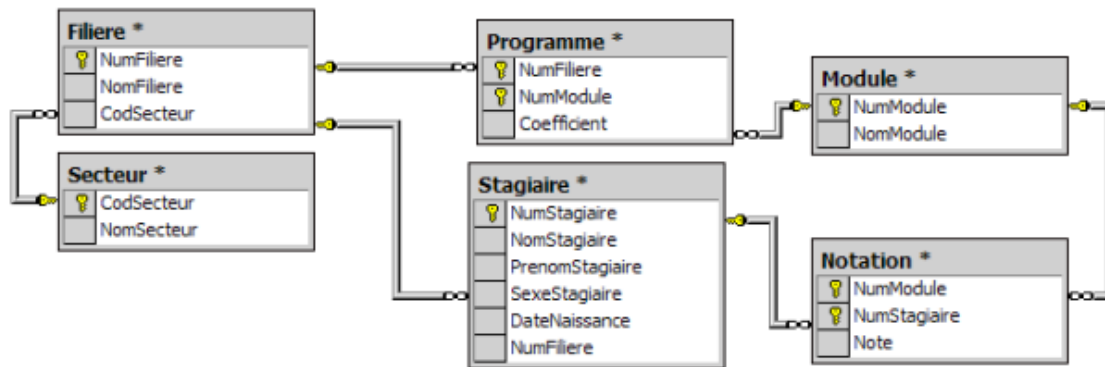


### Exercice 3

Soit la base de données suivante :



Créer les procédures stockées suivantes :

- PS 1.** Qui affiche les numéros et les noms des stagiaires pour lesquels on a pas encore saisi de note
- PS 2.** Qui affiche les filières ayant plus de 10 modules au programme
- PS 3.** SP\_12 qui affiche les noms des modules qui sont étudiés dans toutes les filières d'un secteur donné en paramètre
- PS 4.** Qui affiche pour un stagiaire donné en paramètre, la liste des notes (numéro module, nom du module, note et coefficient)
- PS 5.** Qui affiche pour chaque stagiaire :
- Le nom, le prénom et la filière sous la forme :  
Nom et Prénom :.....Filière : .....
  - S'il existe des modules où le stagiaire n'a pas de notes attribuée afficher le message 'En cours de traitement' ainsi que la liste des modules où il n'a pas encore de notes'
  - S'il existe plus de deux modules où le stagiaire a obtenu une note <3 afficher 'Notes Eliminatoires' et afficher les modules concernés
  - Sinon afficher la liste des modules (Module + Coefficients+note) ainsi que la moyenne du stagiaire

<pre> delimiter \$\$ CREATE PROCEDURE ps1() begin select s.* from stagiaire s where s.code_Stagiaire not in (select code_Stagiaire from notation);  end\$\$ </pre>	<p>Call ps1() ;</p>
<pre> DELIMITER \$\$ create PROCEDURE ps2() begin select f.Nom_Filiere, count(p.code_Module) from filiere f , programme p where f.code_filiere=p.code_filiere group by f.Nom_Filiere having count(p.code_Module)&gt;10;  end\$\$ </pre>	<p>Call ps2()</p>

<pre> DELIMITER \$\$ create procedure ps3(s int) begin select m.Nom_Module,count(p.code_filiere) from filiere f ,programme p , module m where f.code_filiere=p.code_filiere and p.code_Module=m.code_Module and f.code_secteur=s  group by m.Nom_Module HAVING count(p.code_filiere)= (select count(f.code_filiere)FROM filiere f where f.code_secteur=s); end\$\$ </pre>	Call ps3(1)
<pre> select m.code_Module,m.Nom_Module,n.Note,p.Coefficient, n.Note*p.Coefficient from programme p , module m , notation n, stagiaire s where p.code_Module=m.code_Module and n.code_Module =m.code_Module and s.code_Stagiaire=n.code_Stagiaire and p.code_filiere=s.code_filiere and s.code_Stagiaire=st; </pre>	

```

CREATE PROCEDURE ps44(st int)

begin declare m FFloat;

declare rem varchar(20);

select m.code_Module,m.Nom_Module,n.Note,p.Coefficient, n.Note*p.Coefficient
from programme p , module m , notation n, stagiaire s

where p.code_Module=m.code_Module and n.code_Module =m.code_Module

and s.code_Stagiaire=n.code_Stagiaire and p.code_filiere=s.code_filiere

and s.code_Stagiaire=st;

select sum(n.Note*p.Coefficient )/sum(p.Coefficient) into m

from programme p , module m , notation n, stagiaire s

where p.code_Module=m.code_Module and n.code_Module =m.code_Module

and s.code_Stagiaire=n.code_Stagiaire and p.code_filiere=s.code_filiere

and s.code_Stagiaire=st;

if m<10 THEN

set rem='redoublant';

ELSE set rem='reussi';

end if;

```

```
select concat(m , ' ',rem) as 'moyenne Decicion';  
end;
```

Un parameter peut être In et Out en même temps

Retourner le nombre de stagiaires d'une filière donnée

```
Create procedure ps0(In cf int, out Nbrst int)
```

```
Begin
```

```
Select count(*) into Nbrst  
from stagiaire s where s.code_filiere= cf;  
end
```

```
Create procedure ps00(Inout x int)
```

```
Begin
```

```
DECLARE nbr int; Select count(*) into nbr  
from stagiaire s  
where s.code_filiere= x; set x=nbr;  
end;
```

```
pour appeler cette procedure:
```

```
set @f=1;  
call ps00(@f);  
select @f as 'nombre de stagiaires';  
$$
```

Création de fonction