

Correction exam BOO 2018

1-

1.1 SQL3

- create type t_ecole as object(
 - list_classe t_list_classes --l'école est composé de plusieurs classe -> table imbriquée des références des objets t_class
);

 create or replace type t_list_classes as table of REF t_class ;
- create type t_class as object(
 - numero number(2),
 - nom varchar2(25),
 - ecole REF t_ecole,
 - list_eleve t_list_eleves, --le nombre des eleves dans une classe est quelconque -> table imbriquée des références des objets de type t_eleve
 - list_cours t_list_cours --chaque classe suit un nombre de cours max 15 cours -> varray(15) des ref des objets de type t_cours
);

 create or replace type t_list_eleves as table of REF t_eleve;
 create or replace type t_list_cours as varray(15) of ref t_cours;
- create type t_eleve as object(
 - numero number(2),
 - nom varchar2(25),
 - prenom varchar2(25),
 - classe REF t_class --un eleve n'appartient qu'à une et une seule classe -> ref vers un objet de type t_class
);
- create type t_cours as object(
 - numero number(2),
 - intit varchar(25),
 - vh number(2),
 - list_classes t_list_classes, --un cours peut être suivie par plusieurs/aucune classes -> table imbriquée des ref des objets de type t_class
 - list_prof t_list_prof --un cours peut être assuré par plusieurs prof max 3, -> varray des ref des objets de type t_prof
);

 create or replace table t_list_classes as table of ref t_class
 create or replace type t_list_prof as varray(3) of ref t_prof;

- create or replace type `t_prof` as object(
 - numero number(2),
 - nom varchar2(25),
 - prenom varchar2(25),
 - adresse varchar(75),
 - list_cours t_list_prof_cours --chaque prof assure au max 5 cours -> varray(5) des ref des objets de type t_cours
-);
- create or replace type t_list_prof_cours as varray(5) of ref t_cours;

1.2 Objet pur

```
Public class Ecole{
    List<Classe> classes = new ArrayList<Classe>() ; // le new je pense que c'est pas obligatoire
}
```

```
Public class Classe{
    Int numero ;
    String nom ;
    Ecole ecole ;
    List<Eleve> eleves = new ArrayList<Eleve>() ;
    Cours[] cours = new Cours[15] ;
}
```

```
Public class Eleve{
    Int numero ;
    String nom ;
    String prenom ;
    Classe classe ;
}
```

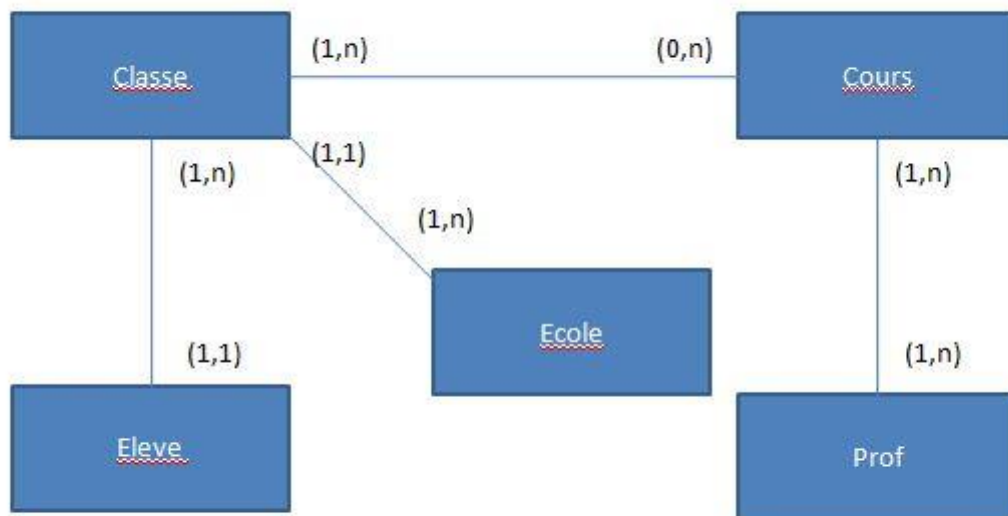
```
Public class Cours{
    Int numero ;
    String intit ;
    Int vh ;
    List<Classe> classes = new ArrayList<Classe>() ;
    Prof[] prof = new prof[3] ;
}
```

```

Public class Prof{
    Int numero ;
    String nom ;
    String prenom ;
    String adresse ;
    Cours[] cours = new Cours[5] ;
}

```

1.3 SQL2



2- SQL3

Premièrement on redéclare le type t_class comme suivant :

- create type `t_class` as object(
 - numero number(2),
 - nom varchar2(25),
 - ecole REF t_ecole,
 - list_eleve `t_list_eleves`,
 - list_cours `t_list_cours` ,
 member function calculer_vh return number ; // nouveaux
);

Puis on définit le corps du type avec implémentation de la méthode calculer_vh

Sol1 :

Create or replace type body t_class as

Member function calculer_vh return number is
 count number ; i number ; total number ;

Begin

total := 0 ; i :=1 ;

count := SELF.cours.COUNT()-- retourne le nombre des éléments dans la liste

while i <= count loop

total := total + cours(i).vh ; -- on somme

i := i + 1 ;

end loop ;

return total ;

end calculer_vh ;

Sol2 : si on suppose qu'une table des classes est déjà créée sous le nom **tab_classes**

Create or replace type body t_class as

Member function calculer_vh return number is

cours t_list_cours;

count number ; i number ; total number ;

Begin

Select list_cours into cours from **tab_classes** where numero = SELF.numero ;

If cours is null then return 0 ;

else

total := 0 ; i :=1 ;

count := cours.COUNT();

while i <= count loop

total := total + cours(i).vh ;

i := i + 1 ;

end loop;

return total ;

end if ;

end calculer_vh ;

DB40 (je suis pas sur, je l'ai pas fait dans l'examen)

Public class Classe{

Int numero ;

String nom ;

Ecole ecole ;

List<Eleve> eleves = new ArrayList<Eleve>() ;

Cours[] cours = new Cours[15] ;

```

    }
    Public int getVh(){
        Int total = 0 ;
        for(Cours c : cours) total = total + c.vh ;
        return total ;
    }
}

```

SQL2 :

La méthode sera une simple requête jointure entre la table classe_cours (intermediare) et chacune des autres tables classe et cours.

2-

Comparaison

	Objet pur ^{DB4O}	Relationnel étendu ^{SQL3}	Relationnel (Pur) ^{SQL2}
Capture de la sémantique	✓	✓	✗
1ere forme normale	✗	✗	✓
Types personnalisés	✓	✓	✗
Navigation	Identité	Reference	Jointure
	Objet	ADT	
	Identité	Reference (pinteur)	
	Collections	VARRAY & Nested Tables	
	heritage	heritage de type/table	