



Bases de Données Avancées

Révision SQL3

- USTHB Master 01 IL-
M. AZZOUZ

Dernière mise à jour : Juin 2020

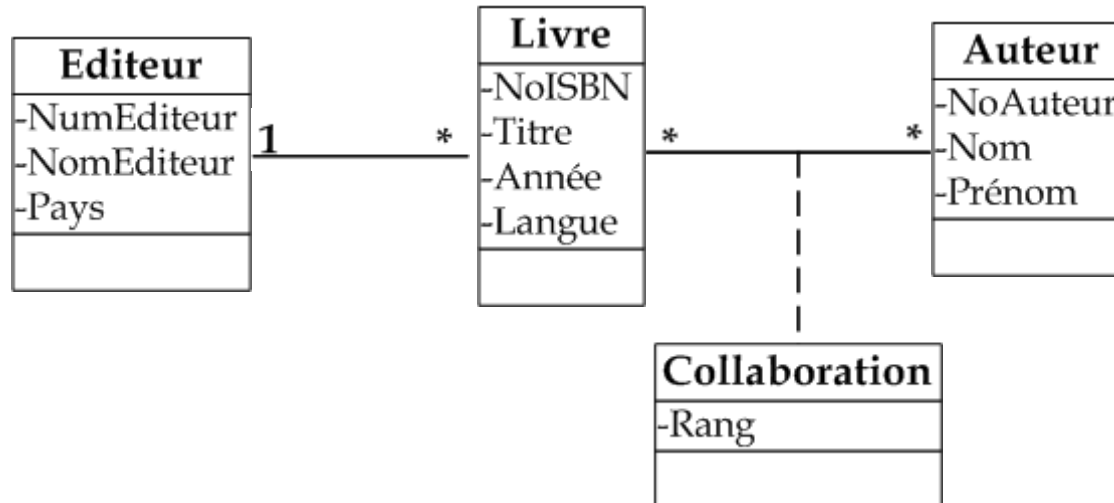
Exercice 04

❑ Soit la base de données relationnelle suivante, modélisant les livres (les clés des relations sont soulignées):

Relation	Description
LIVRE (<u>NoISBN</u> , Titre, Année, Langue, NumEditeur)	NoISBN: numéro du livre Titre : titre du livre Année : année de publication du livre Langue : langue de publication du livre : français, anglais ou espagnol NumEditeur : numéro de la maison d'édition où a été publié ce livre
AUTEUR (<u>NoAuteur</u> , Nom, Prénom)	NoAuteur: numéro de l'auteur Nom : nom de l'auteur Prénom : premier prénom de l'auteur
COLLABORATION (<u>NoISBN</u> , <u>NoAuteur</u> , Rang)	NoISBN: numéro du livre NoAuteur: numéro de l'auteur Rang : rang de l'auteur. Cette relation décrit les collaborations des auteurs à la rédaction d'un livre ainsi que le rang de l'auteur dans la liste des auteurs du livre (1er auteur, 2ème auteur, etc...).
EDITEUR (<u>NumEditeur</u> , NomEdit, Pays)	NumEditeur: numéro de l'éditeur NomEdit : nom de l'éditeur Pays : pays de l'éditeur

Exercice 04

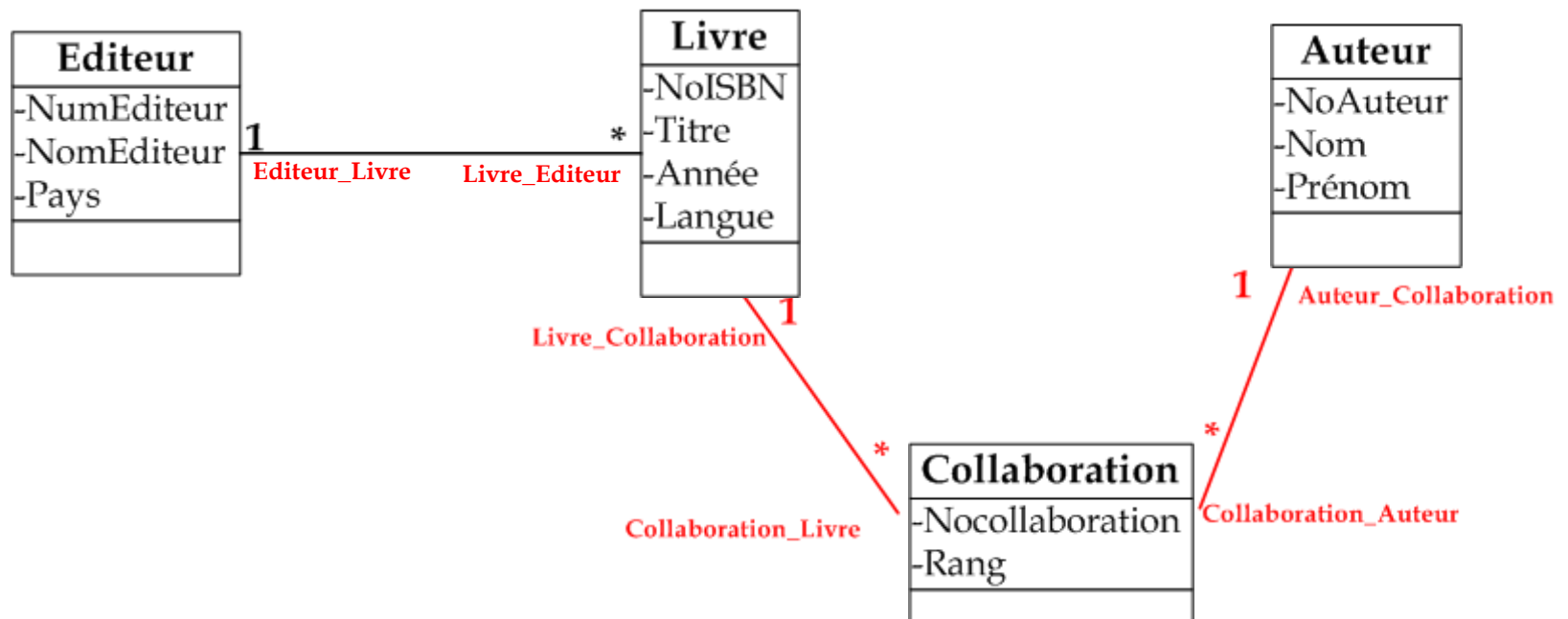
1. Transformer ce schéma relationnel en un diagramme de classes.



Exercice 04

2. On souhaite stocker ce diagramme de classes dans un SGBD relationnel objet en utilisant le langage de définition de données SQL3 :

➤ Transformation de diagramme de classes (ajout des rôles ainsi que la transformation des associations à classe d'association



Exercice 04

2. On souhaite stocker ce diagramme de classes dans un SGBD relationnel objet en utilisant le langage de définition de données SQL3 :
 - a. Définir tous les types nécessaires. Prendre en compte toutes associations qui existent.

➤ créer les types de manière incomplète

```
create type tlivre;  
/  
create type tauteur;  
/  
create type tediteur;  
/  
create type tcollaboration;  
/
```

➤ créer les types nécessaires aux associations « les tables imbriquées des références »

```
create type tset_ref_collaboration as  
table of ref tcollaboration;  
/  
create type tset_ref_livre as table of  
ref tlivre;  
/
```

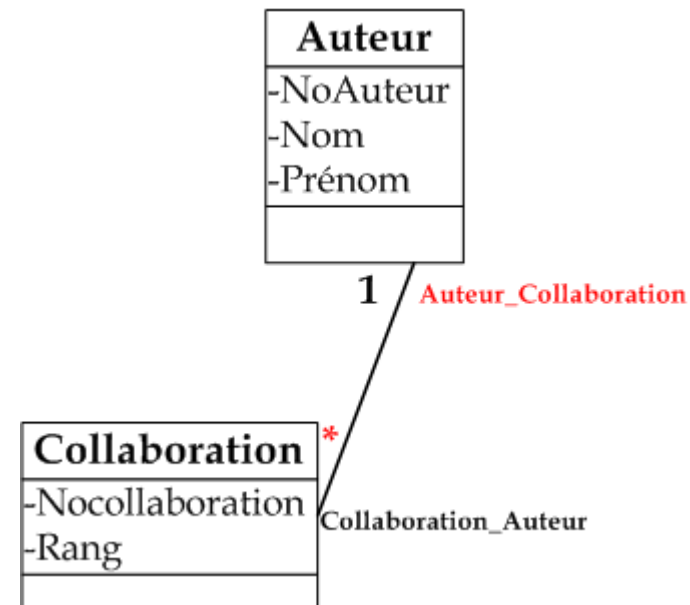
Exercice 04

2. On souhaite stocker ce diagramme de classes dans un SGBD relationnel objet en utilisant le langage de définition de données SQL3 :
 - a. Définir tous les types nécessaires. Prendre en compte toutes associations qui existent.

➤ Type tauteur

```
create or replace type tauteur as  
object(Noauteur integer, nom varchar(20),  
prenom varchar(20),
```

```
auteur_collaboration tset_ref_collaboration  
);  
/
```



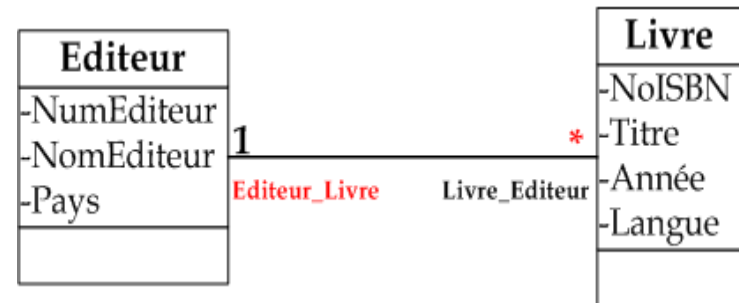
Exercice 04

2. On souhaite stocker ce diagramme de classes dans un SGBD relationnel objet en utilisant le langage de définition de données SQL3 :
 - a. Définir tous les types nécessaires. Prendre en compte toutes associations qui existent.

➤ Type téditeur

```
create or replace type téditeur as  
object(NumEditeur integer,  
nomEditeur varchar(50), Pays  
varchar(30),
```

```
éditeur_livre tset_ref_livre);  
/
```

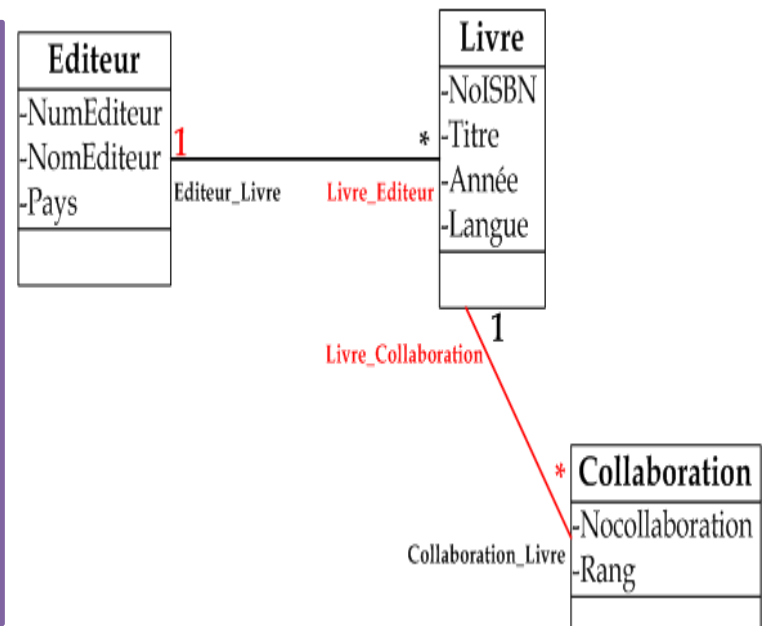


Exercice 04

2. On souhaite stocker ce diagramme de classes dans un SGBD relationnel objet en utilisant le langage de définition de données SQL3 :
 - a. Définir tous les types nécessaires. Prendre en compte toutes associations qui existent.

➤ Type tlivre

```
create or replace type tlivre as
object(NoISBN varchar(30),
titre varchar(100), Annee integer,
Langue varchar(20),
livre_editeur Ref teditteur,
livre_collaboration tset_ref_collaboration);
/
```



Exercice 04

b. Définir les tables nécessaires à votre base de données

```
➤ create table auteur of tauteur(constraint pk_auteur primary  
key(Noauteur))  
nested table auteur_collaboration store as tab_auteur_collaboration;  
  
➤ create table editeur of tEditeur(constraint pk_editeur primary  
key(NumEditeur))  
nested table editeur_livre store as table_editeur_livre;  
  
➤ create table livre of tlivre(constraint pk_livre primary  
key(NoISBN), constraint ck_langue check (langue in ('français',  
'anglais', 'espagnol')), constraint fk_editeur foreign key(livre_editeur)  
references editeur)  
nested table livre_collaboration store as table_livre_collaboration;  
  
➤ create table collaboration of tcollaboration(  
constraint pk_collaboration primary key(Nocollaboration), constraint  
fk_livre foreign key(collaboration_livre) references livre, constraint  
fk_auteur foreign key(collaboration_auteur) references auteur);
```

Exercice 04

3. La base de données contient les auteurs: David Fayon et Michaël Tartar et l'éditeur Pearson. Ecrire l'instruction SQL3 permettant d'insérer le livre de titre « Transformation digitale 2.0 », ISBN : 978-2-7440-6709-9 et édité par l'éditeur Pearson en juin 2019. Ce livre est écrit par David Fayon premier auteur et Michaël Tartar.

❑ Les insertions des données existantes(n'est pas demandée)

➤ Les deux auteurs:

```
insert into auteur values (tauteur('1', 'Fayon',  
'David',tset_ref_collaboration()));  
insert into auteur values (tauteur('2', 'Tartar',  
'Michaël',tset_ref_collaboration()));
```

➤ L'éditeur

```
insert into editeur values(tediteur('1',  
'Pearson',",tset_ref_livre()));
```

Exercice 04

❑ Insertion dans la table livre

```
insert into livre values (tlivre(  
'978-2-7440-6709-9',  
'Transformation digitale 2.0' ,  
2019, 'anglais',
```

```
(select ref (e) from editeur e  
where nomEditeur='Pearson'),
```

```
tset_ref_collaboration()  
));
```

❑ Insertion dans la table collaboration

```
➤ insert into collaboration  
values(tcollaboration('1',  
(select ref(l) from livre l where  
NoISBN='978-2-7440-6709-9'),  
(select ref(a) from Auteur a where  
nom='Fayon' and prenom='David'),  
1));
```

```
➤ insert into collaboration  
values(tcollaboration('2',  
(select ref(l) from livre l where  
NoISBN='978-2-7440-6709-9'),  
(select ref(a) from Auteur a where  
nom='Tartar' and prenom='Michaël') ,  
2));
```

Exercice 04

❑ Insertion dans la table imbriquée «livre_collaboration » de livre NoISBN='978-2-7440-6709-9'

➤ insert into table (select l.livre_collaboration from livre l where NoISBN='978-2-7440-6709-9')

(select ref(c) from collaboration c where Nocollaboration=1);

➤ insert into table (select l.livre_collaboration from livre l where NoISBN='978-2-7440-6709-9')

(select ref(c) from collaboration c where Nocollaboration=2);

❑ Insertion dans la table imbriquée «auteur_collaboration » des auteurs Fayon David et Tartar Michaël

➤ insert into table (select a.auteur_collaboration from auteur a where nom='Fayon' and prenom='David')

(select ref(c) from collaboration c where Nocollaboration=1);

➤ insert into table (select a.auteur_collaboration from auteur a where nom='Tartar' and prenom='Michaël')

(select ref(c) from collaboration c where Nocollaboration=2);

Exercice 04

3. On souhaite ajouter dans la base de données une liste de numéros de téléphone pour chaque auteur. Chaque numéro de téléphone a un libellé indiquant la nature du numéro (mobile, bureau, maison). Ecrire les requêtes permettant de stocker ces informations dans la base de données.

```
➤ create type tTel as object (num varchar(15),  
label varchar(10));
```

```
/
```

```
➤ create type tListTel as table of tTel;
```

```
/
```

```
➤ Alter type tauteur add attribute telephone tListTel  
cascade;
```

Exercice 04

3. On complète le type `auteur` avec la méthode *MesLivres* retournant l'ensemble (sans doublons) des livres où l'auteur est de rang égale à 1. Donner la signature et le corps de cette méthode.

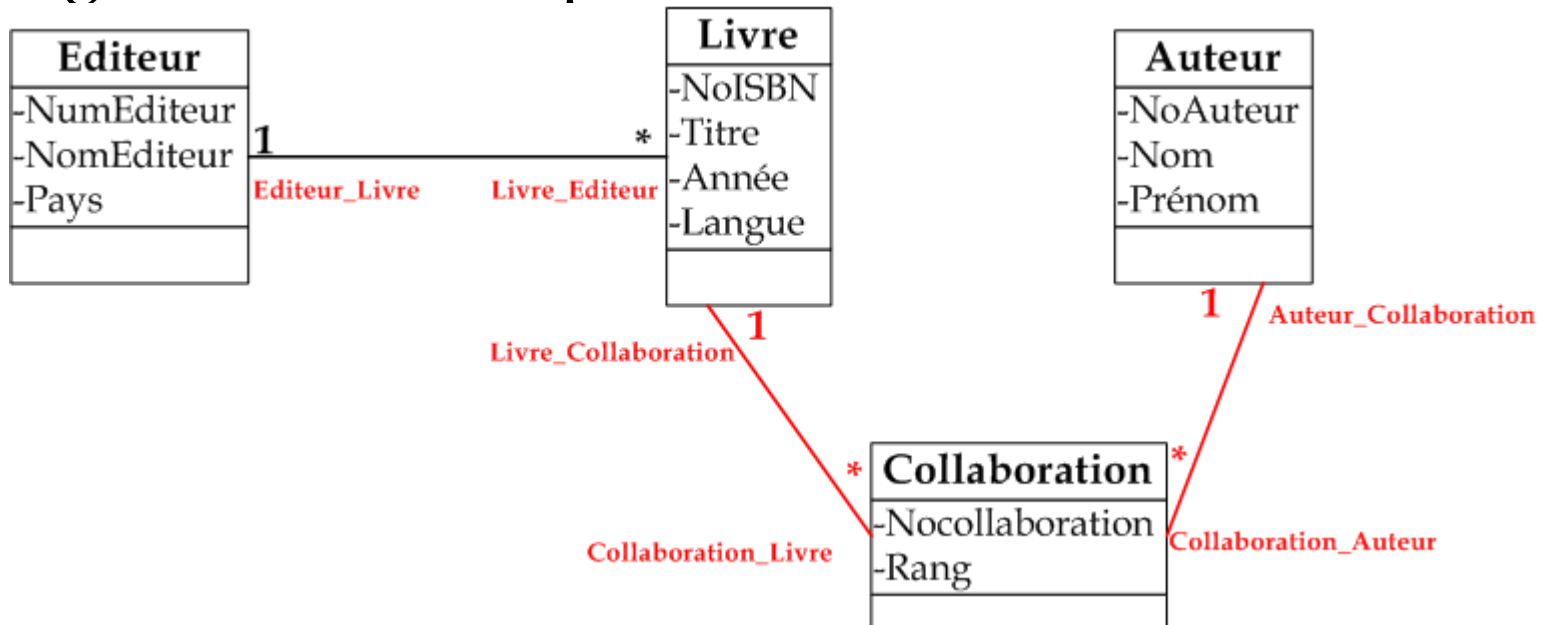
➤ create type `tset_livre` as table of tlivre;

/

➤ Alter type `tauteur` add member function *MesLivres* return `tset_livre` cascade;

Exercice 04

3. On complète le type auteur avec la méthode *MesLivres* retournant l'ensemble (sans doublons) des livres où l'auteur est de rang égale à 1. Donner la signature et le corps de cette méthode.



Exercice 04

➤ *Corps de la méthode MesLivres*

create or replace type body tAuteur

as member function MesLivres return tset_livre

is

livresauteurs tset_livre;

Begin

Select CAST (MULTISET(

select deref(deref(value(c)).collaboration_livre)

from auteur a, table(a.auteur_collaboration) c

where deref(value(c)).rang=1

and a.Noauteur=self.Noauteur) as tset_livre)

into livresauteurs

from dual;

return livresauteurs;

end;

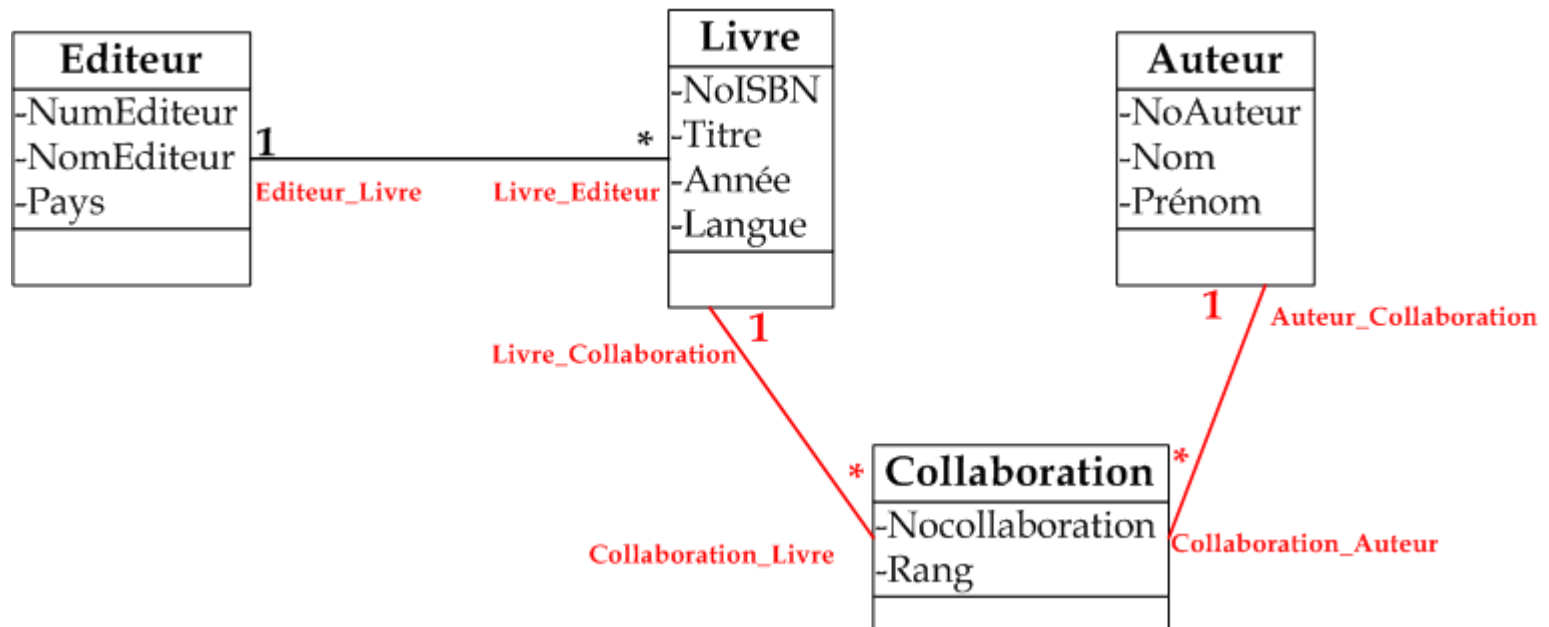
End;

/

Exercice 04

6. Ecrire en SQL3 les requêtes suivantes :

a. Quels sont les auteurs qui ont participé dans l'écriture de même livre. Afficher des couples formés de numéro de livre et des objets auteurs.



Exercice 04

6. Ecrire en SQL3 les requêtes suivantes :

a. Quels sont les auteurs qui ont participé dans l'écriture de même livre. Afficher des couples formés de numéro de livre et des objets auteurs.

```
Select deref(value(c1).collaboration_livre). NoISBN,  
value(a1), value(a2)
```

```
From auteur a1, table(a1.auteur_collaboration) c1,  
auteur a2, table(a2.auteur_collaboration) c2
```

```
Where a1.Noauteur > a2.Noauteur
```

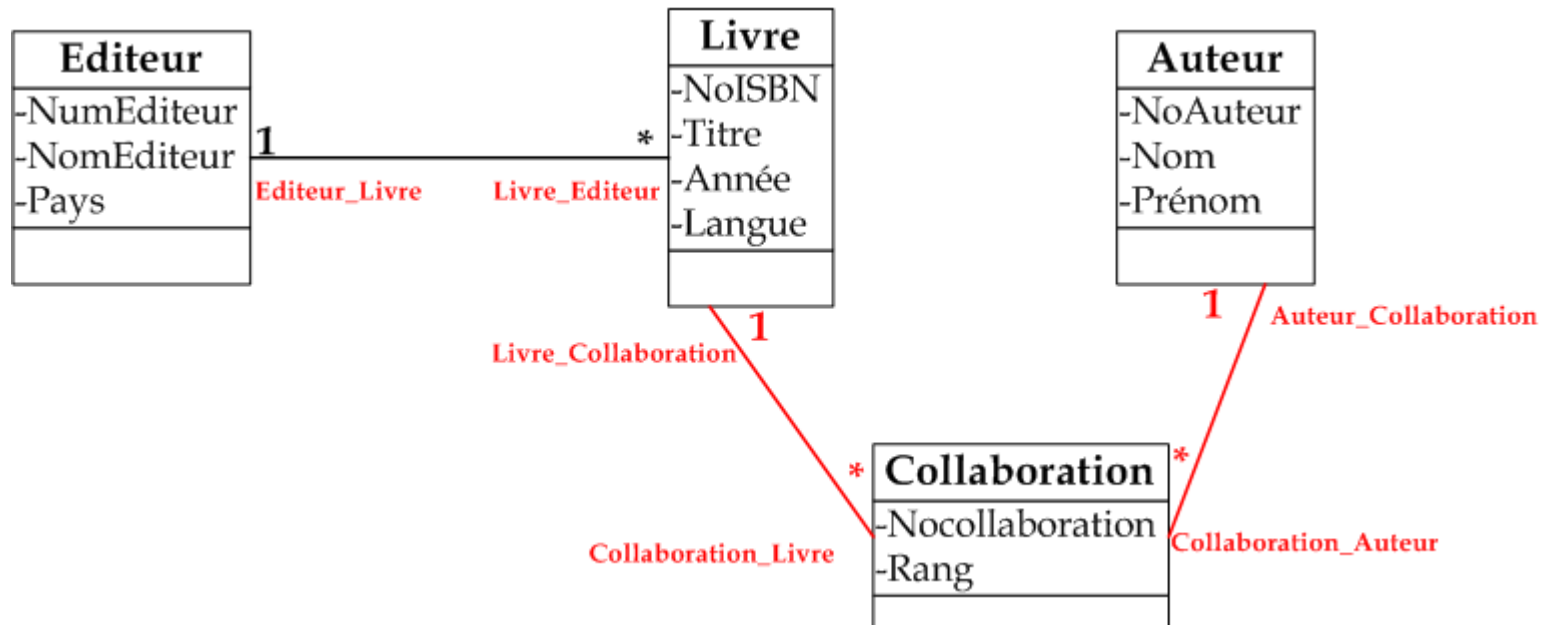
```
And
```

```
value(c1).collaboration_livre = value(c2).collaboration_livre;
```

Exercice 04

6. Ecrire en SQL3 les requêtes suivantes :

b. Quels sont les auteurs qui éditent chez tous les éditeurs.



Exercice 04

6. Ecrire en SQL3 les requêtes suivantes :

b. Quels sont les auteurs qui éditent chez tous les éditeurs.

```
select value(a)
```

```
from auteur a, table(a.auteur_collaboration) c
```

```
having
```

```
count(distinct(deref(value(c).collaboration_livre).livre_editeur))
```

```
=
```

```
(select count(*) from editeur)
```

```
group by a.Noauteur;
```