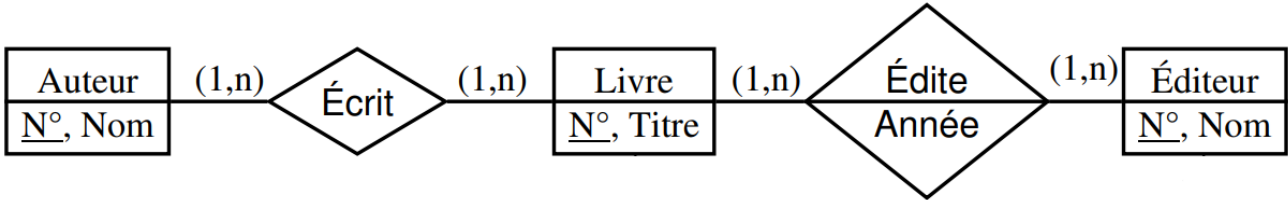


Corrigé type examen

Question de cours : voir les définitions dans le support de cours.

Exercice 1: Modélisation (4 pts)

1) Modèle E/A correspondant: (1,5 pt)



Modèle relationnel déduit: (1,5 pts)

Table "Livre" : ISBN (clé primaire), Titre.

Relation: Livre(ISBN,Titre)

Table "Auteur" : Numéro d'auteur (clé primaire), Nom de l'auteur

Relation: Auteur(Numéro-auteur,Nom-auteur)

Table "Éditeur" : Numéro d'éditeur (clé primaire), Nom de l'éditeur

Relation: Éditeur(Numéro-éditeur,Nom-éditeur)

Table "Écrit par" : ISBN (clé étrangère référençant la table "Livre"), Numéro d'auteur (clé étrangère référençant la table "Auteur")

Relation: Écrit par(#ISBN,#Numéro-auteur)

Table "Édité par" : ISBN (clé étrangère référençant la table "Livre"), Numéro d'éditeur (clé étrangère référençant la table "Éditeur")

Relation: Édité par(#ISBN,#Numéro-éditeur)

Les tables "Écrit par" et "Édité par" sont des tables de jonction qui permettent de représenter la relation plusieurs-à-plusieurs entre les livres, les auteurs et les éditeurs.

2. La requête SQL de création de la table LIVRE (1 pt)

```
create table LIVRE (  
ISBN(20) not null,  
Titre varchar(50)not null,  
primary key (ISBN)  
);
```

Exercice 2: Algèbre relationnel et SQL (10pts)

Algèbre relationnel :

1. Le nom des auteurs.

$\pi(\text{NomAut})(\text{Auteurs})$

2. Donner le numéro des auteurs qui éditent tous les livres qu'ils écrivent.

$\pi_{\text{NumAut}}(\text{Ecrit}) - \pi_{\text{NumAut}}(\text{Ecrit} - \pi_{\text{NumEdit, NumLivre}}(\text{livres}))$

3. Quels sont les éditeurs (numéro et nom) qui n'ont édité aucun livre de l'auteur Numéro 120.

$\pi_{\text{NumEdit, NomEdit}}(\text{Editeur} \bowtie (\pi_{\text{NumEdit}}(\text{Editeur}) - \pi_{\text{NumEdit}}(\text{Livres} \bowtie \sigma_{\text{NumAut}=120}(\text{Ecrit}))))$

SQL:

1. Sélectionner tous les auteurs.

```
SELECT * FROM Auteurs;
```

2. Les livres publiés entre 2020 et 2023.

```
SELECT *  
FROM Livres  
WHERE AnnéeEdition BETWEEN 2020 AND 2023;
```

3. Les auteurs ayant écrit au moins un livre.

```
SELECT *  
FROM Auteurs  
WHERE NumAut IN (SELECT DISTINCT NumAut FROM Ecrit);
```

4. Donner le titre des livres écrits par des auteur qui habitent à Alger par ordre croissant des années.

```
SELECT Livres.Titre  
FROM Livres  
JOIN Ecrit ON Livres.NumLivre = Ecrit.NumLivre  
JOIN Auteurs ON Ecrit.NumAut = Auteurs.NumAut  
WHERE Auteurs.AdrAut = 'Alger'  
ORDER BY Livres.AnnéeEdition ASC;
```

Remarque: ici vous pouvez utiliser la jointure comme vu au TDs ou utilisant JOIN.

5. La liste des maisons d'édition avec lesquelles l'auteur numéro 10 a travaillé.

```
SELECT Éditeur.NumEdition, Éditeur.NomEdition  
FROM Éditeur  
JOIN Livres ON Éditeur.NumEdition = Livres.NumEdit  
JOIN Ecrit ON Livres.NumLivre = Ecrit.NumLivre  
WHERE Ecrit.NumAut = 10;
```

6. Calculer la quantité totale des livres écrits par chaque auteur qui a écrit plus d'un livre, trié par ordre alphabétique des noms d'auteur.

```
SELECT NumAut, NomAut, COUNT(NumLivre)
FROM Écrit
INNER JOIN Auteurs ON Écrit.NumAut = Auteurs.NumAut
GROUP BY NumAut, NomAut
HAVING COUNT(NumLivre) > 1
ORDER BY NomAut ASC;
```

7. Quels sont les numéros des auteurs dont les livres n'ont été édités que dans la maison d'édition de nom SCIENCE ET SAVOIR.

Solution 1 (avec MINUS)

```
(SELECT NumAut FROM Écrit x, Livre y, Editeur z
WHERE x.Numlivre=y.NumLivre AND y.NumEdit=z.NumEdit
AND z.NomEdit ='SCIENCE ET SAVOIR')
```

MINUS

```
(SELECT NumAut FROM écrit x, Livre y, Editeur z
WHERE x.Numlivre=y.NumLivre
AND y.NumEdit=z.NumEdit
AND z.NomEdit <>'SCIENCE ET SAVOIR');
```

Solution 2 (avec NOT EXISTS)

```
SELECT NumAut FROM Auteur a
WHERE NOT EXISTS (SELECT * FROM écrit x
WHERE x.NumAut= a.NumAut
AND NOT EXISTS (SELECT * FROM Livre y, Editeur z
WHERE x.NumLivre=y.numLivre
AND y.NumEdit=z.NumEdit
AND z.NomEdit ='SCIENCE ET SAVOIR'));
```

Exercice : Normalisation (4 pts)

Soit deux relations:

1) **Réservation**(NumR, NomClient, AdresseClient, CodeP, NomP, Prix)

Avec les dépendances fonctionnelles suivantes :

1. **NumR** → **NomClient, AdresseClient, CodeP**

2. **CodeP** → **NomP, Prix**

Identifier les clés candidates La relation "Réservation" a une clé candidate : **{NumR}** qui est une primaire.

Car:

NumR → NomClient, AdresseClient, CodeP et CodeP → NomP, Prix donc:

NumR → NomClient, AdresseClient, CodeP, NomP, Prix (par transitivité).

La relation est en 1FN car *tout attribut contient une valeur atomique*.

La relation est en 2FN elle est en 1FN et tout attribut n'appartenant pas à une clé ne dépend pas que d'une partie de cette clé (i.e, il n'y a pas une partie de la clé qui détermine un attribut non clé).

La relation n'est pas en 3FN à cause de la DF CodeP → NomP, Prix (un attribut non clé détermine des attributs non clé).

Rappelant que pour qu'une relation soit en 3FN, il faut qu'elle soit en 2FN et que tout attribut n'appartenant pas à une clé ne dépend pas d'un attribut non clé

La décomposition : Clients(NumR, NomClient, AdresseClient, CodeP)

Produits(CodeP, NomP, Prix)

2) C (NP, V, D, NE, A, NC, N)

Soit l'ensemble des dépendances fonctionnelles initiales suivant :

1. **NP** → **V**

2. **V** → **D**

3. **NP** → **D**

4. **NE** → **A**

5. **NE, NC** → **N**

6. **NC** → **NP**

{1, 2, 4, 5, 6} est une couverture minimale.

Une relation est en 1ère Forme Normale (1FN) si tout attribut contient une valeur atomique

Une relation est en 2FN si elle est en 1FN et tout attribut n'appartenant pas à une clé ne dépend pas que d'une partie de cette clé.

1 seule clé (**NE, NC**)

Les DF : 1. NP --> V 2. V --> D 3. NE --> A

4. NE, C--> N 5. C--> NP.

C (NE, NC, N)

R1 (NC, NP, V, D) }
R2 (NE, A) } sont en 2FN

Une relation est en troisième forme normale (3FN) si :

1. elle est en 2 FN

2. tout attribut n'appartenant pas à une clé ne dépend pas d'un attribut non clé

C (NE, NC, N)

R1 (NC, NP, V, D)

R2 (NE, A)

R1 n'est pas en 3 FN

car NP \rightarrow V et V \rightarrow D

R1 (NC, NP)

R3 (NP, V, D)

R3 n'est pas en 3 FN car V \rightarrow D

R3 (NP, V) R4 (V, D)