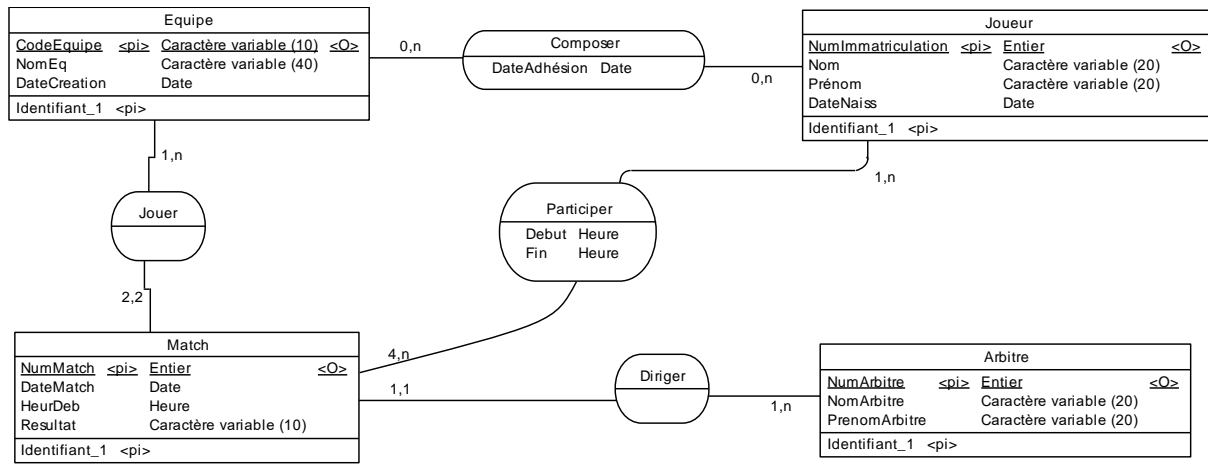




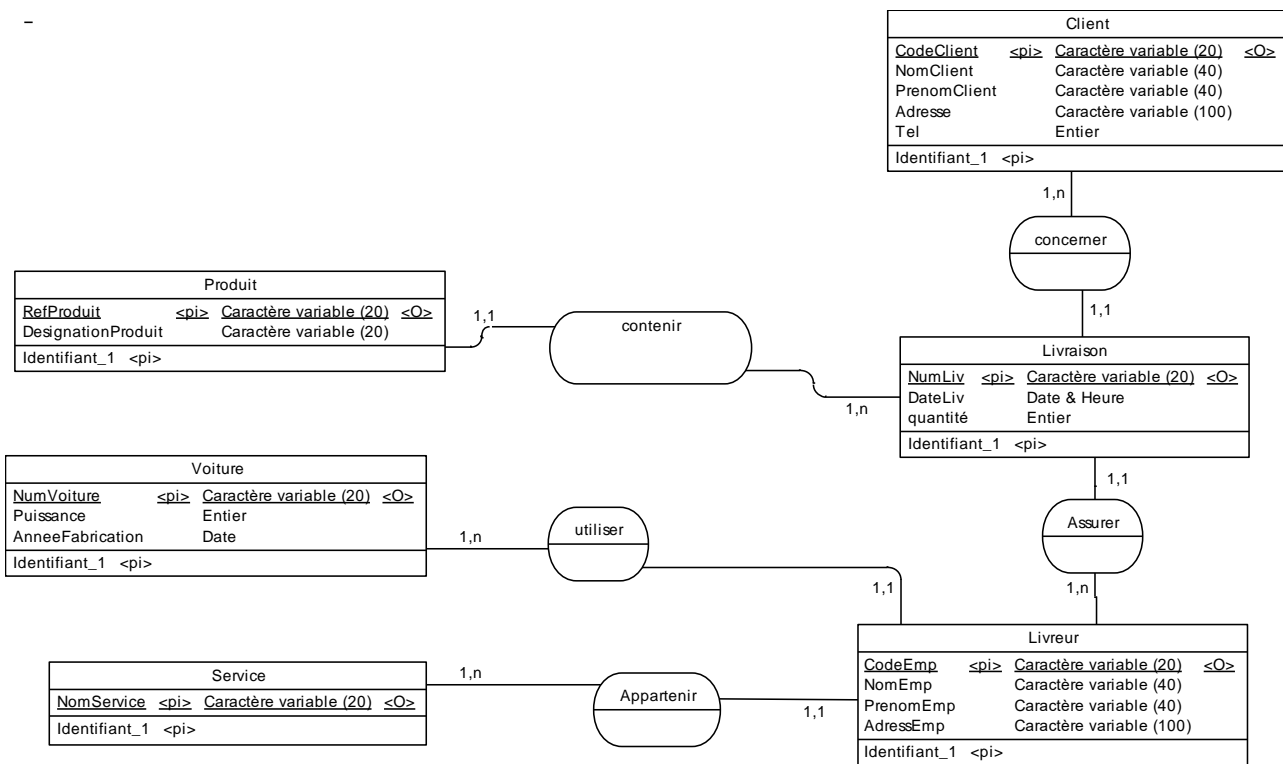
Correction de l'examen BD

Exercice 1 : Modélisation MCD (6 points)

A.



B.



Exercice 2 : Normalisation et Dépendances fonctionnelles (3 points)

Soit la relation suivante :

CarnetDeVoyage (numAuteur, nomAuteur, prenomAuteur, numVille, nomVille, nomPays, descriptionVoyage)

numAuteur \rightarrow nomAuteur

numAuteur \rightarrow prenomAuteur

numVille \rightarrow nomVille

numVille \rightarrow nomPays

numAuteur, numVille \rightarrow description

1. Une clé est un groupe d'attributs minimal qui détermine tous les attributs de la relation.

Il y a une unique clé (**numAuteur, numVille**).

numAuteur, numVille \rightarrow nomAuteur

numAuteur, numVille \rightarrow prenomAuteur

numAuteur, numVille \rightarrow nomVille

numAuteur, numVille \rightarrow nomPays

numAuteur, numVille \rightarrow description

La réflexivité et la transitivité sont utilisées à chaque fois de la même façon, par exemple, pour le premier :

numAuteur, numVille \rightarrow numAuteur

ET numAuteur \rightarrow numAuteur

DONC numAuteur, numVille \rightarrow numAuteur

2. La relation est en 1NF, on a identifié une clé et les attributs sont atomiques. Elle n'est pas en 2NF car des attributs faisant partie de la clé déterminent d'autres attributs, par exemple :
numAuteur \rightarrow nomAuteur.

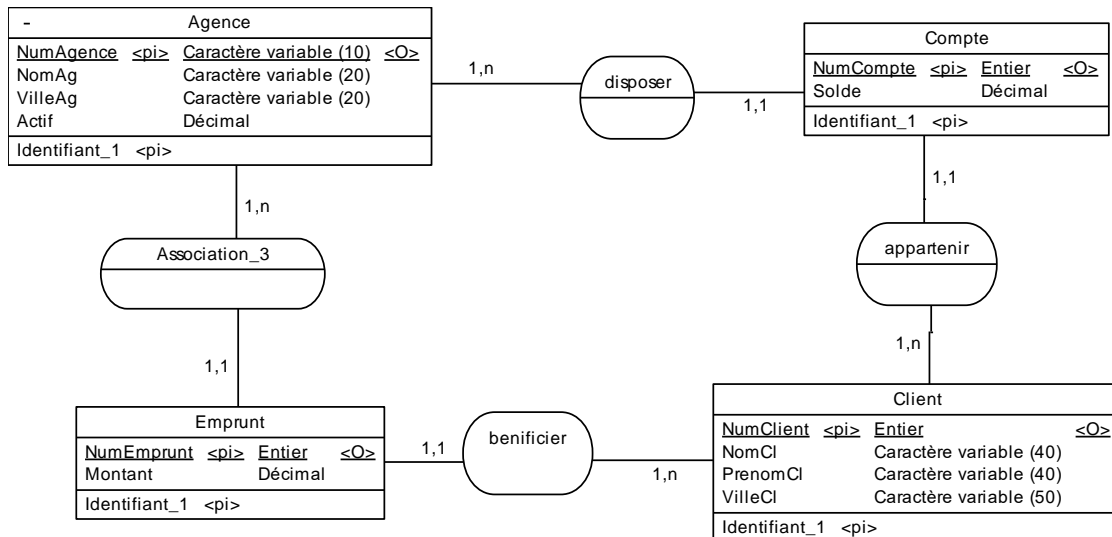
3. Auteur(#numAuteur, nomAuteur, prenomAuteur)
Ville(#numVille, nomVille, nomPays)
CarnetDeVoyage(#numAuteur, #numVille, description)

Exercice 3 : Algèbre relationnelle (4 points)

Soit la base de données BANQUE contenant les tables suivantes :

- AGENCE (Num_Agence, Nom, Ville, Actif)
- CLIENT (Num_Client, Nom, Prenom, Ville)
- COMPTE (Num_Compte, #Num_Agence, #Num_Client, Solde)
- EMPRUNT (Num_Emprunt, #Num_Agence, #Num_Client, Montant)

- 1.



2. Les clients résidant à Kénitra, avec un compte dont le solde est supérieur à 10 000 et un emprunt dont le montant est inférieur à 100 000.

$$\pi_{\text{Num_Client}}(\sigma_{\text{ville}='Kénitra'}(\text{Client}) \bowtie \sigma_{\text{solde}>1000}(\text{Compte}) \bowtie \sigma_{\text{montant}<100000}(\text{Emprunt}))$$

3. Les clients n'ayant contracté aucun emprunt.

$$\pi_{\text{Num_Client}}(\text{Client}) - \pi_{\text{Num_Client}}(\text{Emprunt})$$

4. Les clients ayant un compte dans la même agence que Mohammed Idrissi.

$$\pi_{\text{Num_Client}}(\pi_{\text{Num_Agence}}(\sigma_{\text{Prenom}='Mohammed' \text{ AND } \text{Nom}='Idrissi'}(\text{Client}) \bowtie \text{Compte}) \bowtie \text{Compte})$$

Exercice 4 : Langage SQL (7 points)

Soit la base de données intitulée "gestion_projet" permettant de gérer les projets relatifs au développement de logiciels suivante :

- Developpeur (NumDev, NomDev, ADRDev, EmailDev, TelDev)
- Projet (NumProj, TitreProj, DateDeb, DateFin)
- Logiciel (CodLog, NomLog, PrixLog, #NumProj)
- Realisation (#NumProj, #NumDev)

1. Créer les tables Projet et Logiciel :

```
CREATE TABLE Projet (
  NumProj INT UNSIGNED PRIMARY KEY AUTO_INCREMENT,
  TitreProj VARCHAR(40),
  DateDeb Date,
  DateFin Date,
)
ENGINE=InnoDB;
```

```
CREATE TABLE Logiciel (
  CodeLog INT UNSIGNED PRIMARY KEY AUTO_INCREMENT,
  NomLog VARCHAR(40),
  PrixLog DECIMAL,
  NumProj INT UNSIGNED,
  CONSTRAINT fk_projet_numero -- On donne un nom à notre clé
  FOREIGN KEY (NumProj) -- Colonne sur laquelle on crée la clé
  REFERENCES Projet(NumProj) -- Colonne de référence
```

)
ENGINE=InnoDB;

2. Afficher les noms et les prix des logiciels appartenant au projet ayant comme titre « gestion de stock », triés dans l'ordre décroissant des prix.

```
SELECT L.NomLog, L.PrixLog FROM Logiciel L INNER JOIN Projet P
ON L.NumProj=P.NumProj WHERE P.TitreProj="gestion de stock"
ORDER BY L.PrixLog DESC
```

3. Afficher le total des prix des logiciels du projet numéro 10. Lors de l'affichage, le titre de la colonne sera « coût total du projet ».

```
SELECT SUM(PrixLog) as "cout total du projet" FROM Logiciel WHERE NumProj=10
```

4. Afficher le nombre de développeurs qui ont participé au projet intitulé « gestion de stock »

```
SELECT count(*) FROM Developpeur D INNER JOIN Realisation R
ON D.NumDev=R.NumDev INNER JOIN Projet P ON P.NumProj=R.NumProj
WHERE P.TitreProj= "gestion de stock"
```

5. Afficher les projets qui ont plus que 5 logiciels

```
SELECT NumProj, TitreProj FROM PProjet P INNER JOIN Logiciel L ON P.NumProj=L.NumProj
GROUP BY NumProj, TitreProj
HAVING count(*)>5
```

6. Les numéros et noms des développeurs qui ont participé dans tous les projets.

```
SELECT NumDev, NomDev FROM Developpeur D INNER JOIN Realisation R ON
D.NumDev=R.NumDev
GROUP BY NumDev, NomDev
HAVING count(*)=(SELECT COUNT(*) FROM Projet)
```

7. UPDATE Projet
SET DateFin = '17/04/2022'
WHERE TitreProj like '%Web%' ;