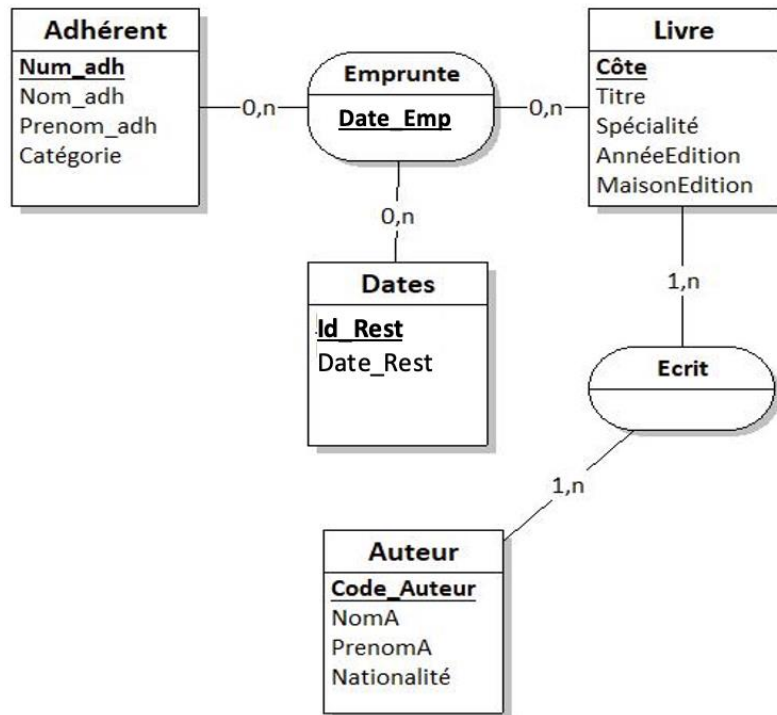


Corrigé TD#1 BDA - Rappels

Exercice 1

1. Modèle Entité/Association :



2. Après application des règles de transformation du modèle E/A vers le modèle relationnel, on obtient :

Adhérent (Num_adh, Nom_adh, Prenom_adh, Catégorie)
Livre (Côte, Titre, Spécialité, AnnéeEdition, MaisonEdition)
Auteur (Code_Auteur, NomA, PrenomA, Nationalité)
Ecrit (Code_Auteur*, Côte*)
Emprunte (Date_Emp, Num_Adh*, Côte*, Id_Rest*)
Dates (Id_date, Date_Rest)

Rappel des règles de transformation E/A → Relationnel :

1. Pour chaque entité du modèle, créer une relation portant le même nom. Les propriétés de l'entité deviennent les attributs de la relation et l'identifiant constitue la clé de la relation.
2. Pour chaque association entre deux entités A et B, de cardinalité 1 à plusieurs (à une occurrence de A correspond plusieurs occurrences de B, et à une occurrence de B correspond au plus une seule occurrence de A), intégrer la clé de la relation A dans la relation B (comme étant clé étrangère).
3. Pour chaque association entre deux entités A et B de cardinalité plusieurs à plusieurs, créer une relation portant le même nom dont la clé est constituée de la composition des clés de A et B. Si l'association est porteuse d'informations (comporte des propriétés), ces propriétés deviennent des attributs de la relation.
4. Pour chaque association n-aire entre N entités A_1, A_2, \dots, A_n , créer une relation portant le même nom, dont la clé est constituée de la composition des clés des relations A_1, A_2, \dots, A_n . Si l'association est porteuse d'informations (comporte des propriétés), ces propriétés deviennent des attributs de la relation.

Exercice 2

```
CREATE TABLE Adherent(  
    Num_adh Number(10) NOT NULL PRIMARY KEY,  
    Nom_adh char(25),  
    Prenom_adh char(25),  
    Categorie char(10)  
);  
  
CREATE TABLE Livre(  
    Cote char(10) NOT NULL PRIMARY KEY,  
    Titre char(200),  
    Specialite char(25),  
    AnneeEdition Number(4),  
    MaisonEdition char(25)  
);  
  
CREATE TABLE Auteur(  
    Code_Auteur char(10) NOT NULL PRIMARY KEY,  
    NomA char(25),  
    PrenomA char(25),  
    Nationalite char (25)  
);
```

```
CREATE TABLE Ecrit(
    Code_Auteur char(10),
    Cote char(10),
    FOREIGN KEY (Code_Auteur) REFERENCES Auteur(Code_Auteur),
    FOREIGN KEY (Cote) REFERENCES Livre(Cote),
    PRIMARY KEY (Code_Auteur, Cote)
);
```

```
CREATE TABLE Emprunte(
    Date_Emp Date,
    Num_adh Number(10),
    Cote char(10),
    Id_Rest Number(10),
    FOREIGN KEY (Num_adh) REFERENCES Adherent(Num_adh),
    FOREIGN KEY (Cote) REFERENCES Livre(Cote),
    FOREIGN KEY (Id_Rest) REFERENCES Dates(Id_Rest),
    PRIMARY KEY (Date_Emp, Num_adh, Cote, Id_Rest)
);
```

```
CREATE TABLE Dates(
    Id_Rest Number(10) NOT NULL PRIMARY KEY,
    Date_Rest Date
);
```

Exercice 3

1. SELECT E.Matricule, E.Nom, E.Prenom FROM Etudiant E
WHERE E.Groupe NOT IN (SELECT S.Groupe FROM Seance S
WHERE S.Jour_Semaine = 'Jeudi');
2. SELECT S.Groupe FROM Séance
WHERE S.Code_Ens IN (SELECT E.Code_Ens FROM Enseignant E
WHERE E.Grade = 'Maître Assistant');
3. SELECT S.code_Ens, count(*) FROM Séance S
GROUP BY S.Code_Ens;
4. SELECT E.Code_Ens, E.Nom FROM Enseignant E
WHERE NOT EXISTS (SELECT S.Code_Ens FROM Séance S
WHERE S.Enseignant.Code_Ens = S.Seance.Code_Ens
AND S.Code_Salle = 204) ;
5. SELECT E.Groupe FROM Etudiant GROUP BY E.Groupe
HAVING count(*)>=12;
6. SELECT S.Code_Salle FROM Salle S
WHERE S.Superficie IN (SELECT max(S1.Superficie) FROM Salle S1);

Exercice 4

1. SELECT Num_adh, nom, prenom FROM Adherent A
WHERE EXISTS (SELECT * FROM Emprunt E1, Livre L1

```

WHERE E1.Cote = L1.Cote AND E1.Num_adh = A.Num_adh
AND L1.MaisonEdition = 'Casbah Editions')
AND NOT EXISTS (SELECT * FROM Emprunt E2, Livre L2
WHERE E2.Cote = L2.Cote AND E1.Num_adh = A.Num_adh
AND L2.MaisonEdition <> 'Casbah Editions');

```

```

2. SELECT Num_adh, Nom_adh, Prenom_adh FROM Adherent A
WHERE NOT EXISTS (SELECT * FROM Emprunte E
WHERE E.Num_adh = A.Num_adh
AND EXISTS (SELECT * FROM Livre L
WHERE L.Cote = E.Cote
AND Specialite = 'Réseaux'));

```

3. **Les titres des livres pour lesquels on ne trouverait PAS d'adhérent qui ne les aient PAS empruntés**

```

SELECT Titre FROM Livre L
WHERE NOT EXISTS (SELECT * FROM Adherent A
WHERE NOT EXISTS (SELECT * FROM Emprunt E
WHERE E.Num_adh = A.Num_adh
AND E.Cote = L.Cote));

```

```

4. SELECT adherent.num_adh, adherent.nom_adh, adherent.prenom_adh
FROM Adherent, Emprunte WHERE adherent.num_adh NOT IN (SELECT
num_adh FROM Emprunte);

```

Exercice 5

1. π NomEtd, PrenomEtd, AdresseEtd (σ NomEtd = 'Alami' Etudiant)

3. π NomAuteur, AdresseAuteur (σ TitreLivre = 'comment avoir 20 en BDD' (Livre \bowtie Auteur))

4. π Livre.* (σ NomAuteur = 'Aalami' and NomEditeur='nulle part' (Livre \bowtie Editeur) \bowtie Auteur)

5. π Livre.* (σ NomAuteur = 'Aalami' ou NomAuteur = 'Belhadj' (Livre \bowtie Auteur))

6. π Livre.* ((π NumLivre Livre - π NumLivre Prêt) \bowtie Livre)