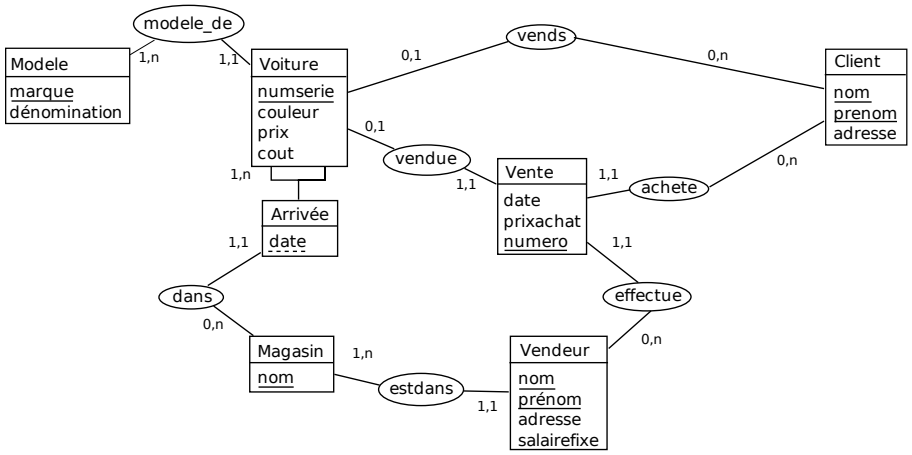
LIF4 - TD9
Requêtes SQL
Correction

Exercice 1:

On considère une entreprise de ventes de voitures. Un modèle de voiture est décrit par une marque, une dénomination. Une voiture est identifiée par un numéro de série, et a un modèle, une couleur et un prix affiché et un coût (prix auquel la voiture est revenue). Des clients, on connaît le nom, le prénom et l'adresse. Parmi les clients, on trouve les anciens propriétaires des voitures d'occasion, ainsi que les personnes ayant acheté une voiture au magasin. Lorsqu'une vente est réalisée, on en connaît le vendeur (dont on connaît le nom, le prénom, l'adresse et le salaire fixe) et le prix d'achat réel (en tenant compte d'un rabais éventuel). Chaque vendeur touche une prime de 5% de la différence entre le prix d'achat affiché et le coût de la voiture. L'entreprise est répartie sur un certain nombre de magasins et chaque vendeur opère dans un magasin unique. Chaque voiture est, ou à été, stockée dans certains magasins et est vendue dans le dernier magasin où elle a été stockée. On garde trace des dates d'arrivée dans et de départ des magasins. Un transfert de voiture entre deux magasins se fait dans la journée.

1. Donner un diagramme Entité/Association pour représenter ces données

   Correction:



2. Donner un schéma de base de données correspondant à ce diagramme

   Correction:

```
Voiture(numserie, couleur, prix, cout, marque, modele,nomp?,prenomp?)
Arrivee(numserie,date,magasin)
Client(nom,prenom,adresse)
Vendeur(nom,prenom,adresse,salairefixe,magasin)
Vente(numero,date,prixachat,numserie,noma,prenoma,nomv,prenomv)
```

3. Écrire les requêtes suivantes en SQL:

(a) Donner la liste des voitures (numéro) vendues après le 15 avril 2007.
Correction:

```
SELECT Voiture.numserie
FROM Voiture, Vente
WHERE Voiture.numserie = Vente.numserie
  AND date > '2007-04-15'
```

(b) Donner la voiture qui rapporté le plus d'argent.
Correction:

```
SELECT Voiture.numserie
FROM Voiture,Vente
WHERE Voiture.numserie = Vente.numserie
  AND prixachat-cout >= ALL (SELECT prixachat-cout
                             FROM Voiture,Vente
                             WHERE Voiture.numserie = Vente.numserie)
```

(c) Donner le vendeur ayant accordé le plus gros rabais.
Correction:

```
SELECT nomv, prenomv
FROM Vente,Voiture
WHERE Voiture.numserie = Vente.numserie
  AND prix-prixachat >= ALL (SELECT prix-prixachat
                             FROM Voiture,Vente
                             WHERE Voiture.numserie = Vente.numserie)
```

(d) Les bénéfices de chaque magasin pour le mois de janvier 2007.
Correction:

```
SELECT magasin, SUM(benef) as benefice
FROM (SELECT nom,prenom,magasin,
             SUM(prixachat-cout-0.5*(prix-cout))-salaire as benef
      FROM Voiture,Vente,Vendeur
      WHERE Voiture.numserie = Vente.numserie
        AND nomv = nom
        AND prenomv = prenom
        AND date BETWEEN '2007-01-01' AND '2007-01-31'
      GROUP BY nom,prenom,magasin) BeneficesVendeurs
GROUP BY magasin
```

(e) Le meilleur client (celui ayant rapporté le plus d'argent à l'entreprise).
Correction:

```
SELECT nomc,prenomc
FROM Vente,Voiture
WHERE Voiture.numserie = Vente.numserie
GROUP BY nomc,prenomc
HAVING SUM(prixachat-cout-0.5*(prix-cout)) >= ALL
        (SELECT SUM(prixachat-cout-0.5*(prix-cout))
         FROM Vente,Voiture
         WHERE Voiture.numserie = Vente.numserie
         GROUP BY nomc,prenomc)
```

(f) La marque pour laquelle on a accordé le plus de rabais.
Correction: A faire

4. Écrire en algèbre relationnelle une requête pour obtenir la voiture ayant le coût le plus élevé. Donner, parmi les requêtes précédentes, celles qui peuvent être traduites en algèbre relationnelle.

   Correction: Requête en algèbre (les voitures moins celles ayant au moins une voiture plus chère):

   $$\pi_{numserie}(Voiture) \setminus \pi_{numserie}(\sigma_{cout<cout_2} Voiture \times \rho_{cout/cout_2}(\pi_{cout}(Voiture)))$$

   Les requêtes a,b et c peuvent être traduites en algèbre relationnelle.

Exercice 2:

On considère le Schéma de la base de données CINEMA:

- FILM (NUMF,TITRE, GENRE, ANNEE, DUREE, BUDGET, REALISATEUR, SALAIRE_REAL)
- DISTRIBUTION (NUMF, NUMA, ROLE, SALAIRE)
- PERSONNE (NUMP, PRENOM, NOM, DATENAIS)
- ACTEUR (NUMA, AGENT, SPECIALITE, TAILLE, POIDS)

L'attribut REALISATEUR de la relation FILM est l'identifiant d'une PERSONNE. Il en est de même pour les attributs NUMA et AGENT de la relation ACTEUR.
Donner les requêtes SQL permettant de répondre aux questions suivantes. Lorsque cela est possible, on donnera également les requêtes équivalentes en calcul relationnel de n-uplets (tuples) et en algèbre relationnelle, puis on donnera un plan d'exécution en utilisant l'optimisation à base de règles.

1. Retrouver la liste de tous les films.

   Correction:

   ```
   SELECT *
   FROM FILM
   ```

2. Retrouver la liste des films dont la longueur dépasse 180 min.

   Correction:

   ```
   SELECT *
   FROM FILM
   WHERE DUREE >180
   ```

3. Donner la liste de tous les genres de film.

   Correction:

   ```
   SELECT DISTINCT GENRE
   FROM FILM
   ```

4. Donner le nombre de films par genre.

   Correction:

```
SELECT GENRE, COUNT(*)
FROM FILM
GROUP BY GENRE
```

5. Trouver le/les titre(s) et l'/les année(s) du/des film(s) le(s) plus long(s).

   Correction:

```
SELECT TITRE, ANNEE
FROM FILM
WHERE DUREE =
(SELECT MAX(DUREE)
FROM FILM)
```

6. Trouver tous les "couples d'acteurs", i.e., les acteurs ayant joués le "Premier" rôle dans un même film (sans doublons).

7. Trouver le nom des personnes qui ne sont ni agents, ni acteurs et ni réalisateurs.

   Correction: Pour les questions suivantes, proposer deux requêtes différentes : une avec des jointures et une autre avec le mot clé "IN".

8. Donner le nom et le prénom des réalisateurs qui ont joué dans au moins un de leurs propres films

   Correction:

   - forme plate:

```
SELECT DISTINCT P.PRENOM, P.NOM
FROM PERSONNE P, FILM F, DISTRIBUTION D
WHERE P.NUMP = F.REALISATEUR
   AND WHERE F.NUMF = D.NUMF
   AND D.NUMA = F.REALISATEUR
```

   - forme imbriquée:

```
SELECT DISTINCT PRENOM, NOM
FROM PERSONNE
WHERE P.NUMP IN (
   SELECT REALISATEUR
   FROM FILM
   WHERE (REALISATEUR, NUMF) IN (
      SELECT NUMA, NUMF
      FROM DISTRIBUTION))
```

9. Quel est le total des salaires des acteurs du film "Nuits blanches à Seattle".

   Correction:

   - forme plate:

```
SELECT SUM(D.SALAIRE)
FROM FILM F, DISTRIBUTION D
WHERE F.NUMF = D.NUMF
AND F.TITRE = 'Nuits blanches à SEATLE'
```

- forme imbriquée:

```
SELECT SUM(SALAIRE)
FROM DISTRIBUTION
WHERE NUMF IN (
   SELECT NUMF
   FROM FILM WHERE TITLE="NUITS BLANCHES à SEATLE")
```

10. Pour chaque film de Spielberg (titre, année), donner le total des salaires des acteurs.

    Correction:

    - forme plate

```
SELECT F.TITRE, F.ANNEE, SUM(D.SALAIRE)
FROM FILM F, DISTRIBUTION D, PERSONNE P
WHERE F.NUMF = D.NUMF
AND F.REALISATEUR = P.NUMP
AND P.NOM = 'Spielberg'
GROUP BY F.TITRE, F.ANNEE
```

    - forme imbriquée

```
SELECT F.TITRE, F.ANNEE, X.SUMSAL
FROM FILM F, (
   SELECT NUMF, SUM(SALAIRE) AS SUMSAL
   FROM DISTRIBUTION
   GROUP BY NUMF ) AS X
WHERE F.NUMPF = X.NUMF
AND F.REALISATEUR IN(
   SELECT NUMP
   FROM PERSONNE
   WHERE NOM='SPIELBERG')
```