

# **Cahier d'exercices TD/TP BDDR**

# TD/TP1

## Exercices sur les modèles Entité Association et Schéma Relationnels

### Exercice 1

Nous souhaitons réaliser une base de données « Routes » pour organiser les informations sur les autoroutes. Les autoroutes sont identifiées par un code CodA dont on doit enregistrer le nom. Les autoroutes sont composées de tronçons d'autoroutes, correspondant au découpage de l'autoroute en plusieurs morceaux afin de faciliter sa gestion. Chaque tronçon est identifié par un code (CodT), qui appartient à une autoroute donnée (ex: A6) et est délimité par un Kilomètre initial (DuKm) et un Kilomètre final (AuKm). Les autoroutes contiennent des sorties d'autoroute portant un nom de sortie (Libelle) et un numéro de sortie (Numero) de valeur 1,2,... pour l'autoroute en question. Les villes qui sont joignables par autoroute sont identifiées par un code postal (CodP) et décrites par un Nom.

1. Proposez un schéma conceptuel de données MCD
2. Traduire le MCD en schéma relationnel

### Exercice 2

1. Proposez un modèle Entité Association à un éditeur qui souhaite mémoriser les informations suivantes dans une base de données :
  - Les livres :
    - Les livres sont identifiés par leur numéro ISBN.
    - Un livre possède un titre et un prix de vente.
    - Il est écrit par un ou plusieurs auteurs.
    - Chaque livre est tiré en une ou plusieurs éditions, datées et identifiées par leur ordre (première édition, seconde édition, etc.).
    - Chaque édition comporte un certain nombre d'exemplaires.
    - Le prix de vente peut changer d'une édition à l'autre.
    - Un livre peut être primé (Goncourt, Fémina etc.)
  - Les auteurs :
    - Les auteurs sont identifiés par leur nom et prénoms.
    - Ils peuvent avoir un pseudonyme.
    - Pour chaque livre, un auteur perçoit des droits d'auteur annuels, calculés comme un pourcentage des ventes (il est aussi fonction du nombre d'auteurs).
  - Les libraires :
    - Ils sont identifiés par leur nom et adresse complète.
    - Ils commandent des livres en précisant l'édition et le nombre d'exemplaires désiré.
2. Traduire le MCD en schéma relationnel.

### Exercice 3

Dix magasins de location de DVD vidéo se sont regroupés pour mettre en commun les DVD dont ils disposent et ont fondé un club de location. A la suite d'une rencontre avec les représentants de ce club, il ressort que chaque point de vente disposera d'un ordinateur personnel relié à un site central et qu'il faudra pouvoir prendre en compte les éléments suivants :

- Un client verse une caution. Suivant le montant de cette caution il aura le droit d'emprunter en même

- temps de 1 à 6 DVD.
- Les DVD empruntés doivent être retournés dans un délai de 3 jours dans n'importe quelle boutique du club.
- Plusieurs DVD peuvent contenir le même film.
- Un film est rattaché à un genre cinématographique (défini par un nom et le type de public auquel il est destiné). Il est caractérisé par sa durée, son réalisateur et la liste des acteurs principaux.
- Une location n'est permise que si le client est en règle (pas de dépassement du nombre d'emprunts maximum, pas de DVD en retard).
- La consultation d'un client permettra d'obtenir son nom, son adresse, son nombre d'emprunts en cours, la liste des numéros de DVD et des titres qu'il a actuellement empruntés.
- La consultation d'un genre permettra d'obtenir la liste des films de ce genre disponibles dans un magasin donné.
- Périodiquement, on veut obtenir la liste des retardataires ; on veut pour chaque DVD non retourné à temps les informations suivantes : nom et adresse du client, date de l'emprunt, numéro(s) de DVD et titre du (des) film(s) concerné(s).
- On veut pouvoir connaître pour chaque DVD (identifié par une numérotation commune aux dix magasins) où il est, quand il a été mis en service, quel film y est enregistré, combien de fois il a déjà loué, et quel est son état (de très bon à mauvais).

Questions : Proposer un modèle conceptuel de données qui décrit le club de location. Donner le modèle relationnel dérivé

#### Exercice 4

On veut modéliser le système d'informations d'une bibliothèque d'entreprise. Cette bibliothèque gère le stockage, l'indexation et l'emprunt d'un ensemble de documents. Chaque document stocké est un des exemplaires d'un ouvrage. Un ouvrage se caractérise par un titre, ses auteurs (nom, prénom, date de naissance), l'année de parution, l'éditeur. Chaque exemplaire d'un ouvrage est caractérisé par une clé alphanumérique appelée « cote » calculée à partir de sa localisation (étage, travée, rayon) et d'un numéro d'exemplaire.

Un ouvrage peut être un livre, un périodique, ou un document interne de l'entreprise. Les documents internes comportent l'indication du service pour lequel ils ont été rédigés. L'auteur d'un périodique est le responsable de la rédaction. La durée du prêt est d'un mois pour un livre ou un document interne, et d'une semaine pour un périodique. On peut renouveler le prêt d'un ouvrage.

Les emprunteurs sont des membres de l'entreprise, travaillant dans un service. Un emprunteur possède un matricule, un nom, prénom, une date de naissance, un service dans lequel il travaille, et un numéro de téléphone professionnel, les exemplaires en cours d'emprunt avec leur date, le projet auquel l'emprunt doit être imputé (pour pouvoir faire des statistiques de prêts par projet), ainsi que tous les emprunts clôturés, avec leur date de début et de fin et les projets auxquels ils ont été imputés.

#### Questions

- Elaborer le modèle Entité/Association correspondant à la description d'une bibliothèque donnée ci-dessus.
- Donner le schéma relationnel dérivé.

#### **Exercice 5 : (transformation de modèles)**

Soient les relations suivantes qui décrivent une auto-école. Les clés primaires sont soulignées, les clés étrangères sont précédées d'un # :

Client (ClientID, Nom, Prénom, Adresse, DateNaissance)

CD-Rom (CdRomID, Editeur)

Question (QuestionID, intitulé, réponse, NiveauDifficulté, Thème)

Série (SerieID, #CdRomID)

Seance-Code (SeanceID, Date, Heure, #SerieID)

ContenuSerie (#QuestionID, #SerieID, Numéro)

Participation (#ClientID, #SeanceID, Nombre de fautes)

Examen\_Code (PassageCodeID, Date, Heure, LieuExamen)

PassageCode (#PassageCodeID, #ClientID, NombreFautes)

**Question** : Déduisez le modèle Entité/Association

## **Exercice 6**

On considère le schéma relationnel de gestion de recettes de cuisine d'un grand restaurant parisien :

- TypesIngrédients(numéro\_type, description) contenant par exemple le tuple (2, viande)
- Ingrédients(numingrédient, nom, #numéro\_type, #numtypemesure) contenant par exemple le tuple (3, bœuf, 2, 5) le numéro de type de mesure décrit l'unité de mesure utilisée pour cet ingrédient.
- Mesures(numtypemesure, description) contenant par exemple le tuple (5, gramme)
- IngrédientsRecette(#numrecette, #numingrédient, numordreingrédient, quantité) contenant par exemple le tuple (12, 3, 1, 500)
- Recettes(numrecette, nomrecette, #numtyperecette, duréepréparation, niveaudifficulté) contenant par exemple le tuple (12, bœuf Stroganoff, 1, 45, 1)
- TypeRecettes(numtyperecette, description) contenant par exemple (1, plat principal)

Remarque : les clés primaires sont soulignées et les clés étrangères sont marquées par #

Proposez le MCD équivalent à ce schéma logique relationnel.

# TD/TP 2

## Normalisation et Dépendances Fonctionnelles

### Exercice 1

On considère la relation suivante, d'attributs, A, B, C, D :

A	B	C	D
a1	b1	c1	d1
a1	b2	c1	d2
a2	b2	c2	d3
a3	b1	c1	d2
a4	b4	c3	d2

### Questions

1. Quel est le degré de cette relation ?
2. Quelle est sa cardinalité ?
3. Peut-il exister une dépendance fonctionnelle  $A \rightarrow B$  ?  $A \rightarrow C$  ?
4. L'un des attributs A, B, C ou D peut-il jouer le rôle de clé ?
5. Quelles sont les associations d'attributs qui pourraient jouer un rôle de clé dans la relation telle qu'elle est décrite ?

### Exercice 2

Pour constituer une base de données sur la scolarité des étudiants, on dispose des éléments suivants :

N° Etudiant, N°-UV, Nom-UV, Nbr-Heures-Enseignement, N°-SS-Enseignant, Nom-Ens, Grade, Indice, UER-Rattachement, Nbr-Enseignant-UER, Resultat-UV, Nom-Etudiant.

UV : unité de valeur (un module ou unité d'enseignement )

UER : unité d'enseignement et de recherche ( équivalent au département )

On fait les hypothèses suivantes :

- Un étudiant peut s'inscrire à plusieurs UV
- Résultat-UV caractérise un étudiant pour une UV donnée
- Nbr-Heures-Enseignement est spécifique à une UV
- Chaque UV n'est assurée que par un enseignant
- Un enseignant est rattaché à une seule UER
- Chaque enseignant à un grade
- A chaque grade correspond un indice
- Nbr-Enseignant-UER est spécifique à chaque UER-Rattachement

### Questions

1. Quelle est la clé de la relation construite sur l'ensemble des attributs donnés ci-dessus ?
2. Trouver la couverture minimale des dépendances fonctionnelles.
3. Tracer le graphe des dépendances fonctionnelles de la couverture minimale.
4. Trouver graphiquement la fermeture transitive de l'ensemble des dépendances.
5. Proposer si nécessaire une décomposition 3FN

### **Exercice 3**

On considère la relation suivante décrivant des voitures :

R (N°-Immat, Puissance, Marque, Pays, Agence, Chiffre-Aff)

Soient les hypothèses suivantes :

- Chaque véhicule est caractérisé par une puissance et une marque.
- Une marque est spécifique d'un pays.
- Le chiffre d'affaires fait référence à une agence pour une marque donnée.
- Une agence peut représenter plusieurs marques.

#### **Questions**

1. Quelle est la clé de la relation ?
2. En quelle forme normale est cette relation ?
3. Donner des exemples de dépendances transitives et pseudotransitive.
4. Donner la couverture minimale des DF. Tracer le graphe correspondant.
5. Trouver la fermeture transitive (utiliser la propriété de transitivité et des dépendances).
6. Proposer si nécessaire une décomposition 3FN

### **Exercice 4**

Soient les relations R1 et R2 suivantes et l'ensemble de DF F1 et F2 définis respectivement sur ces relations :

R1 = (N°-Piece, Prix-unit, Taux-Tva, Libelle, Catégorie)

F1 = {N°-Piece → Prix-unit ; N°Pice→Libelle, N°Pice→Catégorie; Catégorie→Taux-TVA ; N°-Pice → Taux-TVA}

R2 = (N°-Gamme, Nom-Gamme, N°-Oper, Rang-Oper, Nom-Oper)

F2 = {N°-Gamme →Nom-Gamme ; N°-Oper → Nom-Oper ; N°-Gamme, N°-Oper → Rang-Oper}

#### **Questions**

1. Quelles sont les clés primaires des relations R1 et R2 ?
2. F1 et F2 constituent-ils une couverture minimale.
3. En quelle forme normale sont R1 et R2 ?
4. Proposer si nécessaire une décomposition 3FN

### **Exercice 5**

Les organisateurs d'un tournoi d'équipes de (football, basket, volley) veulent utiliser une base de données relationnelle pour l'organisation des matchs. Soit le schéma relationnel suivant :

MATCH (num-tour, num-match, ville, stade, date, arbitre, equipe1, equipe2, resultat)

Les dépendances fonctionnelles sont :

num-tour, num-match -> ville, stade, date, arbitre, equipe1, equipe2, resultat

stade -> ville

1/ Donner une (01) clé de la relation MATCH. Justifier votre réponse.

2/ Est-ce que la relation MATCH est en 3FN ? Justifier votre réponse.

3/ Donner une décomposition sans perte d'information, qui préserve les dépendances fonctionnelles et dont le résultat est en 3FN.

### **Exercice 6**

Soit la relation universelle **commande** de schéma suivant :

**Commande** (num-commande, Num-produit, Quantité-commande, Num-client, Num-représentant)

Avec les dépendances fonctionnelles suivantes :

Num-commande, Num-produit  $\rightarrow$  Quantité-commande, Num-client, Num-représentant

Num-commande  $\rightarrow$  Num-client , Num-représentant

Num-client  $\rightarrow$  Num-représentant

Questions :

1/ Expliquer pourquoi cette relation n'est pas en 2FN

2/ Décomposer la relation **Commande** pour obtenir un schéma relationnel en 2FN tout en préservant les dépendances fonctionnelles. Attention, on ne demande ici que la 2FN.

3/ Les relations obtenues sont-elles en 3FN ? Expliquer votre réponse. Si ce n'est pas le cas, modifier le schéma afin d'obtenir un résultat en 3FN en utilisant l'algorithme de synthèse (algorithme de décomposition en 3FN).

# TD/TP 3

## SQL1

Avant de faire le TD/TP, il faut installer MySQL (voir le document installation MySQL )

### Exercice 1

A titre d'exemple voici une base de données relationnelle rudimentaire utilisant 3 tables représentant les commandes de produits à des fournisseurs.

produits	pno	design	prix	poids	couleur
	102	fauteuil	1500	9	rouge
	103	bureau	3500	30	vert
	101	fauteuil	2000	7	gris
	105	armoire	2500	35	rouge
	104	bureau	4000	40	gris
	107	caisson	1000	12	jaune
	106	caisson	1000	12	gris
	108	classeur	1500	20	bleu

fournisseurs	fno	nom	adresse	ville
	10	Dupont		Lille
	15	Durand		Lille
	17	Lefebvre		Lille
	12	Jacquet		Lyon
	14	Martin		Nice
	13	Durand		Lyon
	11	Martin		Amiens
	19	Maurice		Paris
	16	Dupont		Paris

commandes	cno	fno	pno	qute
	1001	17	103	10
	1003	15	103	2
	1005	17	102	1
	1007	15	108	1
	1011	19	107	12
	1013	13	107	5
	1017	19	105	3
	1019	14	103	10
	1023	10	102	8
	1029	17	108	15



## 1- Les requêtes de création et d'insertion

Créer la base de données et insérer les données présentées ci-dessus (analyser le script ci-dessous et recopier dans un fichier « BDD.sql »)

**Pour écrire vos scripts SQL vous devez utiliser un éditeur de texte :**

**Notepad++ pour Windows (<https://notepad-plus-plus.org/downloads/>)** -

**ou**

**sublimetext pour Mac**

(<https://www.sublimetext.com>) ou un autre éditeur

sur le SGBD 'MySQL' vous importez pour exécuter ce script pour créer la base de données

```
DROP DATABASE IF EXISTS BDCOMMANDES;  
CREATE DATABASE BDCOMMANDES ;  
USE BDCOMMANDES ;  
DROP TABLE IF EXISTS Fournisseurs ;  
DROP TABLE IF EXISTS Produits ;  
DROP TABLE IF EXISTS Commandes ;
```

```
CREATE TABLE Fournisseurs (  
  fno Numeric(6) NOT NULL primary key,  
  nom VARCHAR(25) NOT NULL ,  
  adresse VARCHAR(25) ,  
  ville VARCHAR(25) NOT NULL  
);
```

```
CREATE TABLE Produits (  
  pno Numeric(6) NOT NULL primary key ,  
  design VARCHAR(25) NOT NULL ,  
  prix Numeric(6,2) NOT NULL ,  
  poids Numeric(6,2) NOT NULL ,  
  couleur VARCHAR(25) NOT NULL  
);
```

```
CREATE TABLE Commandes (  
  cno Numeric(6) NOT NULL primary key ,  
  fno Numeric(6) NOT NULL ,  
  pno Numeric(6) NOT NULL ,  
  qute INT(5) NOT NULL ,
```

```
  constraint FK1 foreign key (fno) references Fournisseurs(fno),  
  constraint FK2 foreign key (pno) references Produits(pno)
```

```
);
```

```
INSERT INTO FOURNISSEURS (Fno,Nom,Adresse,Ville) VALUES  
(10,"Dupont",NULL,"Lille"),(15,"Durand",NULL,"Lille"),  
(17,"Lefebvre",NULL,"Lille"),(12,"Jacquet",NULL,"Lyon"),  
(14,"Martin",NULL,"Nice"),(13,"Durand",NULL,"Lyon"),  
(11,"Martin",NULL,"Amiens"),(19,"Maurice",NULL,"Paris"),  
(16,"Dupont",NULL,"Paris");
```

```
INSERT INTO PRODUITS (Pno,Design,Prix,Poids,Couleur) VALUES  
(102,"fauteuil",1500,9,"rouge"),(103,"bureau",3500,30,"vert"),  
(101,"fauteuil",2000,7,"gris"),(105,"armoire",2500,35,"rouge"),  
(104,"bureau",4000,40,"gris"),(107,"caisson",1000,12,"jaune"),  
(106,"caisson",1000,12,"gris"),(108,"classeur",1500,20,"bleu");
```

```
INSERT INTO COMMANDES VALUES  
(1001,17,103,10),(1003,15,103,2),(1005,17,102,1),  
(1007,15,108,1),(1011,19,107,12),(1013,13,107,5),
```

(1017,19,105,3),(1019,14,103,10),(1023,10,102,8),  
(1029,17,108,15);

## **2- Requête d'exploitation**

1. Lister la table fournisseurs
2. Lister les numéros et noms des fournisseurs
3. Lister les différentes désignations de produits
4. Lister les données sur les produits dont le poids est supérieur à 15
5. Lister les désignations de produits différents dont le poids est supérieur à 15
6. Lister les produits dont le poids est compris entre 15 et 40
7. Lister les fournisseurs habitant Lille, Lyon ou Nice
8. Lister les produits dont le poids n'est compris entre 15 et 35
9. Lister les fournisseurs dont le nom ne commence pas par 'd'
10. Lister les fournisseurs dont l'adresse n'est pas renseignée
11. Lister les fournisseurs dont l'adresse est renseignée
12. Afficher les produits rouge ou verts ou bleus triés sur le nom croissant et la couleur décroissante
13. Afficher les produits et leurs couleurs avec des noms de colonnes compréhensibles
14. Lister le produit cartésien fournisseur \* produits
15. Lister les noms des fournisseurs avec les numéros de produits commandés ainsi que la qualité commandée
16. Utilisation des synonymes de noms de tables
17. Lister les couples (nom de fournisseur, nom de produit) en commande
18. Lister les couples de références de fournisseurs situés dans la même ville
19. Afficher tous les produits de moins de 20kg avec les quantités en cours de commande si possible.
20. Afficher les produits qui ne sont pas commandés
21. Donner Prix des produits avec une TVA à 20,6%
22. Donner le volume financier commandé pour les commandes de moins de 10 articles
23. Compter le nombre de commandes
24. Afficher la somme de toutes les qualités commandées
25. Compter les noms de fournisseurs différents
26. Compter le nombre de livraisons du produit numéro 102
27. Récupérer toutes les statistiques sur les quantités en commande
28. Afficher la somme des quantités commandées à chaque fournisseur
29. Lister le nombre de commandes par fournisseur
30. Lister les fournisseurs qui ont moins de 3 commandes
31. Lister les références des fournisseurs qui fournissent plus d'un produit
32. On s'intéresse ici qu'aux commandes dont le numéro est supérieur à 6 et, pour ces commandes, on souhaite lister les noms des fournisseurs avec les quantités maximum commandées, dont la moyenne des quantités commandées est supérieure à 50, le résultat doit être trié par ordre décroissant du nom de fournisseur
33. Lister les fournisseurs qui habitent la même ville que le fournisseur 10
34. Lister les fournisseurs d'au moins un des produits fournis par au moins un des fournisseurs d'un produit rouge
35. Lister les noms des fournisseurs livrant tous les produits
36. Lister les références des fournisseurs livrant au moins un produit an quantité supérieure à chacun des produits livrés par fournisseur 19
37. Lister les numéros de produits dont le poids est supérieur à 20 ainsi que les produits commandés par le fournisseur 15

# TD/TP 5 (4h)

## SQL2

### I – Description de la base de données :

Analyser puis exécuter le script sql fourni ci-dessous puis déduire le schéma relationnel de la base de données. Recopier ce script dans un fichier cinema.sql puis l'importer dans le SGBD pour créer la base des données

### II – Interrogation de la base de données :

Trouvez les ordres SQL correspondants aux requêtes suivantes.

**Remarque :** (Il faut donner la requête SQL et son résultat d'exécution)

#### Partie 1 (Sélections simples)

1. Les titres des films triés par ordre croissant.
2. Nom et année de naissance des artistes nés avant 1950.
3. Les cinémas du 12ème arrondissement.
4. Les artistes dont le nom commencent par 'H' (commande LIKE).
5. Quels sont les acteurs dont on ignore la date de naissance ? (Attention : cela signifie que la valeur n'existe pas).
6. Combien de fois Bruce Willis a-t-il joué le rôle de McLane ?

#### Partie 2 (Négation , Fonctions de groupe)

7. Quels acteurs n'ont jamais mis en scène de film ?
8. Les cinémas (nom, adresse) qui ne passent pas un film de Tarantino.
9. Total des places dans les salles du Rex.
10. Année du film le plus ancien et du film le plus récent.
11. Total des places offertes par cinéma.

#### Partie 3 (Jointures)

12. Qui a joué Tarzan (nom et prénom) ?
13. Nom des acteurs de Vertigo.
14. Quels films peut-on voir au Rex, et à quelle heure ?
15. Titre des films dans lesquels a joué Woody Allen. Donner aussi le rôle.
16. Quel metteur en scène a tourné dans ses propres films ? Donner le nom, le rôle et le titre des films.
17. Quel metteur en scène a tourné en tant qu'acteur ? Donner le nom, le rôle et le titre des films où le metteur en scène a joué.
18. Où peut-on voir Shining ? (Nom et adresse du cinéma, horaire).
19. Dans quels films le metteur-en-scène a-t-il le même prénom que l'un des interprètes ? (titre, nom du metteur-en-scène, nom de l'interprète). Le metteur-en-scène et l'interprète ne doivent pas être la même personne.
20. Où peut-on voir un film avec Clint Eastwood ? (Nom et adresse du cinéma, horaire).
21. Quel film peut-on voir dans le 12e arrondissement, dans une salle climatisée ? (Nom du cinéma, numéro de la salle, horaire, titre du film).
22. Liste des cinémas (Adresse, Arrondissement) ayant une salle de plus de 150 places et passant un film avec Bruce Willis.
23. Liste des cinémas (Nom, Adresse) dont TOUTES les salles ont plus de 100 places

24. Nom et prénom des réalisateurs, et nombre de films qu'ils ont tournés.
25. Nom des cinémas ayant plus de 1 salle climatisée.
26. Les artistes (nom, prénom) ayant joué au moins dans trois films depuis 1985, dont au moins un, passe à l'affiche à Paris (donner aussi le nombre de films).
27. Le nom des cinémas qui passent tous les films de

Kubrick Le script cinema.sql

```

-----
-- Nom de la base de donnees : cinema --
-- SGBD : MySql --
-- Destruction des tables

create database
cinema; use cinema;

DROP TABLE IF EXISTS
Seance ; DROP TABLE IF
EXISTS Salle ; DROP TABLE
IF EXISTS Cinema ; DROP
TABLE IF EXISTS Role ; DROP
TABLE IF EXISTS Film ;
DROP TABLE IF EXISTS
Artiste ;

-- Structure de la table 'Artiste'

CREATE TABLE Artiste (
  Nom varchar(20) NOT NULL,
  Prenom varchar(15),
  Annee_naissance decimal(4,0),
  PRIMARY KEY (Nom)
)Engine='InnoDB';

-- Contenu de la table 'Artiste'

INSERT INTO Artiste VALUES( 'Allen', 'Woody',
'1938'); INSERT INTO Artiste VALUES( 'Lynch',
'David', '1946');
INSERT INTO Artiste VALUES( 'Kusturica', 'Emir',
'1959'); INSERT INTO Artiste VALUES( 'Lang', 'Fritz',

```

```

INSERT INTO Artiste VALUES( 'Novak', 'Kim', NULL);
INSERT INTO Artiste VALUES( 'Hunt', 'Greg', '1950');
INSERT INTO Artiste VALUES( 'Tarantino', 'Quentin', '1948');
INSERT INTO Artiste VALUES( 'Willis', 'Bruce', '1952');
INSERT INTO Artiste VALUES( 'Spielberg', 'Steven', '1943');
INSERT INTO Artiste VALUES( 'Hudson', 'Hugh', NULL);
INSERT INTO Artiste VALUES( 'Gillian', 'Terry', '1944');
INSERT INTO Artiste VALUES( 'Truffaut', 'Francois', '1938');
INSERT INTO Artiste VALUES( 'Lambert', 'Christophe', '1953');
INSERT INTO Artiste VALUES( 'Keitel', 'Harvey', '1940');
INSERT INTO Artiste VALUES( 'Woo', 'John', '1951');
INSERT INTO Artiste VALUES( 'Travolta', 'John', '1953');
INSERT INTO Artiste VALUES( 'Cage', 'Nicolas', '1954');
INSERT INTO Artiste VALUES( 'DiCaprio', 'Leonardo', '1973');
INSERT INTO Artiste VALUES( 'Cameron', 'James', '1943');
INSERT INTO Artiste VALUES( 'Cruise', 'Tom', '1960');
INSERT INTO Artiste VALUES( 'De Palma', 'Brian', '1953');
INSERT INTO Artiste VALUES( 'Depp', 'Johnny', '1967');
INSERT INTO Artiste VALUES( 'Ricci', 'Christina', '1974');
INSERT INTO Artiste VALUES( 'Burton', 'Tim', '1958');

```

-----

-- Structure de la table 'Cinema'

```

CREATE TABLE Cinema (
    Nom_cinema varchar(10) NOT NULL,
    Arrondissement decimal(2,0),
    Adresse varchar(30),
    PRIMARY KEY (Nom_cinema)
)Engine='InnoDB';

```

-- Contenu de la table 'Cinema'

```

INSERT INTO Cinema VALUES( 'Rex', '2', '22 Bd Poissoniere');
INSERT INTO Cinema VALUES( 'Kino', '15', '3 Bd Raspail');
INSERT INTO Cinema VALUES( 'Nations', '12', '3 Rue de Reuilly');
INSERT INTO Cinema VALUES( 'Halles', '1', 'Forum des Halles');

```

-- Structure de la table 'Film'

```
CREATE TABLE Film (
  ID_film decimal(10,0) DEFAULT '0' NOT NULL,
  Titre varchar(30),
  Annee decimal(4,0),
  Nom_Réalisateur varchar(20),
  PRIMARY KEY (ID_film)
)Engine='InnoDB';
```

-- Contenu de la table 'Film'

```
INSERT INTO Film VALUES( '10', 'Annie Hall', '1977', 'Allen');
INSERT INTO Film VALUES( '57', 'Brazil', '1984', 'Gillian');
INSERT INTO Film VALUES( '5', 'Underground', '1995', 'Kusturica');
INSERT INTO Film VALUES( '38', 'Metropolis', '1926', 'Lang');
INSERT INTO Film VALUES( '45', 'Impitoyable', '1992', 'Eastwood');
INSERT INTO Film VALUES( '65', 'Vertigo', '1958', 'Hitchcock');
INSERT INTO Film VALUES( '7', 'Shining', '1980', 'Kubrick');
INSERT INTO Film VALUES( '6', 'Psychose', '1960', 'Hitchcock');
INSERT INTO Film VALUES( '3', 'Twin Peaks', '1990', 'Lynch');
INSERT INTO Film VALUES( '90', 'Casablanca', '1942', 'Curtis');
INSERT INTO Film VALUES( '85', 'Greystocke', '1984', 'Hudson');
INSERT INTO Film VALUES( '89', 'Le dernier metro', '1980', 'Truffaut');
INSERT INTO Film VALUES( '1', 'Reservoir Dogs', '1992', 'Tarantino');
INSERT INTO Film VALUES( '43', 'Manhattan', '1979', 'Allen');
INSERT INTO Film VALUES( '11', 'Jurassic Park', '1992', 'Spielberg');
INSERT INTO Film VALUES( '32', 'Rencontres du 3eme type', '1978', 'Spielberg');
INSERT INTO Film VALUES( '33', 'Piege de cristal', '1990', 'Hunt');
INSERT INTO Film VALUES( '34', 'Une journee en enfer', '1994', 'Hunt');
INSERT INTO Film VALUES( '35', '48 minutes pour vivre', '1992', 'Hunt');
INSERT INTO Film VALUES( '73', 'Pulp Fiction', '1995', 'Tarantino');
INSERT INTO Film VALUES( '101', 'Broken Arrow', '1996', 'Woo');
INSERT INTO Film VALUES( '102', 'Volte-Face', '1997', 'Woo');
INSERT INTO Film VALUES( '104', 'Titanic', '1998', 'Cameron');
INSERT INTO Film VALUES( '135', 'Mission Impossible 2', '2000', 'Woo');
INSERT INTO Film VALUES( '136', 'Mission Impossible', '1997', 'De Palma');
INSERT INTO Film VALUES( '142', 'Edward scissorhands', '1990', 'Burton');
INSERT INTO Film VALUES( '141', 'Sleepy Hollow', '1999', 'Burton');
```

-- Structure de la table 'Role'

CREATE TABLE Role (

Nom\_Role varchar(20) NOT NULL,

ID\_film decimal(10,0) DEFAULT '0' NOT NULL,

Nom\_acteur varchar(20) NOT NULL,

PRIMARY KEY (ID\_film, Nom\_acteur)

)Engine='InnoDB';

-- Contenu de la table 'Role'

INSERT INTO Role VALUES( 'Bernard', '32', 'Truffaut');

INSERT INTO Role VALUES( 'Davis', '43', 'Allen');

INSERT INTO Role VALUES( 'Tarzan', '85', 'Lambert');

INSERT INTO Role VALUES( 'Ferguson', '65', 'Stewart');

INSERT INTO Role VALUES( 'Elster', '65', 'Novak');

INSERT INTO Role VALUES( 'Jonas', '10', 'Allen');

INSERT INTO Role VALUES( 'McLane', '33', 'Willis');

INSERT INTO Role VALUES( 'McLane', '34', 'Willis');

INSERT INTO Role VALUES( 'McLane', '35', 'Willis');

INSERT INTO Role VALUES( 'Mr Brown', '1', 'Tarantino');

INSERT INTO Role VALUES( 'Munny', '45', 'Eastwood');

INSERT INTO Role VALUES( 'Mr White', '1', 'Keitel');

INSERT INTO Role VALUES( 'Wolf', '73', 'Keitel');

INSERT INTO Role VALUES( 'Coolidge', '73', 'Willis');

INSERT INTO Role VALUES( 'Vega', '73', 'Travolta');

INSERT INTO Role VALUES( 'Deakins', '101', 'Travolta');

INSERT INTO Role VALUES( 'Archer', '102', 'Travolta');

INSERT INTO Role VALUES( 'Troy', '102', 'Cage');

INSERT INTO Role VALUES( 'Dowson', '104', 'DiCaprio');

INSERT INTO Role VALUES( 'Howard', '135', 'Cruise');

INSERT INTO Role VALUES( 'Howard', '136', 'Cruise');

INSERT INTO Role VALUES( 'Crane', '141', 'Depp');

INSERT INTO Role VALUES( 'Edward', '142', 'Depp');

INSERT INTO Role VALUES( 'Van Tassel', '141', 'Ricci');

-- Structure de la table 'Salle'

```

CREATE TABLE Salle (
    Nom_cinema varchar(10) NOT NULL,
    No_salle decimal(2,0) DEFAULT '0' NOT NULL,
    Climatise char(1),
    Capacite decimal(4,0),
    PRIMARY KEY (Nom_cinema, No_salle)
)Engine='InnoDB';

-- Contenu de la table 'Salle'

INSERT INTO Salle VALUES( 'Rex', '1', 'O', '150');
INSERT INTO Salle VALUES( 'Rex', '2', 'O', '100');
INSERT INTO Salle VALUES( 'Rex', '3', 'N', '80');
INSERT INTO Salle VALUES( 'Rex', '4', 'N', '80');
INSERT INTO Salle VALUES( 'Kino', '1', 'N', '280');
INSERT INTO Salle VALUES( 'Kino', '2', 'O', '120');
INSERT INTO Salle VALUES( 'Kino', '3', 'O', '130');
INSERT INTO Salle VALUES( 'Nations', '1', 'O', '130');
INSERT INTO Salle VALUES( 'Nations', '2', 'N', '90');
INSERT INTO Salle VALUES( 'Nations', '3', 'N', '60');
INSERT INTO Salle VALUES( 'Halles', '1', 'O', '75');
INSERT INTO Salle VALUES( 'Halles', '2', 'N', '60');
INSERT INTO Salle VALUES( 'Halles', '3', 'N', '60');

-- Structure de la table 'Seance'

CREATE TABLE Seance (
    Nom_cinema varchar(10) NOT NULL,
    No_salle decimal(2,0) DEFAULT '0' NOT NULL,
    No_seance decimal(2,0) DEFAULT '0' NOT NULL,
    Heure_debut decimal(4,2),
    Heure_fin decimal(4,2),
    ID_film decimal(10,0) DEFAULT '0' NOT NULL,
    PRIMARY KEY (Nom_cinema, No_salle, No_seance)
)Engine='InnoDB';

-- Contenu de la table 'Seance'

```



```

INSERT INTO Seance VALUES( 'Rex', '1', '3', '9,99', '9,99', '1');
INSERT INTO Seance VALUES( 'Rex', '1', '4', '9,99', '9,99', '6');
INSERT INTO Seance VALUES( 'Rex', '2', '1', '9,99', '9,99', '34');
INSERT INTO Seance VALUES( 'Rex', '2', '2', '9,99', '9,99', '34');
INSERT INTO Seance VALUES( 'Rex', '2', '3', '9,99', '9,99', '7');
INSERT INTO Seance VALUES( 'Rex', '2', '4', '9,99', '9,99', '65');
INSERT INTO Seance VALUES( 'Rex', '3', '1', '9,99', '9,99', '11');
INSERT INTO Seance VALUES( 'Rex', '3', '2', '9,99', '9,99', '11');
INSERT INTO Seance VALUES( 'Rex', '3', '3', '9,99', '9,99', '11');
INSERT INTO Seance VALUES( 'Rex', '4', '1', '9,99', '9,99', '38');
INSERT INTO Seance VALUES( 'Rex', '4', '2', '9,99', '9,99', '38');
INSERT INTO Seance VALUES( 'Rex', '4', '3', '9,99', '9,99', '38');
INSERT INTO Seance VALUES( 'Kino', '1', '1', '9,99', '9,99', '34');
INSERT INTO Seance VALUES( 'Kino', '1', '2', '9,99', '9,99', '73');
INSERT INTO Seance VALUES( 'Kino', '1', '3', '9,99', '9,99', '34');
INSERT INTO Seance VALUES( 'Kino', '2', '1', '9,99', '9,99', '43');
INSERT INTO Seance VALUES( 'Kino', '2', '2', '9,99', '9,99', '7');
INSERT INTO Seance VALUES( 'Kino', '2', '3', '9,99', '9,99', '43');
INSERT INTO Seance VALUES( 'Kino', '3', '1', '9,99', '9,99', '101');
INSERT INTO Seance VALUES( 'Kino', '3', '2', '9,99', '9,99', '102');
INSERT INTO Seance VALUES( 'Kino', '3', '3', '9,99', '9,99', '104');
INSERT INTO Seance VALUES( 'Kino', '3', '4', '9,99', '9,99', '104');
INSERT INTO Seance VALUES( 'Nations', '1', '1', '9,99', '9,99', '65');
INSERT INTO Seance VALUES( 'Nations', '1', '2', '9,99', '9,99', '65');
INSERT INTO Seance VALUES( 'Nations', '1', '3', '9,99', '9,99', '65');
INSERT INTO Seance VALUES( 'Nations', '2', '1', '9,99', '9,99', '43');
INSERT INTO Seance VALUES( 'Nations', '2', '2', '9,99', '9,99', '43');
INSERT INTO Seance VALUES( 'Nations', '2', '3', '9,99', '9,99', '43');
INSERT INTO Seance VALUES( 'Nations', '3', '1', '9,99', '9,99', '7');
INSERT INTO Seance VALUES( 'Nations', '3', '2', '9,99', '9,99', '7');
INSERT INTO Seance VALUES( 'Nations', '3', '3', '9,99', '9,99', '7');
INSERT INTO Seance VALUES( 'Halles', '1', '1', '9,99', '9,99', '32');
INSERT INTO Seance VALUES( 'Halles', '1', '2', '9,99', '9,99', '32');
INSERT INTO Seance VALUES( 'Halles', '1', '3', '9,99', '9,99', '32');
INSERT INTO Seance VALUES( 'Halles', '2', '1', '9,99', '9,99', '5');
INSERT INTO Seance VALUES( 'Halles', '2', '2', '9,99', '9,99', '5');
INSERT INTO Seance VALUES( 'Halles', '2', '3', '9,99', '9,99', '45');
INSERT INTO Seance VALUES( 'Halles', '3', '1', '9,99', '9,99', '3');
INSERT INTO Seance VALUES( 'Halles', '3', '2', '9,99', '9,99', '3');
INSERT INTO Seance VALUES( 'Halles', '3', '3', '9,99', '9,99', '3');

```

**COMMIT:**

## TD/TP 6 (2h)

### SQL3

Soit le schéma relationnel suivant :

Emp (empno, ename, job, *mgr*, hiredate, sal, comm, *#deptno*)

Dept (deptno, dname, loc)

Mission (missno, ciename, *#empno*, local, enddate)

Les attributs soulignés sont les clés primaires et les attributs en italique sont les clés étrangères.

#### Relation dept

<u>deptno</u>	<u>dname</u>	<u>loc</u>
10	Accounting	New-York
20	Research	Dallas
30	Sales	Chicago
40	Operations	Boston

#### Relation emp

<u>empno</u>	<u>ename</u>	<u>job</u>	<u>mgr</u>	<u>hiredate</u>	<u>sal</u>	<u>comm</u>	
<u>deptno</u>							
7369	Smith	Clerk	7902	1981-12-17	800.00	\N	20
7499	Allen	Salesman	7698	1981-02-20	1600.00	300.00	30
7521	Ward	Salesman	7698	1981-02-22	1250.00	500.00	30
7566	Jones	Manager	7839	1981-04-02	2975.00	\N	20
7654	Martin	Salesman	7698	1981-09-28	1250.00	1400.00	30
7698	Blake	Manager	7839	1981-05-01	2850.00	\N	30
7782	Clark	Manager	7839	1981-06-09	2450.00	\N	10
7788	Scott	Analyst	7566	1981-11-09	\N	20	
				3000.00			
7839	King	President	\N	1981-11-17	5000.00	\N	10
7844	Turner	Salesman	7698	1981-09-08	1500.00	0.00	30
7876	Adams	Clerk	7788	1981-09-23	1100.00	\N	20
7900	James	Clerk	7698	1981-12-03	950.00	\N	30
7902	Ford	Analyst	7566	1981-12-03	3000.00	\N	20
7934	Miller	Clerk	7782	1982-01-23	1300.00	\N	10

**Relation mission**

<b><u>Missno</u></b>	<b><u>empno</u></b>	<b><u>ciename</u></b>	<b><u>local</u></b>	<b><u>enddate</u></b>
209	7654	BMW	Berlin	2001-02-09
212	7698	MacDoChicago		2001-03-04
213	7902	Oracle	Dallas	2001-04-11

214	7900	FIDAL	Paris	2001-06-07
216	7698	IBM	Chicago	2001-02-09
218	7499	Décathlon	Clermont	2002-12-24
219	7782	BMW	Chicago	2001-08-16

La valeur \N est la valeur NULL.

#### **Table dept**

deptno int not null primary key,  
dname varchar(20) not null,  
loc varchar(20) not null

#### **Table emp**

empno int not null primary key,  
ename varchar(20) not null,  
job varchar(20) not null,  
mgr int,  
hiredate date not null,  
sal double(7,2) not null,  
comm double(7,2),  
deptno int not null

#### **Table mission**

missno int not null primary key,  
empno int not null,  
ciename varchar(20) not null,  
local varchar(20) not null,  
enddate date

## **Question 1**

Donner le script complet pour créer les tables et insérer les données.

## **Question 2 : Manipulation de données :**

Répondez aux requêtes suivantes.

### a) Projections :

- R1. Extraire les différentes valeurs de salaire présentes dans emp.
- R2. Extraire tous les salaires de emp.
- R3. Extraire l'ensemble des jobs des employés de emp.
- R4. Faire afficher les tables emp, dept et mission en entier.
- R5. Extraire les couples (job,mgr) correspondant aux employés de emp.

### b) Sélections :

- R6. Extraire les missions qui se terminent avant le premier mai 2001 et qui ont lieu aux USA.  
R7. Extraire les missions qui ont lieu dans une ville commençant par 'C'.  
R8. Extraire les numéros et noms des employés ayant été embauchés entre les mois de mai et octobre (compris) 1981.  
R9. Donner les numéros des différents départements qui possèdent des employés ayant une commission non nulle.

c) Jointures :

- R10. Donner les noms des départements qui possèdent au moins un employé dont le job est 'Clerk'.  
R11. Donner les noms des employés qui effectuent une mission dans la ville où se trouve leur département. Donner dans la même requête le nom du département et la ville concernée.  
R12. Trouver les noms des vendeurs et des managers n'ayant pas de mission et ayant soit une commission non nulle, soit un salaire compris entre 2500.00 et 3000.00\$.  
R13. Trouver les noms des départements qui n'ont pas d'employé en mission.  
R14. Trouver les couples (employé,manager) qui n'appartiennent pas au même département. Présenter le résultat sous la forme : Num de manager | Num de l'employé

d) Calculs verticaux :

- R15. Trouver le total des salaires du département sales.  
R16. Trouver les nom, job et salaire de l'employé ayant le salaire le plus élevé.  
R17. Trouver le nombre de managers.  
R18. Trouver le salaire moyen des employés qui sont en mission.  
R19. S'il est en mission, trouver la date de fin de mission de l'employé qui est dans l'entreprise depuis le plus longtemps.

e) Calculs horizontaux :

- R20. Trouver les noms, salaire, commission, salaire+commission de tous les vendeurs.  
R21. Trouver les noms des employés qui touchent une commission comprise entre 35 et 120% de leur salaire.

f) ORDER BY – GROUP BY – HAVING :

- R22. Donner les noms des employés par ordre alphabétique.  
R23. Donner les noms des employés par ordre alphabétique inversé.  
R24. Donner les informations sur les employés par ordre décroissant de date d'embauche puis par ordre alphabétique de nom.  
R25. Présenter les employés par groupes de même valeur de salaire, et par ordre décroissant de ces valeurs.  
R26. Afficher le salaire moyen par job.  
R27. Nom des départements qui ont 2 missions ou plus.  
R28. Ville et nom des départements qui ont plus de quatre employés.  
R29. Nom des départements où les employés gagnent en moyenne plus de 2500\$.  
R30. Nom des villes qui accueillent 2 missions ou plus.

