

Correction du TP 2 SQL

Thème Bases de données

Terminale NSI

Exercice Requêtes sur une table

1. Écrire une requête qui renvoie toutes les lignes de la table **personne** dont la valeur de la colonne **prenom** est 'John'.

```
SELECT *  
FROM personne  
WHERE prenom = 'John';
```

2. Écrire une requête qui sélectionne les lignes de la table **personne** dont la valeur de la colonne **prenom** commence par la lettre 'C' et qui renvoie une table avec deux colonnes **prenom** puis **nom**.

```
SELECT prenom, nom  
FROM personne  
WHERE prenom LIKE 'C%';
```

3. Écrire une requête qui renvoie le nombre de films dont l'année de sortie est postérieure à 1980.

```
SELECT COUNT(*)  
FROM film  
WHERE 1980 < annee;
```

4. Écrire une requête qui renvoie le titre du film et l'année de sortie pour tous les films sortis dans les années 80 (entre 1980 et 1989).

```
SELECT titre, annee  
FROM film  
WHERE 1980 <= annee AND annee <= 1989;
```

5. Écrire une requête qui renvoie une table avec une seule ligne et une seule colonne **annee_min** contenant l'année de sortie la plus ancienne pour les films de la table **film**.

```
SELECT MIN(annee) AS annee_min  
FROM film ;
```

6. Écrire une requête qui renvoie le titre et l'année de sortie du ou des film(s) les plus ancien(s) pour les films de la table **film**. *Utiliser une sous-requête.*

```
SELECT titre, annee
FROM film
WHERE annee = (
    SELECT MIN(annee)
    FROM film
);
```

- Écrire une requête qui renvoie les films de la table `film` dans une table avec deux colonnes `annee` puis `titre`.

Les lignes doivent être triées par ordre décroissant pour la colonne `annee` puis croissant pour la colonne `titre`.

```
SELECT titre, annee
FROM film
ORDER BY annee DESC, titre ASC ;
```

Exercice Requetes sur deux tables

- Écrire une requête qui renvoie les films de la base sous la forme d'une table avec quatre colonnes : `titre`, `annee`, `nom du réalisateur`, `prénom du réalisateur`. Les lignes doivent être classées par ordre croissant sur la colonne `nom du réalisateur` puis ordre croissant sur la colonne `annee`.

```
SELECT titre, annee, personne.nom AS nom_realisateur, personne.prenom AS
    prenom_realisateur
FROM film JOIN personne ON film.idr = personne.idp
ORDER BY nom_realisateur ASC, annee ASC;
```

- Modifier la requête précédente pour qu'elle sélectionne uniquement les films réalisés par Charlie Chaplin, classés par `annee` croissante.

```
SELECT titre, annee, personne.nom AS nom_realisateur, personne.prenom AS
    prenom_realisateur
FROM film JOIN personne ON film.idr = personne.idp
WHERE nom_realisateur = 'Chaplin' AND prenom_realisateur = 'Charlie'
ORDER BY annee ;
```

- Écrire une requête qui renvoie les noms et prénoms des personnes qui sont des réalisateurs. La table résultat ne doit pas comporter de doublons.

```
SELECT DISTINCT personne.nom AS nom_realisateur, personne.prenom AS
    prenom_realisateur
FROM film JOIN personne ON film.idr = personne.idp
ORDER BY nom_realisateur ASC ;
```

- Écrire une requête qui renvoie les films du genre Drame avec leur titre et leur année de sortie, trié par année de sortie décroissante.

```
SELECT film.titre, film.annee
FROM film JOIN genre ON film.idg = genre.idg
WHERE genre.categorie = "Drame"
ORDER BY annee DESC ;
```

5. Écrire une requête qui renvoie le nombre de projections du film Austin Powers .

```
SELECT COUNT(*)
FROM film JOIN projection ON film.idf = projection.idf
WHERE film.titre = "Austin Powers" ;
```

6. Écrire une requête qui renvoie le nombre de projections au cinéma Comedia .

```
SELECT COUNT(*)
FROM projection JOIN cinema ON projection.idc = cinema.idc
WHERE cinema.nom = "Comedia" ;
```

Exercice Requêtes sur des tables multiples

1. Écrire une requête qui renvoie les noms et prénoms des personnes qui sont des acteurs.

```
SELECT DISTINCT personne.nom AS nom_acteur, personne.prenom AS
    prenom_acteur
FROM film JOIN jouer ON film.idf = jouer.idf
    JOIN personne ON jouer.ida = personne.idp
ORDER BY nom_acteur ASC ;
```

2. Écrire une requête qui renvoie la distribution du film *Austin Powers 2 : L'Espion qui m'a tirée* avec les noms et prénoms des acteurs et leurs rôles.

```
SELECT personne.nom AS nom_acteur, personne.prenom AS prenom_acteur,
    jouer.role
FROM film JOIN jouer ON film.idf = jouer.idf
    JOIN personne ON jouer.ida = personne.idp
WHERE film.titre LIKE "Austin Powers 2%"
ORDER BY nom_acteur ASC ;
```

3. Écrire une requête qui renvoie les films du genre Western avec leur titre , leur année et les noms et prénoms de leur réalisateur.

```
SELECT titre, annee, personne.nom AS nom_realisateur, personne.prenom AS
    prenom_realisateur, genre.categorie
FROM personne JOIN film ON personne.idp = film.idr
    JOIN genre ON film.idg = genre.idg
WHERE genre.categorie = 'Western';
```

4. Écrire une requête qui renvoie les films où Clint Eastwood a joué un rôle, avec leur titre et leur année.

```
SELECT titre, annee
FROM personne JOIN jouer ON jouer.ida = personne.idp
      JOIN film ON jouer.idf = film.idf
WHERE personne.nom = 'Eastwood' AND personne.prenom = 'Clint' ;
```

5. Écrire une requête qui renvoie les films où Clint Eastwood a joué un rôle, avec leur titre, leur année et leur réalisateur.

```
SELECT titre, annee, realisateur.nom AS nom_realisateur, realisateur.
      prenom AS prenom_realisateur
FROM personne AS acteur JOIN jouer ON acteur.idp = jouer.ida
      JOIN film ON jouer.idf = film.idf
      JOIN personne AS realisateur ON film.idr = realisateur.idp
WHERE acteur.nom = 'Eastwood';
```

6. Écrire une requête qui renvoie le nombre de films du genre Comédie projetés au cinéma Comedia .

```
SELECT COUNT(*)
FROM genre JOIN film ON genre.idg = film.idg
      JOIN projection ON film.idf = projection.idf
      JOIN cinema ON projection.idc = cinema.idc
WHERE genre.categorie = 'Comédie' AND cinema.nom = 'Comedia' ;
```

7. Écrire une requête qui renvoie les titres des films réalisés par David Cronenberg qui ont été projetés au cinéma UGC.

```
SELECT film.titre, personne.nom, personne.prenom, cinema.nom, projection
      .jour
FROM personne JOIN film ON personne.idp = film.idr
      JOIN projection ON film.idf = projection.idf
      JOIN cinema ON projection.idc = cinema.idc
WHERE personne.nom = 'Cronenberg'
AND personne.prenom = 'David'
AND cinema.nom = 'UGC';
```

8. Écrire une requête qui renvoie (sans doublons) les acteurs des films projetés au cinéma UGC depuis l'an 2000.

```
SELECT DISTINCT personne.nom, personne.prenom
FROM personne JOIN jouer ON personne.idp = jouer.ida
      JOIN film ON jouer.idf = film.idf
      JOIN projection ON film.idf = projection.idc
      JOIN cinema ON projection.idc = cinema.idc
WHERE projection.jour >= '2000-01-01'
AND cinema.nom = 'UGC' ;
```

9. Écrire une requête qui renvoie (sans doublons) les titres des films où Stellan Skarsgard a joué un rôle et qui ont été projetés au cinéma UGC.

```

SELECT DISTINCT film.titre
FROM personne JOIN jouer ON personne.idp = jouer.ida
      JOIN film ON jouer.idf = film.idf
      JOIN projection ON jouer.idf = projection.idf
      JOIN cinema ON projection.idc = cinema.idc
WHERE personne.nom = 'Skarsgard' AND personne.prenom = 'Stellan' AND
      cinema.nom = 'UGC'
ORDER BY film.titre ;

```

10. Écrire une requête qui renvoie les noms et prénoms des acteurs ayant joué dans un film réalisé par Jay Roach. La table résultat ne doit pas comporter de doublons.

```

SELECT DISTINCT acteur.nom AS nom_acteur, acteur.prenom AS prenom_acteur
FROM personne AS acteur JOIN jouer ON acteur.idp = jouer.ida
      JOIN film ON jouer.idf = film.idf
      JOIN personne AS realisateur ON film.idr = realisateur.idp
WHERE realisateur.nom = 'Roach' AND realisateur.prenom = 'Jay' ;

```

11. Écrire une requête qui renvoie les titres, noms et prénoms de réalisateur des films où le réalisateur est aussi acteur du film.

```

SELECT titre, annee, realisateur.nom AS nom_realisateur, realisateur.
      prenom AS prenom_realisateur
FROM personne AS acteur JOIN jouer ON acteur.idp = jouer.ida
      JOIN film ON jouer.idf = film.idf
      JOIN personne AS realisateur ON film.idr = realisateur.idp
WHERE acteur.nom = realisateur.nom ;

```

12. Écrire une requête qui renvoie les titres films où Clint Eastwood a joué un rôle mais qu'il n'a pas réalisé, avec leur titre et leur année.

```

SELECT titre, annee, realisateur.nom AS nom_realisateur, realisateur.
      prenom AS prenom_realisateur
FROM personne AS acteur JOIN jouer ON acteur.idp = jouer.ida
      JOIN film ON jouer.idf = film.idf
      JOIN personne AS realisateur ON film.idr = realisateur.idp
WHERE acteur.nom = 'Eastwood' AND realisateur.nom <> 'Eastwood' ;

```

Exercice Insertion, modification, suppression, contraintes d'intégrité

- On ne peut enregistrer dans cette base de données un film avec plusieurs réalisateurs. En effet dans la relation **film**, un film ne peut être enregistré que dans un seul nuplet, l'attribut **idf** étant une clef primaire. De plus l'attribut **idr** est une clef étrangère référençant la clef primaire **idp** de la relation **personne** c'est-à-dire qu'une seule personne peut être associée à un film comme réalisateur. Si on veut définir plusieurs réalisateurs pour un film il faudrait créer, sur le modèle de la relation **jouer**, une relation **realise** avec deux clefs étrangères référençant la clef primaire **personne(idp)** et la clef primaire **film(idf)**.

2. Pour insérer dans la base le rôle d'Edie Stall joué par l'actrice Maria Bello dans le film *A history of violence*, il faut d'abord insérer Maria Bello dans la table **personne** avec :

```
INSERT INTO "personne" ("idp","nom","prenom") VALUES (33, 'Bello', 'Maria');
```

Attention à la valeur de l'attribut **idp** (clef primaire) qui doit être unique dans la relation.

Ensuite, on insère son rôle dans la table **jouer** avec :

```
INSERT INTO "jouer" ("ida","idf","role") VALUES (33, 18,"Edie Stall");
```

Attention aux valeurs des attributs des clefs étrangères **ida** (référéncant la clef primaire **personne(idp)**) et **idf** (référéncant la clef primaire **film(idf)**), qui doivent correspondre à des valeurs existantes (et cohérentes avec le rôle) dans les tables référencées.

3. Le film *Le bon, la brute et le truand* est sorti en Italie en 1966 mais en France seulement en 1968. Quelle instruction SQL permet de modifier l'année de sortie de ce film dans la table **film** ?

```
UPDATE film
SET annee = 1966
WHERE titre = "Le bon, la brute et le truand";
```

4. Si on veut supprimer le film "Le dictateur" de la base, l'instruction **DELETE FROM film WHERE titre = "Le Dictateur"** ne suffit pas. Le SGBD affiche le message d'erreur **FOREIGN KEY constraint failed**. En effet, la clef primaire **idf** de la relation **film** est référencée par une clef étrangère de la table **jouer** et une clef étrangère de la table **projection**. Il faudrait donc d'abord supprimer toutes les projections du film "Le Dictateur" dans la table **projection** et tous les rôles associés à ce film dans la table **jouer**.

Il faut donc exécuter les suppressions dans l'ordre :

```
DELETE FROM projection WHERE idf = (SELECT idf FROM film WHERE titre = "Le Dictateur") ;

DELETE FROM jouer WHERE idf = (SELECT idf FROM film WHERE titre = "Le Dictateur") ;

DELETE FROM film WHERE titre = "Le Dictateur" ;
```