



Soit une base de données de film modélisée par le schéma relationnel suivant :

FILMS (id, titre, annee, score, nbvotant, #idrealisateur)

ACTEURS (id, prenom, nom)

JOUER (#idfilm, #idacteur, rang)

REALISATEURS (id, prenom, nom)

Exercice 1: Rechercher un film

1. Ecrire un programme JAVA qui demande à l'utilisateur de saisir une chaîne de caractère **ch** et qui affiche tous les films qui contiennent la chaîne **ch** dans leur titre.
2. Ecrire un programme JAVA qui demande à l'utilisateur de saisir un acteur par son et son prénom et qui affiche tous les films dans lesquels il a joué.
3. Ecrire un programme JAVA qui demande à l'utilisateur de saisir un entier N et qui affiche tous les films dans lesquels ont joué plus de N acteurs.
4. Ecrire un programme qui demande à l'utilisateur de saisir le nom et le prénom d'un nouvel acteur et l'ajoute à la base de données.

Exercice 2: Fonctions Transact-SQL dans JAVA

1. En utilisant SQL Server Management Studio, créer une fonction **nbFilmsSortisEn** qui prend en paramètre une année et qui renvoie le nombre de films sortis cette année.
2. Ecrire un programme JAVA qui demande à l'utilisateur de saisir une année et qui affiche le nombre de films sortis l'année saisie par l'utilisateur.

Exercice 3: Fonctions T-SQL VS JAVA ??

1. En utilisant SQL Server Management Studio, écrire la fonction **NbFilmActeur** qui prend en paramètre le nom et le prénom d'un acteur et renvoie le nombre de films dans lesquels il a joué.
2. Ecrire cette même fonction, cette fois dans une méthode JAVA.
3. Utiliser l'une et l'autre dans un programme JAVA pour affiche le nombre de film joué par un acteur saisi par l'utilisateur.