**SQL – EXERCICES (2)**

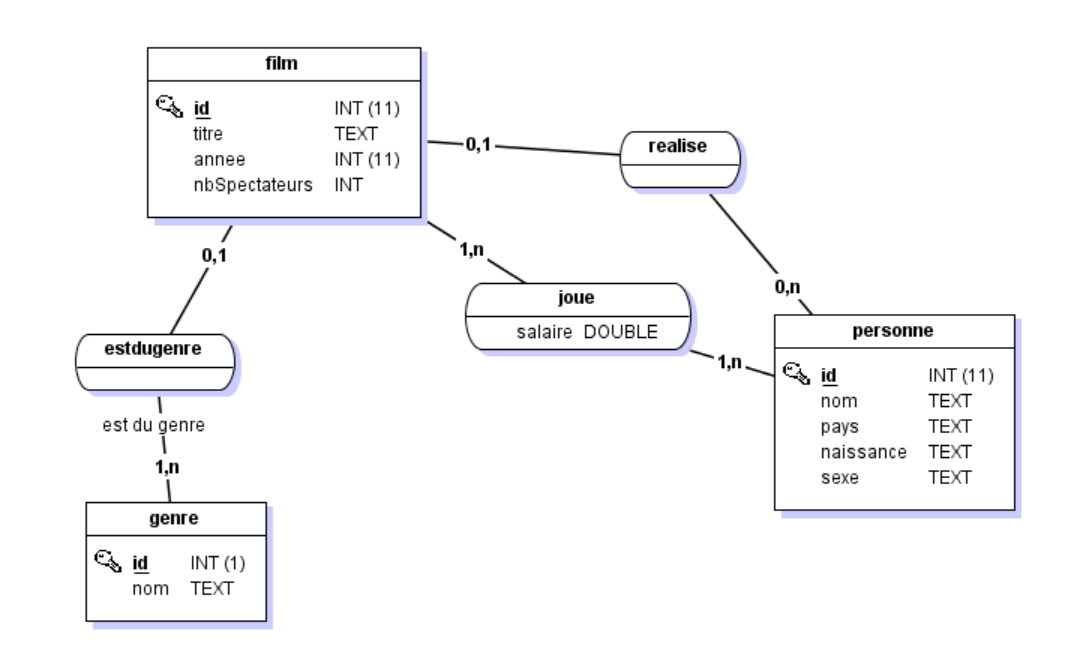
08

**Automne**

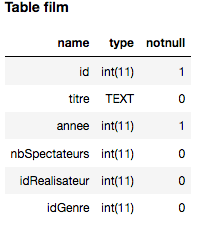
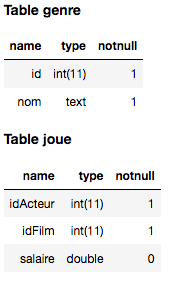
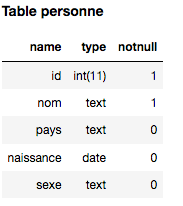
Dans cette série d´exercices, on utilise le logiciel **DB Browser** pour effectuer toutes les requêtes.

1. **LES FILMS**

La base de données sur les films est issue du *MCD* suivant:

****

Elle donne naissance aux tables du *modèle relationnel:*

**  **

*Ecrivons les requêtes SQL qui permettent d´afficher:*

**Requêtes simples**

1) Dix noms de personnes (réalisateurs ou acteurs).  
  
**SELECT nom FROM personne LIMIT 10***;*

#### 2) Liste des titres des films commençant par la lettre M et se terminant par la lettre r ou s. SELECT titre FROM film WHERE titre LIKE('M%') AND (titre LIKE('%r') OR titre LIKE('%s') );

3) Liste des titres des films sortis entre 2002 et 2004 (sur les 3 années).  
  
**SELECT titre FROM film WHERE annee BETWEEN 2002 AND 2004;**

BETWEEN est une clause inclusive

4) Liste des noms et dates de naissance des personnes connues de la base dont le prénom commence par 'Ro' et de nationalité Française (le pays vaut 'France'), par ordre alphabétique.

**SELECT nom, naissance FROM personne WHERE nom LIKE('Ro%') AND pays='France' ORDER BY nom ASC;**L'ordre est croissant par défaut, la clause ASC est donc optionnelle.

5) Les pays (une fois chacun) dont ont connaît au moins une personne de sexe féminin (donc une réalisatrice ou une actrice).

**SELECT DISTINCT pays FROM personne WHERE sexe='F';***Il faut d´abord regarder dans la table 'personne' comme est codé le sexe des personnes* .

**JOINTURES**

6) Liste des films sortis en 2006 en indiquant le titre, et le genre (en texte), par ordre alphabétique des titres.

**SELECT titre, genre.nom FROM film   
INNER JOIN genre   
ON film.idGenre=genre.id   
WHERE annee=2006  
ORDER BY titre;**  
  
7) Liste des drames sortis strictement **avant 1970** en donnant le titre et l'année de sortie.

**SELECT titre, annee FROM film   
INNER JOIN genre   
ON film.idGenre=genre.id   
WHERE genre.nom='Drame' AND annee<1970;**

8) Liste alphabétique des acteurs du film "Le Convoyeur".

**SELECT nom FROM personne   
INNER JOIN joue  
ON personne.id=joue.idActeur   
INNER JOIN film ON film.id=joue.idFilm   
WHERE film.titre='Le Convoyeur'   
ORDER BY nom;***On fait ici une double jointure entre les tables 'personne', 'joue' et ' film'.*

**AGRÉGATIONS**

10) Table contenant seulement l'année de sortie du plus ancien film, et l'année de sortie du film le plus récent.

**SELECT Min(annee),MAX(annee) FROM film;**

11) Moyenne du nombre de spectateurs par film.

**SELECT AVG(nbSpectateurs) FROM film;**

12) Liste des 10 films comptant le plus d'acteurs.

**SELECT titre, nbSpectateurs FROM film ORDER BY nbSpectateurs DESC LIMIT 10;**

**SOUS-REQUÊTES**

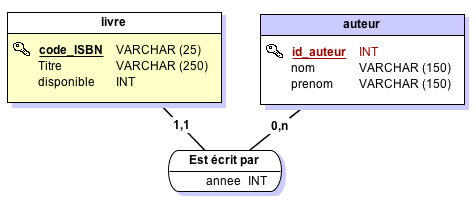
13) Liste des films (avec leur année de sortie) sortis **au moins 50 ans** après le film le plus ancien de la base.

**SELECT titre, annee FROM film WHERE annee>=(SELECT MIN(annee) FROM film)+50;***La sous requête doit être entre parenthèses.*

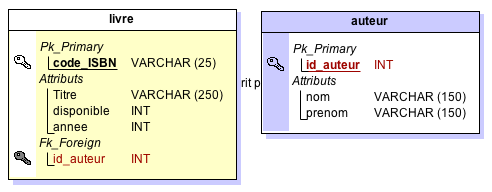
1. **CReation des tables d´une base de données**

Nous allons créer les tables de la base de donnée simplifiée pour le CDI décrite dans le premier chapitre.

1) Reproduire dans un premier temps le MCD dans le logiciel **JMerise**:



2) Validons maintenant le MCD pour générer le modèle relationnel. Les tables sont créées ainsi que le code SQL.



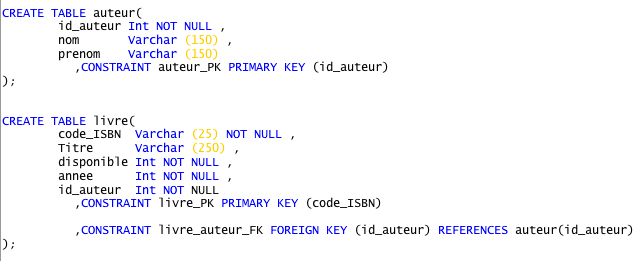
Syntaxe de création:

CREATE TABLE table\_name (  
    colonne1 datatype,  
    colonne2 datatype,  
    colonne3 datatype,  
   ....  
);

On ajoute si besoin PRIMARY KEY (colonne) ou FOREIGN KEY (colonne) REFERENCES table\_name (colonne) en fin d´instructions.

Les instructions soulignées ne sont pas nécessaires.

🖙 Commentons les lignes de code intéressantes.



3) Copiez et collez ces lignes dans **DB Browser** et vérifiez que les tables sont bien créées.

4) Ajoutez maintenant les enregistrements donnés en page suivante. Vérifiez ensuite le contenu des tables avec quelques requêtes simples.

*On commence par insérer des enregistrements dans la table sans clé étrangère. La syntaxe d´insertion est du type:   
INSERT INTO table (colonne1, colonne2, ...) VALUES (valeur1, valeur2, ...);***INSERT INTO auteur VALUES (1,'Stendhal','Henri'),(2,'Hugo','Victor'),(3,'Pessoa','Fernando');**

*Remarquons que l´on peut insérer les trois enregistrements en une seule ligne de code et que le nom des colonnes n´est pas nécessaire si on remplit tous les champs.*

*On insère ensuite les enregistrements de l´autre table:*

**INSERT INTO livre VALUES ('12342','Le Rouge et le Noir',1,1830,1),  
('97865','Mensagem',1,1934,3),
  
('36548','La Chartreuse de Parme',0,1839,1),  
('98734','Les Misérables',1,1862,2);**

Un élève se présente au CDI et veut savoir quels sont les livres de Stendhal disponibles:

**SELECT Titre, disponible FROM livre  
INNER JOIN auteur  
ON livre.id\_auteur=auteur.id\_auteur   
WHERE nom='Stendhal' AND disponible=1;**

Table **auteur**

|  |  |  |
| --- | --- | --- |
| **id\_auteur** | **nom** | **prenom** |
| 1 | Stendhal | Henri |
| 2 | Hugo | Victor |
| 3 | Pessoa | Fernando |

Table **livre**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **code\_ISBN** | **Titre** | **disponible\*** | **annee** | **id\_auteur** |
| 12342 | Le Rouge et le Noir | 1 | 1830 | 1 |
| 97865 | Mensagem | 1 | 1934 | 3 |
| 36548 | La Chartreuse de Parme | 0 | 1839 | 1 |
| 98734 | Les Misérables | 1 | 1862 | 2 |

\* 1 si le livre est disponible, 0 sinon.