

Labo5 : JDBC

Objectifs

L'objectif de ce laboratoire est de vous familiariser avec l'utilisation d'une base de données au sein d'une application Java. Plus précisément, vous allez utiliser l'API Java Database Connectivity (JDBC), qui permet aux applications Java d'accéder à une base de données.

Démarche à suivre

Téléchargez le driver JDBC pour MySQL via <http://dev.mysql.com/downloads/connector/j/> en choisissant votre plateforme.

Créez un projet Java avec Eclipse intitulé BDR_Labo5. Ensuite, ajoutez le fichier `mysql-connector-java-5.1.34-bin.jar` dans les librairies du projet BDR_Labo5 via `Properties>Java Build Path>Libraries>Add External JARs...`

Nous avons fourni un exemple d'utilisation de driver JDBC dans le fichier `JDBCTest.java`. Ajoutez ce fichier à votre projet. Il faut adapter le *numéro de port*, le *nom de l'utilisateur* et le *mot de passe*. Utilisez cette classe comme un template pour réaliser votre code.

Consultez l'API de `java.sql` sur <http://docs.oracle.com/javase/7/docs/api/java/sql/package-summary.html>.

Le laboratoire est à rendre le **mercredi 24 décembre avant minuit**.

Réalisation

1. Ecrivez une requête permettant de récupérer tous les champs des 10 premières lignes de la table `film` dans un objet `ResultSet` et afficher le résultat sur la console.
2. Ecrivez un programme permettant d'insérer dans la table `actor`, les noms et les prénoms de vos dix acteurs préférés qui n'existent pas dans la base de données. Faites les insertions en utilisant le `PreparedStatement`.
3. Ecrivez un programme permettant à un membre de staff du magasin de location de film d'effectuer la location des films pour un client.

Pour cela, vous allez procéder par étapes :

- Vous commencerez par écrire un programme Java qui permet de saisir le numéro de staff.
- Pour effectuer une location le client doit spécifier le nom d'un acteur
- Le programme affiche la liste des acteurs ayant ce nom. Le résultat devra être affiché sous la forme présentée ci-dessous (une liste de lignes composées d'un numéro suivi du prénom et du nom de l'acteur ou actrice). Ici les entrées du programme sont marquées en gras :

//Exemple de l'exécution pour la question 3 :

Entrez votre numéro de staff:

1

Entrez le nom d'un acteur :

NOLTE

Choisissez parmi les acteurs suivants :

1 ALBERT NOLTE

2 JAYNE NOLTE

3 SALMA NOLTE

4 WARREN NOLTE

Votre choix :

3

Vous avez choisi : SALMA NOLTE

...

- Complétez ce programme en permettant à l'utilisateur de choisir l'un de ces acteurs, en indiquant le numéro de la ligne retourné à l'étape précédente. Vous afficherez alors la liste des titres des films dans lesquels l'acteur a joué. De nouveau, présentez une liste numérotée.
- Une fois on a saisi le numéro du titre, le programme doit afficher les numéros d'exemplaire de ce film disponibles dans ce magasin.
- Une fois on a saisi le numéro d'exemplaire, le programme doit demander le nom et le prénom du client.
- Le client doit effectuer le paiement pour pouvoir effectuer la location du film choisi.
- Le nombre maximal des locations qu'un client peut posséder à la fois est 10.
- Si le client n'existe pas il faut l'ajouter dans la base avant d'effectuer la location. Pour cela vous allez demander les coordonnées du client et ajouter au besoin.

Utiliser les transactions quand c'est nécessaire.

4. Ecrivez une fonction stockée qui calcule le nombre total de films dans une catégorie donnée (le nom de catégorie est passé en paramètre). Ecrivez un programme Java qui utilise la procédure stockée pour retourner le nombre des films correspondant à un nom de catégorie donné. Complétez le programme pour afficher les résultats pour toutes les catégories de films dans la base.
5. Ecrivez un programme java qui crée une vue nommée *revenu_par_localite* qui contient la somme total des paiements des clients regroupée par localité : le code postal, la ville et le pays de clients. Le programme utilise ensuite cette vue pour afficher la somme total des paiements de clients en fonction d'un nom de pays donné en entrée.