

TP JAVA JDBC

Fabien Coelho & Pierre Guillou

https://www.cri.enscm.fr/people/coelho/tp_jdbc.html

1 Driver JDBC

Pour se connecter à PostgreSQL on utilisera le *driver* nommé `postgresql` fournit par la classe `org.postgresql.Driver` du fichier `jar /usr/share/java/postgresql.jar`.

Interface graphique Eclipse : il est nécessaire d'ajouter le jar explicitement. Suivre *Projet, Build Path, Add External Archives...* puis coller le nom complet du fichier. Par ailleurs, il faudra aussi changer la configuration de lancement du programme pour lui passer des arguments.

Mode terminal : il faut ajouter le jar à la variable d'environnement `CLASSPATH`. Pour cela, exécuter dans le terminal :

```
export CLASSPATH=/usr/share/java/postgresql.jar:$CLASSPATH
```

2 Interface texte simple

L'idée est de développer un client texte interactif à la manière de la commande `psql` de Postgres. En vous basant sur le squelette `RequeteSQL.java`, faites une interface texte très simple en JAVA, qui attend une requête `SELECT` sur une ligne de l'entrée standard, la soumet à la base, et affiche le résultat en colonnes séparées par des tabulations. On pensera bien sûr à afficher les entêtes des colonnes.

Dans un second temps, adapter le programme pour traiter d'autres types de requêtes `INSERT` `UPDATE` `DELETE` puis les commandes `CREATE` `DROP`...

Dans un troisième temps, ajouter une commande `\q` pour quitter l'interface texte, et une commande `\d` pour décrire une relation (nombre d'attributs, noms et types des attributs).

Ajouter enfin le support des transactions explicites avec `BEGIN` `COMMIT` `ROLLBACK` en vous appuyant sur les fonctionnalités de la connexion. Réfléchir au traitement des erreurs en cas d'exception au cours d'une transaction.

```
shell> java RequeteSQL jdbc:postgresql://pagode/comics calvin
pass: ...
sql> SELECT * FROM auteur;
titre          annee
Jean-Jean la Terreur    2001
Le Geant qui Pleure    2001
Politique Etrangere    2000
Heracles             2002
...
sql> UPDATE langue SET ... WHERE ...;
```

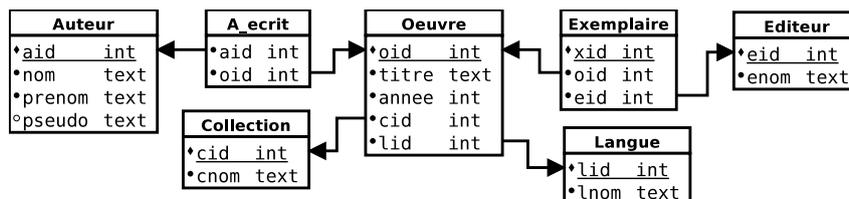


FIGURE 1 – comics database

3 Interface graphique d’interrogation

On souhaite développer une interface graphique pour ajouter simplement des albums à une base de donnée sur les bandes dessinées, présentée à la figure 1.

L’armature de l’application graphique en AWT vous est proposée, et la base contient déjà des fonctions qui implémentent les fonctionnalités (ajout d’un album, ajout d’un auteur supplémentaire à un album que l’on vient d’insérer).

Il vous faut juste compléter les aspects JDBC. Les classes suivantes sont fournies :

Titre	Nouveau titre
Annee	2017
Editeur	Albin Michel
Collection	Achille Talon
Langue	fr_FR
Auteur	Yoann Chivard (Yoann)
Ajouter	
Auteurs...	
Quitter	
status	*

dialogue AWT DatabaseChooser.java

Il permet de demander les informations concernant la connexion à la base de donnée. Différentes méthodes permettent de récupérer les informations, voir le `main` de test à la fin.

Vous n’avez **rien** à compléter dans cette classe !

composant graphique de choix ChoiceId.java

Il s’agit d’un composant graphique qui permet de choisir un élément dans une liste présentée sous forme de menu déroulant. Un numéro est associé au texte affiché, par exemple le numéro de l’auteur à partir de son nom.

Vous n’avez **rien** à compléter dans cette classe !

interface graphique Comics.java

Vous devez compléter le `main` afin de construire une `Connection` à partir des informations fournies par l’utilisateur via le `DatabaseChooser`.

connexion à la base de donnée ComicsDB.java

Vous devez compléter le constructeur pour initialiser le `statement` et les `prepared statement`.

Vous devez ensuite compléter la méthode `getSomething` pour construire un `ChoiceId` à partir d’une requête retournant deux colonnes.

Vous devez également compléter les méthodes `get...` en précisant les différentes requêtes SQL pour récupérer les informations souhaitées.

Vous devez enfin compléter les méthodes `add...` pour invoquer les `prepared statements`. Cette dernière partie peut être faite dans un second temps.