Institut National Universitaire Champollion

L3 Informatique 2023

UE Bases de Données 3 : TP2

suite du TP1 :

connexion et interaction en Java à une BDD PostgreSQL

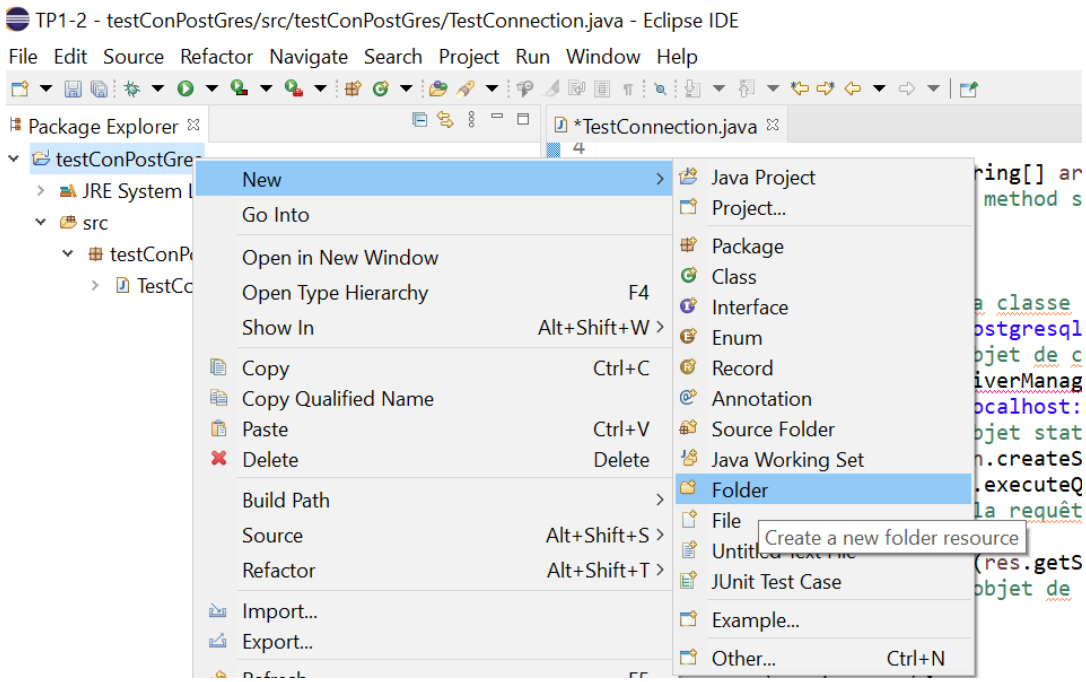
L'objectif de ce TP est de vous apprendre à vous connecter à une BDD Postgres en Java et à interagir avec.

Partie 1 : connexion Java – Postgres

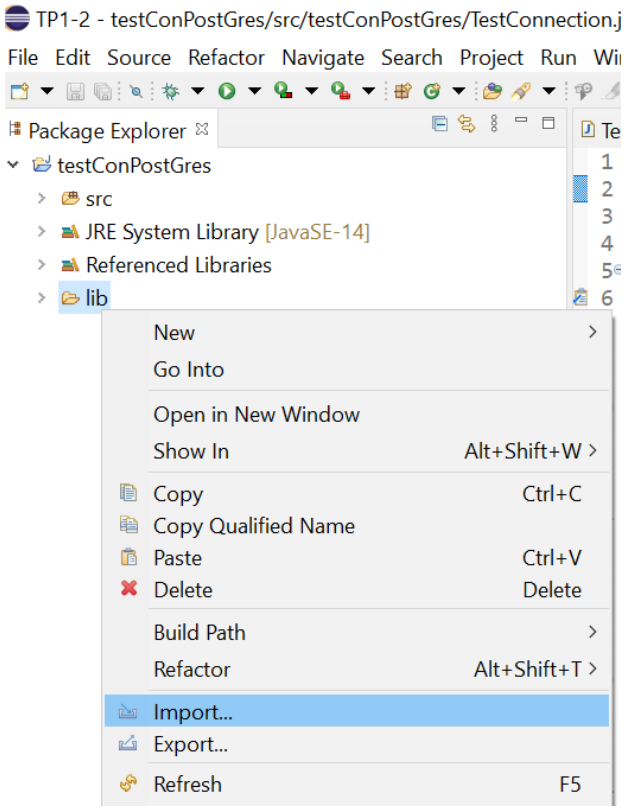
A – import du fichier de connexion

1- Créez un projet Eclipse avec la classe TestConnexion qui vous est fournie.

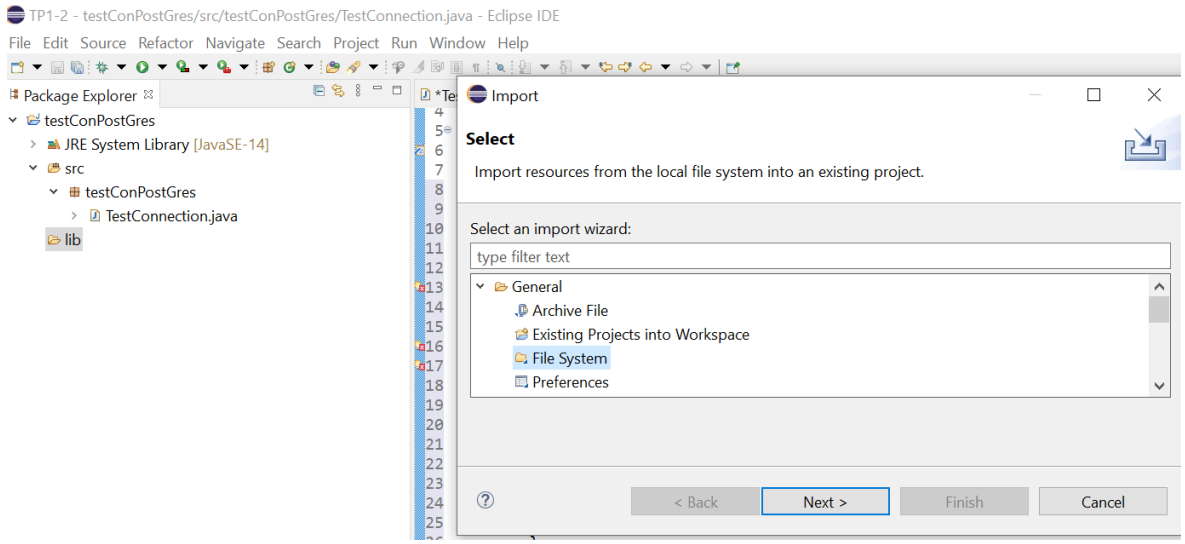
2- cliquez droit sur votre projet et créez un répertoire (nommé lib par exemple)



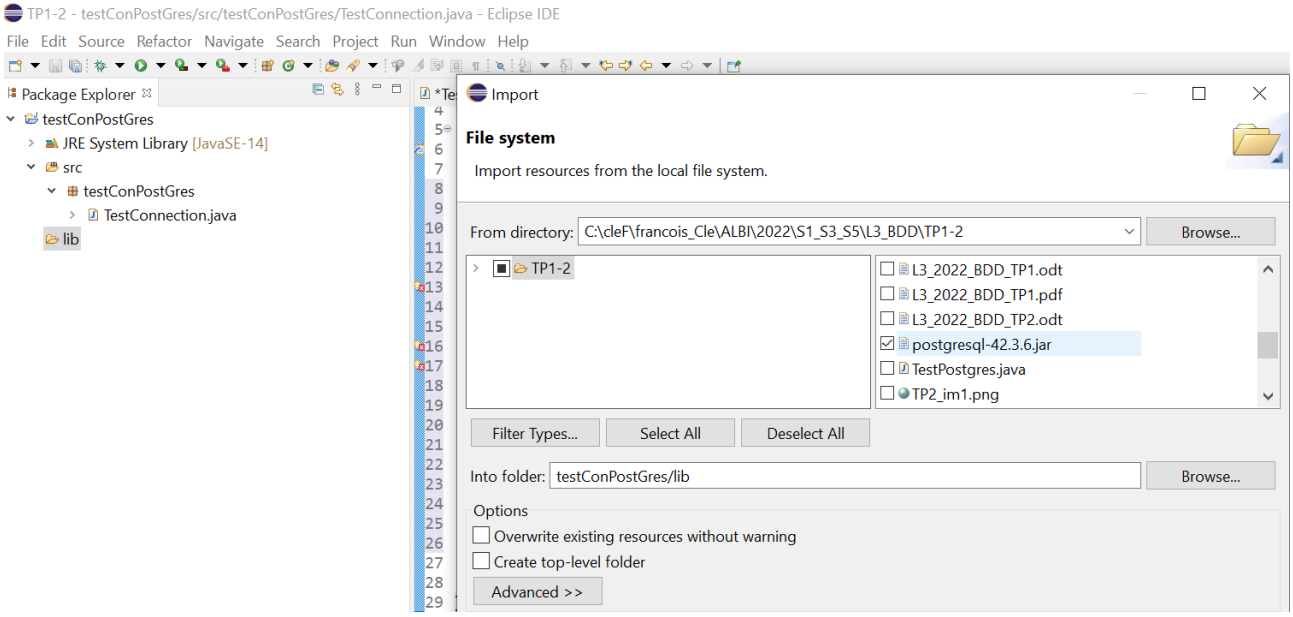
3- cliquez droit sur ce dossier et puis « import »



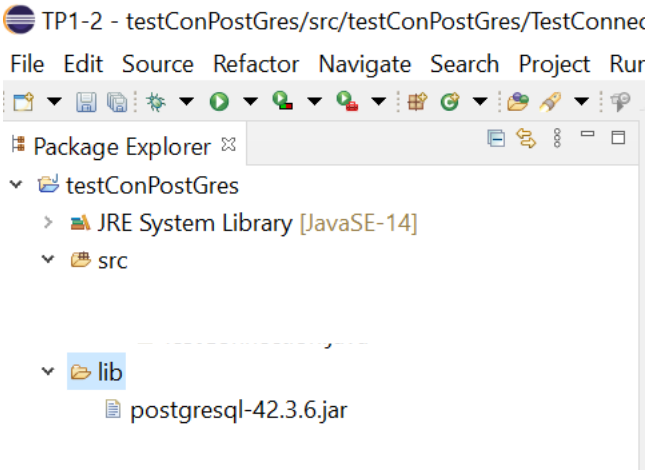
4- Dans le menu import, sélectionnez File System, puis cliquez sur « next »



5- déplacez vous dans le répertoire ou vous avez stocké le fichier postgresql-43.3.6 .jar, et sélectionnez-le.

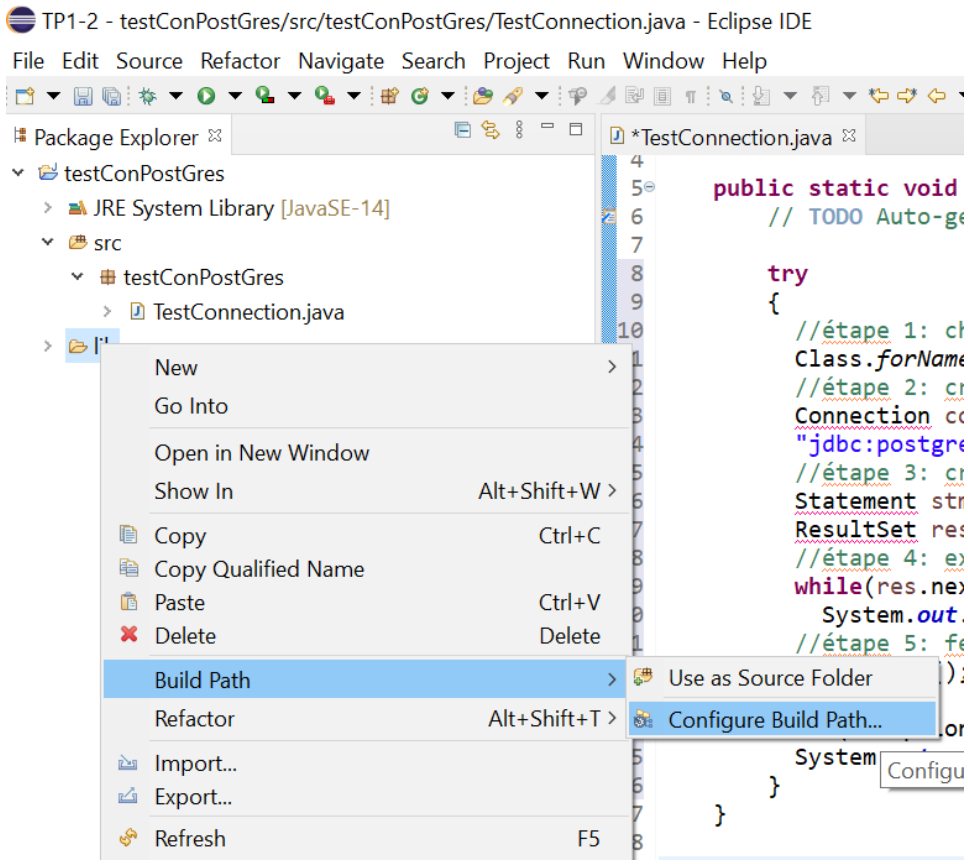


vous pouvez voir ce fichier maintenant dans le répertoire lib :

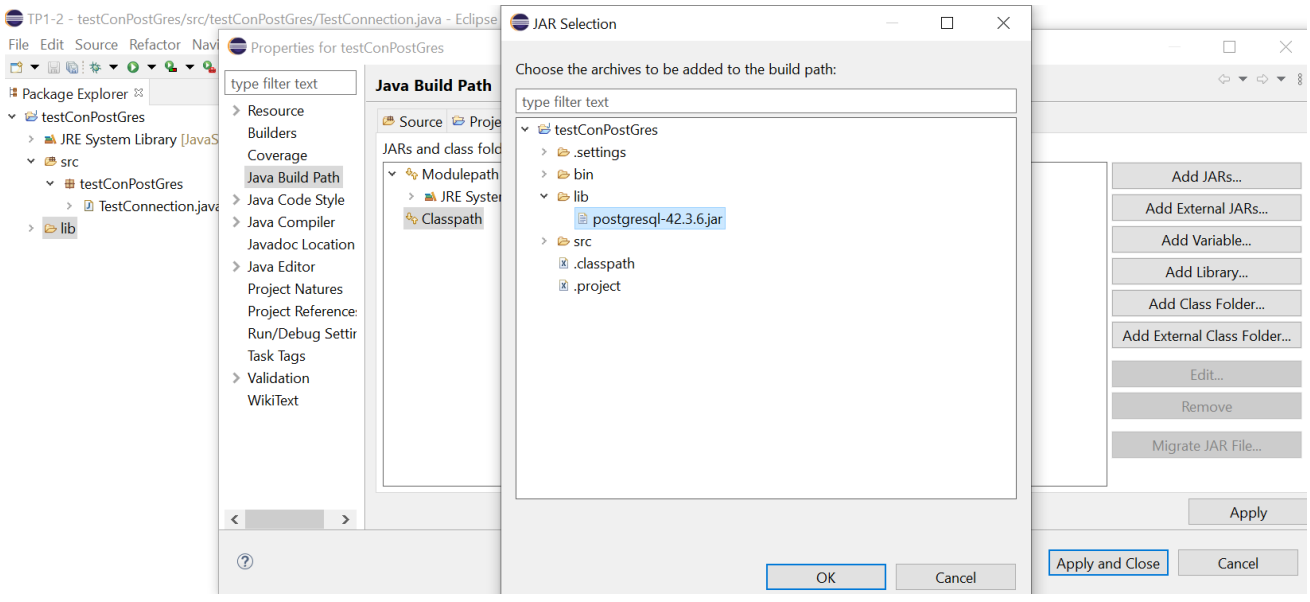


B - construction du path vers ce fichier

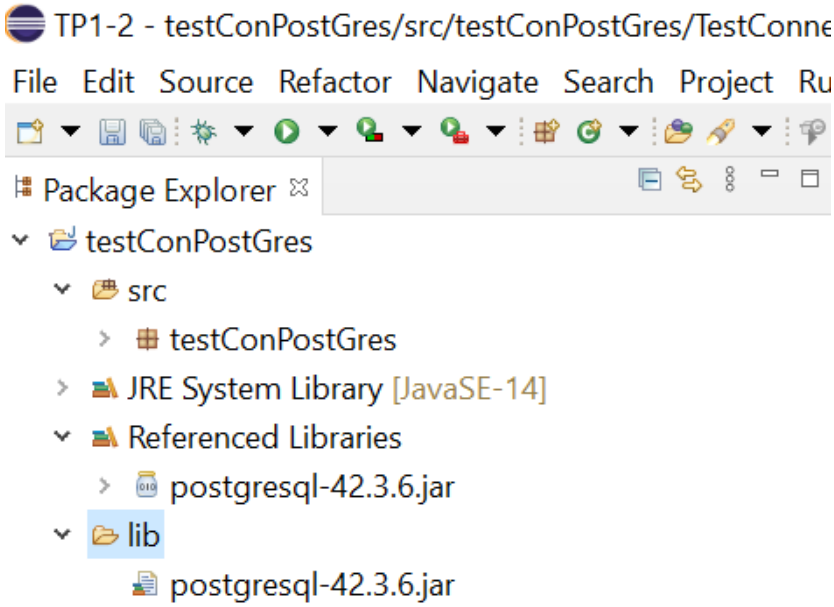
1 – Cliquez droit sur le dossier lib et sélectionnez « Build path » puis « Configure Build Path »



2 – Dans l'onglet *Libraries* sélectionnez ClassPath, puis cliquez sur Add Jars, puis sélectionnez le fichier postgresql-43.3.6 .jar qui se trouve dans votre répertoire lib.



Vous pouvez voir maintenant dans le répertoire Referenced Libraries le fichier postgresql-43.3.6 .jar



C - Test de la connexion

Vous pouvez maintenant exécuter votre classe TestConnexion, prenez soin de préciser correctement le nom de la base, votre login et mot de passe de connexion dans la ligne « étape 2 ».

Partie 2 :
Création d’une application Java utilisant une BDD PostgreSQL.

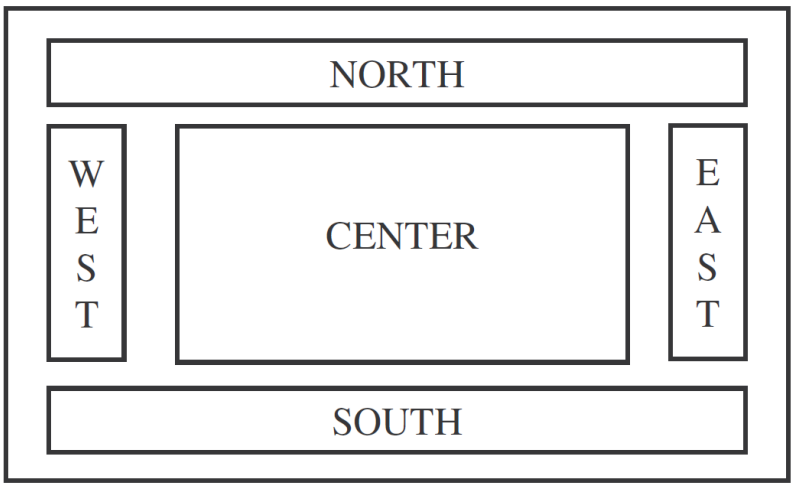
Vous allez dans cette partie développer une application Java interagissant avec la base de données du TP1.
Si nécessaire vous pouvez utiliser le script de création des tables, ainsi que le script permettant de vider les tables pour les tests. Sinon, vous pouvez utiliser directement votre BDD créée lors du TP1.
Il va falloir créer des fenêtres avec des boutons, des zones de texte pour saisir les données et des listes déroulantes pour choisir des données existantes.
Pour cela vous allez réinvestir ce que vous avez appris en programmation évènementielle l’année dernière.
Le cours de l’année dernière est mis à votre disposition.

Etape 1

Pour démarrer on va mettre en place une fenêtre de saisie permettant d’insérer un éditeur dans la table **éditeur**.
Pour cela, on vous fournit trois classes :

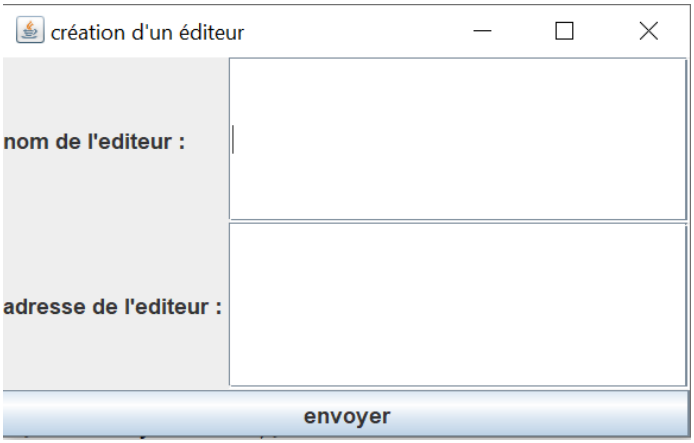
La classe **FenetreEditeur** dont la tâche est de créer les zones de saisie de texte (JTextField), les étiquettes associées (Jlabel), un bouton (Jbutton) pour envoyer les données dans la table éditeur via la classe EcouteurBoutonEditeur.
Vous remarquerez que pour placer ces éléments dans la fenêtre, on utilise deux gestionnaires de placement. Le gestionnaire BorderLayout qui partage la fenêtre en cinq parties comme indiqué dans la figure suivante :

gestionnaire BorderLayout



et le gestionnaire GridLayout, qui créé une grille n lignes et p colonnes permettant de mettre les éléments dans les cases.
On construit donc deux « panneaux » (JPanel) (associés à des gestionnaires de type GridLayout à deux lignes et une colonne), dans lesquels on mettra dans l'un deux étiquettes, et dans l'autre deux zones de texte. On les placera ensuite dans la zone WEST pour le premier et CENTER pour le second. Ainsi les zones de texte et les étiquettes seront alignées.
Le bouton lui, sera placé dans la zone SOUTH.

Cela donne la fenêtre suivante :



La classe **EcouteurBoutonEditeur** dont la tâche est de réagir au clic sur le bouton : c'est à dire d'insérer un t-uple correspondant aux données présentes dans les champs de texte dans la table éditeur.

La classe **FenetreErreur** qui permet d'afficher un message d'erreur en cas de saisie erronée de la part de l'utilisateur. Vous remarquerez que cette classe est une sous classe de la classe JFrame et implémente aussi l'interface MouseListener, ainsi il est possible d'intégrer la réaction au clic sur le bouton à cette même classe.

- Testez ce programme en insérant un t-uple dans éditeur.
- Essayez d'insérer un t-uple en ne respectant pas la contrainte de clé primaire : essayez d'insérer un t-uple avec un nom d'éditeur existant déjà. Que se passe t-il ? Modifiez les classes FenetreErreur et EcouteurBoutonEditeur pour que l'utilisateur ait le message adéquat et puisse corriger. Vous pouvez utiliser la méthode getMessage() (contenant le nom de la contrainte non respectée) sur l'objet de type SQLException pour personnaliser le message affiché qui sera plus adapté pour un utilisateur quelconque.

- Essayez d’insérer un t-uple en ne respectant pas la contrainte de clé primaire : essayez d’insérer un t-uple sans nom d’éditeur. Que se passe t-il ? Modifiez les classes FenetreErreur et EcouteurBoutonEditeur pour que l’utilisateur ait le message adéquat et puisse corriger.
- Essayez d’insérer un t-uple en omettant l’adresse de l’éditeur (contrainte NOT NULL dans la base) Que se passe t-il ? Comme précédemment modifiez le code pour que l’utilisateur puisse corriger.

Etape 2

Utilisez la même démarche pour insérer un t-uple dans la table **Client**.
Pour cela vous ferez une classe FenetreClient en vous inspirant du modèle FenetreEditeur, mais en y intégrant comme pour FenetreErreur la réaction au clic sur le bouton : le codage est plus simple.
Vous ferez ce qu’il faut pour la contrainte d’unicité et de non nullité de l’email du client.
La clé primaire n’est pas à renseigner : utilisez la séquence cleClient, ou l’auto incrémentation.

Etape 3

Utilisez la même démarche pour insérer un t-uple dans la table **Auteur**.
Vous remarquerez que on ne peut créer un Auteur sans y associer au moins un livre et inversement : contraintes qu’on ne peut prendre en compte directement au remplissage des tables. Ces contraintes sont donc vérifiées à posteriori à l’aide des vues crées à cet effet. On créera donc un auteur sans y associer de suite un livre.

En revanche on profitera de cette fenêtre pour insérer un ou plusieurs t-uples dans la table **contrat**, en proposant à l’utilisateur deux champs pour proposer les éditeurs chez qui l’auteur a un contrat. Pour cela vous utiliserez un menu déroulant proposant la liste des éditeurs existant dans la table éditeur.
(utilisation de la classe JcomboBox, avec le constructeur prenant en paramètre un Vector, cet objet Vector sera rempli avec les éditeurs possibles. Vous aurez donc besoin de faire une requête pour obtenir les éditeurs de la table éditeur)

Essayez de mettre un contrat entre un auteur et un éditeur de mêmes noms. Que se passe t-il ? Le t-uple dans la table contrat est il inséré ? Est-il possible d'informer l'utilisateur sans recoder le trigger dans l'application ?

Etape 4

Utilisez la même démarche pour insérer un t-uple dans la table **Livre**.
Pour le type du livre vous utiliserez un menu déroulant proposant soit « POCHE » soit «BD ».
Cette fois – ci on associera un auteur parmi ceux existant dans la table Auteur encore à l'aide d'un menu déroulant : il faudra exécuter une requête sur la table Auteur pour connaître la liste des auteurs à proposer. On proposera bien dans cette liste déroulante les noms et prénoms des auteurs (proposer un identifiant numérique n'est pas explicite pour l'utilisateur), par contre ce sera bien l'identifiant de l'auteur qui sera utilisé.
Ce faisant, on suppose que les auteurs ont tous des noms et prénoms différents ...
On proposera au maximum 2 auteurs (on peut faire plus bien sûr).
Ainsi cette fenêtre aura pour action à la fois d'insérer un t-uple dans la table auteur mais aussi de 1 ou 2 tuples dans la table **aecrit**.

Attention à la contrainte de cycle : dans un premier temps, on ne proposera pas dans cette fenêtre de proposer un éditeur, car cela doit dépendre du ou des auteurs proposés. Ainsi lorsque l'utilisateur a cliqué sur le bouton « envoyer », une nouvelle fenêtre est mise en place pour proposer un éditeur à ce livre parmi ceux possibles (ceux chez qui le ou les auteurs ont un contrat)

Etape 5

Utilisez la même démarche pour insérer un t-uple dans la table **Exemplaire**.
Si un Client est choisi alors la date correspondante sera la date du jour par défaut.

Etape 6

Faites enfin une fenêtre proposant via des boutons les insertions dans les différentes table. Vous fermerez l'application à l'aide d'un bouton.