**TP1 (JDBC-DAO)**

**JDBC**

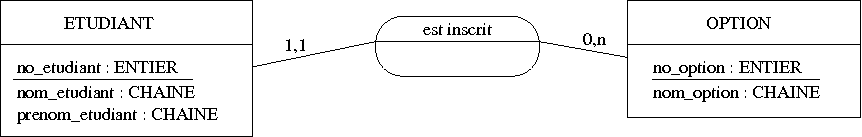
La technologie JDBC (*Java DataBase Connectivity*) est une API fournie avec Java (depuis sa version 1.1) permettant de se connecter à des bases de données, c'est-à-dire que JDBC constitue un ensemble de classes permettant de développer des applications capables de se connecter à des serveurs de bases de données ([SGBD](http://www.commentcamarche.net/contents/bdd/bddintro.php3)).

L'API JDBC a été développée de telle façon à permettre à un programme de se connecter à n'importe quelle base de données en utilisant la même syntaxe, c'est-à-dire que l'API JDBC est indépendante du SGBD.

De plus, JDBC bénéficie des avantages de Java, dont la portabilité du code, ce qui lui vaut en plus d'être indépendant de la base de données d'être indépendant de la plate-forme sur laquelle elle s'exécute.

Exemple Suivie :

Voici un MCD représentant la relation entre les étudiants et les options (Informatique, Télécom….) :



On veut créer une petite application java qui permet de gérer ceci (ajout, suppression, modification des étudiants et des options sans oublier l’affectation des options aux étudiants) dans une base de données MySQL, pour ce faire, on doit commencer par créer la base et son schéma.

1. Création de la base :

|  |  |
| --- | --- |
|  | En utilisant MySQLAdministrator :  -Catalogue🡪 « create new schema »  - insertion du nom de la base exemple (« esprit »). |

1. Utilisateurs et droits:

|  |  |
| --- | --- |
|  | User administration🡪  »add new user »  Nom utilisateur : « esprituser »  Mot de pass : « espritpwd »  Ensuite en bas a droite🡪 « Apply changes »  Puis clic droit sur l’utilisateur esprituser 🡪 « add hosts from wich the user can connect »  Nom du host : « localhost ». |

|  |  |
| --- | --- |
|  | Sélectionner « localhost » (sous « esprituser ») puis l’onglet «Schema privilège ».  Choisir la base de donnee « esprit » puis assigner tous les droits.  Enfin « Apply changes » |

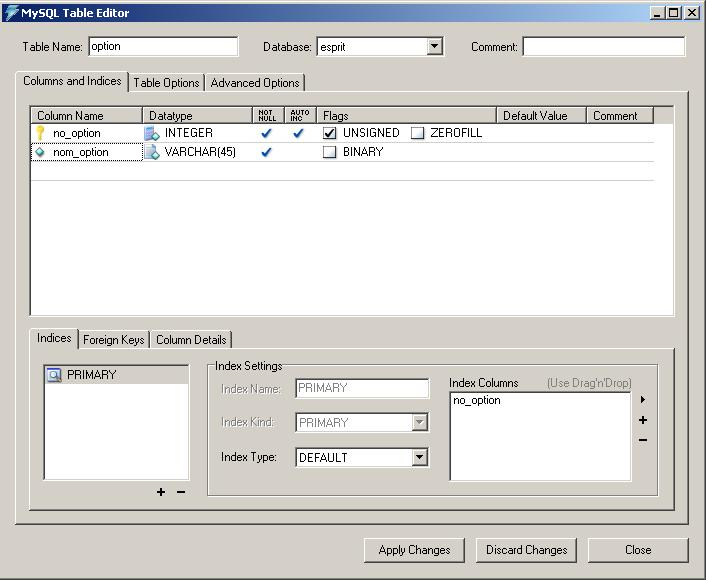
Voila on a terminé la préparation de la base de donnée, on va maintenant créer nos tables.

1. Création des tables :

Nous allons utiliser un autre outil mieux adapté à cette tache : MySQL Query Browser.

|  |  |
| --- | --- |
|  | On en profitera pour tester nos nouveaux paramètres de connexion |
|  | Une fois connecté vous verrez à droite de l’écran les différentes bases de données, faite un clic droit sur la base de données «esprit »  puis « Create New Table ». | |

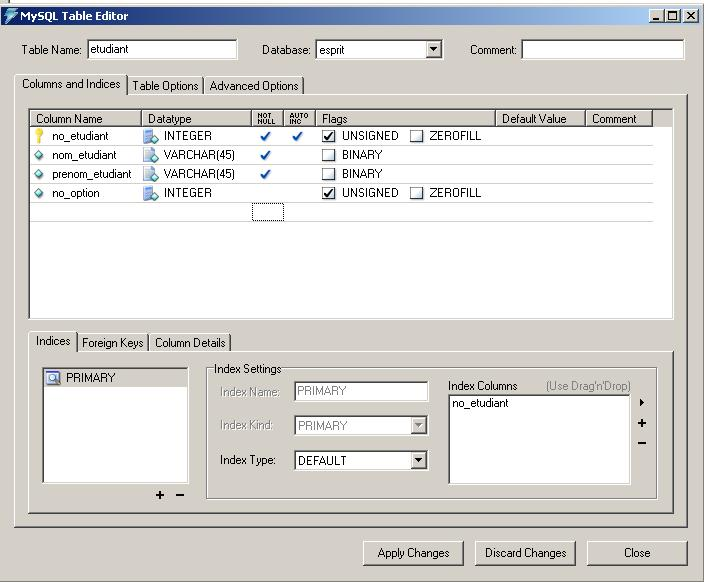
Paramétrez les colonnes comme suit :



Puis « Apply Changes » puis  « execute ».

Pour la table « etudiant » :

Il faut aussi ajouter une colonne  « no\_option»



|  |  |
| --- | --- |
|  | Choisissez l’onglet Foreign Keys, puis « + ».  Pour le nom : FK\_opton  Pour ref Table : option  Column : no\_option  Référence Column : no\_option  Enfin : Apply Changes, et execute. |

Maintenant nous allons créer sous éclipse un nouveau projet java qui va traiter nos objets métiers et stocker leurs données relatives dans notre base de données.

1. Création du nouveau projet :

Lancer eclipse.

|  |  |
| --- | --- |
|  | File->New->Project  Java->JavaProject |

Puis « Next », l’assistant vous demandera le nom du projet : « ProjetJdbc »

Cliquer sur « Finish ».

Vous verrez alors le nouveau projet au niveau du Package explorer (à gauche).

Clique droit sur le repertoir SRC du projet, puis New->package.

Nom du package : « org.esprit.jdbc »

1. Mise en place d’une librairie utilisateur : le driver JDBC pour MySql :

Window->Preferences, une fenêtre s’affiche:

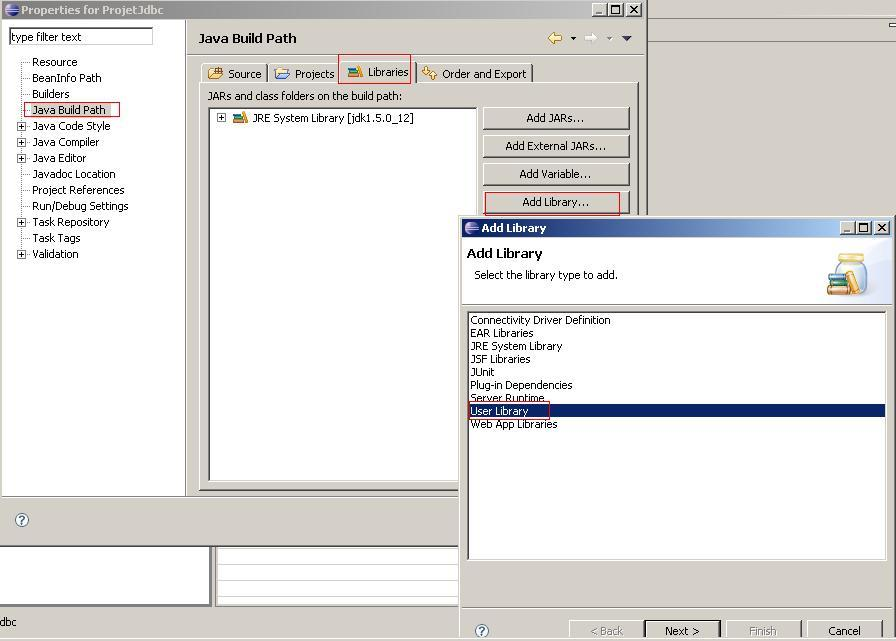
|  |  |
| --- | --- |
|  | Java->BuildPath-> User Libraries puis appuiez sur “New”.  Nom : Mysql puis OK.  La nouvelle librairie s’affiche.  Choisissez « Add Jars » |

|  |  |
| --- | --- |
|  | Parcourir jusqu'à trouver le fichier : mysql-connector-java-5.0.4-bin.jar .  Puis «Load ».  Enfin « OK ». |

1. Relier le Projet à la librairie :

Choisir la racine du projet->properties->Java BuildPath

L’onglet « Libraries » puis « Add Library ».

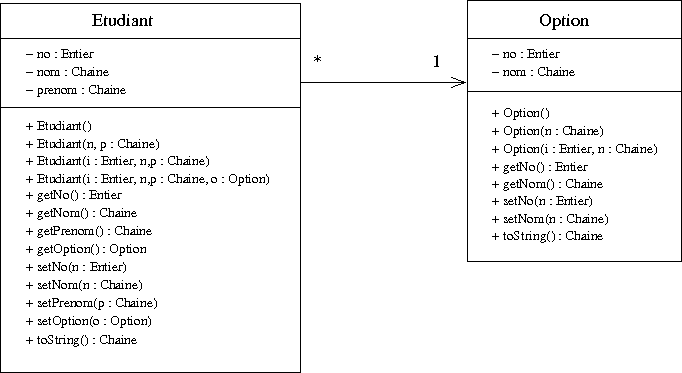


Choisissez « User Library » puis, « Next »une liste des librairies s’affiche :

|  |  |
| --- | --- |
|  |  |

Cocher la librerie puis« Finish »

1. Les Classes métiers :



Sous Eclipse choisissez le package org.esprit.jdbc -> clic droit ->new->Class.

Nom : « Option ». Puis « Finish ».

Code :

**package** org.esprit.jdbc;

**public** **class** Option {

**private** **int** no;

**private** String nom;

**public** **int** getNo() {

**return** no;

}

**public** **void** setNo(**int** no) {

**this**.no = no;

}

**public** String getNom() {

**return** nom;

}

**public** **void** setNom(String nom) {

**this**.nom = nom;

}

}

La classe Etudiant :

**package** org.esprit.jdbc;

**public** **class** Etudiant {

**private** **int** no;

**private** String nom;

**private** String prenom;

**private** Option option;

**public** **int** getNo() {

**return** no;

}

**public** **void** setNo(**int** no) {

**this**.no = no;

}

**public** String getNom() {

**return** nom;

}

**public** **void** setNom(String nom) {

**this**.nom = nom;

}

**public** String getPrenom() {

**return** prenom;

}

**public** **void** setPrenom(String prenom) {

**this**.prenom = prenom;

}

**public** Option getOption() {

**return** option;

}

**public** **void** setOption(Option option) {

**this**.option = option;

}

}

1. Le code de la classe principale : Test.java

**package** org.esprit.jdbc;

**import** java.sql.Connection;

**import** java.sql.DriverManager;

**import** java.sql.ResultSet;

**import** java.sql.SQLException;

**import** java.sql.Statement;

**public** **class** Test {

**static** Connection *connection*=**null**;

**static** Statement *st*=**null**;

**public** **static** **void** enregistrerEtudiant(Etudiant etd)

{

String nom=etd.getNom();

String prenom=etd.getPrenom();

Option opt=etd.getOption();

**int** noOption=opt.getNo();

String sql="INSERT INTO esprit.etudiant(nom\_etudiant,prenom\_etudiant,no\_option)" +

"VALUES('"+nom+"','"+prenom+"',"+noOption+")";

**try** {

*st*=*connection*.createStatement();

*st*.executeUpdate(sql);

} **catch** (SQLException e1) {

// **TODO** Auto-generated catch block

e1.printStackTrace();

}

}

**public** **static** **void** enregistrerOption(Option opt){

String nom=opt.getNom();

String sql="INSERT INTO esprit.option(nom\_option)VALUES('"+nom+"')";

**try** {

*st*=*connection*.createStatement();

*st*.executeUpdate(sql);

} **catch** (SQLException e1) {

// **TODO** Auto-generated catch block

e1.printStackTrace();

}

}

**public** **static** **void** supprimerOption(Option opt){

String sql="DELETE FROM esprit.option WHERE nom\_option='"+opt.getNom()+"'";

**try** {

*st*=*connection*.createStatement();

*st*.executeUpdate(sql);

} **catch** (SQLException e1) {

// **TODO** Auto-generated catch block

e1.printStackTrace();

}

}

**public** **static** Option lireOption(**int** no){

Option opt=**new** Option();

String sql="SELECT \* FROM esprit.option WHERE no\_option='"+no+"'";

**try** {

ResultSet rs;

*st*=*connection*.createStatement();

rs=*st*.executeQuery(sql);

**while** (rs.next()){

opt.setNo(rs.getInt("no\_option"));

opt.setNom(rs.getString("nom\_option"));

}

} **catch** (SQLException e1) {

// **TODO** Auto-generated catch block

e1.printStackTrace();

}

**return** opt;

}

**public** **static** **void** main(String[] args) {

System.*out*.println("Bonjour");

**try** {

Class.*forName*("com.mysql.jdbc.Driver").newInstance();

} **catch** (InstantiationException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

} **catch** (IllegalAccessException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

} **catch** (ClassNotFoundException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

String url="jdbc:mysql://localhost/4a1";

String login="esprituser";

String mdp="espritpwd";

**try** {

*connection*=DriverManager.*getConnection*(url,login,mdp);

System.*out*.println("Connection réussie...");

} **catch** (SQLException e) {

System.*out*.println("Connexion echoée....");

e.printStackTrace();

}

Option option=**new** Option();

option.setNom("Informatique");

Etudiant etudiant;

etudiant=**new** Etudiant();

etudiant.setNom("foulen");

etudiant.setPrenom("benfoulen");

etudiant.setOption(option);

*enregistrerOption*(option);

*enregistrerEtudiant*(etudiant);

//supprimerOption(option);

//Option opt1=lireOption(1) ;

}

}

1. Le design pattern DAO :

Le patron de conception DAO propose la création d'une classe DAO par classe métier.

Chaque classe DAO contient les méthodes de liaison avec la base de données, parfois appelées CRUD (pour *Create, Request, Update, Delete*).

Les méthodes de suppression et de modification renvoient un booléen indiquant le succès de l'opération, la méthode d'insertion renvoie l'identifiant affecté à la nouvelle ligne de la table (utile en cas d'identifiant auto-incrémenté par le SGBD), et plusieurs méthodes « get »permettent d'obtenir un objet en fonction de différents critères de recherche.

1. Notre nouvelle organisation :

Encapsuler tous le code relatif aux détails techniques de la connexion dans une classe « ConnexionMySQL » :

**import** java.sql.Connection;

**import** java.sql.DriverManager;

**import** java.sql.ResultSet;

**import** java.sql.SQLException;

**import** java.sql.Statement;

**public** **class** ConnexionMySql {

**private** Connection connection;

**private** Statement st;

**private** ResultSet rs;

**private** String sql;

**public** **boolean** seConnecter(){

String url="jdbc:mysql://localhost/esprit";

String login="esprituser";

String mdp="espritpwd";

**try** {

Class.*forName*("com.mysql.jdbc.Driver").newInstance();

} **catch** (InstantiationException e) {

e.printStackTrace();

**return** **false**;

} **catch** (IllegalAccessException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

**return** **false**;

} **catch** (ClassNotFoundException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

**return** **false**;

}

**try** {

connection=DriverManager.*getConnection*(url,login,mdp);

} **catch** (SQLException e1) {

e1.printStackTrace();

**return** **false**;

}

**return** **true**;

}

**public** **boolean** executerUpdate(String sql){

**try** {

st=connection.createStatement();

st.executeUpdate(sql);

} **catch** (SQLException e) {

e.printStackTrace();

**return** **false**;

}

**return** **true**;

}

**public** **boolean** executerRequete(String sql){

**try** {

st=connection.createStatement();

setRs(st.executeQuery(sql));

} **catch** (SQLException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

**return** **false**;

}

**return** **true**;

}

**public** **void** setRs(ResultSet rs) {

**this**.rs = rs;

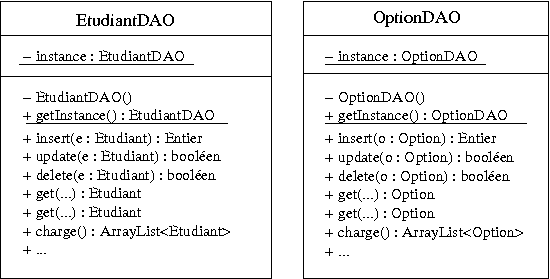
}

**public** ResultSet getRs() {

**return** rs;

}}

En plus des classes Etudiant et option nous auront des classes EtudiantDAO et OptionDao



Notre nouveau Projet :

Organisation :

