

Diallo Thierno Yaya

2016 – 2017

Application Informatique (JEE)

Brigitte Copin

Rapport de Projet JEE

Partie PL/SQL

# Sujet :

Nous avons décidé de mettre en place une plateforme de gestion des stages. Le but principal est de permettre aux différents acteurs (Etudiants, Superviseurs, coordinatrice) impliquer dans le processus de stage de pouvoir consulter les informations concernant le stage d’un élève donné.

Cela permettra aussi de savoir précisément qui contacter en cas de problème et à qui envoyer les différents documents. Le programme sera principalement alimenté par la personne qui s’occupe des conventions de stage, ainsi que l’étudiant qui pourra entre ses informations directement sur la plateforme et choisir son professeur référent.

# La base de données

## Capture d’écranSchéma MCD :

Figure 1: Schéma de la base de données

### Explication

Une coordinatrice (JEE\_Coordinator) possède un identifiant (idcoordinator), un login (usn) et un mot de passe (pwd) ainsi qu’une information classique (NAME).

Elle peut coordonner un ou plusieurs Etudiants (JEE\_STUDENT) qui possèdent des informations classiques telles que : NAME, MAIL, LOGIN, MOT DE PASSE, ADDRESSE ainsi que les informations relatives à l’établissement d’enseignement.

Chaque Etudiant effectue un Stage (JEE\_STAGE) qui contient des informations sur son Thème du TFE(TFETHEME), la date de début et de fin. Cette dernière table est reliée à deux autres tables qui sont respectivement JEE\_CORPORATE et JEE\_SCHOOLMEMBER.

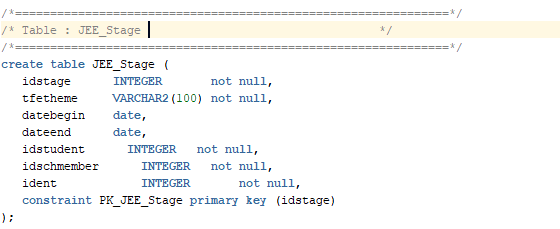
La table JEE\_CORPORATE renferme les informations sur l’entreprise de stage Nom Entreprise (entname), l’adresse de l’entreprise et des informations sur le superviseur du stage (nom, email et téléphone).

La dernière table JEE\_SCHOOLMEMBER contient des informations sur le professeur chargé de superviser le stage de l’étudiant. Il est composé des informations classique (nom, prénom, email, tél, login et mot de passe).

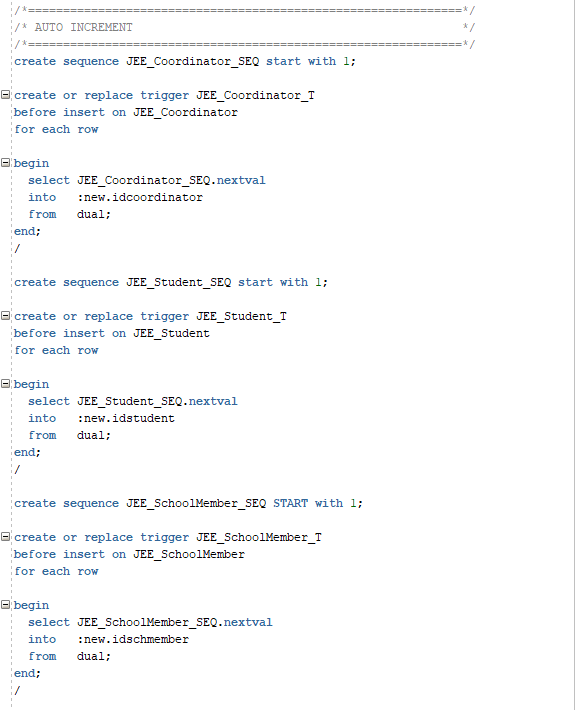
# Script de création de la base de données











# Capture d’écran

# Scripts PL/SQL

* Package :

Création d’un package « **ConnexionDB »** qui contient des fonctions d’authentifications des différents utilisateurs du programme

*connexionCoordinator*: check la connexion d’un coordinateur.

*connexionSchoolMember*: check la connexion d’un professeur.

*connexionStudent*: check la connexion d’un étudiant.

Ces fonctions ont deux paramètre le login et le mot de passe et il renvoie un boolean pour les deux premiers et un number pour le dernier. Ce sont ces données qui seront traitées du code java.

Quant aux exceptions elles sont traitées dans chacune des fonctions. Lorsqu’une une exception ce déclenche, le programme pl/sql renvoie automatiquement l’exception. Celle-ci sera non seulement intercepté et gérée par le programme pl/sql mais le programme renvoie aussi l’exception côte java. Côté java, l’exception viendra sous forme SQLException et sera traitée.

Création d’un package « **AddData»** qui contient les différentes procédures d’ajout de données.

*addCoordinator*: ajoute un coordinateur.

*addEntreprise* : ajoute une entreprise.

*addShoolMember*: ajoute un schoolMember.

*addStudent*: ajoute un student

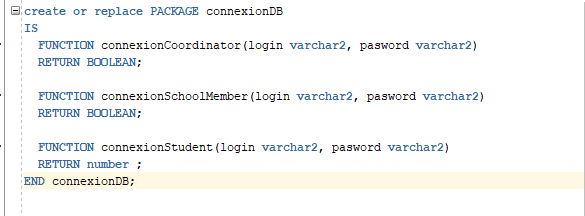
*addStage*: ajoute un stage.

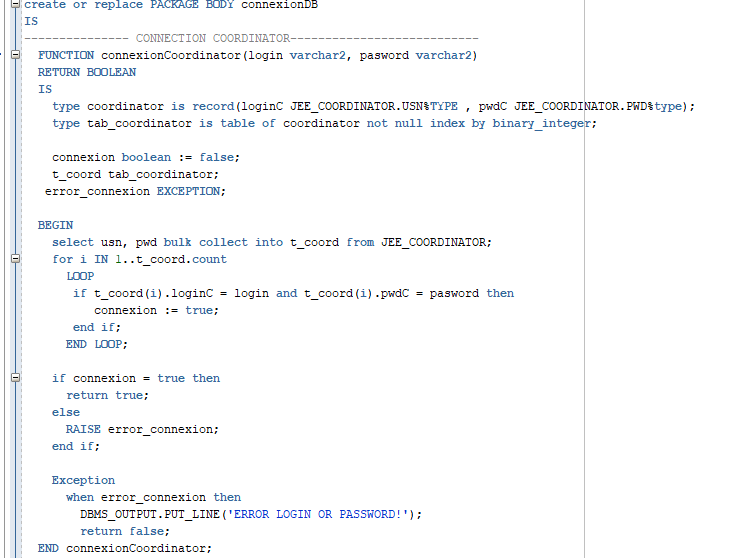
les exceptions dans cet package sont gérer par cette déclaration ci-dessous :

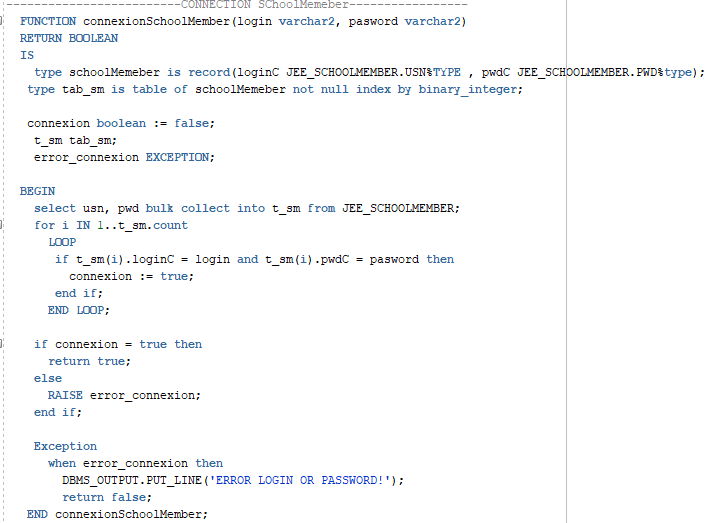
fk\_violationException EXCEPTION;

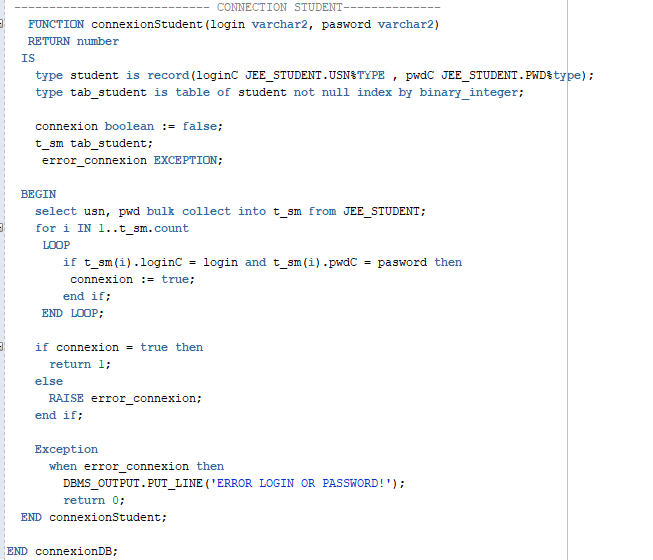
PRAGMA EXCEPTION\_INIT(fk\_violationException,-2291);

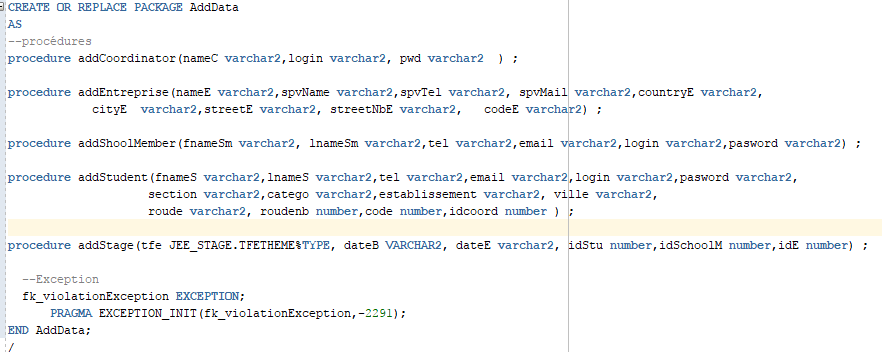
S’il y a une violation des contraintes, pl/sql renvoie automatiquement une exception qui sera intercepté comme une SQLException en Java. Le message contenu dans l’exception sera traité côté java.





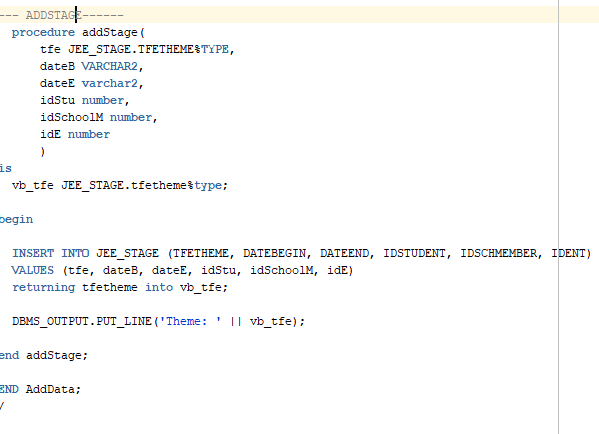




\*\*PacKage addData\*\*\*\*

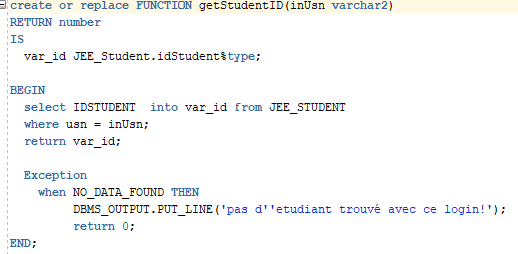






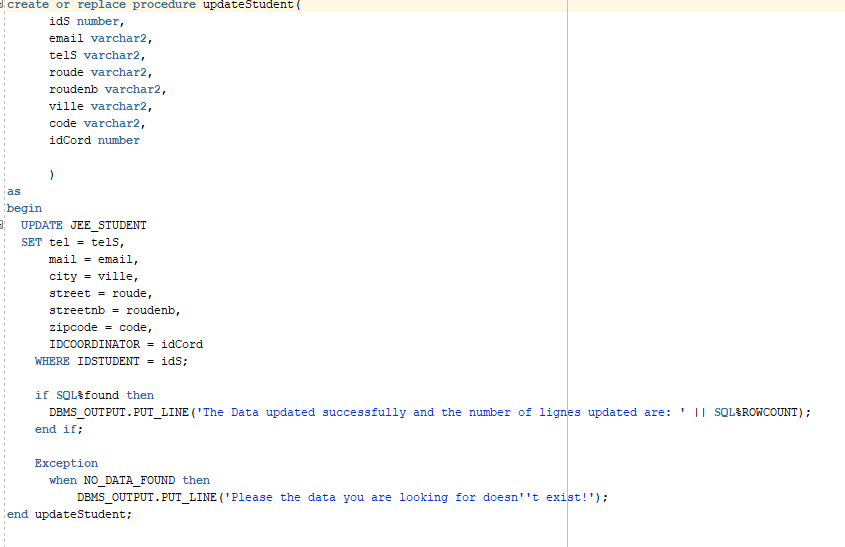
Fonctions :

* **« getStudentID** » : cette fonction nous permet de retourner l’ID d’un étudiant dont son nom est passé en paramètre. Si aucun n’étudiant avec ce nom est trouvé une exception est déclenchée.

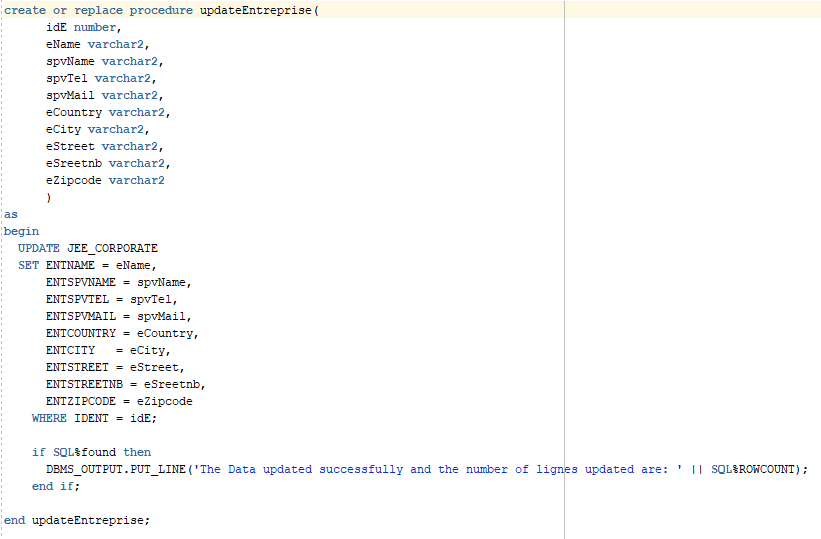


Procédures :

“***updateStudent*** “: cette procédure permet la mise à jour d’un étudiant ; il déclenche une exception lorsqu’il ne trouve pas l’étudiant correspondant.



***«addSchoolMember »***: cette procédure ajoute

***updateEntreprise*** “: cette procédure permet la mise à jour d’une entreprise.

« ***updateStage*** » : cette procédure permet la mise à jour d’un stage ; vous remarquerez l’utilisation de la clause returning ainsi que le nombre de lignes qui ont été modifiées. Si aucune correspondance n’est retrouvée, une Exception de type « NO\_DATA\_FOUND » est déclenchée.



***« deleteStage »*** : cette dernière procédure permet la suppression d’un stage dont l’ID est passé en argument. Vous remarquerez l’utilisation d’un curseur.

