M1 MIAGE FC/FA – 2010/2011

# Bases de Données Avancées

octobre 2010

# TP PLSQL - JDBC

**Exercice 1 :** Un programme de simulation d'un supermarché utilise une base de données. On s'intéresse ici à la gestion du flux des clients aux caisses. Le schéma de la base est le suivant :

```
create table caisse (
  identifiant varchar2(2) constraint caisse_pkey primary key,
  ouverte number(1),
  constraint ouverte_bool check (ouverte between 0 and 1)
) ;

create table client (
  numero number(3) constraint client_pkey primary key,
  duree_prevue number(3),
  idCaisse varchar2(3) constraint affect_caisse_fkey references caisse,
  rang number(2)
) ;
```

Un client est affecté à au plus une caisse, et il peut y avoir plusieurs clients affectés à une caisse. Voici quelques informations sur les attributs définis dans les tables :

- Dans la table `CAISSE`, l'attribut `ouverte` permet de savoir si la caisse est ouverte (valeur 1) ou fermée (valeur 0).
- Dans la table `CLIENT`, l'attribut `duree_prevue` est la durée prévue de traitement du caddy du client. Le temps d'attente à une caisse (pour un nouveau client arrivant) est la somme des temps prévus pour le traitement de chaque client affecté à cette caisse.
- Dans la table `CLIENT`, l'attribut `rang` est le rang qu'occupe le client à la caisse. Donc si un client a le rang 3, ça signifie qu'il y a 2 clients devant lui.

Question 1.1 : Donnez le code SQL permettant de créer les trois vues suivantes :

1. Une vue `caisses_ouvertes`, dont le schéma est `caisses_ouvertes(identifiant)`, et qui ne contient que les caisses ouvertes.
2. Une vue `durees_attentes`, qui contient pour chaque caisse ouverte son identifiant et la duree d'attente prévue. Le schéma de cette vue sera `durees_attentes(identifiant, duree)`. Une caisse qui n'a aucun client a bien sûr une durée d'attente de 0.
3. Une vue `meilleures_caisses` qui contient les identifiants de caisses dont la durée d'attente est minimale. Le schéma de cette vue sera `meilleures_caisses(identifiant)`

Vous avez le droit (c'est même conseillé) d'utiliser une vue pour en définir une autre.

On définit un paquetage pour gérer les affectations des clients aux caisses. Voici sa spécification :

```
create or replace package PAQ_CAISSE as
  procedure affectation_client(leNumClient CLIENT.numero%type) ;

  procedure passer_client(la_caisse caisse.identifiant%type) ;

  procedure ouvrir_caisse(la_caisse caisse.identifiant%type) ;

  procedure fermer_caisse(la_caisse caisse.identifiant%type) ;

  function taille_max return NUMBER ;

  function nb_clients(la_caisse caisse.identifiant%type) return NUMBER ;

  PAS_DE_CAISSES_DISPO Exception ;
  PRAGMA exception_init(PAS_DE_CAISSES_DISPO,-20105);

  CLIENT_DEJA_EN_CAISSE Exception ;
  PRAGMA exception_init(CLIENT_DEJA_EN_CAISSE,-20106);

  PAS_DE_CLIENT Exception ;
  PRAGMA exception_init(PAS_DE_CLIENT,-20107);

  CAISSE_INCONNUE Exception ;
  PRAGMA exception_init(PAS_DE_CLIENT,-20108);

end ;
```

*vous pouvez par la suite utiliser les vues définies à la question précédente.*
Question 1.2 : Ecrire le corps de la fonction `taille_max`. Cette fonction est une fonction constante qui renvoie toujours 3. *on aurait pu utiliser une constante mais les constantes PL/SQL ne sont pas accessibles à partir d'un programme JDBC.*

Question 1.3 : Ecrire le corps de la fonction `nb_clients`. Si la caisse est fermée, cette fonction renvoie 0. Si la caisse n'existe pas, la fonction déclenche l'exception `CAISSE_INCONNUE`. *Pour éviter de recalculer à chaque fois le nombre de clients à partir de la table* `CLIENT` *vous pouvez définir un trigger qui met à jour une nouvelle colonne* `nb_clients` *de la table* `CAISSE`.

Question 1.4 : Ecrire le corps de la procédure `passer_client`. Cette procédure enlève le client de rang 1 de la file d'attente de la caisse (identifiant en paramètre) et décale les rangs des clients qui restent dans la file. Si la caisse n'a pas de client en attente, alors cette procédure déclenche l'exception `PAS_DE_CLIENT`.

Question 1.5 : Ecrire le corps de la procédure `affectation_client`. Cette procédure recherche une caisse où il y a le moins d'attente et y affecte le client dont le numéro est passé en paramètre. S'il n'y a aucune caisse ouverte, cette procédure déclenche l'exception `PAS_DE_CAISSES_DISPO`. Si le client est déjà affecté à une caisse, on déclenche l'exception `CLIENT_DEJA_EN_CAISSE`.

Question 1.6 : Ecrire le corps de la procédure `ouvrir_caisse` qui, comme son nom l'indique, ouvre la caisse dont l'identifiant est passé en paramètre. Cette procédure ne déclenche pas d'exception si la caisse est déjà ouverte, par contre elle déclenche `CAISSE_INCONNUE` si la caisse n'existe pas.

Question 1.7 : Ecrire le corps de la procédure `fermer_caisse` qui ferme la caisse dont l'identifiant est passé en paramètre. Cette procédure doit réaffecter les clients de cette caisse aux autres caisses ouvertes. On ne déclenche pas d'exception si la caisse est déjà fermée.

**Exercice 2 :**
*suite de l'exercice 1*
On écrit l'application de simulation en java. Vous renommerez `BaseACompleter.java` en `Base.java` et `MagasinACompleter.java` en `Magasin.java`

### Développement Java avec JDeveloper

Ouvrez `JDeveloper`, créez un `Workspace` et un `Project` dans cet espace de travail. Pour cela, dans l'onglet Applications (à gauche), sélectionnez `Applications` puis avec le clic droit, `new Application`. Le nom de votre application est `Application1` et choisissez son répertoire sur H, par exemple `H:\tpjdbc`. Suivez les fenêtres de dialogue pour créer un projet `Project1` dans le répertoire `H:\tpjdbc\Project1`

Sous windows, dans le poste de travail, allez dans le répertoire `H:\tpjdbc\Project1`, créez-y un sous-répertoire `src` et recopiez-y les fichiers java donnés pour le TP.

Sous jdeveloper, sélectionnez le projet `Project1` puis avec le clic droit, choisissez `Add to project content`. Ajoutez le répertoire `src`.

Pour pouvoir utiliser le pilote JDBC d'oracle, il faut l'ajouter dans le `CLASSPATH` du projet. Pour cela, sélectionnez votre projet, avec le clic droit choisissez `Project Properties` puis la rubrique `Libraries`. Appuyez sur le bouton `Add jar/directory` et suivez les instructions pour ajouter le chemin `C:\Oracle\product\10.2.0\client_1\jdbc\lib\ojdbc14.jar`
Votre environnement de travail est maintenant configuré pour pouvoir travailler.

### Accès à la base

Pour accéder à la base, nous allons utiliser une classe `Base` qui permettra de créer une connexion, et d'obtenir des ressources pour interroger la base.
Question 2.1 : Complétez le fichier `Base.java` et écrivez un petit programme testant votre connexion à la base Oracle via JDBC (ce sera par exemple, le `main` de la classe `Base`).

## Simulation des caisses du magasin

Nous allons nous intéresser à la représentation des caisses dans cette simulation. Vous avez besoin pour cela du fichier `Magasin.java`.

### La classe Caisse

Question 2.2 : Ecrire le constructeur de la classe `Caisse`. Ce constructeur prend en paramètre

- le numéro de la caisse à créer (on suppose que c'est un numéro qui n'existe pas déjà dans la base)
- un booléen qui vaut `true` si et seulement si la caisse créée est ouverte.

Le constructeur initialise les variables d'instance (sauf la durée qui est déjà initialisée) et insère dans la base une ligne dans la table `CAISSE` avec les mêmes informations que les paramètres du constructeur.

Question 2.3 : Ecrire les méthode `ouvrir`, `fermer`, `passerClient` et `nbClients` qui font appel à des procédures ou fonctions du paquetage stocké `PAQ_CAISSE`

Question 2.4 : Ecrire la méthode `temps_traitement()` qui renvoie

- La durée prévue de traitement du client de rang 1
- ou -1 s'il n'y a pas de client de rang 1 (donc pas de client à la caisse).

### La classe Magasin

Question 2.5 : Complétez la classe Magasin, dont la principale fonction est la simulation du passage des clients en caisse.