Connexion à la base de données Oracle

## **1. Configuration de l’application dans JDeveloper**

## Création d'une application et d’un projet

Tout d’abord, créez une application commençant par la création d’un projet comme suit:

1. Dans le navigateur de l'application, cliquez sur « New » application.

2. « Create Generic Application » wizard est activé. Entrez le nom de l’application « HRApp » dans le champ « Nom de l'application », et de la liste des applications, sélectionnez l’Application générique. Cliquez sur Suivant.

3. Entrez le nom du projet. Cliquez sur Terminer.

4 La nouvelle application « HRApp » est affichée dans le navigateur application. Configurez le projet et l'application dans JDeveloper

5 Enregistrez votre application, dans le menu Fichier, sélectionnez Enregistrer tout.

## **2. Connexion à la base de données Oracle à partir d'une application Java**

## 2.1 Création d'une classe Java dans JDeveloper

Créer une classe Java appelée DataHandler, qui contiendra les méthodes pour récupérer la base de données et modifier les données.

1. Dans le Navigateur application, faire right-click et sélectionnez « New ».

2. Dans « New Gallery », sélectionnez la catégorie « General ». Dans la liste des éléments, sélectionnez une classe Java, puis cliquez sur OK.

3. Dans « Create Java Class », faire entrez le nom de la classe « DataHandler » et le package « hr ».



La classe « DataHandler » est créé et affichée dans l'éditeur Java Source. La déclaration de package, la déclaration de classe et le constructeur sont créé par défaut.



## 2.3 Ajout de JDBC et JSP bibliothèques

Pour inclure les bibliothèques dans le projet, suivez les étapes suivants:

1. Double-cliquez sur « View » projet dans le Navigateur application pour afficher les Propriétés du projet.

2. Cliquez sur « Bibliothèques et Classpath », puis cliquez sur « Add Library ».



3. Sélectionnez « JSP Runtime » et cliquez sur OK pour l'ajouter à la liste des bibliothèques. De même, ajouter la bibliothèque JDBC Oracle.

4 Cliquez sur OK.

## 2.4 Import packages JDBC dans une application Java

Pour utiliser JDBC dans l'application Java, import les packages JDBC comme ci-dessous:

package hr;

import java.sql.Connection;

import oracle.jdbc.pool.OracleDataSource;

## 2.5 Déclaration des variables de connexion liés

Déclarer les trois variables de connexion comme suit:

String jdbcUrl = null;

String userid = null;

String password = null;

Ces variables seront utilisées dans l'application pour contenir des valeurs fournies par l'utilisateur lors de la connexion pour identidier l'utilisateur et créer une connexion à la base de données. La variable JDBCURL est utilisé pour maintenir l'URL de la base de données. Les variables d'ID utilisateur et mot de passe sont utilisés pour identifier l'utilisateur.

Déclarez une instance de connexion comme suit:

Connection conn;

Déclarer des variables de connexion et l'objet de connexion

package hr;

import java.sql.Connection;

import oracle.jdbc.pool.OracleDataSource;

public class DataHandler {

public DataHandler() {

}

String jdbcUrl = null;

String userid = null;

String password = null;

Connection conn;

}

## 2.6 Création de la méthode de connexion

Pour se connecter à la base de données, créer une méthode comme suit:

1. Ajoutez la déclaration de méthode, après la déclaration de connexion:

public void getDBConnection() throws SQLException

2. Déclarez une instance OracleDataSource comme suit:

OracleDataSource ds;

3. Entrez la commande suivante pour créer un nouvel objet OracleDataSource:

ds = new OracleDataSource ();

4. Définir l'URL de l'objet DataSource:

ds.setURL(jdbcUrl);

5. Entrez le suivant :

conn = ds.getConnection(userid,password);

Ajouter une méthode pour se connecter à la base de données

package hr;

import java.sql.Connection;

import java.sql.SQLException;

import oracle.jdbc.pool.OracleDataSource;

public class DataHandler {

public DataHandler() {

}

String jdbcUrl = null;

String userid = null;

String password = null;

Connection conn;

public void getDBConnection() throws SQLException{

OracleDataSource ds;

ds = new OracleDataSource();

ds.setURL(jdbcUrl);

conn=ds.getConnection(userid,password);

}

}

6. Compiler la classe.