

Bases de Données Relationnelles

TP3b :Gestion d'un emploi du temps (suite)

SI2 et MAM2

1 Exposé du problème

1.1 Création d'une table pour l'affichage d'un EDT

Cette table doit contenir les items qu'il faut afficher dans l'emploi du temps d'une semaine donnée d'un cursus donné. Pour un même jour, il peut y avoir plusieurs lignes s'il y a des cours en parallèle.

Pour déterminer, journée par journée, sur quelle ligne placer chacun des modules se déroulant ce jour-là, on pourra utiliser la table SQL suivante :

`affichage (module, nom_creneau, jour, heure_deb,duree, ligne)`

qu'on remplira par programme.

```
/*  
  
Table pour l'affichage de EDT d'une semaine  
  
*/  
  
drop table affichage;  
  
create table affichage (  
    code_module  varchar(20), -- ne peut référencer module car devra prendre  
                                -- la valeur nulle pour les créneaux vides  
    groupe      varchar(10),  
    jour        numeric(1) ,  
    heured      numeric(2) ,  
    duree       numeric(2),  
    ligne       numeric(2),  
    salle       varchar(10),  
    unique (code_module, groupe),  
    primary key (jour,ligne , heured)
```

```
);

/*
Attribution des droits (necessaires pour l'accès
via le server resin)
*/

grant select on
    module, creneau_type, cursus, cursus_groupe, edt_sem
to invited;

grant select, insert, delete, update on affichage to invited;
```

Pour placer dans cette table les créneaux à afficher, y compris les créneaux vides (avec un champ module égal à NULL), on remplira la colonne "ligne" à l'aide d'un programme java qui devra effectuer les opérations suivantes:

1. Sélectionner un par un les créneaux à placer (avec une requête SQL dans la table générale des créneaux) en fonction de la semaine et du cursus choisi .
2. Pour chaque créneau à placer, déterminer (là-encore avec une requête SQL) s'il existe un créneau vide susceptible de l'accueillir. S'il n'en n'existe pas, introduire une nouvelle "ligne " d'affichage pour le jour concerné, en ajoutant simplement à la table affichage un créneau vides de 8h démarrant en 1 pour le jour concerné, avec une nouvelle ligne.
Pour "placer" le créneau dans l'emploi du temps, dans le créneau vide d'accueil sélectionné, il suffit alors d'assigner la ligne correspondante à ce créneau vide, et de remplacer le créneau vide par 0,1 ou 2 nouveaux créneaux vides, selon la configuration.
Il faut aussi mettre des modules NULL dans les "trous".

1.2 Remplissage via JDBC de la table d'affichage

On vous demande d'utiliser jdbc pour vous connecter a votre base de données pour remplir la table affichage avec les enseignements de la semaine 49 pour les étudiants de votre cursus.

Comme vous ne desirez pas faire apparaitre en clair votre mot de passe dans le code java, la connexion se fera sous couvert de l'utilisateur invited dont le mot de passe est invited. Vous devez donner à cet utilisateur uniquement les droits dont il a besoin.

Vous aurez besoin d'un driver postgres. Si vous utilisez le serveur postgres dbms.essi.fr (version 8.14) et jdk 1.4 ou 1.5 (ce qui est le cas si vous travaillez sur morag), vous pouvez utiliser postgresql-8.1-407.jdbc2_1.jar (qui se trouve dans ~rueher/BDR/TP3b). D'autres drivers sont disponibles à l'adresse jdbc.postgresql.org/download.html.

```
import java.sql.*;
import java.io.*;
```

```

public class EdtRemplissage{

    public static void main(String args[]) {

        String sqlUser = "invited ";
        String sqlPwd = "invited ";
        //String url = "jdbc:postgresql://dbms.essi.fr:5432/claudine";
        String url = "jdbc:postgresql://localhost:5432/rueher ";
        Connection conn;
        Statement stmt;
        String sqlTxt;

        String errMsg;
        // String errPage = "edt_result.jsp ";
        // String errFwd;

        // Parametres du formulaire de selection
        String reqCursus ="SI2 ";
        // String reqSem;
        int reqSemInt=49;
        try {
            Class.forName("org.postgresql.Driver");
        }
        catch (java.lang.ClassNotFoundException e) {
            System.err.print ("ClassNotFoundException: ");
            System.err.println (e.getMessage());
        };

        try {
            conn = DriverManager.getConnection(url , sqlUser , sqlPwd);
            stmt = conn.createStatement();
            stmt.executeUpdate("DELETE FROM affichage ");
            int [] nbLignes = new int [5]; //Nombre de lignes d'affichage pour chaque jour
            for (int i=1; i<=5; i++){
                stmt.executeUpdate(
                    "insert into affichage (jour , heured , duree , ligne) " +
                    " values(" + Integer.toString(i) + ", 1, 8, 1) "
                );
            nbLignes[i-1]=1;
        };
    }
}

```

```

sqlTxt =
" SELECT E.code_module, E.groupe, E.jour, E.heured, CT.duree, E.salle " +
" FROM edt_sem E, cursus_groupe CCT, creneau_type CT " +
" WHERE CCT.nomcursus = ? and E.sem= ? and " +
"      E.groupe=CCT.groupe and " +
"      CT.code_module= E.code_module and CT.groupe=E.groupe " +
" ORDER BY 3,4,5 " ;
errMsg="PB SQL dans insertion des creneaux a placer : \n" + sqlTxt;
PreparedStatement stmtModules = conn.prepareStatement(sqlTxt);

stmtModules.setString(1, reqCursus);
stmtModules.setInt(2, reqSemInt);
ResultSet rsetModules = stmtModules.executeQuery();

while (rsetModules.next()) {
String code_module = rsetModules.getString(1);
String groupe = rsetModules.getString(2);
int jour = rsetModules.getInt(3);
int heured = rsetModules.getInt(4);
int duree = rsetModules.getInt(5);
int heuref = heured + duree-1;
String salle = rsetModules.getString(6);
//parametres du creneau vide a trouver
boolean cv_trouve=false;
int cv_ligne=1;
int cv_heured=1;
int cv_duree=1;
int cv_heuref=1;

// -----
// Calcul de la ligne de placement
// recherche de creneau vide
// -----
errMsg="PB SQL dans selection des creneaux libres";
PreparedStatement stmtSelCrenVide = conn.prepareStatement(
" SELECT A.ligne, A.heured, A.duree " +
" FROM affichage A " +
" WHERE A.code_module IS NULL and A.jour = ? and " +
"      A.heured <= ? and ? <= A.heured+duree-1 " +
" ORDER BY 1, 2"
);

```

```

stmtSelCrenVide.setInt(1, jour);
stmtSelCrenVide.setInt(2, heured);
stmtSelCrenVide.setInt(3, heuref);
ResultSet rsetSelCrenVide = stmtSelCrenVide.executeQuery();
while (rsetSelCrenVide.next()) {
    cv_trouve=true;
    cv_ligne= rsetSelCrenVide.getInt(1);
    cv_heured= rsetSelCrenVide.getInt(2);
    cv_duree= rsetSelCrenVide.getInt(3);
    cv_heuref= cv_heured + cv_duree -1;
    break;
}
rsetSelCrenVide.close();
if (! cv_trouve ) {
    // -----
    // insertion d'une nouvelle ligne dans le jour
    // par creation d'un nouveau creneau vide de 8h
    // recherche de la ligne maximale pour le jour
    // -----
    nbLignes[jour-1] +=1;
    cv_ligne=nbLignes[jour-1];
    errMsg="PB SQL dans ajout nouvelle ligne";
    stmt.executeUpdate(
        "insert into affichage (jour, heured, duree, ligne) " +
        " values(" + Integer.toString(jour) + ", 1, 8, " +
        Integer.toString(cv_ligne) + ")"
    );
    cv_heured = 1; cv_duree=8; cv_heuref=8;
} //Eof (! cv_trouve )

// -----
// Le creneau vide existe donc maintenant.
// -----

// -----
// on supprime le creneau vide
// -----
errMsg="PB SQL dans suppression du creneau vide";
stmt.executeUpdate(
    "delete from affichage " +
    "where jour=" + Integer.toString(jour) + " and heured=" +
    Integer.toString(cv_heured) + " and " +

```

```

        ligne=" + Integer.toString(cv_ligne)
    );

    // -----
    // On place le creneau sur la ligne.
    // -----
    errMsg="PB SQL dans insertion du creneau place";
    stmt.executeUpdate(
        "insert into affichage (code_module, groupe,jour, heured, duree, ligne, salle) " +
        " values('" + code_module + "', '" + groupe + "', "
        + Integer.toString(jour) + ", " + Integer.toString(heured) +
        " , " + Integer.toString(duree) + ", " + Integer.toString(cv_ligne)
        + ",'" + salle + "')"
    );
    if (cv_heured < heured ) {
        // -----
        // insertion d'un creneau vide a gauche
        // -----
        errMsg="PB SQL dans insertion du creneau vide gauche";
        stmt.executeUpdate(
            "insert into affichage (jour, heured, duree, ligne) " +
            " values(" + Integer.toString(jour) + ", "
            + Integer.toString(cv_heured) + ", "
            + Integer.toString(heured - cv_heured) + ", "
            + Integer.toString(cv_ligne) + ")"
        );
    }
    if (heuref < cv_heuref ) {
        // -----
        // insertion d'un creneau vide a droite
        // -----
        errMsg="PB SQL dans insertion du creneau vide droite";
        stmt.executeUpdate(
            "insert into affichage (jour, heured, duree, ligne) " +
            " values(" + Integer.toString(jour) + ", "
            + Integer.toString(heuref+1) + ", "
            + Integer.toString(cv_heuref-heuref) + ", "
            + Integer.toString(cv_ligne) + ")"
        );
    }
}
}
rsetModules.close();

```

```
        stmt.close();
    conn.close();

    }
    catch(SQLException ex) {
        System.err.println("SQLException: " + ex.getMessage());
    }
}

};
```
