

# Rapport TP2 BD

Antoine Thebaud

Aurélien Monnet-Paquet

23 octobre 2015

## 1 Partie Java

1. Fonction pour afficher la liste des animaux du ZOO

```
1 private static void listeAnimaux() throws SQLException {
2     Statement stmt = conn.createStatement();
3     ResultSet rs = stmt.executeQuery("SELECT * FROM LesAnimaux");
4
5     //Affichage attributs de la table
6     ResultSetMetaData resultSetMetaData = rs.getMetaData();
7     int nCols = resultSetMetaData.getColumnCount();
8     for (int i=1; i<=nCols; i++){
9         System.out.print(String.format("%-15s", resultSetMetaData.getColumnName(i)));
10    }
11
12    //Affichage contenu de la table
13    while (rs.next()) {
14        System.out.println(String.format("%-15s%-15s%-15s%-15s%-15s%-15s%-15s",
15        "%-15s",
16        rs.getString("nomA"),
17        rs.getString("sexe"),
18        rs.getString("type_an"),
19        rs.getString("fonction_cage"),
20        rs.getString("pays"),
21        rs.getString("annais"),
22        rs.getString("nocage"),
23        rs.getString("nb_maladies")));
24    }
25 }
```

## 2. Fonction qui déplace un animal de cage

```
1 private static void deplacerAnimal() throws SQLException {
2     System.out.println("Quel animal voulez-vous déplacer ? (saisir 'liste',
3                         pour afficher la liste des animaux)");
4     String animal= LectureClavier.lireChaine();
5     if (animal.equals("liste")) {
6         listeAnimaux();
7         System.out.println("Alors ?");
8         animal= LectureClavier.lireChaine();
9     }
10    int numCage = LectureClavier.lireEntier("Dans quelle cage ?");
11    Statement stmt = conn.createStatement();
12
13    try {
14        //UPDATE LesAnimaux SET NB_MALADIES = 2 WHERE nomA = 'Milou';
15        stmt.executeUpdate("UPDATE LesAnimaux SET nocage = " + numCage +
16                           " WHERE nomA = '" + animal+"'");
17        listeAnimaux();
18    } catch (SQLException e) {
19        System.out.println("La mise à jour a échoué");
20        displayError(e);
21    }
22}
```

## 3. Fonction qui ajoute une maladie à un animal

```
1 private static void ajouterMaladie() throws SQLException {
2     Statement stmt = conn.createStatement();
3     listeAnimaux();
4     System.out.println("Ajouter une maladie à qui ?");
5     String animal = LectureClavier.lireChaine();
6     System.out.println("Ajouter quelle maladie à "+animal+" ?");
7     String maladie = LectureClavier.lireChaine();
8     try {
9         stmt.executeUpdate("INSERT INTO LesMaladies VALUES ('"+ animal +"', '"+ +
10                           maladie +"')");
11         listeAnimaux();
12     } catch (SQLException e) {
13         System.out.println("L'insertion a échoué");
14         displayError(e);
15     }
16 }
```

## 2 Partie Triggers

- Le calcul du nombre de maladies pour chaque animal doit être automatisé en fonction de l'ajout ou de la suppression des maladies.

```
1 CREATE OR REPLACE TRIGGER TRG_UPD_MALADIE_Q2_1
2 AFTER INSERT OR DELETE ON LesMaladies
3 FOR EACH ROW
4 Begin
5   IF INSERTING Then
6     UPDATE LesAnimaux SET nb_maladies = nb_maladies+1 WHERE nomA = :new
7     .nomA;
8
9   ELSIF DELETING Then
10    UPDATE LesAnimaux SET nb_maladies = nb_maladies-1 WHERE nomA = :old
11    .nomA;
12
13  END IF;
14 End;
15 /
```

- Incrémente le nombre de maladies de Milou de 1 dans la table lesAnimaux
- Décrémente le nombre de maladies de Milou de 1 dans la table lesAnimaux

```
1 INSERT INTO LesMaladies VALUES ( 'Milou' , 'Bronchiolite' );
2 DELETE FROM LesMaladies WHERE nomA = 'Milou' AND nomM = 'angine' ;
3
```

- Des animaux ne peuvent pas être placés dans une cage dont la fonction est incompatible avec ces animaux. On prendra en compte le fait que des animaux peuvent être ajoutés, mais aussi déplacés d'un cage.

```
CREATE OR REPLACE TRIGGER TRG_CHECK_CAGE_Q2_2
1 AFTER INSERT OR UPDATE OF noCage ON LesAnimaux
2 FOR EACH ROW
3 Declare
4   str VARCHAR2(100);
5 Begin
6   SELECT Fonction INTO str
7   FROM LesCages
8   WHERE noCage = :new.noCage;
9
10  IF str <> :new.fonction_cage THEN
11    RAISE_APPLICATION_ERROR(-20000, 'Cage incompatible avec cet animal.');
12  END IF;
13 End;
14 /
15
```

1. Insertion refusée car on essaye de mettre un dans une cage à fauve (la numéro 11) un animal qui va normalement dans un aquarium.
2. Mise à jour refusée car on essaye de déplacer un ours dans une cage à fauve.

```

1 INSERT INTO LesAnimaux VALUES ('Bob', 'male', 'truite', 'aquarium', 'France'
2   , 2013, 11, 0);
3 UPDATE LesAnimaux SET noCage = 11 WHERE nomA = 'Arthur';

```

3. Des animaux ne peuvent pas être placés dans une cage non gardée.

```

CREATE OR REPLACE TRIGGER TRG_CAGE_GARDEE_Q2_3
1 AFTER INSERT OR UPDATE OF noCage ON LesAnimaux
2 FOR EACH ROW
3 Declare
4   nb NUMBER;
5 Begin
6   SELECT COUNT(*) INTO nb
7   FROM LesGardiens
8   WHERE noCage = :new.noCage;
9
10  IF nb = 0 THEN
11    RAISE_APPLICATION_ERROR(-20001, 'deplacement impossible dans une cage
12      non gardee');
13  END IF;
14 End;
15 /
16

```

Insertion refusée car l'aquarium dans lequel on essaye d'ajouter un nouveau poisson n'est pas gardé.

```

1 INSERT INTO LesAnimaux VALUES ('Bob', 'male', 'truite', 'aquarium', 'France'
2   , 2013, 2, 0);

```

4. Des animaux de type différent ne peuvent pas cohabiter dans une même cage.

```

1 CREATE OR REPLACE TRIGGER TRG_CHECK_TYPE_Q2_4
2 AFTER INSERT OR UPDATE OF noCage ON LesAnimaux
3 Declare
4     verify NUMBER;
5 Begin
6     SELECT COUNT(*) into verify
7     FROM (   SELECT noCage , COUNT(DISTINCT type_an)
8             FROM LesAnimaux
9             GROUP BY noCage
10            HAVING COUNT(DISTINCT type_an) > 1 );
11
12    IF ( verify > 0 ) THEN
13        RAISE_APPLICATION_ERROR(-20002, 'cohabitation impossible entre animaux
14          de types différents');
15    END IF;
16 End;
17 /

```

Mise à jour refusée car on essaye de déplacer un lion dans une autre cage à fauve déjà occupée par des leopards.

```

1 UPDATE LesAnimaux SET noCage = 11 WHERE nomA = 'Charly';
2

```

### 3 Partie Java + Triggers

Chaque requête s'effectue dans un try{} catch{} donc, lorsqu'une erreur est renvoyée par un trigger, on la capture et on affiche le message d'erreur grâce à la fonction displayError(SQLException e).

La requête est donc annulée par le trigger, l'utilisateur peut alors faire un Rollback grâce au menu.