

```

import java.io.*; // pour la fonction lireClavier uniquement
import java.sql.*;

class ExJDBC2 {

    //-----
    static void ex1(Connection c) throws SQLException {
        String name = lireClavier("entrez le nom : ");
        String texte =
            "update personne set age = age + 1 where nom = '"+name+"'";
        // erreur, tester :
        // "update personne set age = age + 1 where nom = "+name;
        System.out.println(texte);
        Statement s = c.createStatement();
        int n;
        n = s.executeUpdate(texte);
        System.out.println("nombre de lignes traitees : "+n);

        n = s.executeUpdate("update personne set age = age + 1");
        System.out.println("nombre de lignes traitees : "+n);
        s.close();

        lireClavier("Pause programme (entrez return) :");
        Connection c2 = DriverManager.getConnection(
            "jdbc:oracle:thin:waller2_a/AQWzxsx34@soracle:1521:dbinfo");
        s = c2.createStatement();
        texte = "insert into t values(42,52)";
        System.out.println(texte);
        s.executeUpdate(texte);
        s.close();
        c2.close();
    }

    //-----
    static void ex2(Connection c) throws SQLException {
        String texte = "select nom, age from personne";
        System.out.println(texte);
        Statement s = c.createStatement();
        ResultSet r = s.executeQuery(texte);
        while (r.next())
            System.out.println(r.getString(1)+" , "+r.getInt(2)+" ans");
        r.close();
        s.close();

        texte = "select nom, age from personne where age > "+25;
        System.out.println(texte);
        s = c.createStatement();
        r = s.executeQuery(texte);
        while (r.next())
            System.out.println(r.getString(1)+" , "+r.getInt(2)+" ans");
        r.close();
        s.close();
    }

    //-----
    static void ex7(Connection c)
        throws SQLException {
        String texte = "select nom, age from personne where age > 40";
        System.out.println(texte);
        Statement s = c.createStatement();
        ResultSet r = s.executeQuery(texte);
        if (r.next())
            System.out.println(r.getString(1)+" , "+r.getInt(2)+" ans");
        else
            System.out.println("vide");
        r.close();
        s.close();
    }

    //-----
    static void ex3(Connection c) throws SQLException {
        String texte = "select seq.nextval from dual";
        System.out.println(texte);
        Statement s = c.createStatement();
    }
}

```

```

    ResultSet r = s.executeQuery(texte);
    r.next();
    System.out.println(r.getInt(1));
    r.close();
    s.close();
}

//-----
static void ex4(Connection c) throws SQLException {
    // variables : PreparedStatement (but : repetitions)

    // comme ex1.a mais plusieurs fois :

    String texte = "update personne set age = age + 1 where nom = ?"; // ? in
    PreparedStatement p = c.prepareStatement(texte);

    p.setString(1, "Riton");
    p.executeUpdate();

    lireClavier("tapez return : ");

    p.setString(1, "Rita");
    p.executeUpdate();

    p.close();

    // comme ex2.b mais plusieurs fois :
    // repetition select : variables :

    texte = "select nom, age from personne where age > ?";
    p = c.prepareStatement(texte);

    p.setInt(1, 22);
    ResultSet r = p.executeQuery();
    while (r.next())
        System.out.println(r.getString(1)+", "+r.getInt(2)+" ans");
    r.close();

    lireClavier("tapez return (sup 26) : ");

    p.setInt(1, 26);
    r = p.executeQuery();
    while (r.next())
        System.out.println(r.getString(1)+", "+r.getInt(2)+" ans");
    r.close();
    // rem : code duplique => boucle sur {22, 26}

    p.close();
}

//-----
static void ex5(Connection c) throws SQLException {
    // appel procedure : CallableStatement

    String texte = "{ call anniversaire('Riton') }";
    CallableStatement cs = c.prepareCall(texte);
    cs.execute();
    cs.close();

    lireClavier("tapez return 1 :");

    String n = "Riton";
    texte = "{ call anniversaire('"+n+"') }";
    System.out.println(texte);
    cs = c.prepareCall(texte);
    cs.execute();
    cs.close();

    lireClavier("tapez return 2 :");

    // ex. PL/SQL : ... n int; begin n := age('Rita'); end;
    texte = "{ ? = call age('Rita') }";
}

```

```

cs = c.prepareCall(texte);
cs.registerOutParameter(1, Types.INTEGER); // si chaine : Types.VARCHAR
cs.execute();
System.out.println(cs.getInt(1)); // "comme dans curseur"
cs.close();

lireClavier("tapez return 3 :");

texte = "{ call age_out('Rita',?) }";
// rem : exactement pareil (5 lignes) que ci-dessus :
cs = c.prepareCall(texte);
cs.registerOutParameter(1, Types.INTEGER);
cs.execute();
System.out.println(cs.getInt(1));
cs.close();

lireClavier("tapez return 4 :");

texte = "{ call deux_out('Rita',?,?) }";
cs = c.prepareCall(texte);
cs.registerOutParameter(1, Types.INTEGER);
cs.registerOutParameter(2, Types.INTEGER);
cs.execute();
System.out.println(cs.getInt(1));
System.out.println(cs.getInt(2));
cs.close();
}

//-----
static void ex6(Connection c) throws SQLException {
    // repetitions appel procedure
    // fonctionne exactement comme pour PreparedStatement :

    String texte = "{ call anniversaire(?) }";
    CallableStatement cs = c.prepareCall(texte);
    cs.setString(1, "Riton");
    cs.execute();
    cs.setString(1, "Rita");
    cs.execute();
    cs.close();

    lireClavier("ex6 ???");

    texte = "{ ? = call age(?) }";
    cs = c.prepareCall(texte);
    cs.registerOutParameter(1, Types.INTEGER);
    // Rita :
    cs.setString(2, "Rita"); // comme pour PreparedStatement
    cs.execute();
    System.out.println(cs.getInt(1));
    // rebelote pour Riton :
    cs.setString(2, "Riton");
    cs.execute();
    System.out.println(cs.getInt(1));
    cs.close();

    lireClavier("ex6 ???");

    texte = "{ call age_out(?,?) }";
    cs = c.prepareCall(texte);
    cs.registerOutParameter(2, Types.INTEGER);
    cs.setString(1, "Rita");
    cs.execute();
    System.out.println(cs.getInt(2));
    cs.setString(1, "Riton");
    cs.execute();
    System.out.println(cs.getInt(2));
    cs.close();
}

//-----
static String lireClavier(String message) {

```

```

// Dans cette fonction l'exception est catchee pour ne pas avoir a la
// gerer dans le main, et donc mieux voir les exceptions BD.
try {
    System.out.print(message);
    BufferedReader clavier =
        new BufferedReader(new InputStreamReader(System.in));
    return clavier.readLine();
} catch (Exception e) {
    return "erreur dans fonction lireClavier";
}
}

//-----
public static void main(String[] args)
    throws SQLException, ClassNotFoundException {

    System.out.println("coucou1");
    Class.forName("oracle.jdbc.driver.OracleDriver");
    System.out.println("coucou2");
    Connection c = DriverManager.getConnection
        (
            // "jdbc:oracle:thin:ewaller_a/ewaller_a@tp-soracle:1521:dbinfo");
            "jdbc:oracle:thin:c##ewaller_a/ewaller_a@tp-oracle:1522:dbinfo");
    System.out.println("coucou3");

    while (true) {
        System.out.println("-----");
        System.out.println("Bienvenue dans ExJDBC1");
        System.out.println("1 : bases : ");
        System.out.println("2 : sequence : ");
        System.out.println("3 : curseur : ");
        System.out.println("4 : PreparedStatement : ");
        System.out.println("5 : appel procedure : ");
        System.out.println("6 : repetition appel proc : ");
        System.out.println("7 : if : ");
        System.out.println("9 : deconnexion");
        System.out.println("0 : terminer");
        int n = Integer.parseInt(lireClavier("Entrez votre choix : "));
        switch (n) {
            case 1 : ex1(c); break;
            case 2 : ex2(c); break;
            case 3 : ex3(c); break;
            case 4 : ex4(c); break;
            case 5 : ex5(c); break;
            case 7 : ex7(c); break;
            // case 6 : ex6(c); break;
            case 9 : c.close(); break;
            case 0 : return;
        }
    }
}
}

//-----

```