

```

// ElementsCorrJDBC1234.java
// 20 avril 2017
// Tourne le 17 decembre 2013 sur ancienne version corr_pl.sql de B02.
// Attention : non verifie le 20 avril 2017, et pas forcement complet.
//
import java.io.*;
import java.sql.*;

class CorrJDBC {

    //-----
    static Connection connexion(String login, String password,
                               String serveur, String base)
        throws SQLException, ClassNotFoundException {
        System.out.println("ici connexion "+login);
        Connection c = DriverManager.getConnection(
            "jdbc:oracle:thin:@" + login + "/" + password + "@" + serveur + ":1521:" + base);
        return c;
    }

    //-----
    static void deconnexion(Connection c)
        throws SQLException {
        System.out.println("ici deconnexion");
        c.close();
    }

    //-----
    static void creerVillage(Connection c)
        throws SQLException {
        // parametres :
        String v = lireClavier("entrez ville : ");
        String a = lireClavier("entrez activite : ");
        int p = Integer.parseInt(lireClavier("entrez prix : "));
        int cap = Integer.parseInt(lireClavier("entrez capacite : "));

        // construction texte ordre :
        String texte =
            "insert into village values(seq_village.nextval, '"
            + v + "', '" + a + "', " + p + ", " + cap + "')";
        System.out.println(texte);

        // acces serveur :
        Statement stmt = c.createStatement();
        stmt.executeUpdate(texte);
        stmt.close();

        // retours : neant
    }

    //-----
    static void traitement3(Connection c)
        throws SQLException {
        System.out.println("ici traitement3");

        // parametres :
        int jour = Integer.parseInt(lireClavier("entrez jour : "));

        // construction texte ordre :
        String texte = "delete sejour where jourc=" + jour;
        System.out.println(texte);

        // acces serveur :
        Statement stmt = c.createStatement();
        int nb = stmt.executeUpdate(texte);
        stmt.close();

        // retours :
    }

    //-----
    static void traitement1(Connection c)
        throws SQLException {
        System.out.println("ici traitement1");

        // parametres :
        String n = lireClavier("entrez nom : ");
        int a = Integer.parseInt(lireClavier("entrez age : "));

        String texte = "select seq_client.nextval from dual";
        System.out.println(texte);
        Statement s = c.createStatement();
        ResultSet r = s.executeQuery(texte);
        r.next();
        int ic = r.getInt(1);
        r.close();
        // ideal serait un select into au lieu d'un curseur puisqu'il y a
        // exactement une valeur de retour

        texte =
            "insert into client(idc,nom,age) values('"+ic+"','"+n+"','"+a+"')";
        System.out.println(texte);
        // acces serveur :
        Statement stmt = c.createStatement();
        stmt.executeUpdate(texte);
        stmt.close();

        // retour(s) :
        System.out.println("identifiant client : " + ic);
    }

    //-----
    static void traitement2(Connection c)
        throws SQLException {
        System.out.println("ici traitement2");

        // parametres :
        int ic = Integer.parseInt(lireClavier("entrez identifiant client : "));
        String v = lireClavier("entrez ville : ");
        int j = Integer.parseInt(lireClavier("entrez jour : "));

        // initialisation valeurs retour pour cas ou pas de village :
        int iv=-1;
        int is=-1;
        String a="neant";

        int p;
        String texte = "select idv, activite, prix from village"
            + " where ville = '" + v + "' order by prix desc";
        System.out.println(texte);

        Statement s = c.createStatement();
        ResultSet r = s.executeQuery(texte);
        if (r.next()) {
            iv = r.getInt(1); // affectation valeur de retour
            a = r.getString(2); // affectation valeur de retour
            p = r.getInt(3);
            r.close();

            // achat sejour :
            texte = "insert into sejour values(seq_sejour.nextval, '"+ic+"', "
                + "iv', " + p + "')";
            System.out.println(texte);
            s.executeUpdate(texte);
            texte = "update client set avoir=avoir-'"+p+" where idc='"+ic;
            System.out.println(texte);
            s.executeUpdate(texte);

            // affectation valeur de retour :
            texte = "select seq_sejour.currval from dual";
            // ideal serait select into au lieu curseur, comme dans traitement1
        }
    }
}

```

```

        System.out.println("Retour : nombre de sejours detruits : " + nb);
    }

    //-----
    static void traitement3TryCatch(Connection c) {
        System.out.println("ici traitement3TryCatch");
        int jour = Integer.parseInt(lireClavier("entrez jour : "));
        String texte = "delete sejour where jourc=" + jour;
        System.out.println(texte);
        try {
            Statement stmt = c.createStatement();
            int nb = stmt.executeUpdate(texte);
            stmt.close();
            System.out.println("Retour : nombre de sejours detruits : " + nb);
        } catch (SQLException e) {
            System.out.println("argl ! : ");
            System.err.println(e);
        }
    }

    //-----
    static void consulterVillagesEmploye(Connection c)
        throws SQLException {
        System.out.println("ici consulterVillagesEmploye");
        String texte = "select * from village";
        System.out.println(texte);
        Statement stmt = c.createStatement();
        ResultSet r = stmt.executeQuery(texte);
        while (r.next())
            System.out.println(r.getInt(1) + ", " + r.getString(2) + ", "
                + r.getString(3) + ", " + r.getInt(4) + ", "
                + r.getInt(5));
        r.close();
        stmt.close();
    }

    //-----
    static void consulterVillagesClient(Connection c)
        throws SQLException {
        System.out.println("ici consulterVillagesClient");
        int ic = Integer.parseInt(lireClavier("entrez identifiant client : "));
        String texte =
            "select village.idv, ville, activite, prix, capacite\n"
            + "from village, sejour\n"
            + "where village.idv = sejour.idv and idc = " + ic;
        System.out.println(texte);
        Statement stmt = c.createStatement();
        ResultSet r = stmt.executeQuery(texte);
        while (r.next())
            System.out.println(r.getInt(1) + ", " + r.getString(2) + ", "
                + r.getString(3) + ", " + r.getInt(4) + ", "
                + r.getInt(5));
        r.close();
        stmt.close();
    }

    //-----
    static void consulterVillagesSansSejoursClient(Connection c)
        throws SQLException {
        System.out.println("ici consulterVillagesSansSejoursClient");
        String texte = "select * from vue_village_sans_sejour";
        System.out.println(texte);
        Statement stmt = c.createStatement();
        ResultSet r = stmt.executeQuery(texte);
        while (r.next())
            System.out.println(r.getInt(1) + ", " + r.getString(2) + ", "
                + r.getString(3) + ", " + r.getInt(4));
        r.close();
        stmt.close();
    }

    //-----
    // retours :
    System.out.println("village "+iv+", sejour "+is+", activite "+a);
}

//-----
static void traitement3Fun(Connection c)
    throws SQLException {
    int j = Integer.parseInt(lireClavier("entrez le jour : "));
    CallableStatement cs = c.prepareCall("{? = call traitement3(?,?)");
    cs.registerOutParameter(1, Types.INTEGER);
    cs.execute();
    int nb = cs.getInt(1);
    cs.close();
    System.out.println(" nombre de sejours detruits : " + nb);
}

//-----
static void traitement1Fun(Connection c)
    throws SQLException {
    CallableStatement cs =
        c.prepareCall("{? = call traitement1('"+
            lireClavier("nom client : ") + "','" +
            Integer.parseInt(lireClavier("age : ")) + "')");
    cs.registerOutParameter(1, Types.INTEGER);
    cs.execute();
    System.out.println("identifiant client : " + cs.getInt(1));
    cs.close();
}

//-----
static void traitement2Proc(Connection c)
    throws SQLException {
    int idc = Integer.parseInt(lireClavier("identifiant client : "));
    String ville = lireClavier("ville ? ");
    int jour = Integer.parseInt(lireClavier("jour ? "));
    CallableStatement cs =
        c.prepareCall("{call traitement2(?, '"+ville+"', " + jour +
            ", ?, ?)}");
    cs.registerOutParameter(1, Types.INTEGER);
    cs.registerOutParameter(2, Types.INTEGER);
    cs.registerOutParameter(3, Types.VARCHAR);
    cs.execute();
    int idvcs = cs.getInt(1);
    int idscs = cs.getInt(2);
    String activite = cs.getString(3);
    cs.close();
    System.out.println("identifiants village "+idvcs+", sejour "+idscs
        + ", activite "+activite);
}

```



```
//
// ExRefcursseurs.java
//

import java.sql.*;
import oracle.jdbc.OracleTypes; // cause ci-dessous OracleTypes

class ExRefcursseurs {
    public static void main(String[] args)
        throws SQLException, ClassNotFoundException {

        Class.forName("oracle.jdbc.driver.OracleDriver");
        Connection c = DriverManager.getConnection
            ("jdbc:oracle:thin:cn#ewaller_a/ewaller_a@tp-oracle:1522:dbinfo");

        String texte = "{ ? = call ex_ref_cursor_fun(23) }";
        CallableStatement cs = c.prepareCall(texte);
        cs.registerOutParameter(1, OracleTypes.CURSOR); // on aurait écrit Types.CURSOR
        cs.execute();
        ResultSet r = (ResultSet) cs.getObject(1); // on aurait écrit cs.getResultSet(1)
        // Un objet c'est une adresse ; getObject renvoie adresse, puis on dit que
        // l'objet recupere est de la classe ResultSet.
        // Ci-dessous on parcourt et affiche le ResultSet independamment de comment
        // il a ete rempli :
        while (r.next())
            System.out.println(r.getString(1)+"", "+r.getInt(2)+" ans");
        r.close();
        cs.close();
        c.close();
    }
}
```

-UU-(DOS)----F1 ExRefcursseurs.java Top L14 (Java/L Abbrev)

```

fA@vr. 21, 17 18:26      ex_pl_refcursseurs.sql.txt      Page 1/1
-----
-- ex_pl_refcursseurs.sql
-- tourné le 21 fevrier 2017

-- Base pour l'exemple : personne(nom, age, activite) dans ex pl bases.sql

/* Le but d'un ref cursor qui nous interesse :
- si une table(s) est encapsulee, une procedure privee peut evaluer une requete
dessus et en renvoyer le resultat, sans que l'appelant connaisse la table
(confidentialite).
- De plus l'appelant doit juste savoir parcourir un curseur, pas faire une
requete (architecture en couches). */

create or replace function ex_ref_cursor_fun(a personne.age%type)
return sys_refcursor
is
    c sys_refcursor;
begin
    open c for
        select nom, age
        from personne
        where age < a;
    return c;
end;

create or replace procedure aff_ref_cursor(a personne.age%type) is
    c sys_refcursor;
    x personne.nom%type; -- rem : il faut connaitre les colonnes du select
    y personne.age%type;
begin
    -- remplace le open c :
    c := ex_ref_cursor_fun(a); -- zone renvoyee par fonction mise dans c
    -- encapsulation respectee
    -- ex_ref_cursor_proc(n, c); -- pour variante ci-dessous via variable out
    loop
        fetch c into x, y;
        exit when c%notfound;
        dbms_output.put_line(x||' '||y);
    end loop;
end;

select* from personne;
set serveroutput on
exec aff_ref_cursor(30);

-- Variante avec variable out pour s'entraîner (mais fonction plus elegante) :

create or replace procedure ex_ref_cursor_proc (
    a personne.age%type,
    c out sys_refcursor) is
begin
    open c for
        select nom, age
        from personne
        where age < a;
end;
/

```

ex_pl_refcursseurs

mardi fA@vr. 21, 2017