Océane Muller
Launois Camille

PROJET SQL EASY CAR

Partie A : Création de la base de données :

1. Création de la base :

Questions A1 & A2 :

```
create table PAYS(
codePays number(2),
nomPays varchar(25),
primary key(codePays));

create table CLIENT(
numClient varchar(6),
civilite varchar(5),
nom varchar(20),
prenom varchar(15),
codePostal varchar(5),
ville varchar(50),
mail varchar(50),
dateNaiss date,
numPermis varchar(6),
dateObtention date,
nubCB varchar(5),
pointBonus number(4),
codePays number(3),
primary key(numClient),
foreign key(codePays) references PAYS(codePays));

create table AGENCE(
numAgence varchar(8),
NomAgence varchar(35),
telAgence varchar(11),
villeAgence varchar(50),
codePays number(3),
primary key(numAgence),
foreign key(codePays) references PAYS(codePays));

create table JOUR(
codeJour number(1),
libelleJour varchar(10),
primary key(codeJour)) ;
```

```
create table RESERVATION(
numReserv varchar(6),
dateReserv date,
dateDep date,
dateRetour date,
prixEstime number(10,2),
numClient varchar(6),
numAgence varchar(8),
numVehic varchar(10),
RemiseVehic number(1),
primary key(numReserv),
foreign key(numClient) references CLIENT(numClient),
foreign key(numAgence) references AGENCE(numAgence),
foreign key(numVehic) references VEHICULE(numVehic));

create table CATEGORIE(
codeCateg varchar(4),
libelleCateg varchar(15),
nbPorte number(1),
nbPassager varchar(2),
nbBag number(2),
PrixJour number(3),
Clim varchar(1),
primary key(codeCateg));

create table VEHICULE(
numVehic varchar(10),
codeCateg varchar(4),
numAgence varchar(8),
primary key(numVehic),
foreign key(codeCateg) references CATEGORIE(codeCateg),
foreign key(numAgence) references AGENCE(numAgence));

create table FACTURE(
numFact varchar(5),
dateFact date,
montFact number(5,2),
remise number(5,2),
numReserv varchar(6),
Regle varchar(1),
primary key(numFact),
foreign key(numReserv) references RESERVATION(numReserv));

create table OUVERTE(
numAgence varchar(8),
codeJour number(1),
heureOuv date,
HeureFerm date,
```

```
primary key(numAgence,codeJour),
foreign key(numAgence) references AGENCE(numAgence),
foreign key(codeJour) references JOUR(codeJour));

create table LIER(
numAgence1 varchar(8),
numAgence2 varchar(8),
primary key(numAgence1 , numAgence2),
foreign key(numAgence1) references AGENCE(numAgence),
foreign key(numAgence2) references AGENCE(numAgence));
```

2. Exploitation de la base de données :

Question A3 :

```
select *
from agence
where numAgence = (
   select numAgence
   from reservation
   group by numAgence
   having count(*) >= all (
      select count(*)
      from reservation
      group by numAgence));
```

Question A4 :

```
select nom, codePostal, ville
from Client
where numclient not in(select numclient
from Reservation
where datereserv like '%/02/%');
```

Il existe aussi une méthode avec minus :

```
select nom, codePostal, ville
from client
where numclient in(select numclient
from client
minus
select numclient
from Reservation
where datereserv like '%/02/%');
```

Partie B : Programmation sous ORACLE en PL/SQL:

Question B1 :

```
create or replace function vehiculedispo(p_categ varchar, p_datedeb date, p_datefin date)
return varchar
is
v_res varchar(20);

cursor lvehiculedispo is
select *
from Vehicule
where numVehic not in(select numVehic
from vehicule natural join Reservation
where dateretour >= p_datefin
or datedep <= p_datedeb
and numvehic not in(
select numvehic
from vehicule natural join reservation
where dateretour <= p_datefin
or datedep >= p_datedeb))
and codecateg = p_categ;

BEGIN
for v_vl in lvehiculedispo loop
   v_res := v_vl.numvehic;
   exit when lvehiculedispo%rowcount=1;
end loop;

return v_res;

EXCEPTION
when no_data_found then return('null');

END vehiculedispo;
```

PL/SQL AFIN DE TESTER LA FONCTION :

```
DECLARE
resultat varchar(20);

BEGIN
resultat := vehiculedispo('C005','10/03/20','18/03/20');
dbms_output.put_line(resultat);

END;
```

Question B2 :

```
create sequence seqreserv
start with 100
increment by 1;

create or replace procedure AddReserv (p_dateReserv date, p_dateDep date, p_dateRetour date,
p_numClient varchar, p_numAgence varchar, p_numVehic varchar, p_remiseVehic number, p_sortie
out number)
is
v_numReserv varchar(6);
v_prixEstime number(10,2);

BEGIN
p_sortie := 0;
v_numReserv := ('R00' || to_char(seqreserv.nextval));

select PrixJour * (p_dateretour - p_dateDep) into v_prixEstime
from vehicule inner join categorie
on vehicule.codecateg = categorie.codecateg
where vehicule.numVehic = p_numVehic;

insert into reservation values (v_numReserv, p_dateReserv, p_dateDep, p_dateRetour, v_prixEstime,
p_numClient, p_numAgence, p_numVehic, p_remiseVehic);

EXCEPTION
when no_data_found then p_sortie := 1;
dbms_output.put_line('La reservation n a pas pu etre effectuee');

END AddReserv;
```

PL/SQL AFIN DE TESTER LA PROCEDURE :

```
DECLARE
v_sortie number(1);
v_numReserv varchar(6);
v_dateReserv date;
v_dateDep date;
v_dateRetour date;
v_prixEstime number(10,2);
v_numClient varchar(6);
v_numAgence varchar(8);
v_numVehic varchar(10);
v_RemiseVehic number(1);

BEGIN
v_dateReserv := '11/10/19';
v_dateDep := '11/10/20';
v_dateRetour := '12/10/21';
v_numClient := 'CL0001';
```

```
v_numAgence := 'AG010024';
v_numVehic :='100XA10' ;
v_RemiseVehic := 1;

AddReserv(v_dateReserv,v_dateDep,v_dateRetour,v_numClient,v_numAgence,v_numVehic,v_Remis
eVehic,v_sortie );

END ;
```

Question B3 :

```
create or replace function devis(p_categ varchar, p_datedeb date, p_datefin date)
return number
is
v_devis number(6,2);
dateretour date;
datedep date;

BEGIN
select (p_datefin - p_datedeb)*prixjour into v_devis
from Categorie
where codecateg = p_categ;

return v_devis;

END devis;
```

PL/SQL AFIN DE TESTER LA FONCTION :

```
DECLARE
v_devis number(6,2);

BEGIN
v_devis := devis('C001','12/02/20','24/02/20');
dbms_output.put_line('Le montant du devis est de : ' || v_devis);

END;
```

Partie C : Programme en Java exploitant une base de données :

Question C1 :

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

public class projetC1 {
```

```
public static void main(String[] args) throws Exception {
   try
   {
      String url = "jdbc:oracle:thin:@charlemagne.iutnc.univ-lorraine.fr:1521:infodb";
       Connection cnt = DriverManager.getConnection(url,"launois15u", "Basile2499");

       String reqVille = "select distinct villeAgence from agence order by villeAgence ASC";
       String reqAgence = "select nomAgence, telAgence from agence where villeAgence = ?";
       String reqCode = "select distinct codeCateg from categorie";
       String reqCateg = "select distinct libelleCateg, PrixJour from categorie inner join vehicule on
       categorie.codecateg = vehicule.codecateg inner join agence on agence.numagence =
       vehicule.numagence where vehicule.codeCateg = ?" ;
       String reqVehic = "select numVehic from categorie inner join vehicule on categorie.codecateg
       = vehicule.codecateg where vehicule.codecateg = ?";

      PreparedStatement p_ville = cnt.prepareStatement(reqVille);
      PreparedStatement p_agence = cnt.prepareStatement(reqAgence);
      PreparedStatement p_code = cnt.prepareStatement(reqCode);
      PreparedStatement p_categ = cnt.prepareStatement(reqCateg);
      PreparedStatement p_vehic = cnt.prepareStatement(reqVehic);

      ResultSet resp_ville = p_ville.executeQuery();

      while(resp_ville.next())
      {
         String ville = resp_ville.getString("villeAgence");
         System.out.println(ville);

         p_agence.setString(1, ville);
         ResultSet resp_agence = p_agence.executeQuery();

         while (resp_agence.next())
         {
            String nom = resp_agence.getString("nomAgence");
            System.out.print("  " + nom + " ");

            String tel = resp_agence.getString("telAgence");
            System.out.println(tel);

            ResultSet resp_code = p_code.executeQuery();

            while(resp_code.next())
            {
               String code = resp_code.getString("codeCateg");

               p_categ.setString(1, code);
               ResultSet resp_categ = p_categ.executeQuery();
```

```
                while (resp_categ.next())
                {
                   String libelle = resp_categ.getString("libelleCateg");
                   double prix = resp_categ.getDouble("PrixJour");

                   System.out.println("\t" + code + " " + libelle + " " + prix + " ");

                   p_vehic.setString(1, code);
                   ResultSet resp_vehic = p_vehic.executeQuery();

                   while(resp_vehic.next())
                   {
                      String numvehic = resp_vehic.getString("numVehic");

                      System.out.println("\t\t" + " - " + numvehic + " " + "\n");
                   }

                   System.out.println("");
                   resp_vehic.close();
                }
                resp_categ.close();
             }
             resp_code.close();
          }
          resp_agence.close();
       }
       resp_ville.close();
       p_categ.close();
       p_vehic.close();
       p_ville.close();
       cnt.close();
    }

    catch(Exception e)
    {
       System.out.println("An error occured");
       e.printStackTrace();
    }
  }
}
```

Question C2 :

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

public class TESTC2 {
```

```
   public static void main(String[] args) throws Exception {
   String url = "jdbc:oracle:thin:@charlemagne:1521:infodb";
   Connection cnt = DriverManager.getConnection(url, "muller664u", "Poulet58");

   CallableStatement stmt = cnt.prepareCall("{ ? = call devis(?,?,?)}");
   stmt.registerOutParameter(1, Types.DECIMAL, 2);
   Scanner sc = new Scanner(System.in);
   System.out.println("Veuillez renseigner le type de categorie :");
   String categ = sc.nextLine();
   stmt.setString(2,categ);

   System.out.println("Veuillez renseigner la date de debut :");
   String datedebut = sc.nextLine();
   stmt.setString(3,datedebut);

   System.out.println("Veuillez renseigner la date de fin :");
   String datefin = sc.nextLine();
   stmt.setString(4,datefin);

   stmt.execute();
   double devis = stmt.getDouble(1);

   System.out.println("Votre devis est de : " + devis + " euros");

   }
}
```

Question C3 :

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

public class TestJDBC {

 public static void main(String[] args) throws Exception {
 String url = "jdbc:oracle:thin:@charlemagne:1521:infodb";
 Connection cnt = DriverManager.getConnection(url, "muller664u", "Poulet58");

  CallableStatement stmt = cnt.prepareCall("{ ? = call vehiculedispo(?,?,?)}");
  stmt.registerOutParameter(1, Types.CHAR);
  Scanner sc = new Scanner(System.in);
  System.out.println("Veuillez renseigner le type de categorie :");
  String categ = sc.nextLine();
  stmt.setString(2,categ);

  System.out.println("Veuillez renseigner la date de debut :");
```

```
 String datedebut = sc.nextLine();
 stmt.setString(3,datedebut);

 System.out.println("Veuillez renseigner la date de fin :");
 String datefin = sc.nextLine();
 stmt.setString(4,datefin);

 stmt.execute();
 String dispo = stmt.getString(1);

          if (dispo==null){
                System.out.println("Aucun vehicule disponible");
          } else {
                System.out.println("L'immatriculation du premier vehicule est : "+dispo);
                }
          }
}
```

Question C4 :

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

public class TestJDBC{

   public static void main(String []args) throws Exception {

      Connection cnt =
DriverManager.getConnection("jdbc:oracle:thin:@charlemagne:1521:infodb","muller664u","Poulet5
8");
      CallableStatement stmt = cnt.prepareCall("{ call AddReserv(?,?,?,?,?,?,?,?)}");
      Scanner sc = new Scanner(System.in);
      System.out.println("Veuillez renseigner la date de reservation :");
      String datereserv = sc.nextLine();
      stmt.setString(1,datereserv);
      System.out.println("Veuillez renseigner la date de départ :");
      String datedep = sc.nextLine();
      stmt.setString(2,datedep);
      System.out.println("Veuillez renseigner la date de retour :");
      String dateretour = sc.nextLine();
      stmt.setString(3,dateretour);
      System.out.println("Veuillez renseigner le numero du client :");
      String numcl = sc.nextLine();
      stmt.setString(4,numcl);
      System.out.println("Veuillez renseigner le numero de l'agence :");
      String numag = sc.nextLine();
      stmt.setString(5,numag);
```

```
        System.out.println("Veuillez renseigner le numero du vehicule :");
        String numvehic = sc.nextLine();
        stmt.setString(6,numvehic);
        System.out.println("Veuillez renseigner la remise de vehicule :");
        int remvehic = sc.nextInt();
        stmt.setInt(7,remvehic);
        stmt.registerOutParameter(8, Types.NUMERIC);
        stmt.execute();
        int sortie = stmt.getInt(8);

        if (sortie==0){
           System.out.println("La reservation a ete enregistree");
        } else {
           System.out.println("La reservation n'a pas pu avoir lieu");
        }
     }
}
```

Partie D : Synthèse :

Question D1 :

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

public class Synthese {

   public static void main(String[] args) throws Exception {
      try
      {
         Scanner sc = new Scanner(System.in);
         String url = "jdbc:oracle:thin:@charlemagne.iutnc.univ-lorraine.fr:1521:infodb";
         Connection cnt = DriverManager.getConnection(url,"launois15u", "Basile2499");

         System.out.println("\n" + "          Agence EasyCar         " + "\n");
         System.out.println("       1 - Etablir un devis       "+ "\n");
         System.out.println("       2 - Verifier la disponibilite d'un vehicule"+ "\n");
         System.out.println("       3 - Ajouter une reservation "+ "\n");
         System.out.println("       4 - Supprimer une reservation" + "\n");
         System.out.println("       5 - Afficher une reservation" + "\n");
         System.out.println("       6 - Afficher le catalogue   " + "\n");
         System.out.println("       7 - Quitter                 " + "\n");
         System.out.println("             Choix :           " + "\n");

         int choix = 0;
```

```
while(choix!=7){

   System.out.println("Que voulez vous choisir a present : " + "\n");
   choix = sc.nextInt();
   String blanc = sc.nextLine();

   switch(choix){

      case 1 :

         CallableStatement stmt = cnt.prepareCall("{ ? = call devis(?,?,?)}");
         stmt.registerOutParameter(1, Types.DECIMAL, 2);

         System.out.println("Veuillez renseigner le type de categorie :");
         String categ = sc.nextLine();

         System.out.println("Veuillez renseigner la date de debut :");
         String datedebut = sc.nextLine();

         System.out.println("Veuillez renseigner la date de fin :");
         String datefin = sc.nextLine();

         stmt.setString(2,categ);
         stmt.setString(3,datedebut);
         stmt.setString(4,datefin);

         stmt.execute();
         double devis = stmt.getDouble(1);

         System.out.println("Votre devis est de : " + devis + " euros" + "\n");

      break;

      case 2 :

         CallableStatement stmt2 = cnt.prepareCall("{ ? = call vehiculedispo(?,?,?)}");
         stmt2.registerOutParameter(1, Types.CHAR);

         System.out.println("Veuillez renseigner le type de categorie :");
         String categ2 = sc.nextLine();
         stmt2.setString(2,categ2);

         System.out.println("Veuillez renseigner la date de debut :");
         String datedebut2 = sc.nextLine();
         stmt2.setString(3,datedebut2);

         System.out.println("Veuillez renseigner la date de fin :");
         String datefin2 = sc.nextLine();
```

```
    stmt2.setString(4,datefin2);

    stmt2.execute();
    String dispo2 = stmt2.getString(1);

    if (dispo2==null){
    System.out.println("Aucun vehicule disponible" + "\n");
    } else {
    System.out.println("L'immatriculation du premier vehicule est : " + dispo2 + "\n");
    }

break;

case 3 :

    CallableStatement stmt3 = cnt.prepareCall("{ call AddReserv(?,?,?,?,?,?,?,?)}");

    System.out.println("Veuillez renseigner la date de reservation :");
    String datereserv3 = sc.nextLine();
    stmt3.setString(1,datereserv3);

    System.out.println("Veuillez renseigner la date de depart :");
    String datedep3 = sc.nextLine();
    stmt3.setString(2,datedep3);

    System.out.println("Veuillez renseigner la date de retour :");
    String dateretour3 = sc.nextLine();
    stmt3.setString(3,dateretour3);

    System.out.println("Veuillez renseigner le numero du client :");
    String numcl3 = sc.nextLine();
    stmt3.setString(4,numcl3);

    System.out.println("Veuillez renseigner le numero de l'agence :");
    String numag3 = sc.nextLine();
    stmt3.setString(5,numag3);

    System.out.println("Veuillez renseigner le numero du vehicule :");
    String numvehic3 = sc.nextLine();
    stmt3.setString(6,numvehic3);

    System.out.println("Veuillez renseigner la remise de vehicule :");
    int remvehic3 = sc.nextInt();
    stmt3.setInt(7,remvehic3);

    stmt3.registerOutParameter(8, Types.NUMERIC);
    stmt3.execute();
    int sortie3 = stmt3.getInt(8);
```

```
                /**
                Recherche du numero de reservation crée
                */

                PreparedStatement p_trouve3 = cnt.prepareStatement("select numreserv from
reservation where datedep = ? and dateretour = ? and numclient = ? and numagence = ? and
numvehic = ?");
                p_trouve3.setString(1, datedep3);
                p_trouve3.setString(2, dateretour3);
                p_trouve3.setString(3, numcl3);
                p_trouve3.setString(4, numag3);
                p_trouve3.setString(5, numvehic3);

                ResultSet res_trouve3 = p_trouve3.executeQuery();

                while(res_trouve3.next()){

                   String numres3 = res_trouve3.getString("numreserv");

                   if (sortie3==0){
                      System.out.println("La reservation a ete enregistree sous le numero de reservation
: " + numres3 + "\n");
                   } else {
                      System.out.println("La reservation n'a pas pu avoir lieu" + "\n");
                   }
                }

                res_trouve3.close();
                p_trouve3.close();


             break;

             case 4 :

                System.out.println("Veuillez entrez le numero de reservation : ");
                String numreserv4 = sc.nextLine();

                CallableStatement req4 = cnt.prepareCall("{call suppreserv(?,?)}");
                req4.setString(1, numreserv4);

                req4.registerOutParameter(2, Types.INTEGER);
                req4.execute();

                int sortie4 = req4.getInt(2);

                if (sortie4==0){
```

```
                    System.out.println("La reservation " + numreserv4 + " a bien ete supprime." + "\n");
                } else {
                    System.out.println("La suppression n'a pas pu avoir lieu car le numero de reservation
n'existe pas" + "\n");
                }


            break;

            case 5 :

                System.out.println("Veuillez entrer le numero de votre reservation s'il vous plait : " +
"\n");

                String numres5 = sc.nextLine();

                CallableStatement reserv5 = cnt.prepareCall("{ ? = call reservfact(?)}");
                reserv5.registerOutParameter(1, Types.DECIMAL);
                reserv5.setString(2, numres5);
                reserv5.execute();
                int existe5 = reserv5.getInt(1);

                if (existe5==0){

                    CallableStatement reserv51 = cnt.prepareCall("{ ? = call reserv(?)}");
                    reserv51.registerOutParameter(1, Types.DECIMAL);
                    reserv51.setString(2, numres5);
                    reserv51.execute();
                    int existe51 = reserv51.getInt(1);

                    if (existe51==0){
                        System.out.println("La reservation n'existe pas" + "\n");

                    } else {

                        CallableStatement afficher51 = cnt.prepareCall("{ call affichres(?,?,?,?,?,?,?,?,?)}");
                        afficher51.setString(1, numres5);
                        afficher51.registerOutParameter(2, Types.VARCHAR);
                        afficher51.registerOutParameter(3, Types.VARCHAR);
                        afficher51.registerOutParameter(4, Types.VARCHAR);
                        afficher51.registerOutParameter(5, Types.VARCHAR);
                        afficher51.registerOutParameter(6, Types.VARCHAR);
                        afficher51.registerOutParameter(7, Types.VARCHAR);
                        afficher51.registerOutParameter(8, Types.INTEGER);
                        afficher51.registerOutParameter(9, Types.VARCHAR);

                        afficher51.execute();
```

```
                    String numclient51 = afficher51.getString(2);
                    String nom51 = afficher51.getString(3);
                    String prenom51 = afficher51.getString(4);
                    String datedep51 = afficher51.getString(5);
                    String datefin51 = afficher51.getString(6);
                    String codecateg51 = afficher51.getString(7);
                    int montant51 = afficher51.getInt(8);
                    String regle51 = afficher51.getString(9);

                    System.out.println(" Le numero du client est : " + numclient51 + "\n");
                    System.out.println(" Le nom et le prenom du client sont : " + nom51 + " " +
prenom51 + "\n");
                    System.out.println(" La date de depart est : " + datedep51 + "\n");
                    System.out.println(" La date de retour est : " + datefin51 + "\n");
                    System.out.println(" Le code de categorie de la voiture est : " + codecateg51 +
"\n");
                    System.out.println(" Le montant de la location est : " + montant51 + "\n");
                    System.out.println(" La somme est regle (o) sinon (n) : " + regle51 + "\n");
                }

            } else {
                CallableStatement afficher5 = cnt.prepareCall("{ call affichfact(?,?,?,?,?,?,?,?,?)}");
                afficher5.setString(1, numres5);
                afficher5.registerOutParameter(2, Types.VARCHAR);
                afficher5.registerOutParameter(3, Types.VARCHAR);
                afficher5.registerOutParameter(4, Types.VARCHAR);
                afficher5.registerOutParameter(5, Types.VARCHAR);
                afficher5.registerOutParameter(6, Types.VARCHAR);
                afficher5.registerOutParameter(7, Types.VARCHAR);
                afficher5.registerOutParameter(8, Types.INTEGER);
                afficher5.registerOutParameter(9, Types.VARCHAR);

                afficher5.execute();

                String numclient5 = afficher5.getString(2);
                String nom5 = afficher5.getString(3);
                String prenom5 = afficher5.getString(4);
                String datedep5 = afficher5.getString(5);
                String datefin5 = afficher5.getString(6);
                String codecateg5 = afficher5.getString(7);
                int montant5 = afficher5.getInt(8);
                String regle5 = afficher5.getString(9);

                System.out.println(" Le numero du client est : " + numclient5 + "\n");
                System.out.println(" Le nom et le prenom du client sont : " + nom5 + " " + prenom5 +
"\n");
                System.out.println(" La date de depart est : " + datedep5 + "\n");
                System.out.println(" La date de retour est : " + datefin5 + "\n");
```

```
                System.out.println(" Le code de categorie de la voiture est : " + codecateg5 + "\n");
                System.out.println(" Le montant de la location est : " + montant5 + "\n");
                System.out.println(" La somme est regle (o) sinon (n) : " + regle5 + "\n");}

            break;

            case 6 :

                String reqVille6 = "select distinct villeAgence from agence order by villeAgence ASC";
                String reqAgence6 = "select nomAgence, telAgence from agence where villeAgence =
?";
                String reqCode6 = "select distinct codeCateg from categorie";
                String reqCateg6 = "select distinct libelleCateg, PrixJour from categorie inner join
vehicule on categorie.codecateg = vehicule.codecateg inner join agence on agence.numagence =
vehicule.numagence where vehicule.codeCateg = ?" ;
                String reqVehic6 = "select numVehic from categorie inner join vehicule on
categorie.codecateg = vehicule.codecateg where vehicule.codecateg = ?";

                PreparedStatement p_ville6 = cnt.prepareStatement(reqVille6);
                PreparedStatement p_agence6 = cnt.prepareStatement(reqAgence6);
                PreparedStatement p_code6 = cnt.prepareStatement(reqCode6);
                PreparedStatement p_categ6 = cnt.prepareStatement(reqCateg6);
                PreparedStatement p_vehic6 = cnt.prepareStatement(reqVehic6);

                ResultSet resp_ville6 = p_ville6.executeQuery();

                while(resp_ville6.next())
                {
                   String ville6 = resp_ville6.getString("villeAgence");
                   System.out.println(ville6);

                   p_agence6.setString(1, ville6);
                   ResultSet resp_agence6 = p_agence6.executeQuery();

                   while (resp_agence6.next())
                   {
                      String nom6 = resp_agence6.getString("nomAgence");
                      System.out.print("  " + nom6 + " ");

                      String tel6 = resp_agence6.getString("telAgence");
                      System.out.println(tel6);

                      ResultSet resp_code6 = p_code6.executeQuery();

                      while(resp_code6.next())
                      {
                         String code6 = resp_code6.getString("codeCateg");
```

```
                    p_categ6.setString(1, code6);
                    ResultSet resp_categ6 = p_categ6.executeQuery();

                    while (resp_categ6.next())
                    {
                       String libelle6 = resp_categ6.getString("libelleCateg");
                       double prix6 = resp_categ6.getDouble("PrixJour");

                       System.out.println("\t" + code6 + " " + libelle6 + " " + prix6 + " ");

                       p_vehic6.setString(1, code6);
                       ResultSet resp_vehic6 = p_vehic6.executeQuery();

                       while(resp_vehic6.next())
                       {
                          String numvehic6 = resp_vehic6.getString("numVehic");

                          System.out.println("\t\t" + " - " + numvehic6 + " " + "\n");
                       }

                       System.out.println("");
                       resp_vehic6.close();
                    }
                    resp_categ6.close();
                 }
                 resp_code6.close();
              }
              resp_agence6.close();
           }
           resp_ville6.close();
           p_categ6.close();
           p_vehic6.close();
           p_ville6.close();

        break;

        case 7 :

           System.out.println("\n" + "Au revoir et a tres bientot ;) \n");
           System.out.println("MULLER Oceane, LAUNOIS Camille \n");
           System.exit(0);

        break;
     }
  }
}

catch(Exception e)
```

```
      {
         System.out.println("Une erreur s'est produite \n");
         e.printStackTrace();
      }
   }
}
```

Fonctions PL/SQL utilisée pour le cas 5 :

```
create or replace function reservfact (p_numreserv VARCHAR)
return number
IS
v_reserv number(1);

BEGIN
select count(*) into v_reserv
from Facture
where numreserv = p_numreserv;
return v_reserv;

EXCEPTION
when no_data_found then v_reserv := 0;

END;
```

Cette fonction permet de déterminer si le numéro de réservation est présent dans la table facture.

```
create or replace function reserv (p_numreserv VARCHAR)
return number
IS
v_reserv number(1);

BEGIN
select count(*) into v_reserv
from Reservation
where numreserv = p_numreserv;
return v_reserv;

EXCEPTION
when no_data_found then v_reserv := 0;

END;
```

Cette fonction permet de déterminer si le numéro de réservation est présent dans la table réservation.

Procédures PL/SQL utilisée pour le cas 5 :

```
create or replace procedure affichefact(p_numreserv varchar2, p_numclient out varchar2, p_nom
out varchar2, p_prenom out varchar2, p_datedep out date, p_datefin out date, p_codecateg out
varchar2, p_montant out number, p_regle out varchar2)
is

v_numclient varchar(10) ;
v_nom varchar(20) ;
v_prenom varchar(20) ;
v_datedep date ;
v_datefin date ;
v_codecateg varchar(6) ;
v_montant number(5) ;
v_regle varchar(1) ;

begin

select numclient, nom, prenom, datedep, dateretour, codecateg, montfact, regle into v_numclient,
v_nom, v_prenom, v_datedep, v_datefin, v_codecateg, v_montant, v_regle
from reservation natural join client natural join vehicule natural join facture where numreserv =
p_numreserv ;

p_numclient := v_numclient ;
p_nom := v_nom ;
p_prenom := v_prenom ;
p_datedep := v_datedep ;
p_datefin := v_datefin ;
p_codecateg := v_codecateg ;
p_montant := v_montant ;
p_regle := v_regle ;

exception
when no_data_found then dbms_output.put_line('rien trouve');

END;
```

Cette procédure permet d'afficher la réservation si la facture a été édité.

```
create or replace procedure affichres(p_numreserv varchar2, p_numclient out varchar2, p_nom out
varchar2, p_prenom out varchar2, p_datedep out date, p_datefin out date, p_codecateg out
varchar2, p_montant out number, p_regle out varchar2)
is

v_numclient varchar(10) ;
v_nom varchar(20) ;
v_prenom varchar(20) ;
v_datedep date ;
v_datefin date ;
v_codecateg varchar(6) ;
v_montant number(5) ;
v_regle varchar(1) ;
```

```
begin

select numclient, nom, prenom, datedep, dateretour, codecateg, prixestime into v_numclient,
v_nom, v_prenom, v_datedep, v_datefin, v_codecateg, v_montant
   from reservation natural join client natural join vehicule where numreserv = p_numreserv ;
   p_numclient := v_numclient ;
   p_nom := v_nom ;
   p_prenom := v_prenom ;
   p_datedep := v_datedep ;
   p_datefin := v_datefin ;
   p_codecateg := v_codecateg ;
   p_montant := v_montant ;
   p_regle := 'P' ;

exception
when no_data_found then dbms_output.put_line('rien trouve');

END;
```

Cette procédure permet d'afficher la réservation si la facture n'a pas été édité.

Procédure PL/SQL utilisée pour le cas 4 :

```
create or replace procedure suppreserv (p_numreserv varchar, p_sortie out number)
is
v_num varchar(18);

BEGIN
p_sortie := 0;

select numreserv
into v_num
from reservation
where numreserv = p_numreserv;

delete from reservation
where numreserv = v_num;

EXCEPTION

when no_data_found then p_sortie := 1;

END suppreserv;
```