HomeWork 2 Solution

Que 1.

```
CREATE OR REPLACE PROCEDURE Q1 (TYPE_ARG IN CLASS.TYPE%TYPE) AS
TITLE_VAR CLASS.TYPE%TYPE;
F_NAME_VAR INSTRUCTOR.F_NAME%TYPE;
L_NAME_VAR INSTRUCTOR.L_NAME%TYPE;
CURSOR C IS
SELECT C.TITLE ,I.F_NAME,I.L_NAME FROM CLASS C JOIN INSTRUCTOR I ON
C.INSTRUCTOR=I.ID  WHERE  C.TYPE=TYPE_ARG;

BEGIN
OPEN C;
LOOP
FETCH C INTO TITLE_VAR,F_NAME_VAR ,L_NAME_VAR ;
EXIT WHEN C%NOTFOUND;
DBMS_OUTPUT.PUT_LINE('TITLE IS : ' || TITLE_VAR || '     FIRST NAME IS :'||F_NAME_VAR ||
'     LAST NAME IS :' || L_NAME_VAR);
END LOOP;
CLOSE C;
EXCEPTION
  WHEN NO_DATA_FOUND THEN
    DBMS_OUTPUT.PUT_LINE('No data');
  WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE('Error occured'|| SQLCODE || ' ' || SQLERRM);
END;
```

Que 2:

```
CREATE OR REPLACE PROCEDURE Q2 (SEASON_ARG IN
CLASS.SEASON%TYPE,YEAR_ARG IN CLASS.YEAR%TYPE) AS
TITLE_VAR CLASS.TITLE%TYPE;
CURSOR C1 IS
SELECT TITLE FROM CLASS WHERE SEASON=SEASON_ARG AND YEAR=YEAR_ARG ;
CURSOR C2 IS
SELECT TITLE FROM CLASS WHERE SEASON=SEASON_ARG ;
CURSOR C3 IS
SELECT TITLE FROM CLASS WHERE YEAR=YEAR_ARG;

BEGIN

IF (SEASON_ARG is NULL AND YEAR_ARG is not NULL) THEN
OPEN C3;
LOOP
FETCH C3 INTO TITLE_VAR ;
```

```
EXIT WHEN C3%NOTFOUND;
DBMS_OUTPUT.PUT_LINE('CLASSES  OFFERED DURING' || YEAR_ARG || '  IS  '
||TITLE_VAR );
END LOOP;
CLOSE C3;

ELSIF (YEAR_ARG is NULL AND SEASON_ARG is not NULL) THEN
OPEN C2;
LOOP
FETCH C2 INTO TITLE_VAR;
EXIT WHEN C2%NOTFOUND;
DBMS_OUTPUT.PUT_LINE('CLASSES TITLE OFFERED DURING' || SEASON_ARG ||'  IS  '
||TITLE_VAR);
END LOOP;
CLOSE C2;

ELSIF(YEAR_ARG is not NULL AND SEASON_ARG is not NULL) THEN
OPEN C1;
LOOP
FETCH C1 INTO TITLE_VAR;
EXIT WHEN C1%NOTFOUND;
DBMS_OUTPUT.PUT_LINE('CLASSES OFFERED DURING ' || YEAR_ARG || 'AND' ||
SEASON_ARG || ' IS '|| TITLE_VAR);
END LOOP;
CLOSE C1;

ELSE
DBMS_OUTPUT.PUT_LINE('NO CLASSES ARE OFFERED');
END IF;
END;
```

Que 3:

```
CREATE OR REPLACE PROCEDURE Q3 (CLASS_TYPE_ARG CLASS.TYPE%TYPE)AS
TOTAL_REVENUE_VAR NUMBER;

BEGIN

SELECT SUM(COST) INTO TOTAL_REVENUE_VAR FROM ENROLLMENT E JOIN CLASS C
ON C.ID=E.CLASS_ID  WHERE TYPE=CLASS_TYPE_ARG GROUP BY TYPE;

DBMS_OUTPUT.PUT_LINE('TOTAL REVENUE FOR CLASS TYPE ' || CLASS_TYPE_ARG || '
IS :  '|| TOTAL_REVENUE_VAR);

END;
```

Que 4:

```
CREATE OR REPLACE PROCEDURE Q4 (F_NAME_ARG IN INSTRUCTOR.F_NAME%TYPE ,
L_NAME_ARG IN INSTRUCTOR.L_NAME%TYPE )AS
COUNTER_INS NUMBER;
COUNTER_REC NUMBER;

CURSOR C_INS IS SELECT COUNT(ID) FROM INSTRUCTOR  WHERE
F_NAME=F_NAME_ARG AND L_NAME= L_NAME_ARG;
CURSOR C_REC IS SELECT COUNT(ID) INTO COUNTER_REC FROM RECCENTERMEMBER
WHERE F_NAME=F_NAME_ARG AND L_NAME= L_NAME_ARG;

BEGIN

OPEN C_INS;
FETCH C_INS INTO COUNTER_INS ;
CLOSE C_INS;

OPEN C_REC;
FETCH C_REC INTO  COUNTER_REC;
CLOSE C_REC;

IF COUNTER_INS = 0 AND COUNTER_REC > 0 THEN
DBMS_OUTPUT.PUT_LINE(F_NAME_ARG  ||' ' ||  L_NAME_ARG ||' IS A RECCENTER
MEMBER');

ELSIF COUNTER_INS >0 AND COUNTER_REC = 0 THEN
DBMS_OUTPUT.PUT_LINE(F_NAME_ARG  ||' ' ||  L_NAME_ARG ||' IS AN INSTRUCTOR');

ELSIF COUNTER_INS >0 AND COUNTER_REC >0 THEN
DBMS_OUTPUT.PUT_LINE(F_NAME_ARG  ||' ' ||  L_NAME_ARG ||' IS AN  INSTRUCTOR AND
A RECCENTER MEMBER');

ELSE DBMS_OUTPUT.PUT_LINE(F_NAME_ARG  ||' ' ||  L_NAME_ARG ||' IS NOT PRESENT IN
DATABASE');

END IF;

END;
```

Question 5:

```
package menk;

import java.sql.*;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.util.*;
```

```
import java.lang.*;

public class Main {

	private static final String USERNAME = "";
	private static final String PASSWORD = "";
	private static final String CONN_STRING =
		"jdbc:oracle:thin:@fourier.cs.iit.edu:1521:orcl";

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

	   System.out.println();
	   Scanner scan = new Scanner(System.in);
	   String first="NULL";
	   String last="NULL";
	   String id;
	   int idIn = 0;
	   int flIn = 0;
	   int none = 0;
	   int fid =-1;
	   String address = "null";
	   long phone = 123;
	   System.out.println("Enter first name (case sensitive)(press enter to skip to ID entry): ");
	   first = scan.nextLine();
	   if(first.equals("")){
	      first = "NULL";
	   }

	   if(!(first.equals("NULL"))){
	      System.out.println("Enter last name (case sensitive): ");
	      last = scan.nextLine();

	      if(last.equals("")){
	         last = "NULL";
	         System.out.println("Error: Last Name not entered previous input (first name) will be ig-
nored");
	         System.out.println();
	      }
	   }

	   if(!first.equals("NULL") && !last.equals("NULL")){
	      id = "NULL";
	   }
	   else{
	      System.out.println("Enter ID Number: ");
	      id = scan.nextLine();
	      if(id.equals("")){
	         id = "NULL";
	      }
```

```
        }

        if(first.equals("NULL") && last.equals("NULL") && id.equals("NULL")){
           System.out.println("Nothing Entered");
           none = 1;
        }
        else if(!(id.equals("NULL"))){
           idIn = 1;
        }
        else if(id.equals("NULL")){
           flIn=1;
        }
        if(none == 0) {
           Connection conn = null;
           Statement stmt = null;
           try {
              Class.forName("oracle.jdbc.driver.OracleDriver");
              conn = DriverManager.getConnection(CONN_STRING, USERNAME, PASSWORD);
              //System.out.println("Connected!");
              System.out.println();
              if (flIn == 1) {
                 //user inputted first and last name
                 stmt = conn.createStatement();
                 String sql = "SELECT family_id FROM RECCENTERMEMBER WHERE f_name = "
+ """ + first + """ + " and l_name = " + """ + last + """;
                 //System.out.println(sql);
                 ResultSet rs = stmt.executeQuery(sql);
                 while (rs.next()) {
                    //Retrieve by column name
                    fid = rs.getInt("family_id");
                    //Display values
                    //System.out.print("fid: " + fid);
                 }
                 rs.close();
                 //System.out.println(fid);
                 stmt = conn.createStatement();
                 String sql2 = "select address,phone from familypackage where id = " + fid;
                 //System.out.println(sql2);
                 ResultSet rs2 = stmt.executeQuery(sql2);
                 while (rs2.next()) {
                    //Retrieve by column name
                    address = rs2.getString("ADDRESS");
                    phone = rs2.getLong("PHONE");
                    //Display values
                    //System.out.print("address: "+ address + " phone: " + phone);
                 }
                 if (fid == -1 && address.equals("null") && phone == 123) {
                    System.out.println("Person Not Found");
                 } else if (fid != -1 && address.equals("null") && phone == 123) {
```

```
                System.out.println( first + " " +last +" | contact info not available in the database");
                System.out.println();
                System.out.println("Would you like to input contact info for the " + last + " family?
(y/n)");
                String yn = scan.nextLine();
                if(yn.charAt(0) == 'y' || yn.charAt(0) == 'Y'){
                   System.out.println("Entered Y/y");
                   System.out.println("Enter Address");
                   address=scan.nextLine();
                   if(address.equals("")){
                      System.out.println("Error: No Address entered quitting");
                   }
                   else{
                      System.out.println("Enter Phone Number WITHOUT dashes or spaces");
                      String temp;
                      temp = scan.nextLine();
                      if(temp.length()<10){
                         System.out.println("Error: Invalid Phone Number quitting");
                      }
                      else if(temp.length()>10){
                         System.out.println("Error: Invalid Phone Number quitting");
                      }
                      else{
                         phone=Long.valueOf(temp).longValue();
                         //System.out.println(address+ " " +phone);
                      }
                      stmt = conn.createStatement();
                      int tempid=0;
                      String sql3 = "Select max(id) from FAMILYPACKAGE";
                      //System.out.println(sql2);
                      ResultSet rs3 = stmt.executeQuery(sql3);
                      while (rs3.next()) {
                         //Retrieve by column name
                         tempid=rs3.getInt("MAX(ID)");
                         //Display values
                         //System.out.print("address: "+ address + " phone: " + phone);
                      }
                      stmt = conn.createStatement();
                      String sql4="Insert into FamilyPackage values("+(tempid+1) + "," + "'"+ad-
dress+"'" + "," + phone +")";
                      stmt.executeUpdate(sql4);
                      stmt = conn.createStatement();
                      String sql5 ="Update RECCENTERMEMBER set family_id =" + (tempid+1) +
"where l_name ="+"'"+last+"'";
                      stmt.executeUpdate(sql5);
                      System.out.println("Contact Information for the " + last + " family has been up-
dated in the database");
                   }
```

```
                }
            } else {
                System.out.println(first + " " + last + " | " + address + " | " + phone);
            }

        }
        else if (idIn == 1) {
            //user inputted id
            stmt = conn.createStatement();
            String sql = "Select family_id,f_name,l_name from RECCENTERMEMBER where id =
" + id;
            ResultSet rs = stmt.executeQuery(sql);
            while (rs.next()) {
                //Retrieve by column name
                fid = rs.getInt("family_id");
                first = rs.getString("f_name");
                last = rs.getString("l_name");
                //Display values
                //System.out.print(fid);
            }
            stmt = conn.createStatement();
            String sql2 = "select address,phone from familypackage where id = " + fid;
            ResultSet rs2 = stmt.executeQuery(sql2);
            while (rs2.next()) {
                //Retrieve by column name
                address = rs2.getString("ADDRESS");
                phone = rs2.getLong("PHONE");
                //Display values
                //System.out.print("address: "+ address + " phone: " + phone);
            }
            if (fid == -1 && address.equals("null") && phone == 123) {
                System.out.println("Person Not Found");
            } else if (fid != -1 && address.equals("null") && phone == 123) {
                System.out.println( first + " " +last +" | contact info not available in the database");
                System.out.println();
                System.out.println("Would you like to input contanct info for the " + last + " family?
(y/n)");
                String yn = scan.nextLine();
                if(yn.charAt(0) == 'y' || yn.charAt(0) == 'Y'){
                    System.out.println("Entered Y/y");
                    System.out.println("Enter Address");
                    address=scan.nextLine();
                    if(address.equals("")){
                        System.out.println("Error: No Address entered quitting");
                    }
                    else{
                        System.out.println("Enter Phone Number WITHOUT dashes or spaces");
                        String temp;
                        temp = scan.nextLine();
```

```
                        if(temp.length()<10){
                           System.out.println("Error: Invalid Phone Number quitting");
                           System.exit(0);
                        }
                        else if(temp.length()>10){
                           System.out.println("Error: Invalid Phone Number quitting");
                           System.exit(0);
                        }
                        else{
                           phone=Long.valueOf(temp).longValue();
                           //System.out.println(address+ " " +phone);
                        }
                        stmt = conn.createStatement();
                        int tempid=0;
                        String sql3 = "Select max(id) from FAMILYPACKAGE";
                        //System.out.println(sql2);
                        ResultSet rs3 = stmt.executeQuery(sql3);
                        while (rs3.next()) {
                           //Retrieve by column name
                           tempid=rs3.getInt("MAX(ID)");
                           //Display values
                           //System.out.print("address: "+ address + " phone: " + phone);
                        }
                        stmt = conn.createStatement();
                        String sql4="Insert into FamilyPackage values("+(tempid+1) + "," + "'"+ad-
dress+"'" + "," + phone +")";
                        stmt.executeUpdate(sql4);
                        stmt = conn.createStatement();
                        String sql5 ="Update RECCENTERMEMBER set family_id =" + (tempid+1) +
"where l_name ="+"'"+last+"'";
                        stmt.executeUpdate(sql5);
                        System.out.println("Contact Information for the " + last + " family has been up-
dated in the database");
                     }

                  }
               }
               else {
                  System.out.println(first + " " + last + " | " + address + " | " + phone);
               }
            }
            stmt.close();
            conn.close();
         } catch (SQLException e) {
            System.err.println(e);
         } catch (ClassNotFoundException e) {
            e.printStackTrace();
         } finally {
            if (conn != null) {
```

```
                conn.close();
            }
        }
    }
    System.out.println("Exiting Program");
  }
}
```

Question 6:

```
package menk;

import java.sql.*;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.util.*;
import java.lang.*;

public class Number6 {

   private static final String USERNAME = "";
   private static final String PASSWORD = "";
   private static final String CONN_STRING =
         "jdbc:oracle:thin:@fourier.cs.iit.edu:1521:orcl";

   public static void main(String[] args) throws SQLException {
      float discount=0;
      int ageGroup=0;
      float rev1=0;
      int younger=-1;
      String temp;
      Scanner scan = new Scanner(System.in);
      while(true){
         System.out.println("Enter Discount:");
         String input = scan.next();
         float intInputValue = 0;
         try{
            intInputValue = Float.parseFloat(input);
            discount=intInputValue;
            //System.out.println("Correct input");
            break;
         }
         catch (NumberFormatException ne){
            System.out.println("Error: Incorrect input");
         }
```

```
        }


        System.out.println();
        System.out.println("Enter Age Group:");
        while(true){
           //System.out.println("Enter Age Group:");
           String input = scan.next();
           //temp=input;
           if(input.equals("younger") || input.equals("Younger")){
              younger=1;
           }
           int intInputValue = 0;
           try{
              intInputValue = Integer.parseInt(input);
              ageGroup=intInputValue;
              //System.out.println("Correct input");
              break;
           }
           catch (NumberFormatException ne){
              //System.out.println("Error: Incorrect input");
           }
        }

        //System.out.println(younger);

        //System.out.println(discount + " " + ageGroup);



        Connection conn = null;
        Statement stmt = null;
        try {
           float decimal = 1-(discount/100);
           Class.forName("oracle.jdbc.driver.OracleDriver");
           conn = DriverManager.getConnection(CONN_STRING, USERNAME, PASSWORD);
           //System.out.println("Connected!");
           System.out.println();
           stmt=conn.createStatement();
           if(younger == -1) {
              String sql = "Select (sum(cost)-sum(cost*"+decimal+")) as impact from
ENROLLMENT,(select id from RECCENTERMEMBER where (extract(year from
dob))<((extract(year from sysdate))-" + ageGroup + "))temp where ENROLLMENT.member_id =
temp.id";
              ResultSet rs = stmt.executeQuery(sql);
              while (rs.next()) {
                 //Retrieve by column name
                 rev1 = rs.getFloat("impact");
                 //Display values
```

```
                //System.out.print("fid: " + fid);
            }
            rs.close();
        }
        else{
            String sql = "Select (sum(cost)-sum(cost*"+decimal+")) as impact from
ENROLLMENT,(select id from RECCENTERMEMBER where (extract(year from
dob))>((extract(year from sysdate))-" + ageGroup + "))temp where ENROLLMENT.member_id =
temp.id";
            ResultSet rs = stmt.executeQuery(sql);
            while (rs.next()) {
                //Retrieve by column name
                rev1 = rs.getFloat("impact");
                //Display values
                //System.out.print("fid: " + fid);
            }
            rs.close();
        }
        System.out.println("Revenue Impact: $" + rev1);
        stmt.close();
        conn.close();
    }
    catch (SQLException e) {
        System.err.println(e);
    } catch (ClassNotFoundException e) {
        e.printStackTrace();
    } finally {
        if (conn != null) {
            conn.close();
        }
    }
  }
}
```

Question 7:

Trigger:

```
create or replace trigger Prob7
After update of season,year on class
for each row
declare
 varNewSeasonCount number:=0;
 varOldSeasonCount number:=0;

begin
```

```
  select decode(trim(:new.Season), 'Spring',1,
                        'Winter', 2,
                        'Summer', 3,
                        'Fall', 4,
                        0) into varNewSeasonCount from dual;
  select decode(trim(:old.Season), 'Spring',1,
                        'Winter', 2,
                        'Summer', 3,
                        'Fall', 4,
                        0) into varOldSeasonCount from dual;
  if( :new.year>=:old.year  or  varNewSeasonCount>= varOldSeasonCount) then

     update Instructor
     set LastClassTaught = :old.Type
     where ID = :old.Instructor;

     end if;


  end;
```

Procedure for initializing the last class:

```
create or replace PROCEDURE HW2_7GETLASTCLASS(in_id IN INSTRUCTOR.ID%TYPE) AS
CURSOR getSeason is
SELECT *
FROM
(SELECT *
FROM CLASS
WHERE YEAR= (
SELECT MAX(CLASS.YEAR) as max_yr
FROM INSTRUCTOR INNER JOIN CLASS
ON INSTRUCTOR.ID = CLASS.INSTRUCTOR
WHERE INSTRUCTOR.ID = in_id))
ORDER BY
CASE SEASON
WHEN 'Winter' THEN 1
WHEN 'Fall' THEN 2
WHEN 'Summer' THEN 3
WHEN 'Spring' THEN 4
END;
p_a getSeason%ROWTYPE;
lastclass varchar2(999);
BEGIN
OPEN getSeason;
FETCH getSeason into p_a;
if getSeason%notfound then
DBMS_OUTPUT.PUT_LINE('Instructor not found');
ELSE
```

```
SELECT LISTAGG(Title, ',') WITHIN GROUP (ORDER BY Title) as l into lastclass
FROM (SELECT *
FROM
(SELECT *
FROM CLASS
WHERE YEAR= (
SELECT MAX(CLASS.YEAR) as max_yr
FROM INSTRUCTOR INNER JOIN CLASS
ON INSTRUCTOR.ID = CLASS.INSTRUCTOR
WHERE INSTRUCTOR.ID = in_id)) WHERE SEASON=p_a.SEASON)
end;
DBMS_OUTPUT.PUT_LINE('Last Class for Instructor ' || in_id || ' is ' || lastclass);
Update instructor set lastclasstaught=lastclass where id=in_id;
commit;
end if;
CLOSE getSeason;
EXCEPTION
WHEN NO_DATA_FOUND THEN
DBMS_OUTPUT.PUT_LINE('No data');
WHEN TOO_MANY_ROWS THEN
DBMS_OUTPUT.PUT_LINE('More than one row fetched');
WHEN OTHERS THEN
DBMS_OUTPUT.PUT_LINE('Error occured'|| SQLCODE || ' ' || SQLERRM);
END;
```

This procedure needs to be called from a java class, passing the instructor id as the input.