**Assignment – Day 10**

**Time : 90 mins**

1. The following table shows a few characters from the Harry Potter series of books.

Status column refers to the status with respect to Harry Potter.

1. Write a function that will create a table and insert these records into the table

**package** com.verizon.day1;

**import** java.sql.Connection;

**import** java.sql.DriverManager;

**import** java.sql.PreparedStatement;

**import** java.sql.SQLException;

**public** **class** Potter {

**public** **static** **void** main(String[] args) {

**try** {

Class.*forName*("oracle.jdbc.driver.OracleDriver");

Connection con=DriverManager.*getConnection*("jdbc:oracle:thin:@localhost:1521:orcl","hr","Admin");

PreparedStatement pst=con.prepareStatement(" CREATE TABLE Harry\_POTTER(name varchar(50) not null,house varchar(50),role varchar(10),status varchar(10),dies varchar(10))");

} **catch** (ClassNotFoundException e) {

e.printStackTrace();

} **catch** (SQLException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

}

}

1. Write a function that will return the list of names of all characters who belonged to the Gryffindor house

**public** **static** **void** main(String[] args) {

**try** {

Class.*forName*("oracle.jdbc.driver.OracleDriver");

Connection con=DriverManager.*getConnection*("jdbc:oracle:thin:@localhost:1521:orcl","hr","Admin");

PreparedStatement pst=con.prepareStatement("select NAME from HARRY\_POTTER where house like ? ");

pst.setString(1,"Gryffindor%");

ResultSet rs=pst.executeQuery();

**while** (rs.next())

{

System.***out***.println(rs.getString(1) + "\t");

}

rs.close();

pst.close();

con.close();

} **catch** (ClassNotFoundException e) {

e.printStackTrace();

} **catch** (SQLException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

}

1. Write a function that will return the details of all the characters who are alive, grouped according to their houses

**public** **static** **void** main(String[] args) {

**try** {

Class.*forName*("oracle.jdbc.driver.OracleDriver");

Connection con=DriverManager.*getConnection*("jdbc:oracle:thin:@localhost:1521:orcl","hr","Admin");

PreparedStatement pst=con.prepareStatement("select NAME,house from HARRY\_POTTER where dies like ? order by house ");

pst.setString(1,"No%");

ResultSet rs=pst.executeQuery();

**while** (rs.next())

{

System.***out***.println(rs.getString(1) + "\t" +rs.getString(2) + "\t");

}

rs.close();

pst.close();

con.close();

}

**catch** (ClassNotFoundException e) {

e.printStackTrace();

} **catch** (SQLException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

}

1. Write a function that will return all the details of the family members of Harry Potter

**public** **static** **void** main(String[] args) {

**try** {

Class.*forName*("oracle.jdbc.driver.OracleDriver");

Connection con=DriverManager.*getConnection*("jdbc:oracle:thin:@localhost:1521:orcl","hr","Admin");

PreparedStatement pst=con.prepareStatement("select \* from HARRY\_POTTER where status like ? ");

pst.setString(1,"Family%");

ResultSet rs=pst.executeQuery();

**while** (rs.next())

{

System.***out***.println(rs.getString(1) + "\t" +rs.getString(2) + "\t"+rs.getString(3) + "\t"+rs.getString(4) + "\t" +rs.getString(5) + "\t");

}

rs.close();

pst.close();

con.close();

}

**catch** (ClassNotFoundException e) {

e.printStackTrace();

} **catch** (SQLException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

}

1. Write a function that will return the faculty members who die, sorted by alphabetical order

**public** **static** **void** main(String[] args) {

**try** {

Class.*forName*("oracle.jdbc.driver.OracleDriver");

Connection con=DriverManager.*getConnection*("jdbc:oracle:thin:@localhost:1521:orcl","hr","Admin");

PreparedStatement pst=con.prepareStatement("select \* from HARRY\_POTTER where role like ? AND dies like ? order by name ");

pst.setString(1,"Faculty%");

pst.setString(2,"Yes%");

ResultSet rs=pst.executeQuery();

**while** (rs.next())

{

System.***out***.println(rs.getString(1) + "\t" +rs.getString(2) + "\t"+rs.getString(3) + "\t"+rs.getString(4) + "\t" +rs.getString(5) + "\t");

}

rs.close();

pst.close();

con.close();

}

**catch** (ClassNotFoundException e) {

e.printStackTrace();

} **catch** (SQLException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

}

2) Salary procedures

create or replace

PROCEDURE salaryproc( dept\_id IN NUMBER, avgsal OUT NUMBER, maxsal OUT NUMBER, minsal OUT NUMBER) as

begin

select min(salary) ,max(salary),avg(salary) into minsal,maxsal,avgsal from ecopy where department\_id=dept\_id;

end salaryproc;

package com.verizon.day1;

import java.sql.CallableStatement;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.SQLException;

import java.sql.Types;

import java.util.Scanner;

public class Potter {

public static void main(String[] args) {

try {

Class.forName("oracle.jdbc.driver.OracleDriver");

Connection con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:orcl","hr","Admin");

CallableStatement cst=con.prepareCall("{call salaryproc( ?, ? ,?, ?)}");

Scanner scanner=new Scanner(System.in);

System.out.println("enter the department id");

int deptid=scanner.nextInt();

cst.setInt(1, deptid);

cst.registerOutParameter(2, Types.INTEGER);

cst.registerOutParameter(3, Types.INTEGER);

cst.registerOutParameter(4, Types.INTEGER);

cst.execute();

System.out.println(cst.getInt(2)+ "\t" +cst.getInt(3) + "\t"+cst.getInt(4) + "\t");

scanner.close();

cst.close();

con.close();

} catch (ClassNotFoundException e) {

e.printStackTrace();

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

}