EV Assignment Oracle PLSQL – Medicine Database

Oracle PLSQL Code: -

Set serveroutput on;

Create table medicine (id number generated always as identity primary key,

name varchar(100), description varchar(100), quantity number, price number);

insert into medicine (name, description, quantity, price) values ('Crocin', 'Body Pain, Muscle Pain, Fever', 100, 100);

insert into medicine (name, description, quantity, price) values ('Combiflam', 'Body Pain, Muscle Pain, Join Pain, Fever', 50, 150);

insert into medicine (name, description, quantity, price) values ('Dolo 650', 'Body Pain, Fever, Headaches, Stomach Discomfort', 30, 60);

insert into medicine (name, description, quantity, price) values ('Azithromycin', 'Bacterial Infections for Throat, Ears, Tonsils, etc.', 10, 240);

insert into medicine (name, description, quantity, price) values ('Gelusil', 'Reduces acidity', 70, 140);

insert into medicine (name, description, quantity, price) values ('Cofsil Cough Syrup', 'Reduces Cough. Quantity given is in no.of bottles of 100 ml each', 5, 350);

insert into medicine (name, description, quantity, price) values ('Paracetamol', 'Pain reliever and fever reducer', 90, 90);

insert into medicine (name, description, quantity, price) values ('Ibuprofen', 'Pain relief and anti-inflammatory', 20, 120);

insert into medicine (name, description, quantity, price) values ('Aspirin', 'Pain, fever, inflammation relief', 40, 200);

select \* from medicine;

-- Procedure to add medicine

CREATE OR REPLACE PROCEDURE add\_medicine (

p\_name IN VARCHAR2,

p\_description IN VARCHAR2,

p\_quantity IN NUMBER,

p\_price IN NUMBER

) AS

BEGIN

INSERT INTO medicine (name, description, quantity, price)

VALUES (p\_name, p\_description, p\_quantity, p\_price);

END;

/

-- Procedure to read medicine

CREATE OR REPLACE PROCEDURE read\_medicine (

p\_id IN NUMBER,

p\_name OUT VARCHAR2,

p\_description OUT VARCHAR2,

p\_quantity OUT NUMBER,

p\_price OUT NUMBER

) AS

BEGIN

SELECT name, description, quantity, price

INTO p\_name, p\_description, p\_quantity, p\_price

FROM medicine

WHERE id = p\_id;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

p\_name := NULL;

p\_description := NULL;

p\_quantity := NULL;

p\_price := NULL;

END;

/

-- Procedure to update medicine

CREATE OR REPLACE PROCEDURE update\_medicine (

p\_id IN NUMBER,

p\_name IN VARCHAR2,

p\_description IN VARCHAR2,

p\_quantity IN NUMBER,

p\_price IN NUMBER

) AS

BEGIN

UPDATE medicine

SET name = p\_name,

description = p\_description,

quantity = p\_quantity,

price = p\_price

WHERE id = p\_id;

END;

/

-- Procedure to delete medicine

CREATE OR REPLACE PROCEDURE delete\_medicine (

p\_id IN NUMBER

) AS

BEGIN

DELETE FROM medicine

WHERE id = p\_id;

END;

/

-- Procedure to find medicine name by ID

CREATE OR REPLACE PROCEDURE findMedicineNameById(

p\_id IN NUMBER,

p\_name OUT VARCHAR2

) IS

BEGIN

SELECT name

INTO p\_name

FROM medicine

WHERE id = p\_id;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

p\_name := NULL;

END;

/

-- Procedure to find all medicine names

CREATE OR REPLACE PROCEDURE findAllMedicines (

p\_names OUT SYS\_REFCURSOR

) IS

BEGIN

OPEN p\_names FOR

SELECT name FROM medicine;

END;

/

-- Procedure to find medicines by price

CREATE OR REPLACE PROCEDURE findMedicinesByPrice (

p\_price IN NUMBER,

p\_names OUT SYS\_REFCURSOR

) IS

BEGIN

OPEN p\_names FOR

SELECT name FROM medicine

WHERE price = p\_price;

END;

/

select \* from medicine;

Java Code: -

MedicineDao.java: -

package com.medicine.dao;

import java.sql.Connection;

import java.sql.SQLException;

import com.medicine.model.Medicine;

public interface MedicineDao {

public void addMedicine(Medicine med) throws SQLException;

public Medicine readMedicine(int Id) throws SQLException; // return details of medicine for reference, not returning void

public void updateMedicine(Medicine med) throws SQLException;

public void deleteMedicine(int Id) throws SQLException;

public String findMedicineNameById(int Id) throws SQLException;

public String findAllMedicines() throws SQLException;

public String findMedicinesByPrice(int price) throws SQLException;

public Connection getConnection() throws SQLException;

}

MedicineDaoImpl.java: -

**package** com.medicine.dao;

**import** java.sql.CallableStatement;

**import** java.sql.Connection;

**import** java.sql.DriverManager;

**import** java.sql.ResultSet;

**import** java.sql.SQLException;

**import** java.sql.Types;

**import** com.medicine.model.Medicine;

**public** **class** MedicineDaoImpl **implements** MedicineDao {

**private** Connection con;

**public** **void** addMedicine(Medicine medicine) **throws** SQLException {

con = getConnection();

String sql = "{call add\_medicine(?, ?, ?, ?)}";

CallableStatement cst = con.prepareCall(sql);

cst.setString(1, medicine.getName());

cst.setString(2, medicine.getDescription());

cst.setInt(3, medicine.getQuantity());

cst.setInt(4, medicine.getPrice());

cst.executeUpdate();

cst.close();

con.close();

}

**public** Medicine readMedicine(**int** id) **throws** SQLException {

con = getConnection();

String sql = "{call read\_medicine(?, ?, ?, ?, ?)}";

CallableStatement cst = con.prepareCall(sql);

cst.setInt(1, id);

cst.registerOutParameter(2, Types.***VARCHAR***);

cst.registerOutParameter(3, Types.***VARCHAR***);

cst.registerOutParameter(4, Types.***INTEGER***);

cst.registerOutParameter(5, Types.***INTEGER***);

cst.execute();

Medicine medicine = **new** Medicine(id, cst.getString(2), cst.getString(3), cst.getInt(4), cst.getInt(5));

cst.close();

con.close();

**return** medicine;

}

**public** **void** updateMedicine(Medicine medicine) **throws** SQLException {

con = getConnection();

String sql = "{call update\_medicine(?, ?, ?, ?, ?)}";

CallableStatement cst = con.prepareCall(sql);

cst.setInt(1, medicine.getId());

cst.setString(2, medicine.getName());

cst.setString(3, medicine.getDescription());

cst.setInt(4, medicine.getQuantity());

cst.setInt(5, medicine.getPrice());

cst.executeUpdate();

cst.close();

con.close();

}

**public** **void** deleteMedicine(**int** id) **throws** SQLException {

con = getConnection();

String sql = "{call delete\_medicine(?)}";

CallableStatement cst = con.prepareCall(sql);

cst.setInt(1, id);

cst.executeUpdate();

cst.close();

con.close();

}

**public** String findMedicineNameById(**int** id) **throws** SQLException {

con = getConnection();

String sql = "{call findMedicineNameById(?, ?)}";

CallableStatement cst = con.prepareCall(sql);

cst.setInt(1, id);

cst.registerOutParameter(2, Types.***VARCHAR***);

cst.execute();

String medicineName = cst.getString(2);

cst.close();

con.close();

**return** medicineName;

}

**public** String findAllMedicines() **throws** SQLException {

con = getConnection();

String sql = "{call findAllMedicines(?)}";

CallableStatement cst = con.prepareCall(sql);

cst.registerOutParameter(1, Types.***REF\_CURSOR***);

cst.execute();

ResultSet rs = (ResultSet) cst.getObject(1);

StringBuilder allMedicines = **new** StringBuilder();

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

allMedicines.append(rs.getString("name")).append("\n");

}

cst.close();

con.close();

**return** allMedicines.toString();

}

**public** String findMedicinesByPrice(**int** price) **throws** SQLException {

con = getConnection();

String sql = "{call findMedicinesByPrice(?, ?)}";

CallableStatement cst = con.prepareCall(sql);

cst.setInt(1, price);

cst.registerOutParameter(2, Types.***REF\_CURSOR***);

cst.execute();

ResultSet rs = (ResultSet) cst.getObject(2);

StringBuilder medicinesByPrice = **new** StringBuilder();

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

medicinesByPrice.append(rs.getString("name")).append("\n");

}

cst.close();

con.close();

**return** medicinesByPrice.toString();

}

**public** Connection getConnection() **throws** SQLException {

String URL = "jdbc:oracle:thin:@localhost:1521:xe";

String USER = "SYS AS SYSDBA";

String PWD = "Atharva2003";

**return** DriverManager.*getConnection*(URL, USER, PWD);

}

}

MedicineService.java: -

**package** com.medicine.service;

**import** java.sql.SQLException;

**import** java.util.Scanner;

**import** com.medicine.dao.MedicineDaoImpl;

**import** com.medicine.model.Medicine;

**public** **class** MedicineService {

**private** **static** MedicineDaoImpl *dao*;

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

*dao* = **new** MedicineDaoImpl();

Medicine medicine = **null**;

**int** id, quantity, price;

String name, description;

Scanner sc = **new** Scanner(System.***in***);

System.***out***.println("Welcome to Medicine Store");

**int** choice;

**do** {

System.***out***.println("Enter your choice: ");

System.***out***.println("1. Add new medicine");

System.***out***.println("2. Read the medicine details");

System.***out***.println("3. Update the medicine details");

System.***out***.println("4. Delete a medicine");

System.***out***.println("5. Find medicine name by ID");

System.***out***.println("6. Read all medicine names");

System.***out***.println("7. Find medicine names by price");

System.***out***.println("8. Exit");

choice = sc.nextInt();

sc.nextLine();

**try** {

**switch** (choice) {

**case** 1:

System.***out***.println("Enter the medicine ID: ");

id = sc.nextInt();

sc.nextLine();

System.***out***.println("Enter the medicine name: ");

name = sc.nextLine();

System.***out***.println("Enter the medicine description: ");

description = sc.nextLine();

System.***out***.println("Enter the medicine quantity: ");

quantity = sc.nextInt();

System.***out***.println("Enter the medicine price: ");

price = sc.nextInt();

medicine = **new** Medicine(id, name, description, quantity, price);

*dao*.addMedicine(medicine);

System.***out***.println("Medicine added successfully.");

**break**;

**case** 2:

System.***out***.println("Enter the medicine ID: ");

id = sc.nextInt();

sc.nextLine();

medicine = *dao*.readMedicine(id);

**if** (medicine != **null**) {

System.***out***.println(medicine);

} **else** {

System.***out***.println("No medicine found with ID: " + id);

}

**break**;

**case** 3:

System.***out***.println("Enter the medicine ID to update: ");

id = sc.nextInt();

sc.nextLine();

System.***out***.println("Enter the new medicine name: ");

name = sc.nextLine();

System.***out***.println("Enter the new medicine description: ");

description = sc.nextLine();

System.***out***.println("Enter the new medicine quantity: ");

quantity = sc.nextInt();

System.***out***.println("Enter the new medicine price: ");

price = sc.nextInt();

medicine = **new** Medicine(id, name, description, quantity, price);

*dao*.updateMedicine(medicine);

System.***out***.println("Medicine updated successfully.");

**break**;

**case** 4:

System.***out***.println("Enter the medicine ID to delete: ");

id = sc.nextInt();

sc.nextLine();

*dao*.deleteMedicine(id);

System.***out***.println("Medicine deleted successfully.");

**break**;

**case** 5:

System.***out***.println("Enter the medicine ID to find its name: ");

id = sc.nextInt();

sc.nextLine();

String medicineName = *dao*.findMedicineNameById(id);

**if** (medicineName != **null**) {

System.***out***.println("Medicine Name: " + medicineName);

} **else** {

System.***out***.println("No medicine found with ID: " + id);

}

**break**;

**case** 6:

System.***out***.println("All Available Medicines are: ");

String allMedicines = *dao*.findAllMedicines();

System.***out***.println(allMedicines);

**break**;

**case** 7:

System.***out***.println("Enter the price to find medicines: ");

price = sc.nextInt();

sc.nextLine();

String medicinesByPrice = *dao*.findMedicinesByPrice(price);

**if** (medicinesByPrice != **null** && !medicinesByPrice.isEmpty()) {

System.***out***.println("Medicines with price " + price + ":");

System.***out***.println(medicinesByPrice);

} **else** {

System.***out***.println("No medicines found with price: " + price);

}

**break**;

**case** 8:

System.***out***.println("Exiting...");

**break**;

**default**:

System.***out***.println("Invalid choice. Please enter a number between 1 and 8.");

**break**;

}

} **catch** (SQLException e) {

System.***out***.println("An error occurred: " + e.getMessage());

}

} **while** (choice != 8);

sc.close();

}

}