1. Write a Java program to connect to a MySQL database using JDBC.

Code; package JDBC\_conn;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.SQLException;

public class Jdbc\_conn {

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

//import packages

//create object for connection

//Retrieve data

//close connection

String url="jdbc:mysql://localhost:3306/mydb";

String user="root";

String password="Reddy@7125";

Class.forName("com.mysql.cj.jdbc.Driver");

Connection con=DriverManager.getConnection(url,user,password);

System.out.println("Connection created");

con.close();

}

}output; Connection created

1. Create a Java class to insert student records into a database table.

Code; **package** JDBC\_conn;

**import** java.sql.Connection;

**import** java.sql.DriverManager;

**import** java.sql.ResultSet;

**import** java.sql.SQLException;

**import** java.sql.Statement;

**public** **class** Create\_table {

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

String url="jdbc:mysql://localhost:3306/mydb";

String user="root";

String password="Reddy@7125";

//create table

//S

String sql="create table if not exists students1(rollno int,"

+ "name varchar(50),"

+"per int,"

+"email varchar(50))";

//insert record

String insertSQL="insert into students1 values(101,'Neeva Sharma',97,'abc@gmail.com'),"

+"(102,'Reeva Sharma',94,'Reeva@gmail.com'),"

+"(103,'Shiva Upadhyay',96,'Shiva@gmail.com')";

//String updateSQL="update students set email='Neeva sharma@gmail.com' where name ='Neeva Sharma";

**try** {

//driver connection

Class.*forName*("com.mysql.cj.jdbc.Driver");

Connection con=DriverManager.*getConnection*(url,user,password);

System.***out***.println("Connection created");

//create statement for query execution

Statement stmt =con.createStatement();

//call create query using statement

stmt.executeUpdate(sql);

System.***out***.println("Students1 table created");

//call insert query using statement

**int** rowInserted =stmt.executeUpdate(insertSQL);

//int updaterow =stmt.executeUpdate(updateSQL);

**if**(rowInserted>0) {

System.***out***.println("new Student record inserted ");

}

stmt.close();

con.close();

}**catch**(SQLException e) {

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

}**catch**( ClassNotFoundException e) {

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

}

}

}

Output; Connection created

Students1 table created

new Student record inserted

1. Write a JDBC program to fetch and display all student records from the database.

Code; **package** JDBC\_conn;

**import** java.sql.Connection;

**import** java.sql.DriverManager;

**import** java.sql.ResultSet;

**import** java.sql.SQLException;

**import** java.sql.Statement;

**public** **class** Create\_table {

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

String url="jdbc:mysql://localhost:3306/mydb";

String user="root";

String password="Reddy@7125";

//create table

//S

String sql="create table if not exists students1(rollno int,"

+ "name varchar(50),"

+"per int,"

+"email varchar(50))";

//insert record

String insertSQL="insert into students1 values(101,'Neeva Sharma',97,'abc@gmail.com'),"

+"(102,'Reeva Sharma',94,'Reeva@gmail.com'),"

+"(103,'Shiva Upadhyay',96,'Shiva@gmail.com')";

//String updateSQL="update students set email='Neeva sharma@gmail.com' where name ='Neeva Sharma";

**try** {

//driver connection

Class.*forName*("com.mysql.cj.jdbc.Driver");

Connection con=DriverManager.*getConnection*(url,user,password);

System.***out***.println("Connection created");

//create statement for query execution

Statement stmt =con.createStatement();

//call create query using statement

stmt.executeUpdate(sql);

System.***out***.println("Students1 table created");

//call insert query using statement

**int** rowInserted =stmt.executeUpdate(insertSQL);

//int updaterow =stmt.executeUpdate(updateSQL);

**if**(rowInserted>0) {

System.***out***.println("new Student record inserted ");

}

//Print table

ResultSet rs=stmt.executeQuery("Select \* from students1");

System.***out***.println("rollno \t name \t Percentage \tEmail");

//while loop for fetching all the table records

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

**int** rollno =rs.getInt("rollno");

String name=rs.getString("name");

**int** per =rs.getInt("per");

String email=rs.getString("email");

System.***out***.println(rollno +" "+ name +" "+ per +" "+ email);

}

stmt.close();

con.close();

}**catch**(SQLException e) {

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

}**catch**( ClassNotFoundException e) {

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

}

}

}

Output; Students1 table created

new Student record inserted

rollno name Percentage Email

101 Neeva Sharma 97 abc@gmail.com

102 Reeva Sharma 94 Reeva@gmail.com

101 Neeva Sharma 97 abc@gmail.com

102 Reeva Sharma 94 Reeva@gmail.com

104 Ajay Sharma 95 ajay@gmail.com

104 Ajay Sharma 95 ajay@gmail.com

104 Ajay Sharma 95 ajay@gmail.com

104 Ajay Sharma 95 ajay@gmail.com

101 Neeva Sharma 97 abc@gmail.com

102 Reeva Sharma 94 Reeva@gmail.com

103 Shiva Upadhyay 96 Shiva@gmail.com

101 Neeva Sharma 97 abc@gmail.com

102 Reeva Sharma 94 Reeva@gmail.com

103 Shiva Upadhyay 96 Shiva@gmail.com

101 Neeva Sharma 97 abc@gmail.com

102 Reeva Sharma 94 Reeva@gmail.com

103 Shiva Upadhyay 96 Shiva@gmail.com

1. Develop a program to search a student by ID using JDBC.

Code; **package** JDBC\_conn;

**import** java.io.IOException;

**import** java.sql.Connection;

**import** java.sql.DriverManager;

**import** java.sql.ResultSet;

**import** java.sql.SQLException;

**import** java.sql.Statement;

**public** **class** Search\_Student\_by\_ID {

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

String url="jdbc:mysql://localhost:3306/mydb";

String user="root";

String password="Reddy@7125";

**try** {

Class.*forName*("com.mysql.cj.jdbc.Driver");

Connection con=DriverManager.*getConnection*(url,user,password);

System.***out***.println("Connection created");

Statement stmt =con.createStatement();

ResultSet rs=stmt.executeQuery("Select \* from students1 Where rollno=102 ");

//ResultSet rs1=stmt.executeQuery("");

System.***out***.println("rollno \t name \t Percentage \tEmail");

//while loop for fetching all the table records

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

**int** rollno =rs.getInt("rollno");

String name=rs.getString("name");

**int** per =rs.getInt("per");

String email=rs.getString("email");

System.***out***.println(rollno +" "+ name +" "+ per +" "+ email);

}

stmt.close();

con.close();

}**catch**(SQLException e) {

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

}**catch**( ClassNotFoundException e) {

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

}

}

}

Output; Connection created

rollno name Percentage Email

102 Reeva Sharma 94 Reeva@gmail.com

102 Reeva Sharma 94 Reeva@gmail.com

102 Reeva Sharma 94 Reeva@gmail.com

102 Reeva Sharma 94 Reeva@gmail.com

102 Reeva Sharma 94 Reeva@gmail.com

1. Implement an update operation to modify student details in the database using JDBC.

Code; **package** JDBC\_conn;

**import** java.sql.Connection;

**import** java.sql.DriverManager;

**import** java.sql.ResultSet;

**import** java.sql.SQLException;

**import** java.sql.Statement;

**public** **class** Update\_Student1 {

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

String url="jdbc:mysql://localhost:3306/mydb";

String user="root";

String password="Reddy@7125";

String updateSQL="insert into students1 values(105,'Rohit Sharma',90,'Rohit@gmail.com')";

**try** {

Class.*forName*("com.mysql.cj.jdbc.Driver");

Connection con=DriverManager.*getConnection*(url,user,password);

System.***out***.println("Connection created");

Statement stmt =con.createStatement();

**int** rowInserted =stmt.executeUpdate(updateSQL);

**if**(rowInserted>0) {

System.***out***.println("new Student record inserted ");

}

ResultSet rs=stmt.executeQuery("Select \* from students1");

System.***out***.println("rollno \t name \t per \t email");

//while loop for fetching all the table records

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

**int** rollno=rs.getInt("rollno");

String name=rs.getString("name");

**int** per =rs.getInt("per");

String email=rs.getString("email");

System.***out***.println(rollno +" "+ name +" "+ per+" "+ email);

}

stmt.close();

con.close();

}**catch**(SQLException e) {

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

}**catch**( ClassNotFoundException e) {

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

}

}

}

Output; Connection created

new Student record inserted

rollno name per email

101 Neeva Sharma 97 abc@gmail.com

102 Reeva Sharma 94 Reeva@gmail.com

101 Neeva Sharma 97 abc@gmail.com

102 Reeva Sharma 94 Reeva@gmail.com

104 Ajay Sharma 95 ajay@gmail.com

104 Ajay Sharma 95 ajay@gmail.com

104 Ajay Sharma 95 ajay@gmail.com

104 Ajay Sharma 95 ajay@gmail.com

101 Neeva Sharma 97 abc@gmail.com

102 Reeva Sharma 94 Reeva@gmail.com

103 Shiva Upadhyay 96 Shiva@gmail.com

101 Neeva Sharma 97 abc@gmail.com

102 Reeva Sharma 94 Reeva@gmail.com

103 Shiva Upadhyay 96 Shiva@gmail.com

101 Neeva Sharma 97 abc@gmail.com

102 Reeva Sharma 94 Reeva@gmail.com

103 Shiva Upadhyay 96 Shiva@gmail.com

105 Rohit Sharma 90 Rohit@gmail.com

105 Rohit Sharma 90 Rohit@gmail.com

1. Write a Java program to delete a student record from the database using JDBC.

Code; **package** JDBC\_conn;

**import** java.sql.Connection;

**import** java.sql.DriverManager;

**import** java.sql.ResultSet;

**import** java.sql.SQLException;

**import** java.sql.Statement;

**public** **class** Delete\_student {

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

String url="jdbc:mysql://localhost:3306/mydb";

String user="root";

String password="Reddy@7125";

String deleteSQL="DELETE FROM students1 WHERE rollno = 102";

**try** {

Class.*forName*("com.mysql.cj.jdbc.Driver");

Connection con=DriverManager.*getConnection*(url,user,password);

System.***out***.println("Connection created");

Statement stmt =con.createStatement();

**int** deleterow =stmt.executeUpdate(deleteSQL);

ResultSet rs=stmt.executeQuery("Select \* from students1");

System.***out***.println("rollno \t name \t per \t email");

//while loop for fetching all the table records

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

**int** rollno=rs.getInt("rollno");

String name=rs.getString("name");

**int** per =rs.getInt("per");

String email=rs.getString("email");

System.***out***.println(rollno +" "+ name +" "+ per+" "+ email);

}

stmt.close();

con.close();

}**catch**(SQLException e) {

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

}**catch**( ClassNotFoundException e) {

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

}

}

}

Output; Connection created

rollno name per email

101 Neeva Sharma 97 abc@gmail.com

101 Neeva Sharma 97 abc@gmail.com

104 Ajay Sharma 95 ajay@gmail.com

104 Ajay Sharma 95 ajay@gmail.com

104 Ajay Sharma 95 ajay@gmail.com

104 Ajay Sharma 95 ajay@gmail.com

101 Neeva Sharma 97 abc@gmail.com

103 Shiva Upadhyay 96 Shiva@gmail.com

101 Neeva Sharma 97 abc@gmail.com

103 Shiva Upadhyay 96 Shiva@gmail.com

101 Neeva Sharma 97 abc@gmail.com

103 Shiva Upadhyay 96 Shiva@gmail.com

105 Rohit Sharma 90 Rohit@gmail.com

105 Rohit Sharma 90 Rohit@gmail.com

1. Design a Java application to perform all CRUD (Create, Read, Update, Delete) operations on an **Employee** table using JDBC.

Code; **package** JDBC\_conn;

**import** java.sql.Connection;

**import** java.sql.DriverManager;

**import** java.sql.ResultSet;

**import** java.sql.SQLException;

**import** java.sql.Statement;

**public** **class** CRUD\_Employee {

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

String url="jdbc:mysql://localhost:3306/mydb";

String user="root";

String password="Reddy@7125";

String sql="create table if not exists Employee2(id int,"

+ "name varchar(50),"

+"salary int,"

+"dept varchar(50))";

String insertSQL="insert into Employee2 values(101,'Neeva Sharma',27000,'Testing'),"

+"(102,'Reeva Sharma',24000,'Management'),"

+"(103,'Shiva Upadhyay',26000,'Business')";

String updateSQL="UPDATE Employee2 SET salary = 30000, dept = 'DevOps' WHERE id = 101";

String deleteSQL="DELETE FROM Employee2 WHERE id = 102";

**try** {

Class.*forName*("com.mysql.cj.jdbc.Driver");

Connection con=DriverManager.*getConnection*(url,user,password);

System.***out***.println("Connection created");

Statement stmt =con.createStatement();

stmt.executeUpdate(sql);

System.***out***.println("Employee2 table created");

//call insert query using statement

**int** rowInserted =stmt.executeUpdate(insertSQL);

**if**(rowInserted>0) {

System.***out***.println("new Employee record inserted ");

}

System.***out***.println("Employee2 table before updating and delete operations:");

*displayTable*(stmt);

**int** updaterow =stmt.executeUpdate(updateSQL);

**int** deleterow=stmt.executeUpdate(deleteSQL);

System.***out***.println("Employee2 table after update/delete:");

*displayTable*(stmt);

stmt.close();

con.close();

}**catch**(SQLException e) {

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

}**catch**( ClassNotFoundException e) {

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

}

}

**public** **static** **void** displayTable(Statement stmt) **throws** SQLException {

ResultSet rs = stmt.executeQuery("SELECT \* FROM Employee2");

System.***out***.println("ID\tName\t\tSalary\tDept");

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

**int** id = rs.getInt("id");

String name = rs.getString("name");

**int** salary = rs.getInt("salary");

String dept = rs.getString("dept");

System.***out***.println(id + "\t" + name + "\t" + salary + "\t" + dept);

}

rs.close();

}

}

Output; Connection created

Employee2 table created

new Employee record inserted

Employee2 table before updating and delete operations:

ID Name Salary Dept

101 Neeva Sharma 27000 Testing

102 Reeva Sharma 24000 Management

103 Shiva Upadhyay 26000 Business

Employee2 table after update/delete:

ID Name Salary Dept

101 Neeva Sharma 30000 DevOps

103 Shiva Upadhyay 26000 Business

1. Create a JDBC-based program to count the total number of rows in a table.

Code; **package** JDBC\_conn;

**import** java.io.IOException;

**import** java.sql.Connection;

**import** java.sql.DriverManager;

**import** java.sql.PreparedStatement;

**import** java.sql.ResultSet;

**import** java.sql.SQLException;

**import** java.sql.Statement;

**public** **class** Row\_Count\_Employee2 {

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

String url="jdbc:mysql://localhost:3306/mydb";

String user="root";

String password="Reddy@7125";

String query = "SELECT COUNT(\*) FROM students1";

**try** (Connection connection = DriverManager.*getConnection*(url, user, password);

PreparedStatement statement = connection.prepareStatement(query);

ResultSet resultSet = statement.executeQuery()) {

**if** (resultSet.next()) {

**int** rowCount = resultSet.getInt(1);

System.***out***.println("Total number of rows in students1 table: " + rowCount);

}

} **catch** (SQLException e) {

System.***err***.println("Database error: " + e.getMessage());

}

}

}

Output; Total number of rows in students1 table: 14

1. Develop a program to sort student data in ascending order by name using SQL in JDBC.

Code; **package** JDBC\_conn;

**import** java.sql.Connection;

**import** java.sql.DriverManager;

**import** java.sql.PreparedStatement;

**import** java.sql.ResultSet;

**import** java.sql.SQLException;

**public** **class** Sort\_Students\_By\_Name\_ASC {

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

String url="jdbc:mysql://localhost:3306/mydb";

String user="root";

String password="Reddy@7125";

String query = "SELECT \* FROM students1 ORDER BY name ASC";

**try** (Connection connection = DriverManager.*getConnection*(url, user, password);

PreparedStatement statement = connection.prepareStatement(query);

ResultSet resultSet = statement.executeQuery()) {

System.***out***.println("Sorted Student Data by Name:");

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

**while** (resultSet.next()) {

**int** rollno = resultSet.getInt("rollno");

String name = resultSet.getString("name");

**int** per = resultSet.getInt("per");

String email=resultSet.getString("email");

System.***out***.println("Rollno: " + rollno + ", Name: " + name + ", Age: " + per+",Email:"+email);

}

} **catch** (SQLException e) {

System.***err***.println("Database error: " + e.getMessage());

}

}

}

Output; Sorted Student Data by Name:

--------------------------------

Rollno: 104, Name: Ajay Sharma, Age: 95,Email:ajay@gmail.com

Rollno: 104, Name: Ajay Sharma, Age: 95,Email:ajay@gmail.com

Rollno: 104, Name: Ajay Sharma, Age: 95,Email:ajay@gmail.com

Rollno: 104, Name: Ajay Sharma, Age: 95,Email:ajay@gmail.com

Rollno: 101, Name: Neeva Sharma, Age: 97,Email:abc@gmail.com

Rollno: 101, Name: Neeva Sharma, Age: 97,Email:abc@gmail.com

Rollno: 101, Name: Neeva Sharma, Age: 97,Email:abc@gmail.com

Rollno: 101, Name: Neeva Sharma, Age: 97,Email:abc@gmail.com

Rollno: 101, Name: Neeva Sharma, Age: 97,Email:abc@gmail.com

Rollno: 105, Name: Rohit Sharma, Age: 90,Email:Rohit@gmail.com

Rollno: 105, Name: Rohit Sharma, Age: 90,Email:Rohit@gmail.com

Rollno: 103, Name: Shiva Upadhyay, Age: 96,Email:Shiva@gmail.com

Rollno: 103, Name: Shiva Upadhyay, Age: 96,Email:Shiva@gmail.com

Rollno: 103, Name: Shiva Upadhyay, Age: 96,Email:Shiva@gmail.com

1. Write a program to display all students whose percentage is greater than 75 using JDBC and SQL WHERE clause.

Code; **package** JDBC\_conn;

**import** java.sql.Connection;

**import** java.sql.DriverManager;

**import** java.sql.PreparedStatement;

**import** java.sql.ResultSet;

**import** java.sql.SQLException;

**public** **class** Sort\_Students\_By\_percentage {

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

String url="jdbc:mysql://localhost:3306/mydb";

String user="root";

String password="Reddy@7125";

String query = "SELECT \* FROM students1 WHERE per>75 ";

**try** (Connection connection = DriverManager.*getConnection*(url, user, password);

PreparedStatement statement = connection.prepareStatement(query);

ResultSet resultSet = statement.executeQuery()) {

System.***out***.println("Sorted Student Data by Percentage Greater than 75:");

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

**while** (resultSet.next()) {

**int** rollno = resultSet.getInt("rollno");

String name = resultSet.getString("name");

**int** per = resultSet.getInt("per");

String email=resultSet.getString("email");

System.***out***.println("Rollno: " + rollno + ", Name: " + name + ", Percentage: " + per+",Email:"+email);

}

} **catch** (SQLException e) {

System.***err***.println("Database error: " + e.getMessage());

}

}

Output; Sorted Student Data by Percentage Greater than 75:

--------------------------------

Rollno: 101, Name: Neeva Sharma, Percentage: 97,Email:abc@gmail.com

Rollno: 101, Name: Neeva Sharma, Percentage: 97,Email:abc@gmail.com

Rollno: 104, Name: Ajay Sharma, Percentage: 95,Email:ajay@gmail.com

Rollno: 104, Name: Ajay Sharma, Percentage: 95,Email:ajay@gmail.com

Rollno: 104, Name: Ajay Sharma, Percentage: 95,Email:ajay@gmail.com

Rollno: 104, Name: Ajay Sharma, Percentage: 95,Email:ajay@gmail.com

Rollno: 101, Name: Neeva Sharma, Percentage: 97,Email:abc@gmail.com

Rollno: 103, Name: Shiva Upadhyay, Percentage: 96,Email:Shiva@gmail.com

Rollno: 101, Name: Neeva Sharma, Percentage: 97,Email:abc@gmail.com

Rollno: 103, Name: Shiva Upadhyay, Percentage: 96,Email:Shiva@gmail.com

Rollno: 101, Name: Neeva Sharma, Percentage: 97,Email:abc@gmail.com

Rollno: 103, Name: Shiva Upadhyay, Percentage: 96,Email:Shiva@gmail.com

Rollno: 105, Name: Rohit Sharma, Percentage: 90,Email:Rohit@gmail.com

Rollno: 105, Name: Rohit Sharma, Percentage: 90,Email:Rohit@gmail.com

1. Use **PreparedStatement** to insert multiple student records into the database.

Code; **package** JDBC\_conn;

**import** java.sql.Connection;

**import** java.sql.DriverManager;

**import** java.sql.PreparedStatement;

**import** java.sql.ResultSet;

**import** java.sql.SQLException;

**public** **class** insert\_multiple\_studdata {

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

String url="jdbc:mysql://localhost:3306/mydb";

String user="root";

String password="Reddy@7125";

String insertQuery = "INSERT INTO students1 (rollno, name, per, email) VALUES (?, ?, ?, ?)";

**try** (Connection connection = DriverManager.*getConnection*(url, user, password);

PreparedStatement statement = connection.prepareStatement(insertQuery)) {

statement.setInt(1, 106);

statement.setString(2, "Suriya");

statement.setInt(3, 90);

statement.setString(4, "suri@gmail.com");

statement.addBatch();

statement.setInt(1, 107);

statement.setString(2, "Raja");

statement.setInt(3, 89);

statement.setString(4, "raja@gmail.com");

statement.addBatch();

statement.setInt(1, 108);

statement.setString(2, "Nagireddy");

statement.setInt(3, 96);

statement.setString(4, "nagi@gmail.com");

statement.addBatch();

**int**[] rowsInserted = statement.executeBatch();

System.***out***.println("Inserted " + rowsInserted.length + " student records successfully.");

} **catch** (SQLException e) {

System.***err***.println("Database error: " + e.getMessage());

}

}

}

Output; Inserted 3 student records successfully.

1. Implement a program using **transaction management** in JDBC (i.e., commit and rollback).

Code; **package** JDBC\_conn;

**import** java.sql.\*;

**public** **class** TransactionExample {

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

String url="jdbc:mysql://localhost:3306/mydb";

String user="root";

String password="Reddy@7125";

String createsql="create table if not exists accounts(id int PRIMARY KEY,"

+ "name varchar(50),"

+"balance DOUBLE)";

// String insertSQL="insert into accounts values(1,'Neeva Sharma',2000),"

// +"(2,'Reeva Sharma',4000),"

// +"(3,'Shiva Upadhyay',6000)";

Connection conn = **null**;

**try** {

conn = DriverManager.*getConnection*(url, user, password);

Statement stmt =conn.createStatement();

//call create query using statement

stmt.executeUpdate(createsql);

//stmt.executeUpdate(insertSQL);

conn.setAutoCommit(**false**); // Start transaction

// Withdraw ₹1000 from Account A (id = 1)

PreparedStatement withdrawStmt = conn.prepareStatement(

"UPDATE accounts SET balance = balance - ? WHERE id = ?");

withdrawStmt.setDouble(1, 1000);

withdrawStmt.setInt(2, 1);

withdrawStmt.executeUpdate();

// Deposit ₹1000 to Account B (id = 2)

PreparedStatement depositStmt = conn.prepareStatement(

"UPDATE accounts SET balance = balance + ? WHERE id = ?");

depositStmt.setDouble(1, 1000);

depositStmt.setInt(2, 2);

depositStmt.executeUpdate();

// Commit transaction

conn.commit();

System.***out***.println("Transaction successful!");

System.***out***.println("Accounts table after transaction.");

ResultSet rs=stmt.executeQuery("Select \* from accounts");

System.***out***.println("id \t name \t balance ");

//while loop for fetching all the table records

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

**int** id =rs.getInt("id");

String name=rs.getString("name");

**double** balance =rs.getInt("balance");

System.***out***.println(id +" "+ name +" "+ balance);

}

} **catch** (SQLException e) {

System.***out***.println("Transaction failed. Rolling back...");

**try** {

**if** (conn != **null**) conn.rollback();

} **catch** (SQLException rollbackEx) {

System.***out***.println("Rollback failed: " + rollbackEx.getMessage());

}

e.printStackTrace();

} **finally** {

**try** {

**if** (conn != **null**) conn.setAutoCommit(**true**); // Restore default

**if** (conn != **null**) conn.close();

} **catch** (SQLException closeEx) {

closeEx.printStackTrace();

}

}

}

}

Output; Transaction successful!

Accounts table after transaction.

id name balance

1 Neeva Sharma 0.0

2 Reeva Sharma 6000.0

3 Shiva Upadhyay 6000.0

Write a JDBC program to handle exceptions (like invalid ID, connection errors) gracefully.

Code; **package** JDBC\_conn;

**import** java.sql.\*;

**public** **class** Transaction\_exception\_Handling {

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

String url = "jdbc:mysql://localhost:3306/mydb";

String user = "root";

String password = "Reddy@7125";

String createSQL = "CREATE TABLE IF NOT EXISTS accounts (" +

"id INT PRIMARY KEY, " +

"name VARCHAR(50), " +

"balance DOUBLE)";

Connection conn = **null**;

**try** {

conn = DriverManager.*getConnection*(url, user, password);

conn.setAutoCommit(**false**);

**try** (Statement stmt = conn.createStatement()) {

stmt.executeUpdate(createSQL);

// Optional: Insert sample data if needed

//stmt.executeUpdate("INSERT INTO accounts VALUES (1,'Sharma',6000), (2,'Reeva Sharma',4000)");

// Withdraw ₹1000 from Account A (id = 1)

PreparedStatement withdrawStmt = conn.prepareStatement(

"UPDATE accounts SET balance = balance - ? WHERE id = ?");

withdrawStmt.setDouble(1, 1000);

withdrawStmt.setInt(2, 1);

**int** rowsWithdrawn = withdrawStmt.executeUpdate();

// Deposit ₹1000 to Account B (id = 2)

PreparedStatement depositStmt = conn.prepareStatement(

"UPDATE accounts SET balance = balance + ? WHERE id = ?");

depositStmt.setDouble(1, 1000);

depositStmt.setInt(2, 2);

**int** rowsDeposited = depositStmt.executeUpdate();

**if** (rowsWithdrawn == 0 || rowsDeposited == 0) {

**throw** **new** SQLException("Invalid account ID(s). Transaction aborted.");

}

conn.commit();

System.***out***.println("Transaction successful!");

// Display updated table

ResultSet rs = stmt.executeQuery("SELECT \* FROM accounts");

System.***out***.println("\nid\tname\t\tbalance");

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

**int** id = rs.getInt("id");

String name = rs.getString("name");

**double** balance = rs.getDouble("balance");

System.***out***.printf("%d\t%-15s\t%.2f%n", id, name, balance);

}

} **catch** (SQLException e) {

System.***out***.println("Error during transaction: " + e.getMessage());

**if** (conn != **null**) conn.rollback();

System.***out***.println("Rolled back transaction.");

} **finally** {

**if** (conn != **null**) conn.setAutoCommit(**true**);

}

} **catch** (SQLException connEx) {

System.***out***.println("Connection failed: " + connEx.getMessage());

} **finally** {

**try** {

**if** (conn != **null**) conn.close();

} **catch** (SQLException closeEx) {

System.***out***.println("Error closing connection: " + closeEx.getMessage());

}

}

}

}

Output; Transaction successful!

id name balance

1 Neeva Sharma -3000.00

2 Reeva Sharma 9000.00

3 Shiva Upadhyay 6000.00

1. Create a login system using JDBC where user credentials are verified from the database.

Code; **package** JDBC\_conn;

**import** java.sql.\*;

**import** java.util.Scanner;

**public** **class** LoginSystem {

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

String url = "jdbc:mysql://localhost:3306/mydb";

String user = "root";

String password = "Reddy@7125";

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

System.***out***.print("Enter username: ");

String inputUsername = sc.nextLine();

System.***out***.print("Enter password: ");

String inputPassword = sc.nextLine();

String createTableSQL = "CREATE TABLE IF NOT EXISTS users (" +

"username VARCHAR(50) PRIMARY KEY, " +

"password VARCHAR(50) NOT NULL)";

String query = "SELECT \* FROM users WHERE username = ? AND password = ?";

String insertquery=" INSERT INTO users VALUES ('admin', 'admin123'), ('jagadheeswar', 'java@2025')";

**try** (Connection conn = DriverManager.*getConnection*(url, user, password);

Statement stmt = conn.createStatement()) {

// Create table if not exists

stmt.executeUpdate(createTableSQL);

stmt.executeUpdate(insertquery);

// Verify credentials

**try** (PreparedStatement pstmt = conn.prepareStatement(query)) {

pstmt.setString(1, inputUsername);

pstmt.setString(2, inputPassword);

ResultSet rs = pstmt.executeQuery();

**if** (rs.next()) {

System.***out***.println("Login successful. Welcome, " + inputUsername + "!");

} **else** {

System.***out***.println("Invalid username or password.");

}

}

} **catch** (SQLException e) {

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

}

}

}

Output; Enter username: admin

Enter password: admin123

Login successful. Welcome, admin!

1. Implement a Java application to take dynamic input from the user and perform insertion, search, or update using menu-driven logic.

Code; **package** JDBC\_conn;

**import** java.sql.\*;

**import** java.util.Scanner;

**public** **class** MenuDriven\_System {

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

String url = "jdbc:mysql://localhost:3306/mydb";

String user = "root";

String password = "Reddy@7125";

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

**try** (Connection conn = DriverManager.*getConnection*(url, user, password);

Statement stmt = conn.createStatement()) {

// Create table if not exists

// String createSQL = "CREATE TABLE IF NOT EXISTS accounts (" +

// "id INT PRIMARY KEY, " +

// "name VARCHAR(50), " +

// "balance DOUBLE)";

// stmt.executeUpdate(createSQL);

**while** (**true**) {

System.***out***.println("\n=== MENU ===");

System.***out***.println("1. Insert Account");

System.***out***.println("2. Search Account by ID");

System.***out***.println("3. Update Account Balance");

System.***out***.println("4. Exit");

System.***out***.print("Choose an option: ");

**int** choice = sc.nextInt();

sc.nextLine(); // consume newline

**switch** (choice) {

**case** 1:

System.***out***.print("Enter ID: ");

**int** id = sc.nextInt();

sc.nextLine();

System.***out***.print("Enter Name: ");

String name = sc.nextLine();

System.***out***.print("Enter Balance: ");

**double** balance = sc.nextDouble();

String insertSQL = "INSERT INTO accounts VALUES (?, ?, ?)";

**try** (PreparedStatement pstmt = conn.prepareStatement(insertSQL)) {

pstmt.setInt(1, id);

pstmt.setString(2, name);

pstmt.setDouble(3, balance);

pstmt.executeUpdate();

System.***out***.println("Account inserted successfully.");

} **catch** (SQLException e) {

System.***out***.println("Error inserting account: " + e.getMessage());

}

**break**;

**case** 2:

System.***out***.print("Enter ID to search: ");

**int** searchId = sc.nextInt();

String searchSQL = "SELECT \* FROM accounts WHERE id = ?";

**try** (PreparedStatement pstmt = conn.prepareStatement(searchSQL)) {

pstmt.setInt(1, searchId);

ResultSet rs = pstmt.executeQuery();

**if** (rs.next()) {

System.***out***.println("ID: " + rs.getInt("id"));

System.***out***.println("Name: " + rs.getString("name"));

System.***out***.println("Balance: ₹" + rs.getDouble("balance"));

} **else** {

System.***out***.println("Account not found.");

}

}

**break**;

**case** 3:

System.***out***.print("Enter ID to update: ");

**int** updateId = sc.nextInt();

System.***out***.print("Enter new balance: ");

**double** newBalance = sc.nextDouble();

String updateSQL = "UPDATE accounts SET balance = ? WHERE id = ?";

**try** (PreparedStatement pstmt = conn.prepareStatement(updateSQL)) {

pstmt.setDouble(1, newBalance);

pstmt.setInt(2, updateId);

**int** rows = pstmt.executeUpdate();

**if** (rows > 0) {

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

} **else** {

System.***out***.println("Account not found.");

}

}

**break**;

**case** 4:

System.***out***.println("Exiting program. Goodbye!");

**return**;

**default**:

System.***out***.println("Invalid choice. Try again.");

}

}

} **catch** (SQLException e) {

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

}

}

}

Output; === MENU ===

1. Insert Account

2. Search Account by ID

3. Update Account Balance

4. Exit

Choose an option: 1

Enter ID: 3

Enter Name: shiva aadhav

Enter Balance: 3000

Error inserting account: Duplicate entry '3' for key 'accounts.PRIMARY'

=== MENU ===

1. Insert Account

2. Search Account by ID

3. Update Account Balance

4. Exit

Choose an option: 1

Enter ID: 4

Enter Name: ravi

Enter Balance: 4000

Account inserted successfully.

=== MENU ===

1. Insert Account

2. Search Account by ID

3. Update Account Balance

4. Exit

Choose an option: 2

Enter ID to search: 2

ID: 2

Name: Reeva Sharma

Balance: ₹9000.0

=== MENU ===

1. Insert Account

2. Search Account by ID

3. Update Account Balance

4. Exit

Choose an option: 3

Enter ID to update: 2

Enter new balance: 10000

Balance updated successfully.

=== MENU ===

1. Insert Account

2. Search Account by ID

3. Update Account Balance

4. Exit

Choose an option: 4

Exiting program. Goodbye!

1. Design the schema for a **Library Management System** and write JDBC programs for:

* Adding a book
* Viewing all books
* Issuing a book to a member
* Returning a book

Code; **package** JDBC\_conn;

**import** java.sql.\*;

**import** java.util.Scanner;

**public** **class** LibrarySystem {

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

String url = "jdbc:mysql://localhost:3306/mydb";

String user = "root";

String password = "Reddy@7125";

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

**try** (Connection conn = DriverManager.*getConnection*(url, user, password);

Statement stmt = conn.createStatement()) {

// Create tables if not exist

String createBooks = "CREATE TABLE IF NOT EXISTS books (" +

"book\_id INT PRIMARY KEY, " +

"title VARCHAR(100), " +

"author VARCHAR(100), " +

"available BOOLEAN DEFAULT TRUE)";

String createIssued = "CREATE TABLE IF NOT EXISTS issued\_books (" +

"issue\_id INT PRIMARY KEY AUTO\_INCREMENT, " +

"book\_id INT, " +

"member\_name VARCHAR(100), " +

"issue\_date DATE, " +

"return\_date DATE, " +

"FOREIGN KEY (book\_id) REFERENCES books(book\_id))";

stmt.executeUpdate(createBooks);

stmt.executeUpdate(createIssued);

**while** (**true**) {

System.***out***.println("\n=== LIBRARY MENU ===");

System.***out***.println("1. Add Book");

System.***out***.println("2. View All Books");

System.***out***.println("3. Issue Book");

System.***out***.println("4. Return Book");

System.***out***.println("5. Exit");

System.***out***.print("Choose an option: ");

**int** choice = sc.nextInt();

sc.nextLine(); // consume newline

**switch** (choice) {

**case** 1: // Add Book

System.***out***.print("Enter Book ID: ");

**int** bookId = sc.nextInt();

sc.nextLine();

System.***out***.print("Enter Title: ");

String title = sc.nextLine();

System.***out***.print("Enter Author: ");

String author = sc.nextLine();

String insertBook = "INSERT INTO books VALUES (?, ?, ?, TRUE)";

**try** (PreparedStatement pstmt = conn.prepareStatement(insertBook)) {

pstmt.setInt(1, bookId);

pstmt.setString(2, title);

pstmt.setString(3, author);

pstmt.executeUpdate();

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

} **catch** (SQLException e) {

System.***out***.println("Error adding book: " + e.getMessage());

}

**break**;

**case** 2: // View All Books

ResultSet rs = stmt.executeQuery("SELECT \* FROM books");

System.***out***.println("\nID\tTitle\t\tAuthor\t\tAvailable");

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

System.***out***.printf("%d\t%-15s\t%-15s\t%s%n",

rs.getInt("book\_id"),

rs.getString("title"),

rs.getString("author"),

rs.getBoolean("available") ? "Yes" : "No");

}

**break**;

**case** 3: // Issue Book

System.***out***.print("Enter Book ID to issue: ");

**int** issueBookId = sc.nextInt();

sc.nextLine();

System.***out***.print("Enter Member Name: ");

String member = sc.nextLine();

String checkAvailability = "SELECT available FROM books WHERE book\_id = ?";

**try** (PreparedStatement checkStmt = conn.prepareStatement(checkAvailability)) {

checkStmt.setInt(1, issueBookId);

ResultSet checkRs = checkStmt.executeQuery();

**if** (checkRs.next() && checkRs.getBoolean("available")) {

String issueSQL = "INSERT INTO issued\_books (book\_id, member\_name, issue\_date) VALUES (?, ?, CURDATE())";

String updateBook = "UPDATE books SET available = FALSE WHERE book\_id = ?";

**try** (PreparedStatement issueStmt = conn.prepareStatement(issueSQL);

PreparedStatement updateStmt = conn.prepareStatement(updateBook)) {

issueStmt.setInt(1, issueBookId);

issueStmt.setString(2, member);

issueStmt.executeUpdate();

updateStmt.setInt(1, issueBookId);

updateStmt.executeUpdate();

System.***out***.println("Book issued to " + member);

}

} **else** {

System.***out***.println("Book not available or doesn't exist.");

}

}

**break**;

**case** 4: // Return Book

System.***out***.print("Enter Book ID to return: ");

**int** returnBookId = sc.nextInt();

String returnSQL = "UPDATE issued\_books SET return\_date = CURDATE() WHERE book\_id = ? AND return\_date IS NULL";

String updateAvailability = "UPDATE books SET available = TRUE WHERE book\_id = ?";

**try** (PreparedStatement returnStmt = conn.prepareStatement(returnSQL);

PreparedStatement updateStmt = conn.prepareStatement(updateAvailability)) {

returnStmt.setInt(1, returnBookId);

**int** rows = returnStmt.executeUpdate();

**if** (rows > 0) {

updateStmt.setInt(1, returnBookId);

updateStmt.executeUpdate();

System.***out***.println("Book returned successfully.");

} **else** {

System.***out***.println("No active issue found for this book.");

}

}

**break**;

**case** 5:

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

**return**;

**default**:

System.***out***.println("Invalid choice. Try again.");

}

}

} **catch** (SQLException e) {

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

}

}

}

Output; === LIBRARY MENU ===

1. Add Book

2. View All Books

3. Issue Book

4. Return Book

5. Exit

Choose an option: 1

Enter Book ID: 1

Enter Title: wings of fire

Enter Author: Dr.A P J Abdul kalam

Book added successfully.

=== LIBRARY MENU ===

1. Add Book

2. View All Books

3. Issue Book

4. Return Book

5. Exit

Choose an option: 2

ID Title Author Available

1 wings of fire Dr.A P J Abdul kalam Yes

=== LIBRARY MENU ===

1. Add Book

2. View All Books

3. Issue Book

4. Return Book

5. Exit

Choose an option: 3

Enter Book ID to issue: 1

Enter Member Name: ravi

Book issued to ravi

=== LIBRARY MENU ===

1. Add Book

2. View All Books

3. Issue Book

4. Return Book

5. Exit

Choose an option: 4

Enter Book ID to return: 1

Book returned successfully.

=== LIBRARY MENU ===

1. Add Book

2. View All Books

3. Issue Book

4. Return Book

5. Exit

Choose an option: 5

Exiting Library System.

1. Create a **Hospital Management System** database. Using JDBC, implement:

* Register new patient
* Assign doctor
* Generate billing

Code; **package** JDBC\_conn;

**import** java.sql.\*;

**import** java.util.Scanner;

**public** **class** HospitalSystem {

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

String url = "jdbc:mysql://localhost:3306/mydb";

String user = "root";

String password = "Reddy@7125";

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

**try** (Connection conn = DriverManager.*getConnection*(url, user, password);

Statement stmt = conn.createStatement()) {

// Create tables

stmt.executeUpdate("CREATE TABLE IF NOT EXISTS patients (" +

"patient\_id INT PRIMARY KEY, " +

"name VARCHAR(100), " +

"age INT, " +

"disease VARCHAR(100))");

stmt.executeUpdate("CREATE TABLE IF NOT EXISTS doctors (" +

"doctor\_id INT PRIMARY KEY, " +

"name VARCHAR(100), " +

"specialization VARCHAR(100))");

stmt.executeUpdate("CREATE TABLE IF NOT EXISTS billing (" +

"bill\_id INT PRIMARY KEY AUTO\_INCREMENT, " +

"patient\_id INT, " +

"doctor\_id INT, " +

"amount DOUBLE, " +

"billing\_date DATE, " +

"FOREIGN KEY (patient\_id) REFERENCES patients(patient\_id), " +

"FOREIGN KEY (doctor\_id) REFERENCES doctors(doctor\_id))");

**while** (**true**) {

System.***out***.println("\n=== HOSPITAL MENU ===");

System.***out***.println("1. Register New Patient");

System.***out***.println("2. Assign Doctor");

System.***out***.println("3. Generate Billing");

System.***out***.println("4. Exit");

System.***out***.print("Choose an option: ");

**int** choice = sc.nextInt();

sc.nextLine(); // consume newline

**switch** (choice) {

**case** 1: // Register Patient

System.***out***.print("Enter Patient ID: ");

**int** pid = sc.nextInt();

sc.nextLine();

System.***out***.print("Enter Name: ");

String pname = sc.nextLine();

System.***out***.print("Enter Age: ");

**int** age = sc.nextInt();

sc.nextLine();

System.***out***.print("Enter Disease: ");

String disease = sc.nextLine();

String insertPatient = "INSERT INTO patients VALUES (?, ?, ?, ?)";

**try** (PreparedStatement pstmt = conn.prepareStatement(insertPatient)) {

pstmt.setInt(1, pid);

pstmt.setString(2, pname);

pstmt.setInt(3, age);

pstmt.setString(4, disease);

pstmt.executeUpdate();

System.***out***.println("Patient registered.");

} **catch** (SQLException e) {

System.***out***.println("Error: " + e.getMessage());

}

**break**;

**case** 2: // Assign Doctor

System.***out***.print("Enter Doctor ID: ");

**int** did = sc.nextInt();

sc.nextLine();

System.***out***.print("Enter Name: ");

String dname = sc.nextLine();

System.***out***.print("Enter Specialization: ");

String spec = sc.nextLine();

String insertDoctor = "INSERT INTO doctors VALUES (?, ?, ?)";

**try** (PreparedStatement pstmt = conn.prepareStatement(insertDoctor)) {

pstmt.setInt(1, did);

pstmt.setString(2, dname);

pstmt.setString(3, spec);

pstmt.executeUpdate();

System.***out***.println("Doctor assigned.");

} **catch** (SQLException e) {

System.***out***.println("Error: " + e.getMessage());

}

**break**;

**case** 3: // Generate Billing

System.***out***.print("Enter Patient ID: ");

**int** billPid = sc.nextInt();

System.***out***.print("Enter Doctor ID: ");

**int** billDid = sc.nextInt();

System.***out***.print("Enter Amount: ");

**double** amount = sc.nextDouble();

String billSQL = "INSERT INTO billing (patient\_id, doctor\_id, amount, billing\_date) " +

"VALUES (?, ?, ?, CURDATE())";

**try** (PreparedStatement pstmt = conn.prepareStatement(billSQL)) {

pstmt.setInt(1, billPid);

pstmt.setInt(2, billDid);

pstmt.setDouble(3, amount);

pstmt.executeUpdate();

System.***out***.println("Billing generated successfully.");

} **catch** (SQLException e) {

System.***out***.println("Error: " + e.getMessage());

}

**break**;

**case** 4:

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

**return**;

**default**:

System.***out***.println("Invalid choice. Try again.");

}

}

} **catch** (SQLException e) {

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

}

}

}

Output; === HOSPITAL MENU ===

1. Register New Patient

2. Assign Doctor

3. Generate Billing

4. Exit

Choose an option: 1

Enter Patient ID: 100

Enter Name: abdul khadeer

Enter Age: 22

Enter Disease: migrane

Patient registered.

=== HOSPITAL MENU ===

1. Register New Patient

2. Assign Doctor

3. Generate Billing

4. Exit

Choose an option: 2

Enter Doctor ID: 201

Enter Name: jai prakash

Enter Specialization: ENT

Doctor assigned.

=== HOSPITAL MENU ===

1. Register New Patient

2. Assign Doctor

3. Generate Billing

4. Exit

Choose an option: 3

Enter Patient ID: 100

Enter Doctor ID: 201

Enter Amount: 2000

Billing generated successfully.

=== HOSPITAL MENU ===

1. Register New Patient

2. Assign Doctor

3. Generate Billing

4. Exit

Choose an option: 4

Exiting Hospital System.

1. Write a JDBC-based report generator that exports data from a MySQL table to a text or CSV file.

Code; **package** JDBC\_conn;

**import** java.sql.\*;

**import** java.io.\*;

**public** **class** ReportGenerator {

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

String url = "jdbc:mysql://localhost:3306/mydb";

String user = "root";

String password = "Reddy@7125";

String query = "SELECT \* FROM students1";

**try** (Connection conn = DriverManager.*getConnection*(url, user, password);

Statement stmt = conn.createStatement();

ResultSet rs = stmt.executeQuery(query)) {

File file = **new** File("students\_report.csv");

**try** (BufferedWriter writer = **new** BufferedWriter(**new** FileWriter(file))) {

// Write header

writer.write("Rollno,Name,per,Email");

writer.newLine();

// Write data rows

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

**int** rollno = rs.getInt("rollno");

String name = rs.getString("name");

**int** per = rs.getInt("per");

String email = rs.getString("email");

String line = rollno + "," + name + "," + per + "," + email ;

writer.write(line);

writer.newLine();

}

System.***out***.println("Report generated: " + file.getAbsolutePath());

} **catch** (IOException e) {

System.***out***.println("File writing error: " + e.getMessage());

}

} **catch** (SQLException e) {

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

}

}

}

Output; Report generated: C:\Users\tatip\git\Java\_Selenium\_Projects\JDBC\_Connectivity\_Project\JDBC\_connectivity\students\_report.csv