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

Program:

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

public class Connection {
   public static void main(String[] args) {
          String url="jdbc:mysql://localhost:3306/java";

                 String user="root";
                 String password="root";

      try {
         Connection conn = DriverManager.getConnection(url, user, password);
         System.out.println("Connected to database");
         conn.close();
      } catch (SQLException e) {
         System.out.println("Error"+e.getMessage());
      }
   }
}
```

Output:

```
Connected to database
```

---

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

Program:

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.SQLException;

public class insertDetails {
   public static void main(String[] args) {
      String url="jdbc:mysql://localhost:3306/java";
      String user="root";
      String password="root";

      try (Connection con=DriverManager.getConnection(url, user, password)) {
         String insertSQL="INSERT INTO student (rollno, name, per, email) VALUES (?, ?, ?, ?)";
         try (
        PreparedStatement pstmt=con.prepareStatement(insertSQL)) {
            pstmt.setInt(1, 105);
            pstmt.setString(2, "Jayanth");
            pstmt.setInt(3, 99);
            pstmt.setString(4, "jayanth@gmail.com");
            int rowInserted=pstmt.executeUpdate();
            if (rowInserted>0) {
                System.out.println("New record inserted");
            }
        }
    } catch (SQLException e) {
        System.out.println(e);
    }
  }
}
```

Output:

```
New record inserted
```

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

Program:

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

public class FetchDetails {
    public static void main(String[] args) {
        String url="jdbc:mysql://localhost:3306/java";
        String user="root";
        String password="root";

        try (Connection con=DriverManager.getConnection(url, user, password);
             Statement stmt=con.createStatement();
             ResultSet rs=stmt.executeQuery("select * from Student")) {

            System.out.println("rollno\tname\tpercent\tEmail");
            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+"\t"+per+"\t"+email);
            }

        } catch (SQLException e) {
            System.out.println(e);
        }
    }
}
```

OutPut:

```
rollno   name   percent           Email
105    Jayanth 99       jayanth@gmail.com
```

---

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

Program:

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.Scanner;

public class SearchDetails {
    public static void main(String[] args) {
        String url="jdbc:mysql://localhost:3306/java";
        String user="root";
        String password="root";

        try (Connection con=DriverManager.getConnection(url, user, password)) {
            Scanner scanner=new Scanner(System.in);
            System.out.print("Enter student Id");
            int rollno=scanner.nextInt();
            scanner.close();

            String query = "select * from Student where rollno = ?";
            try (PreparedStatement pstmt = con.prepareStatement(query)) {
                pstmt.setInt(1, rollno);
                try (ResultSet rs=pstmt.executeQuery()) {
                    if (rs.next()) {
                        System.out.println("Student found");
                        System.out.println("Roll No"+rs.getInt("rollno"));
                        System.out.println("Name"+rs.getString("name"));
                        System.out.println("Percentage"+rs.getInt("per"));
                        System.out.println("Email"+ rs.getString("email"));
                    } else {
                        System.out.println("Student not found");
                    }
                }
            }
        } catch (SQLException e) {
            System.out.println(e);
        }
    }
}
```

Output:

```
Enter student Id105
Student found
Roll No105
NameJayanth
Percentage99
Emailjayanth@gmail.com
```

---

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

Program:

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.SQLException;

public class ModifyDetails {
    public static void main(String[] args) {
        String url="jdbc:mysql://localhost:3306/java";
        String user="root";
        String password="root";

        int rollno=1;
        String name="raju";
        int per=60;
        String email="raju@gmail.com";
        try (Connection con=DriverManager.getConnection(url, user, password)) {
            String query="update Student set name=?, per=?, email=? where rollno =?";
            try (PreparedStatement pstmt = con.prepareStatement(query)) {
                pstmt.setString(1, name);
                pstmt.setInt(2, per);
                pstmt.setString(3, email);
                pstmt.setInt(4, rollno);
                int rowsUpdated = pstmt.executeUpdate();
                if (rowsUpdated>0) {
                    System.out.println("Student details updated");
                } else {
                    System.out.println("Student not found");
                }
            }
        } catch (SQLException e) {
            System.out.println(e);
        }
    }
}
```

Output:

```
Student details updated
```

---

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

Program:

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.SQLException;

public class DeleteDetails {
    public static void main(String[] args) {
        String url="jdbc:mysql://localhost:3306/java";
        String user="root";
        String password="root";
        int rollno=10;

        try (Connection con=DriverManager.getConnection(url, user, password)) {
            String query = "delete from Student where rollno = ?";
            try (PreparedStatement pstmt=con.prepareStatement(query)) {
                pstmt.setInt(1, rollno);
                int rowsDeleted=pstmt.executeUpdate();
                if (rowsDeleted>0) {
                    System.out.println("Student record deleted");
                } else {
                    System.out.println("Student not found");
                }
            }
        } catch (SQLException e) {
            System.out.println(e);
        }
    }
}
```

Output:

```
Student record deleted
```

---

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

Program:

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;

public class CRUD {
    public static void main(String[] args) {
        String url = "jdbc:mysql://localhost:3306/java";
        String user = "root";
        String password = "root";

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

            createTable(con);
            insertEmployee(con,1,"A", 60000,"development","delhi");
            insertEmployee(con,2,"B", 30000, "testing","hyd");
            insertEmployee(con,3,"C", 90000,"management","banglore");
            insertEmployee(con,4,"D", 40000,"development","mumbai");
            insertEmployee(con,5,"E", 60000, "testing","pune");
            System.out.println("All Employees:");
            displayEmployees(con);
            System.out.println("\nUpdate Employee:");
            updateEmployee(con,2,"B Updated",35000,"testing updated","hyd
updated",9000001);
            displayEmployees(con);
            System.out.println("\nDelete Employee:");
            deleteEmployee(con, 3);
            displayEmployees(con);
        } catch (SQLException e) {
            System.out.println(e);
        }
    }

    public static void createTable(Connection con) throws SQLException {
        String query = "create table if not exists Emp12 (id int, name varchar(50), salary int,
department varchar(50), city varchar(50))";
        try (PreparedStatement pstmt=con.prepareStatement(query)) {
            pstmt.executeUpdate();
        }
    }

    public static void insertEmployee(Connection con, int id, String name, int salary, String
department, String city) throws SQLException {
        String query = "insert into Emp12 values (?, ?, ?, ?, ?)";
        try (PreparedStatement pstmt=con.prepareStatement(query)) {
            pstmt.setInt(1, id);
            pstmt.setString(2, name);
            pstmt.setInt(3, salary);
            pstmt.setString(4, department);
            pstmt.setString(5, city);
            pstmt.executeUpdate();
        }
    }

    public static void displayEmployees(Connection con) throws SQLException {
        String query="select * from Emp12";
        try (PreparedStatement pstmt=con.prepareStatement(query);
             ResultSet rs=pstmt.executeQuery()) {
            while (rs.next()) {
                System.out.println(rs.getInt("id")+" "+rs.getString("name")+" "+rs.getInt("salary")+"
"+rs.getString("department")+" "+rs.getString("city"));
            }
        }
    }

    public static void updateEmployee(Connection con, int id, String name, int salary, String
department, String city, long phone) throws SQLException {
        String query = "update Emp12 set name=?, salary=?, department=?, city=? where id=?";
        try (PreparedStatement pstmt = con.prepareStatement(query)) {
            pstmt.setString(1, name);
            pstmt.setInt(2, salary);
            pstmt.setString(3, department);
            pstmt.setString(4, city);
            pstmt.setInt(5, id);
            pstmt.executeUpdate();
        }
    }

    public static void deleteEmployee(Connection con, int id) throws SQLException {
        String query = "delete from Emp12 where id=?";
        try (PreparedStatement pstmt=con.prepareStatement(query)) {
            pstmt.setInt(1, id);
            pstmt.executeUpdate();
        }
    }
}
```
Output:
```
All Employees:
1 A 60000 development delhi
2 B 30000 testing hyd
3 C 90000 management banglore
4 D 40000 development mumbai
5 E 60000 testing pune

Update Employee:
1 A 60000 development delhi
2 B Updated 35000 testing updated hyd updated
3 C 90000 management banglore
4 D 40000 development mumbai
5 E 60000 testing pune

Delete Employee:
1 A 60000 development delhi
2 B Updated 35000 testing updated hyd updated
```

```
4 D 40000 development mumbai
5 E 60000 testing pune
```

---

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

Program:

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;

public class CountRows {
   public static void main(String[] args) {
      String url="jdbc:mysql://localhost:3306/java";
      String user="root";
      String password="root";
      String tableName="Emp12";

      try (Connection con=DriverManager.getConnection(url, user, password)) {
         int rowCount=countRows(con, tableName);
         System.out.println("Total rows in"+tableName+rowCount);
      } catch (SQLException e) {
         System.out.println(e);
      }
   }

   public static int countRows(Connection con, String tableName) throws SQLException {
      String query = "Sselect count(*) from"+tableName;
      try (PreparedStatement pstmt=con.prepareStatement(query);
          ResultSet rs=pstmt.executeQuery()) {
         if (rs.next()) {
            return rs.getInt(1);
         } else {
            return 0;
         }
      }
   }
}
```

---

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

Program;

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;

public class SortDetails {
   public static void main(String[] args) {
      String url="jdbc:mysql://localhost:3306/java";
      String user="root";
      String password="root";
      String tableName="Student";

      try (Connection con = DriverManager.getConnection(url, user, password)) {
         System.out.println("Students in ascendingby name:");
         displayStudents(con, tableName);
      } catch (SQLException e) {
         System.out.println(e);
      }
   }

   public static void displayStudents(Connection con, String tableName) throws SQLException
{
      String query = "select * from"+tableName+"order by name ASC";
      try (PreparedStatement pstmt=con.prepareStatement(query);
          ResultSet rs=pstmt.executeQuery()) {
         while (rs.next()) {
            System.out.println(rs.getInt("rollno")+" "+rs.getString("name")+"
"+rs.getInt("per")+" "+rs.getString("email"));
         }
      }
   }
}
```

Output:

```
Students in ascendingby name:
101 Ajay 85 ajay@gmail.com
105 Jayanth 99 jayanth@gmail.com
110 Rahul 75 rahul@example.com
```

---

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

Program:

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;

public class Percentage {
   public static void main(String[] args) {
      String url="jdbc:mysql://localhost:3306/java";
      String user="root";
      String password="root";
      String tableName="Student";

      try (Connection con=DriverManager.getConnection(url, user, password)) {
         System.out.println("Students with percentage greater75");
         displayStudents(con, tableName);
      } catch (SQLException e) {
         System.out.println(e);
      }
   }

   public static void displayStudents(Connection con, String tableName) throws SQLException
{
      String q="select * from"+tableName+"where per>75";
      try (PreparedStatement pstmt=con.prepareStatement(q);
          ResultSet rs=pstmt.executeQuery()) {
         while (rs.next()) {
            System.out.println(rs.getInt("rollno")+"
"+rs.getString("name")+rs.getInt("per")+rs.getString("email"));
         }
      }
   }
}
```

OutPut:

```
Students with percentage greater75
105 Jayanth 99 jayanth@gmail.com
101 Ajay 85 ajay@gmail.com
```

---

11. Use PreparedStatement to insert multiple student records into the database.

Program;

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.SQLException;

public class InsertMultiple {
   public static void main(String[] args) {
      String url="jdbc:mysql://localhost:3306/java";
      String user="root";
      String password="root";
      String tableN="Student";

      try (Connection con=DriverManager.getConnection(url,user,password)) {
         insertStudents(con,tableN);
      } catch (SQLException e) {
         System.out.println(e);
      }
   }

   public static void insertStudents(Connection con, String tableN) throws SQLException {
      String query="insert into"+tableN+"values(?, ?, ?, ?)";
      try (PreparedStatement pstmt=con.prepareStatement(query)) {
         con.setAutoCommit(false);
         pstmt.setInt(1, 106);
         pstmt.setString(2, "R");
         pstmt.setInt(3, 90);
         pstmt.setString(4, "r@gmail.com");
         pstmt.addBatch();

         pstmt.setInt(1, 107);
         pstmt.setString(2, "ravi");
         pstmt.setInt(3, 85);
         pstmt.setString(4, "ravi@gmail.com");
         pstmt.addBatch();

         pstmt.setInt(1, 108);
         pstmt.setString(2, "Ram");
         pstmt.setInt(3, 95);
         pstmt.setString(4, "ram@gmail.com");
         pstmt.addBatch();
```

```
            pstmt.executeBatch();
            con.commit();
            System.out.println("Multiplerecords inserted");
        } catch (SQLException e) {
            con.rollback();
            throw e;
        }
    }
}
```

Output:

```
Multiplerecords inserted
```

---

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

Program:

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.SQLException;

public class CommitRollback {
    public static void main(String[] args) {
        String url="jdbc:mysql://localhost:3306/java";
        String user="root";
        String password="root";
        String tableName="Student";

        try (Connection con=DriverManager.getConnection(url, user, password)) {
            con.setAutoCommit(false);

            try {
                insertStudent(con, tableName,109,"R", 90,"r@gmail.com");
                insertStudent(con,tableName, 110,"Ramu",95,"riya@gmail.com");
                System.out.println("Transaction committed successfully.");
            } catch (SQLException e) {
                con.rollback();
                System.out.println("error"+e.getMessage());
            }
        } catch (SQLException e) {
            System.out.println(e);
        }
    }

    public static void insertStudent(Connection con, String tableName, int rollno, String name,
int per, String email) throws SQLException {
        String query = "insert into"+tableName+"values(?, ?, ?, ?)";
        try (PreparedStatement pstmt=con.prepareStatement(query)) {
            pstmt.setInt(1, rollno);
            pstmt.setString(2, name);
            pstmt.setInt(3, per);
            pstmt.setString(4, email);
            pstmt.executeUpdate();
        }
    }
}
```

---

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

Program:

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;

public class HandleExceptions {
    public static void main(String[] args) {
        String url="jdbc:mysql://localhost:3306/java";
        String user="root";
        String password="root";
        String tableName="Student";
        int rollno = 101;
        try (Connection con = DriverManager.getConnection(url, user, password)) {
            displayStudent(con, tableName, rollno);
        } catch (SQLException e) {
            handleSQLException(e);
        } catch (Exception e) {
            System.out.println("error"+e.getMessage());
        }
    }
    public static void displayStudent(Connection con, String tableName, int rollno) throws
SQLException {
        String query="select * from"+tableName+"where rollno = ?";
        try (PreparedStatement pstmt=con.prepareStatement(query)) {
            pstmt.setInt(1, rollno);
            try (ResultSet rs = pstmt.executeQuery()) {
                if (rs.next()) {
                    System.out.println("Student found:");
                    System.out.println("Roll No: "+rs.getInt("rollno"));
                    System.out.println("Name"+rs.getString("name"));
                    System.out.println("Percentage"+rs.getInt("per"));
                    System.out.println("Email"+rs.getString("email"));
                } else {
                    System.out.println("not found"+rollno);
                }
            }
        }
    }

    public static void handleSQLException(SQLException e) {
        System.out.println("Exception occurred:");

    }
}
```

---

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

Program:

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.Scanner;

public class VerifyCredentials {
    public static void main(String[] args) {
        String url="jdbc:mysql://localhost:3306/java";
        String user="root";
        String password="root";
        String tableName="Users";

        try (Connection con=DriverManager.getConnection(url,user,password)) {
            Scanner scanner=new Scanner(System.in);
            System.out.print("Enter username");
            String username=scanner.nextLine();
            System.out.print("Enter password");
            String pwd=scanner.nextLine();

            if (verifyCredentials(con, tableName, username, pwd)) {
                System.out.println("login successful!");
            } else {
                System.out.println("Invalid");
            }
        } catch (SQLException e) {
            System.out.println(e);
        }
    }

    public static boolean verifyCredentials(Connection con, String tableName, String
username, String password) throws SQLException {
        String query = "select * from"+tableName + "where username=? and password=?";
        try (PreparedStatement pstmt=con.prepareStatement(query)) {
            pstmt.setString(1, username);
            pstmt.setString(2, password);
            try (ResultSet rs = pstmt.executeQuery()) {
                return rs.next();
            }
        }
    }
}
```

---

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

Program

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.Scanner;
```

```
public class Application {
    public static void main(String[] args) {
        String url="jdbc:mysql://localhost:3306/java";
        String user="root";
        String password="root";
        String tableName="Student";

        try (Connection con=DriverManager.getConnection(url, user, password);
             Scanner scanner=new Scanner(System.in)) {

            while (true) {
                System.out.println("Menu");
                System.out.println("Insert Student");
                System.out.println("Search Student");
                System.out.println("Update Student");
                System.out.println("Exit");
                System.out.print("Choose an option");
                int option = scanner.nextInt();
                scanner.nextLine();

                switch (option) {
                    case 1:
                        insertStudent(con,tableName,scanner);
                        break;
                    case 2:
                        searchStudent(con,tableName,scanner);
                        break;
                    case 3:
                        updateStudent(con,tableName,scanner);
                        break;
                    case 4:
                        System.out.println("Exiting");
                        return;
                    default:
                        System.out.println("invalid option.");
                }
            }
        } catch (SQLException e) {
            System.out.println(e);
        }
    }

    public static void insertStudent(Connection con, String tableName, Scanner scanner)
throws SQLException {
        System.out.print("Enter rollno");
        int rollno = scanner.nextInt();
        scanner.nextLine();
        System.out.print("Enter name");
        String name = scanner.nextLine();
        System.out.print("Enter percentage");
        int per = scanner.nextInt();
        scanner.nextLine();
        System.out.print("Enter email");
        String email = scanner.nextLine();

        String query="insert into"+tableName +"values(?, ?, ?, ?)";
        try (PreparedStatement pstmt = con.prepareStatement(query)) {
            pstmt.setInt(1, rollno);
            pstmt.setString(2, name);
            pstmt.setInt(3, per);
            pstmt.setString(4, email);
            pstmt.executeUpdate();
            System.out.println("Student inserted");
        }
    }

    public static void searchStudent(Connection con, String tableName, Scanner scanner)
throws SQLException {
        System.out.print("Enter rollno to search: ");
        int rollno = scanner.nextInt();
        scanner.nextLine();

        String query = "select * from"+tableName+"where rollno = ?";
        try (PreparedStatement pstmt = con.prepareStatement(query)) {
            pstmt.setInt(1, rollno);
            try (ResultSet rs = pstmt.executeQuery()) {
                if (rs.next()) {
                    System.out.println("Student found");
                    System.out.println("roll no"+rs.getInt("rollno"));
                    System.out.println("Name"+rs.getString("name"));
                    System.out.println("Percentage"+rs.getInt("per"));
                System.out.println("Email"+rs.getString("email"));
            } else {
                System.out.println("Student not found.");
            }
        }
    }
}

public static void updateStudent(Connection con, String tableName, Scanner scanner)
throws SQLException {
        System.out.print("enter rollno to update");
        int rollno = scanner.nextInt();
        scanner.nextLine();
        System.out.print("enter new name");
        String name = scanner.nextLine();
        System.out.print("enter new percentage");
        int per = scanner.nextInt();
        scanner.nextLine();
        System.out.print("Enter new email");
        String email = scanner.nextLine();

        String query="update"+tableName+"set name=?, per=?, email=? where rollno=?";
        try (PreparedStatement pstmt =con.prepareStatement(query)) {
            pstmt.setString(1,name);
            pstmt.setInt(2, per);
            pstmt.setString(3,email);
            pstmt.setInt(4, rollno);
            pstmt.executeUpdate();
            System.out.println("Student updated");
        }
    }
}
```

---

16. 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

Program:

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.Scanner;

public class LibraryManagement {
    public static void main(String[] args) {
        String url="jdbc:mysql://localhost:3306/java";
        String user="root";
        String password="root";

        try (Connection con = DriverManager.getConnection(url, user, password);
             Scanner scanner = new Scanner(System.in)) {

            while (true) {
                System.out.println("Menu");
                System.out.println("Add a book");
                System.out.println("View all books");
                System.out.println("Issue a book to a member");
                System.out.println("Return a book");
                System.out.println("Exit");
                System.out.print("Choose an option: ");
                int option = scanner.nextInt();
                scanner.nextLine();

                switch (option) {
                    case 1:
                        addBook(con,scanner);
                        break;
                    case 2:
                        viewAllBooks(con);
                        break;
                    case 3:
                        issueBook(con,scanner);
                        break;
                    case 4:
                        returnBook(con, scanner);
                        break;
                    case 5:
                        System.out.println("Exiting");
                        return;
                    default:
                        System.out.println("Invalid option");
                }
            }
        } catch (SQLException e) {
            System.out.println(e);
        }
    }

    public static void addBook(Connection con, Scanner scanner) throws SQLException {
        System.out.print("Enter book ID: ");
        int bookId = scanner.nextInt();
        scanner.nextLine(); // Consume newline left-over
        System.out.print("Enter book title: ");
        String title = scanner.nextLine();
        System.out.print("Enter book author: ");
        String author = scanner.nextLine();
        System.out.print("Enter publication year: ");
        int publicationYear = scanner.nextInt();
        scanner.nextLine(); // Consume newline left-over

        String query = "INSERT INTO Books (book_id, title, author, publication_year) VALUES (?,
?, ?, ?)";
        try (PreparedStatement pstmt = con.prepareStatement(query)) {
            pstmt.setInt(1, bookId);
            pstmt.setString(2, title);
            pstmt.setString(3, author);
            pstmt.setInt(4, publicationYear);
            pstmt.executeUpdate();
            System.out.println("Book added successfully.");
        }
    }

    public static void viewAllBooks(Connection con) throws SQLException {
        String query = "SELECT * FROM Books";
        try (PreparedStatement pstmt = con.prepareStatement(query);
             ResultSet rs = pstmt.executeQuery()) {
            while (rs.next()) {
                System.out.println("Book ID: " + rs.getInt("book_id"));
                System.out.println("Title: " + rs.getString("title"));
            System.out.println("Author: " + rs.getString("author"));
            System.out.println("Publication Year: " + rs.getInt("publication_year"));
            System.out.println("Status: " + rs.getString("status"));
            System.out.println();
          }
        }
      }

      public static void issueBook(Connection con, Scanner scanner) throws SQLException {
        System.out.print("Enter book ID: ");
        int bookId = scanner.nextInt();
        scanner.nextLine(); // Consume newline left-over
        System.out.print("Enter member ID: ");
        int memberId = scanner.nextInt();
        scanner.nextLine(); // Consume newline left-over

        String query = "SELECT * FROM Books WHERE book_id = ? AND status = 'Available'";
        try (PreparedStatement pstmt = con.prepareStatement(query)) {
          pstmt.setInt(1, bookId);
          try (ResultSet rs = pstmt.executeQuery()) {
            if (rs.next()) {
              String updateQuery = "UPDATE Books SET status = 'Issued' WHERE book_id = ?";
              try (PreparedStatement updatePstmt = con.prepareStatement(updateQuery)) {
                updatePstmt.setInt(1, bookId);
                updatePstmt.executeUpdate();
              }

              String insertQuery = "INSERT INTO Borrowings (book_id, member_id, issue_date)
    VALUES (?, ?, CURDATE())";
              try (PreparedStatement insertPstmt = con.prepareStatement(insertQuery)) {
                insertPstmt.setInt(1, bookId);
                insertPstmt.setInt(2, memberId);
                insertPstmt.executeUpdate();
              }
              System.out.println("Book issued successfully.");
            } else {
              System.out.println("Book is not available.");
            }
          }
        }
      }
    public static void returnBook(Connection con, Scanner scanner) throws SQLException {
        System.out.print("Enter book ID: ");
        int bookId = scanner.nextInt();
        scanner.nextLine(); // Consume newline left-over

        String query = "UPDATE Books SET status = 'Available' WHERE book_id = ?";
        try (PreparedStatement pstmt = con.prepareStatement(query)) {
            pstmt.setInt(1, bookId);
            pstmt.executeUpdate();
        }

        String updateQuery = "UPDATE Borrowings SET return_date = CURDATE() WHERE
book_id = ? AND return_date IS NULL";
        try (PreparedStatement updatePstmt = con.prepareStatement(updateQuery)) {
            updatePstmt.setInt(1, bookId);
            updatePstmt.executeUpdate();
        }
        System.out.println("Book returned successfully.");
        }
        }
```

Output:

```
Book added successfully
Book ID: 101
Title: Java Basics
Author: John Doe
Publication Year: 2020
Status: Available
Book issued successfully.
```

---

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

· Register new patient

· Assign doctor

· Generate billing

Program

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.Scanner;

public class HospitalManagement{
    public static void main(String[] args) {
        String url="jdbc:mysql://localhost:3306/java";
        String user="root";
        String password="root";

        try (Connection con=DriverManager.getConnection(url, user, password);
             Scanner scanner=new Scanner(System.in)) {

            while (true) {
                System.out.println("Menu");
                System.out.println("Register new patient");
                System.out.println("Assign doctor");
                System.out.println("Generate billing");
                System.out.println("Exit");
                System.out.print("Choose an option: ");
                int option = scanner.nextInt();
                scanner.nextLine();

                switch (option) {
                    case 1:
                        registerP(con, scanner);
                        break;
                    case 2:
                        assignD(con, scanner);
                        break;
                    case 3:
                        generateB(con, scanner);
                        break;
                    case 4:
                        System.out.println("Exiting");
                        return;
                    default:
                        System.out.println("Invalid option");
                }
            }
        } catch (SQLException e) {
            System.out.println(e);
        }
    }

    public static void registerP(Connection con, Scanner scanner) throws SQLException {
        System.out.print("Enter patient id");
        int patientId = scanner.nextInt();
        scanner.nextLine();
        System.out.print("Enter patient name");
        String name = scanner.nextLine();
        System.out.print("Enter patient age");
        int age = scanner.nextInt();
        scanner.nextLine();
        System.out.print("Enter patient contact number");
        String contactNumber = scanner.nextLine();

        String query = "insert into patients(patient_id, name, age, contact_number) values(?, ?, 
?, ?)";
        try (PreparedStatement pstmt = con.prepareStatement(query)) {
            pstmt.setInt(1, patientId);
            pstmt.setString(2, name);
            pstmt.setInt(3, age);
            pstmt.setString(4, contactNumber);
            pstmt.executeUpdate();
            System.out.println("Patient registered");
        }
    }

    public static void assignD(Connection con, Scanner scanner) throws SQLException {
        System.out.print("Enter patient id");
        int patientId = scanner.nextInt();
        scanner.nextLine();
        System.out.print("Enter doctor ID: ");
        int doctorId = scanner.nextInt();
        scanner.nextLine();

        String query = "insert into Patient_Doctor(patient_id, doctor_id) values(?, ?)";
        try (PreparedStatement pstmt =con.prepareStatement(query)) {
            pstmt.setInt(1, patientId);
            pstmt.setInt(2, doctorId);
            pstmt.executeUpdate();
            System.out.println("Doctor assigned");
        }
    }

    public static void generateB(Connection con, Scanner scanner) throws SQLException {
        System.out.print("Enter patient id");
        int patientId = scanner.nextInt();
        scanner.nextLine();
        System.out.print("Enter bill amount: ");
        double amount = scanner.nextDouble();
        scanner.nextLine();
        System.out.print("Enter payment status: ");
        String paymentStatus = scanner.nextLine();

        String query = "insert into Billing (patient_id, amount, payment_status) values(?, ?, ?)";
        try (PreparedStatement pstmt = con.prepareStatement(query)) {
            pstmt.setInt(1, patientId);
            pstmt.setDouble(2, amount);
            pstmt.setString(3, paymentStatus);
            pstmt.executeUpdate();
            System.out.println("Bill generated ");
        }
    }
}
```

Output:

```
Patient registered
Doctor assigned
Bill generated
Exiting
```

---

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

Program

```
import java.io.FileWriter;
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 JDBC {
    public static void main(String[] args) {
        String url ="jdbc:mysql://localhost:3306/java";
        String user="root";
        String password="root";
        String tableName="Student";
        String outputFile="student_report.csv";

        try (Connection con=DriverManager.getConnection(url, user, password)) {
            generateReport(con,tableName,outputFile);
        } catch (SQLException e) {
            System.out.println(e);
        }
    }

    public static void generateReport(Connection con, String tableName, String outputFile)
throws SQLException {
        String query ="select * from"+tableName;
        try (Statement stmt=con.createStatement();
             ResultSet rs =stmt.executeQuery(query);
             FileWriter writer=new FileWriter(outputFile)) {

            int columnCount=rs.getMetaData().getColumnCount();
            for (int i = 1; i <= columnCount; i++) {
                writer.write(rs.getMetaData().getColumnName(i));
                if (i < columnCount) {
                    writer.write(",");
                }
            }
            writer.write("\n");
            while (rs.next()) {
                for (int i=1;i<=columnCount; i++) {
                    writer.write(rs.getString(i));
                    if (i <columnCount) {
                        writer.write(",");
                    }
                }
                writer.write("\n");
            }

            System.out.println("Report generated");
        } catch (IOException e) {
            System.out.println("Error"+e.getMessage());
    }
}}
```