
LAB 8

Question:

- **Implement JDBC Concept in your domain.**
- **Minimum two tables based on your domain you need to create it.**
- **Minimum 10 records add it in your table.**

CRUD Operation performs in your domain based example.

Code:

```
import java.sql.*;

public class App {
    public static void main(String[] args) {
        try {
            // Registering the driver
            Class.forName("com.mysql.cj.jdbc.Driver");

            // Creating a connection
            String url = "jdbc:mysql://localhost:3306/bookstore";
            String username = "root";
            String password = "";
            Connection con = DriverManager.getConnection(url,
username, password);

            // Creating a table for books
            String query = "CREATE TABLE books (id INT AUTO_INCREMENT
PRIMARY KEY, title VARCHAR(255), author VARCHAR(255), price INT)";
            Statement stmt = con.createStatement();
            stmt.executeUpdate(query);

            // Creating a table for customers
            query = "CREATE TABLE customers (id INT AUTO_INCREMENT
PRIMARY KEY, name VARCHAR(255), address VARCHAR(255), bill INT)";
            stmt.executeUpdate(query);

            // Inserting records in books
```

```
            query = "INSERT INTO books (title, author, price) VALUES
('The Alchemist', 'Paulo Coelho', 170)";
            stmt.executeUpdate(query);
            query = "INSERT INTO books (title, author, price) VALUES
('The Da Vinci Code', 'Dan Brown', 120)";
            stmt.executeUpdate(query);
            query = "INSERT INTO books (title, author, price) VALUES
(' Catcher ', 'J.D', 67)";
            stmt.executeUpdate(query);
            query = "INSERT INTO books (title, author, price) VALUES
('T', 'J.D. Salinger', 900)";
            stmt.executeUpdate(query);
            query = "INSERT INTO books (title, author, price) VALUES
('One Hundred', 'Gabriel Garcia Marquez', 305)";
            stmt.executeUpdate(query);
            query = "INSERT INTO books (title, author, price) VALUES
('The Hobbit', 'J.R.R. Tolkien', 220)";
            stmt.executeUpdate(query);
            query = "INSERT INTO books (title, author, price) VALUES
('Lord of the Rings', 'J.R.R. Tolkien', 459)";
            stmt.executeUpdate(query);
            query = "INSERT INTO books (title, author, price) VALUES
('The Odyssey', 'Homer', 145)";
            stmt.executeUpdate(query);
            query = "INSERT INTO books (title, author, price) VALUES
('Harry Potter ', 'J.K. Rowling', 875)";
            stmt.executeUpdate(query);
            query = "INSERT INTO books (title, author, price) VALUES
('Catcher in the Rye', 'J.D. Salinger', 900)";
            stmt.executeUpdate(query);

            // Inserting records in customers
            query = "INSERT INTO customers (name, address, bill)
VALUES ('Ravi', 'Patna', 100)";
            stmt.executeUpdate(query);
            query = "INSERT INTO customers (name, address, bill)
VALUES ('Vansh', 'GAya', 2650)";
            stmt.executeUpdate(query);
```

```
            query = "INSERT INTO customers (name, address, bill)
VALUES ('Amit', 'Lalitpur', 345)";
            stmt.executeUpdate(query);

            // Updating records
            query = "UPDATE books SET price = 110 WHERE title = 'The
Alchemist'";
            stmt.executeUpdate(query);

            // Deleting records
            query = "DELETE FROM books WHERE title = 'T'";
            stmt.executeUpdate(query);

            // Retrieving records from books
            query = "SELECT * FROM books";
            ResultSet rs = stmt.executeQuery(query);
            while (rs.next()) {
                System.out.println(rs.getInt("id") + " " +
rs.getString("title") + " " + rs.getString("author") + " "
                        + rs.getFloat("price"));
            }

            // Closing the connection
            con.close();
        } catch (Exception e) {
            System.out.println(e);
        }
    }
}
```

Screenshot:

```
PS D:\Christ MCA\Sem - 2\Java\Lab\Lab8>  & 'C:'
pp'
1 The Alchemist Paulo Coelho 110.0
2 The Da Vinci Code Dan Brown 120.0
3  Catcher  J.D 67.0
5 One Hundred Gabriel Garcia Marquez 305.0
6 The Hobbit J.R.R. Tolkien 220.0
7 Lord of the Rings J.R.R. Tolkien 459.0
8 The Odyssey Homer 145.0
9 Harry Potter  J.K. Rowling 875.0
10 Catcher in the Rye J.D. Salinger 900.0
PS D:\Christ MCA\Sem - 2\Java\Lab\Lab8>
```

DB Tables

	Table	Action						Rows	Type	Collation	Size	Overhead
☐	books	☆	Browse	Structure	Search	Insert	Empty	Drop	9	InnoDB	utf8mb4_general_ci	16.0 KiB	-
☐	customers	☆	Browse	Structure	Search	Insert	Empty	Drop	3	InnoDB	utf8mb4_general_ci	16.0 KiB	-
	2 tables	Sum							12	InnoDB	utf8mb4_general_ci	32.0 KiB	0 B

Check all With selected:

Books Table

Extra options

←T→				id	title	author	price
☐	Edit	Copy	Delete	1	The Alchemist	Paulo Coelho	110
☐	Edit	Copy	Delete	2	The Da Vinci Code	Dan Brown	120
☐	Edit	Copy	Delete	3	Catcher	J.D	67
☐	Edit	Copy	Delete	5	One Hundred	Gabriel Garcia Marquez	305
☐	Edit	Copy	Delete	6	The Hobbit	J.R.R. Tolkien	220
☐	Edit	Copy	Delete	7	Lord of the Rings	J.R.R. Tolkien	459
☐	Edit	Copy	Delete	8	The Odyssey	Homer	145
☐	Edit	Copy	Delete	9	Harry Potter	J.K. Rowling	875
☐	Edit	Copy	Delete	10	Catcher in the Rye	J.D. Salinger	900

Check all With selected: Edit Copy Delete Export

Customer Table

	id	name	address	bill
Edit Copy Delete	1	Ravi	Patna	100
Edit Copy Delete	2	Vansh	GAya	2650
Edit Copy Delete	3	Amit	Lalitpur	345