

Modern Application Development (Java Spring Boot)

Assignment-3

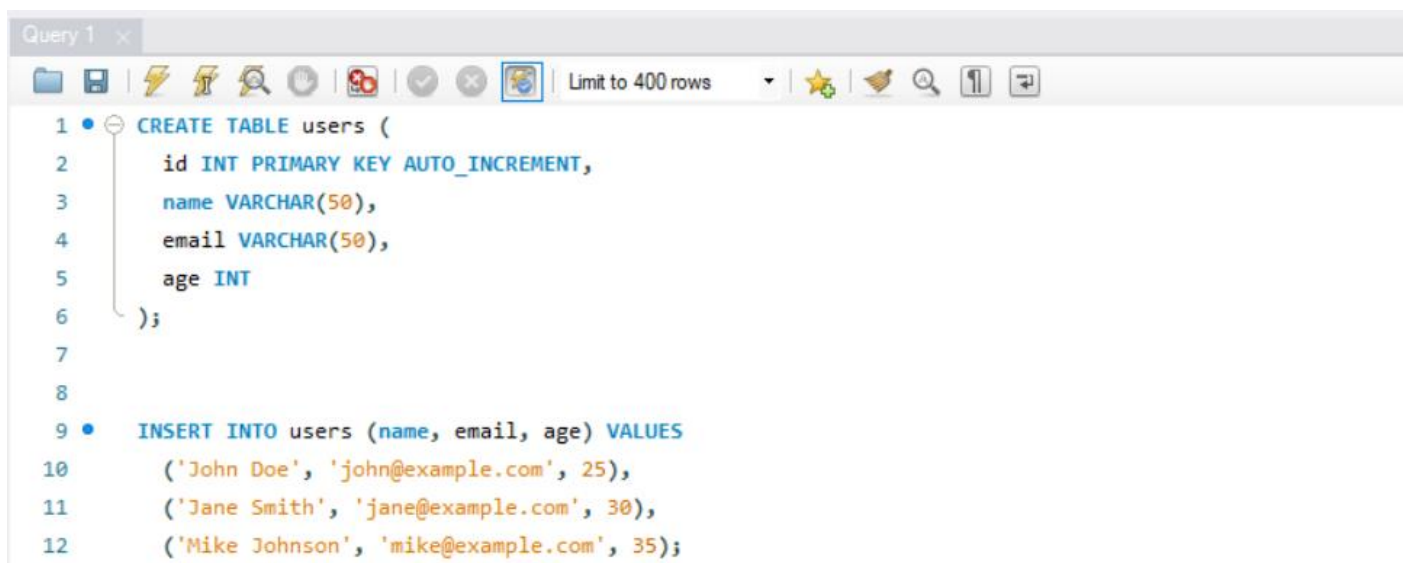
REG.NO: 20BCE2752

NAME: AATHIRAINATHAN P

1) Implement JDBC connectivity using java.

Step 1: Create a database in MySql for JDBC connectivity:

```
CREATE TABLE users (  
    id INT PRIMARY KEY AUTO_INCREMENT,  
    name VARCHAR(50),  
    email VARCHAR(50),  
    age INT  
);  
  
INSERT INTO users (name, email, age) VALUES  
    ('John Doe', 'john@example.com', 25),  
    ('Jane Smith', 'jane@example.com', 30),  
    ('Mike Johnson', 'mike@example.com', 35);
```



The screenshot shows a MySQL query editor window titled 'Query 1'. The toolbar includes icons for file operations, execution, and search. The SQL code is as follows:

```
1 • CREATE TABLE users (  
2     id INT PRIMARY KEY AUTO_INCREMENT,  
3     name VARCHAR(50),  
4     email VARCHAR(50),  
5     age INT  
6 );  
7  
8  
9 • INSERT INTO users (name, email, age) VALUES  
10     ('John Doe', 'john@example.com', 25),  
11     ('Jane Smith', 'jane@example.com', 30),  
12     ('Mike Johnson', 'mike@example.com', 35);
```

Result Grid	Filter Rows:	Edit:	Export/Import:	Wrap Cell Content:
id	name	email	age	
1	John Doe	john@example.com	25	
2	Jane Smith	jane@example.com	30	
3	Mike Johnson	mike@example.com	35	
NULL	NULL	NULL	NULL	

users 1 x

Step 2: Establish JDBC connectivity using Netbeans IDE and execute some queries:

```
package com.sample.jdbcex;
```

```
import java.sql.*;
```

```
public class NewClass {
```

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

```
        // JDBC connection parameters
```

```
        String url = "jdbc:mysql://localhost:3306/users";
```

```
        String username = "root";
```

```
        String password = "aathirai";
```

```
        // Step 1: Establish the connection
```

```
        try (Connection conn = DriverManager.getConnection(url, username, password)) {
```

```
            System.out.println("Connected to the database.");
```

```
            // Step 2: Execute SQL queries
```

```
            Statement statement = conn.createStatement();
```

```
            // Insert a new user
```

```
            String insertQuery = "INSERT INTO users (name, age, email) VALUES ('John Doe', 25, 'john@example.com')";
```

```
            int rowsInserted = statement.executeUpdate(insertQuery);
```

```
            System.out.println(rowsInserted + " row(s) inserted.");
```

```
            // Select all users
```

```
            String selectQuery = "SELECT * FROM users";
```

```

ResultSet resultSet = statement.executeQuery(selectQuery);
while (resultSet.next()) {
    int userId = resultSet.getInt("id");
    String name = resultSet.getString("name");
    int age = resultSet.getInt("age");
    String email = resultSet.getString("email");
    System.out.println("User ID: " + userId + ", Name: " + name + ", Age: " + age + ", Email: " +
email);
}

// Update a user's age
String updateQuery = "UPDATE users SET age = 30 WHERE name = 'John Doe'";
int rowsUpdated = statement.executeUpdate(updateQuery);
System.out.println(rowsUpdated + " row(s) updated.");

// Delete a user
String deleteQuery = "DELETE FROM users WHERE name = 'John Doe'";
int rowsDeleted = statement.executeUpdate(deleteQuery);
System.out.println(rowsDeleted + " row(s) deleted.");
} catch (SQLException e) {
    System.out.println("Connection error: " + e.getMessage());
}
}
}

```

```
Start Page x NewClass.java x
Source History
1 package com.sample.jdbcex;
2 import java.sql.*;
3
4 public class NewClass {
5     public static void main(String[] args) {
6         // JDBC connection parameters
7         String url = "jdbc:mysql://localhost:3306/users";
8         String username = "root";
9         String password = "aathirai";
10
11         // Step 1: Establish the connection
12         try (Connection conn = DriverManager.getConnection(url, username, password)) {
13             System.out.println("Connected to the database.");
14
15             // Step 2: Execute SQL queries
16             Statement statement = conn.createStatement();
17
18             // Insert a new user
19             String insertQuery = "INSERT INTO users (name, age, email) VALUES ('John Doe', 25, 'john@example.com')";
20             int rowsInserted = statement.executeUpdate(insertQuery);
21             System.out.println(rowsInserted + " row(s) inserted.");
22
23             // Select all users
24             String selectQuery = "SELECT * FROM users";
25             ResultSet resultSet = statement.executeQuery(selectQuery);
26             while (resultSet.next()) {
27                 int userId = resultSet.getInt("id");
28                 String name = resultSet.getString("name");
29                 int age = resultSet.getInt("age");
30                 String email = resultSet.getString("email");
31                 System.out.println("User ID: " + userId + ", Name: " + name + ", Age: " + age + ", Email: " + email);
32             }
33     }
}
```

```
Start Page x NewClass.java x
Source History
26 while (resultSet.next()) {
27     int userId = resultSet.getInt("id");
28     String name = resultSet.getString("name");
29     int age = resultSet.getInt("age");
30     String email = resultSet.getString("email");
31     System.out.println("User ID: " + userId + ", Name: " + name + ", Age: " + age + ", Email: " + email);
32 }
33
34 // Update a user's age
35 String updateQuery = "UPDATE users SET age = 30 WHERE name = 'John Doe'";
36 int rowsUpdated = statement.executeUpdate(updateQuery);
37 System.out.println(rowsUpdated + " row(s) updated.");
38
39 // Delete a user
40 String deleteQuery = "DELETE FROM users WHERE name = 'John Doe'";
41 int rowsDeleted = statement.executeUpdate(deleteQuery);
42 System.out.println(rowsDeleted + " row(s) deleted.");
43 } catch (SQLException e) {
44     System.out.println("Connection error: " + e.getMessage());
45 }
46 }
47 }
48 }
```

Output:



```
Output - Run (jdbcex) x
cd C:\Users\HP\Documents\NetBeansProjects\jdbcex; "JAVA_HOME=C:\\Program Files\\Java\\jdk-17.0.1" cmd /c "%C:\
Running NetBeans Compile On Save execution. Phase execution is skipped and output directories of dependency pro
Scanning for projects...

-----< com.sample:jdbcex >-----
Building jdbcex 1.0-SNAPSHOT
-----[ jar ]-----

--- exec-maven-plugin:3.0.0:exec (default-cli) @ jdbcex ---
Connected to the database.
1 row(s) inserted.
User ID: 1, Name: John Doe, Age: 25, Email: john@example.com
User ID: 2, Name: Jane Smith, Age: 30, Email: jane@example.com
User ID: 3, Name: Mike Johnson, Age: 35, Email: mike@example.com
User ID: 4, Name: John Doe, Age: 25, Email: john@example.com
2 row(s) updated.
2 row(s) deleted.

BUILD SUCCESS

Total time: 4.089 s
Finished at: 2023-06-05T10:46:02+05:30
```

20BCE2752

AATHIRAINATHAN P