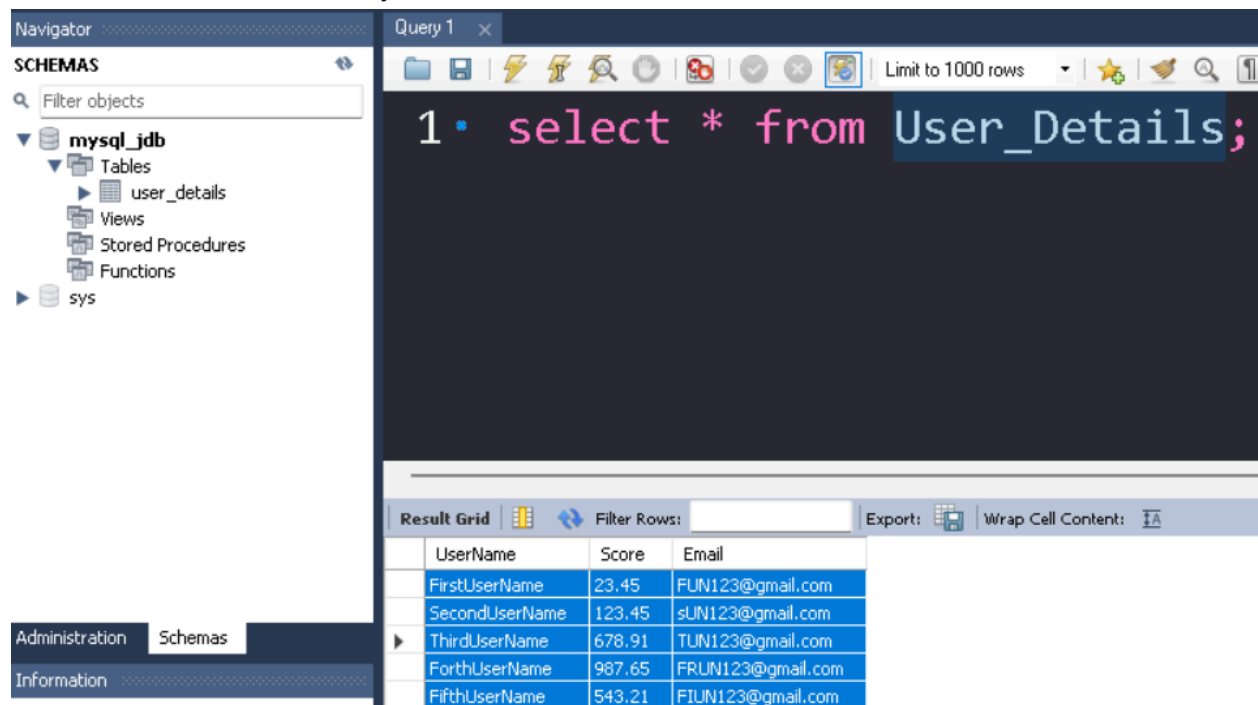


Insert user details, then finally tables be like:



The screenshot shows a database IDE interface. On the left, the 'SCHEMAS' pane displays a tree view for 'mysql\_jdb' containing 'Tables', 'Views', 'Stored Procedures', and 'Functions'. The 'Tables' folder is expanded, showing 'user\_details'. The main query editor, titled 'Query 1', contains the SQL statement: `1. select * from User_Details;`. Below the editor, the 'Result Grid' shows the query results in a table with three columns: 'UserName', 'Score', and 'Email'. The results are as follows:

UserName	Score	Email
FirstUserName	23.45	FUN123@gmail.com
SecondUserName	123.45	sUN123@gmail.com
ThirdUserName	678.91	TUN123@gmail.com
ForthUserName	987.65	FRUN123@gmail.com
FifthUserName	543.21	FIUN123@gmail.com

Initially we have the database like:

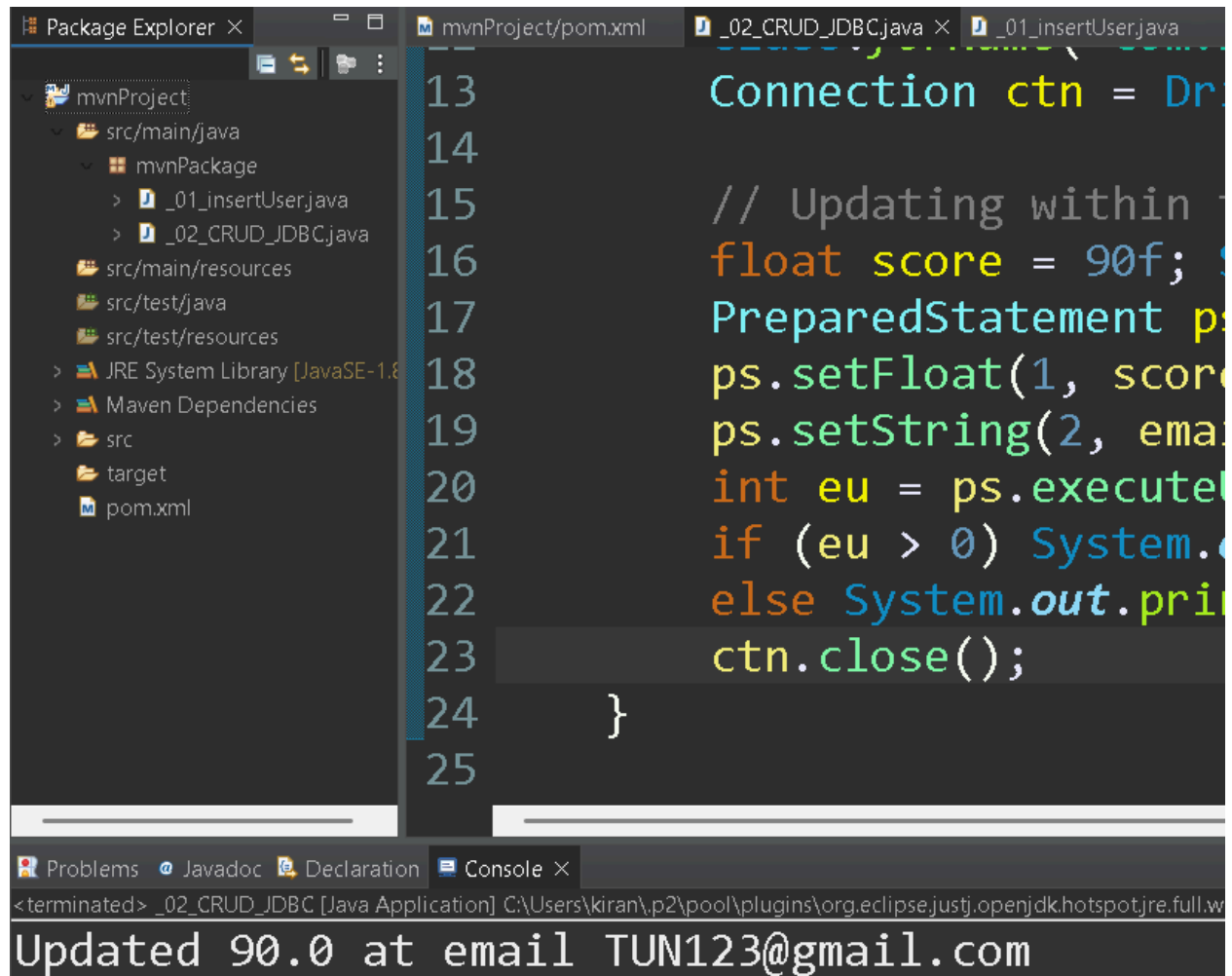
UserName	Score	Email
FirstUserName	23.45	FUN123@gmail.com
SecondUserName	123.45	sUN123@gmail.com
ThirdUserName	678.91	TUN123@gmail.com
ForthUserName	987.65	FRUN123@gmail.com
FifthUserName	543.21	FIUN123@gmail.com

For updating cell in Table:

```
// Updating within table
    float score = 90f; String email =
    "TUN123@gmail.com";
    PreparedStatement ps = ctn.prepareStatement("update
    User_Details set Score=? where Email=?");
    ps.setFloat(1, score);
    ps.setString(2, email);
    int eu = ps.executeUpdate();
    if (eu > 0) System.out.println("Updated " + score +
    " at email " + email);
    else System.out.println("Update at email " + email
    + " got failed.");
```

```
ctn.close();
```

Run the code:



The screenshot shows the Eclipse IDE interface. On the left is the Package Explorer showing the project structure: mvnProject, src/main/java (containing mvnPackage, \_01\_insertUser.java, and \_02\_CRUD\_JDBC.java), src/main/resources, src/test/java, src/test/resources, JRE System Library [JavaSE-1.8], Maven Dependencies, src, target, and pom.xml. The main editor displays the code in \_02\_CRUD\_JDBC.java, with line numbers 13 to 25. The code includes a JDBC connection, an update statement, and a close call. The console at the bottom shows the output: 'Updated 90.0 at email TUN123@gmail.com'.

```
13 Connection ctn = DriverManager.getConnection(url, user, password);
14
15 // Updating within database
16 float score = 90f;
17 PreparedStatement ps = ctn.prepareStatement("UPDATE student SET score = ? WHERE email = ?");
18 ps.setFloat(1, score);
19 ps.setString(2, email);
20 int eu = ps.executeUpdate();
21 if (eu > 0) System.out.println("Updated " + score + " at email " + email);
22 else System.out.println("Not Updated");
23 ctn.close();
24 }
25
```

<terminated> \_02\_CRUD\_JDBC [Java Application] C:\Users\kiran\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.w

Updated 90.0 at email TUN123@gmail.com

Click as shown by arrow:

The screenshot shows the MySQL Workbench interface. On the left, the 'Navigator' pane displays the 'SCHEMAS' tree with 'mysql\_jdb' expanded, showing 'Tables' (including 'user\_details'), 'Views', 'Stored Procedures', and 'Functions'. The main editor window, titled 'Query 1', contains the SQL query: `1. select * from User_Details;`. A red arrow points to the number '1' at the start of the query. Below the editor, the 'Result Grid' shows the query results in a table with columns 'UserName', 'Score', and 'Email'.

UserName	Score	Email
FirstUserName	23.45	FUN123@gmail.com
SecondUserName	123.45	sUN123@gmail.com
ThirdUserName	678.91	TUN123@gmail.com
ForthUserName	987.65	FRUN123@gmail.com
FifthUserName	543.21	FIUN123@gmail.com

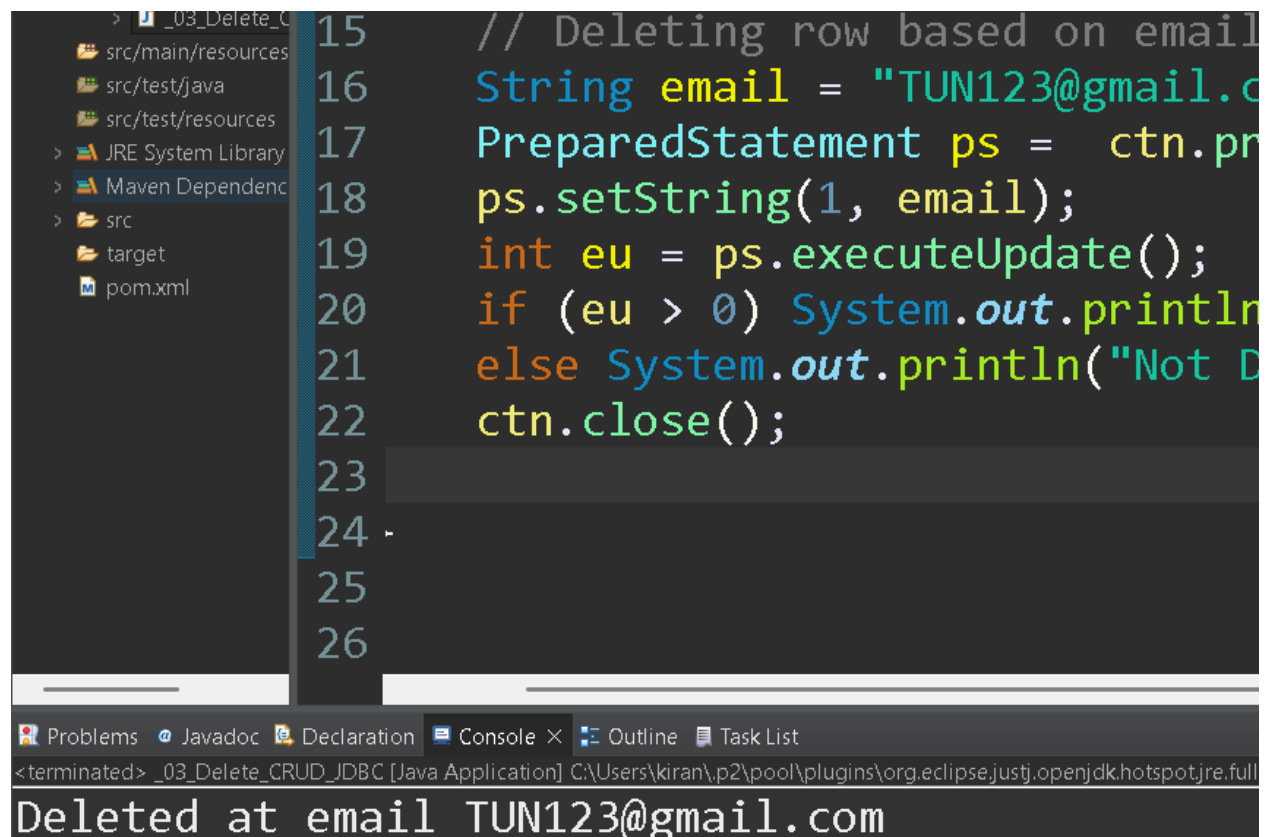
Then the score gets updated as expected without using MySQL workbench:

	UserName	Score	Email
▶	FirstUserName	23.45	FUN123@gmail.com
	SecondUserName	123.45	sUN123@gmail.com
	ThirdUserName	90.00	TUN123@gmail.com
	ForthUserName	987.65	FRUN123@gmail.com
	FifthUserName	543.21	FIUN123@gmail.com

For **Deleting** the row based on email check:

```
// Deleting row based on email
String email = "TUN123@gmail.com";
PreparedStatement ps = ctn.prepareStatement("delete from User_Details where Email=?");
ps.setString(1, email);
int eu = ps.executeUpdate();
if (eu > 0) System.out.println("Deleted at email " + email);
else System.out.println("Not Deleted at email " + email);
ctn.close();
```

Run the code:



The screenshot shows the Eclipse IDE interface. On the left, the Project Explorer displays the file structure of a project named "\_03\_Delete\_CRUD\_JDBC". The main editor area shows a Java file with the following code:

```
15 // Deleting row based on email
16 String email = "TUN123@gmail.com";
17 PreparedStatement ps = ctn.prepareStatement("DELETE FROM users WHERE email = ?");
18 ps.setString(1, email);
19 int eu = ps.executeUpdate();
20 if (eu > 0) System.out.println("Deleted at email " + email);
21 else System.out.println("Not Deleted");
22 ctn.close();
23
24
25
26
```

At the bottom, the Console view shows the output of the application:

```
<terminated> _03_Delete_CRUD_JDBC [Java Application] C:\Users\kiran\AppData\Local\Temp\org.eclipse.justj.openjdk.hotspot.jre.full\
Deleted at email TUN123@gmail.com
```



Query 1 x

Limit to 1000 rows

```
1 select * from User_Details;
```

Result Grid

Filter Rows:

Export:  Wrap Cell Content: 

	UserName	Score	Email
▶	FirstUserName	23.45	FUN123@gmail.com
	SecondUserName	123.45	sUN123@gmail.com
	ThirdUserName	90.00	TUN123@gmail.com
	ForthUserName	987.65	FRUN123@gmail.com
	FifthUserName	543.21	FIUN123@gmail.com

UserName	Score	Email
FirstUserName	23.45	FUN123@gmail.com
SecondUserName	123.45	sUN123@gmail.com
ForthUserName	987.65	FRUN123@gmail.com
FifthUserName	543.21	FIUN123@gmail.com

Similarly, Fetching the data from database:

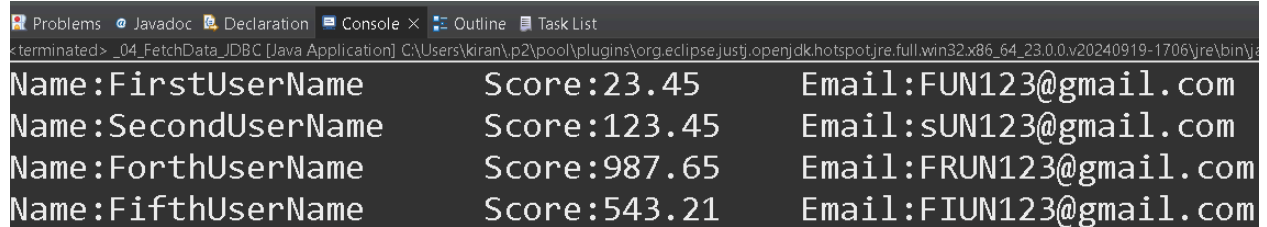
```

Connection ctn = DriverManager.getConnection(url, user, pwd);

PreparedStatement ps = ctn.prepareStatement("select * from User_Details");
ResultSet rs = ps.executeQuery();
while (rs.next()) {
    System.out.print("Name:" + rs.getString("UserName"));
    System.out.print("\tScore:" + rs.getString("Score"));
    System.out.println("\tEmail:" + rs.getString("Email"));
}
ctn.close();

```

Then output looks like:



```

Problems Javadoc Declaration Console × Outline Task List
<terminated> _04_FetchData_JDBC [Java Application] C:\Users\kiran\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_23.0.0.v20240919-1706\jre\bin\java.exe
Name:FirstUserName      Score:23.45      Email:FUN123@gmail.com
Name:SecondUserName     Score:123.45     Email:sUN123@gmail.com
Name:ForthUserName      Score:987.65     Email:FRUN123@gmail.com
Name:FifthUserName      Score:543.21     Email:FIUN123@gmail.com

```