

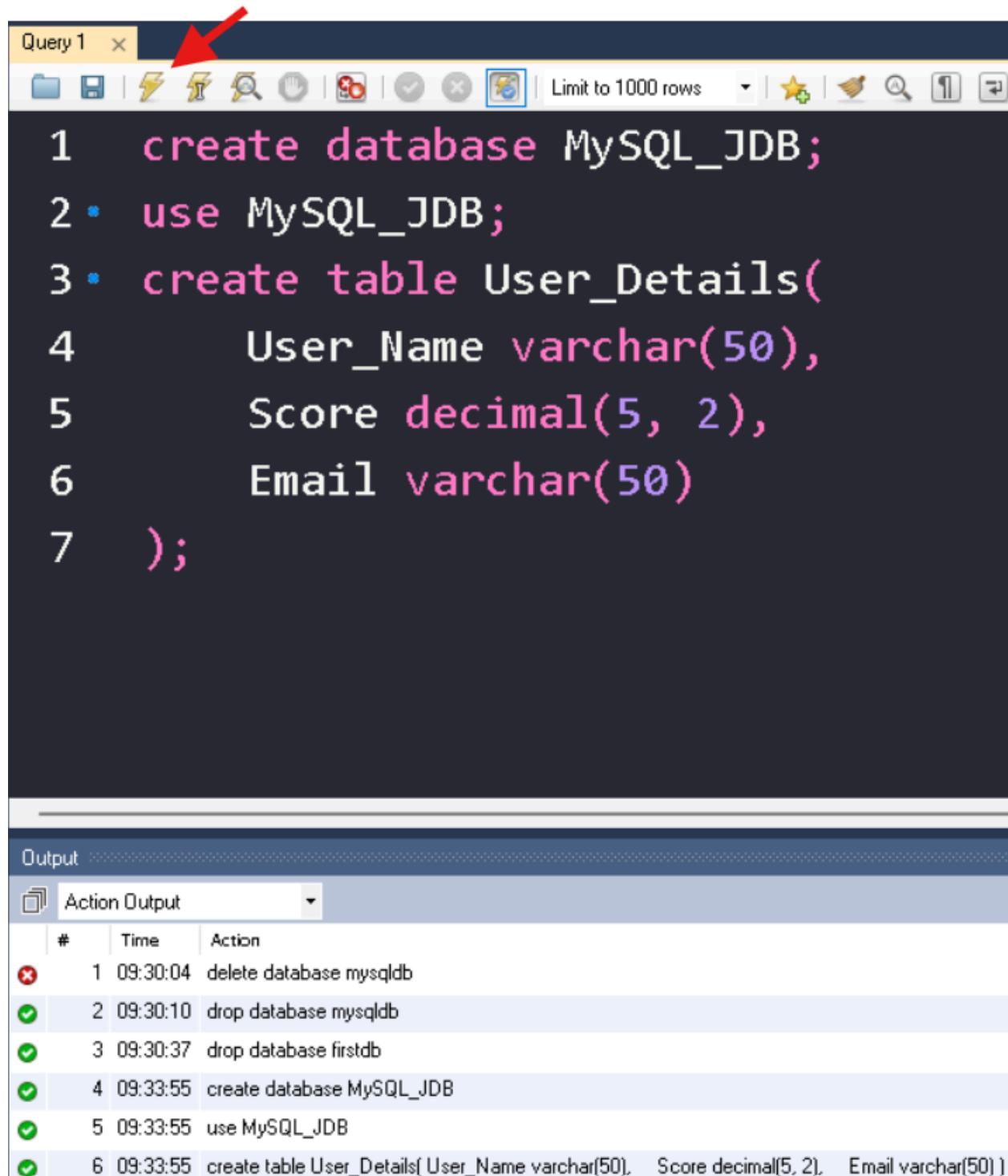
1. Create a Maven project in EclipseIDE
2. Make pom.xml:

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>mvnProject</groupId>
  <artifactId>mvnProject</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <dependencies>
    <!--
https://mvnrepository.com/artifact/com.mysql/mysql-connector-j -->
  <dependency>
    <groupId>com.mysql</groupId>
    <artifactId>mysql-connector-j</artifactId>
    <version>9.5.0</version>
  </dependency>
</dependencies>
</project>
```

Open the MySQL workbench:

Write the below code, then click the arrows as shown:

```
1  -- Creates Database
2 • create database MySQL_JDB;
3  -- Select the database
4 • use MySQL_JDB;
5  -- create table when database is Selected
6 • create table User_Details(
7      User_Name varchar(50),
8      Score decimal(5, 2),
9      Email varchar(50)
10 );
```



The screenshot shows a SQL IDE interface. At the top, a tab labeled "Query 1" is active. A red arrow points to the "Execute" button (a lightning bolt icon) in the toolbar. Below the toolbar, a SQL query is entered in a dark-themed editor:

```
1 create database MySQL_JDB;  
2 use MySQL_JDB;  
3 create table User_Details(  
4     User_Name varchar(50),  
5     Score decimal(5, 2),  
6     Email varchar(50)  
7 );
```

Below the query editor, the "Output" pane is visible, showing the "Action Output" log. The log contains the following entries:

#	Time	Action
1	09:30:04	delete database mysqldb
2	09:30:10	drop database mysqldb
3	09:30:37	drop database firstdb
4	09:33:55	create database MySQL_JDB
5	09:33:55	use MySQL_JDB
6	09:33:55	create table User_Details(User_Name varchar(50), Score decimal(5, 2), Email varchar(50))

Refresh by clicking as below:

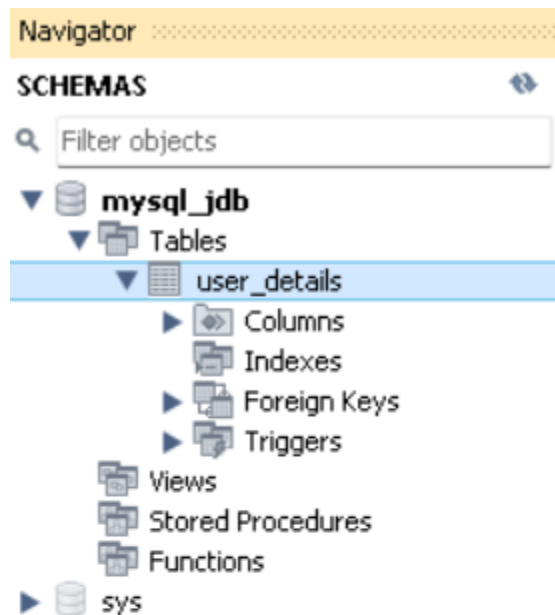
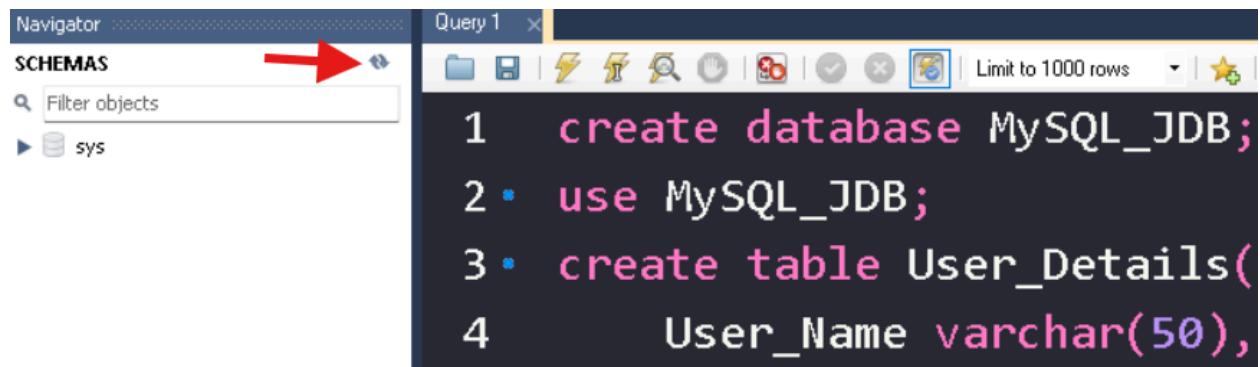


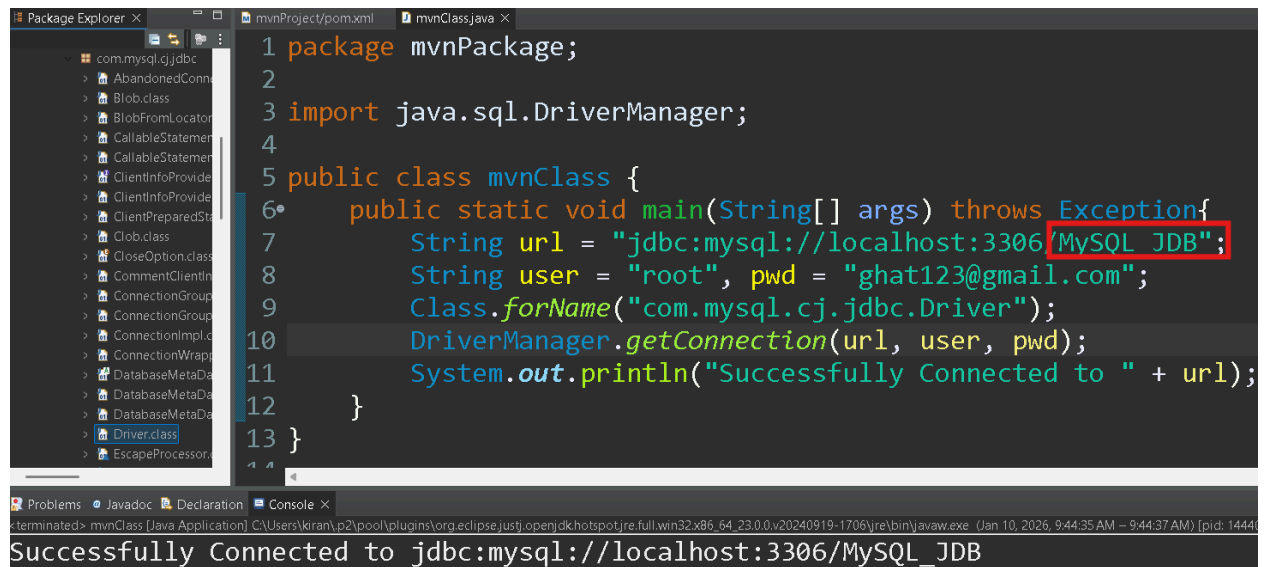
Table: **user_details**

Columns:

User_Name	varchar(50)
Score	decimal(5,2)
Email	varchar(50)

Then clear the code. No need to create database and table once again.

Make sure name of the database must be same



```
1 package mvnPackage;
2
3 import java.sql.DriverManager;
4
5 public class mvnClass {
6     public static void main(String[] args) throws Exception{
7         String url = "jdbc:mysql://localhost:3306/MySQL_JDB";
8         String user = "root", pwd = "ghat123@gmail.com";
9         Class.forName("com.mysql.cj.jdbc.Driver");
10        DriverManager.getConnection(url, user, pwd);
11        System.out.println("Successfully Connected to " + url);
12    }
13 }
```

Successfully Connected to jdbc:mysql://localhost:3306/MySQL_JDB

After creating database and table in workbench(manually done by user) along with driver connection, lets insert values from Java Class

Different ways to creat statement:

```
// PreparedStatement ps = ctn.prepareStatement("insert into
User_Details values('FirstUserName', 2.32,
'FUN123@gmail.com')");
// Alternative of Above
// PreparedStatement ps =
ctn.prepareStatement("insert into User_Details values('" +
username + "', " + score + ", '" + email + "')");
// Alternative of Above
PreparedStatement ps = ctn.prepareStatement("insert
into User_Details values(?, ?, ?)");
ps.setString(1, username);
ps.setFloat(2, score);
ps.setString(3, email);
```

Final code to insert users in userdetails table.

```
package mvnPackage;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
public class _01_insertUser {
    public static void main(String[] args) throws Exception{
```

```

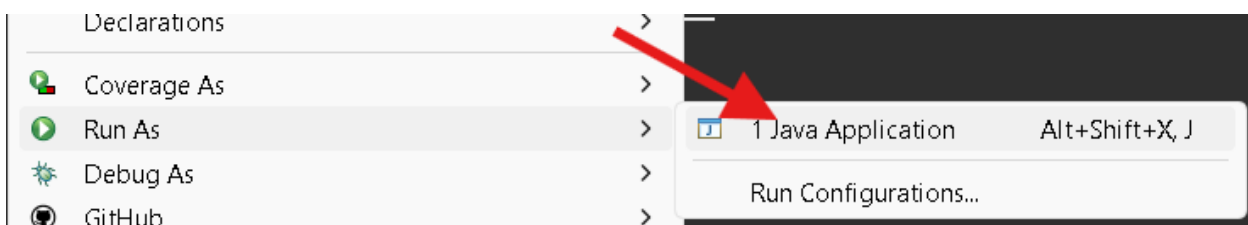
String url = "jdbc:mysql://localhost:3306/MySQL_JDB";
String user = "root", pwd = "ghat123@gmail.com";
Class.forName("com.mysql.cj.jdbc.Driver");
Connection ctn = DriverManager.getConnection(url, user,
pwd);

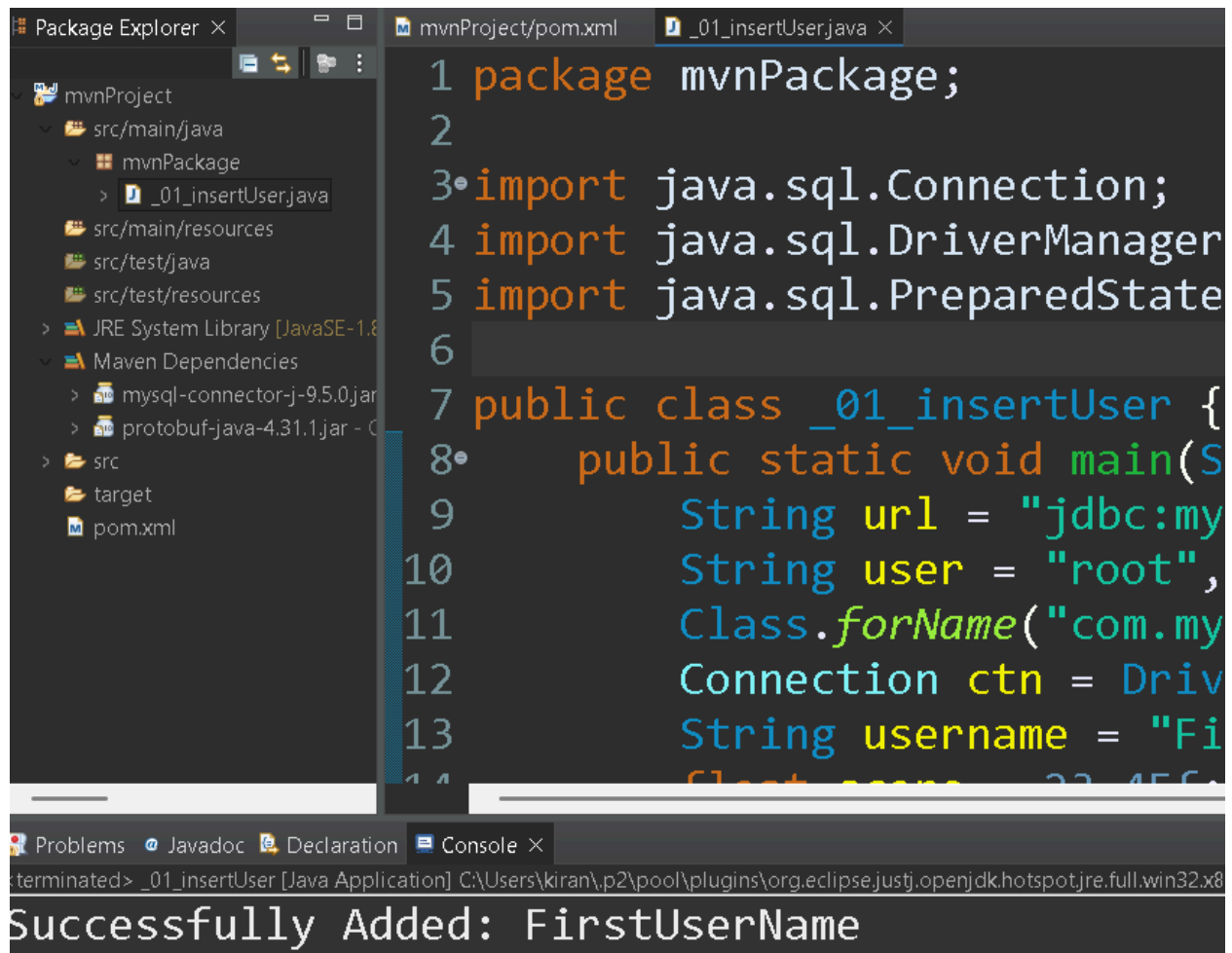
String username = "FirstUserName", email =
"FUN123@gmail.com";
float score = 23.45f;
// PreparedStatement ps = ctn.prepareStatement("insert into
User_Details values('FirstUserName', 2.32, 'FUN123@gmail.com')");
// Alternative of Above
// PreparedStatement ps = ctn.prepareStatement("insert into
User_Details values('" + username + "', " + score + ", '" + email +
"'");

// Alternative of Above
PreparedStatement ps = ctn.prepareStatement("insert into
User_Details values(?, ?, ?)");
// For first ? = username, next is score, then email, this
is decided when table is created in MySQL DB
ps.setString(1, username);
ps.setFloat(2, score);
ps.setString(3, email);
// Return 0 for fail, number > 0 for pass
int is_ps = ps.executeUpdate();
if (is_ps > 0) {
    System.out.println("Successfully Added: " + username);
}else System.out.println("Failed to Added: " + username);
ctn.close();
}
}

```

Run the code:

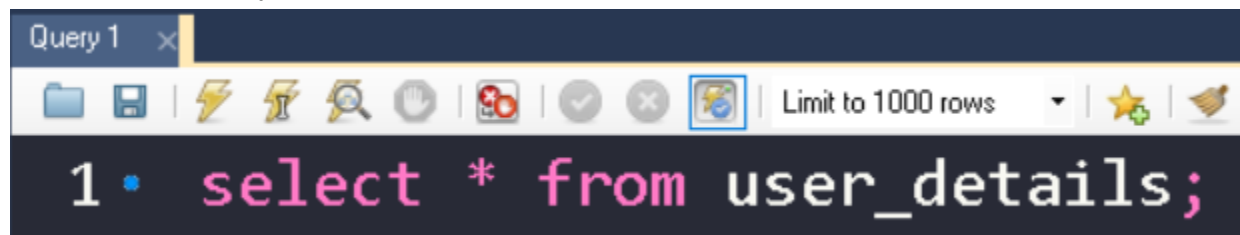




```
1 package mvnPackage;
2
3 import java.sql.Connection;
4 import java.sql.DriverManager;
5 import java.sql.PreparedStatement;
6
7 public class _01_insertUser {
8     public static void main(String[] args) {
9         String url = "jdbc:mysql://localhost:3306/mydb";
10        String user = "root";
11        Class.forName("com.mysql.jdbc.Driver");
12        Connection ctn = DriverManager.getConnection(url, user, "");
13        String username = "FirstUserName";
14        String password = "123456789";
15        String sql = "insert into user_details (username, password) values (?, ?)";
16        PreparedStatement pstmt = ctn.prepareStatement(sql);
17        pstmt.setString(1, username);
18        pstmt.setString(2, password);
19        pstmt.executeUpdate();
20    }
21 }
```

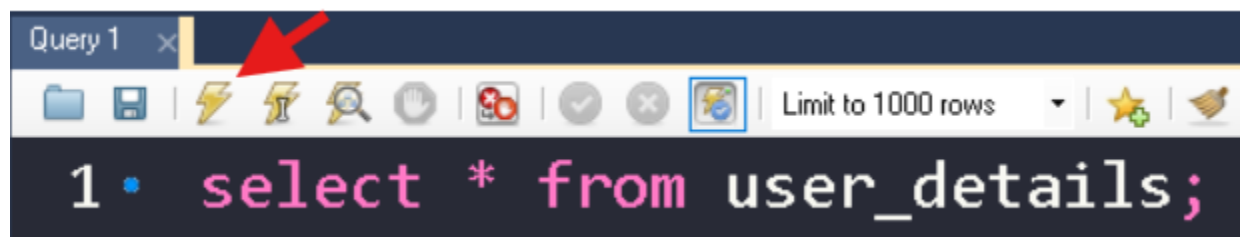
Successfully Added: FirstUserName

Write the code in MySQL:

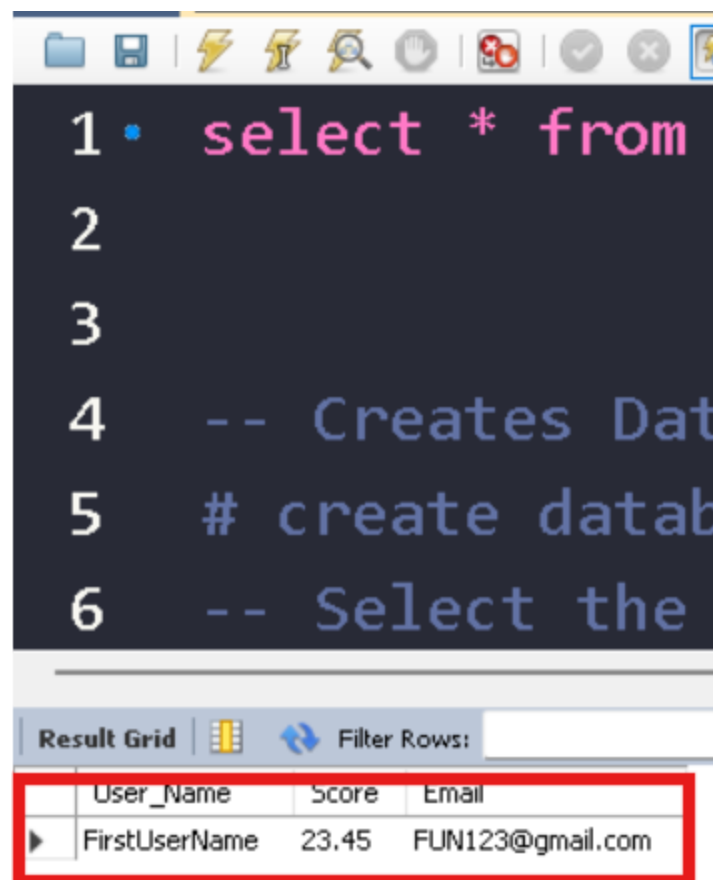


```
1 • select * from user_details;
```

Click:



```
1 • select * from user_details;
```



The screenshot shows a SQL IDE window. The top toolbar contains icons for file operations (folder, save, lightning bolt, copy, paste, search, zoom in, zoom out), editing (undo, redo), and window management (check, close, maximize). The query editor has a dark background with the following SQL code:

```
1 • select * from
2
3
4 -- Creates Dat
5 # create datab
6 -- Select the
```

Below the editor is a "Result Grid" tab with a "Filter Rows:" input field. The grid displays the results of the query, which are highlighted with a red border:

	User_Name	Score	Email
▶	FirstUserName	23.45	FUN123@gmail.com