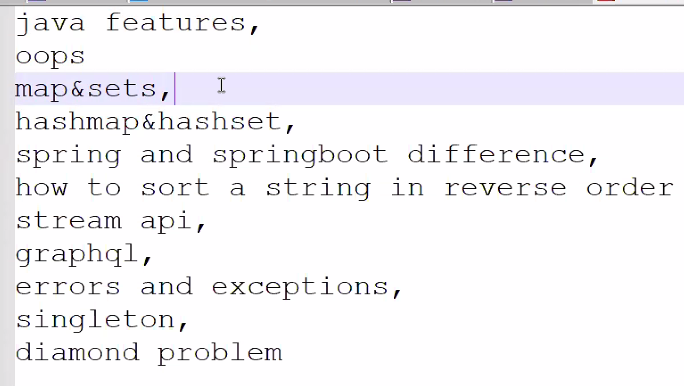
**ResultSet\_SQLInjection\_PreparedStatement**

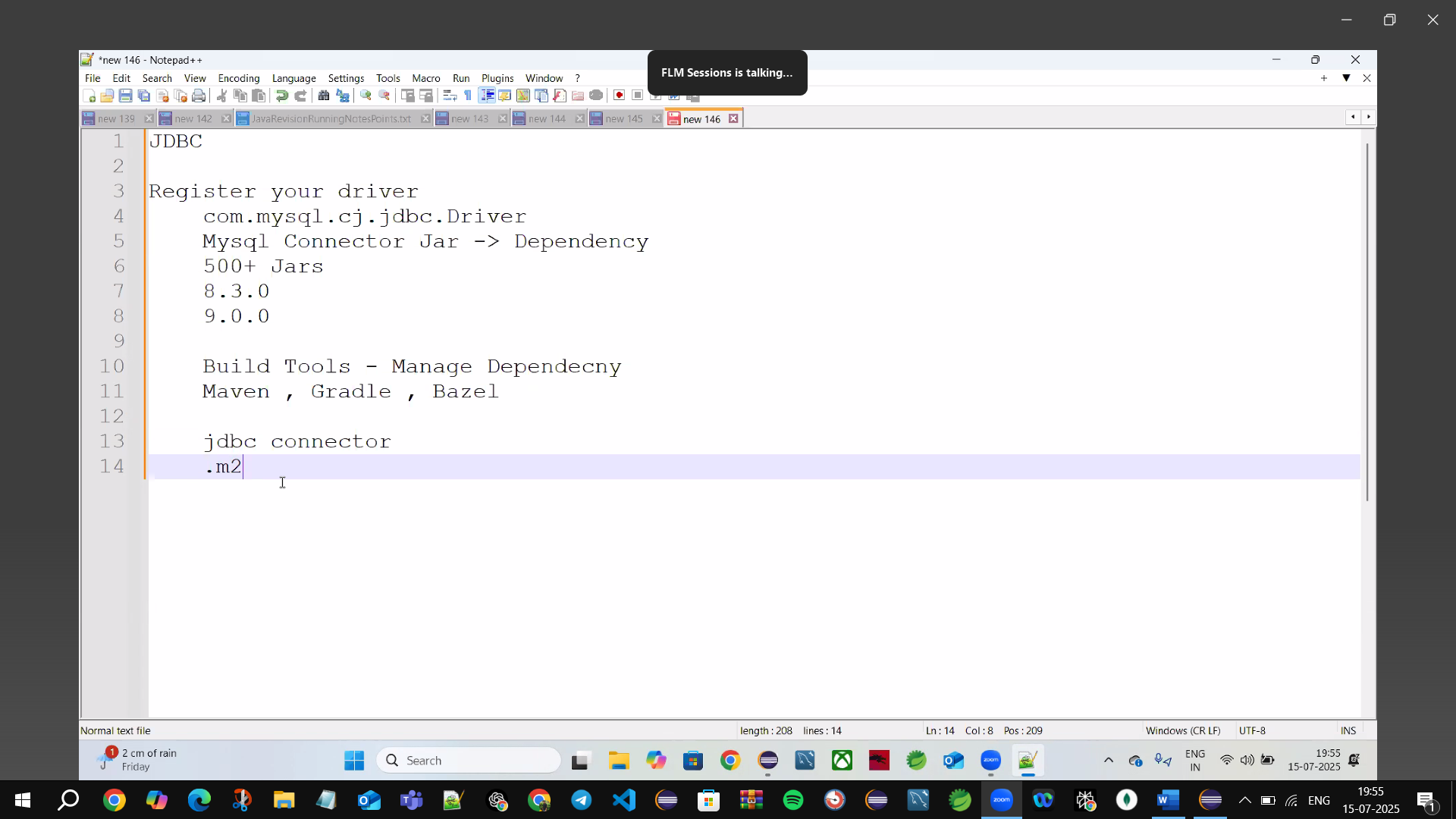
3yrs Exp Interview Q’s

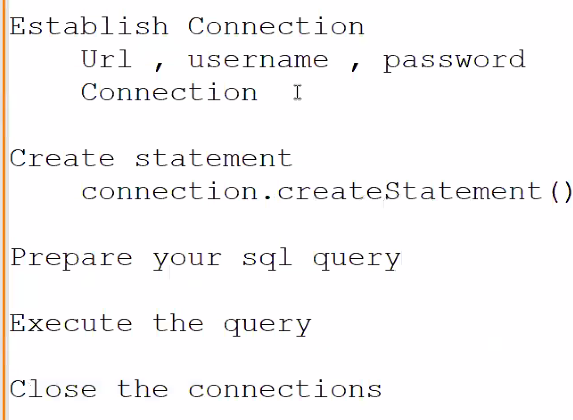


🡪treeMap don’t have nulls & sorts in the insertion order

Graphql🡪RESTAPI’S

Dimond Problem🡪Multiple Inheritance

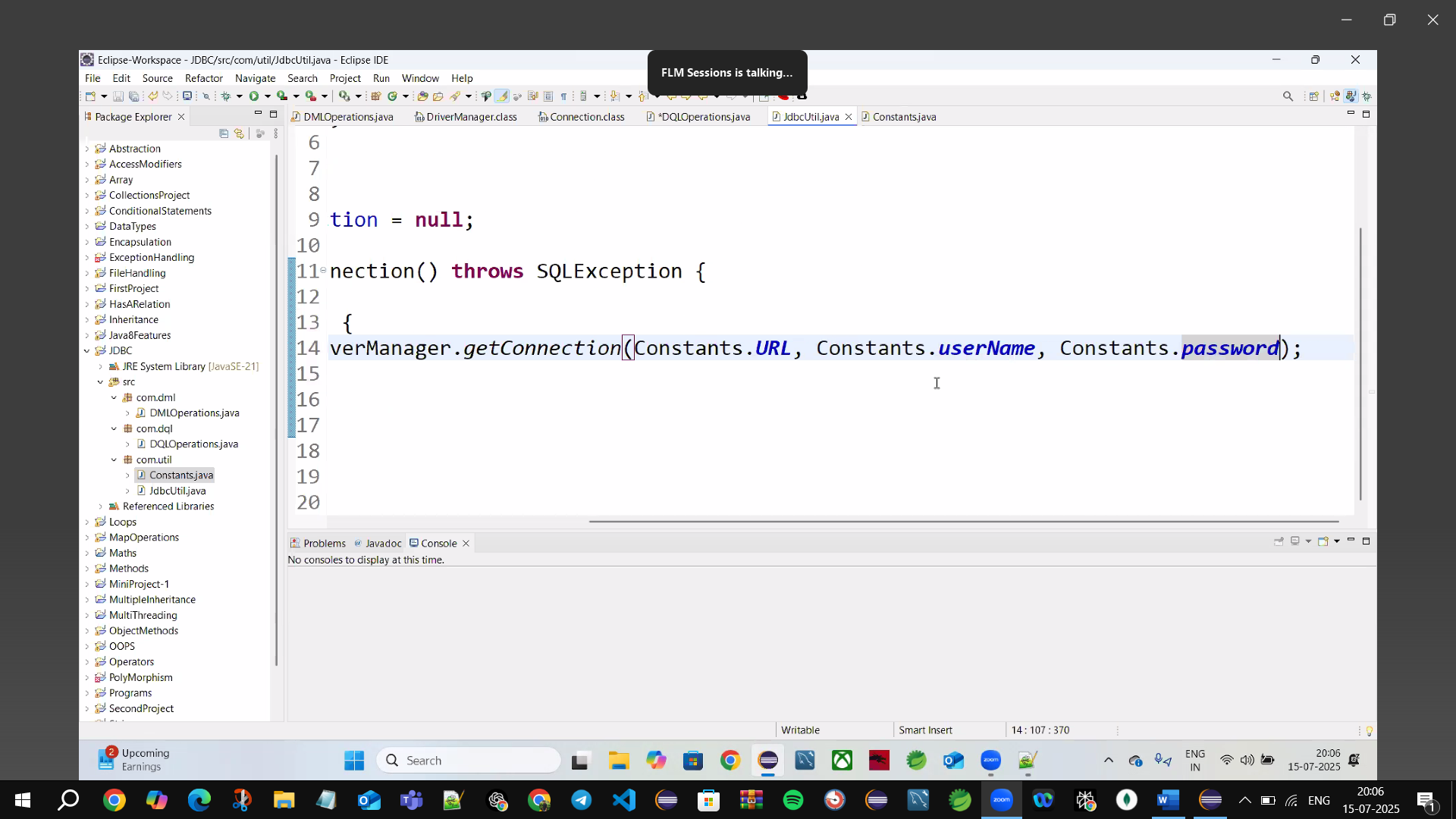




Build Tools 🡪used in Java🡪Maven, Gradle, Bazel (🡪Instead of Java Project🡪Use Maven Project)

🡪Maven manage dependencies initialize once & use any time it stores dependence in .m2 folder(maintain cache)

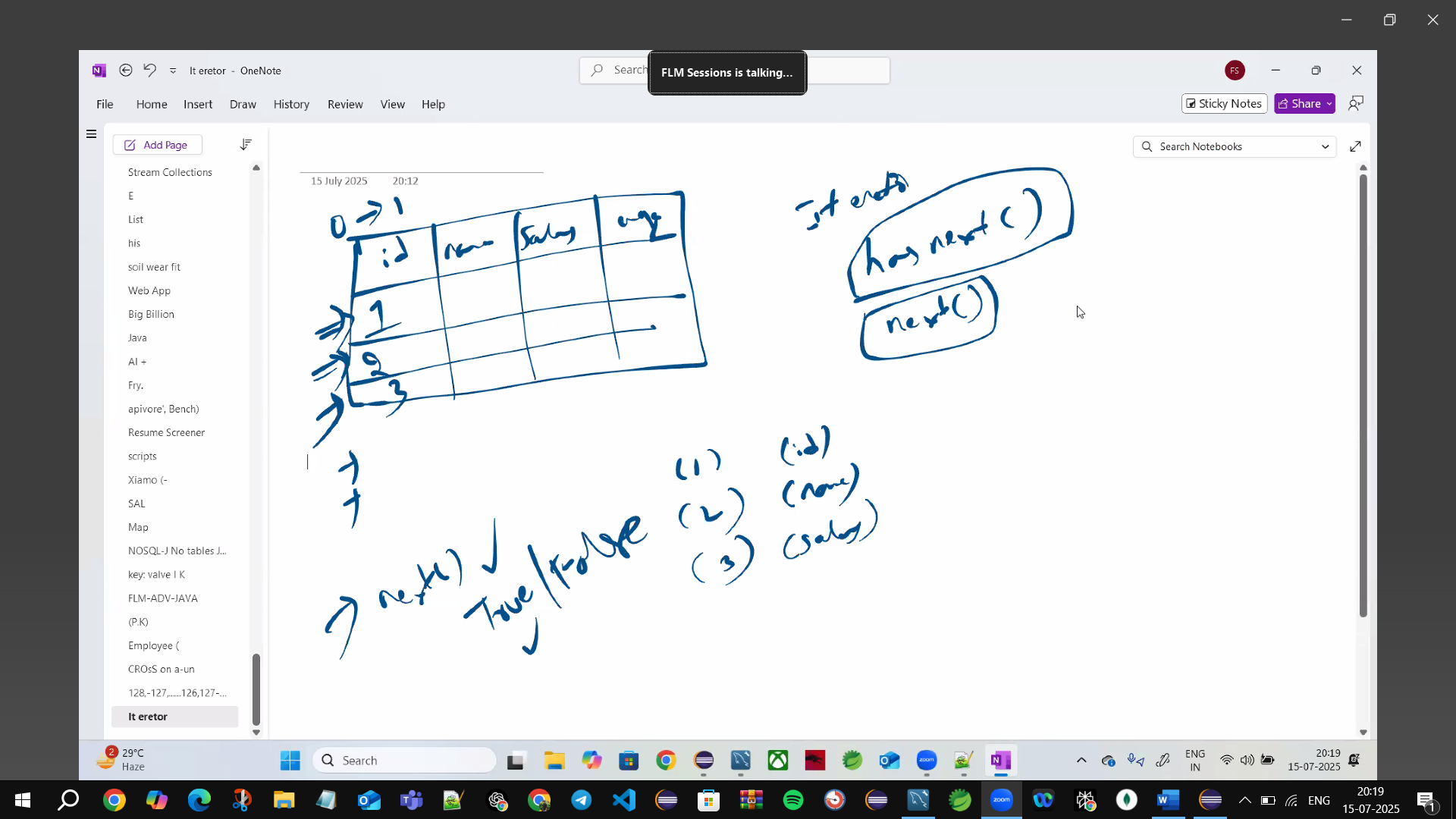






Iterator b/f 🡪0 Zero (iterator b/w 0-1)

Result set 🡪same like table uses next()🡪Moves the cursor to the next row. Returns true if the row exists; otherwise, false.

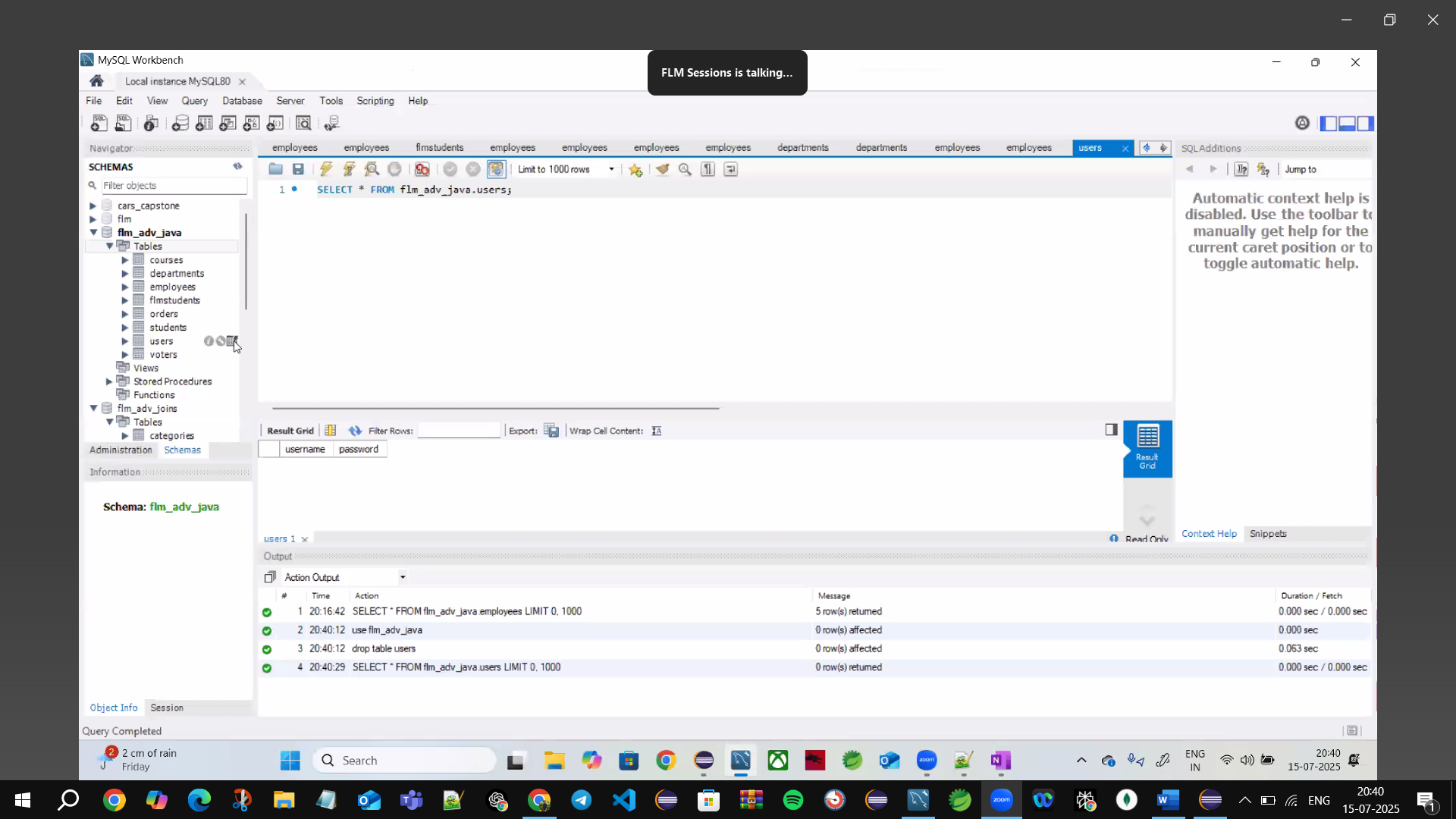


🡪Iterator 🡪use while loop

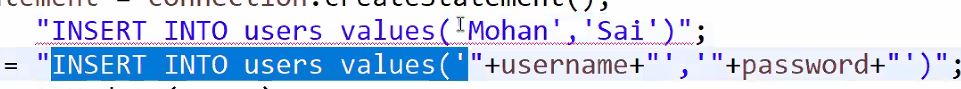
**When we Do DQL operations use 🡪executeQuery(query)**

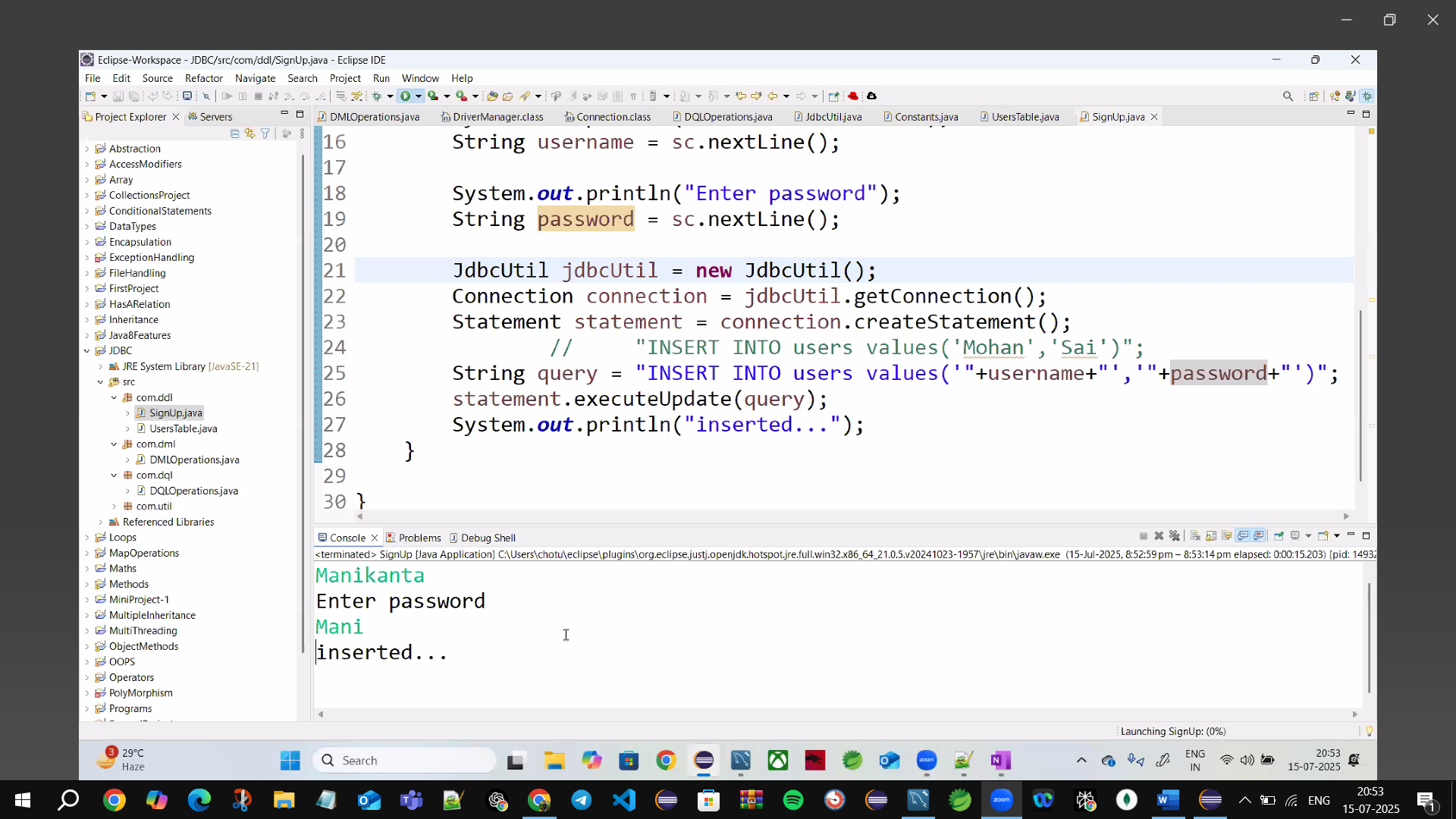
**When we Do DDL Operation use 🡪executeUpdate(query);**

**Whenever we do Select Operation in JDBC we use 🡪ResultSet🡪has directly contains next() tells true/ false🡪then move forward**

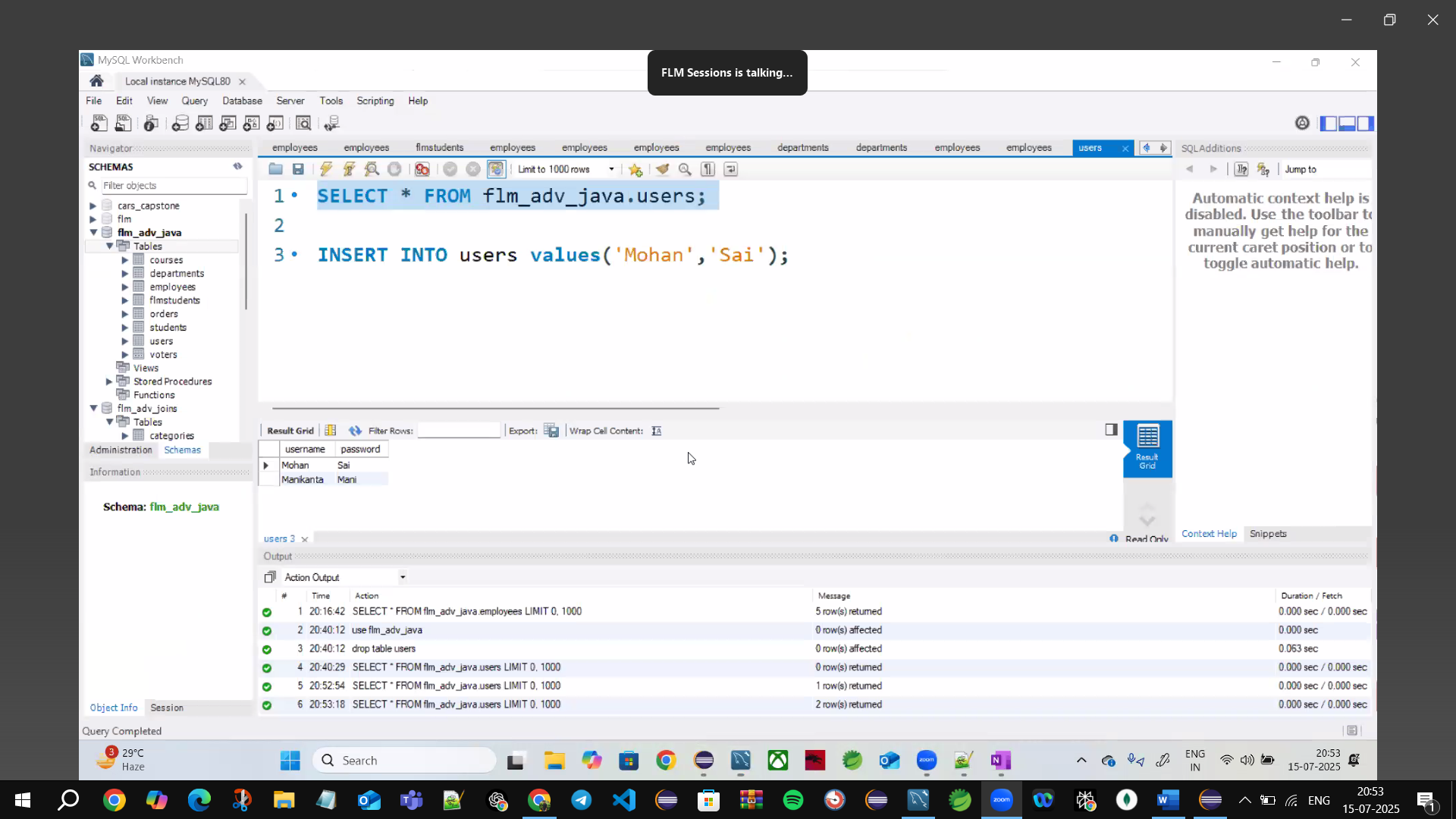


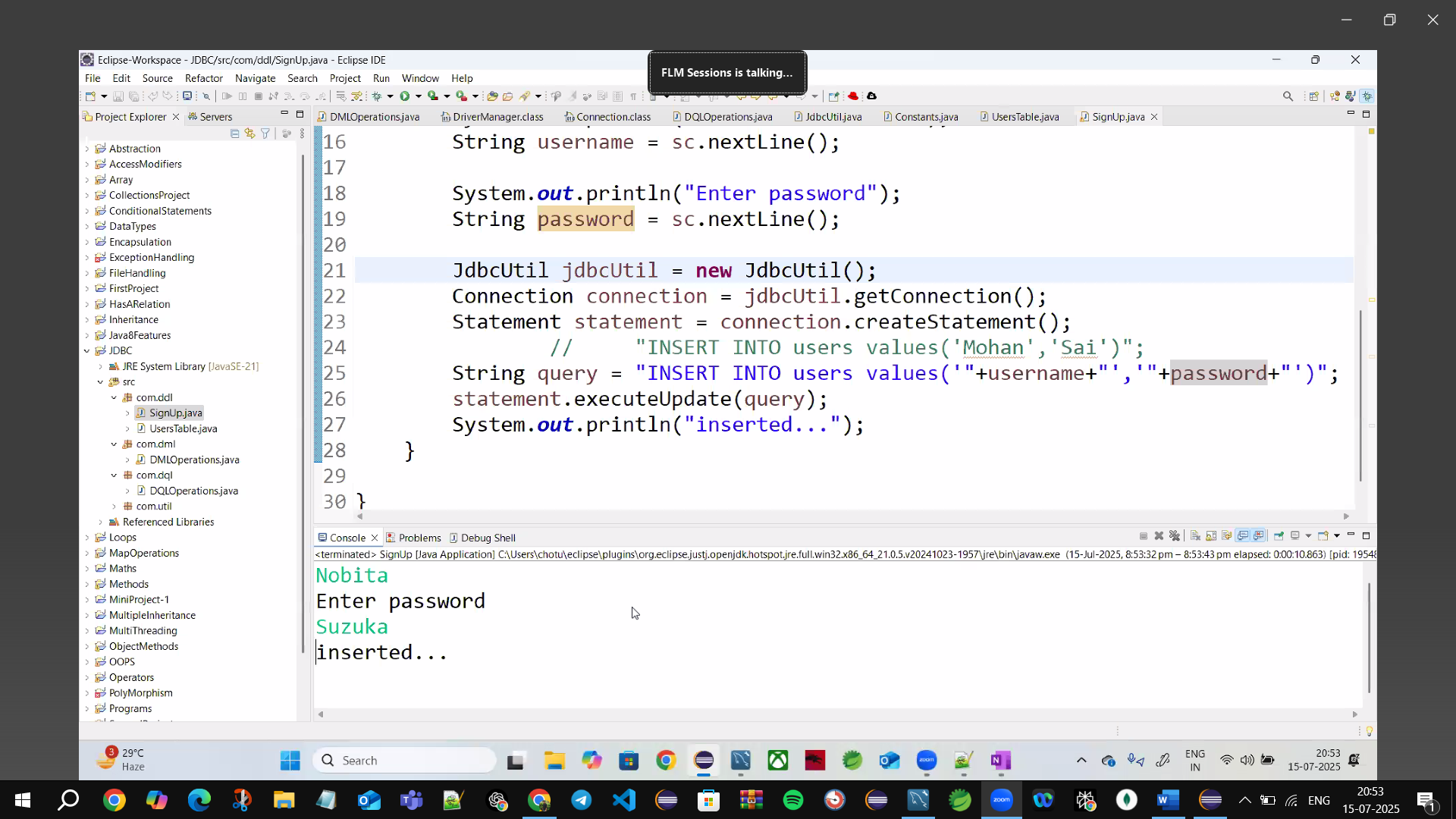
We Don’t do DDL commands in JDBC

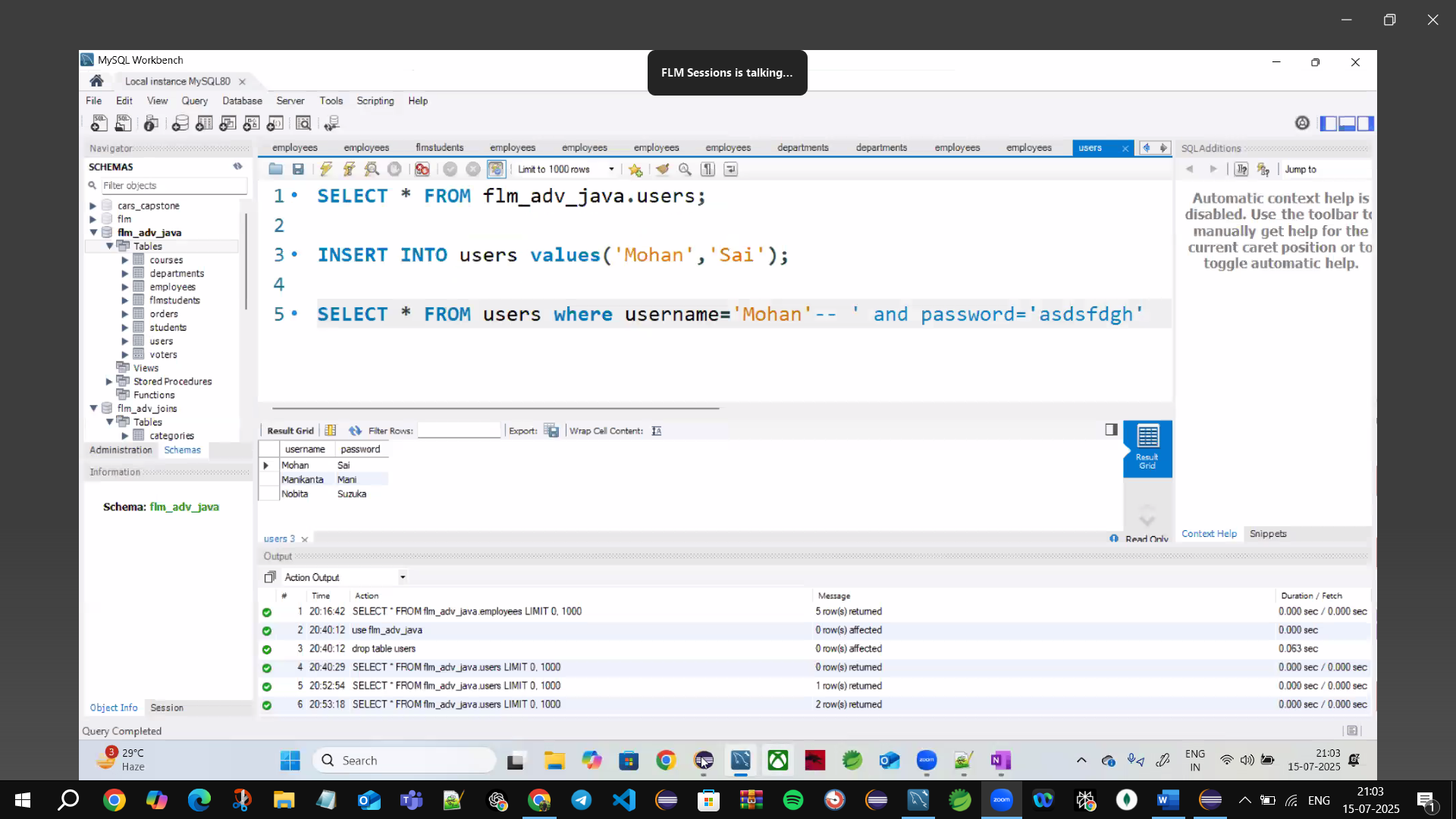




🡪We don’t do Table creation form code not good Approach





SQLInjections🡪to Overcome this use Prepared Statement



**SQL Injection**

**✅ Fix: Add a space after --**

Try this input instead:

Username: Balaji' --

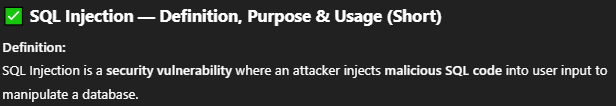
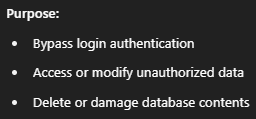
↑ (note the space after --)

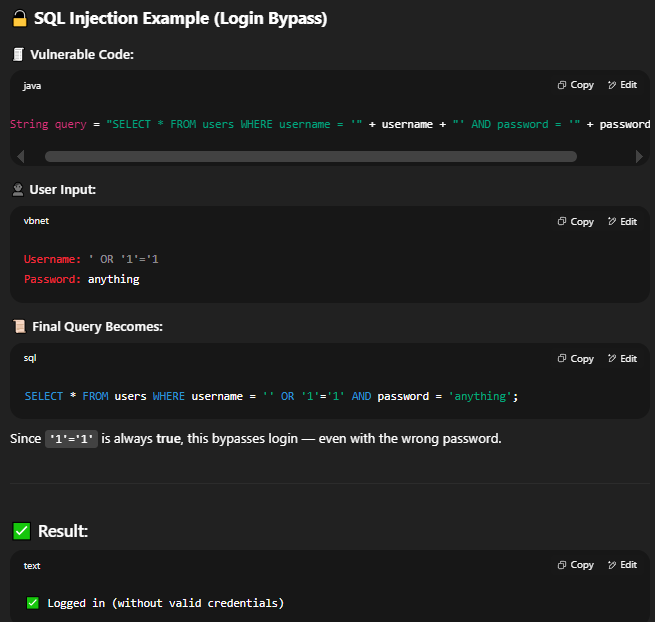
Password: anasa

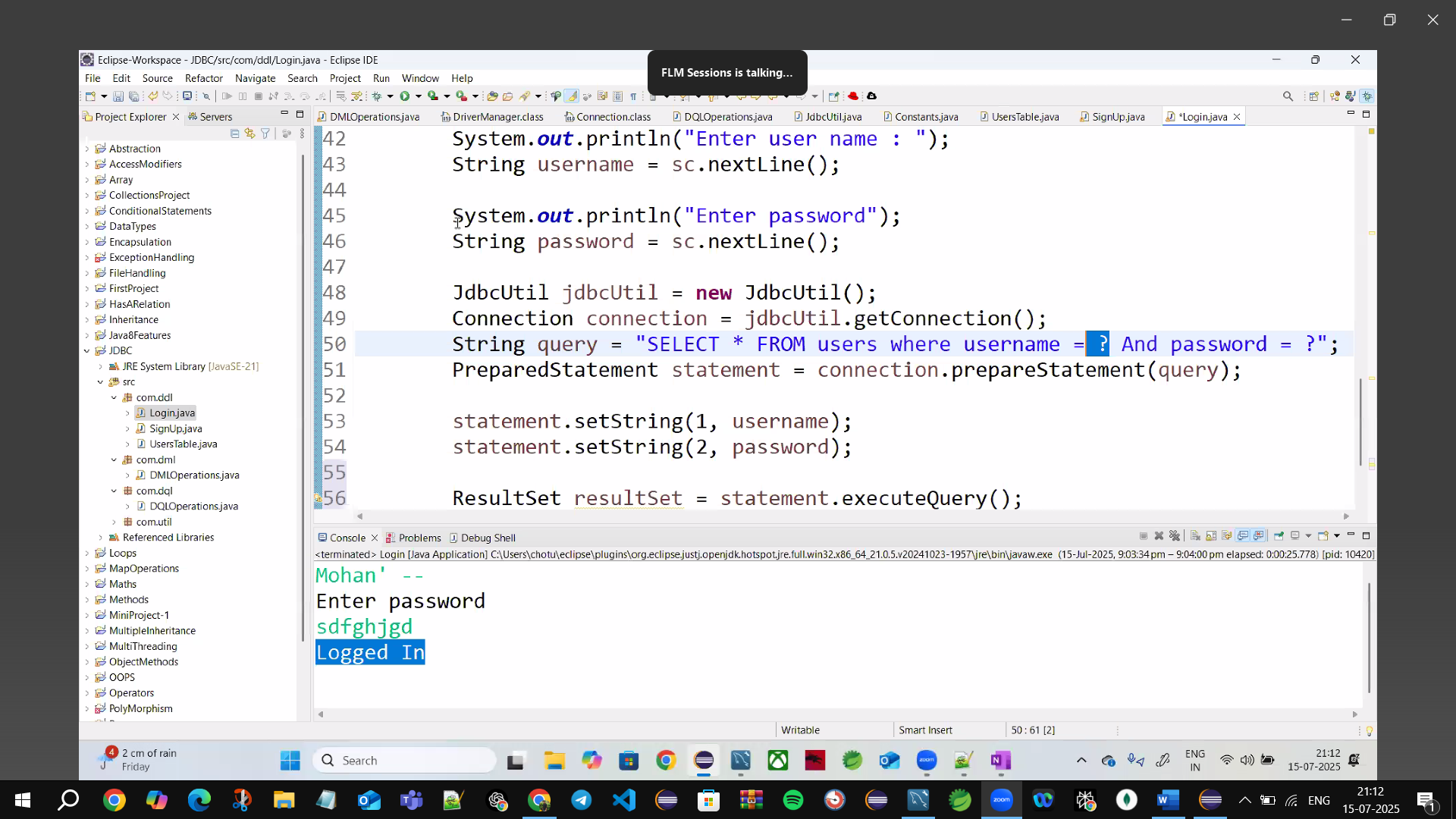
Now the final query becomes:

SELECT \* FROM users WHERE username = 'Balaji' -- AND password = 'anasa'

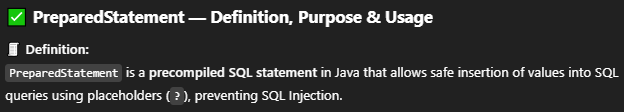
✅ This is valid SQL: the -- comments out the password condition, and if Balaji exists in the table, rs.next() will succeed and print ✅ Logged in.

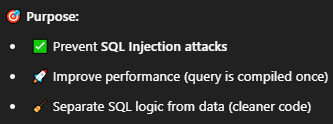
  

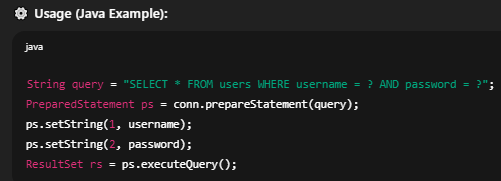




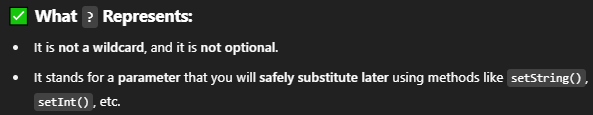
PreparedStatements

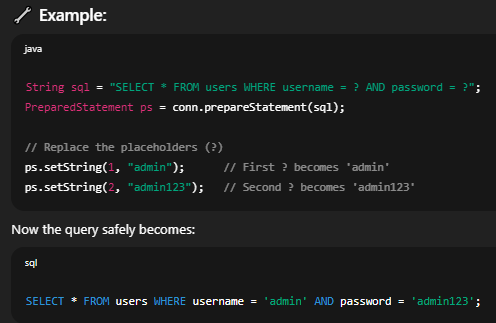


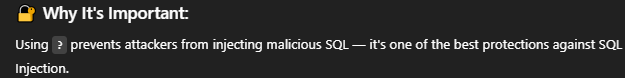


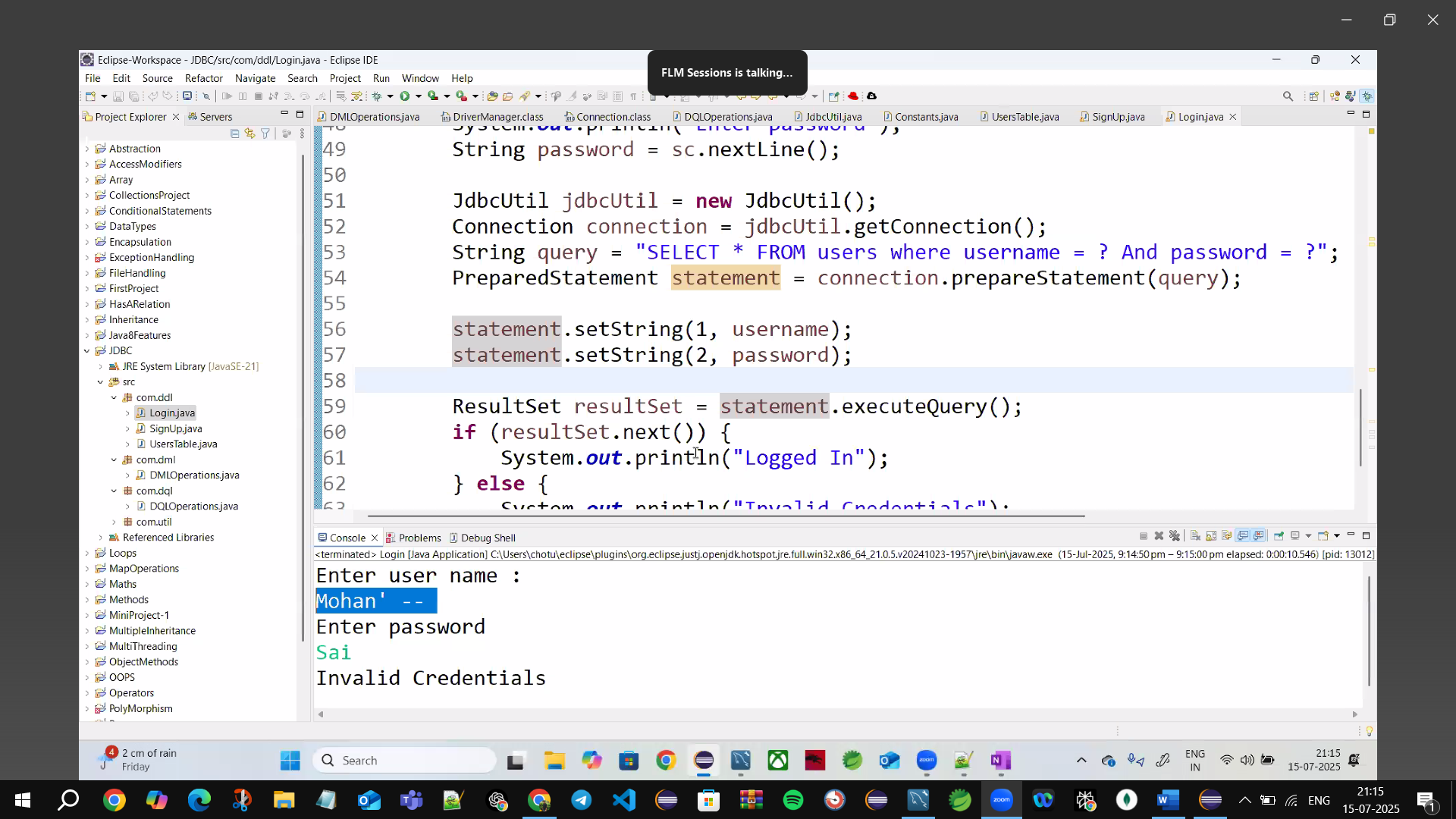


🡪The ? in a PreparedStatement represents a **placeholder for a value**.

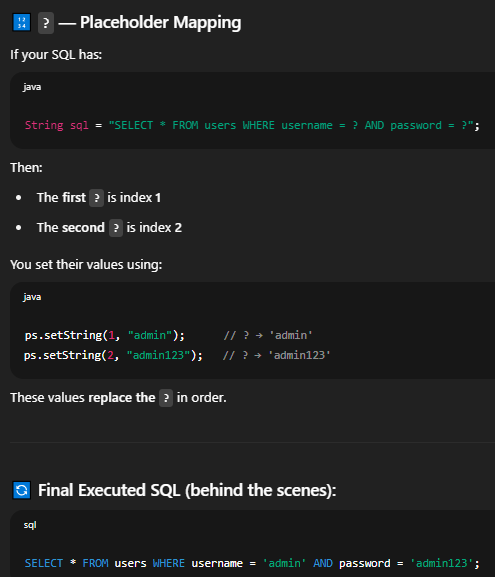




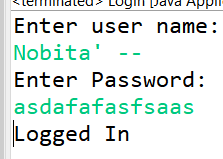


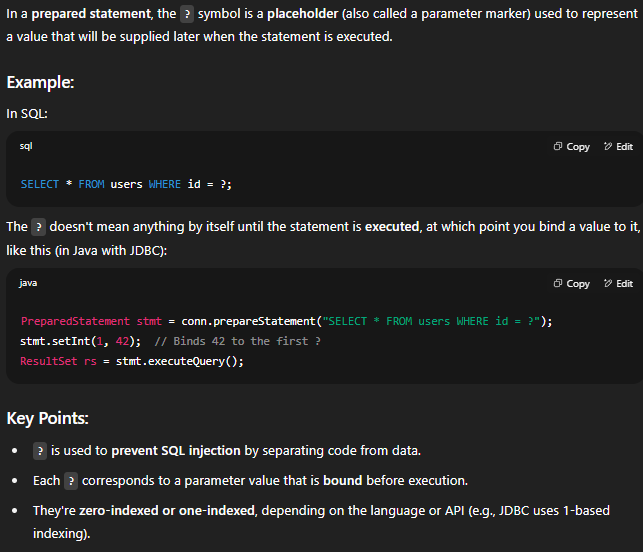


?-->1, ?-->2 these are values



SQL Injection 🡪By pass the user Credentials by commenting after username ‘ -- space🡪take any password (🡪-- comments all password and logged in Successfully Hacked the all user data ) to overcome this SQL injection use🡪Prepared statements with secure 🡪? ? current data using indexes











?--> Placeholder/Parameter marker 🡪represents a value

?->parameter value bound b/f execution