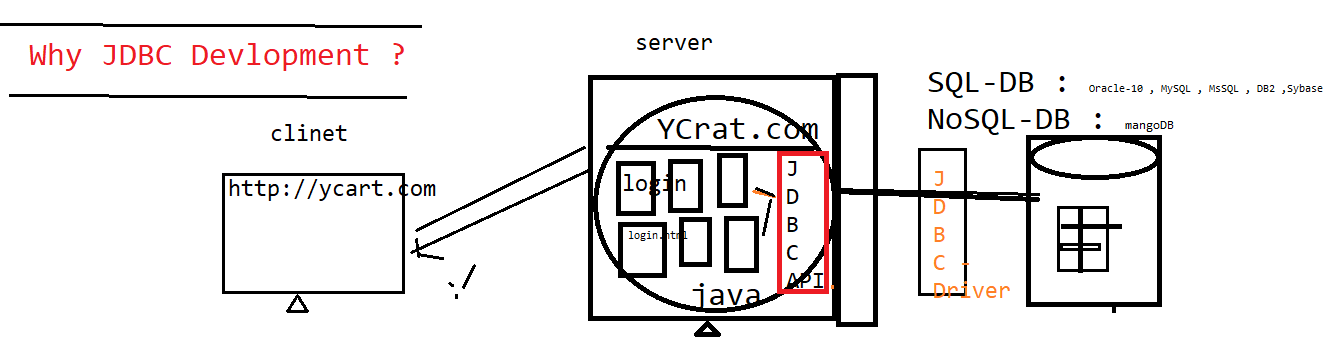
What is JDBC?

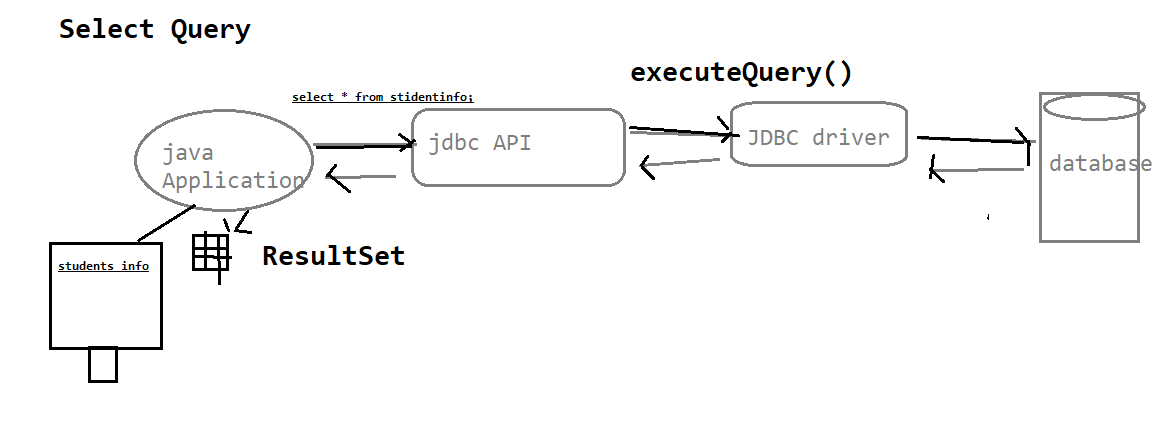
It’s a J2AEE API, which is used to connect to any data base from the java code

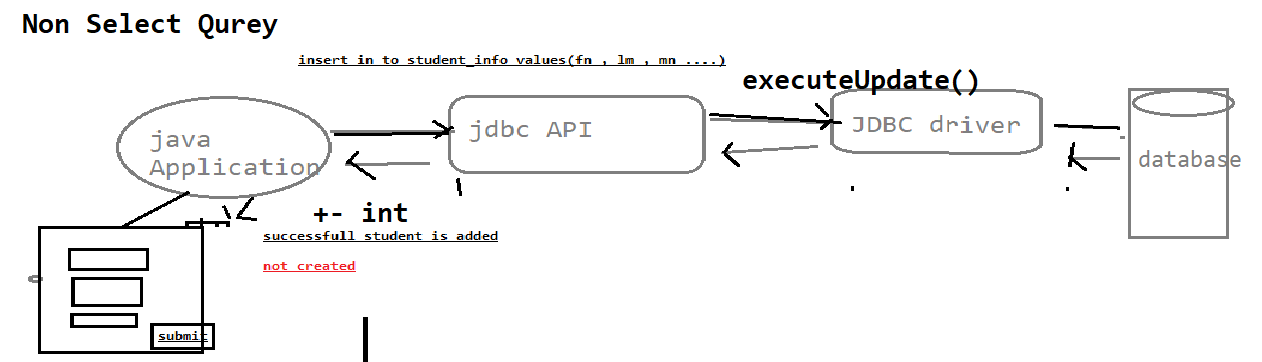




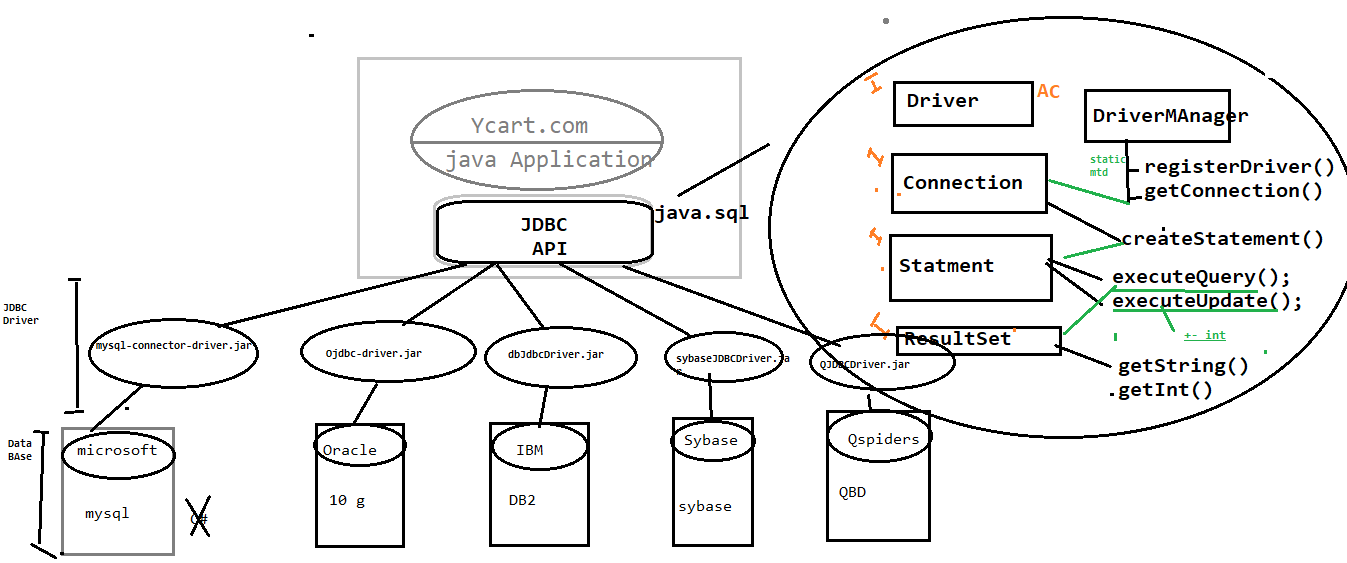
JDBC helps java program to store / retrieve data from the physical database

JDBC Workflow?





What is JDBC Drivers & JDBC API?

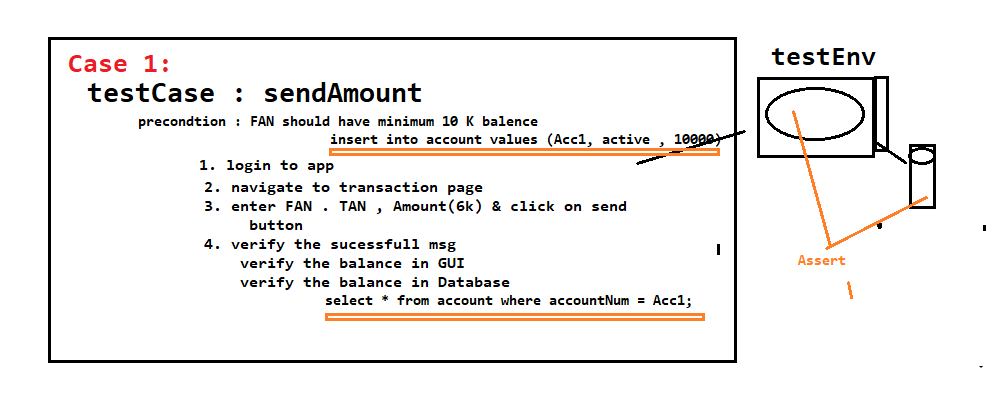


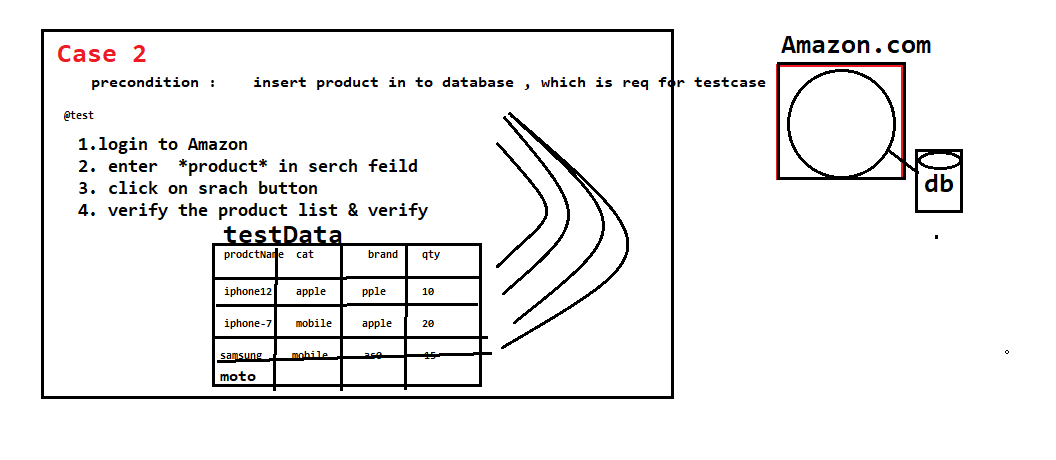
JDBC API: It’s an API’s available in java.sql package, which is collection interfaces with abstract method & class, & its database independent

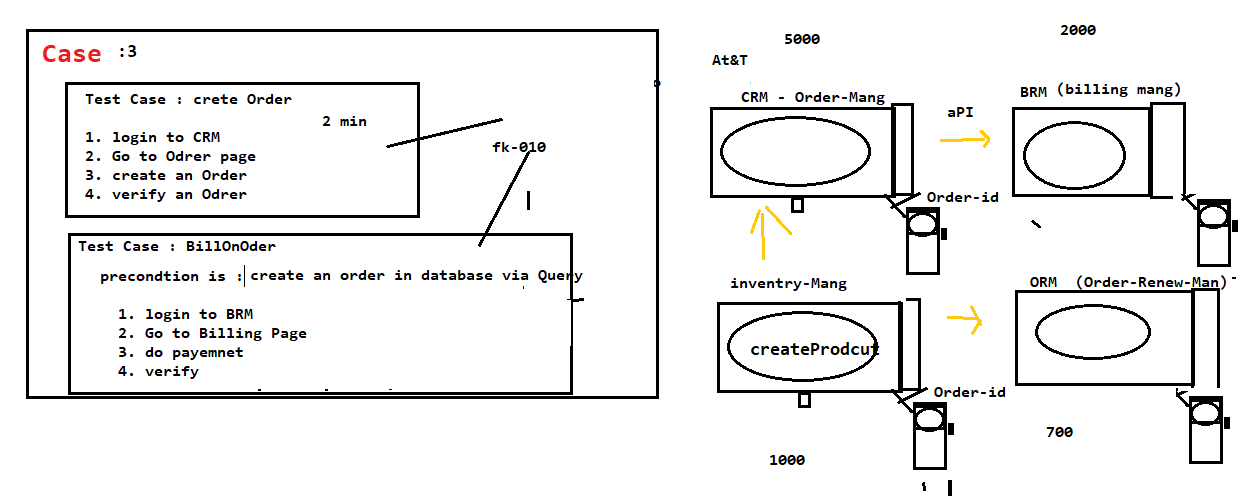
JDBC Driver: its implementation class of JDBC API provided by data base vender. it’s a db dependent

Why JDBC in Automation?

In Automation also required JDBC Connection, Most of the test case Preconditions, required few set of data in Database & also validation GUI WRT Database







**ExecuteQuery()**

@Test

public void getExecuteQuery() throws SQLException

{

//Step 1: Register the database

Driver driverref=new Driver();

DriverManager.registerDriver(driverref);

//Step 2: establish Connection with database

Connection con=DriverManager.getConnection("jdbc:mysql://localhost:3306/projects", "root", "root");

//Step 3: Issue create statement

Statement stat=con.createStatement();

//Step 4: Execute any query

ResultSet result = stat.executeQuery("select \* from project;");

while(result.next())

{ System.out.println(result.getString(1)+"\t"+result.getString(2)+"\t"+result.getString(3)+"\t"+result.getString(4));

}

//Step 5: Close database connection

con.close();

}

**ExecuteUpdate() With finally block**

@Test

public void getExecuteUpdate() throws SQLException

{

Connection con = null;

try {

//Step 1: Register the database

Driver driverref=new Driver();

DriverManager.registerDriver(driverref);

//Step 2: establish Connection with database --- database

con=DriverManager.getConnection("jdbc:mysql://localhost:3306/projects", "root", "root");

//Step 3: Issue create statement

Statement stat=con.createStatement();

//Step 4: Execute any query -- table

int result = stat.executeUpdate("insert project values('TY\_PROJ\_999','RINCY','26/04/2021','SDET-13','ON-GOING',62);");

if(result==1)

{

System.out.println("Project is successfully created");

}

else

{

System.out.println("Query failed");

}

}

catch(Exception e) {

}

finally {

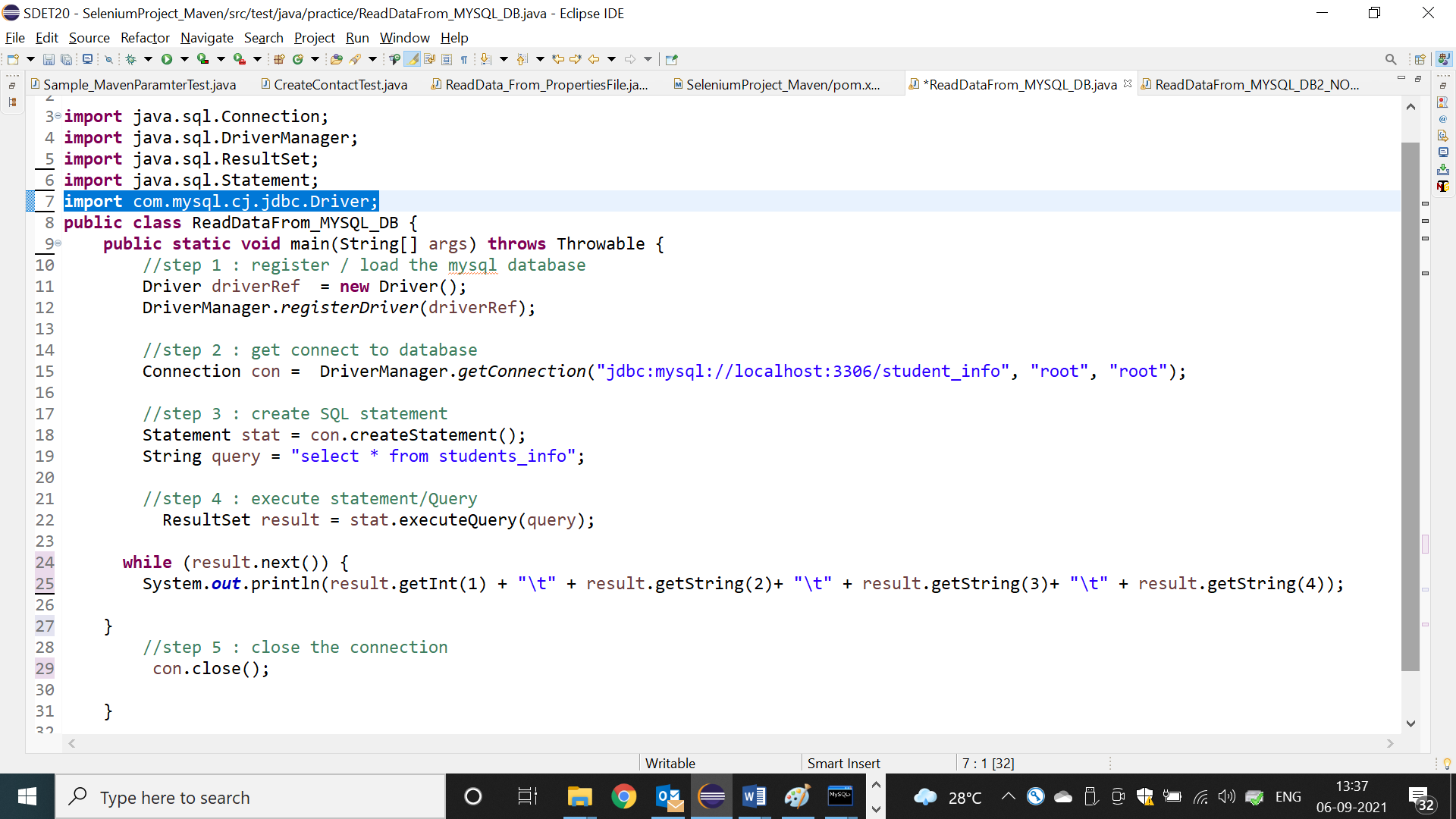
//Step 5: Close database connection

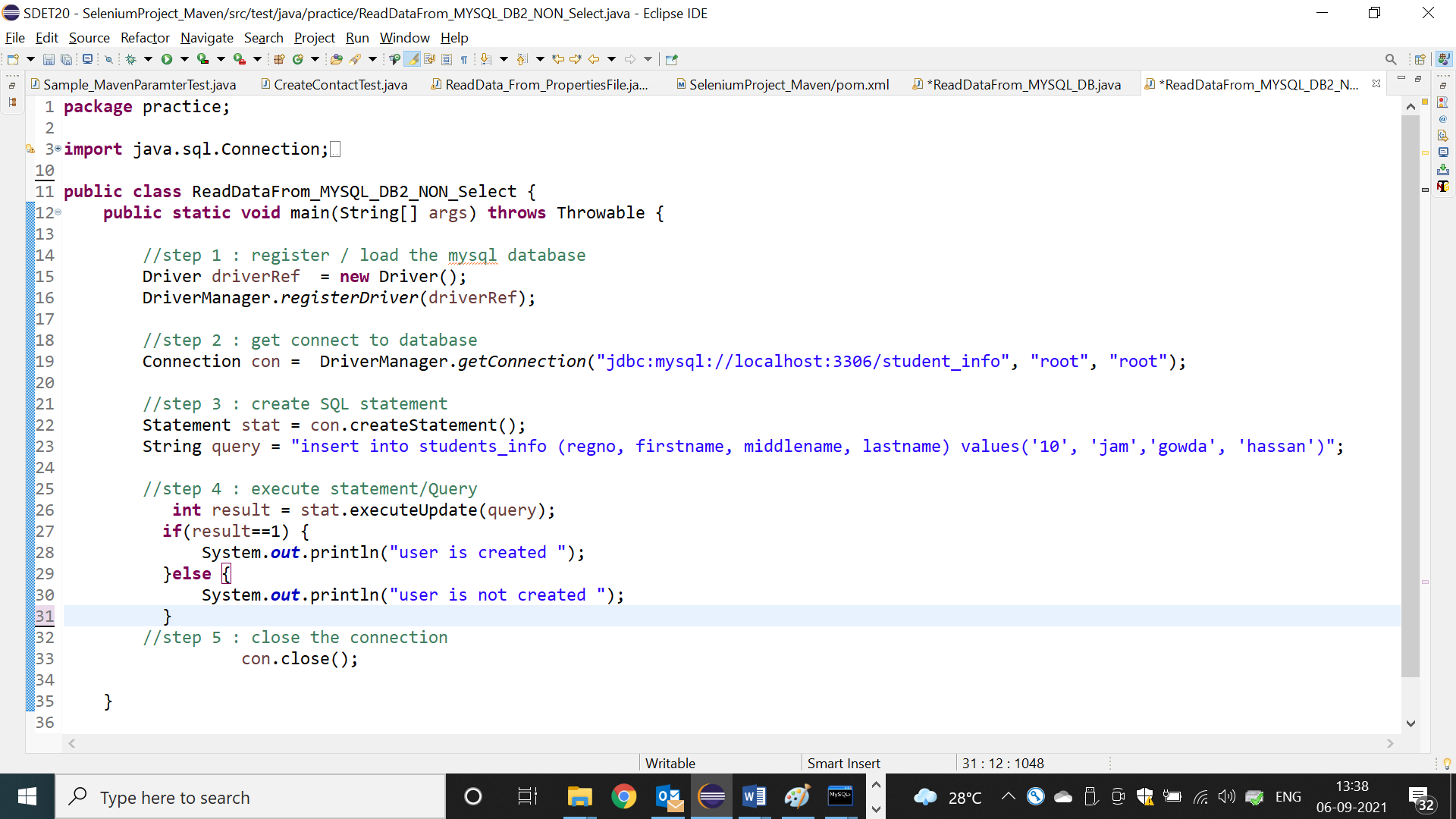
con.close();

System.out.println("Connection closed");

}

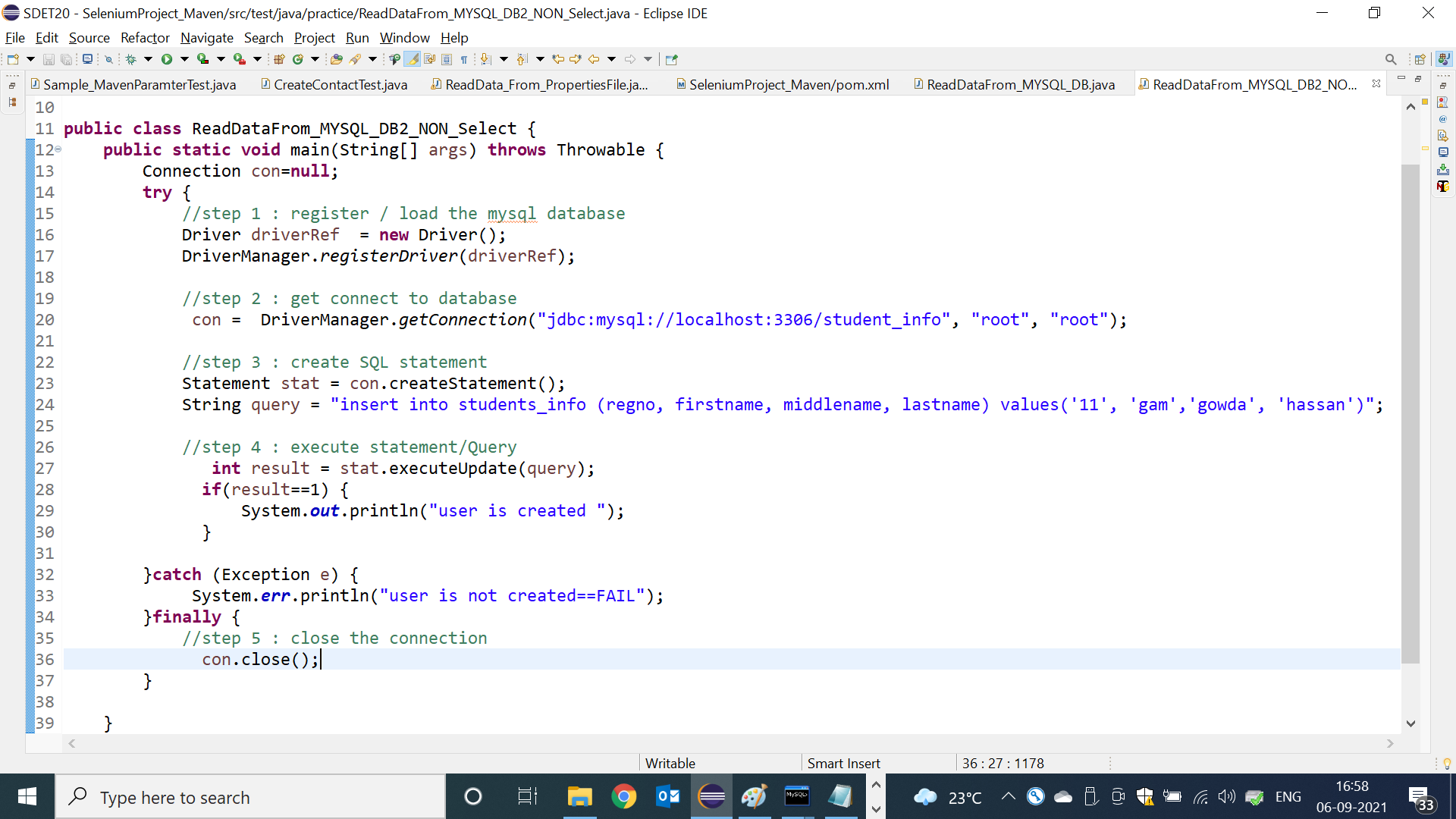
}





**JDBC Program with try /Catch/Finally block**

Note: Real time JDBC code should be inside the try/catch/finally block, because connection object will get closed in positive/ negative scenario



Syntax for creating a table:

CREATE TABLE STUDENT (STU\_NAME VARCHAR(25) NOT NULL, STU\_ID int, PHONE\_NO int NOT NULL, DEPARTMENT VARCHAR(20) NOT NULL, PRIMARY KEY(STU\_ID));

Syntax for inserting the values:

insert into student (stu\_name, stu\_id, phone\_no, department) values ('Jothi',3333,1122112211,'Electrical');

