Course Name: A Level (2nd Sem) **Subject: JAVA**
Topic: JDBC(Part3) **Date: 24-06-20**

JDBC Statement:

java.sql.Statement is an interface. It provides some methods through which we can submit SQL queries to the database. It will be implemented bydriver implementation providers. createStatement() method on Connection object will return Statement object.There are 3 types of JDBC statement which are given below:

Statement	PreparedStatement	CallableStatement
It is used to execute normal SQL queries.	It is used to execute parameterized or dynamic SQL queries.	It is used to call the stored procedures.
It is preferred when a particular SQL query is to be executed only once.	It is preferred when a particular query is to be executed multiple times.	It is preferred when the stored procedures are to be executed.
You cannot pass the parameters to SQL query using this interface.	You can pass the parameters to SQL query at run time using this interface.	You can pass 3 types of parameters using this interface. They are – IN, OUT and IN OUT.
This interface is mainly used for DDL statements like CREATE, ALTER, DROP etc.	It is used for any kind of SQL queries which are to be executed multiple times.	It is used to execute stored procedures and functions.
The performance of this interface is very low.	The performance of this interface is better than the Statement interface (when used for multiple execution of same query).	The performance of this interface is high.

Note: In most of the cases the prepared statement is used frequently.

Basic JDBC Examples IN MYsql:

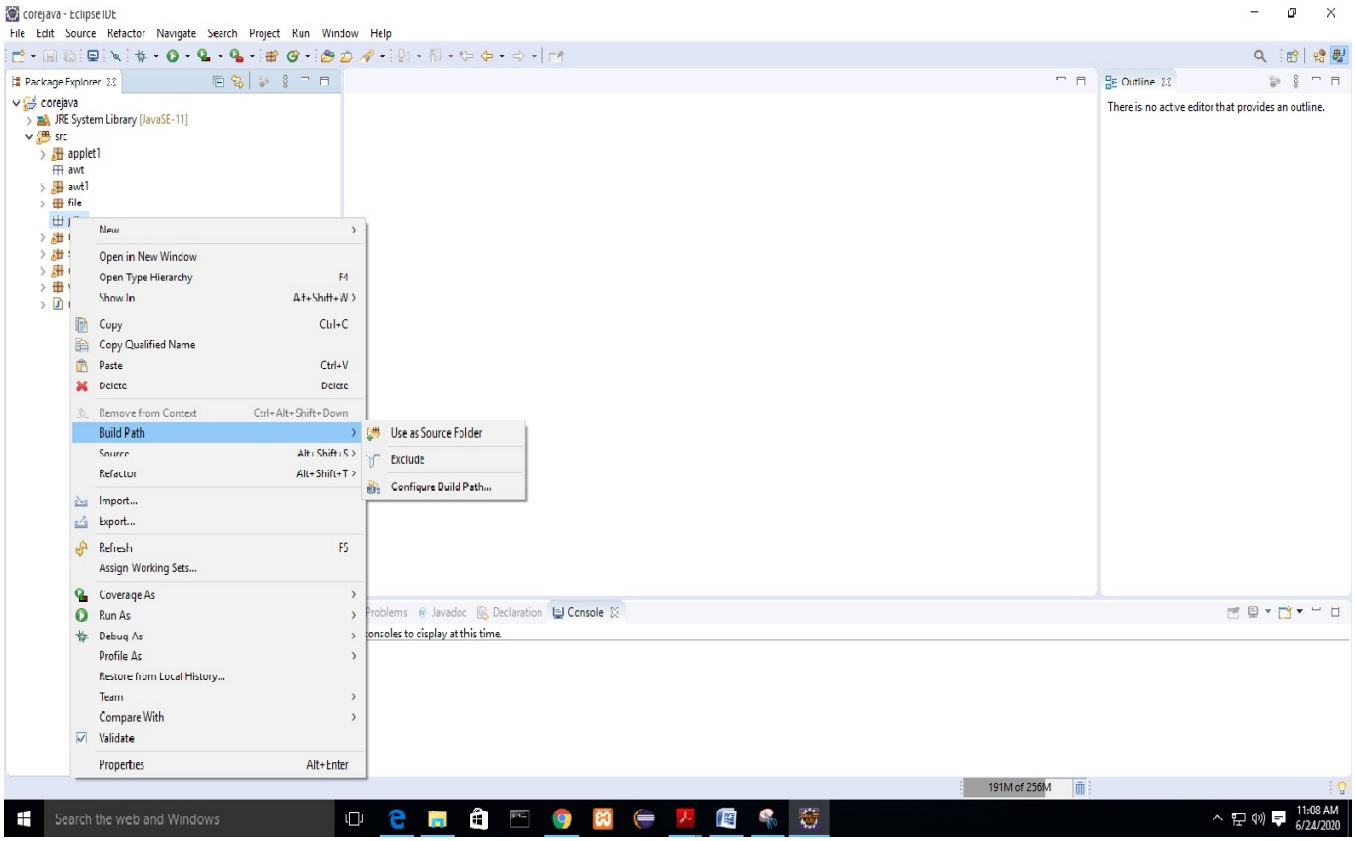
1)To create a Table

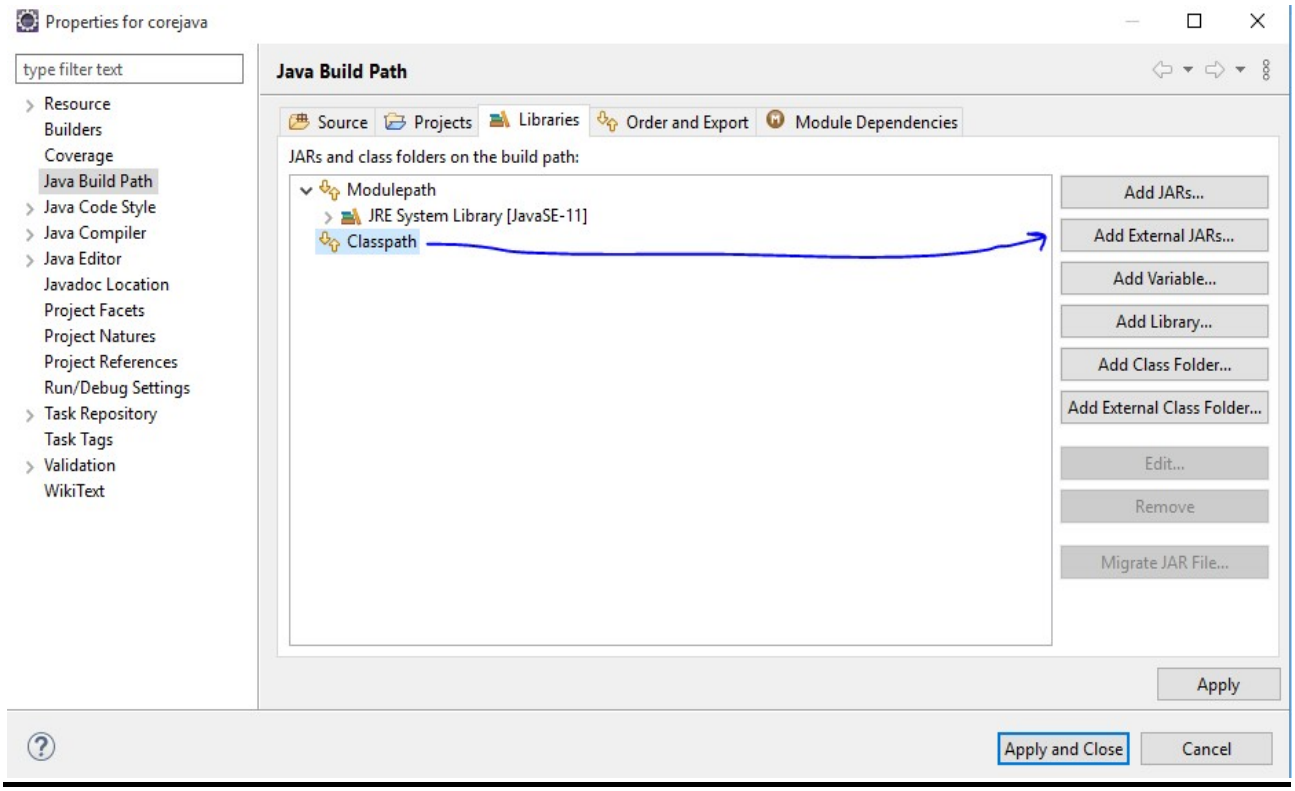
```
public class CreateTable {
public static void main(String[] args) {
try {
Class.forName("com.mysql.jdbc.Driver");
Connection
con=DriverManager.getConnection("jdbc:mysql://localhost:3306/test","root","root");
Statement stmt = con.createStatement();
String sql = "create table emp1(name varchar(25),department varchar(5),salary
varchar(25))";

stmt.executeUpdate(sql);
System.out.println("Table is sucessfully created");
con.close();
}
catch (ClassNotFoundException e)
{
e.printStackTrace();
}
catch (SQLException e)
{
e.printStackTrace();
}
}
}
```

Note: In all JDBC programs, a file called as mysql.jar(available on net) needs to be imported in eclipse project, without that a project will not run. But be sure to match your database version with the mysql.jar version. For a reference on how to add mysql.jar I am adding screenshot.

Steps:

Right click on project→build path→configure build path→libraries→classpath→add external jar


corejava - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
corejava
JRE System Library [JavaSE-11]
src
applet1
awt
awt1
file
New
Open in New Window
Open Type Hierarchy F4
Show In Alt+Shift+W
Copy Ctrl+C
Copy Qualified Name
Paste Ctrl+V
Delete Delete
Remove from Context Ctrl+Alt+Shift+Down
Build Path
Source Alt+Shift+S
Refactor Alt+Shift+T
Import...
Export...
Refresh F5
Assign Working Sets...
Coverage As
Run As
Debug As
Profile As
Restore from Local History...
Team
Compare With
Validate
Properties Alt+Enter
Use as Source Folder
Exclude
Configure Build Path...
Outline
There is no active editor that provides an outline.
Problems Javadoc Declaration Console
consoles to display at this time.
191M of 256M
Search the web and Windows
11:08 AM
6/24/2020


Properties for corejava
type filter text
Resource
Builders
Coverage
Java Build Path
Java Code Style
Java Compiler
Java Editor
Javadoc Location
Project Facets
Project Natures
Project References
Run/Debug Settings
Task Repository
Task Tags
Validation
WikiText
Java Build Path
Source
Projects
Libraries
Order and Export
Module Dependencies
JARs and class folders on the build path:
Modulepath
JRE System Library [JavaSE-11]
Classpath
Add JARs...
Add External JARs...
Add Variable...
Add Library...
Add Class Folder...
Add External Class Folder...
Edit...
Remove
Migrate JAR File...
Apply
Apply and Close
Cancel