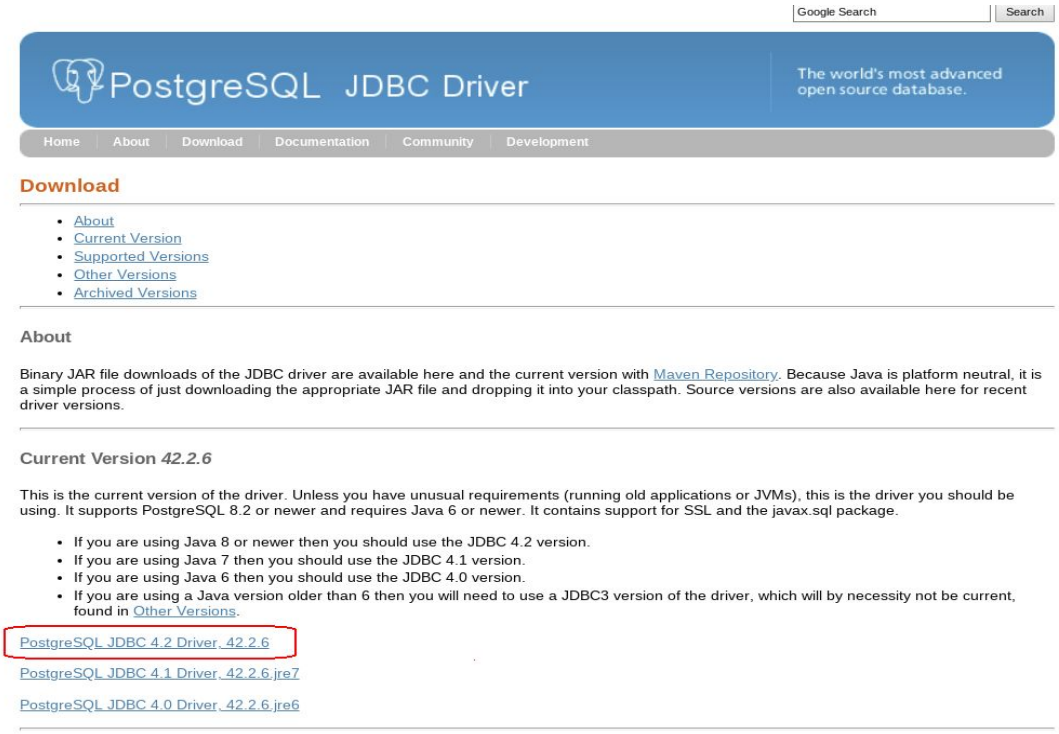
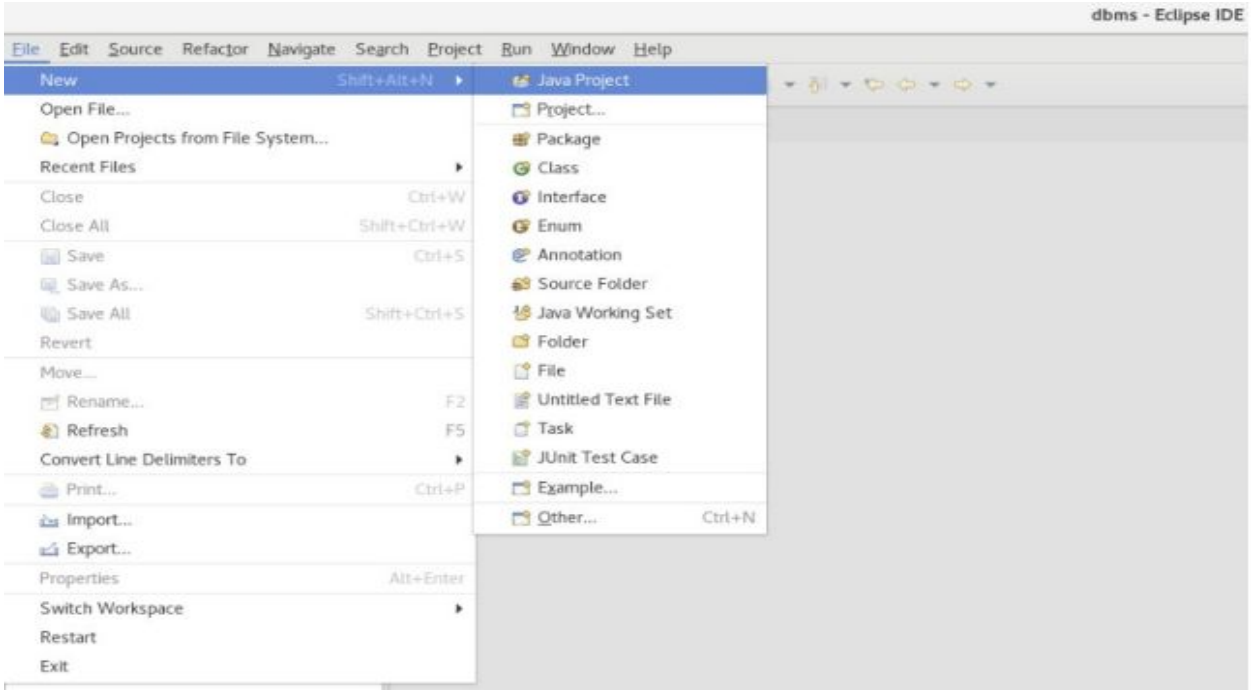
## POSTGRESQL JDBC CONNECTION EXAMPLE

1. Visit http://jdbc.postgresql.org/download.html to download the latest PostgreSQL JDBC Driver.




2. Open Eclipse and click File -> New-> Java project. Give proper project name and click on Finish.



New Java Project

**Create a Java Project**

Create a Java project in the workspace or in an external location.

Project name: postgresConnectionExample

☑ Use default location

Location: /home/sysad/workspace/postgresConnectionExample Browse...

JRE

○ Use an execution environment JRE: OSGi/Minimum-1.2

○ Use a project specific JRE: java-8-openjdk-amd64

◉ Use default JRE (currently 'java-8-openjdk-amd64') Configure JREs...

Project layout

○ Use project folder as root for sources and class files

◉ Create separate folders for sources and class files Configure default...

Working sets

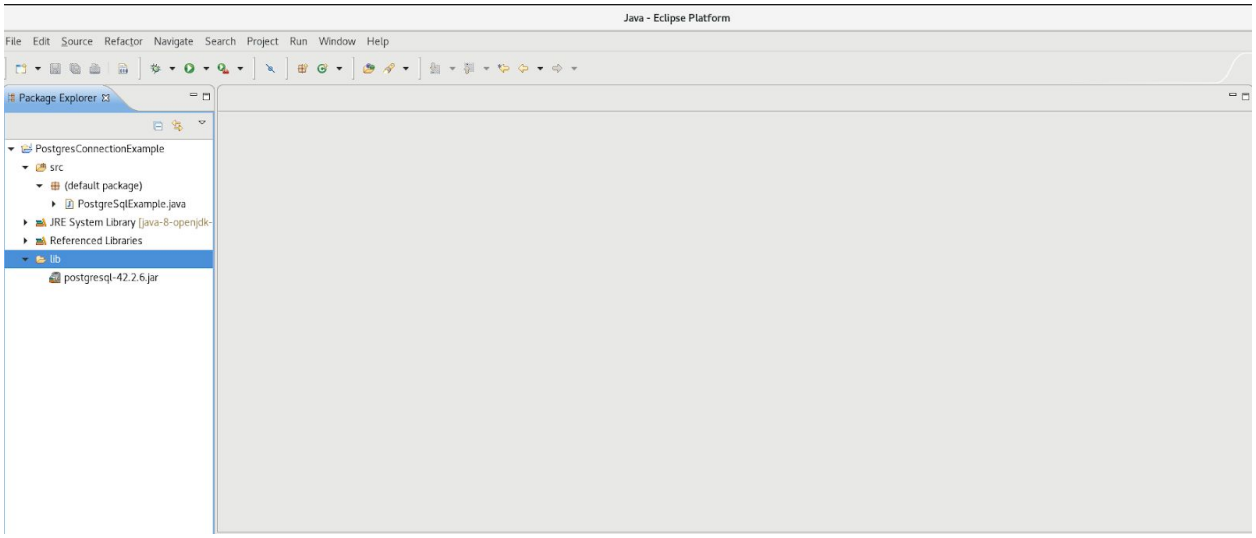
☐ Add project to working sets

Working sets: Select...

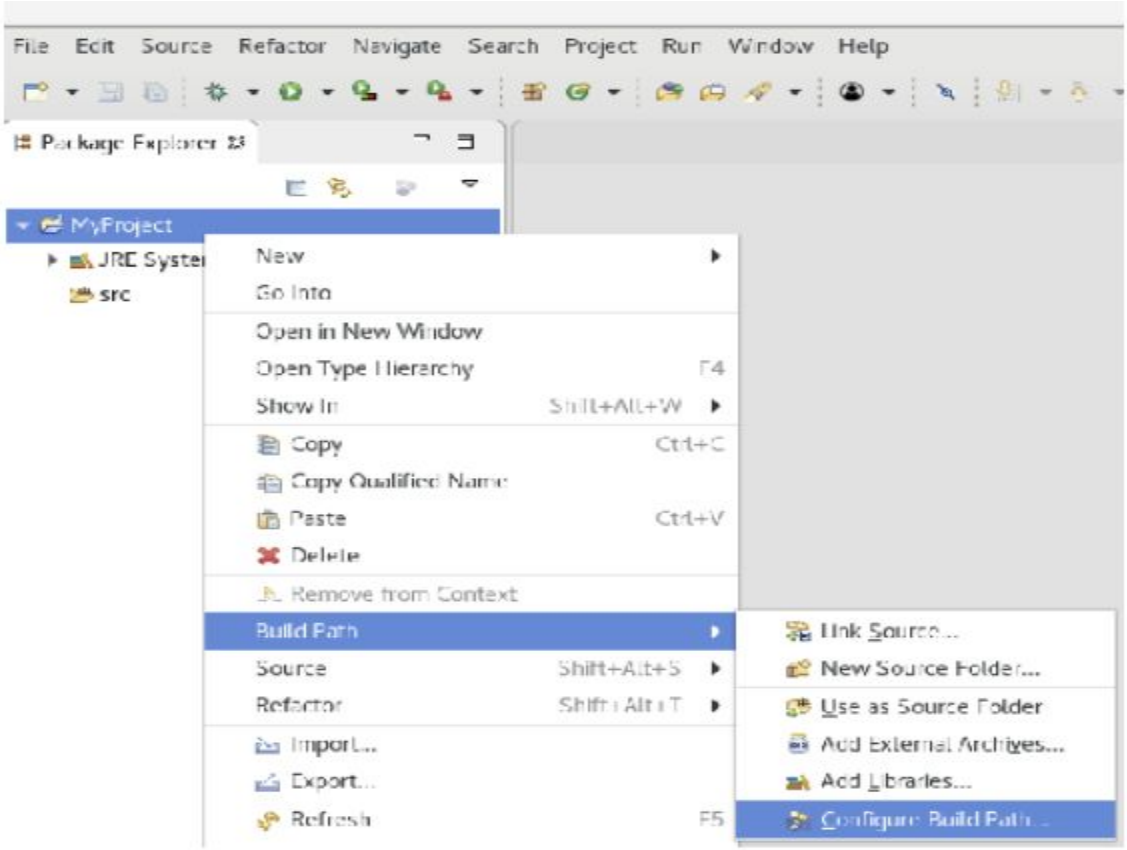
⚠ The current workspace uses a 1.4 JRE with compiler compliance level 1.7. This is not recommended and either the JRE or the compiler compliance level should be changed. Configure...

< Back Next > Cancel Finish

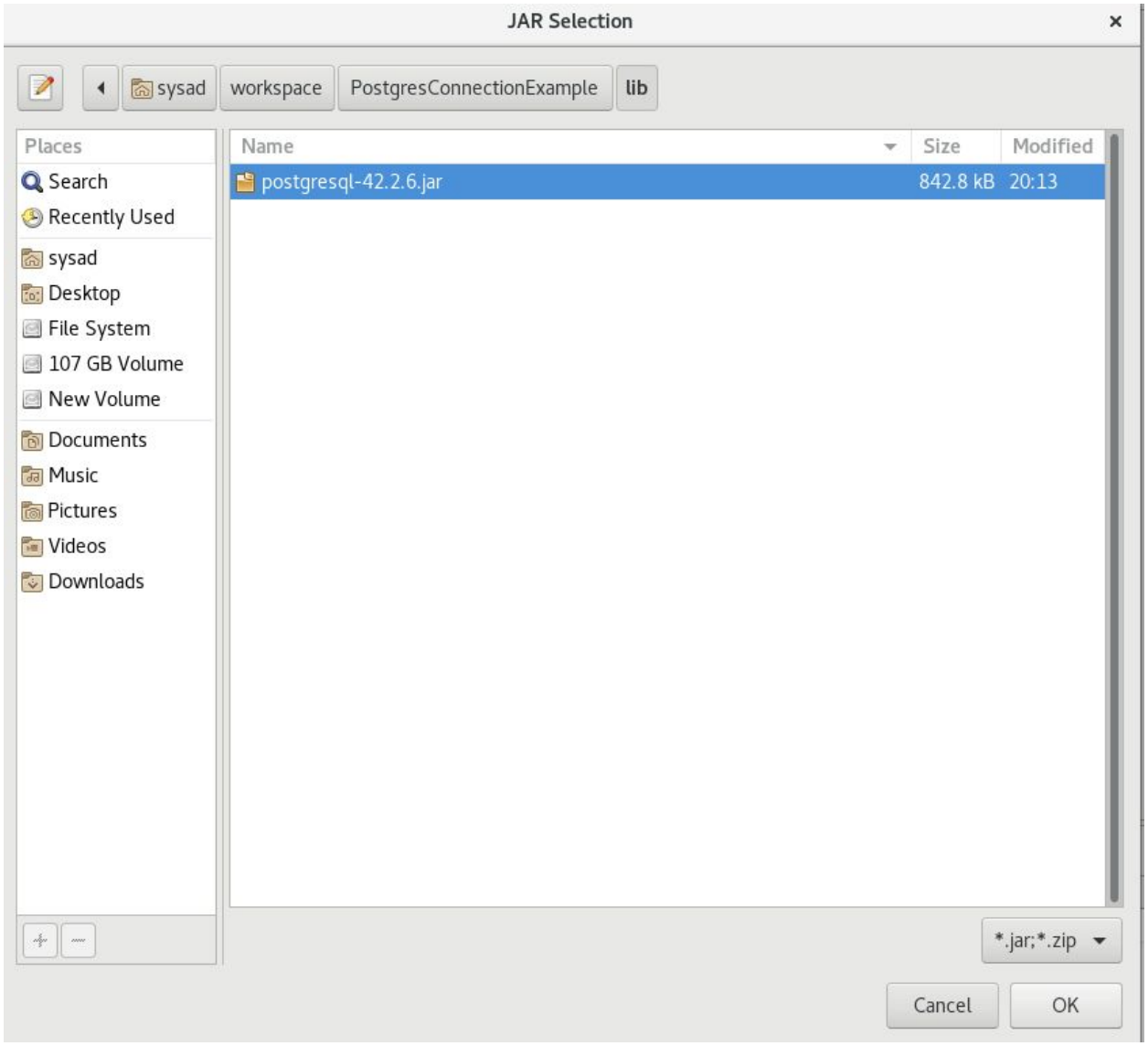
3. Create a folder lib under your project directory and add the postgresql jdbc driver jar file to that.




4. Right click on your project in the Project Explorer, then Build Path and Configure Build Path




5.. Goto the libraries tab --> Add external Jar --> postgresql.jar file. Click apply and close.




6. Right click on your project in the Project Explorer, then File -> Class and create the java class - PostgreSqlExample.java

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

public class PostgreSqlExample {
    public static void main(String[] args) {
```

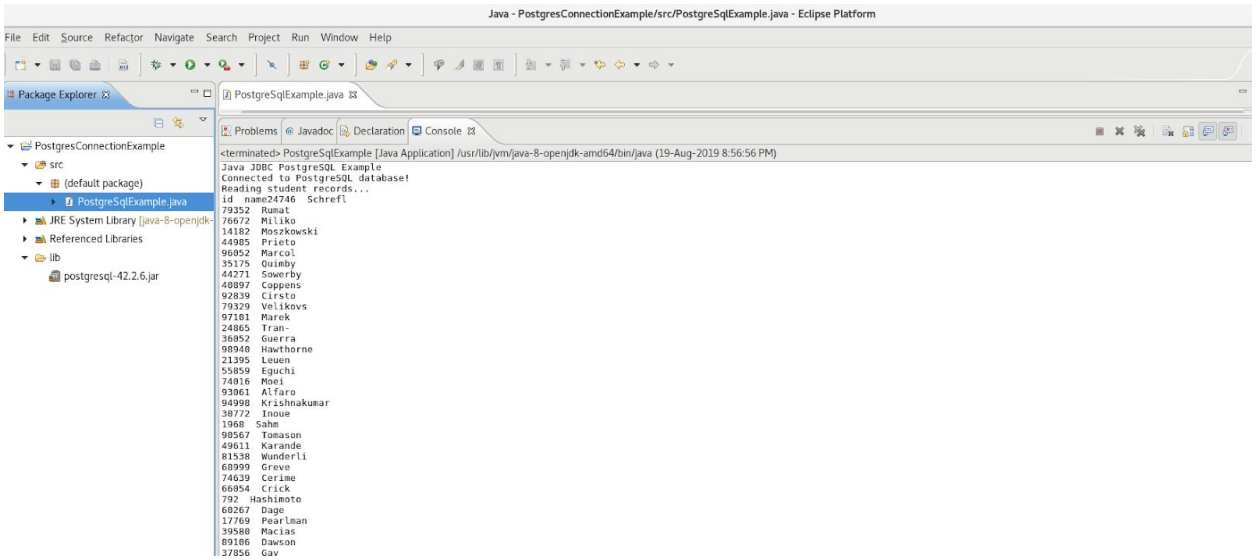
```
        try (Connection connection =
DriverManager.getConnection("jdbc:postgresql://localhost:5432/university", "postgres",
"123456")) {

            System.out.println("Java JDBC PostgreSQL Example");
            System.out.println("Connected to PostgreSQL database!");
            Statement statement = connection.createStatement();
            System.out.println("Reading student records...");
            System.out.printf("%s  %s", "id", "name");
            ResultSet resultSet = statement.executeQuery("SELECT * FROM student");
            while (resultSet.next()) {
                System.out.printf("%s  %s\n", resultSet.getString("id"), resultSet.getString("name"));
            }

        }  catch (SQLException e) {
            System.out.println("Connection failure.");
            e.printStackTrace();
        }
    }
}
```

7. Run the java code.

Output:




**NOTE :**

JDBC database URL for PostgreSQL

The syntax of database URL for PostgreSQL looks like the following forms:

**jdbc:postgresql:database**

**jdbc:postgresql://host/database**

**jdbc:postgresql://host:port/database**

**jdbc:postgresql://host:port/database?param1=value1 & param2=val2& …**

Where:

- host: hostname or IP address of the machine on which PostgreSQL server is running. If omitted, default is localhost.
- port: port number on which the server is listening. If omitted, default is 5432.
- database: name of the database to connect to.
- param1=value1&param2=value2&…: specify additional connection parameters in pairs of key=value form. Most common parameters are user and password.

Here are some examples:

- Connect to the database ProductDB on localhost:

  **jdbc:postgresql:ProductDB**

- Connect to a remote PostgreSQL server on the host dbserver:

  **jdbc:postgresql:dbserver:ProductDB**

- Using host name and port number explicitly:

  **jdbc:postgresql:dbserver:5432:ProductDB**

- Specify username and password for the connection:

  **jdbc:postgresql:ProductDB&user=root&password=secret**