CSI 2132 Lab 6

JDBC Installation and Use

Presented By: Rana Khalil

Outline

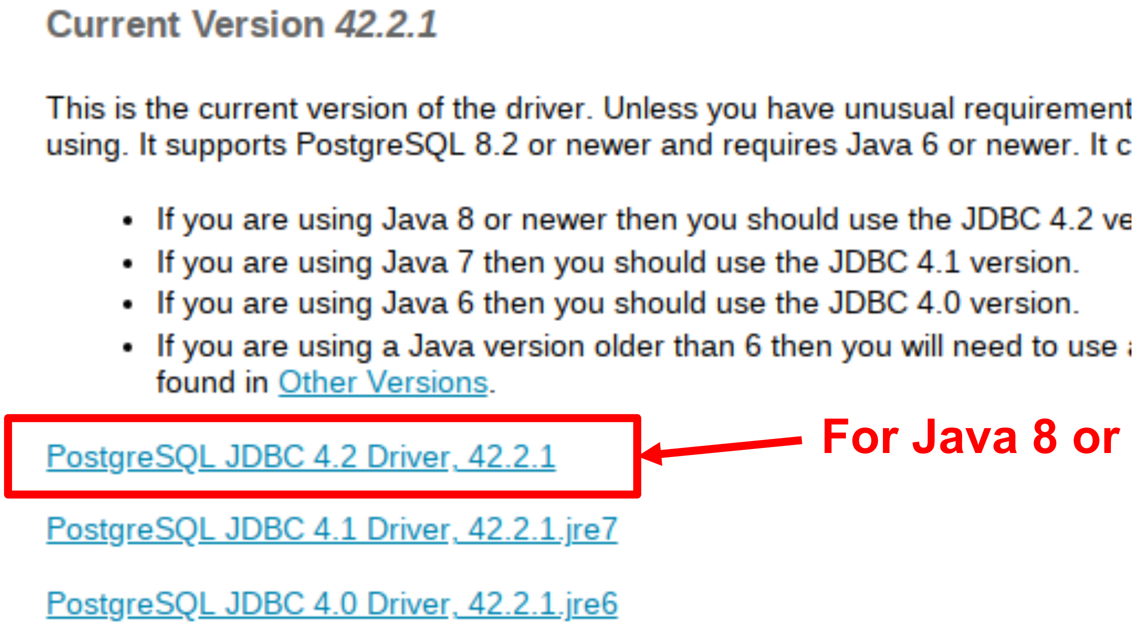
- What is JDBC?
- JDBC Driver (download and use)
- Java programming with JDBC
- Dynamic SQL queries with Java

What is JDBC?

- Java Database Connectivity (JDBC)
- It is an API by Sun Microsystems to allow Java programmers to access SQL databases
- Available since JDK 1.1
- JDBC is an API not a library. It needs to be implemented (as drivers) for a particular DB. i.e. PostgreSQL and MySQL have different JDBC drivers
- In this course we use PostgreSQL so we download PostgreSQL JDBC driver

Download JDBC Driver

- Download JDBC from: https://jdbc.postgresql.org/download.html



Current Version *42.2.1*

This is the current version of the driver. Unless you have unusual requirement using. It supports PostgreSQL 8.2 or newer and requires Java 6 or newer. It c

- If you are using Java 8 or newer then you should use the JDBC 4.2 ve
- If you are using Java 7 then you should use the JDBC 4.1 version.
- If you are using Java 6 then you should use the JDBC 4.0 version.
- If you are using a Java version older than 6 then you will need to use a found in Other Versions.

PostgreSQL JDBC 4.2 Driver, 42.2.1

For Java 8 or newer

PostgreSQL JDBC 4.1 Driver, 42.2.1.jre7

PostgreSQL JDBC 4.0 Driver, 42.2.1.jre6

JDBC Enabled Project

1. Haven’t installed JDK

   Download Netbeans JDK bundle:

   http://www.oracle.com/technetwork/java/javase/downloads/jdk-netbeans-jsp-142931.html

   Have installed JDK, some suggestions:

   1). NetBeans

   https://netbeans.org/downloads/

   2). Eclipse J2EE:

   http://www.eclipse.org/downloads/packages/eclipse-ide-java-ee-developers/oxygen2

   3). IntelliJ

   https://www.jetbrains.com/idea/download/#section=windows

JDBC Enabled Project in Eclipse

2. Create a new Java Project in Eclipse

- File > New > Java Project > Set Project Name and location > Finish

JDBC Enabled Project in eclipse

3. Add PostgreSQL JDBC library

- *Right click on your Project > Select Properties > Java Build Path > Libraries > Add External JARs*
- *Add the downloaded* postgresql-42.2.1.jar *file*

4. Create a new class inside the package with a static main method

5. Write a try – catch structure inside the main method with a generic exception handler

JDBC Coding in Java

1. Import the JDBC driver
2. Load the driver
3. Connect to a Database
4. Issue a Query and process the result

JDBC Coding in Java

1. Import the JDBC driver

- import java.sql.*;

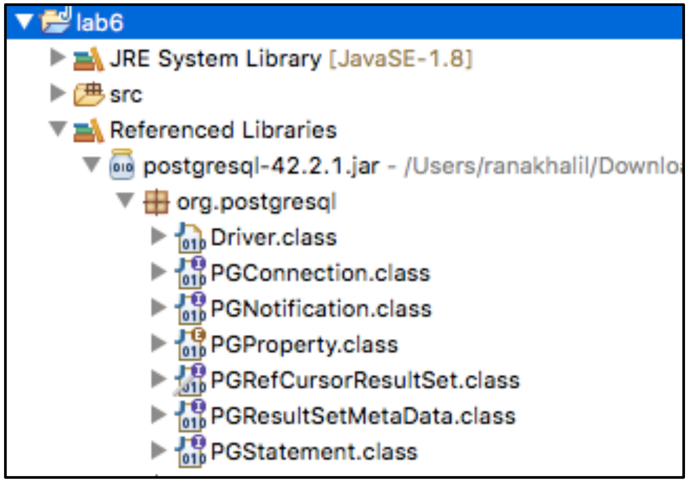
  Note: Do not import the org.postgresql package. If you do, your source will not compile, as javac will get confused.

- Remember the import lines go after the package line

JDBC Coding in Java

2. Load the driver

- Class.forName("org.postgresql.Driver");
- You can check that this class actually exist under: Referenced Libraries > *postgresql-42.2.1.jar > org.postgresql > Driver.class*



```
▼ lab6
    ▶ JRE System Library [JavaSE-1.8]
    ▶ src
    ▼ Referenced Libraries
        ▼ postgresql-42.2.1.jar - /Users/ranakhalil/Downlo
            ▼ org.postgresql
                ▶ Driver.class
                ▶ PGConnection.class
                ▶ PGNotification.class
                ▶ PGProperty.class
                ▶ PGRefCursorResultSet.class
                ▶ PGResultSetMetaData.class
                ▶ PGStatement.class
```

JDBC Coding in Java

3. Connect to a Database

```
Connection db = DriverManager.getConnection(url,
username, password);
```

Where:

- url is in the form of:

  1)jdbc:postgresql://host:port/database?currentSchema=XXX

  Example:

  jdbc:postgresql://www.eecs.uottawa.ca:15432/svale054?currentSchema=labolatories

- username: your SITE username (svale054)
- password: your SITE password (XXXXXX)

JDBC Coding in Java

4. Issue a Query and process the result

```
Statement st = db.createStatement();
ResultSet rs = st.executeQuery("SELECT * FROM artist");
while (rs.next()) {
      System.out.print("Column 1 returned: ");
      System.out.println(rs.getString(1));
}
rs.close();
st.close();
```

Example

Write a Java program (FirstExcercise.java) that connects to our own database and retrieves the name and birthday of all artists. Print the result as a 2D table using System.out.print

Dynamic Queries

```
String field = “aname, Style";
String cond = “aname";
String table = “artist";
String value = “Caravaggio";
Statement st = db.createStatement();
ResultSet rs = st.executeQuery("SELECT " + field + " FROM
 " + table + " WHERE " + cond + " = ‘"+ value + "’;");
```

Dynamic Queries

To get number of columns returned by the query:

```
ResultSetMetaData rsMetaData = rs.getMetaData();
int numberOfColumns=rsMetaData.getColumnCount();
```

Your Turn

- Write Java code (SecondExcercise.java) that returns those fields of table Artist that are in an array named fields: String[] fields = {“AName”, “Style”, ...};
- Allow your program to retrieve information from more than one artist (hint: use keyword IN).
- Try changing the fields array and recompile and run your code. It should work for all valid fields.

REFERENCE

http://jdbc.postgresql.org/documentation/83/index.html