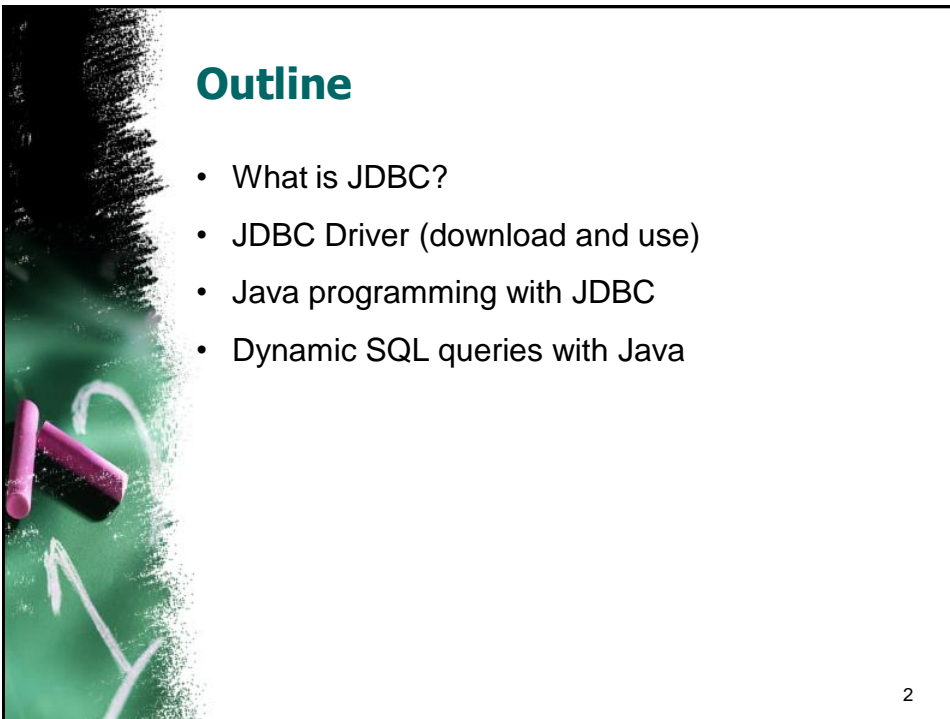




**CSI 2132 Lab 6**

JDBC Installation and Use

1



**Outline**

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

2

## 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:  
<http://jdbc.postgresql.org/download.html>
- *JDBC4 PostgreSQL driver– version 9.0-801*
- *It is class folder*

### Current Version

This is the current version of the driver. Unless newer and requires a 1.4 or newer JVM. It should use the JDBC4 version.

[JDBC3 Postgresql Driver, Version 9.0-801](#)

[JDBC4 Postgresql Driver, Version 9.0-801](#)

### Other Versions

Many other versions of the JDBC driver are available.

To determine JDK/JVM compatibility this follows:



## JDBC Enabled Project in eclipse

1. Create a new Java Project in eclipse (jdbc)
2. From the Project Properties, click on
  - *Java Build Path > Libraries > Add External JARs*
3. Select the downloaded JAR file
4. Create a new package in your project (code)
5. Create a new class inside the package with a static main method (connection.java)
6. 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.*;`
  - *It is NOT appropriate to import org.postgresql directly*
  - *Remember the import lines go after the package line*
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
2. Load the driver
  - `Class.forName("org.postgresql.Driver");`
  - *You can check that this class actually exist under Referenced Libraries > postgresql-9.0-801.jdbc4 > org.postgresql > Driver.class*
3. Connect to a Database
4. Issue a Query and process the result

## JDBC Coding in Java

1. Import the JDBC driver
2. Load the driver
3. Connect to a Database
  - `Connection db = DriverManager.getConnection(url, username, password);`
  - *URL is in the form of:*
    - `jdbc:postgresql:database`
    - `jdbc:postgresql://host/database`
    - `jdbc:postgresql://host:port/database`
    - `jdbc:postgresql://web0.site.uottawa.ca:15432/svale054`
  - *Username: your SITE username (svale054)*
  - *Password: your SITE password (XXXXXX)*
4. Issue a Query and process the result

## 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
 

```
Statement st = db.createStatement();
ResultSet rs = st.executeQuery("SELECT * FROM
laboratories.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 = "laboratories.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>