JDBC

Establishing Connection

Establishing Connection

Agenda

1 **Establishing Connection**

Objectives

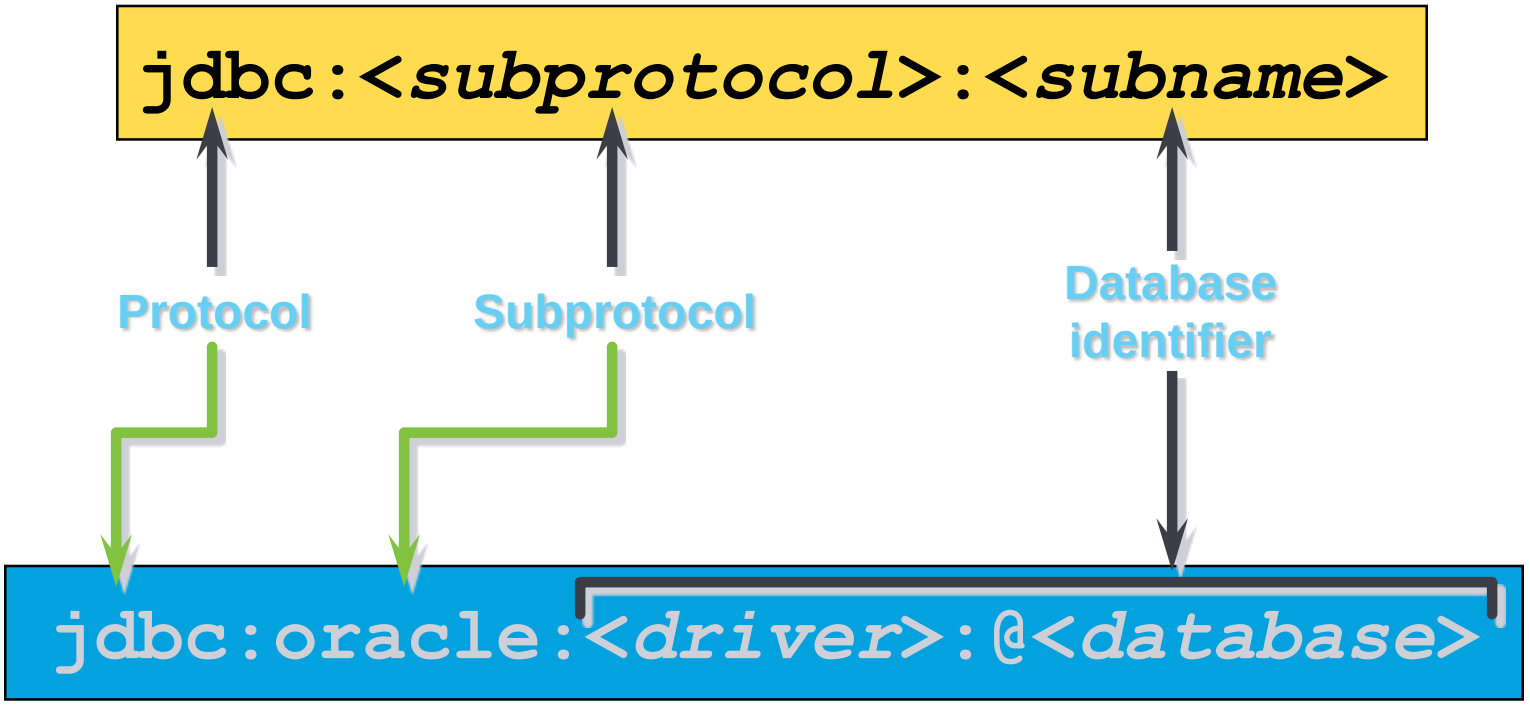
At the end of this module, you will be able to:

- Explain how to connect to a database using Java Database Connectivity (JDBC).

4

Connect: About JDBC URL

URL represents a protocol to connect to the database



Connect: About JDBC URL

- A JDBC driver uses a JDBC URL to identify and connect to a particular database. These URLs are generally of the form:

  jdbc:<subprotocol>:<subname> or
  jdbc:*driver*:*databasename*

- The actual standard is quite fluid, however, as different databases require different information to connect successfully. For example, the Oracle JDBC-Thin driver uses a URL of the form:

  jdbc:oracle:thin:@*site*:*port*:*database*

- while the JDBC-ODBC Bridge uses:

  jdbc:odbc:*datasource:odbcoptions*

- The only requirement is that a driver be able to recognize its own URLs.

Host Details

You should be aware of the IP address of the system or the host name, from where Oracle Database is being accessed.

If your Oracle database is running on the same system, where you are executing your jdbc program, then you can use @localhost (in place of the IP address), in your JDBC URL

jdbc:oracle:thin:@localhost

If the Oracle database is running on a different system, then you need to find out the IP address of that system and include @IP Address in your JDBC URL.

For e.g. if the IP address of the system hosting Oracle database is 192.168.10.5, then your JDBC URL, will begin with

jdbc:oracle:thin:@192.168.10.5

Service Name and Port No.

You should be aware of the service name and port no. on which oracle service is running

How to find the service name and the port no. of the database service ?

8

Service Name and Port No.

- You can find the service name and the port no. of the database that you want to connect, by opening a file called tnsnames.ora

- This file will be found within one of the subdirectories of Oracle installation directory. You will have to look for a folder called app, which contains files and folders related to Oracle Software. In one of the sub-directories in this hierarchy, ADMIN, you will find tnsnames.ora file.
- For e.g. in my machine, tnsnames.ora is found within the following path :
- **E:\app\harb\product\11.1.0\db_1\NETWORK\ADMIN**
- *Note : Please do not change any configuration parameters stored within this file. Even a small change in this file could disrupt the Oracle Service. Just open the file, note down the configuration parameters for port no. and service_name and close the file without saving it.*

tnsnames.ora file

A typical tnsnames.ora file :

```
# tnsnames.ora Network Configuration File:
E:\app\harb\product\11.1.0\db_1\network\admin\tnsnames.ora
# Generated by Oracle configuration tools.

ORCL =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = L-156012383)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = orcl)
    )
  )
```

As you can observe, the service name is **orcl** *and the port no. is* **1521**

JDBC URLs: Examples

- To connect to oracle using thin driver provided by Oracle

```
jdbc:oracle:thin:@192.168.49.107:1521:orcl
```


11

How to make the Connection?

1. To register the driver is to send the driver class name as parameter for Class.forName() method

```
Class c = Class.forName("oracle.jdbc.driver.OracleDriver");
Class c = Class.forName(”sun.jdbc.odbc.JdbcOdbcDriver");
```

2. To connect to a database use getConnection() method

```
Connection
conn=DriverManager.getConnection(URL,userid,password);
```

```
Connection conn = DriverManager.getConnection
("jdbc:oracle:thin:@myhost:1521:orcl", "scott", "tiger");
```


12

How to make the Connection?

- Loading the driver or drivers you want to use is very simple and involves just one line of code. If, for example, you want to use the JDBC-ODBC Bridge driver, the following code will load it:
- Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
- Your driver documentation will give you the class name to use. Class.forName will automatically register the driver with the DriverManager.
- When you have loaded a driver, it is available for making a connection with a DBMS.

How to make the Connection?

- The java.sql.Connection object, which encapsulates a single connection to a particular database, forms the basis of all JDBC data-handling code. The DriverManager.getConnection( ) method creates a connection:

Connection con = DriverManager.getConnection("*url*", "*user*", "*password*");

- You pass three arguments to getConnection( ): a JDBC URL, a database username, and a password. For databases that don't require explicit logins, the user and password strings should be left blank. When the method is called, the DriverManager queries each registered driver, asking if it understands the URL. If a driver recognizes the URL, it returns a Connection object. Because the getConnection( ) method checks each driver in turn, you should avoid loading more drivers than are necessary for your application.

Quiz

1. **Which one of the following statements is used to register the jdbc driver**
   a) Claas.Forname(“oracle.jdbc.driver.OracleDriver”);
   b) Class.forName(“oracle.jdbc.driver.OracleDriver”);
   c) class.forname(“oracle.jdbc.driver.OracleDriver”);
   d) Class.ForName(“oracle.jdbc.driver.OracleDriver”);

2. **Which is the correct syntax for creating the Connection object**
   a) Connection.getConnection(URL, username, password);
   b) Connection.createConnection(URL, username, password);
   c) DriverManager.createConnection(URL, username, password);
   d) DriverManager.getConnection(URL, username, password);

Answers :
1 : b
2 : d

15

Summary

In this module, you were able to:

- Explain how to connect to a database using Java Database Connectivity (JDBC)

Thank You