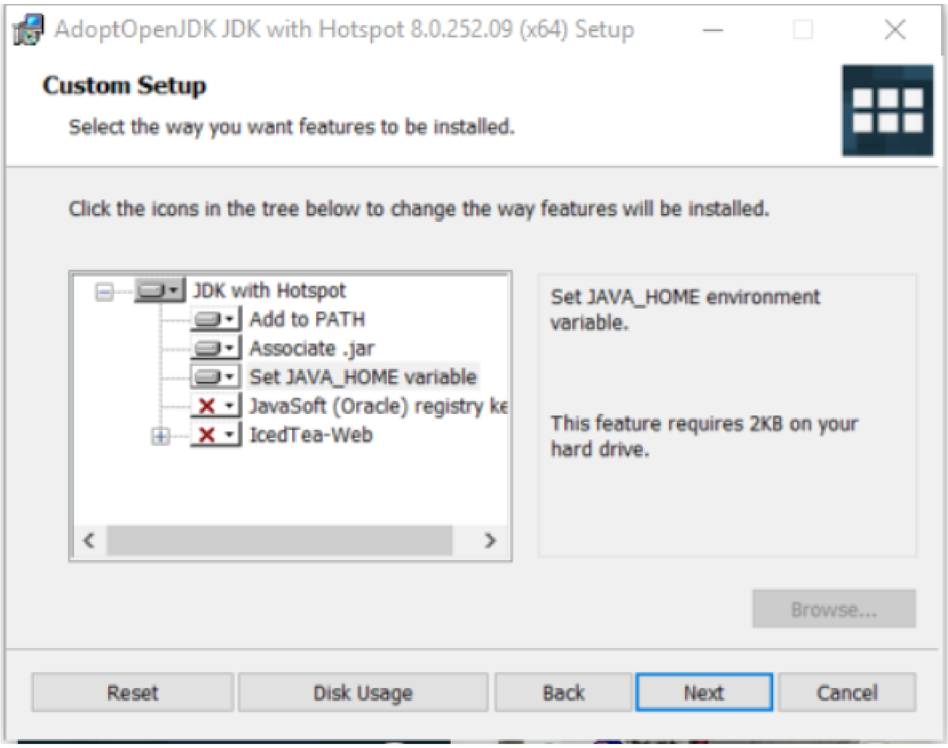
SOFTENG 351 S1 C : Tutorial 8

Due Date: Sunday 24 May 2020 at 11:59pm
10 marks in total = 1% of the finalgrade

This lab practice aims to build up a java environment with the JDBC drive that can connect to the student database.

1 Environment Setting

1. If you have not installed java, please
   - Download OpenJDK8 (LTS) from OpenJDK community `https://adoptopenjdk.net/index.html?variant=openjdk8&jvmVariant=hotspot`.
   - Install the .msi file with SET JAVA_HOME variable set:



AdoptOpenJDK JDK with Hotspot 8.0.252.09 (x64) Setup

Custom Setup

Select the way you want features to be installed.

Click the icons in the tree below to change the way features will be installed.

- JDK with Hotspot
  - Add to PATH
  - Associate .jar
  - Set JAVA_HOME variable
  - ✗ JavaSoft (Oracle) registry ke
  - ✗ IcedTea-Web

Set JAVA_HOME environment variable.

This feature requires 2KB on your hard drive.

2. Install the JDBC driver of mysql by copying the file mysql-connector-java-5.1.48.jar (included in the package) to $JAVA_HOME/jre/lib/ext
   - Another way of using mysql jar, in case things don't work:

```
C:\> java -cp <the location of the jar file>; <the location of your main class>
Example:
C:\> java -cp C:\\city\\windows\\mysql-connector-java-5.1.48.jar; mainProgram
```

3. Due to internal policies, one will not be able to access our MySQL server from home using JDBC, **unless** an SSH tunnel is established. The instruction of construct the SSH tunnel is attached in Section 1.1.

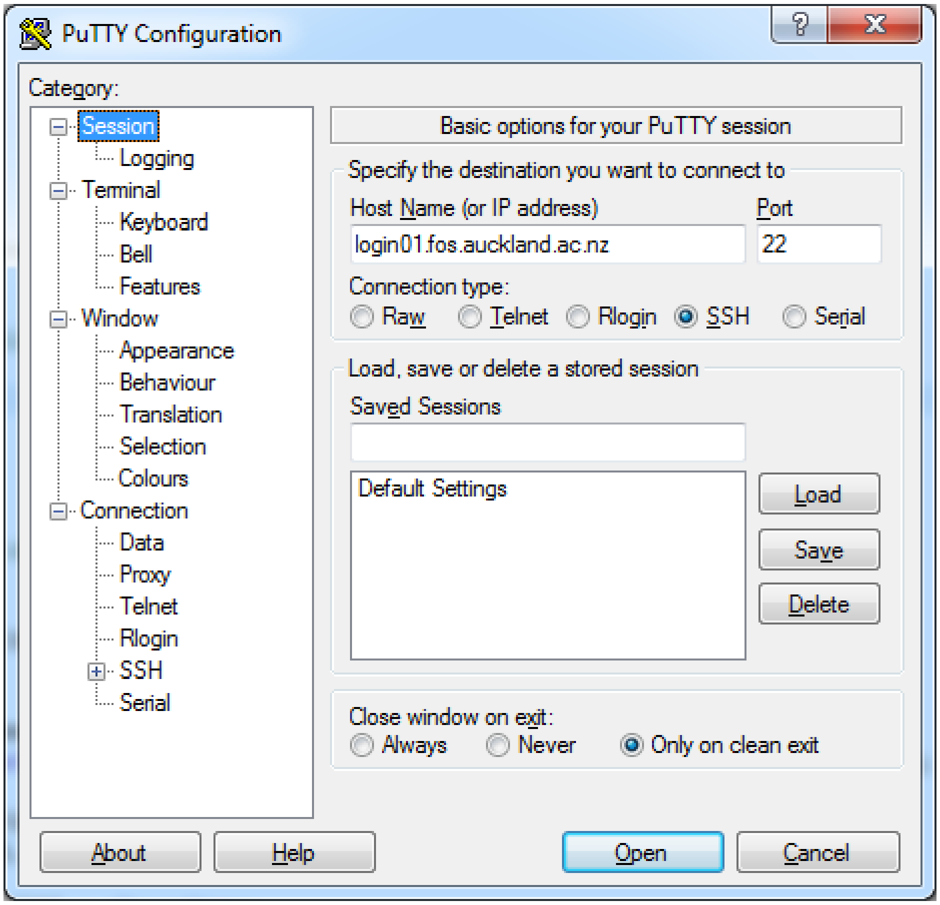
1.1 Establish SSH Tunnel to Connect to the DB

Step 1 – Specify PuTTy Session

On Windows, get Putty from https://www.putty.org/.

On Linux OS, PuTTy can be install by running: *sudo apt-get install putty* Or *sudo yum install putty*.

On Mac OS, PuTTy can be install by running: *sudo brew install putty*



PuTTY Configuration

Basic options for your PuTTY session

Specify the destination you want to connect to

Host Name (or IP address): login01.fos.auckland.ac.nz

Port: 22

Connection type: ○ Raw ○ Telnet ○ Rlogin ● SSH ○ Serial

Load, save or delete a stored session

Saved Sessions

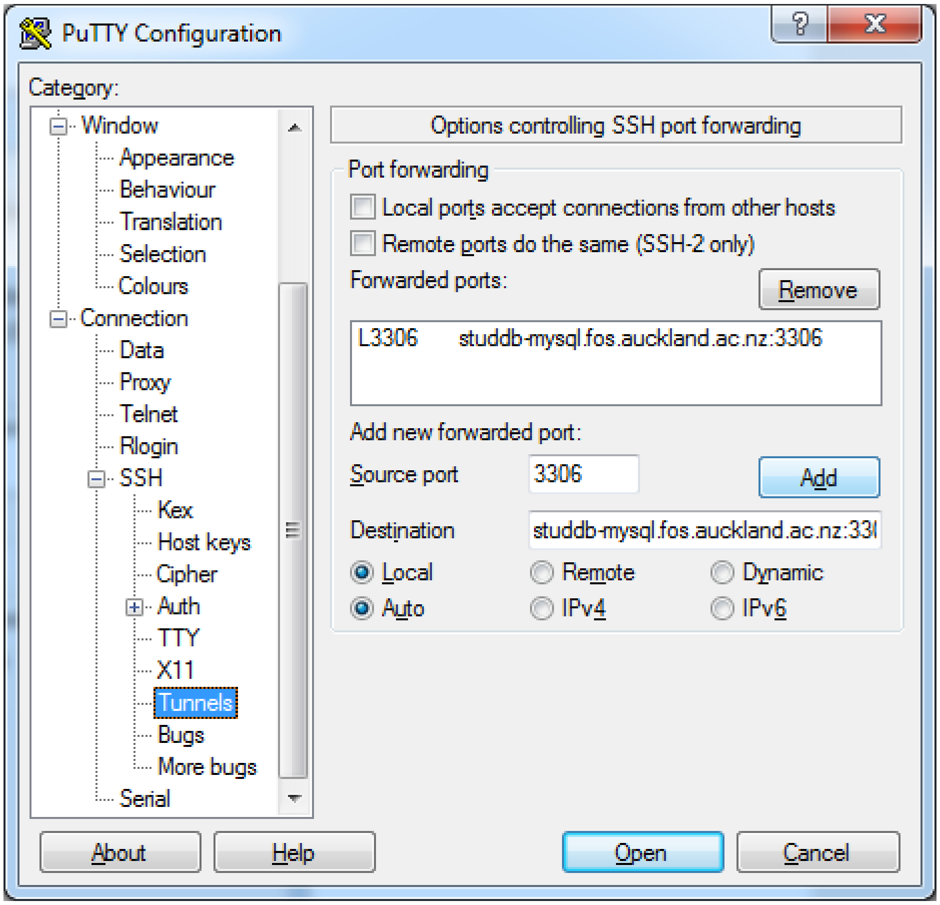
Default Settings

Close window on exit: ○ Always ○ Never ● Only on clean exit

For Science Students, set the Host Name to *login01.fos.auckland.ac.nz*, and Port to 22.

For Engineering Students set the Host Name to *shell.foe.auckland.ac.nz*, and port to 22.

Step 2 – Setup forwarded port



Options controlling SSH port forwarding

Port forwarding

□ Local ports accept connections from other hosts

□ Remote ports do the same (SSH-2 only)

Forwarded ports:

```
L3306      studdb-mysql.fos.auckland.ac.nz:3306
```

Add new forwarded port:

Source port 3306

Destination studdb-mysql.fos.auckland.ac.nz:33

● Local ○ Remote ○ Dynamic

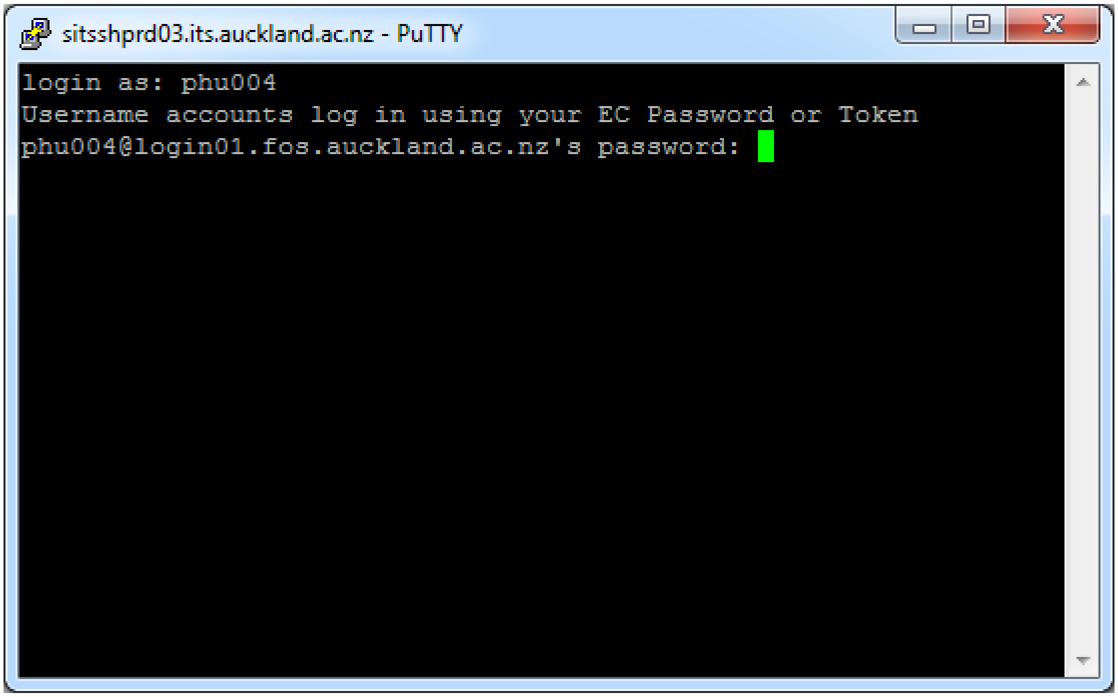
● Auto ○ IPv4 ○ IPv6

In the Tunnels section, set Source port to 3306, and set Destination to the MySQL server:

studdb-mysql.fos.auckland.ac.nz:3306

Once this is done, click on the “Add” button to create the forwarded port, then the “Open” button at the bottom to start the login Session.

Step 3 – Start Login Session



```
login as: phu004
Username accounts log in using your EC Password or Token
phu004@login01.fos.auckland.ac.nz's password:
```

Put your EC user name then password when prompted. Once the Session has been established, keep the session window live as long as you want to connect to the MySQL database server.

Since we have mapped the remote MySQL server port to the local port 3306, The JDBC connection string should look something like:

```
"jdbc:mysql://127.0.0.1:3306/{database_name}"
```

2 Connect to the Database using JDBC

Please adapt the following code (also included in the package) to connect to the student database.

```
import java.sql.*;


public class connect {
        public static void main(String args[]) {
                String username = " "; //"your UPI";
                String password = " "; //"your password";
                String url = "jdbc:mysql://127.0.0.1:3306/<your database name>"; //e.g. "jdbc:mysql://127.0.0.1:3306/stu_UPI_COMPSCI_351_C_S1_2020"


        //Loads the JDBC driver
                try {
                        Class.forName("com.mysql.jdbc.Driver");
                        System.out.println("Driver loaded");
                 //Establishes a connection
                        Connection conn = DriverManager.getConnection(url, username, password);
                        System.out.println("Database connected");
                 //Creates a statement
                        Statement stmt= conn.createStatement();
                 //Executes a statement
                        String command = "SELECT * " + "FROM DEPARTMENT";
                 //Obtains the results as a set of rows
                        System.out.println(command);
                        ResultSet result = stmt.executeQuery(command);

                  //Obtains the metadata associated with the results

                        ResultSetMetaData  metaData = result.getMetaData();
                        //Obtains the number of columns
                        int columnCount = metaData.getColumnCount();
                        System.out.println(columnCount);
                          //Prints the names of the columns obtained from the metadata
                        for (int i=1; i<=columnCount; i++) {
                                if (i > 1) System.out.print('\t');
                                System.out.print(metaData.getColumnLabel(i));
                        }
                        System.out.println();
                        System.out.println("-----------------------------------------------------");

            //Iterates through the results and prints the tuples (rows)
                        while (result.next()) {
                                for (int i=1; i<= columnCount; i++) {
                                        if (i>1) System.out.print('\t');
                                        System.out.print(result.getString(i));
                                }
                                System.out.println();
                        }
                                                         //closes the connection  (optional)
                        conn.close();
                }
                catch (Exception e) {
                        e.printStackTrace();
                }
        }
}
```

Submit the running result of the above code. [10 marks]