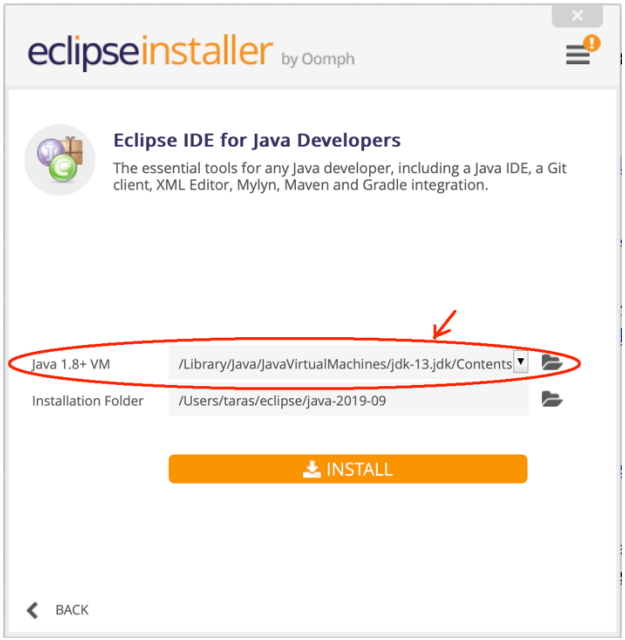
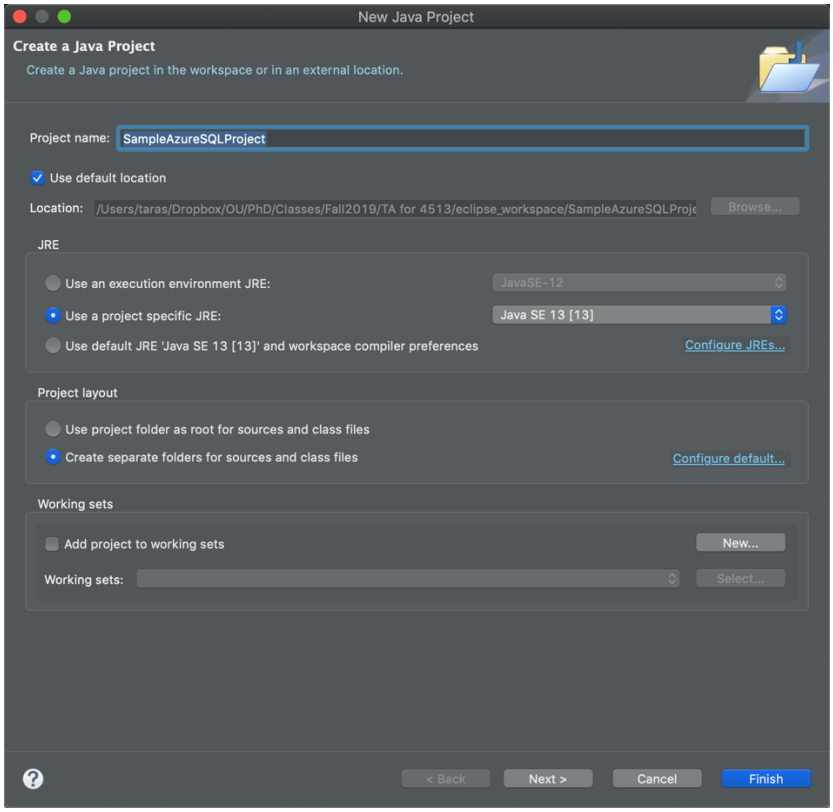
CS/DSA 4513 Fall 2019
Using Java and JDBC to Connect to the Azure SQL Database

1. **Download & Install Java SDK v13**
    - Download JDK v13 from the below link
        - https://www.oracle.com/technetwork/java/javase/downloads/jdk13-downloads-5672538.html
    - Follow the installation instructions at the below link
        - https://docs.oracle.com/en/java/javase/13/install/overview-jdk-installation.html
2. **Download & Install Eclipse IDE for Java Developers**
    - Please follow the instructions at the below link (also see the note below)
        - http://www.eclipse.org/downloads/packages/installer
    - Note, if you asked about Java VM, make sure to select the version 13 which you just installed (see the screenshot below)

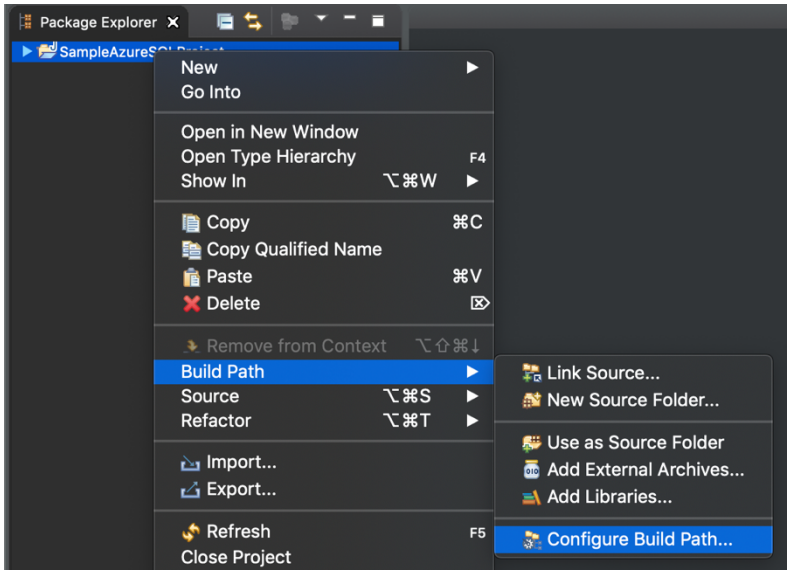


3. **Download & Install Microsoft JDBC Driver 7.4**
    - Download Microsoft JDBC Driver 7.4 for SQL Server from the below link
        - https://www.microsoft.com/en-us/download/details.aspx?id=58505
    - Follow the Install Instructions on the same page. Make sure to note your installation directory.
4. **Connect to Azure SQL DB using Java**
    - Launch Eclipse IDE you just installed
        - We recommend creating a new workspace directory for this class when prompted
    - Close the "Welcome" window if you see it.
    - Click on "File" > "New" > "Java Project"

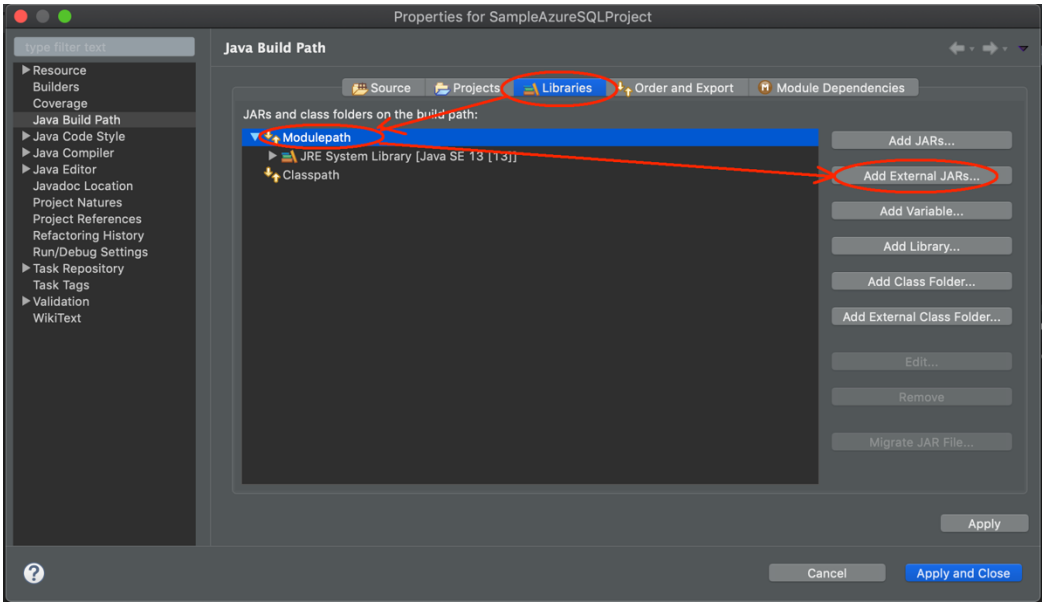
- When presented with the “New Java Project” Window
  - Give the project name (“*SampleAzureSQLProject”* for example)
  - Select “Use project specific JRE” and “Java SE 13”
  - Your window should look like the screenshot below
  - Click on “Finish”. Choose “Don’t create” when asked about creating module-info.java



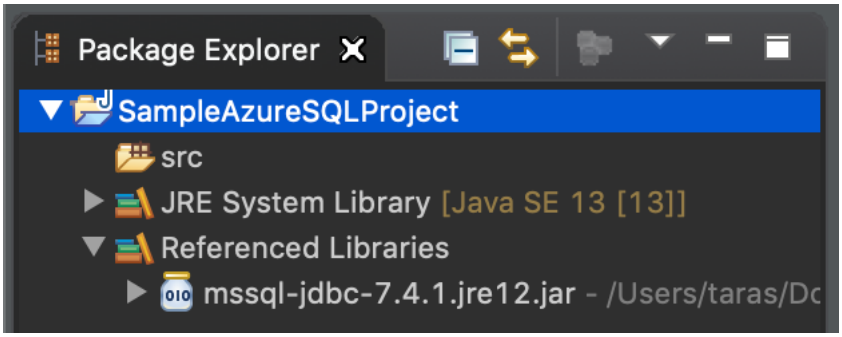
- Now, right-click on the project “SampleAzureSQLProject” > “Build Path” > “Configure Build Path” (see the screenshot below)



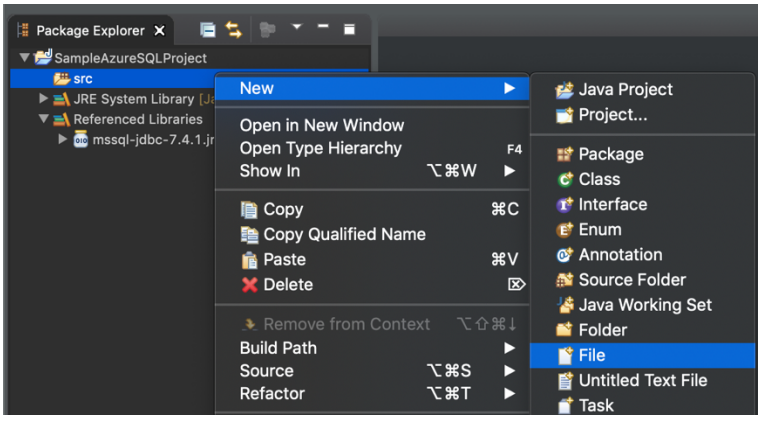
- Click on “Libraries tab”, click on “Modulepath” and “Add External JARs…” and select the “mssql-jdbc-7.4.1.jre12.jar” file available in the installation directory of the JDBC driver (we asked you to note it earlier).



- Click on “Apply and Close”.
- You should now see that JAR file added into referenced libraries of your project.

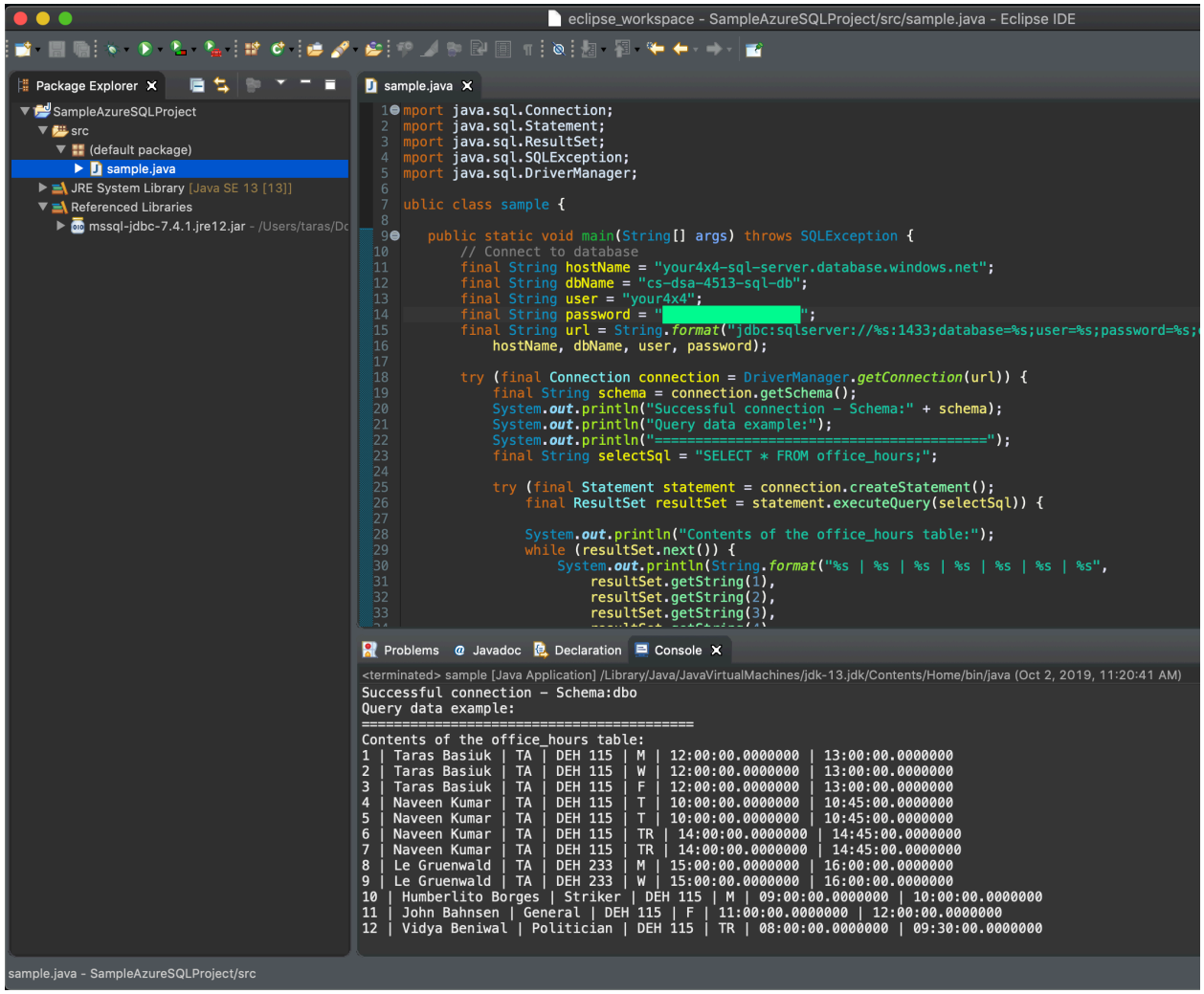


- Create a new java file. Right click on “src” folder > “New” > “File”



- Name it as “*sample.java*” and click “Finish” button.
- Once the new file is opened in a tab copy the code from Appendix B below into it.

- Replace the values “<your4x4>” and “<your password>” in the java program with your SQL server admin username and password. Make sure the hostName and dbName values match your Azure SQL server and database names.
  - FYI: The connection strings (url variable in the sample code) can also be taken from your Azure Dashboard > “cs-dsa-4513-sql-db” > “connection strings” > “JDBC”
- Note, the following step assumes the SQL statements from the Appendix B below were previously executed in your SQL database. If it’s not the case, either execute them now, or modify the select query you want your java program to execute.
- Run your java program by right-clicking on sample.java > “Run As” > “Java Application”. You should see the table data printed out into the console which is retrieved from your Azure SQL database (see the screenshot below).



- You were able to successful connect to your Azure SQL database and should now be ready to work on your Graded Homework #3, good luck!

Helpful links:

1. Microsoft JDBC Driver for Azure SQL Database documentation
    a. https://docs.microsoft.com/en-us/sql/connect/jdbc/microsoft-jdbc-driver-for-sql-server?view=sql-server-2017

Appendix A. Sample Java Program

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

public class sample {

    public static void main(String[] args) throws SQLException {
        // Connect to database
        final String hostName = "<your4x4>-sql-server.database.windows.net";
        final String dbName = "cs-dsa-4513-sql-db";
        final String user = "<your4x4>";
        final String password = "<your password>";
        final String url =
String.format("jdbc:sqlserver://%s:1433;database=%s;user=%s;password=%s;encrypt=true;host
NameInCertificate=*.database.windows.net;loginTimeout=30;",
            hostName, dbName, user, password);

        try (final Connection connection = DriverManager.getConnection(url)) {
            final String schema = connection.getSchema();
            System.out.println("Successful connection - Schema:" + schema);
            System.out.println("Query data example:");
            System.out.println("=========================================");
            final String selectSql = "SELECT * FROM office_hours;";

            try (final Statement statement = connection.createStatement();
                final ResultSet resultSet = statement.executeQuery(selectSql)) {

                System.out.println("Contents of the office_hours table:");
                while (resultSet.next()) {
                    System.out.println(String.format("%s | %s | %s | %s | %s | %s | %s",
                        resultSet.getString(1),
                        resultSet.getString(2),
                        resultSet.getString(3),
                        resultSet.getString(4),
                        resultSet.getString(5),
                        resultSet.getString(6),
```

```
                resultSet.getString(7)));
          }
        }
      }
    }
}
```

Appendix B. Sample SQL statements

```
DROP TABLE office_hours; --Delete the table if it was previously created

--Create the new table for office hours schedule
CREATE TABLE office_hours (
      id INT IDENTITY(1,1) PRIMARY KEY, --IDENTITY(1, 1) auto-increments the id
      faculty_name VARCHAR(64) NOT NULL,
      faculty_position VARCHAR(32) DEFAULT 'TA',
      location VARCHAR(64) NOT NULL,
      weekday VARCHAR(2) NOT NULL,
      start_time TIME NOT NULL,
      end_time TIME NOT NULL,
  CONSTRAINT CHK_non_empty_values CHECK( --Make sure provided strings are not empty
      LEN(faculty_name) > 0 AND
      LEN(location) > 0 AND
      LEN(faculty_position) > 0
  ),
  --Only accept weekday values corresponding to Monday through Friday
  CONSTRAINT CHK_weekday_value CHECK (weekday IN ('M', 'T', 'W', 'TR', 'F')),
  --Only accept start_time and end_time which differ by at least 30 minutes
  CONSTRAINT CHK_time_duration CHECK (DATEDIFF(minute, start_time, end_time) >= 30)
);

--Insert an entry into the table
INSERT INTO office_hours (
  faculty_name, location, weekday, start_time, end_time
) VALUES (
  'Taras Basiuk', 'DEH 115', 'M', '12:00:00', '13:00:00'
);

--Insert multiple records at once
INSERT INTO office_hours
 (faculty_name, location, weekday, start_time, end_time)
VALUES
 ('Taras Basiuk', 'DEH 115', 'W', '12:00:00', '13:00:00'),
 ('Taras Basiuk', 'DEH 115', 'F', '12:00:00', '13:00:00'),
 ('Naveen Kumar', 'DEH 115', 'T', '10:00:00', '10:45:00'),
 ('Naveen Kumar', 'DEH 115', 'T', '10:00:00', '10:45:00'),
 ('Naveen Kumar', 'DEH 115', 'TR', '14:00:00', '14:45:00'),
```

```
 ('Naveen Kumar', 'DEH 115', 'TR', '14:00:00', '14:45:00'),
 ('Le Gruenwald', 'DEH 233', 'M', '15:00:00', '16:00:00'),
 ('Le Gruenwald', 'DEH 233', 'W', '15:00:00', '16:00:00');

--Insert multiple records at once
INSERT INTO office_hours
 (faculty_name, faculty_position, location, weekday, start_time, end_time)
VALUES
 ('Humberlito Borges', 'Striker', 'DEH 115', 'M', '09:00:00', '10:00:00'),
 ('John Bahnsen', 'General', 'DEH 115', 'F', '11:00:00', '12:00:00'),
 ('Vidya Beniwal', 'Politician', 'DEH 115', 'TR', '08:00:00', '09:30:00');
```