Database connectivity

1. Draw virtual schema - DONE

2. Create Database -DONE

3. Create tables -DONE

4. Configuring our App to work with Databases - DONE

—create build.gradle file-

—add dependencies for sql2o - allow our application to execute SQL cmds for us.

—and postgresql - allow our application to connect to our Postgres SQL database.

5. Creating a Database class - DONE

—to manage these database interactions

—create DB.java file

—Import sql2o library from build.gradle we added in step 4.

—Declare a DB class

—Create a static instance of Sql2o. Sql2o constructor has 3 arguments

—public static Sql2o sql2o = new Sql2o(“jdbc:postgresql://localhost:5432/to\_do”, null, null);

6. Testing with Databases

7. Creating a Test Database - DONE

—Create a database for test with same name of the original database but suffix wit word ‘test’

—@Before —Connecting to test Database

—@After —Clearing the test Database—Executing SQL commands from an Application

\*\*\*\*A try without a catch block is known as ‘Try wit Resources’.

8. Refactoring Classes to work with Databases - DONE

—Remove ‘m’ from all names;

—Create Properties or attributes of a class, Create constructors with arguments and Create methods

9. Connecting a Class to the Database

—Returning all objects — Adding code to all() method

\*\*\*\*Create all(), equals(), save() and find(),

\*\*\*\*Test all(), find(), equals(),save()

—Overriding equals() — to compare Objects we retrieve from database with our code .

—Saving new objects to the Database

—Assigning Unique IDs (getId)

—Finding Objects