*Assignment 4*

**JDBC API**

*Toka Alaa Elgindy (14)*

*Nada Salama Ali (55)*

**JDBC API**

* **Description:**

Java Database Connectivity (JDBC) is an application programming interface (API) for the programming language Java, which defines how a client may access a database. It is a Java-based data access technology used for Java database connectivity. It provides methods to query and update data in a database that is used to access databases, regardless of the driver and database product. JDBC presents a uniform interface to databases - change vendors and your applications only need to change their driver. This application provides user with one of the drivers' type. So, the driver task is to create connection with the database. Then the connection creates statement which used as a bridge to the database interface in the previous assignment.

* **User guide:**

1. Enter the URL of the driver if it is valid program will transfer you to the next step otherwise program will stop.
2. Enter the required path.
3. Enter 1 to continue or 0 to stop running program.
4. Enter your query to create database or drop old one.
5. Enter your query to make any operation on your database.
6. If you enter invalid thing the program will stop running.

* **Design decisions:**

1- At first, user should enter valid SQL URL.

2- Then user should enter valid path to create new connection of this path.

3- User choose continue or close console and if he continues, the program will create new statement

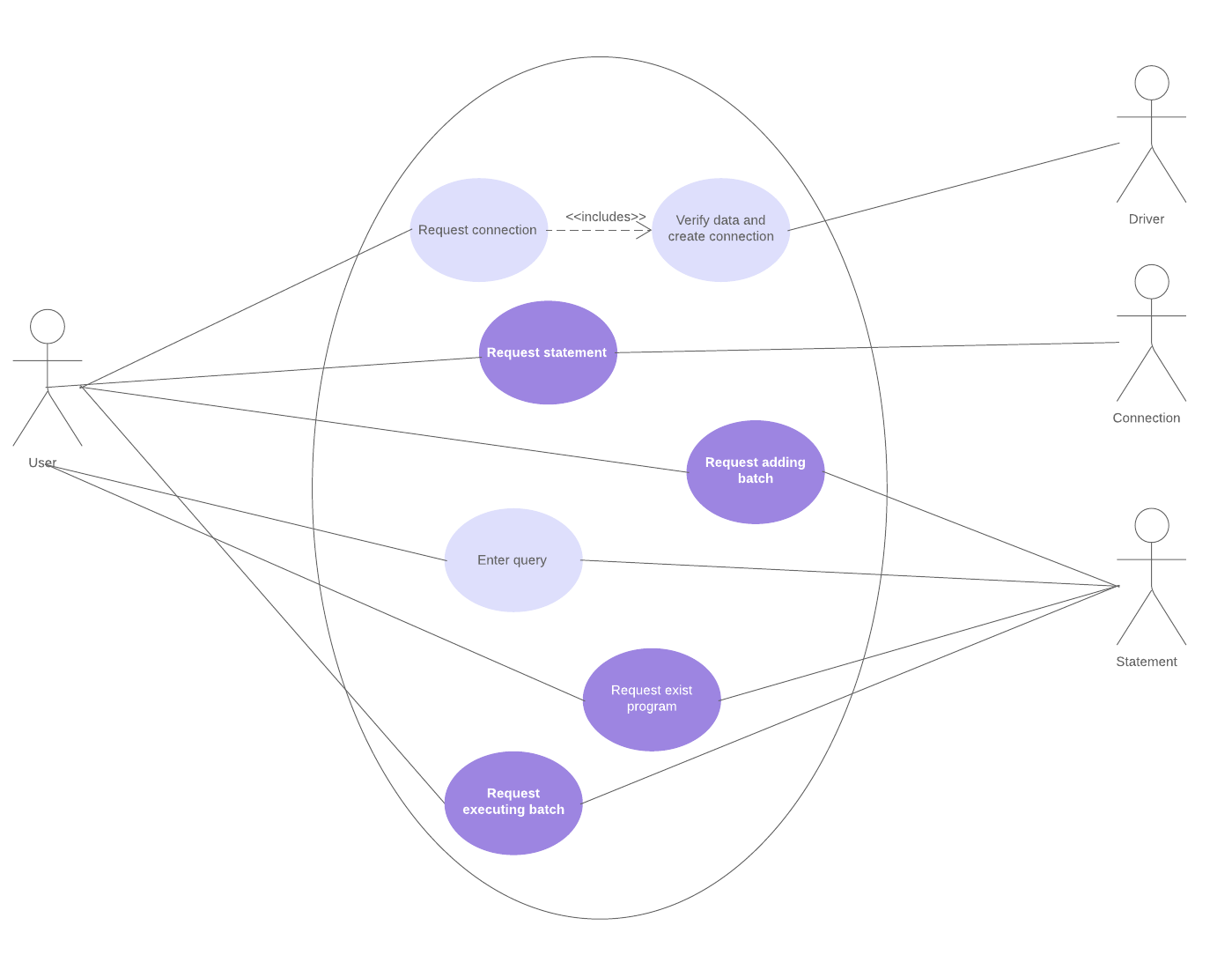
4- Then user enter query of create database to enter in this database or can enter query of drop database to delete exist one.

5- Then he can enter many query to do operation in this database whether create table, drop exist table, select from any table in it, insert in table or delete or update.

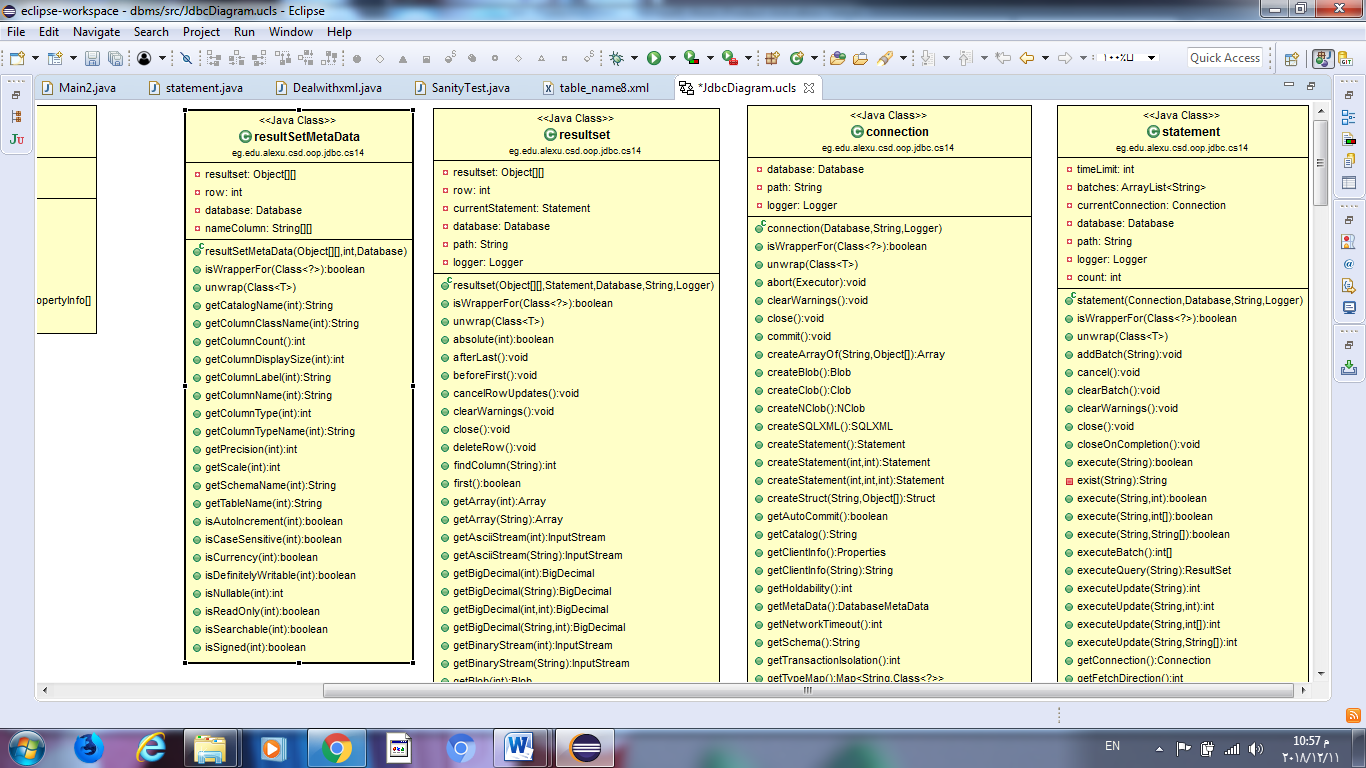
6- There is an option for enter query in batch to be executed later.

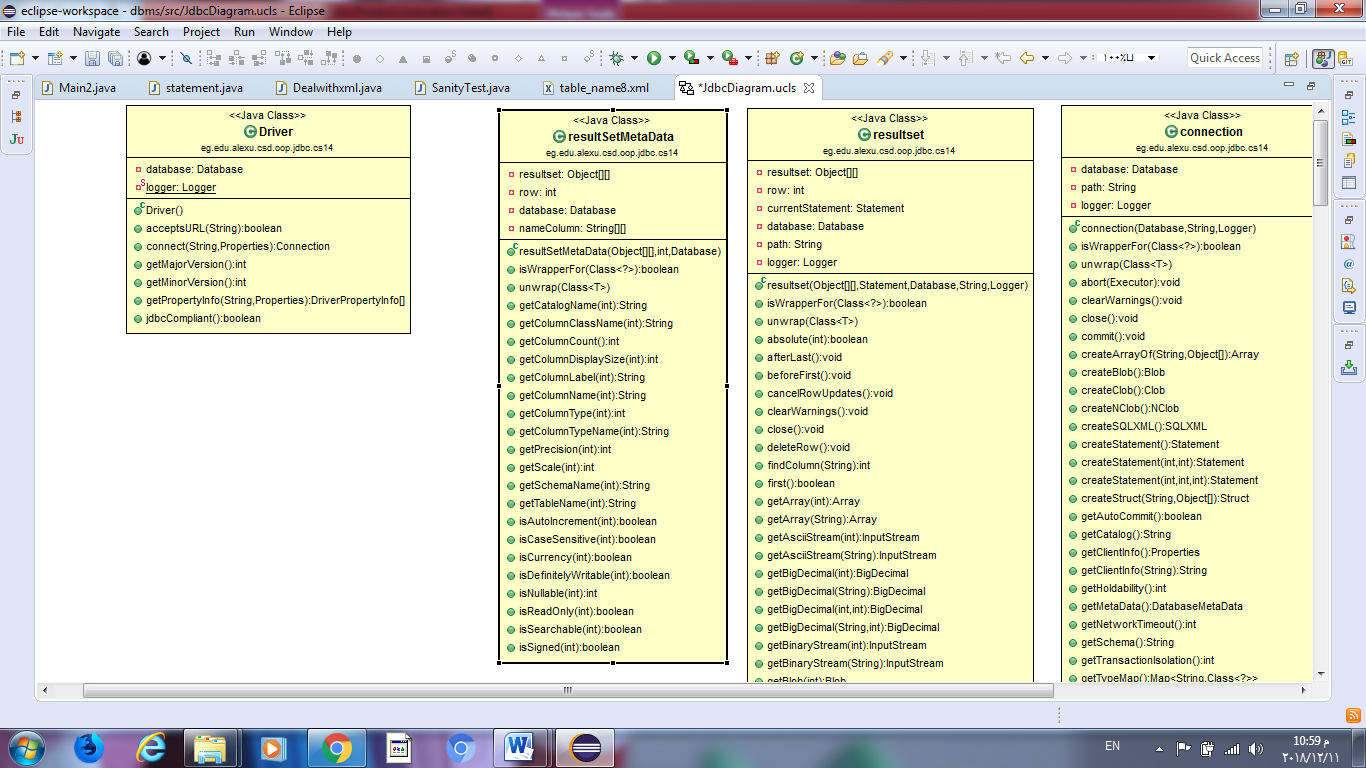
7- User can add in this batch more than query, also can clear all batch or execute all query in batch.

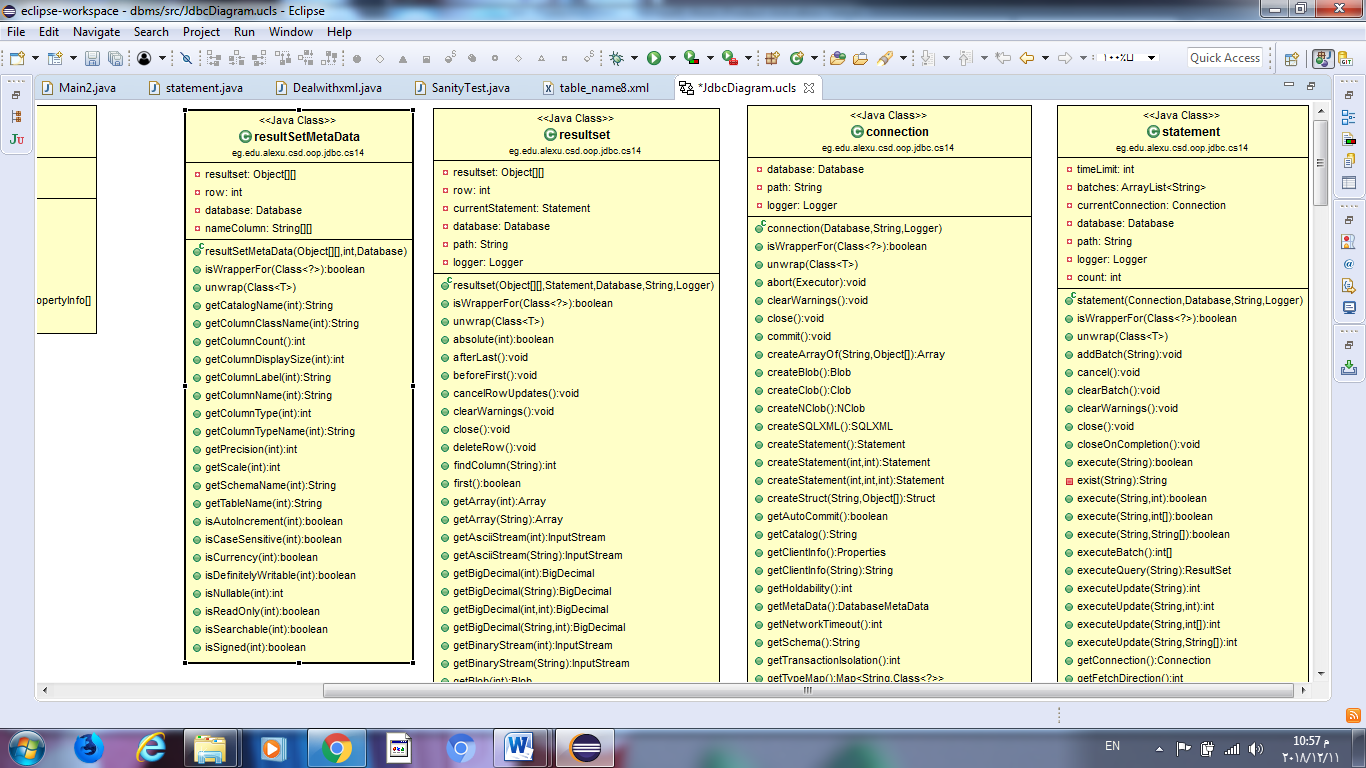
* **UML diagrams:**
* **Use case:**

****

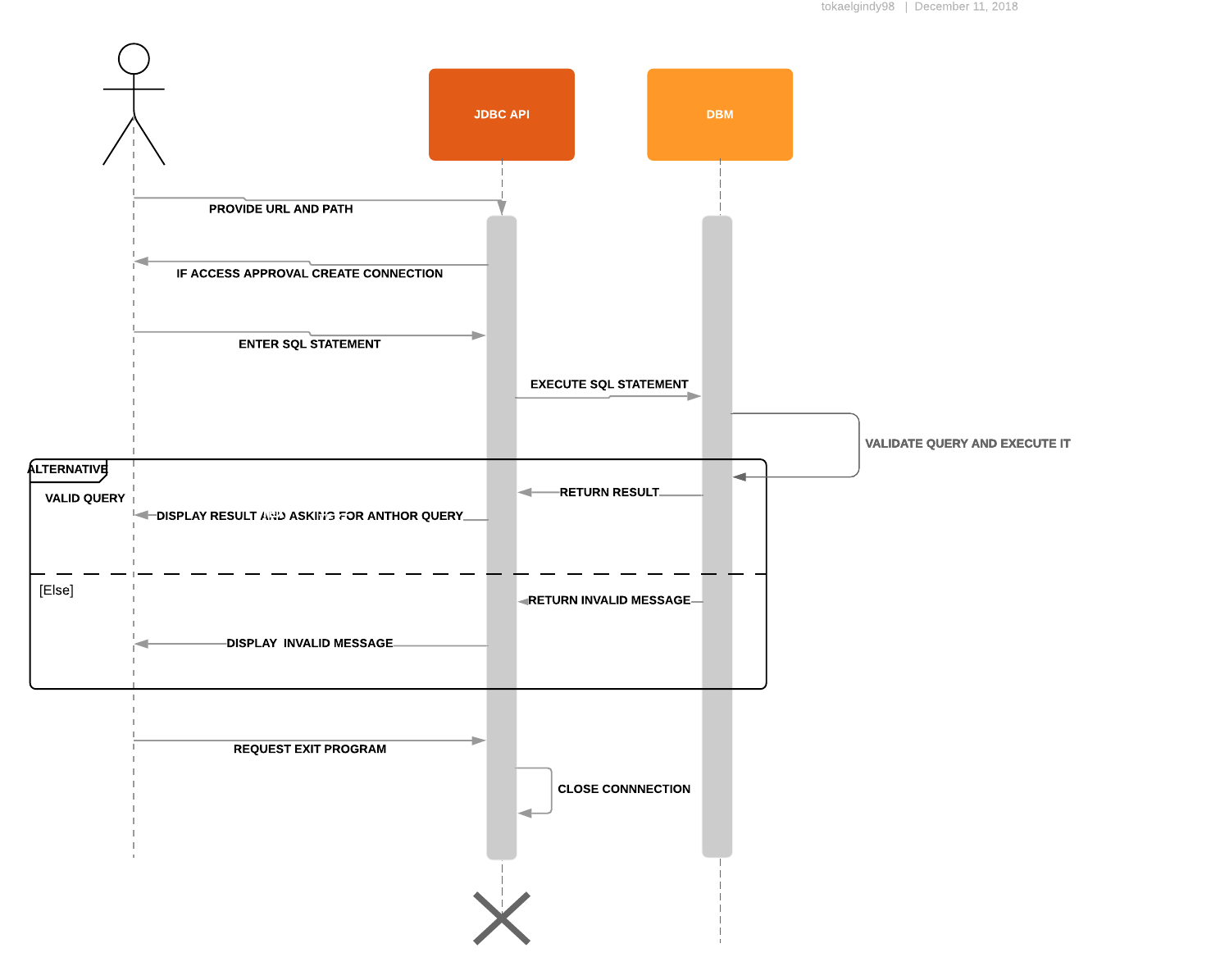
**Class diagram:**





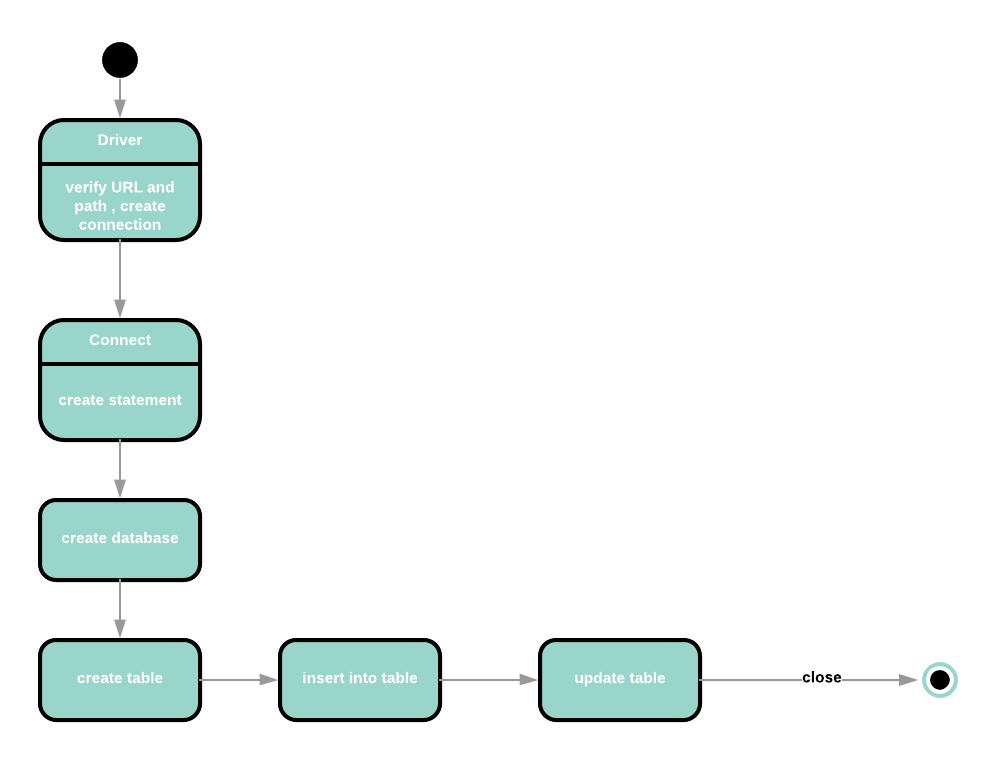


* **Sequence diagram:**

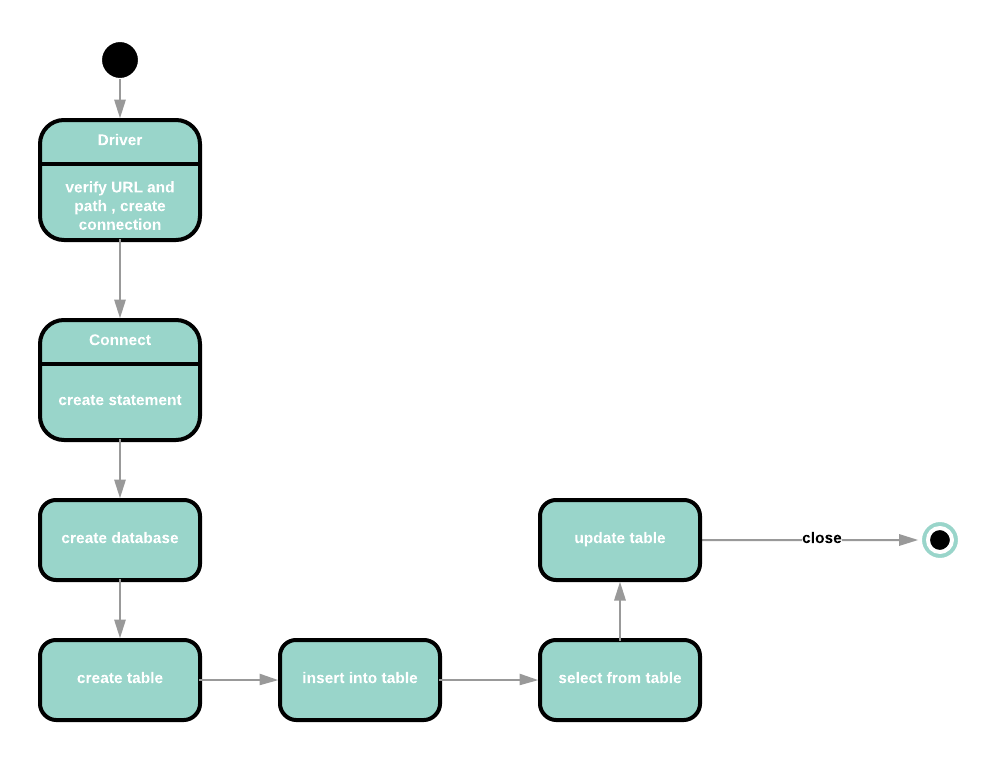
****

* **State diagram:**

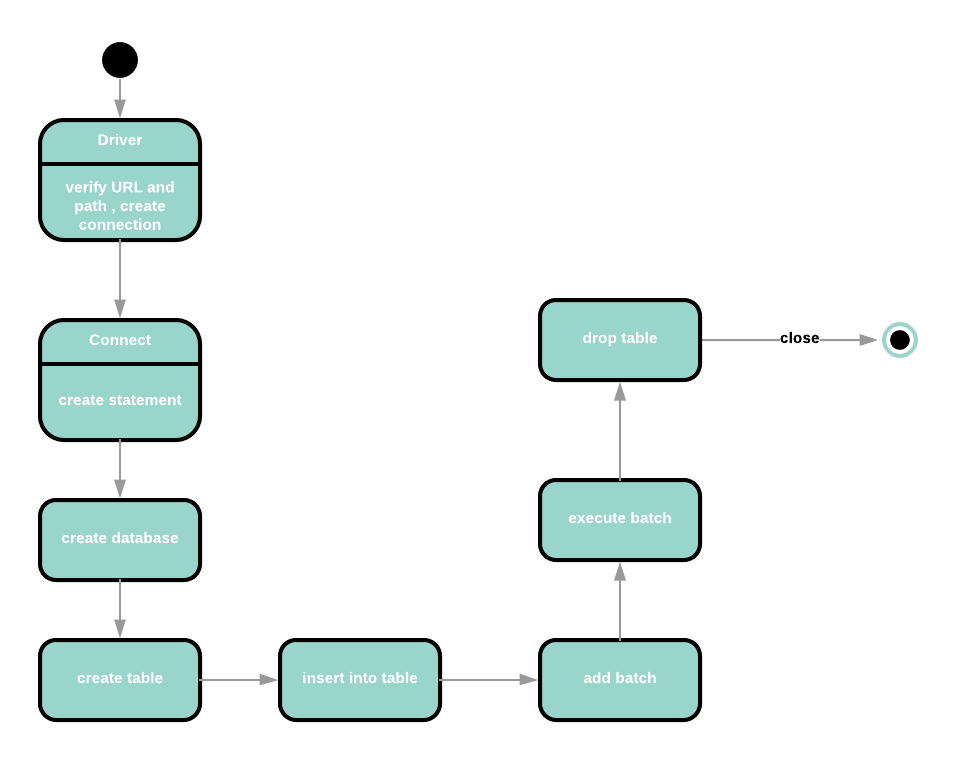
**Scenarios 1:**

****

**Scenarios 2:**

****

**Scenarios 3:**

****