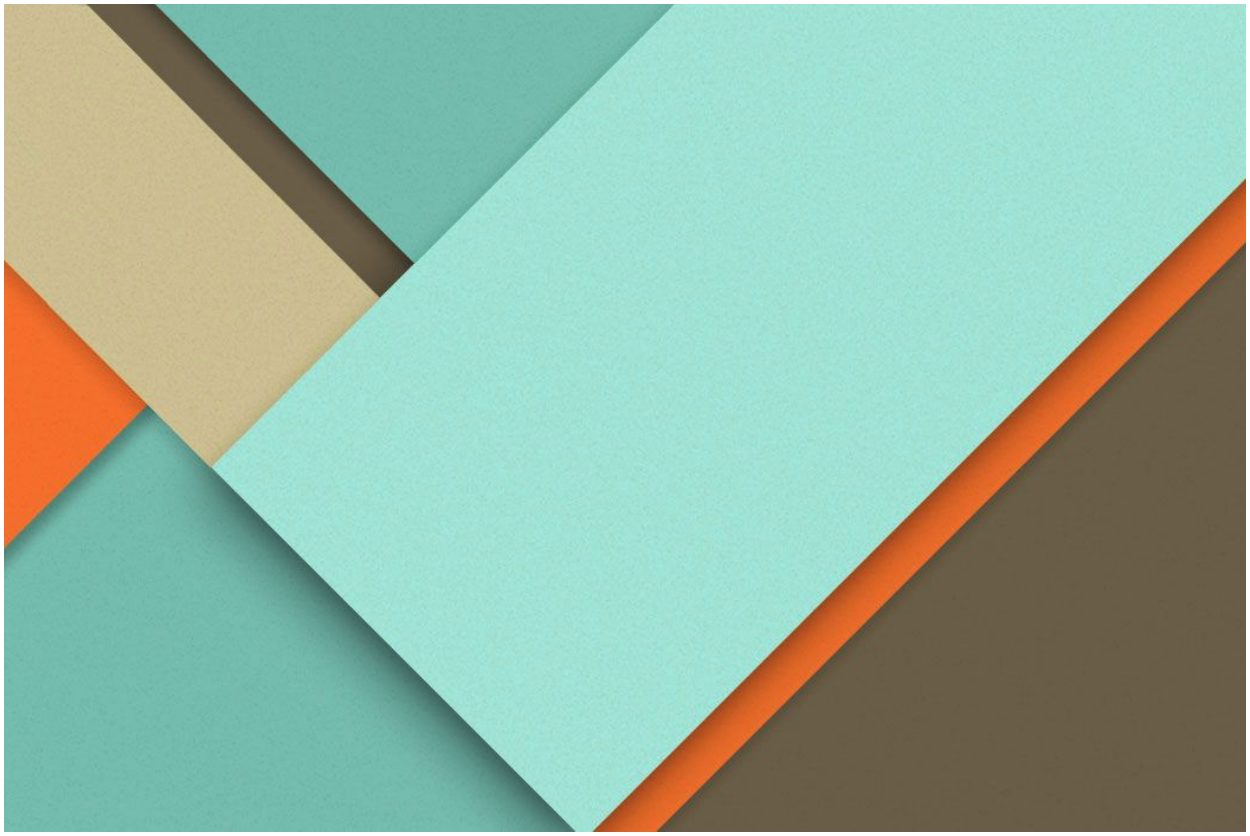


JDBC

Your Name

1- Ahmed Ali

2-Ahmed Kamal

3- Mahmoud Kamal

4-Ahmed Ayman

Overview

.Java Database Connectivity (JDBC) provides Java developers with a standard API 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.

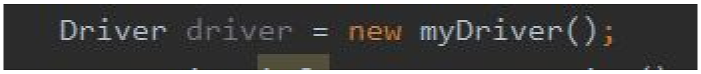
Goals

1. • Implements JDBC standard interface.
2. • Integrate JDBC interface with your earlier DBMS implementation.
3. • Apply different design patterns to your model.
4. • Extend your previously written code and augment it with new features.

User Manual

1- Driver

To use driver take this command then you can access connection , statement , resultset and metaresultSet



```
    Driver driver = new myDriver();
```

Allowing function

* acceptsURL(String url)

∗ connect(String url, Properties info)

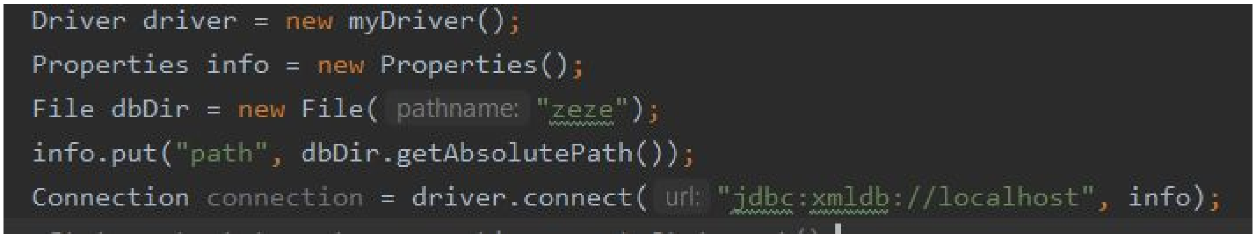
2- Connection

To access connection use

Connection connection = driver.connect("jdbc:xmldb://localhost", info);

Note : info is a properties which contains key (“path”) and the value

Is a file containing the path to dataBase or the path without File



```
Driver driver = new myDriver();
Properties info = new Properties();
File dbDir = new File( pathname: "zeze");
info.put("path", dbDir.getAbsolutePath());
Connection connection = driver.connect( url: "jdbc:xmldb://localhost", info);
```

Allowing function

*close()

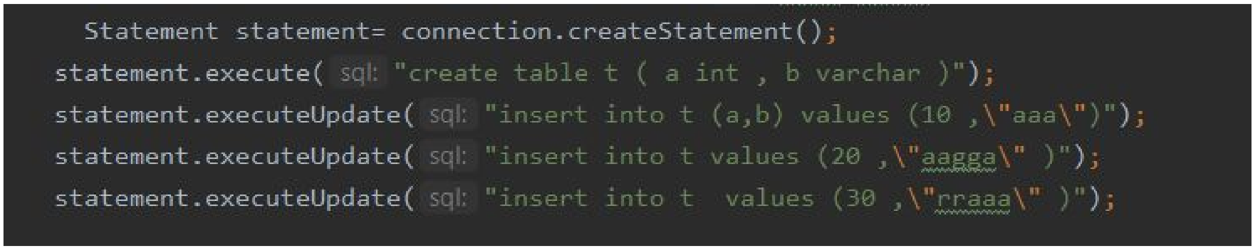
∗ createStatement()

3-Statement

To create Statement use

Statement statement= connection.createStatement();

Then you can access tables and it’s operations (insert , select , update , delete)



```
  Statement statement= connection.createStatement();
statement.execute( sql: "create table t ( a int , b varchar )");
statement.executeUpdate( sql: "insert into t (a,b) values (10 ,\"aaa\")");
statement.executeUpdate( sql: "insert into t values (20 ,\"aagga\" )");
statement.executeUpdate( sql: "insert into t  values (30 ,\"rraaa\" )");
```

Note : if statement is closed or **connection creating it is closed** then you can’t use it again

Allowing function

∗ addBatch(String sql)

∗ clearBatch()

∗ close()

∗ execute(String sql)

∗ executeBatch()

∗ executeQuery(String sql)

∗ executeUpdate(String sql)

∗ getConnection()

∗ getQueryTimeout()

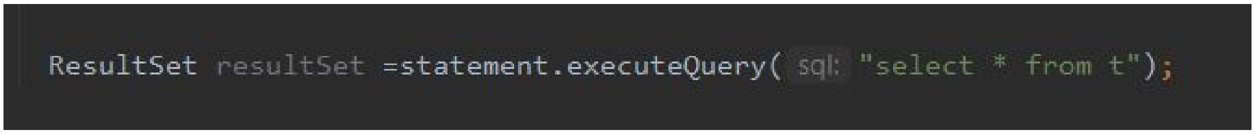
∗ setQueryTimeout(int seconds)

4- Resultset

Result set is a data which get from executeQuery from statement

To create it use

ResultSet resultSet =statement.executeQuery(SQL QUERY);



```
ResultSet resultSet =statement.executeQuery( sql: "select * from t");
```

Allowing function

- absolute(int row)
- afterLast()
- beforeFirst()
- close()
- findColumn(String columnLabel)
- first()
- getInt(int columnIndex)
- getInt(String columnLabel)
- getMetaData()
- getObject(int columnIndex)
- getStatement()
- getString(int columnIndex)
- getString(String columnLabel)
- isAfterLast()
- isBeforeFirst()
- isClosed()
- isFirst()
- isLast()
- last()
- next()
- previous()

5- ResultSetMetaData

ResultSetMetaData is a specific data about result set

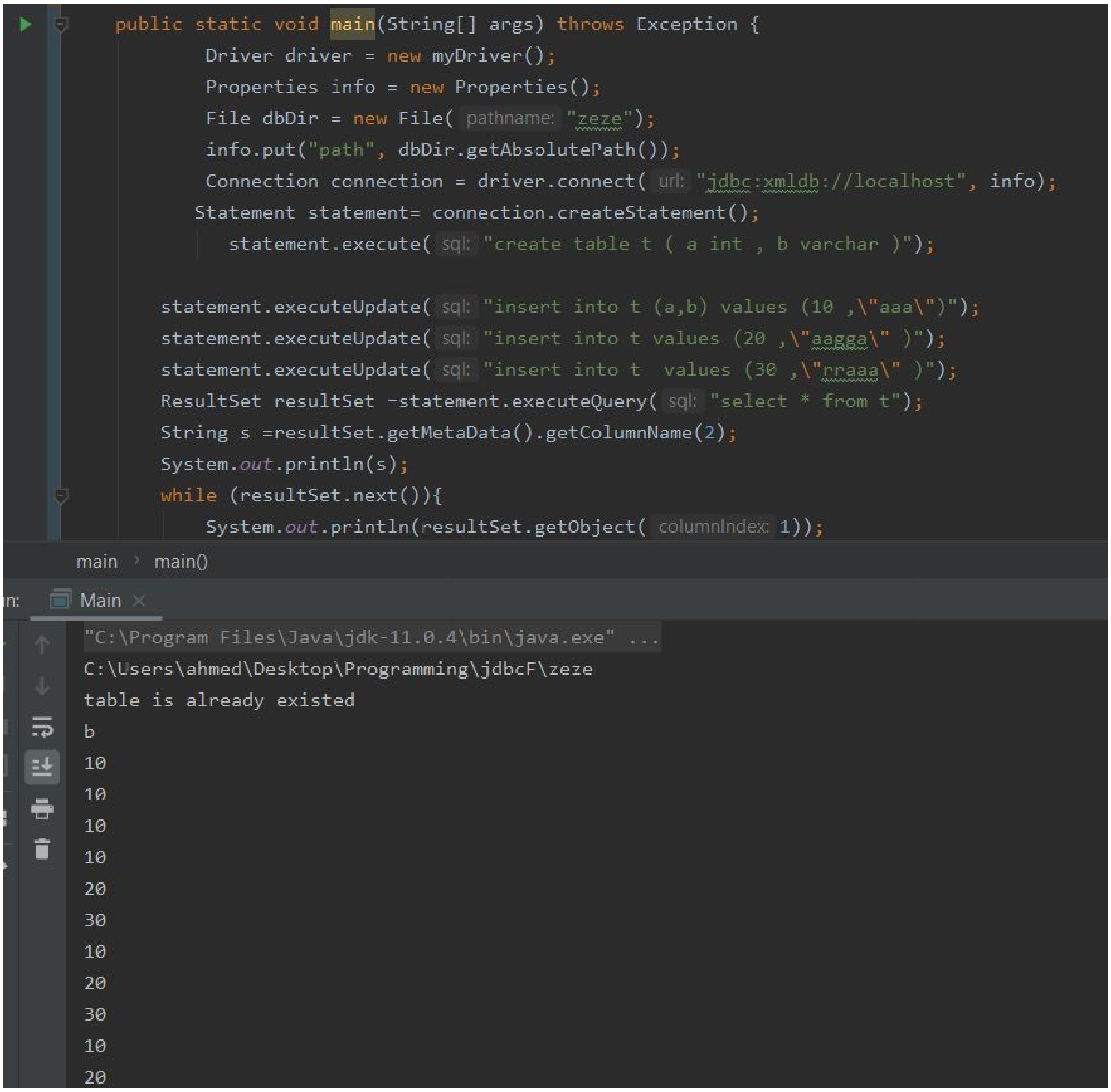
To use it use command

```
ResultSetMetaData resultSetMetaData=resultSet.getMetaData();
```

Allowing function

- getColumnCount()
- getColumnLabel(int column)
- getColumnName(int column)
- getColumnType(int column)
- getTableName(int column)

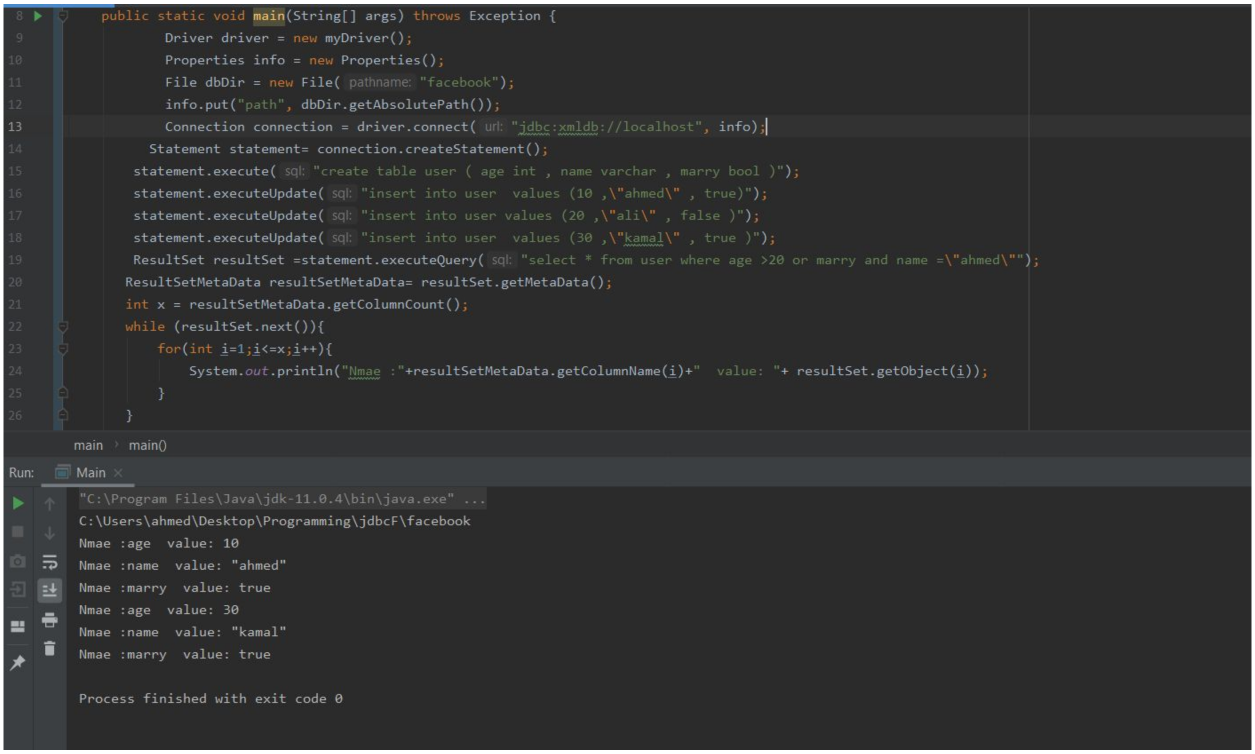
Examples



```
public static void main(String[] args) throws Exception {
        Driver driver = new myDriver();
        Properties info = new Properties();
        File dbDir = new File( pathname: "zeze");
        info.put("path", dbDir.getAbsolutePath());
        Connection connection = driver.connect( url: "jdbc:xmldb://localhost", info);
       Statement statement= connection.createStatement();
          statement.execute( sql: "create table t ( a int , b varchar )");

    statement.executeUpdate( sql: "insert into t (a,b) values (10 ,\"aaa\")");
    statement.executeUpdate( sql: "insert into t values (20 ,\"aagga\" )");
    statement.executeUpdate( sql: "insert into t  values (30 ,\"rraaa\" )");
    ResultSet resultSet =statement.executeQuery( sql: "select * from t");
    String s =resultSet.getMetaData().getColumnName(2);
    System.out.println(s);
    while (resultSet.next()){
        System.out.println(resultSet.getObject( columnIndex: 1));
```

```
"C:\Program Files\Java\jdk-11.0.4\bin\java.exe" ...
C:\Users\ahmed\Desktop\Programming\jdbcF\zeze
table is already existed
b
10
10
10
10
20
30
10
20
30
10
20
```



```
public static void main(String[] args) throws Exception {
        Driver driver = new myDriver();
        Properties info = new Properties();
        File dbDir = new File( pathname: "facebook");
        info.put("path", dbDir.getAbsolutePath());
        Connection connection = driver.connect( url: "jdbc:xmldb://localhost", info);
      Statement statement= connection.createStatement();
    statement.execute( sql: "create table user ( age int , name varchar , marry bool )");
    statement.executeUpdate( sql: "insert into user  values (10 ,\"ahmed\" , true)");
    statement.executeUpdate( sql: "insert into user values (20 ,\"ali\" , false )");
    statement.executeUpdate( sql: "insert into user  values (30 ,\"kamal\" , true )");
    ResultSet resultSet =statement.executeQuery( sql: "select * from user where age >20 or marry and name =\"ahmed\"");
   ResultSetMetaData resultSetMetaData= resultSet.getMetaData();
   int x = resultSetMetaData.getColumnCount();
   while (resultSet.next()){
       for(int i=1;i<=x;i++){
           System.out.println("Nmae :"+resultSetMetaData.getColumnName(i)+"  value: "+ resultSet.getObject(i));
       }
   }
```

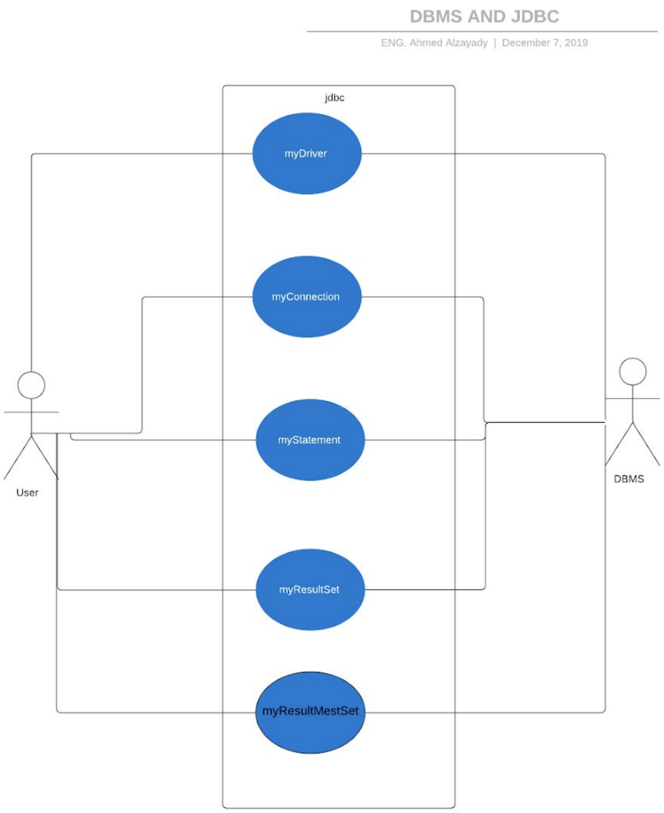
```
"C:\Program Files\Java\jdk-11.0.4\bin\java.exe" ...
C:\Users\ahmed\Desktop\Programming\jdbcF\facebook
Nmae :age  value: 10
Nmae :name  value: "ahmed"
Nmae :marry  value: true
Nmae :age  value: 30
Nmae :name  value: "kamal"
Nmae :marry  value: true

Process finished with exit code 0
```

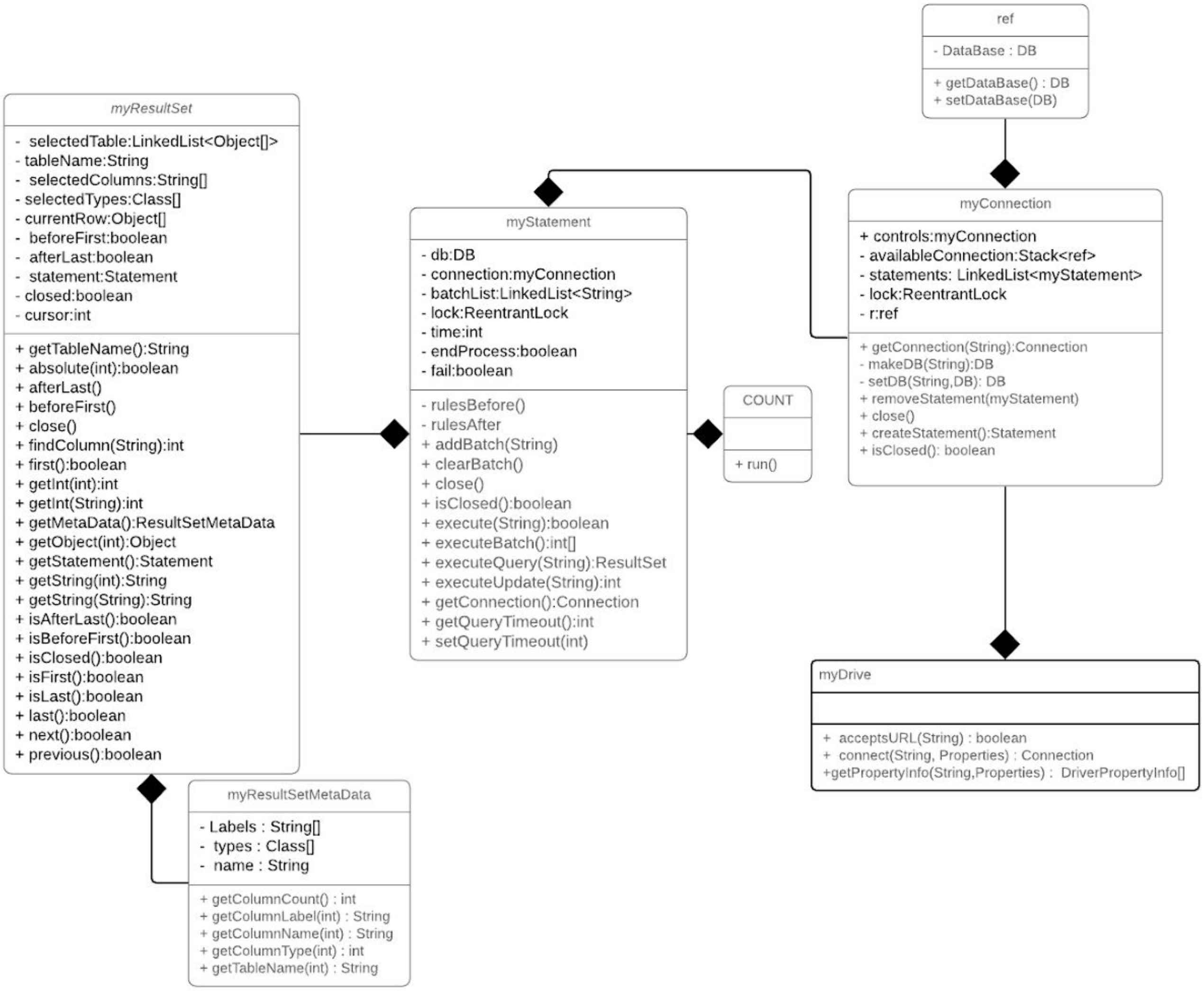
Design Pattern :

Object Pool Pattern

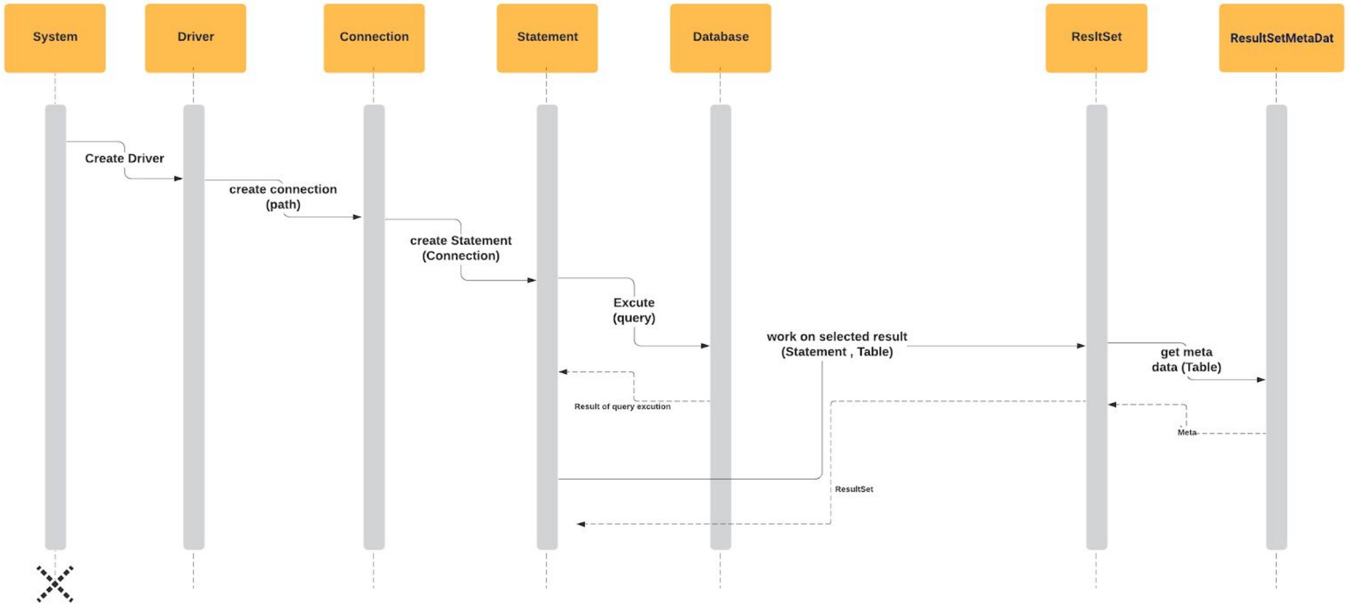
USE CASE DIAGRAM



CLASS DIAGRAM

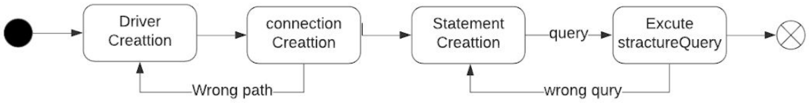


SEQUENCE DIAGRAM

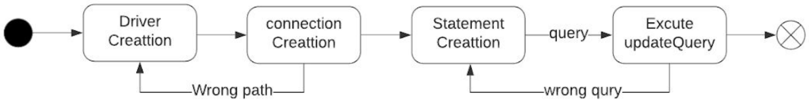


STATE DIAGRAM

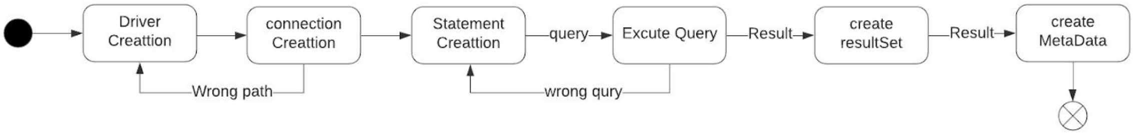
Create Table



insert into Table



select from Table to getResultSet and getMetaData


Driver Creattion
connection Creattion
Statement Creattion
query
Excute Query
Result
create resultSet
Result
create MetaData
Wrong path
wrong qury