

*Alexandria University
Faculty of Engineering
Computer and Systems Engineering Dept.
Second Year
Fall 2016*

Report about JDBC Assignment

Team work:

- 1. Aya Ali Saad Abo Zaid (2)**
- 2. Aya Fouad Metwally (3)**
- 3. Salma Mohammed Elsayed (33)**
- 4. Mona Alaa Darwesh Mostaufa (76)**

Introduction:

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.

We started the project work with modifying the previous project as we organized the design of the program as we can access another backend like json and protocol buffers easily in the program as each one of them is different from each one other as protocol buffers file relies on messages, json files relies on objects, xml file relies on nodes and elements. So we made our project more extensible.

Then we started the current assignment by adding the new features:

- ❖ *First: Adding support for two statements/operators other than those you implemented o the previous phase. You should choose of: ALTER TABLE (Add/delete columns), "SELECT DISTINCT, "SQL UNION, and "SQL ORDER BY.*
 - *We added alter statement that has two possible operations:*
 - *Add column in the table*
 - *Delete column from the table.*
 - *We added also select distinct that mean*

- ❖ *Second: Adding support for these new data types for your XML DBMS (Date, Float). For Date, you use the java.sql.Date class as a representation, and for Float, you use java.lang.Float.*
- ❖ *We have to add a different backend writer for our DBMS, for example, the backend writer for the previous assignment was XML. This time, you need to support another format. You can choose between JSON and Protocol Buffers.*
 - *We added json file in our project.*
- ❖ *In the first of the program when the user connects he should enter "xmldb" for writing in xml file and should enter "jsondb" for writing in json file.*
- ❖ *Then the program should keep this parameter to use it in the program.*

We finished this part so we entered in the connection to the program in this part we implemented the JDBC main interfaces to access the tables' data.

- *Java.sql.Driver.*

This is the interface we will use it so the user can enter and the program will accept his connection or not.

In this class the program makes sure that the entered URL is available. Then he called the connection class.

 - *Accepts URL (String URL).*
 - *Connect (String URL, Properties info).*
 - *Get Property Info (String URL, Properties info).*
- *java.sql.Connection*

This is the interface after accept the user connects so the user will enter and create database

 - *Close ().*
 - *Create Statement ().*
- *java.sql.Statement.*

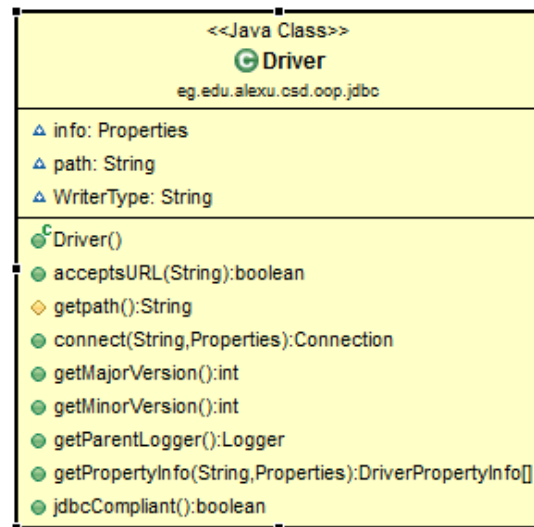
This is an interface used to enter the operation that user wants to be executed like insert, update, alter add, alter delete, distinct, delete, select and delete.
- *java.sql.ResultSet.*
- *java.sql.ResultSetMetaData.*

These used to get the number of columns or rows that are changed or the operation is executed on them and helps us to know the values of current row or next row or the previous row by its name.

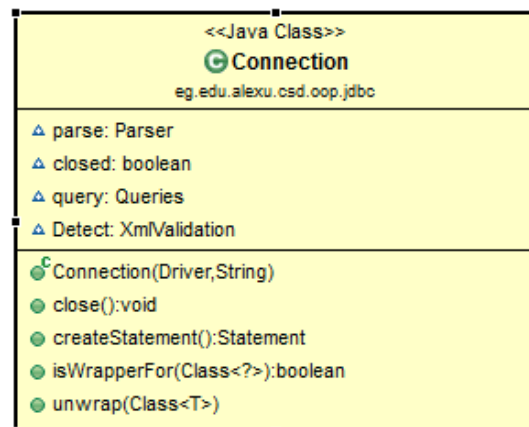
- We added also log4j in the program with its configuration file that records every operation in the program, every error, every connection for any user and every query that the user will enter after accept his connection.

Uml diagrams:

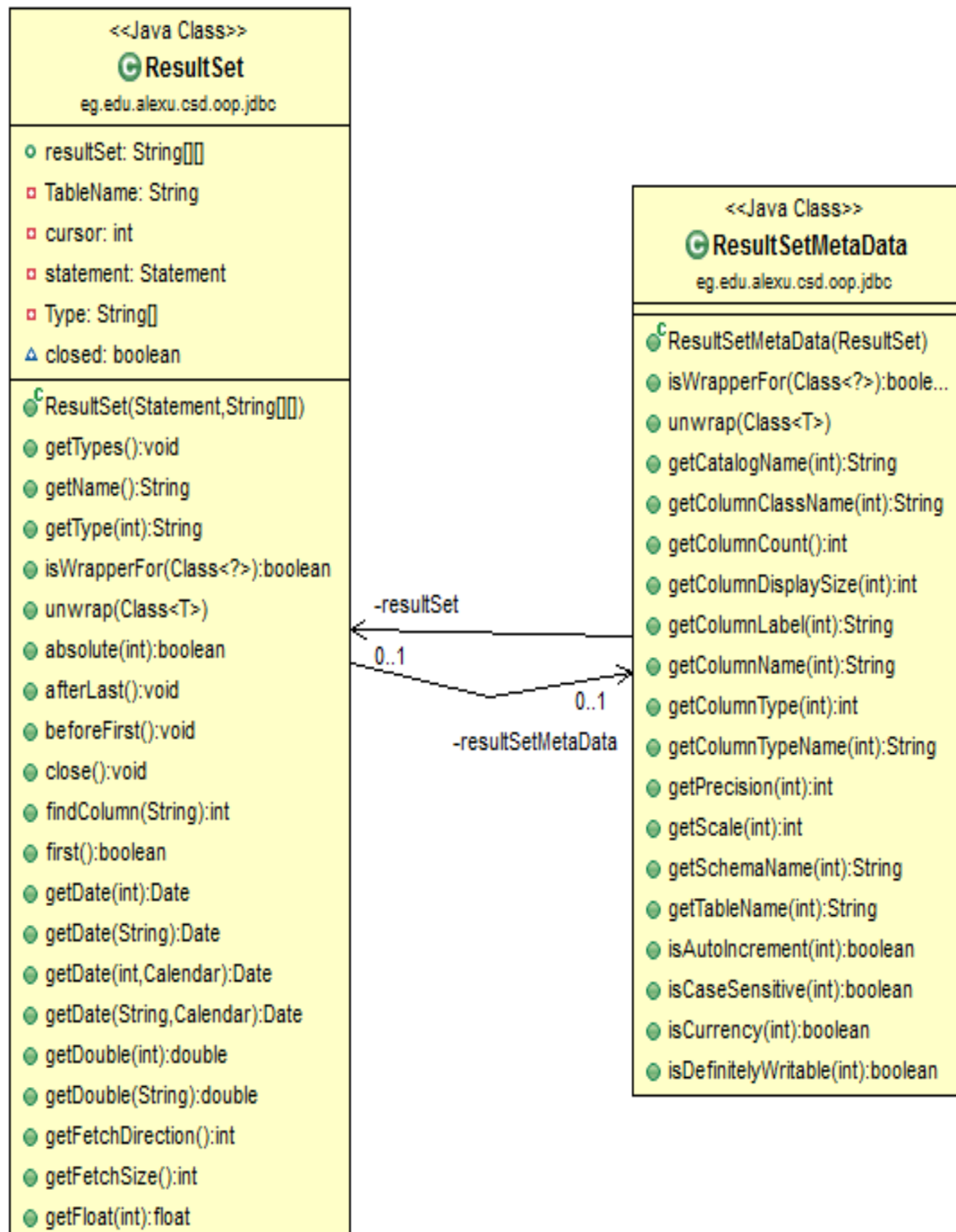
Driver class that implement driver interface (java.sql.Driver)



For the connection (java.sql.Connection)



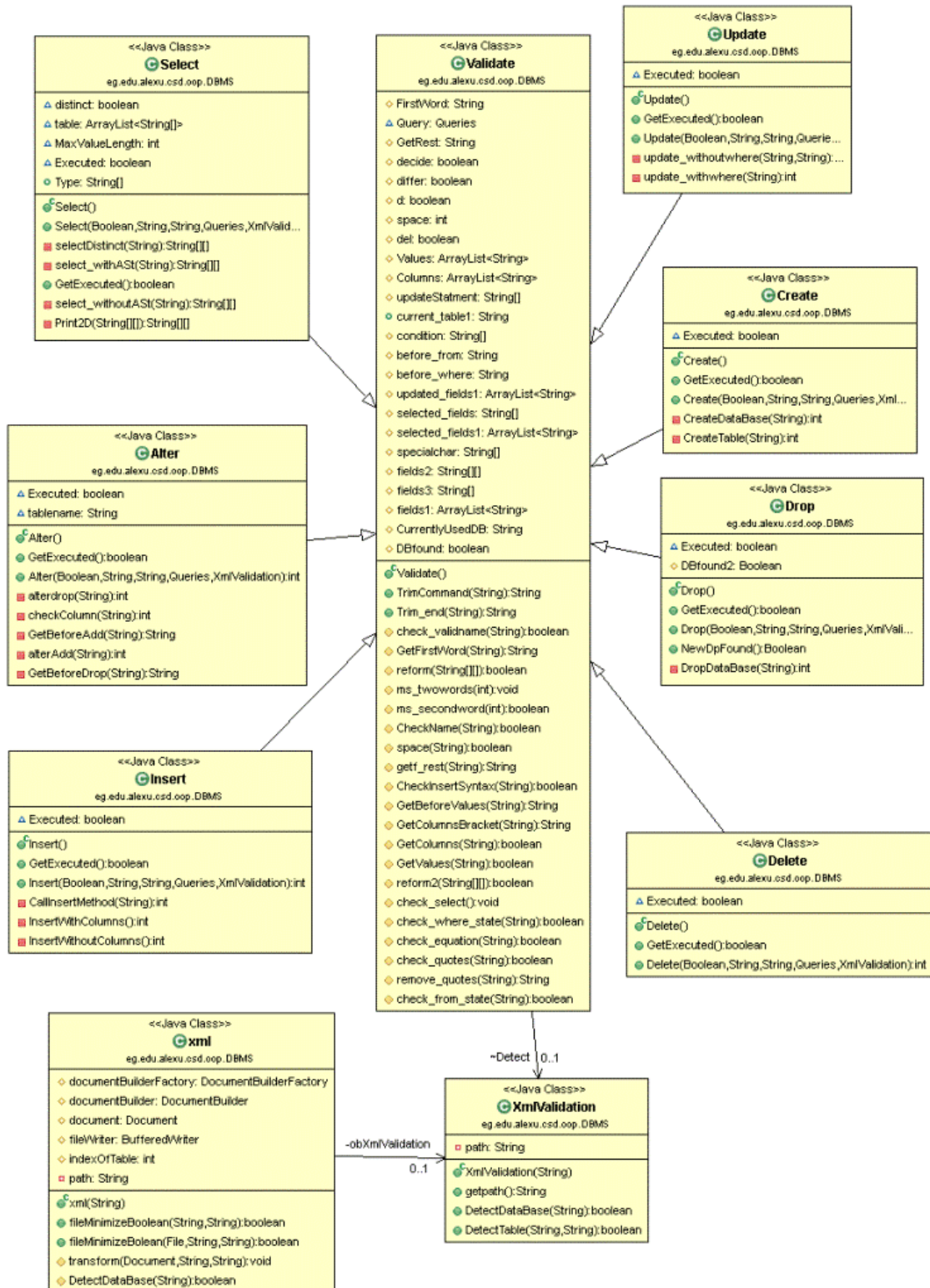
For ResultSet and ResultSetMetaData
(java.sql.ResultSet & java.sql.ResultSetMetaData)



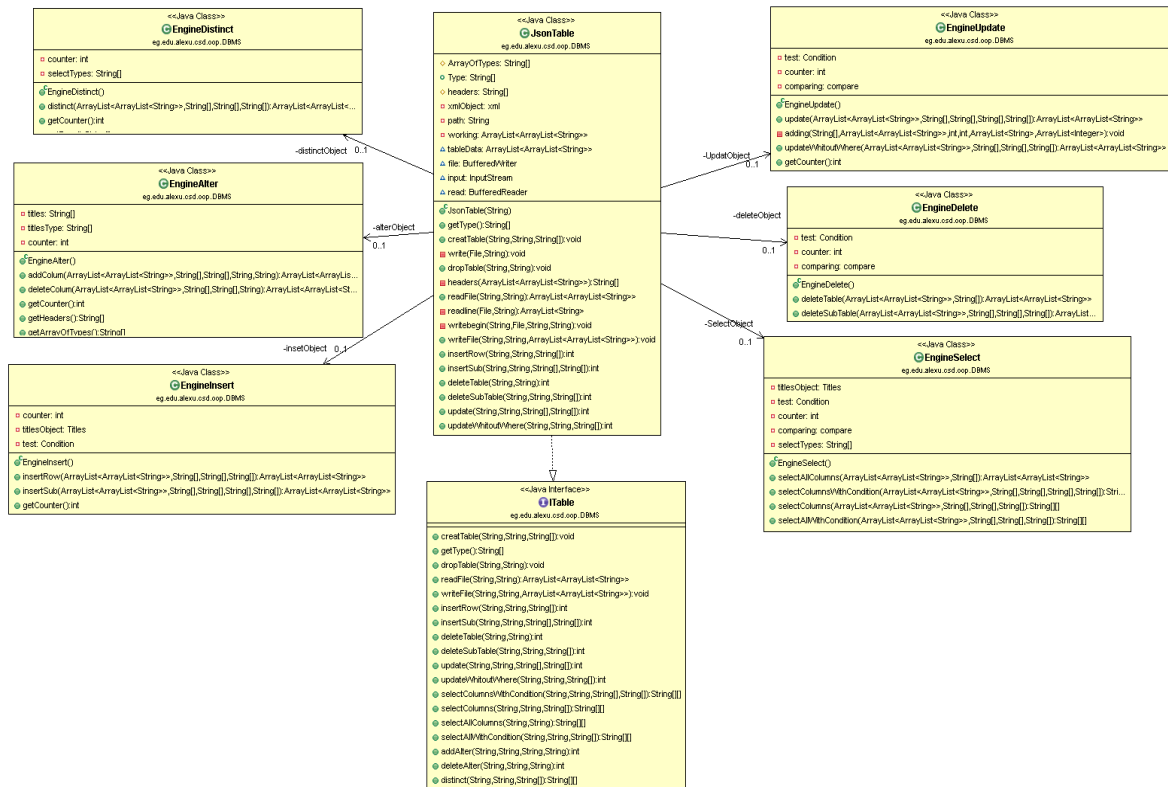
For Statement (java.sql.Statement) and its relation with the other interfaces.



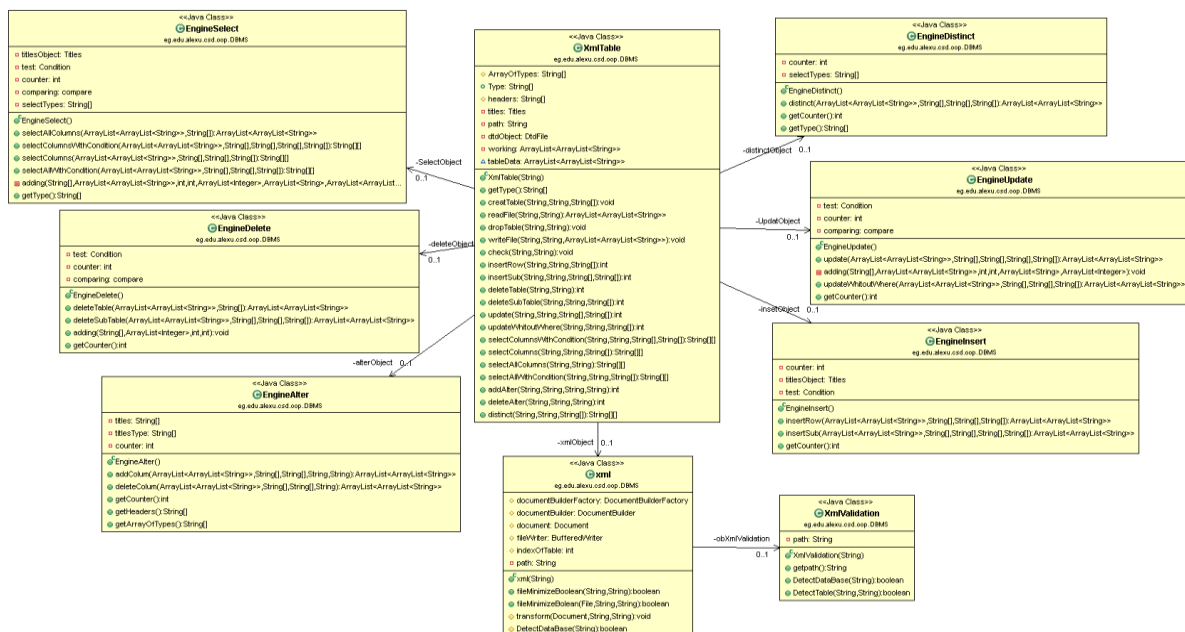
For the database part: We added two features Alter and Distinct.



This uml diagram express json file with its functionalities:

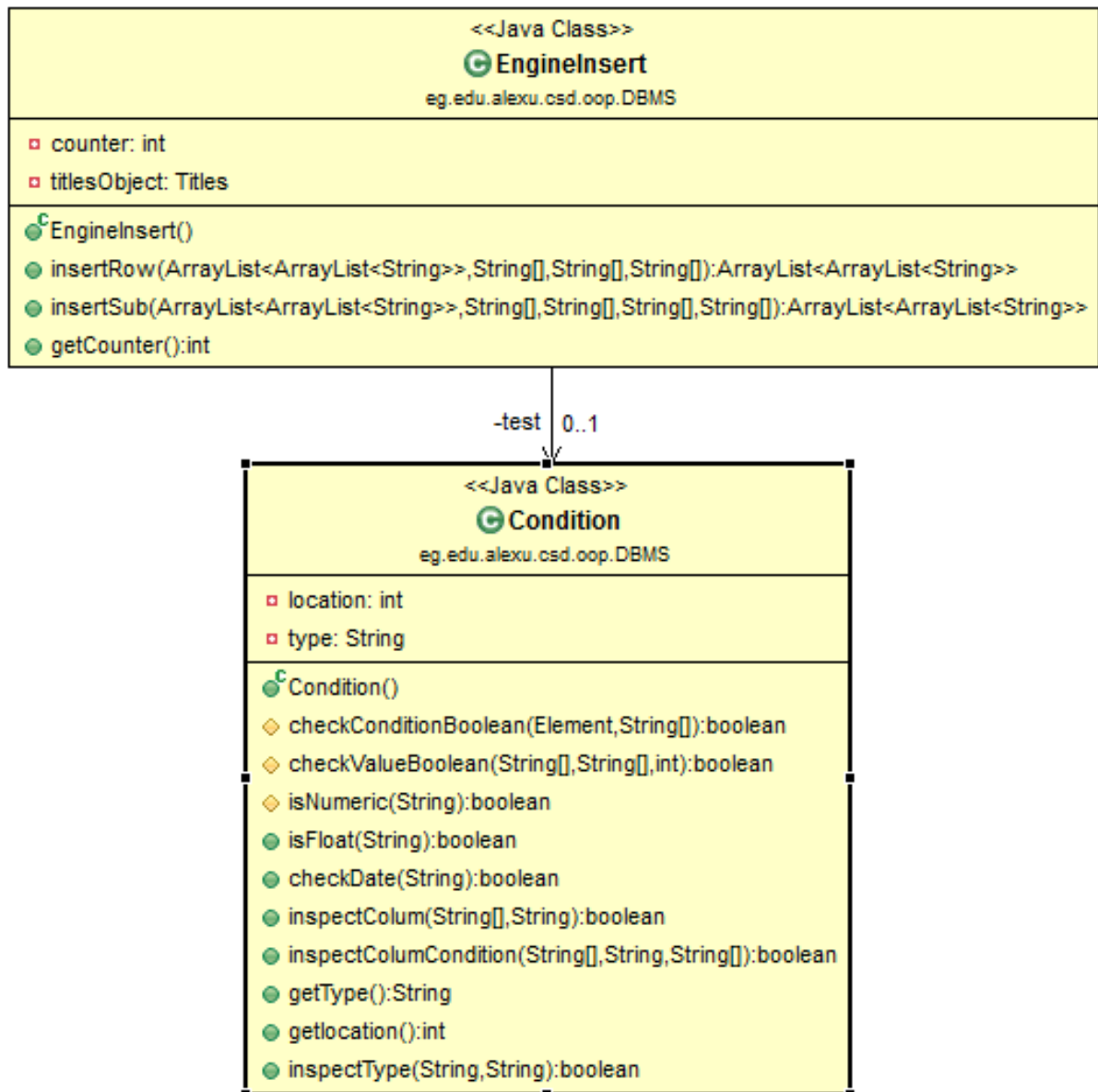


And the same for XML file:

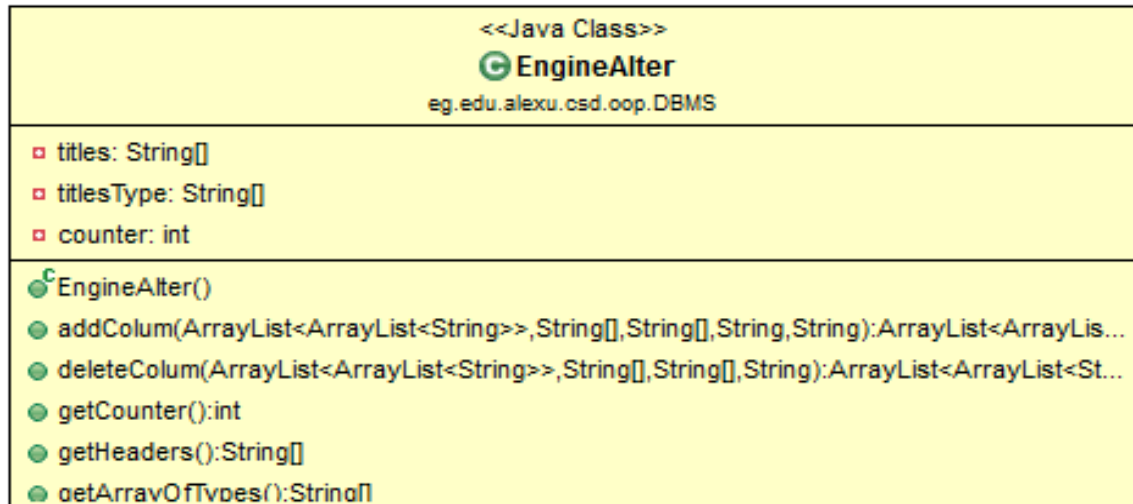


For the functionalities:

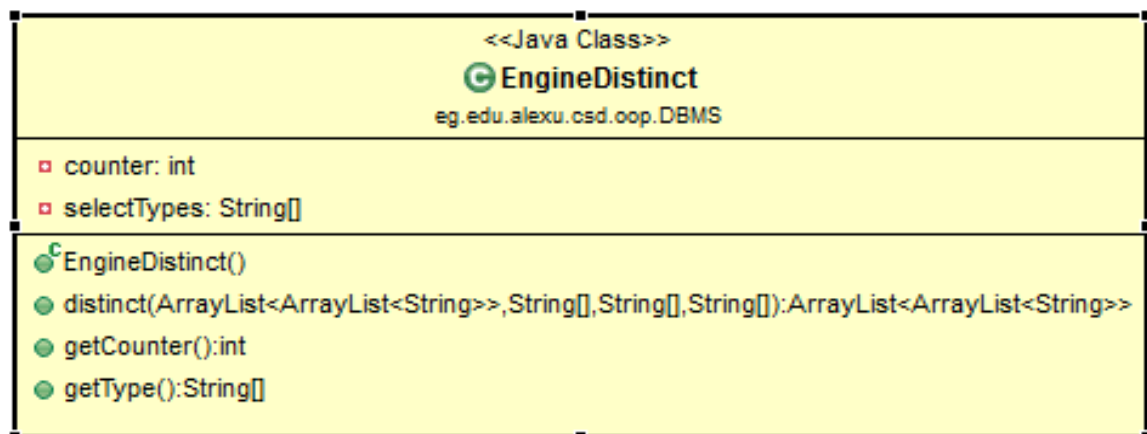
1-Insert



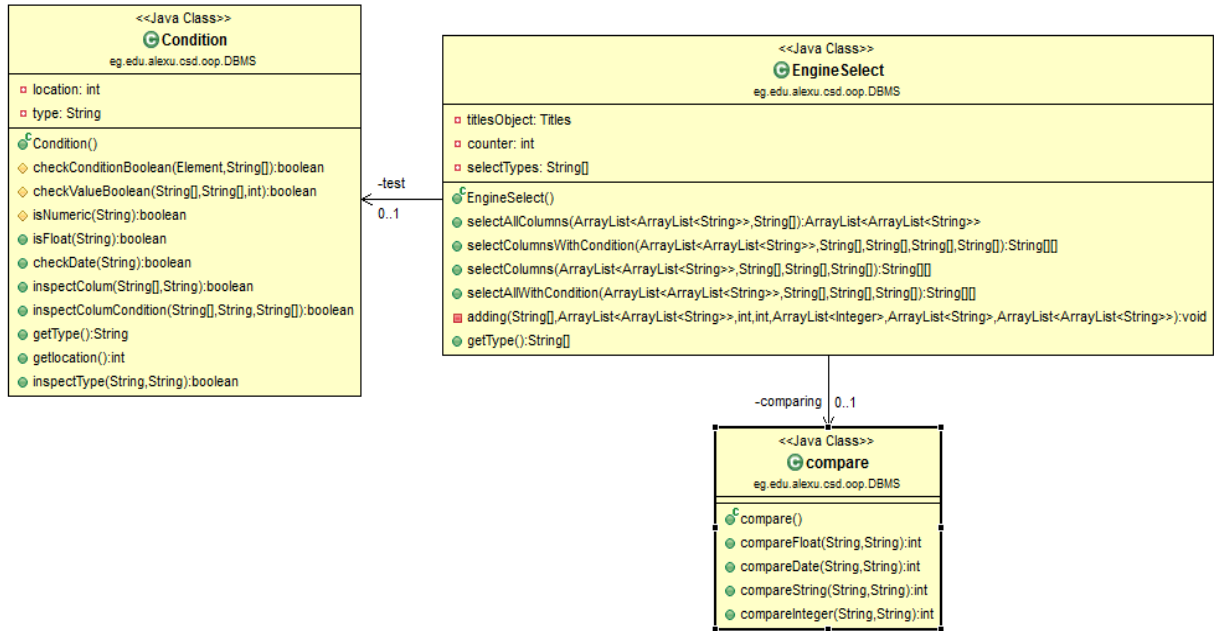
2-Alter:



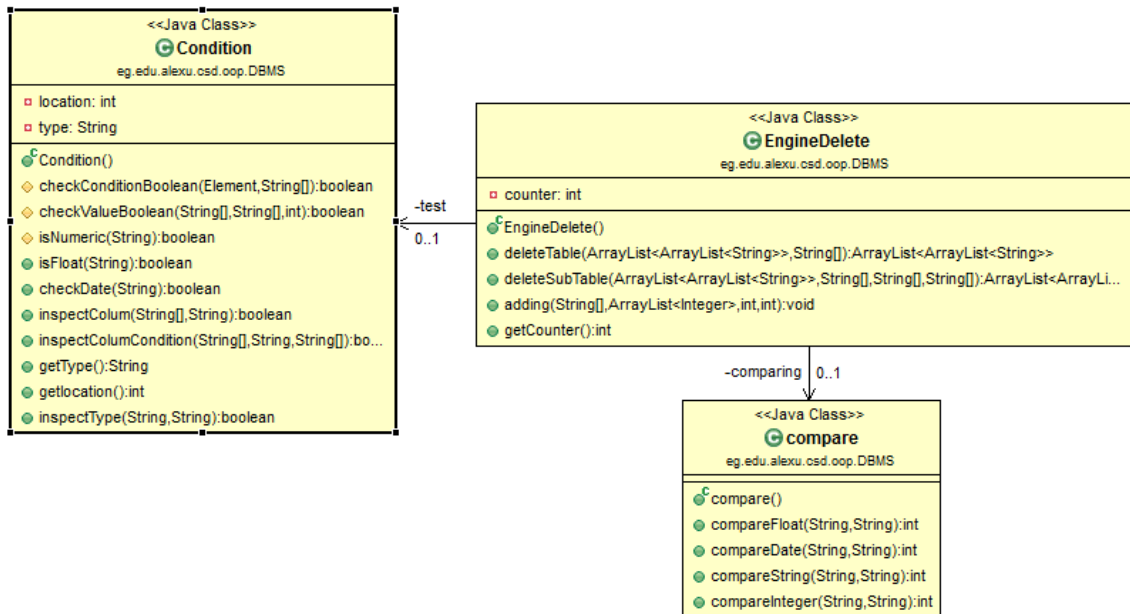
3-Distinct:



4-Select:



5-Delete:



6-Update:

