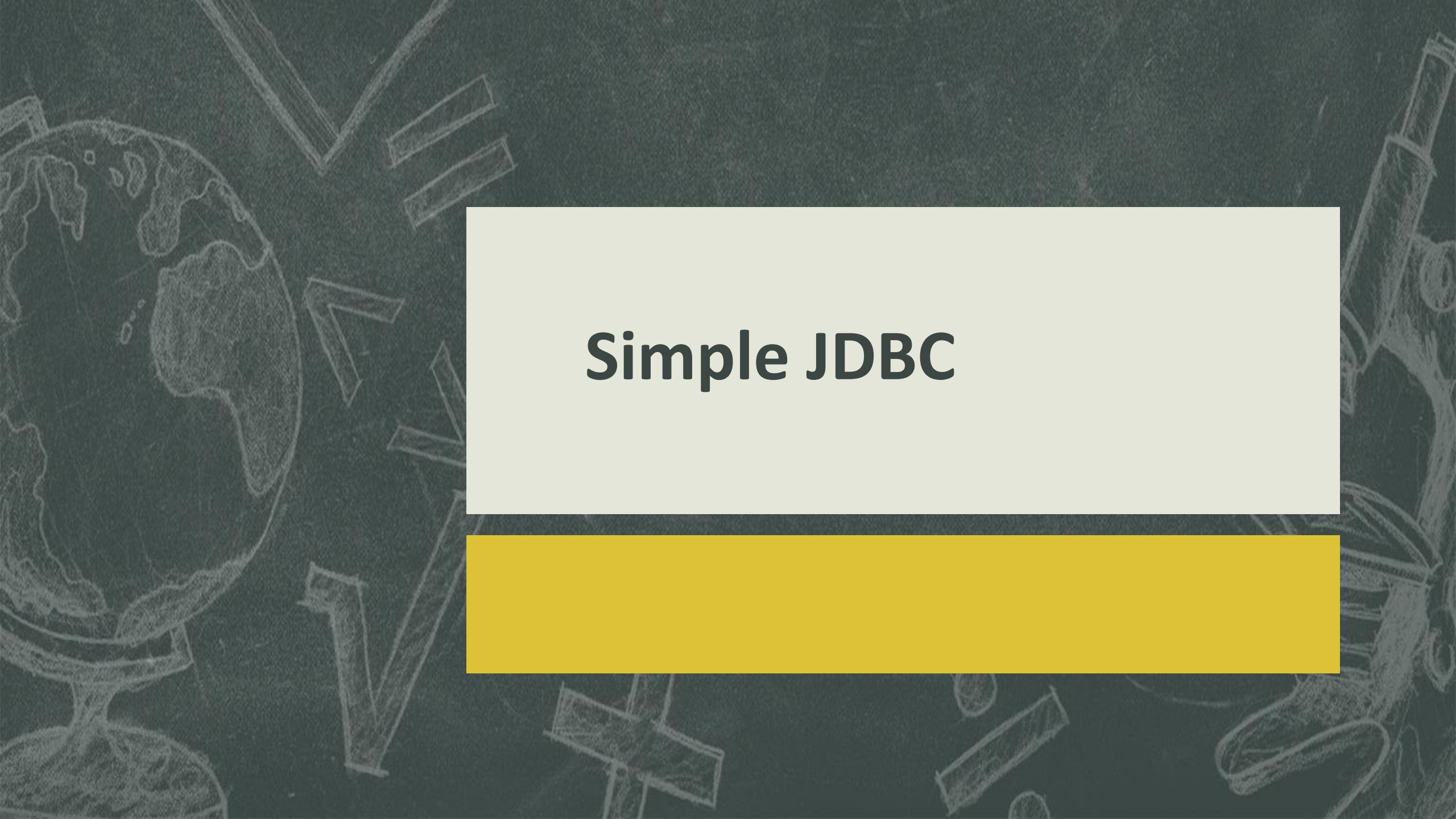


Simple JDBC

Description, features and design decisions

Description

We designed and implemented JDBC. **Java Database Connectivity,** Application Program Interface **(API)** in which user can access database, API can dynamically load correct java package with JDBC Driver Manager. Driver Manager is responsible for creating JDBC connections . JDBC support Creating and executing SQL queries to give instructions to the program to and maybe Update,Delete,Select.

Features and design decisions

- We use the console or GUI (swing) in the User Interface.
- We use SQL queries to give instructions to the program.
- You must enter a correct query otherwise error message will be displayed on screen.
- Driver manger responsible for making Connections.
- You can access database through resultset and metadata.
- In result set You can find column , return object in specific row by entering either column index or column string, moving cursor pointing row to either next of previous row, checking is cursor in first or last row, return either integar or string value from database,close resultset.
- In metdata You can entre column index and get column count,return column name , get tablename,return type of column either varchar or int .
- Any wrong input the program will display wrong message on screen.

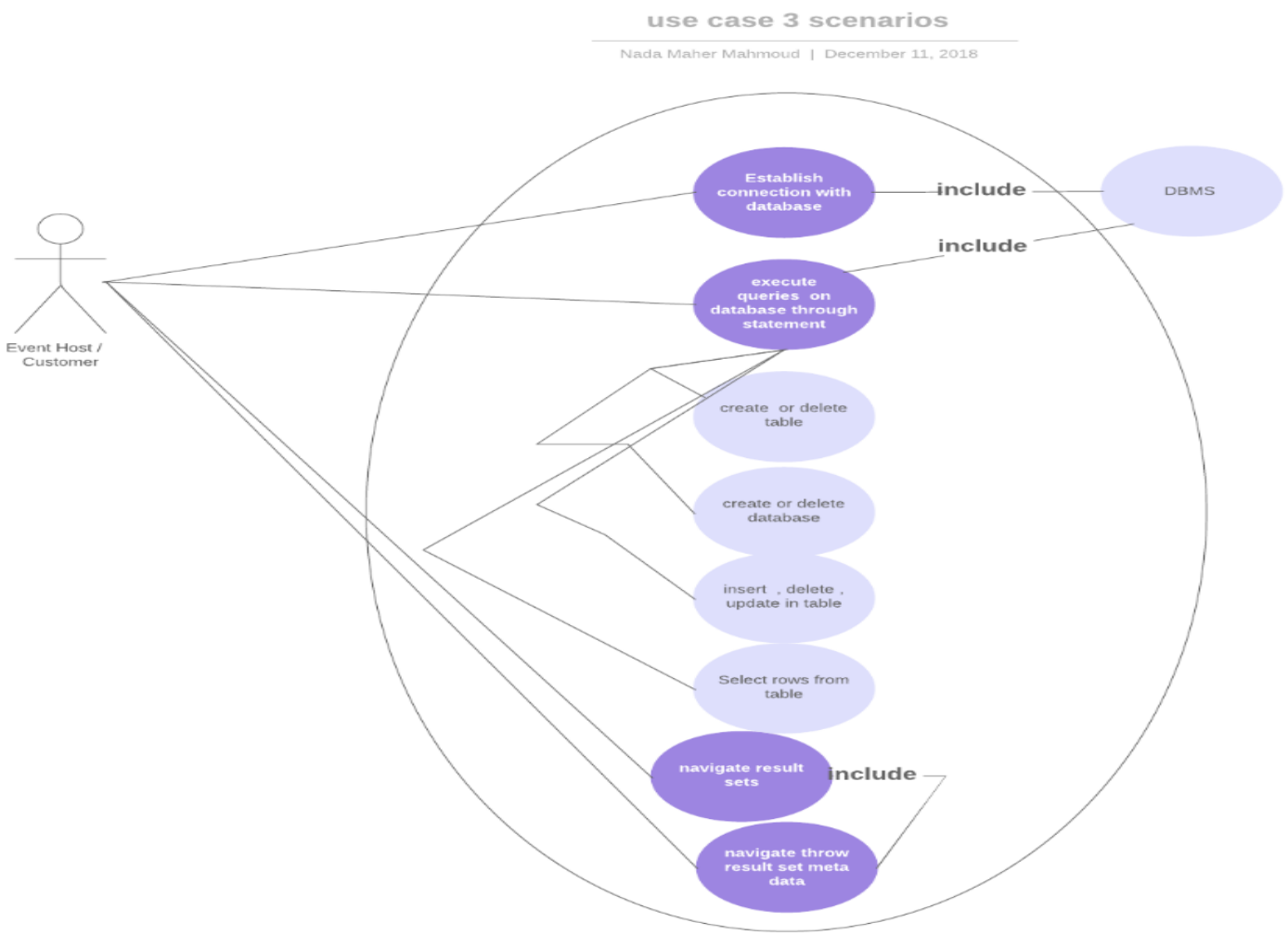
Division of labor among group members

- We divided the java.sql methods one member for each point:
    - Driver and connection
    - Statement
    - Result set
    - Result set metadata

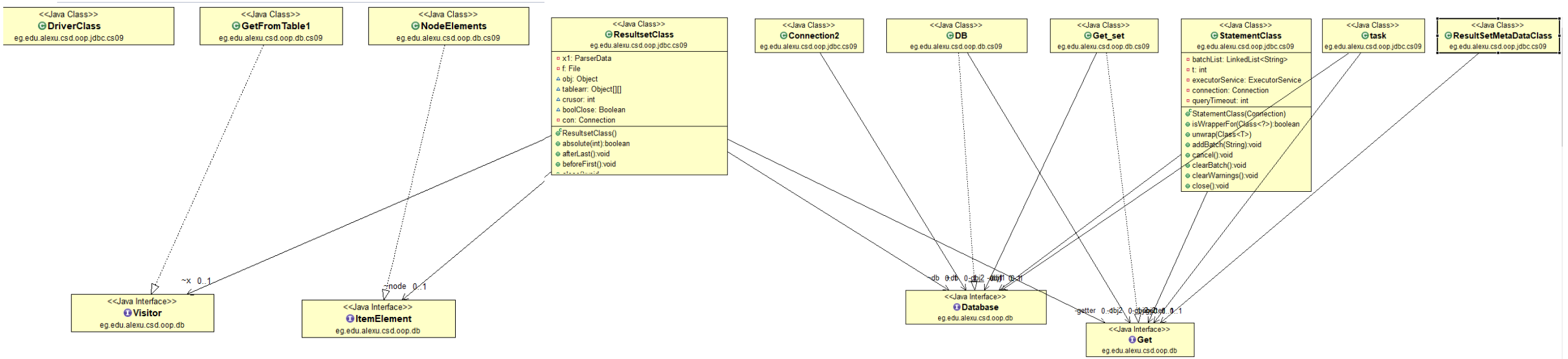
-Two members for loggers and configuration .

-Two members for GUI and time limit.

Use case


use case 3 scenarios
Nada Maher Mahmoud | December 11, 2018
Event Host / Customer
Establish connection with database
include
DBMS
include
execute queries on database through statement
create or delete table
create or delete database
insert , delete , update in table
Select rows from table
navigate result sets
include
navigate throw result set meta data

UML diagram


<<Java Class>>
DriverClass
eg.edu.alexu.csd.oop.jdbc.cs09
<<Java Class>>
GetFromTable1
eg.edu.alexu.csd.oop.db.cs09
<<Java Class>>
NodeElements
eg.edu.alexu.csd.oop.db.cs09
<<Java Class>>
ResultsetClass
eg.edu.alexu.csd.oop.jdbc.cs09
x1: ParserData
f: File
obj: Object
tablearr: Object[][]
crusor: int
boolClose: Boolean
con: Connection
ResultsetClass()
absolute(int):boolean
afterLast():void
beforeFirst():void
<<Java Class>>
Connection2
eg.edu.alexu.csd.oop.jdbc.cs09
<<Java Class>>
DB
eg.edu.alexu.csd.oop.db.cs09
<<Java Class>>
Get_set
eg.edu.alexu.csd.oop.db.cs09
<<Java Class>>
StatementClass
eg.edu.alexu.csd.oop.jdbc.cs09
batchList: LinkedList<String>
t: int
executorService: ExecutorService
connection: Connection
queryTimeout: int
StatementClass(Connection)
isWrapperFor(Class<?>):boolean
unwrap(Class<T>)
addBatch(String):void
cancel():void
clearBatch():void
clearWarnings():void
close():void
<<Java Class>>
task
eg.edu.alexu.csd.oop.jdbc.cs09
<<Java Class>>
ResultSetMetaDataClass
eg.edu.alexu.csd.oop.jdbc.cs09
~x 0..1
~node 0..1
<<Java Interface>>
Visitor
eg.edu.alexu.csd.oop.db
<<Java Interface>>
ItemElement
eg.edu.alexu.csd.oop.db
<<Java Interface>>
Database
eg.edu.alexu.csd.oop.db
getter 0..1
<<Java Interface>>
Get
eg.edu.alexu.csd.oop.db

User manual

1) Any instructions are given to the program in the form of SQL queries.

2) You must enter a query in a correct syntax in order for the program to perform the instruction you require, if syntax is wrong Error message will be displayed on screen.

2) You cannot create a database with an already existing name otherwise error will be shown on screen.

3) The user can Set limiting time for executing query .

4) The User can execute query, execute Batch,Add to Batch,get resultset.

5) The user can access the resultset (find column, get object(col index), get object(string colname), get string(String collabel), get string(col index),close, next, previous, first, last, afterfirst,beforelast,get int)

6) The result will be displayed on the label.

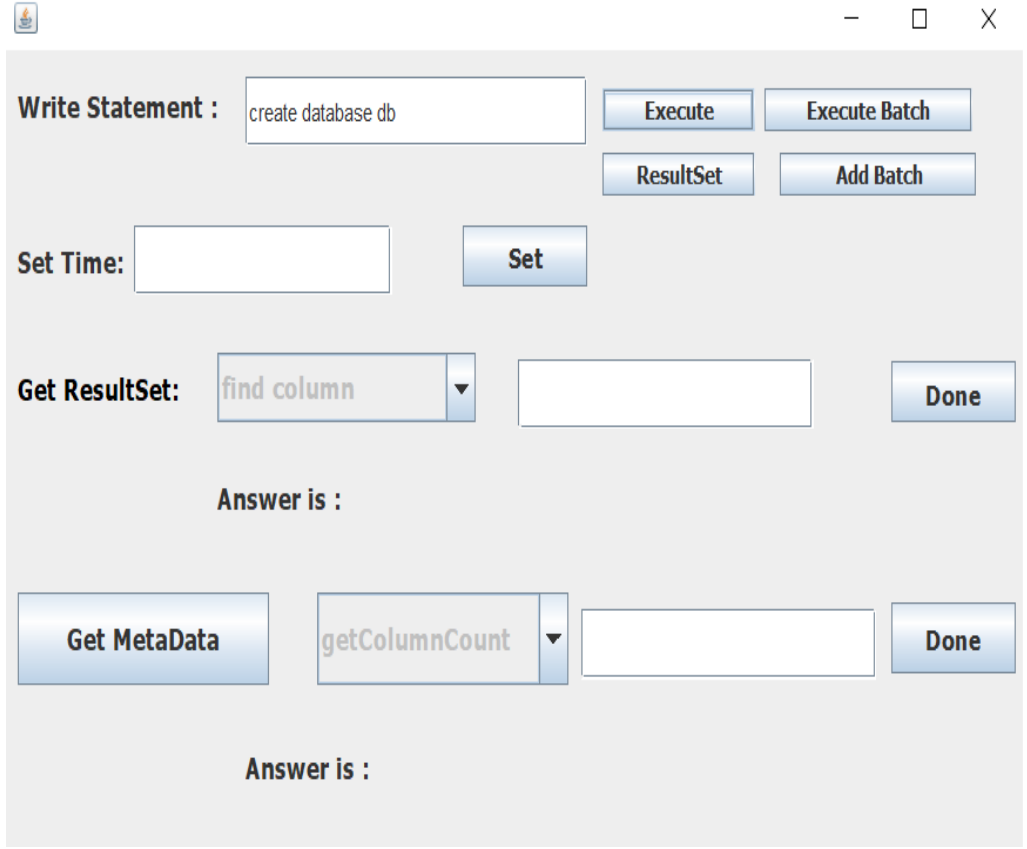
User manual

7) The User can access metadata.

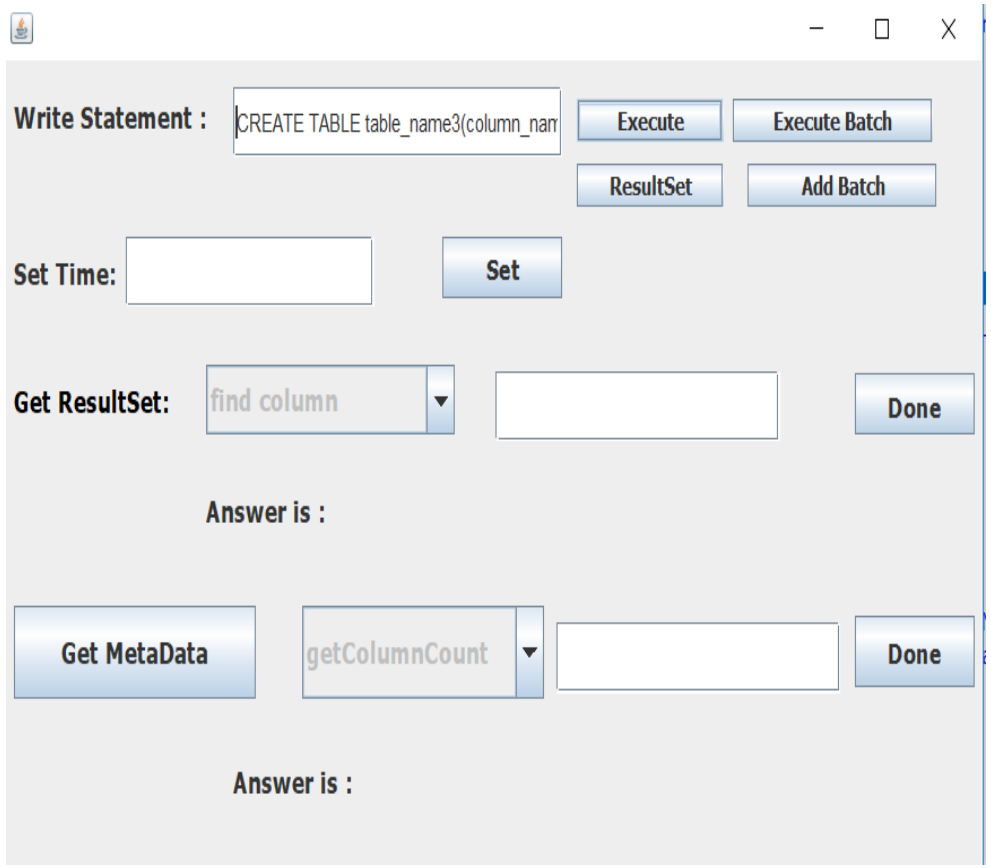
8)The User can (getColumnCount, getColumnLabel, getColumnName, getColumnType, ,getTableName) by entering column index.

9) The Result will be displayed on label on screen.

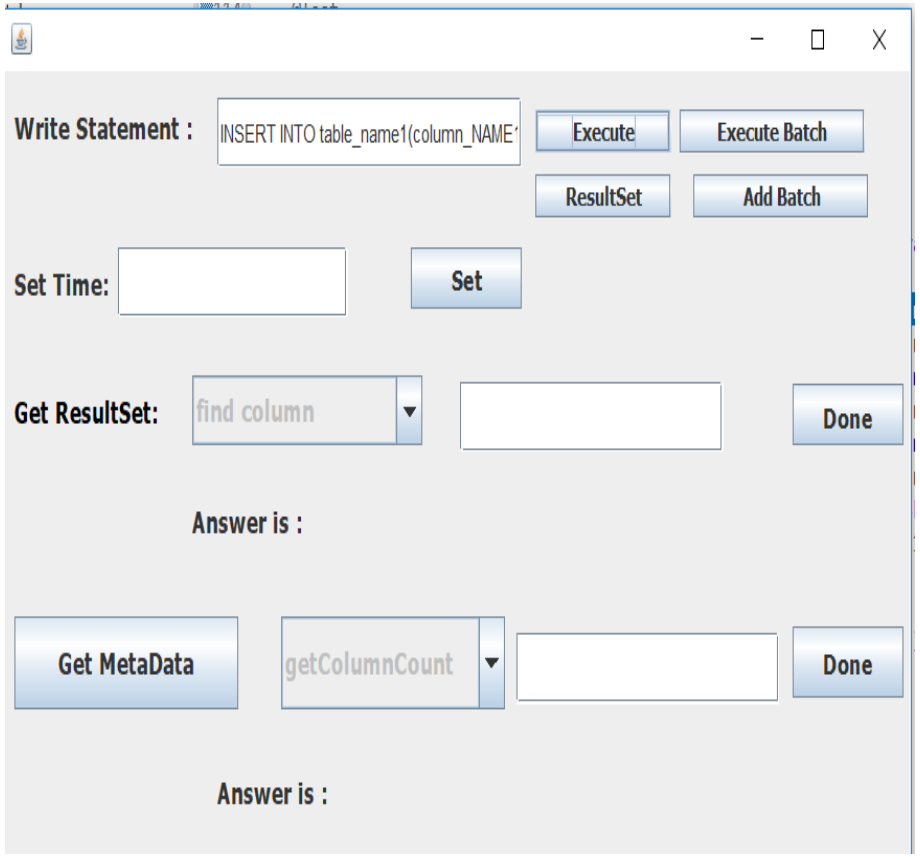
Sample runs



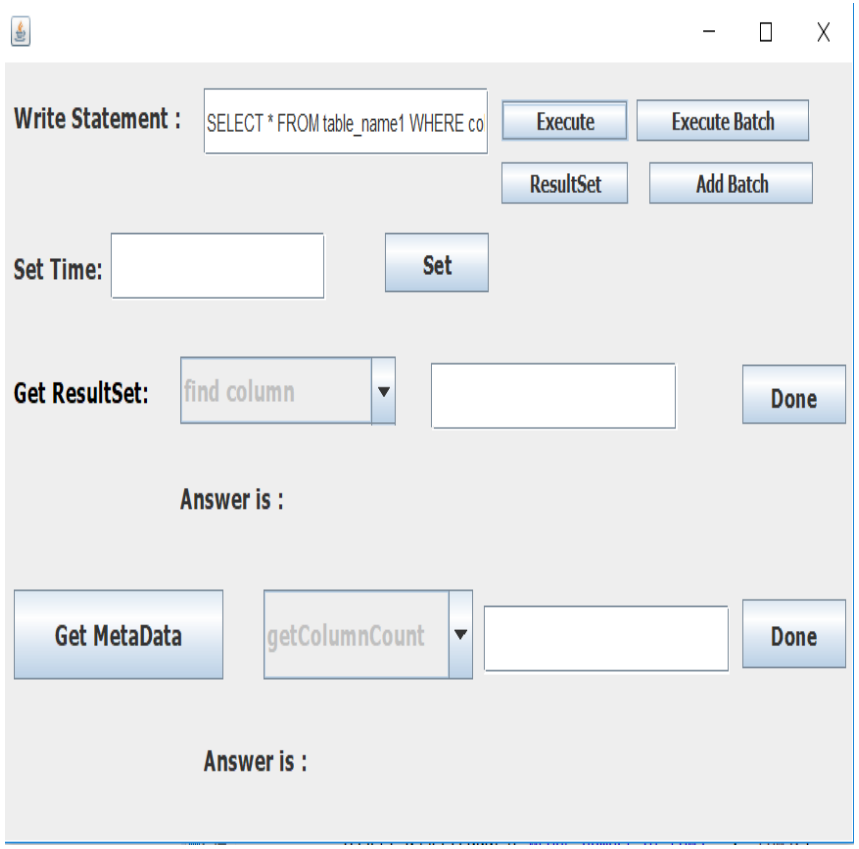
Create database



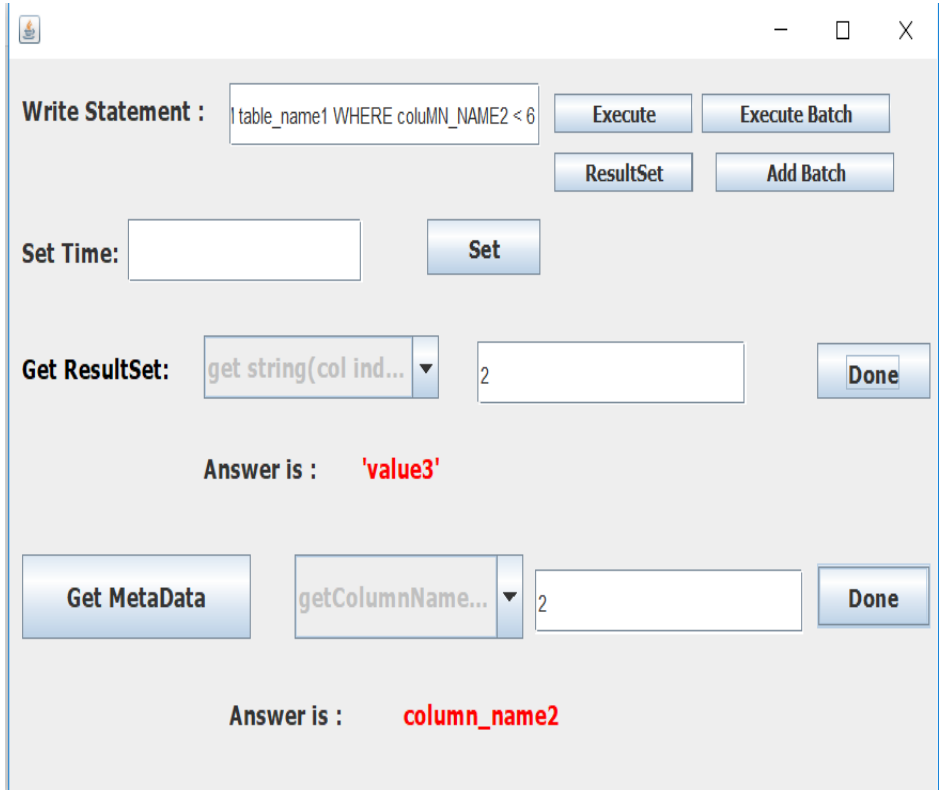
Create table



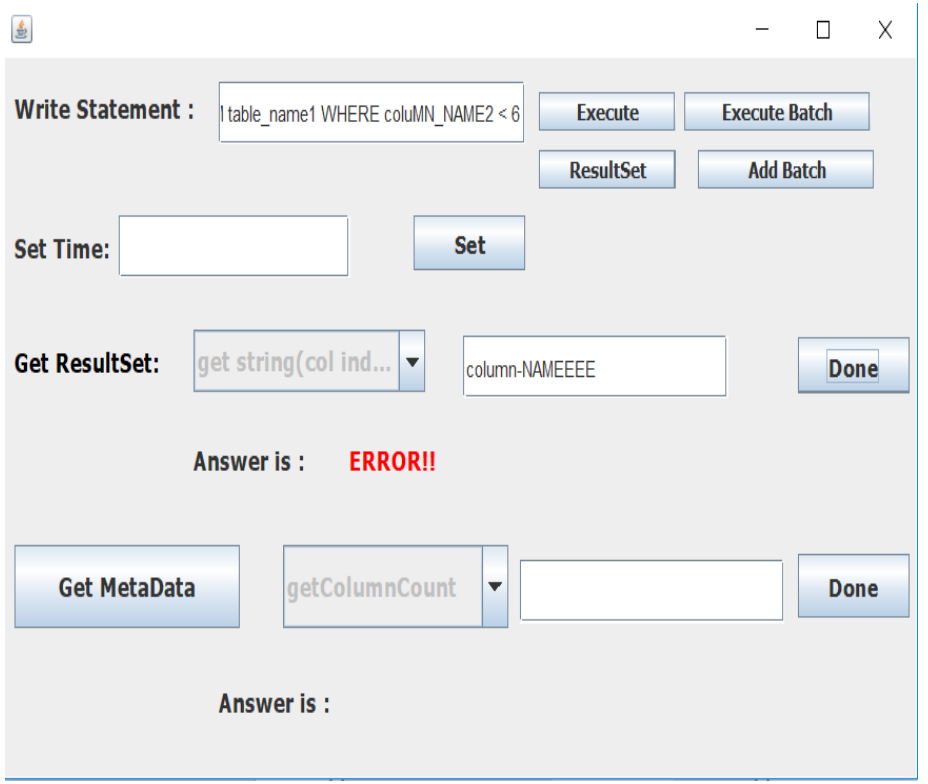
Insert into table_name1



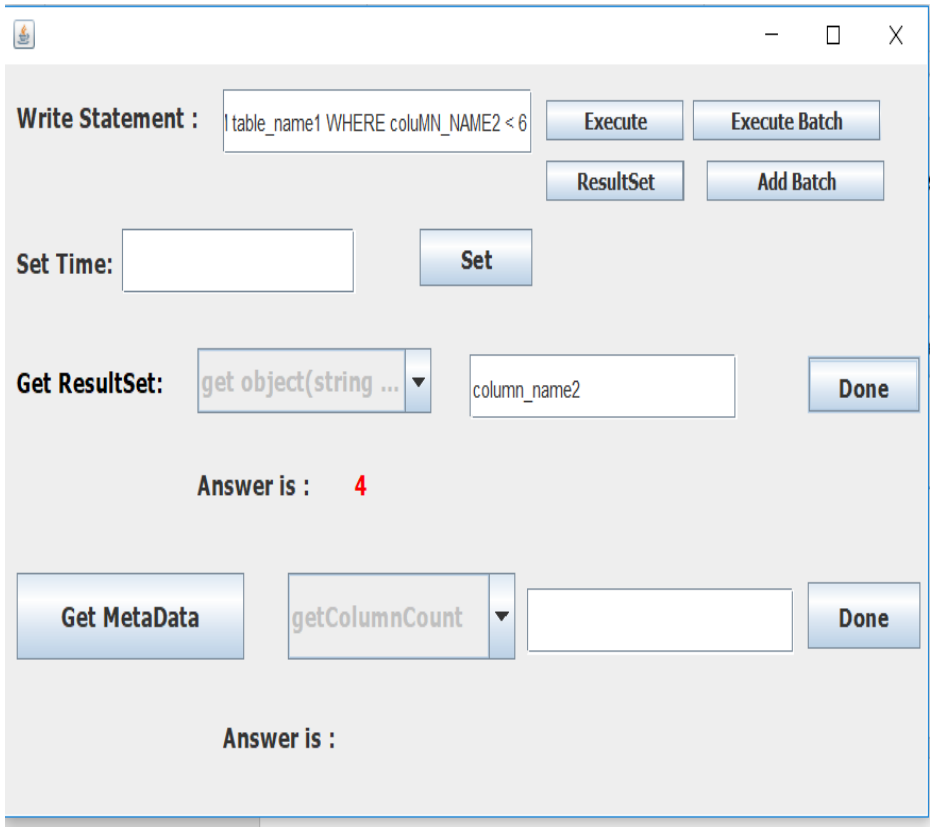
Select from table



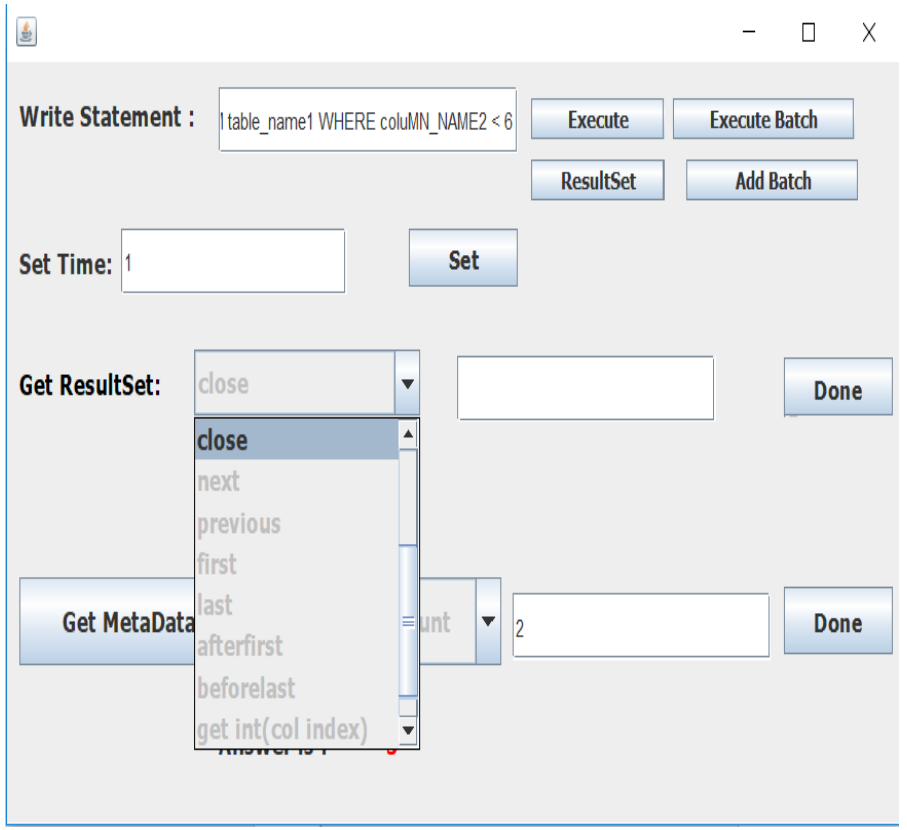
Get string from resultset by column index and get column name from metadata by column index



ERROR message displayed when entering wrong column name



Get object from resultset by column name



displaying functions of resultset by comboBox