User Guide:

1. When the system runs, you must enter URL for JDBC using a driver to connect.
2. Enter username and password .
3. Get connection to your database .
4. Create a statement using this connection .
5. Enter any SQL statement you need to execute.
6. The statement object will take your SQL statement and execute it.
7. Close your resultSet.
8. Close your statement.
9. Close your connection .

Assumptions:

- The system will keep reading a SQL query until ";" is entered.

Design Decisions:

1. When executing any query, data will be cached in an array list of linked hash map from the required file, manipulated, then written back to the same file.

2. There are these interfaces: "IFileWriter", "IFileReader", "QueryValidator", where "JSONWriter" and "XMLWriter" implement "IFileWriter", "JSONReader" and "XMLReader" implement "IFileReader" and "XMLValidator" and "JSONValidator" implement "QueryValidator"
   Using the **Bridge Design Pattern.**

3. Having configuration files for both loggers and CLI.

4. Encapsulation is achieved as the DBMS deals only with the database class which handles the requests to table class if needed.

UML:

Main
<<Java Class>>
eg.edu.alexu.csd.oop.jdbc

- loader: driverLoader
- driver: Driver
- username: String
- password: String
- protocol1: String
- protocol2: String
- Main()
- readConfig():void
- queryScanner(Scanner):String
- getUsername():String
- getPassword():String
- main(String[]):void

DBMS
<<Java Class>>
dbms

- db: DataBase
- homeDir: Path
- writer: IFileWriter
- reader: IFileReader
- fileValidator: FileValidator
- logger: Logs
- DBMS(int,Path)
- createDB(String):boolean
- useDB(String):boolean
- dropDB(String):boolean
- changeDBStructure(int,ArrayList<String>):boolean
- changeTableStructure(int,ArrayList<String>):void
- excuteUpdateQuery(int,ArrayList<String>):int

JStatement
<<Java Class>>
eg.edu.alexu.csd.oop.jdbc

- sqlV: SQLValidator
- query: ArrayList<String>
- batch: Queue<String>
- mode: int
- updated: int
- updateCount: int
- isClosed: boolean
- logger: Logs
- JStatement(JConnection)
- addBatch(String):void
- executeBatch():int[]
- clearBatch():void
- execute(String):boolean
- executeQuery(String):Re...
- executeUpdate(String):int
- getConnection():Connection
- getResultSet():ResultSet
- getUpdateCount():int
- close():void

JResultSet
<<Java Class>>
eg.edu.alexu.csd.oop.jdbc

- tableMap: ArrayList<LinkedHashMap<String,Strin...
- MetaData: ArrayList<LinkedHashMap<String,Stri...
- colMapping: ArrayList<String>
- cursor: int
- isClosed: boolean
- meta: LinkedHashMap<String,String>
- metaD: LinkedHashMap<String,String>
- JResultSet(ArrayList<LinkedHashMap<String,Stri...
- mapCols():void
- setStatement(JStatement):void
- getSize():int
- checkExistance():void
- checkType(String,String):void
- absolute(int):boolean
- isBeforeFirst():boolean
- isAfterLast():boolean
- isFirst():boolean
- isLast():boolean
- beforeFirst():void
- afterLast():void
- first():boolean
- last():boolean
- next():boolean
- previous():boolean
- getObject(int):Object
- getInt(int):int
- getInt(String):int
- getString(int):String
- getString(String):String
- getFloat(int):float
- getFloat(String):float
- getDate(int):Date
- getDate(String):Date
- findColumn(String):int

JDriver
<<Java Class>>
eg.edu.alexu.csd.oop.jdbc

- userName: String
- password: String
- protocolEntered: String
- JDriver()
- acceptsURL(String):boolean
- connect(String,Properties):Connection
- getMajorVersion():int
- getMinorVersion():int
- getParentLogger():Logger
- getPropertyInfo(String,Properties):DriverPropertyInfo[]
- jdbcCompliant():boolean

JConnection
<<Java Class>>
eg.edu.alexu.csd.oop.jdbc

- isClosed: boolean
- JConnection(DBMS)
- getDBMS():DBMS
- isWrapperFor(Class<?>):boolean
- unwrap(Class<T>)
- abort(Executor):void
- clearWarnings():void
- close():void
- commit():void
- createArrayOf(String,Object[]):Array
- createBlob():Blob
- createClob():Clob
- createNClob():NClob
- createSQLXML():SQLXML
- createStatement():Statement

JResultSetMetaData
<<Java Class>>
eg.edu.alexu.csd.oop.jdbc

- columnCount: int
- MetaData: ArrayList<LinkedHashMap<String,String>>
- colMapping: ArrayList<String>
- JResultSetMetaData(int,ArrayList<LinkedHashMap<String,String>>,ArrayList<String>)
- setResultSet(JResultSet):void
- getColumnCount():int
- getTableName(int):String
- getColumnType(int):int
- getColumnLabel(int):String

-dbms 0..1

dbms 0..1

0,1

-connection 0..1

-sqlStatement

0..1 -resultSet

-ResultSet 0..1