Assignment (4)
JDBC: Java Database Connectivity

Members:

Gamal Abdelhamid Ghanem (23)

Sherif Mohamed Abdelrahman (37)

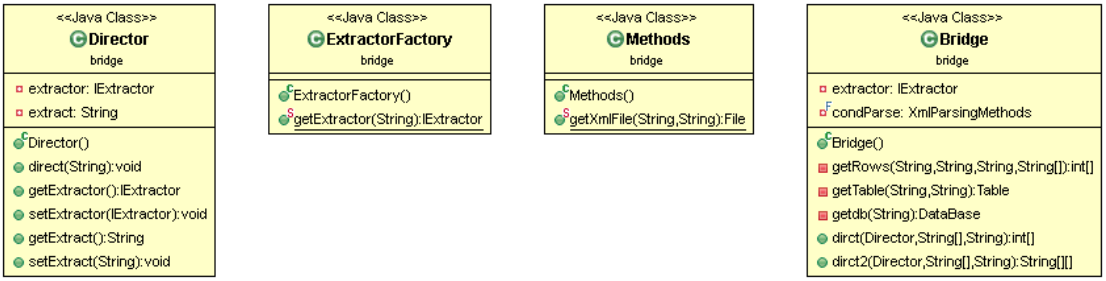
Shaban Sheta (38)

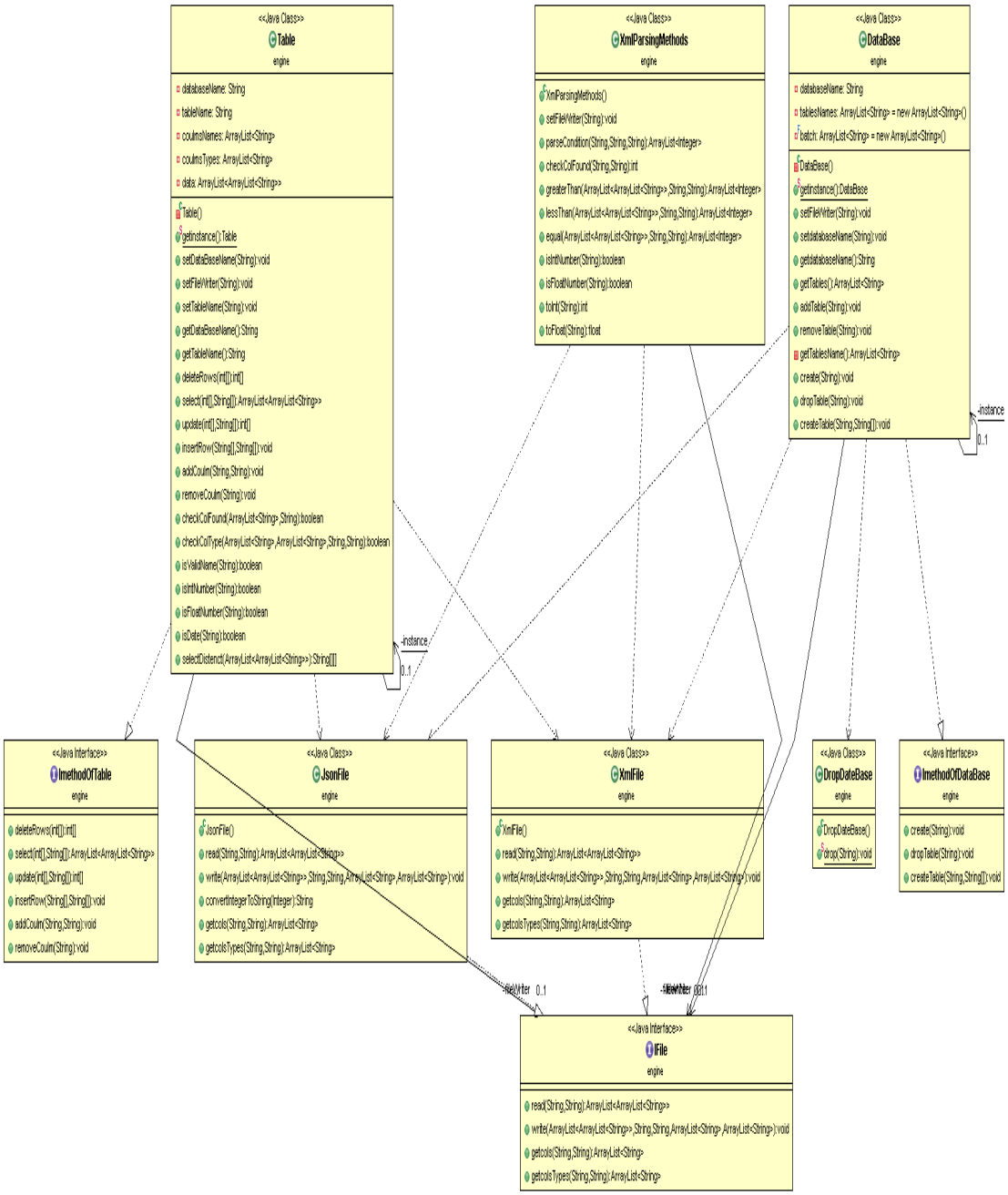
Omar Ahmed Wasfy(42)

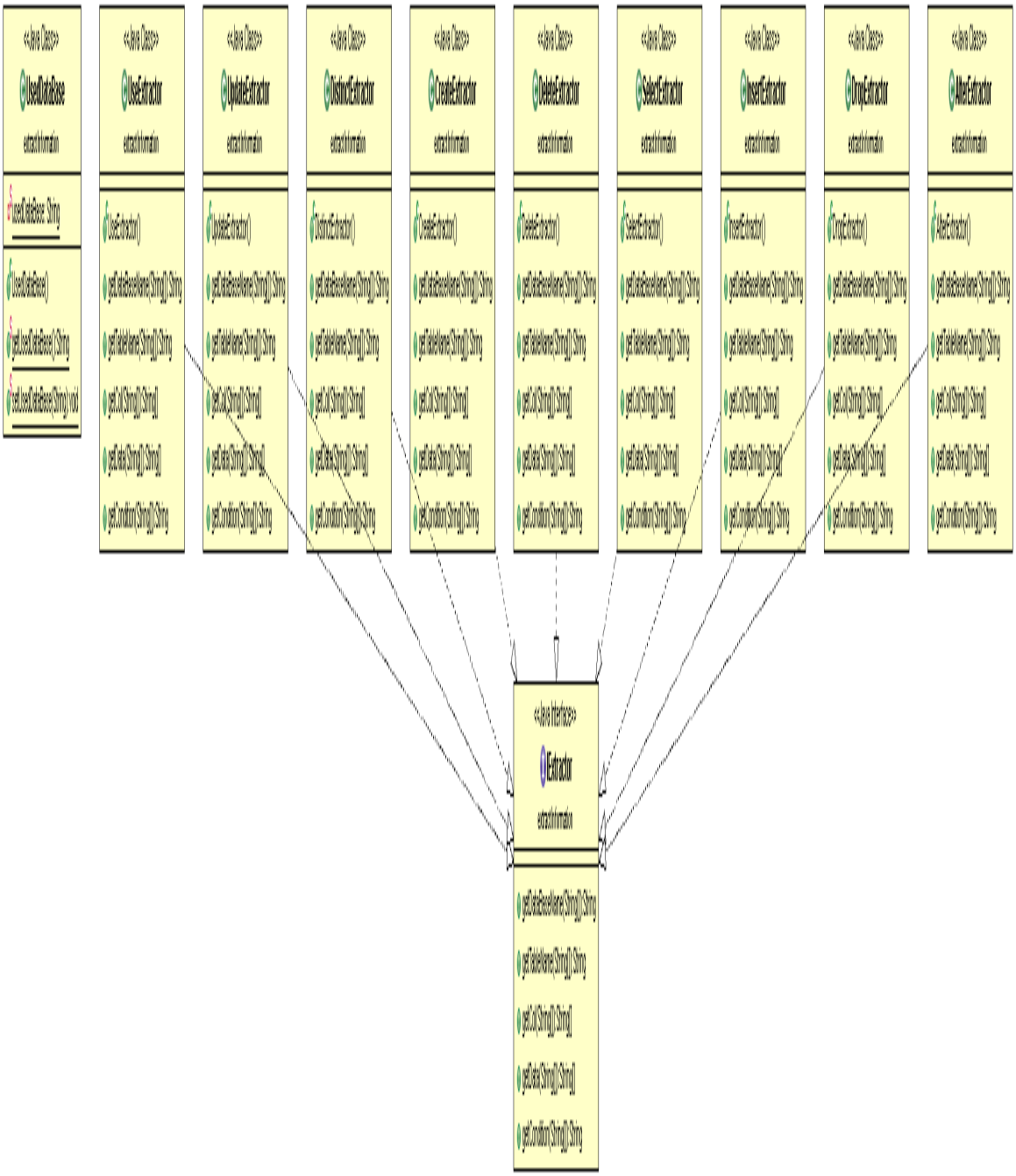
Mostafa Tarek (74)

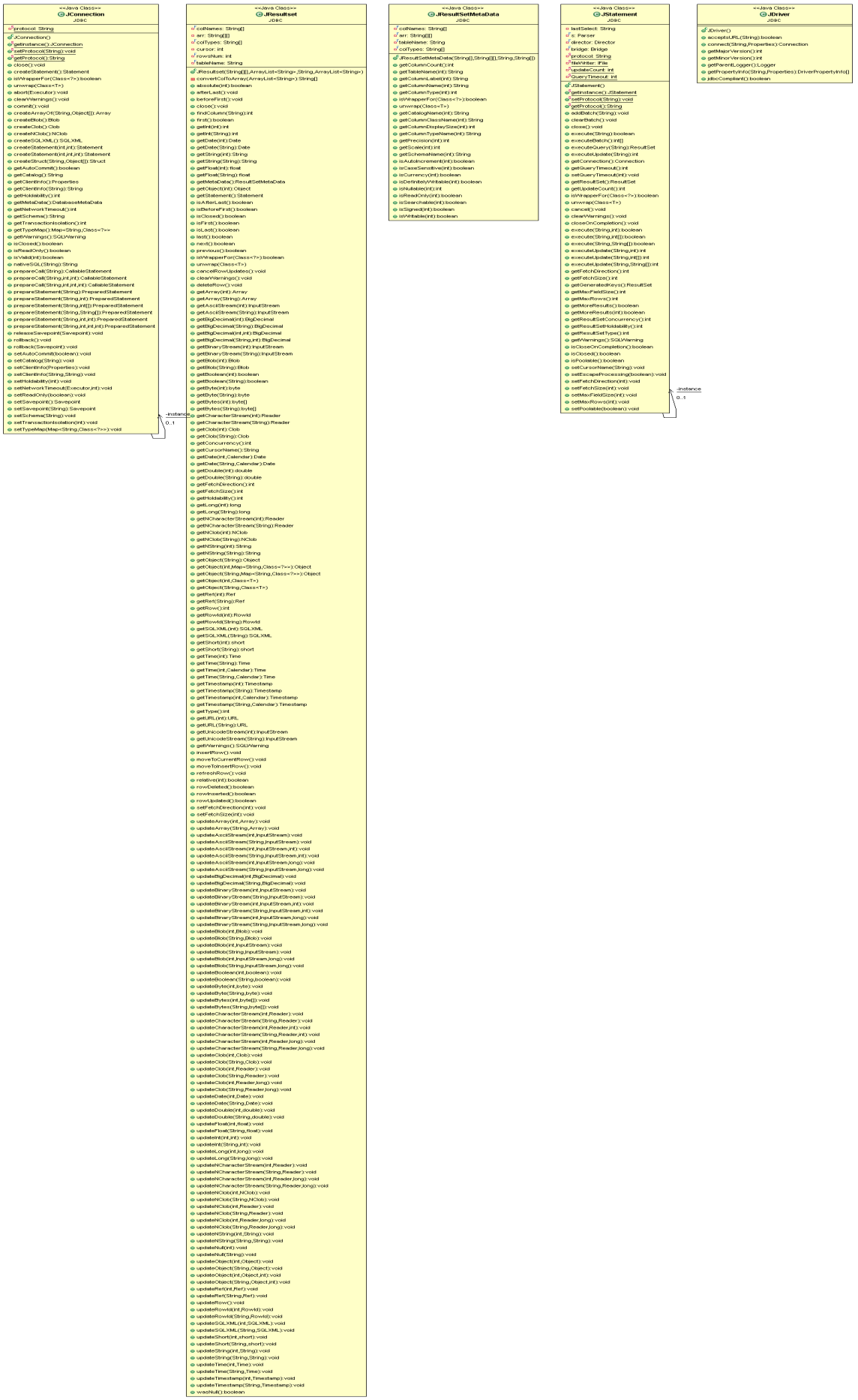
Contents:

- UML Diagram


«Java Class»
Director
bridge
extractor: IExtractor
extract: String
Director()
direct(String):void
getExtractor():IExtractor
setExtractor(IExtractor):void
getExtract():String
setExtract(String):void
«Java Class»
ExtractorFactory
bridge
ExtractorFactory()
getExtractor(String):IExtractor
«Java Class»
Methods
bridge
Methods()
getXmlFile(String,String):File
«Java Class»
Bridge
bridge
extractor: IExtractor
condParse: XmlParsingMethods
Bridge()
getRows(String,String,String,String[]):int[]
getTable(String,String):Table
getdb(String):DataBase
dirct(Director,String[],String):int[]
dirct2(Director,String[],String):String[][]


«Java Class»
Table
engine
databaseName: String
tableName: String
coulmsNames: ArrayList<String>
coulmsTypes: ArrayList<String>
data: ArrayList<ArrayList<String>>
Table()
getInstance():Table
setDataBaseName(String):void
setFileWriter(String):void
setTableName(String):void
getDataBaseName():String
getTableName():String
deleteRows(int[]):int[]
select(int[],String[]):ArrayList<ArrayList<String>>
update(int[],String[]):int[]
insertRow(String[],String[]):void
addCoulm(String,String):void
removeCoulm(String):void
checkColFound(ArrayList<String>,String):boolean
checkColType(ArrayList<String>,ArrayList<String>,String,String[]):boolean
isValidName(String):boolean
isIntNumber(String):boolean
isFloatNumber(String):boolean
isDate(String):boolean
selectDistenct(ArrayList<ArrayList<String>>):String[][]
«Java Class»
XmlParsingMethods
engine
XmlParsingMethods()
setFileWriter(String):void
parseCondition(String,String,String):ArrayList<Integer>
checkColFound(String,String):int
greaterThan(ArrayList<ArrayList<String>>,String,String):ArrayList<Integer>
lessThan(ArrayList<ArrayList<String>>,String,String):ArrayList<Integer>
equal(ArrayList<ArrayList<String>>,String,String):ArrayList<Integer>
isIntNumber(String):boolean
isFloatNumber(String):boolean
toInt(String):int
toFloat(String):float
«Java Class»
DataBase
engine
databaseName: String
tablesNames: ArrayList<String> = new ArrayList<String>()
batch: ArrayList<String> = new ArrayList<String>()
DataBase()
getInstance():DataBase
setFileWriter(String):void
setdatabaseName(String):void
getdatabaseName():String
getTables():ArrayList<String>
addTable(String):void
removeTable(String):void
getTablesName():ArrayList<String>
create(String):void
dropTable(String):void
createTable(String,String[]):void
-instance
0..1
-instance
0..1
«Java Interface»
ImethodOfTable
engine
deleteRows(int[]):int[]
select(int[],String[]):ArrayList<ArrayList<String>>
update(int[],String[]):int[]
insertRow(String[],String[]):void
addCoulm(String,String):void
removeCoulm(String):void
«Java Class»
JsonFile
engine
JsonFile()
read(String,String):ArrayList<ArrayList<String>>
write(ArrayList<ArrayList<String>>,String,String,ArrayList<String>,ArrayList<String>):void
convertIntegerToString(Integer):String
getcols(String,String):ArrayList<String>
getcolsTypes(String,String):ArrayList<String>
«Java Class»
XmlFile
engine
XmlFile()
read(String,String):ArrayList<ArrayList<String>>
write(ArrayList<ArrayList<String>>,String,String,ArrayList<String>,ArrayList<String>):void
getcols(String,String):ArrayList<String>
getcolsTypes(String,String):ArrayList<String>
«Java Class»
DropDateBase
engine
DropDateBase()
drop(String):void
«Java Interface»
ImethodOfDataBase
engine
create(String):void
dropTable(String):void
createTable(String,String[]):void
fileWriter 0..1
fileWriter 0..1
«Java Interface»
IFile
engine
read(String,String):ArrayList<ArrayList<String>>
write(ArrayList<ArrayList<String>>,String,String,ArrayList<String>,ArrayList<String>):void
getcols(String,String):ArrayList<String>
getcolsTypes(String,String):ArrayList<String>


«Java Class»
UseDatabase
extractInformation
useDatabase: String
UseDatabase()
getUseDatabase():String
setUseDatabase(String):void
«Java Class»
UseExtractor
extractInformation
UseExtractor()
getTableRelease(String[]):String
getTableName(String[]):String
getJoin(String[]):String[]
getData(String[]):String[]
getCondition(String[]):String
«Java Class»
UpdateExtractor
extractInformation
UpdateExtractor()
getTableRelease(String[]):String
getTableName(String[]):String
getSet(String[]):String[]
getData(String[]):String[]
getCondition(String[]):String
«Java Class»
DistinctExtractor
extractInformation
DistinctExtractor()
getTableRelease(String[]):String
getTableName(String[]):String
getJoin(String[]):String[]
getData(String[]):String[]
getCondition(String[]):String
«Java Class»
CreateExtractor
extractInformation
CreateExtractor()
getTableRelease(String[]):String
getTableName(String[]):String
getJoin(String[]):String[]
getData(String[]):String[]
getCondition(String[]):String
«Java Class»
DeleteExtractor
extractInformation
DeleteExtractor()
getTableRelease(String[]):String
getTableName(String[]):String
getJoin(String[]):String[]
getData(String[]):String[]
getCondition(String[]):String
«Java Class»
SelectExtractor
extractInformation
SelectExtractor()
getTableRelease(String[]):String
getTableName(String[]):String
getJoin(String[]):String[]
getData(String[]):String[]
getCondition(String[]):String
«Java Class»
InsertExtractor
extractInformation
InsertExtractor()
getTableRelease(String[]):String
getTableName(String[]):String
getJoin(String[]):String[]
getData(String[]):String[]
getCondition(String[]):String
«Java Class»
DropExtractor
extractInformation
DropExtractor()
getTableRelease(String[]):String
getTableName(String[]):String
getJoin(String[]):String[]
getData(String[]):String[]
getCondition(String[]):String
«Java Class»
AlterExtractor
extractInformation
AlterExtractor()
getTableRelease(String[]):String
getTableName(String[]):String
getJoin(String[]):String[]
getData(String[]):String[]
getCondition(String[]):String
«Java Interface»
IExtractor
extractInformation
getTableRelease(String[]):String
getTableName(String[]):String
getJoin(String[]):String[]
getData(String[]):String[]
getCondition(String[]):String


«Java Class»
JConnection
JDBC
protocol: String
JConnection()
getInstance():JConnection
setProtocol(String):void
getProtocol():String
close():void
createStatement():Statement
isWrapperFor(Class<?>):boolean
unwrap(Class<T>)
abort(Executor):void
clearWarnings():void
commit():void
createArrayOf(String,Object[]):Array
createBlob():Blob
createClob():Clob
createNClob():NClob
createSQLXML():SQLXML
createStatement(int,int):Statement
createStatement(int,int,int):Statement
createStruct(String,Object[]):Struct
getAutoCommit():boolean
getCatalog():String
getClientInfo():Properties
getClientInfo(String):String
getHoldability():int
getMetaData():DatabaseMetaData
getNetworkTimeout():int
getSchema():String
getTransactionIsolation():int
getTypeMap():Map<String,Class<?>>
getWarnings():SQLWarning
isClosed():boolean
isReadOnly():boolean
isValid(int):boolean
nativeSQL(String):String
prepareCall(String):CallableStatement
prepareCall(String,int,int):CallableStatement
prepareCall(String,int,int,int):CallableStatement
prepareStatement(String):PreparedStatement
prepareStatement(String,int):PreparedStatement
prepareStatement(String,int[]):PreparedStatement
prepareStatement(String,int,int):PreparedStatement
prepareStatement(String,String[]):PreparedStatement
prepareStatement(String,int,int,int):PreparedStatement
prepareStatement(String,int,int,int):PreparedStatement
releaseSavepoint(Savepoint):void
rollback():void
rollback(Savepoint):void
setAutoCommit(boolean):void
setCatalog(String):void
setClientInfo(Properties):void
setClientInfo(String,String):void
setHoldability(int):void
setNetworkTimeout(Executor,int):void
setReadOnly(boolean):void
setSavepoint():Savepoint
setSavepoint(String):Savepoint
setSchema(String):void
setTransactionIsolation(int):void
setTypeMap(Map<String,Class<?>>):void
-instance
0..1
«Java Class»
JResultSet
JDBC
colNames: String[]
arr: String[][]
colTypes: String[]
cursor: int
rowsNum: int
tableName: String
JResultSet(String[][],ArrayList<String>,String,ArrayList<String>)
convertColToArray(ArrayList<String>):String[]
absolute(int):boolean
afterLast():void
beforeFirst():void
close():void
findColumn(String):int
first():boolean
getInt(int):int
getInt(String):int
getDate(int):Date
getDate(String):Date
getString(int):String
getString(String):String
getFloat(int):float
getFloat(String):float
getMetaData():ResultSetMetaData
getObject(int):Object
getStatement():Statement
isAfterLast():boolean
isBeforeFirst():boolean
isClosed():boolean
isFirst():boolean
isLast():boolean
last():boolean
next():boolean
previous():boolean
isWrapperFor(Class<?>):boolean
unwrap(Class<T>)
cancelRowUpdates():void
clearWarnings():void
deleteRow():void
getArray(int):Array
getArray(String):Array
getAsciiStream(int):InputStream
getAsciiStream(String):InputStream
getBigDecimal(int):BigDecimal
getBigDecimal(String):BigDecimal
getBigDecimal(int,int):BigDecimal
getBigDecimal(String,int):BigDecimal
getBinaryStream(int):InputStream
getBinaryStream(String):InputStream
getBlob(int):Blob
getBlob(String):Blob
getBoolean(int):boolean
getBoolean(String):boolean
getByte(int):byte
getByte(String):byte
getBytes(int):byte[]
getBytes(String):byte[]
getCharacterStream(int):Reader
getCharacterStream(String):Reader
getClob(int):Clob
getClob(String):Clob
getConcurrency():int
getCursorName():String
getDate(int,Calendar):Date
getDate(String,Calendar):Date
getDouble(int):double
getDouble(String):double
getFetchDirection():int
getFetchSize():int
getHoldability():int
getLong(int):long
getLong(String):long
getNCharacterStream(int):Reader
getNCharacterStream(String):Reader
getNClob(int):NClob
getNClob(String):NClob
getNString(int):String
getNString(String):String
getObject(String):Object
getObject(int,Map<String,Class<?>>):Object
getObject(String,Map<String,Class<?>>):Object
getObject(int,Class<T>)
getObject(String,Class<T>)
getRef(int):Ref
getRef(String):Ref
getRow():int
getRowId(int):RowId
getRowId(String):RowId
getSQLXML(int):SQLXML
getSQLXML(String):SQLXML
getShort(int):short
getShort(String):short
getTime(int):Time
getTime(String):Time
getTime(int,Calendar):Time
getTime(String,Calendar):Time
getTimestamp(int):Timestamp
getTimestamp(String):Timestamp
getTimestamp(int,Calendar):Timestamp
getTimestamp(String,Calendar):Timestamp
getType():int
getURL(int):URL
getURL(String):URL
getUnicodeStream(int):InputStream
getUnicodeStream(String):InputStream
getWarnings():SQLWarning
insertRow():void
moveToCurrentRow():void
moveToInsertRow():void
refreshRow():void
relative(int):boolean
rowDeleted():boolean
rowInserted():boolean
rowUpdated():boolean
setFetchDirection(int):void
setFetchSize(int):void
updateArray(int,Array):void
updateArray(String,Array):void
updateAsciiStream(int,InputStream):void
updateAsciiStream(String,InputStream):void
updateAsciiStream(int,InputStream,int):void
updateAsciiStream(String,InputStream,int):void
updateAsciiStream(int,InputStream,long):void
updateAsciiStream(String,InputStream,long):void
updateBigDecimal(int,BigDecimal):void
updateBigDecimal(String,BigDecimal):void
updateBinaryStream(int,InputStream):void
updateBinaryStream(String,InputStream):void
updateBinaryStream(int,InputStream,int):void
updateBinaryStream(String,InputStream,int):void
updateBinaryStream(int,InputStream,long):void
updateBinaryStream(String,InputStream,long):void
updateBlob(int,Blob):void
updateBlob(String,Blob):void
updateBlob(int,InputStream):void
updateBlob(String,InputStream):void
updateBlob(int,InputStream,long):void
updateBlob(String,InputStream,long):void
updateBoolean(int,boolean):void
updateBoolean(String,boolean):void
updateByte(int,byte):void
updateByte(String,byte):void
updateBytes(int,byte[]):void
updateBytes(String,byte[]):void
updateCharacterStream(int,Reader):void
updateCharacterStream(String,Reader):void
updateCharacterStream(int,Reader,int):void
updateCharacterStream(String,Reader,int):void
updateCharacterStream(int,Reader,long):void
updateCharacterStream(String,Reader,long):void
updateClob(int,Clob):void
updateClob(String,Clob):void
updateClob(int,Reader):void
updateClob(String,Reader):void
updateClob(int,Reader,long):void
updateClob(String,Reader,long):void
updateDate(int,Date):void
updateDate(String,Date):void
updateDouble(int,double):void
updateDouble(String,double):void
updateFloat(int,float):void
updateFloat(String,float):void
updateInt(int,int):void
updateInt(String,int):void
updateLong(int,long):void
updateLong(String,long):void
updateNCharacterStream(int,Reader):void
updateNCharacterStream(String,Reader):void
updateNCharacterStream(int,Reader,long):void
updateNCharacterStream(String,Reader,long):void
updateNClob(int,NClob):void
updateNClob(String,NClob):void
updateNClob(int,Reader):void
updateNClob(String,Reader):void
updateNClob(int,Reader,long):void
updateNClob(String,Reader,long):void
updateNString(int,String):void
updateNString(String,String):void
updateNull(int):void
updateNull(String):void
updateObject(int,Object):void
updateObject(String,Object):void
updateObject(int,Object,int):void
updateObject(String,Object,int):void
updateRef(int,Ref):void
updateRef(String,Ref):void
updateRow():void
updateRowId(int,RowId):void
updateRowId(String,RowId):void
updateSQLXML(int,SQLXML):void
updateSQLXML(String,SQLXML):void
updateShort(int,short):void
updateShort(String,short):void
updateString(int,String):void
updateString(String,String):void
updateTime(int,Time):void
updateTime(String,Time):void
updateTimestamp(int,Timestamp):void
updateTimestamp(String,Timestamp):void
wasNull():boolean
«Java Class»
JResultSetMetaData
JDBC
colNames: String[]
arr: String[][]
tableName: String
colTypes: String[]
JResultSetMetaData(String[],String[][],String,String[])
getColumnCount():int
getTableName(int):String
getColumnLabel(int):String
getColumnName(int):String
getColumnType(int):int
isWrapperFor(Class<?>):boolean
unwrap(Class<T>)
getCatalogName(int):String
getColumnClassName(int):String
getColumnDisplaySize(int):int
getColumnTypeName(int):String
getPrecision(int):int
getScale(int):int
getSchemaName(int):String
isAutoIncrement(int):boolean
isCaseSensitive(int):boolean
isCurrency(int):boolean
isDefinitelyWritable(int):boolean
isNullable(int):int
isReadOnly(int):boolean
isSearchable(int):boolean
isSigned(int):boolean
isWritable(int):boolean
«Java Class»
JStatement
JDBC
lastSelect: String
s: Parser
director: Director
bridge: Bridge
protocol: String
fileWriter: PrWe
updateCount: int
queryTimeout: int
JStatement()
getInstance():JStatement
setProtocol(String):void
getProtocol():String
addBatch(String):void
clearBatch():void
close():void
execute(String):boolean
executeBatch():int[]
executeQuery(String):ResultSet
executeUpdate(String):int
getConnection():Connection
getQueryTimeout():int
setQueryTimeout(int):void
getResultSet():ResultSet
getUpdateCount():int
isWrapperFor(Class<?>):boolean
unwrap(Class<T>)
cancel():void
clearWarnings():void
closeOnCompletion():void
execute(String,int):boolean
execute(String,int[]):boolean
execute(String,String[]):boolean
executeUpdate(String,int):int
executeUpdate(String,int[]):int
executeUpdate(String,String[]):int
getFetchDirection():int
getFetchSize():int
getGeneratedKeys():ResultSet
getMaxFieldSize():int
getMaxRows():int
getMoreResults():boolean
getMoreResults(int):boolean
getResultSetConcurrency():int
getResultSetHoldability():int
getResultSetType():int
getWarnings():SQLWarning
isCloseOnCompletion():boolean
isClosed():boolean
isPoolable():boolean
setCursorName(String):void
setEscapeProcessing(boolean):void
setFetchDirection(int):void
setFetchSize(int):void
setMaxFieldSize(int):void
setMaxRows(int):void
setPoolable(boolean):void
-instance
0..1
«Java Class»
JDriver
JDBC
JDriver()
acceptsURL(String):boolean
connect(String,Properties):Connection
getMajorVersion():int
getMinorVersion():int
getParentLogger():Logger
getPropertyInfo(String,Properties):DriverPropertyInfo[]
jdbcCompliant():boolean

Design Description

The whole project now is consisted of two main parts; DBMS and JDBC part.

The DBMS's design and functionalities were accomplished in the last phase but we had to improve the design to enable editing and adding new features to the project like saving in json files instead of xml and adding new sql queries. The new design of DBMS is indirectly consisted of two main parts:

i. The parser which validates the input queries and extracts the data from these queries to pass it to table functions.
ii. The functionality part which also has two main parts:
    i. The file input/output part which is an interface containing reading and writing methods and two classes implementing this interface in Jason and xml format depending on the user's desire.
    ii. The table functions which operate on an input 2D array list as the table data. This array is returned from the 'readFile' method of file interface so these functions are independent of the file format we are reading or writing to.

The JDBC contains five classes implementing the main 5 interfaces of JDBC:

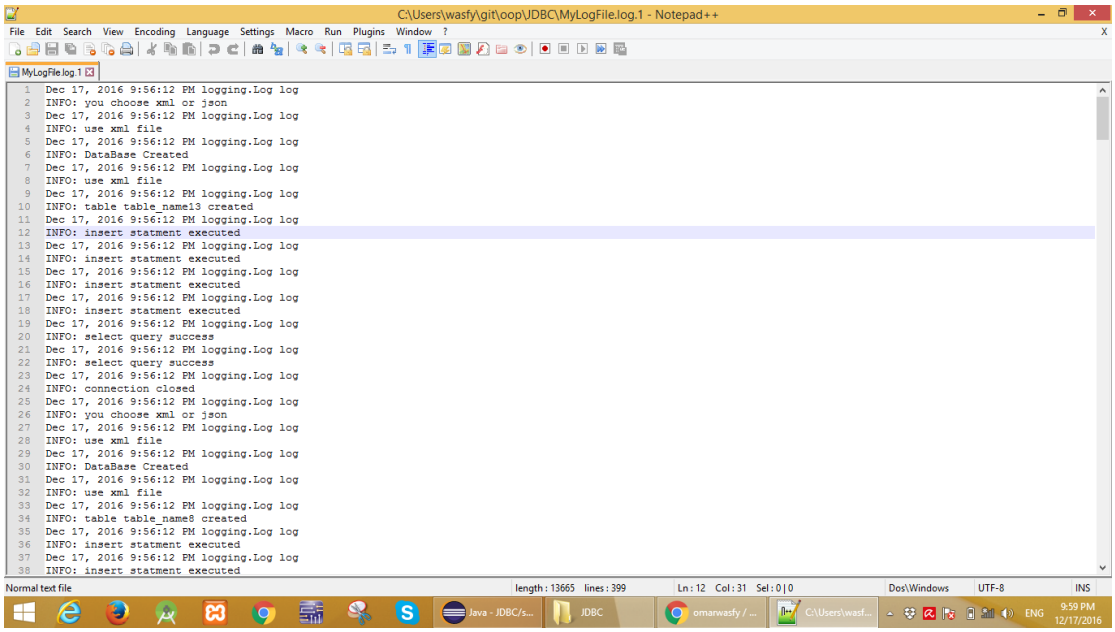
- **java.sql.Drive**: validates URL and creates a connection which is xml or Json based.
- **java.sql.Connection:** implements some of Connection interface methods.
- **java.sql.Statement:** implements some of statement interface methods
- **java.sql.Resultset:** enables getting and manipulating data of returned rows from the select query.
- **java.sql.ResultSetMetaData:** provides the information about the result set like number of columns and types of stored data.

Decisions and Assumptions

- There are no assumptions related to the functionalities due to the existence of the online tester so we were to follow specified criteria.
- We just used 'info' and 'error' in logging operations.

- snapshots:

  Provided here a screen shot of the logs' file



```
Dec 17, 2016 9:56:12 PM logging.Log log
INFO: you choose xml or json
Dec 17, 2016 9:56:12 PM logging.Log log
INFO: use xml file
Dec 17, 2016 9:56:12 PM logging.Log log
INFO: DataBase Created
Dec 17, 2016 9:56:12 PM logging.Log log
INFO: use xml file
Dec 17, 2016 9:56:12 PM logging.Log log
INFO: table table_name13 created
Dec 17, 2016 9:56:12 PM logging.Log log
INFO: insert statment executed
Dec 17, 2016 9:56:12 PM logging.Log log
INFO: insert statment executed
Dec 17, 2016 9:56:12 PM logging.Log log
INFO: insert statment executed
Dec 17, 2016 9:56:12 PM logging.Log log
INFO: insert statment executed
Dec 17, 2016 9:56:12 PM logging.Log log
INFO: select query success
Dec 17, 2016 9:56:12 PM logging.Log log
INFO: select query success
Dec 17, 2016 9:56:12 PM logging.Log log
INFO: connection closed
Dec 17, 2016 9:56:12 PM logging.Log log
INFO: you choose xml or json
Dec 17, 2016 9:56:12 PM logging.Log log
INFO: use xml file
Dec 17, 2016 9:56:12 PM logging.Log log
INFO: DataBase Created
Dec 17, 2016 9:56:12 PM logging.Log log
INFO: use xml file
Dec 17, 2016 9:56:12 PM logging.Log log
INFO: table table_name8 created
Dec 17, 2016 9:56:12 PM logging.Log log
INFO: insert statment executed
Dec 17, 2016 9:56:12 PM logging.Log log
INFO: insert statment executed
```