Mini DBMS specification

The mini relational DBMS system project is for **2 students**.

You will have to implement your own Database Management System (DBMS). It will have a server and a client component, communicating by a communication protocol.

The client can be a visual interface like SQL Developer (of Oracle) or Management Studio of MS SQL Server. You can choose to implement the client in command line too. For command line solution you have to validate the correctness of the statements: use lexical and syntactical analyzer, also regular expressions.

The server will execute typical SQL statements: Create Database, Create Table, INSERT, DELETE, SELECT, etc. and send a message to the client about the success of failure of SQL statement execution. In case of the SELECT statement, the server will send the result rows of the statement to the client. The tables of the system will be stored in a collection file in MongoDB, also the index files.

Lab1. 14p

You have to implement the next facilities:

```
create database, create table, drop database, drop table, create index.
```

You can create one visual interface as in MS SQL Server Management Studio or you can execute the statements from command line.

The structure of the tables, primary key constraints, and foreign key constraints, index file key and name will be stored in an xml or json file. See detailed description in Key-valueSystemsIndex.pdf and examples for system catalog in DataBases.xml and Catalog.xml. Implement only the structure of the tables. Inserts in tables will be implemented in Lab2.

Lab2. 14p One relational table will be stored in a collection file; one row of a table will be stored in a document with key, value structure. In the Key part the primary key will be stored, in the Value part the rest of the record.

Implement a Visual Query Designer for INSERT, DELETE, or execute it from command line. Implement INSERT, DELETE in a relational table. The value of the primary key will be stored in the Key part, the other attributes in the Value part concatenated as strings.

Ex: Let be a relational table Students(StudID, Name, Tel, email) and a row
(12345, 'Rusu Mihai', '0745123456','mrusu@gmail.com')
This row will be stored as follows:
Key: 12345
Value:'Rusu Mihai#0745123456#mrusu@gmail.com'

Lab3. **14p** Implement index files for unique and non-unique keys. Validate the Primary, Unique Key and Foreign Key Constraints (insert in child table, delete from parent table) in a relational table using the implemented index files, see Key-valueSystemsIndex.pdf.

Lab4. **14p** Implement a Visual Query Designer for SELECT statement (possible join of *n* tables) or execute it from command line.

Implement the selection and projection operator, use the existing index files.
Generate 100.000 rows for one of the tables, 10.0000 for another table to test the algorithms.

Lab5. **14p** Implement 2 join algorithms of

- indexed nested loop

- hash join (the partitions have to be written to disk)

- sort merge join (implement external sorting and write the runs to disk)

Use the existing index files where it is possible.

Extend the execution of the SELECT statement for the join of more than too tables.

Lab6. **14p**
Implement group by statement, min, max, avg, count, sum functions and having, use the existing index files.

For a complex SELECT statement execute first selections, then the joins.