Project for 4 students: Implementation of a complete single-user relational database management system. It involves a significant amount of coding. The project is highly structured, but there is enough slack in the specification so that creativity is both allowed and required.

It is recommended to implement a server component and a client one. The client can be implemented as Windows interface, Web client or a command line parser.

The basic project is divided into four parts:

1. *The Record Management* (*RM*) *Component:* implement a set of functions for managing unordered files of database records. (There is recommended to use binary files to implement unordered files). You can consider fix length records; the management of variable length records is optional. One idea to implement the delete operation of a record is the logical delete. It means to store for every record in one bit, which store: the record is deleted or not. In order to not read the whole file to find deleted records and overwrite them with new ones, you can link the deleted records in a stack or queue. The top of the deleted records stack can be stored in the first record of the file.

   You have to store the system catalog. It will contain table names, index file names. For every table the file name, where the table is stored, the structure of the table, the constraints, the associated index files. For every index file, the search key, the type of it. You can implement the catalog in XML file. In Catalog.xml you can find an example.

2. *The Indexing* (*IX*) *Component:* implement a facility for building indexes on records stored in unordered files. The indexing facility will be based on B+ trees or dynamic hashing.
3. *The System Management* (*SM*) *Component:* This part will implement various database and system utilities, including data definition commands (at least integer and character data type), including primary key and foreign key constraint (primary key have to be implemented for one or more columns, but foreign key is optional to implement for more than one column), index definition commands and catalog management. For primary key you will create index file automatic. The System Management component will rely on the Record Management and Indexing components from Parts 1 and 2. It also will use a command-line parser or a graphical user interface.
4. *The Query Language* (*QL*) *Component:* In this part students will implement a query language, which consists of user-level data manipulation commands, both queries and updates (SQL Select, Insert, Update, Delete can be used). The Query Language component will rely on the three components from Parts 1-3, and it will use the command-line parser or a graphical user interface. The queries have to be processed, using algorithms presented at the course. Features you have to implement in Select statement: selection, projection, join of tables, aggregation, cumulative functions, (subquery, order by is optional for extra points).

It is recommended to begin with the part 1, than 2 and every module to be developed by 4 students.

Examples of such minibase you can find at:

www.cse.unsw.edu.au/~cs9315
inst.eecs.berkeley.edu/~cs186/sp07
www.cs.wisc.edu/coral/mini_doc/minibase.html

In students.sql you can find an example script, your mini DBMS will have to process.

The implementation of the buffer management is optional, but you can accumulate extra points with it.

Bonus points:

- Buffer management: 20 points
- Variable length records: 5 points
- Datetime datatype and DateDiff, DateAdd functions. 10 points
- Subqueries: 10 points
- Order by (external sort implementation): 10 points
- One more index technique: 10 points