Design of a Multi-user DBMS supporting dynamic SQL queries

The system architecture is depicted in the diagram below.

The description of the system architecture of the multi-user DBMS is as follows. There are three components, namely, the client, the server, and the communication module. Two levels of abstraction are used in this design (sub-system and modules of a sub-system).

The client contains User Interface and Report Engine; the server contains Authentication Module, SQL Parser, Query Processing Unit, Schema manager and Data Store.

Client and server interact with the help of the communication system. The communication system consists of a communication channel and two communication modules, one at each of the server and client side. Communication system is a volatile one that is likely to change, e.g. the data format of the communication packets. If the communication modules are resided in both server and client, a change in the communication protocol results in changes in both of them. These reasons lead to
this design.

User Interface consists of two sub-systems, namely, Input Module and Output Module. Input Module takes the user's SQL query and the user's login information for authentication. Output Module display the result (for examples, status denoting whether a table can be created, the result tuples of a SQL query, or any error message like fail to login) of the user's action (for examples, login or issuing of a SQL query) to the user, the result is obtained from the Report Engine. Any user's input will be feed into the client side's Communication Module, and the result from the server side will be feed into the Report Engine.

Authentication Module takes the user's login information from the server side's Communication Module, and returns the result (either successful or unsuccessful) of the login to the same Communication Module.

After successful login, the authentication module will pass the SQL query issued by the user to the SQL parser, which in turns parses in a format suitable to be processed by the Query Processing Unit. In case the SQL is not in a correct format, an error message will be feed to the Query Processing Unit as well. The Query Processing Unit, after processing a valid query or after obtained the error message from the SQL Parser, passes the result of the query or the error message to Communication Module.

Query Processing Unit consists of Data Retrieval Module and Data Definition Module. The former is for processing data retrieval language of SQL and interacts with the Data Store to compute the result of the SQL. The later is for processing data definition language of SQL and interacts with the Schema Manager to create/modify/delete the schema and table defined by the SQL. Schema Manager interacts with the Data Store to perform the action specified by the data definition language of SQL.

In case the SQL query is in a correct format but semantically incorrect with respect to the schema of the database (for examples, retrieving tuples from a non-existing table, creating a table with the same name as an existing table, etc.) or the data store cannot function properly (for example, fail to allocate spaces since the underlying storage is running out), the corresponding modules will pass error messages to the communication module.

Finally, Data Store consists of Storage Interface and Index. The former acts as an interface to the underlying storage system, in order to perform the storage job in a way making the best use of the underlying storage system. The later indexes the data stored in the data store so as to support efficient lookup.