DATABASE SYSTEM IMPLEMENTATION PROJECT

Priyam Chandra Amruta Andurkar Shravya Narayan Sarma

ABSTRACT

In this project we have tried to implement newer features over the existing ones in the h2 database.As flexible and convenient as the h2 database is there are few features that could be implemented in order to make h2 a better environment for database operations.It is an open source relational database management system implemented in Java.It has an embedded server with a browser based console application and a command line shell tool as its utilities.Since it is an open source software it gave us the option of modifying and enhancing the database by implementing newer features which would make it an even more powerful tool for performing database operations.It also has the ability to act as PostgreSQL server.It can be operated in clientserver mode and has the option of supporting clustering.It is a relatively faster database engine,has stronger security features and supports mutiversion concurrency.It supports multiple connections and 2 phase commit feature.It supports a wide range of data types such as CLOB and BLOB.Even though H2 supports a wide range of aggregate functions there are a few which could be added to the current set of functions in order to make it a more complete database management system.The features that we have implemented are a few aggregate functions such as Median,Mode the ones returning the last and first elements or records in the database like First and Last.In addition to this we plan to implement the unique shortest substring queries which is what we had stated in the project proposal.The median function would be the numerical value separating the lower half from the higher half of a data sample.It is defined only on ordered onedimensional data.Mode function is based on the fact that it returns a value that appears most often in a set of values.The shortest unique substring queries is based on a research paper implementation which involves having to build a suffix tree algorithm.In the shortest substring queries it provides a wide variety of applications such as information retrieval,bioinformatics and event context analysis.Firstly a suffix tree is built to answer a shortest unique substring query and using that as the basis and having O(nh) time and O(n) space shortest unique substring is computed for every position in the string.For instance let us consider having to use a keyword for searching through a search engine database the first method would to find the occurrence and evaluate the number of occurrences of that keyword.A more sensible approach would be to display the snippet of each occurrence of the keyword and the length of each snippet is predefined globally.Through this feature implementation we address the point of answering shortest substring queries from the algorithmic point of view.Firstly we implement the shortest unique substring queries and explore their properties thoroughly and later implement an algorithm to answer a shortest substring query in O(n) time using suffix tree algorithm.We try to make it as realistic as possible through the usage of real data sets.The reason suffix tree implementation was chosen was because of the drawback of brute force implementation which is not acceptable in practice since the length would vary and for larger lengths it would certainly result in a not so efficient implementation of the algorithm.In order to have all this handled efficiently suffix tree seems to be a better approach and hence a more realistic approach for bringing shortest unique substring queries into picture from a broader sense and more relatable to real world scenarios.Our approach is to have the usage of minimalistic space and having to execute in the shortest period of time and yet having to come up with the one that does the job and provides the required output.It can be considered similar to keyword searching mechanism or fuzzy search algorithm.

1. DESIGN CONSIDERATIONS

Embedded mode

1. This opens the database from within the same JVM using JDBC.
2. Fastest and easiest connection mode.
3. The only disadvantage is it may be open in only one virtual machine.
4. Persistent and inmemory databases are supported.
5. No limit on the number of open connections.

.

Server Mode

1. An application opens a database remotely using JDBC or ODBC API.

2. A server needs to be started either within the same virtual machine or a different virtual machine
3. Many applications can connect to the database through the server which opens the database in the embedded mode.
4. It is slower than embedded mode since it requires TCP/IP connection.

.

Mixed Mode

1. ÂĄIt is a combination of embedded and server mode.
2. ÂĄThe first application is connected to the database in the embedded mode and also starts server so that other applications can concurrently access the database.
3. Local connections are as fast as the ones in embedded mode and the remote ones consume a lot more time.
4. ÂĄThe server can be started and stopped from within the application.

.

2. IMPLEMENTATION

The source code implementation has been included in another file we have specific algorithms for each of the implementations such as the calculation of median,mode and the first and last records specified in the database along with the algorithms for shortest unique substring queries.

3. ARCHITECTURE

The system architecture of the software is the conceptual model that defines the structure behavior and views of the system.Be it the mixed mode embedded mode or the server mode it involves an application be it on the same system or the remote access a jvm which is a must given that it is implemented in java.In the standalone mode or the embedded it is within the same java virtual machine and would not need any tcp/ip connection or access to another application or server in the server mode it would need access to the server and in the mixed mode it is a combination of the above specified modes. It consists of the following features database file locking,database file encryption,custom file access mode,logging and recovery,multidimensional indexing,Triggers,Pluggable and userdefined tables. Since the database files could be encrypted and in order to do so we have two encryption algorithms that are supported AES and XTEA and in order to utilize file encryption algorithm we would need a cipher and the password for connecting to the database. The lock file is created only when the database is opened in order to signal other processes that the database is in use and it is deleted when the database is closed.It utilizes three methods File,Socket and FS.File is the default method and uses a watchdog thread to protect the database file it reads the lock file every second.The second one is socket which opens a server socket and does not read the lock file every second and these should be utilized only if they are accessed by a single computer.The third one is FS which uses native file locking. The custom file access mode involves having to access the database in customized mode while the database usually operates in read write mode as the default access mode we can later reset it as per our convenience by changing the ACCESSMODEDATA variable value.

4. REFERENCES

1. On Shortest Unique Substring Queries Jian Pei Wush Chi-Hsuan Wu Mi-Yen Yeh
2. Minimum Unique Substrings and Maximum Repeats Lucian Ilie and W. F. Smyth