

CS 4420 DATABASE PROJECT (Option 2)

Spring Semester 2004

The overall objective of this project is to try out some advanced database implementation techniques on top of existing database engines. You will be given some open source software packages (in Java). The project requires students to fully understand the open source database engine implementation. You are expected to enhance the existing package by adding more advanced database features (e.g., add aggregate functions, query optimization methods, support larger data files, use additional access methods).

The project will be implemented in two phases. The first phase will focus on two components: (1) the study and understanding of the existing software and report its implementation details (including limitations); and (2) the design decision on which three advanced database features that you intend to add on top of this open source software package and why they are important features. You may use applications you know to justify your decision.

The second phase will extend the open source database engine to provide richer database features. You are required to provide the conceptual and implementation design for each of the three advanced features identified in the Phase I report. Then you are expected to implement these features on top of the given open source database software. Please bear in mind that your implementation should have a user-friendly GUI to demonstrate the new database functionalities.

The project has two milestones:

Milestone 1: A report on Phase I is due on Thursday of the eighth week of the course, which is **February 26, 2004**. You may hand in your report in class on that day or email it to instructor and TA.

Milestone 2: The software package of the entire product of your project and a final report will be produced. Each project team will schedule a demo date and time to demonstrate their product and walk through the code. The **due date** of your final project is the midnight of the Thursday of the last week of the semester, which is **April 22, 2004**. You should send in a compressed tar file including the source code with good documentation, the printout of your data used in the project, some sample screenshots, and the final report.

A detailed description of the deliverables for each of the two milestones will be made available online at the course web site.

Now we describe the detailed requirements for each of the two phases below.

PHASE I:

For the project, you can choose one of the two suggested open source database implementations (both in Java) below:

1. **tinySQL** [1]

tinySQL is a lightweight, 100% Java SQL engine that also includes a JDBC driver. It supports the following capabilities (from the README file):

```
SELECT (with joins)
UPDATE
INSERT
DELETE
CREATE TABLE
DROP TABLE
ALTER TABLE xx ADD coldef           dBase only
ALTER TABLE xx DROP [COLUMN] col    dBase only
ALTER TABLE xx RENAME [COLUMN] foo TO bar dBase only
WHERE                                only AND is supported
```

tinySQL now supports all JDBC-Data types and the ODBC-Minimum grammar is implemented. Although tinySQL is not optimized for speed, its architecture makes it possible to build SQL interfaces to non-SQL data sources, such as .DBF (dBase database file format) files or text files.

2. **hsqldb** [2]

hsqldb is a relational database engine written in Java, with a JDBC driver, supporting a rich subset of ANSI-92 SQL (BNF tree format). It offers a small (less than 160k), fast database engine which offers both in memory and disk based tables. Embedded and server modes are available. Additionally, it includes tools such as a minimal web server, in-memory query and management tools (can be run as applets) and a number of demonstration examples.

hsqldb is more complex than tinySQL database engine. It has many features, including view support, aggregate functions, triggers, multi-dimensional indexes, and most JDBC interface support. For more details, please visit the software's home page [2].

You are required to study the existing database engine and give a thorough report on how the engine is implemented. For example, you can report details on how the engine handle the following:

- Creation of relations and indexes if any
- Inserting records into a file
- Inserting key values and addresses into the index

- Fetching a page
- Basic query processing steps
- Query optimization techniques used if any

You will also need to identify the limitations of the existing software so that you can improve some of them in Phase II of the project.

In addition to the study and understanding of the existing software and report its implementation details (including limitations), you are required to describe your design decision on which three advanced database features that you intend to add on top of this open source software package and why they are important features. You may use applications you know to justify your decision. For example, if you intend to add query optimization component and index to tinySQL, you will need to introduce commands such as CREATE INDEX, LOAD INDEX LOAD RSTATES, LOAD ISTATS in order to collect and maintain statistics about all relations created and populated in the systems catalog. You may refer to the Project Option 1 description to learn more on how to design and implement such commands. At Phase I of this project, you are expected to accomplish the following:

1. identify which open source product to use
2. identify which basic database functions are supported by the chosen open source package
3. identify what additional database features you would like to add to the top of chosen open source package
4. for each feature, identify the list of basic components (e.g. commands or modules) that you need to write and incorporate into the chosen open source package
5. identify potential benefits and difficulties for adding the new features to the chosen open source package.

IMPORTANT: As a milestone of the project, we require that each project team produce a Phase I report documenting all five requirements listed above. You are not required to turn in any code for Phase I.

Due date: Thursday of the 8th Week, which is February 26 2004.

PHASE II:

A main objective of the second phase is to extend the open source database engine to provide richer database features. You are required to improve the existing package in at least 3 aspects. Some suggested aspects include:

- Additional query processing capabilities such as join, aggregate, nested queries, or keyword based search, summarization, etc.

- Support larger data files, such as still images, video clips, voice clips, etc.
- Provide query optimization support or add additional query optimization methods, including creating and using indexing techniques, additional data access methods such as sort merge, hashing, and so forth.
- Support Web query and result display interface with richer capabilities (sorting, summarization, etc).
- Support potential optimization techniques to reduce the latency of query answers. Example techniques include returning only the top matching tuples [3, 4, 5], online aggregation [6], approximate queries [7]. These advanced features allow users to see some portion of the results of their queries earlier and observe the continued update of the query answers continuously, instead of waiting for a long time. These advanced features are particularly useful for queries that return large amount of data and thus take relatively longer time to process.

You are required to provide the conceptual and implementation design for each of the three advanced features identified in the Phase I report. You are expected to provide the best possible implementation of the three features on top of the chosen open source database software. At the end of Phase II, you will need to submit a written report on the design and implementation considerations, the system architecture, and the design choices for the improved database engine. In addition, you are expected to submit the source code of your system and the README instruction file. The source code should be well documented and it should describe which team member made what changes to the original package. A demo will be scheduled for your final product in the last week of the semester. Each team member should hand in a page describing the tasks you have done and your contributions to the team project and your evaluation on the rest of the team members. You may send this page by email to the instructor and the TA separately. Your project grade will be based on the reports from the two phases, the demo quality, and the quality of source code.

References

- [1] tinySQL <http://www.jepstone.net/tinySQL>
- [2] hsqldb <http://hsqldb.sourceforge.net>
- [3] N. Bruno, L. Gravano, and A. Marian. Evaluating Top-K Queries over Web-Accessible Databases. In *Proceedings of the 18th IEEE International Conference on Data Engineering (ICDE 2002)*, 2002. <http://www.cs.columbia.edu/%7Egravano/Papers/2002/icde02.pdf>
- [4] S. Chaudhuri and L. Gravano. Evaluating top-K selection queries. In *Proceedings of the 25th International Conference on Very Large Data Bases (VLDB'99)*, 1999. <http://www.cs.columbia.edu/%7Egravano/Papers/1999/vldb.pdf>

- [5] K. C. Chang and S.W. Hwang. Minimal Probing: Supporting Expensive Predicates for Top-k Queries. In *Proceedings of the 2002 ACM SIGMOD Conference*, Madison, Wisconsin, June 2002. <http://www-faculty.cs.uiuc.edu/%7Ekcchang/Papers/mpro.ps>
- [6] J. M. Hellerstein, P. J. Haas, and H. J. Wang. Online aggregation. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, 1997. <http://control.cs.berkeley.edu/online/online.pdf>
- [7] Kaushik Chakrabarti, Minos Garofalakis, Rajeev Rastogi, and Kyuseok Shim. Approximate query processing using wavelets In *Proceedings of VLDB'2000*, Cairo, Egypt, September 2000. <http://citeseer.nj.nec.com/chakrabarti00approximate.html>