System Development: A Project Based Approach

Hossein Hakimzadeh, Robert Batzinger, Susan Gordon

Department of Computer and Information Sciences, Indiana University, South Bend, IN 46615

Introduction

This poster profiles a project-based approach employed in “Advanced Database Systems” at Indiana University South Bend[3]. The course reintroduces system development topics in today’s contemporary computer science curriculum.

The objective of this course is to design and implement a mini database engine (MiniDB) as the vehicle for fostering system development skills. The engine in turn serves as a platform for research and study of contemporary topics such as security, concurrency control, performance tuning, and data mining.

The Course Structure

Conceptually, the Advanced Database course and its project (MiniDB) are divided into five phases: **preparation**; design and implementation of **core algorithms**; **research** in advanced algorithms; **implementation of advanced algorithms**; and **presentation** of the final project.

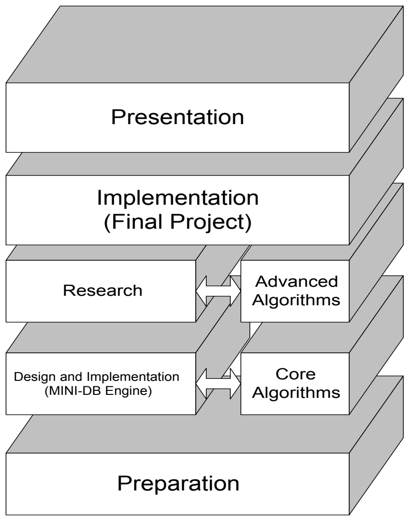
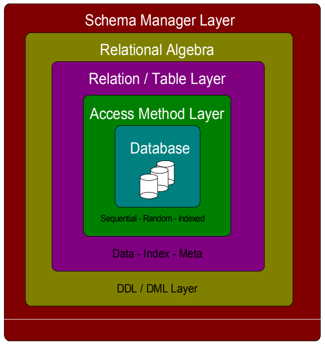


Figure 1. The Five Phases of Constructing MiniDB

Conceptual Model

Conceptually, MiniDB can be viewed as a series of concentric layers. Each layer provides a new level of abstraction and moves the project closer to a fully functioning database management system.

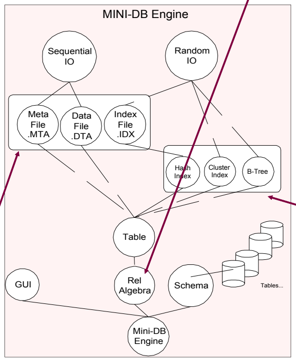
Figure 2. MiniDB Conceptual Model



Implementation Model

The design and implementation are broken down into a series of deliverables. Each deliverable serves two purposes. First, it seeks to incrementally construct new building blocks that move the project toward the goal of a working database engine. Second, it allows the students to systematically refine the previously constructed components.

Figure 3. The MiniDB is implemented as a series of classes which provide the basic access mechanism, data definition (DDL), and data manipulation language (DML).



Access Mechanism

The goal of the first deliverable is to construct a series of classes for creating and manipulating simple **data**, **index** and **meta-data** files. These three classes provide the basis for creating a database table.

Data Definition and Manipulation

The goal of the second and third deliverables is to develop a data definition and data manipulation component for the MiniDB system. Relational algebra is chosen for this purpose.

```
Class Mini_Rel_Algebra {
  bool create(relation, schema);
  bool drop(relation);

  bool insert(relation, attribute_list, value_list);
  bool delete(relation, attribute_name, condition, attribute_value);
  bool modify(relation, search_attribute_name,
           condition, search_attribute_value,
           modify_attribute_list, modify_value_list);
  result_rel select(relation, attribute_name, condition, attribute value);
  result_rel project(relation, attribute_list);
  result_rel cartesian_product(relation_1, relation_2);
  result_rel join(relation_1, relation_2, condition_list);
  result_rel union(relation_1, relation_2);
  result_rel intersect(relation_1, relation_2);
  result_rel difference(relation_1, relation_2);
}
```

Figure 4. Relational Algebra Operations

Refining the Indexing Methods

In order to develop some tools for later optimization of the database, two new index classes are developed. The first class optimizes the existing index class using hashing techniques. The second class develops a cluster index class to handle indexing based on non-key attributes.

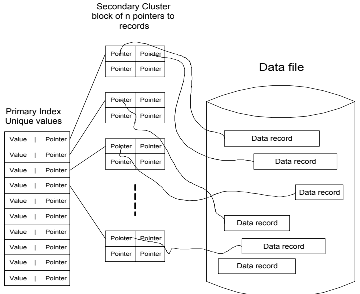

Secondary Cluster block of n pointers to records
Primary Index Unique values
Value | Pointer
Pointer
Data file
Data record

Figure 5. Cluster Index

Advanced Algorithms

The final phase of the course involves the creation and integration of advanced components on top of the basic MiniDB engine. During past offerings of this course, students have been able to develop algorithms for concurrency control, database security, access control, database integrity [4], external sorting, data mining [5], join optimization [6], distributed databases [7], and deductive databases.

Conclusions

This paper profiles the implementation of a course in “Advanced Database Systems”. The primary focus of this course is to study the inner working of system software, specifically how database management systems work. The course systematically leads the students through the design and implementation of a database engine called MiniDB[3,8].

Similar project-based courses in advanced database design were introduced by David DeWitt [9] and later extended by Mike Carey and Raghu Ramakrishnan [10] at the University of Wisconsin. Another system inspired by the DeWitt’s model has been implemented by Albano *et al* [11]. From our perspective, the advantages of MiniDB compared to the above systems are two fold. First, MiniDB has a smaller design footprint which allows the students to more easily understand the concepts. Second, MiniDB uses logical records storage and retrieval vs. physical or page level I/O used by the above systems.

The above project-based, system development approach can be applied to other courses such as computer networks, computer graphics, and computer security. For example, a computer networking course can be augmented such that, through a series of assignments, students finish the course with their own Mini NetworkAPI based on the OSI model. Students in computer graphics could build their own Mini GameEngine. As the computer science curriculum continues to refine and redefine itself, we hope that system development will regain more prominence in the curriculum.

References

[1] IEEE/ACM-CS Computing Curricula 2001,Computer Science, Final Report, The Joint Task Force on Computing Curricula, IEEE Computer Society, Association for Computing Machinery, December 15, 2001.

[2] ACM/AIS/IEEE-CS, Computing Curricula 2005, by The Joint Task Force for Computing Curricula 2005, 30 September 2005.

[3] Hakimzadeh, H., “MINI-DB: A Pedagogical tool for Teaching Advanced Database Systems”, accessed on web on March 2008. www.cs.iusb.edu/technical_reports/TR-20071222-2.pdf

[4] Gordon, S., “Database Integrity: Security, Reliability, and Performance Considerations”, Technical Report: TR-20071226-1, accessed on web on Dec. 2007. www.cs.iusb.edu/technical_reports/TR-20071226-1.pdf

[5] Batzinger, R., “Calling R from Ruby”, Technical Report: TR-20080109-1, accessed on web on Jan. 2008. www.cs.iusb.edu/technical_reports/TR-20080109-1.pdf

[6] Rupley, Michael, Jr., “Introduction to Query Processing and Optimization”, Technical Report: TR-20080105-1, accessed on web on Jan. 2008. www.cs.iusb.edu/technical_reports/TR-20080105-1.pdf

[7] Rababaah, H., “Distributed Databases Fundamentals and Research”, Technical Report: TR-20050525-1, accessed on web on Dec. 2007. www.cs.iusb.edu/technical_reports/TR-20050525-1.pdf

[8] Hakimzadeh, H., “MINI-DB: project website” www.cs.iusb.edu/minidb

[9] DeWitt, D., the Minirel project. A database course project that involved building a small relational DBMS. *www.cs.wisc.edu*

[10] Carey, M., Ramakrishnan, R., the Minibase project. Extension and redesign of Minirel project. www.cs.wisc.edu/coral/mini_doc/project.html

[11] Albano, A., JRS (Java Relational System) is a relational DBMS implemented in Java and designed for educational use. Accessed on the web on March 2008, www.di.unipi.it/~albano/JRS/project.html

For further information

Please contact the authors at *hhakimza@iusb.edu*

Additional information such as assignments, publications, and code libraries for this and related projects can be obtained at *www.cs.iusb.edu/minidb*

A PDF-version of the poster is also provided.

SIGCSE 2009: The 40th ACM Technical Symposium on Computer Science Education, March 4-7, 2009, Chattanooga, TN USA