MINI-DB
Demystifying the Inner Workings of Database Systems

Hossein Hakimzadeh, Robert Batzinger, Susan Gordon

Department of Computer and Information Sciences
Indiana University – South Bend, Indiana

Nov. 8-10, 2010
Las Vegas, NV
MINI-DB
Demystifying the Inner Workings of Database Systems

Hossein Hakimzadeh, Robert Batzinger, Susan Gordon

Department of Computer and Information Sciences
Indiana University – South Bend, Indiana

Nov. 8-10, 2010
Las Vegas, NV

23rd International Conference on
Computers and Their Applications
in Industry and Engineering
(CAINE-2010)
Outline

• The Challenge
• Our Solution!
• MINI-DB
• Lessons learned – Student Feedback
• Conclusions
The Challenge:

- Diversification of the CS Curriculum
  - Advantages
  - Disadvantages
Diversification of CS Curriculum:

• Advantages
  • Ability to expose students to contemporary topics such as cyber security, distributed computing, parallel computing, bioinformatics, and game programming, robotics, etc.
Diversification of CS Curriculum:

• Disadvantages
  • Courses that deal with the internal working of computers, or courses that require system design and system development are being systematically removed from the undergraduate curriculum.

• Merging of (OS and Networking), (Concepts of Programming Languages and Compilers), (File Organizations and Databases)

23rd International Conference on Computers and Their Applications in Industry and Engineering (CAINE-2010)
Our Solution:

- Deliberate review and redesign of elective and required courses to include system design and development.

- Development of more project based courses.

- Development of Open Source Courseware. (e.g. http://www.ocwconsortium.org/)
Case Study:

• Design and Development of Mini-DB
• http://www.cs.iusb.edu/minidb/

Objective:

• To Demystify the Inner Workings of Database Systems
MiniDB Conceptual Model:

Schema Manager Layer

Relational Algebra

Relation / Table Layer

Access Method Layer

Database

Sequential - Random - Indexed

Data - Index - Meta

DDL / DML Layer

23rd International Conference on Computers and Their Applications in Industry and Engineering (CAINE-2010)
Course Structure:

Phase V → Presentation

Phase IV → Implementation (Final Project)

Phase III → Research

Phase II → Design and Implementation (MINI-DB Engine)

Phase I → Preparation

23rd International Conference on Computers and Their Applications in Industry and Engineering (CAINE-2010)
Phase I Preparation

Depending on the focus of the course:

**Advanced Database Systems**
- Students review and examine the code base for Phase II. (next set of slides)

**Database Internals**
- Students survey the I/O facilities of the implementation language. (C++, C, C#, Java, Ruby, etc.)

23rd International Conference on Computers and Their Applications in Industry and Engineering (CAINE-2010)
Phase 2
MiniDB Design

MiniDB Foundation Classes

MINI-DB Engine

Sequential IO
Random IO

XML

Meta File .MTA
Data File .DTA
Index File .IDX

Hash Index
Cluster Index
B-Tree

Table

GUI

Rel Algebra

Schema

Mini-DB Engine

23rd International Conference on Computers and Their Applications in Industry and Engineering (CAINE-2010)
Phase 2
MiniDB Design

Table Class

MINI-DB Engine

Sequential IO
Random IO
XML
Meta File .MTA
Data File .DTA
Index File .IDX
Hash Index
Cluster Index
B-Tree

Table
GUI
Rel Algebra
Schema
Mini-DB Engine

23rd International Conference on Computers and Their Applications in Industry and Engineering (CAINE-2010)
class Table
{
    char TableName[256];
    Data_File *dta;
    Meta_File *mta;
    Index_File *idx;

    int TotalRecords;
    int DeletedRecords;

    public:
    Table(char *tablename);
    ~Table();

    void EraseTable(void);
    int CreateTable(char *schema);
    void OpenTable(void);
    void CloseTable(void);

    int Insert(char *a_record, unsigned long key);
    int Delete(unsigned long key);
    int Update(char *a_new_record, unsigned long key);

    int SearchByKey(unsigned long key);
    int SearchByField(char *field_name, char *value);

    void Print(unsigned long key);
    void PrintSchema(void);
    void Sort();
    void Reorganize();
    int GetTotalRecords(void);
    int GetDeletedRecords(void);
    double GarbageRatio(void);
    void CalculateTotalAndDeletedRecords(void);
};
Phase 2
MiniDB Design

Relational Algebra
Class

23rd International Conference on Computers and Their Applications in Industry and Engineering (CAINE-2010)
Class Mini_Rel_Algebra {
    bool create(relation_name); 
    bool insert(relation_name, attribute_1, value_1,.. attribute_n, value_n); 
    bool delete(relation_name, attribute_name, attribute_value); 
    bool modify(relation_name, attribute_name, attribute_value); 
    result_rel select(relation_name, attribute_name, condition, attribute_value); 
    result_rel project(relation_name, 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); 
}
Phase 2
MiniDB Design

Schema Class

MINI-DB Engine

Sequential IO

Random IO

XML

Meta File .MTA

Data File .DTA

Index File .IDX

Hash Index

Cluster Index

B-Tree

Table

Rel Algebra

Mini-DB Engine

GUI

23rd International Conference on Computers and Their Applications in Industry and Engineering (CAINE-2010)
Phase 3
Research

- Implementing Phase 1 and 2, may take 6 to 10 weeks, leaving approximately 5 to 9 weeks to work on Phase 3, 4 and 5.
Phase 3

- Phase 3, can be implemented in two ways:

1. A course in Database Internals.

Faculty teaching database internals can continue to build additional components to extend the MiniDB engine and incorporate features such as:

- **Indexing algorithms** *(Hash Index, Cluster Index, etc.)*
- **XML**
- **Paging and Buffer Management**
- **Parsing** *(Relational Algebra and/or SQL parser)*
- **Log files**
Phase 3

Database Internals:

**Supporting Classes**

- Hash index
- Cluster index
- XML
- B-tree
- Paging
- Caching
- SQL Parser
- Logging

23rd International Conference on Computers and Their Applications in Industry and Engineering (CAINE-2010)
Phase 3

Advanced Algorithms:

Faculty teaching advanced database concepts can start by quickly familiarizing their students with the MiniDB Foundation Classes by way of an assignment (that uses the MFC to build a simple database and then queries the database using the relational algebra API).

Future assignment can extend the MiniDB engine to incorporate features such as:

- **Transactions** (Start, Commit, Abort, Undo, Redo, checkpoint, write, read)
- **Concurrency Control** (2PL, Optimistic)
- **Distributed Transaction Processing** (Implement a new networking class, and extend the MiniDB engine to accommodate distributed query processing)
- **Query optimization** (Extend the MiniDB engine to include more meta-data as well as runtime information and optimizes the query tree.)
- **New and Novel Algorithm** (Use the MiniDB platform to implement and compare new algorithms vs. traditional/existing algorithms.)
Lessons Learned:

- During the past 3 offering of this class, student feedback indicate that after completing this class, they had found a great appreciation for project based classes.

- The ability to construct a database engine from scratch was specially appealing. Although, among the students who dropped the course, this aspect of the course was sited as the primary reason.

- Students use the code base (MiniDB Foundation Classes) developed in this course in other courses (e.g. Information Organization, and Operating Systems.) as well as after graduation.
Lessons Learned:

**Advanced Database Systems (MiniDB)**

**Operating Systems (ULTIMA)**

**Computer networks (NetApp - Mini Network API)**
Conclusion:

• *We profiled the implementation of a course in “Advanced Database Systems”. The primary focus of this course was to study the inner workings of database management systems and to research advanced database concepts.*

• *The course systematically lead the students through the design and implementation of a database engine called MiniDB, then it allowed them to research advanced DB concepts and implement these concepts as part of the MiniDB system.*

• *This approach has allowed our students to use the MiniDB engine as the starting point for further research.*

• *The course material and the MiniDB project is available as an open courseware.*
Interested?

The MiniDB is available as an open source courseware:

- [www.cs.iusb.edu/minidb](http://www.cs.iusb.edu/minidb)

The site includes:

- Assignments
- Design Documentation
- C++ API
- Source Code (Restricted Distribution to Faculty only)

23rd International Conference on Computers and Their Applications in Industry and Engineering (CAINE-2010)
## Other MiniDB Projects:

<table>
<thead>
<tr>
<th>Project</th>
<th>Minibase (Inspired by Minirel)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Author</td>
<td>Mike Carey and Raghu Ramakrishnan (Univ. of Wisconsin)</td>
</tr>
<tr>
<td>Language</td>
<td>C++</td>
</tr>
<tr>
<td>Status</td>
<td>Active</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Project</th>
<th>Minirel</th>
</tr>
</thead>
<tbody>
<tr>
<td>Author</td>
<td>David DeWitt</td>
</tr>
<tr>
<td>Language</td>
<td>C</td>
</tr>
<tr>
<td>URL</td>
<td>Not available</td>
</tr>
<tr>
<td>Status</td>
<td>May be inactive</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Project</th>
<th>SimpleDB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Author</td>
<td>Edward Sciore (Boston College)</td>
</tr>
<tr>
<td>Language</td>
<td>Java</td>
</tr>
<tr>
<td>URL</td>
<td><a href="http://cs.bc.edu/~sciore/simplesdb/intro.html">http://cs.bc.edu/~sciore/simplesdb/intro.html</a></td>
</tr>
<tr>
<td>Status</td>
<td>Active</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Project</th>
<th>MinSQL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Author</td>
<td></td>
</tr>
<tr>
<td>Language</td>
<td>Java</td>
</tr>
<tr>
<td>URL</td>
<td>Not open source</td>
</tr>
<tr>
<td>Status</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Project</th>
<th>miniDB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Author</td>
<td>Hans Harder</td>
</tr>
<tr>
<td>Language</td>
<td>C</td>
</tr>
</tbody>
</table>
| URL       | [http://freshmeat.net/projects/minidb/](http://freshmeat.net/projects/minidb/)  
| Status    | Active                                    |

<table>
<thead>
<tr>
<th>Project</th>
<th>minidb</th>
</tr>
</thead>
<tbody>
<tr>
<td>Author</td>
<td>jpwarren00</td>
</tr>
<tr>
<td>Language</td>
<td>Java</td>
</tr>
<tr>
<td>Status</td>
<td>May be inactive</td>
</tr>
</tbody>
</table>