Fall 2004

CS 405/605: Introduction to Database Management Systems

Soon M. Chung
Wright State University - Main Campus, soon.chung@wright.edu

F04

CS 405/605 Introduction To Database Management Systems, Fall 2004

Description : Survey of logical and physical aspects of database management systems. Entity Relationship, relational, object-oriented models for databases are presented. Physical implementation methods are discussed.

Prerequisite: CS 400 Data Structures and Software Design.

Instructor : Dr. Soon M. Chung, 403 Russ Center
(937)775-5119, schung@cs.wright.edu, http://www.cs.wright.edu/~schung,

Class : Tu. Th. 4:10-5:25 at 140 HS (Health Science)

Office hour : Tu. Th. 2:30-3:30 p.m. at 403 Russ, or by appointment.
*use e-mail for short questions.

Text Book : R. Elmasri and S. B. Navathe, Fundamentals of Database Systems, 4th edition, Addison Wesley, 2003.

Topics : DBMS concepts and architecture (Chap 1, 2)
Entity-Relationship model (Chap 3)
Relational data model and relational algebra (Chap 5, 6) (= Chap 7 in 3rd Ed.)
ER to Relational mapping (Sec. 9.1) (= Sec. 9.1 in 3rd Ed.)
SQL - a relational database language (Chap 8, Sec. 9.1, 9.2) (= Chap 8 in 3rd Ed.)
Record storage and file organizations (Chap 13) (= Chap 5 in 3rd Ed.)
Index structures (Chap 14) (= Chap 6 in 3rd Ed.)
Functional dependencies and normalization (Chap 10) (= Chap 14 in 3rd Ed.)

Grading : A:[85,100], B:[75,85), C:[65,75), D:[55,65), F:[0,55)
- There is no homework, but solutions of some exercise questions will be given.
- Midterm 30% (10/12, Tu), Project 30%, Final 40% (11/16, Tu, 5:45-7:45 pm)
- Project is paper-review, programming, or DB design. Select one by 10/14.
- The final report is due on 11/16.

(1) paper-review project
{ papers reviewed 6%, technical quality 8%,
written presentation 7%, discussion 9% }
- submit the topic and a list of 4 (or 5) selected papers by 10/14.
- submit the final report (around 25 pages in double space)

(2) programming
Extendible hashing (Ref. Sec. 13.8.3) simulation using a high-level programming language (C, C++, Java, etc.),
{ design 5%, documentation 5%, correctness 15%, discussion 5% },

(3) small database design and SQL programming using MS-Access or some other DBMS
{ description of problem 4%, ER design 6%, Relational Schema 5%, SQL queries and results 8%, discussion 7% }

CS 405/605 DB Design Project

The DB design and implementation project can be performed by the following steps:

1. pickup an application problem.
2. Entity-Relationship model design.
3. Convert the ER design to Relational DB schema.
4. create the relations using MS Access, Oracle, or some other DBMS.
5. Insert sample tuples into the relations
6. Write some meaningful SQL queries and run against the sample relations
7. Print out the results.
8. Discussion.

- Your ER design should include some relationships, so that some of your SQL queries can involve more than one relations. In general, a relational DB design including more than 4 relations is okay.

- Try most of the typical SQL commands to see how they work.

- Your final report should include the description of the problem, data requirements, ER design, relation schemas and instances, query descriptions and SQL queries, query outputs, and discussion.

- The discussion section can include your comments on the logical DB design, implementation, DBMS functionality and performance, future work, etc.

- A guideline for using MS Access is given at http://www.cs.wright.edu/~schung/cs605.html

CS 405/605 Programming Project

Simulation of Extendible Hashing.

1. The hash value of the hash-key attribute of each record is given as a randomly generated integer value.
2. Each data bucket (block) can store 40 records and is dynamically allocated. Each directory entry stores a pointer to a data bucket.
3. Generate up to 10^4 hash values (to simulate up to 10^4 records), and for every 10^2 hash values, display followings:

$$\text{utilization of the directory} = \frac{\text{number of data buckets}}{\text{number of entries in the directory}}$$

$$\text{utilization of the data buckets} = \frac{\text{number of total records}}{\text{number of records that can be stored in all the data buckets}}$$

Note: You can use any programming language, and you can plot the outputs if you want.

Reference:

1. Section 13.8.3 (= section 5.9.3 in 3rd Ed.) of the text book.

2. R. Fagin, J. Nievergelt, N. Pippenger, and H. R. Strong, Extendible Hashing: A fast access method for dynamic files, *ACM Trans. on Database Systems*, **4** (3) (1979) 315-344.

CS 405/605 Paper Review Project

Possible Topics

- Database models
- Database access mechanism (such as indexing, hashing, etc)
- Query optimization
- Concurrency control and recovery
- Parallel algorithms for query processing
- Performance evaluation of DBMS
- Database machines
- Distributed database
- Multidatabases (Heterogeneous databases)
- Expert database
- Logic and database
- Multimedia database
- Object-Oriented database
- Image database
- Engineering database
- CAD/CAM database
- Text retrieval system
- Data mining
- Data warehousing
- Other relevant topics

Reference Sources

- IEEE Trans. on Software Engineering
- IEEE Trans. on Knowledge and Data Engineering
- Computer (IEEE Computer Magazine)
- Communications of ACM
- ACM Trans. on Database Systems
- ACM Trans. on (Office) Information Systems
- Information Systems
- Multimedia Systems (Journal by ACM and Springer International)
- IEEE Multimedia (Magazine by IEEE)
- Data and Knowledge Engineering (Journal)
- Data Mining and Knowledge Discovery (Journal)
- Proc. of IEEE Int'l Conf. on Data Engineering
- Proc. of ACM Conf. on Management of Data (SIGMOD Conference)
  refer to the volumes of SIGMOD RECORD
- Proc. of ACM Symp. on Principles of Database Systems (PODS)
- Proc. of Very Large Data Bases (VLDB) Conference
- IEEE Trans. on Parallel and Distributed Systems
- ACM Computing Surveys
- Proc. of Int'l Conf. on Knowledge Discovery and Data Mining
- Proc. of IEEE Int'l Conf. on Data Mining

and Others