

University of Arizona
CSc 560: Database Systems Implementation
Spring 2004

Instructor: Bongki Moon (bkmoon@cs.arizona.edu)

Office: Gould-Simpson 715, 621-4326 (voice), 621-4246 (fax)

Office hours: Mon 2:45-3:45 and Wed 12:30-1:30, or by appointment.

Grader: Praveen Rao (rpraveen@cs.arizona.edu)

Class schedule: Mon/Wed 1:30-2:45, Gould-Simpson 701

Final exam: May 12, 2004 (Wed), 11:00-13:00, Gould-Simpson 701

Overview and Topics: CSc560 is a graduate-level course in database systems that will emphasize the DBMS architecture and implementation issues. The topics to be covered include

- Storage structures: disk, records, pages and files,
- Operating systems issues and buffer management,
- Access methods: B⁺-tree, multidimensional indexes (*e.g.*, R-trees),
- Query evaluation and optimization,
- Transaction management: concurrency control and recovery,
- Parallel database systems,
- Decision support and OLAP,
- Special topics: *k*-NN similarity search, time-series data, string databases, XML and semi-structured data, broadcasting/streaming data, etc.

Prerequisites: CSc460 (Database Design), or the instructor's permission.

Required texts:

- Raghu Ramakrishnan and Johannes Gehrke, *Database Management Systems*, third edition, McGraw-Hill, ISBN 0-07-246563-8, 2003.
- Bongki Moon, *Lecture Notes on Database Systems Implementation*.
- Most of the research articles in the reading list will be available from the ACM/IEEE Digital Libraries, which are accessible from all the computing labs on campus.

Term project: Students are required to complete a term project in small groups throughout the semester. The goal of the project is to build a working but simplified relational database system called MINIREL. The implementation of the MINIREL will be composed of four layers and a transaction manager. Among the four layers, the *paged file (PF)* layer will be provided; the other layers (*heap file (HF)*, *access method (AM)*, and *front-end (FE)*) and the *transaction manager (TM)* will be implemented by each project team. For each part of the MINIREL project, students are required to submit an archive of source files that will produce a function library and/or an executable program, and a short report that will explain the implementation approach such as overall techniques, important data structures, tricky design decisions, and so on.

Reading assignments and Class participation: You are expected to read every reading assignment *before class*. The reading assignments are mostly research articles and book chapters listed in the *Topics and Tentative Schedule* section of this syllabus. Background materials will be provided by the instructor and then the major points of the day's reading assignment will be discussed in class.

For each reading assignment, you will be required to do something to indicate that you have done the assigned reading and that you understand most of it. You will be often asked to turn in a summary of key points before class; sometimes you will be called on to answer questions; sometimes you will do group activities during class; sometimes you will be given a short quiz; sometimes you will not be given advance warning. However, if you have carefully read each assignment, you will be in good shape.

Grading: This course will be offered for three credits. The grading policy will be class participation (10%), midterm exam (15%), final exam (25%), and project (50%). Except for special circumstances, all members in a group will receive an identical grade for each part of the project. You are required to attend each and every class. The instructor will reserve the right to fail for the course any student who fails to attend more than twice without a valid excuse.

Posting grades: In an effort to comply with the new laws prohibiting the use of any part of student's SSN to post grades, the Department has devised a scheme to generate a unique numeric identifier for each student. The six digit numbers (called CSID) will be used to post the grades in the class web page, the class listserv and my office door.

Class URL and E-mail listserv: <http://www.cs.arizona.edu/classes/cs560/spring04> and cs560@listserv.arizona.edu

You are required to subscribe and read the messages posted to the class listserv on a regular basis (at least twice a week). You can also send messages to the listserv; subscribed class members only can read your messages. For subscription and information, see <http://listserv.arizona.edu>

Class accounts: You will be given accounts on `lectura` (`lectura.cs.arizona.edu`), the Debian Linux systems and Windows XP, plus an access card to Gould-Simpson 228 Lab. The `lectura`, Linux and Windows XP accounts are for instructional purposes only. To obtain your accounts, you will need to go to the 7th floor of Gould-Simpson during the first week of classes. There should be signs on the 7th floor directing you to the location of the electronic account application process. You can use any of these platforms to develop you MINIREL project, but it will be tested and evaluated *only* on `lectura`.

Code of Academic Integrity and Conduct: Students are responsible for understanding and complying with the University's Code of Academic Integrity. See <http://catalog.arizona.edu/policies/974/acacode.htm> and <http://w3.arizona.edu/%7Estudpubs/policies/ppmainpg.html>. Violations of the Code will, at minimum, result in loss of credit for a graded item. An egregious first violation or any second violation will minimally result in failure of the entire course.

Students with Disability: Students requiring accommodation in testing or note-taking must notify the faculty member and must deliver to the faculty member the Disability Resource Center faculty letter within the first few days of the course.

Topics and Tentative Schedule

Date	Topic	Description
Jan/14/04	Introduction	Course and project description, memory hierarchy and disk space management [1, ch. 9.1,9.3]
—	MLK Holiday	No Class on January 19
Jan/21/04	Disk modeling	Disk drive modeling [2]
Jan/26/04	Records and Pages	Fixed and variable length records [1, ch. 9.7], page formats [1, ch. 9.6], heap files [1, ch. 9.5, 8.4]
Jan/28/04	O.S. Issues	Operating System Support for Database Management [3]
Feb/02/04	Buffer Management	Basics [1, ch. 9.4], Non-stochastic buffer management [4], LRU-K: improved LRU replacement [5]
Feb/04/04	B-tree Index	Properties of indexes [1, ch. 8.2], B ⁺ -tree, duplicate keys, and bulk loading [1, ch. 10.3-10.8]
Feb/09/04	Multidimensional index	Overview [1, ch. 28.1-28.3], Grid file [6], k-d tree, quad-tree [7]
Feb/11/04	Multidimensional index	k-D-B tree [8] and hB-tree [9], R-tree and variants [10, 11]
Feb/16/04	Multidimensional index	Space-filling curves [12, 13, 14]
Feb/18/04	Query evaluation	Join algorithms: nested loop, sort-merge, hash join [1, ch. 14.4], Spatial join algorithms [15]
Feb/23/04	Query optimization	Overview of query optimization, Algebra equivalences [1, ch. 15.3]
Feb/25/04	Query optimization	Estimating the cost [16, ch. 11.1-11.8], System-R query optimizer [17]
Mar/01/04	Transaction processing	Concepts and problems, ACID model, and serializability [1, ch. 16.1-16.2]
Mar/03/04	Transaction processing	Transaction schedules: RC, ACA, ST [1, ch. 16.3]
Mar/08/04	Concurrency control	Two phase locking, phantom problems, granular locks, SQL isolation levels [1, ch. 16.4,17.1,16.6]
Mar/10/04	Midterm Exam	In-class Exam
—	Spring Break	No Class on March 15 and 17
Mar/22/04	Concurrency control	Page-oriented B-tree synchronization [18], tuple-oriented B-tree synchronization:ARIES/KVL [19]
Mar/24/04	Concurrency control	Lock management [1, ch. 17.2]
Mar/29/04	Transaction recovery	Failures, propagation and shadow pages, logging and WAL, Steal and Force, checkpointing [20]
Mar/31/04	Transaction recovery	ARIES recovery [1, ch. 18.1-18.6]
Apr/05/04	Parallel databases	Parallel database architectures and overview [21], parallel hash join [1, ch. 22.4.3]
Apr/07/04	Parallel databases	Multi-disk B-tree [22], Parallel R-tree [23]
Apr/12/04	Parallel databases	Multidimensional declustering [24], and other declustering techniques for multimedia
Apr/14/04	Decision support	Star schema, bitmap index, OLAP [1, ch. 25.2,25.6]
Apr/19/04	Decision support	Mining association rules [1, ch. 26.2.1], Apriori algorithm [25], Iceberg queries [1, ch. 26.2.2]
Apr/21/04	<i>k</i> -NN search	Branch-and-bound <i>k</i> -NN search [26], Multi-step <i>k</i> -NN search [27]
Apr/26/04	Sequence match	Whole- and sub-sequence match by DFT [28, 29], and DWT [30]
Apr/28/04	String data	String B-tree [31], Approximate string join [32]
May/03/04	Semi-structured data	Path and twig joins [33, 34, 35]
May/05/04	Data dissemination	Filtering XML documents for selective dissemination [36]
May/12/04	Final Exam	11:00-13:00, Gould-Simpson 701

Reading List and References ¹

Ordered By Lectures

- [1] Raghu Ramakrishnan and Johannes Gehrke. *Database Management Systems*. McGraw-Hill, Inc., New York, NY, third edition, 2003.
- [2] Chris Ruemmler and John Wilkes. An introduction to disk drive modeling. *IEEE Computer*, 27(3):17–28, March 1994.
- [3] Michael Stonebraker. Operating system support for database management. *Communications of the ACM*, 24(7):412–418, July 1981.
- [4] Hong-Tai Chou and David J. DeWitt. An evaluation of buffer management strategies for relational database systems. In *Proceedings of the 11th VLDB Conference*, pages 127–141, Stockholm, 1985.
- [5] Elizabeth J. O’Neil, Patrick E. O’Neil, and Gerhard Weikum. The LRU-K page replacement algorithm for database disk buffering. In *Proceedings of the 1993 ACM-SIGMOD Conference*, pages 297–306, Washington, DC, May 1993.
- [6] J. Nievergelt and H. Hinterberger. The Grid File: An adaptive, symmetric multikey file structure. *ACM Transactions on Database Systems*, 9(1):38–71, March 1984.
- [7] R. A. Finkel and J. L. Bentley. Quad-Trees - a data structure for retrieval on composite keys. *Acta Informatica*, 4:1–9, 1974.
- [8] John T. Robinson. The K-D-B-Tree: A search structure for large multidimensional dynamic indexes. In *Proceedings of the 1981 ACM-SIGMOD Conference*, pages 10–18, Ann Arbor, Michigan, April 1981.
- [9] David B. Lomet and Betty Salzberg. The hB-Tree: A multiattribute indexing method with good guaranteed performance. *ACM Transactions on Database Systems*, 15(4):625–658, December 1990.
- [10] Antonin Guttman. R-Trees: A dynamic index structure for spatial searching. In *Proceedings of the 1984 ACM-SIGMOD Conference*, pages 47–57, Boston, MA, June 1984.
- [11] Norbert Beckmann, Hans-Peter Kriegel, Ralf Schneider, and Bernhard Seeger. The R^* -tree: An efficient and robust access method for points and rectangles. In *Proceedings of the 1990 ACM-SIGMOD Conference*, pages 322–331, Atlantic City, NJ, May 1990.
- [12] J. A. Orenstein and T. H. Merrett. A class of data structures for associative searching. In *the 3rd ACM SIGACT-SIGMOD Symposium on Principles of Database Systems*, pages 181–190, Waterloo, Canada, April 1984.
- [13] Christos Faloutsos. Multi-attribute hashing using Gray codes. In *Proceedings of the 1986 ACM-SIGMOD Conference*, pages 227–238, Washington D.C., May 1986.
- [14] Bongki Moon, H. V. Jagadish, Christos Faloutsos, and Joel H. Saltz. Analysis of the clustering properties of the Hilbert space-filling curve. *IEEE Transactions on Knowledge and Data Engineering*, 13(1):124–141, Jan/Feb 2001.
- [15] Ming-Ling Lo and Chinya V. Ravishankar. Spatial hash-join. In *Proceedings of the 1996 ACM-SIGMOD Conference*, pages 247–258, Montreal, Canada, June 1996.
- [16] Jeffrey D. Ullman. *Principles of Database and Knowledge-base Systems*, volume II. Computer Science Press, Rockville, MD, 1989.
- [17] P. Griffiths Selinger, M. M. Astrahan, D. D. Chamberlin, R. A. Lorie, and T. G. Price. Access path selection in a relational database management system. In *Proceedings of the 1979 ACM-SIGMOD Conference*, pages 23–34, Boston, MA, 1979.
- [18] V. Srinivasan and Machael J. Carey. Performance of B-Tree concurrency control algorithms. In *Proceedings of the 1991 ACM-SIGMOD Conference*, pages 416–425, Denver, Colorado, May 1991.

¹The following 13 articles are required readings: [5, 10, 17, 19, 20, 23, 25, 26, 28, 31, 33, 34, 36]. The other articles will be used as references for lectures and discussions.

- [19] C. Mohan. ARIES/KVL: A key-value locking method for concurrency control of multi-action transactions operating on B-tree indexes. In *Proceedings of the 16th VLDB Conference*, pages 392–405, Brisbane, Australia, 1990.
- [20] Theo Haerder and Andreas Reuter. Principles of transaction-oriented database recovery. *ACM Computing Surveys*, 15(4), December 1983.
- [21] David J. DeWitt and Jim Gray. Parallel database systems: The future of high performance database systems. *Communications of the ACM*, 35(6):85–98, June 1992.
- [22] Bernhard Seeger and Per-Ake Larson. Multi-disk B-trees. In *Proceedings of the 1991 ACM-SIGMOD Conference*, pages 436–455, Denver, Colorado, May 1991.
- [23] Ibrahim Kamel and Christos Faloutsos. Parallel R-trees. In *Proceedings of the 1992 ACM-SIGMOD Conference*, pages 195–204, San Diego, CA, June 1992.
- [24] Bongki Moon and Joel H. Saltz. Scalability analysis of declustering methods for multidimensional range queries. *IEEE Transactions on Knowledge and Data Engineering*, 10(2):310–327, March/April 1998.
- [25] Rakesh Agrawal and Ramakrishnan Srikant. Fast algorithms for mining association rules. In *Proceedings of the 20th VLDB Conference*, pages 487–499, Santiago, Chile, September 1994.
- [26] Nick Roussopoulos, Stephen Kelley, and Frederic Vincent. Nearest neighbor queries. In *Proceedings of the 1995 ACM-SIGMOD Conference*, pages 71–79, San Jose, CA, May 1995.
- [27] Thomas Seidl and Hans-Peter Kriegel. Optimal multi-step k-nearest neighbor search. In *Proceedings of the 1998 ACM-SIGMOD Conference*, pages 154–165, Seattle, WA, June 1998.
- [28] Rakesh Agrawal, Christos Faloutsos, and Arun Swami. Efficient similarity search in sequence databases. In *the 4th International Conference on Foundations of Data Organization and Algorithms (FODO'93)*, pages 69–84, Chicago, IL, October 1993.
- [29] Christos Faloutsos, M. Ranganathan, and Yannis Manolopoulos. Fast subsequence matching in time-series databases. In *Proceedings of the 1994 ACM-SIGMOD Conference*, pages 419–429, Minneapolis, Minnesota, May 1994.
- [30] Kin-Pong Chan and Ada Wai chee Fu. Efficient time series matching by wavelets. In *Proceedings of the 15th Inter. Conference on Data Engineering*, pages 126–133, Sydney, Australia, March 1999.
- [31] Paolo Ferragina and Roberto Grossi. The string B-Tree: A new data structure for string search in external memory and its applications. *Journal of the ACM*, 46(2):236–280, March 1999.
- [32] Luis Gravano, Panagiotis G. Ipeirotis, H. V. Jagadish, Nick Koudas, S. Muthukrishnan, and Divesh Srivastava. Approximate string joins in a database (almost) for free. In *Proceedings of the 27th VLDB Conference*, pages 491–500, Rome, Italy, September 2001.
- [33] Quanzhong Li and Bongki Moon. Indexing and querying XML data for regular path expressions. In *Proceedings of the 27th VLDB Conference*, pages 361–370, Rome, Italy, September 2001.
- [34] Shurug Al-Khalifa, H. V. Jagadish, Nick Koudas, Jignesh M. Patel, Divesh Srivastava, and Yuqing Wu. Structural joins: A primitive for efficient XML query pattern matching. In *Proceedings of the 18th Inter. Conference on Data Engineering*, San Jose, CA, February 2002.
- [35] Nicolas Bruno, Nick Koudas, and Divesh Srivastava. Holistic twig joins: Optimal XML pattern matching. In *Proceedings of the 2002 ACM-SIGMOD Conference*, pages 310–321, Madison, WI, June 2002.
- [36] Mehmet Altinel and Michael J. Franklin. Efficient filtering of XML documents for selective dissemination of information. In *Proceedings of the 26th VLDB Conference*, pages 53–64, Cairo, Egypt, September 2000.