5DV021 Principles of Database Systems Fall 2008

1 Course Staff

Instructor:	Name:	Stephen J. Hegner
	Office:	C444 MIT-huset
	Telephone:	090 786 5760
	E-mail:	`hegner@cs.umu.se`
	URL:	`http://www.cs.umu.se/~hegner`
	Office hours:	1200-1300 on days with a scheduled lecture
Assistant:	Name:	Dennis Olsson
	Office:	D415 MIT-huset
	Telephone:	090 786 6701
	E-mail:	`denniso@cs.umu.se`
	URL:	`http://www.cs.umu.se/~denniso`
	Office hours:	Monday and Thursday 1015-1200

- Stephen J. Hegner is available to answer questions about the lectures and the course material in general, but cannot answer detailed questions about what is or is not acceptable as an answer to an obligatory exercise.
- Questions regarding the obligatory exercises should be directed to Dennis Olsson.

2 Course Language

All lectures will be given in English, and all written work must be submitted in English. Exceptions regarding written submissions may be made upon special arrangement with the instructor. For the final examination, it will be permitted to use an XX-English / English-XX dictionary, where XX is a natural language of the student's choice.

3 Course Literature

The official textbook for this offering of the course is the following.

1. Ramez Elmasri and Shamkant B. Navathe, *Fundamentals of Database Systems*, Fifth Edition, Addison-Wesley, 2006; ISBN: 0-321-41506-X (paper), 0-321-36957-2 (cloth).

While the fifth edition of this book is the official one, and all students are encouraged to obtain a copy, it appears to differ very little from the fourth edition in the sections which are covered in the course. The third edition, which is now somewhat dated, differs much more from these two. For completeness, course topics are listed below in Section 4 for all three editions.

In addition to the course text, there will be relatively detailed overhead slides. These materials will be available for download on the course web page.

4 Course Content and Outline

The official kursplan is available on this link. A more offering-specific outline is shown below. The numbers shown in the single rectangular brackets (i.e., [..]) identify chapters and sections in the fifth edition of the textbook, those in double rectangular brackets (i.e., [[..]]) identify those in the fourth edition, and those in triple rectangular brackets (i.e., [[[..]]]) identify those in the third edition. The numbers in angle brackets ⟨..⟩ indicate the approximate number of 45-minute lecture periods which will be devoted to the topic.

- Reasonably detailed overhead slides will be available for many topics. The authoritative source for relevant (i.e., possible examination) material is the course lectures and these slides. In many cases, material not covered in the textbook may nonetheless be covered in lecture presentations.
- The number of 45-minute lecture "hours" to be devoted to each topic is approximate. Adjustments may be made as the course progresses.

1 Introduction [1, 2] [[1, 2]] [[[1, 2]]] ⟨1⟩

2 Knowledge Representation for Database Systems

- 2.1 Entity-relationship modelling [3] [[3.1-3.7, 4.7]] [[[3, 4.7]]] ⟨2⟩
- 2.2 Overview of the relational model [5, 7.1] [[5, 7.1]] [[[7.1-7.3, 7.7, 9.1]]] ⟨1⟩

3 Query Processing and Constraints

- 3.1 Query Languages
  - 3.1.1 The relational algebra [6.1-6.5], [[6.1-6.5]] [[[7.4-7.6]]] ⟨2⟩
  - 3.1.2 The relational calculus [6.6-6.7] [[6.6-6.7]] [[[9.3-9.4]]] ⟨2⟩
  - 3.1.3 SQL [8.1-8.7, 8.9] [[8, 9.1]] [[[8.1-8.4]]] ⟨2⟩
- 3.2 Views [8.8] [[9.2]] [[[8.5]]] ⟨1⟩
- 3.3 Database programming and CLI/ODBC [9.1-9.3] [[9.3-9.5]] [[[-]]] ⟨3⟩
- 3.4 Dependencies and normalization [10, 11.1-11.2] [[10, 11.1-11.2]] [[[14, 15.1-15.2]]] ⟨4⟩

4 Implementation Issues

- 4.1 Physical database design [13, 14, 16] [[13, 14, 16]] [[[5, 6]]] ⟨3⟩
- 4.2 Database system architectures [-] [[-]] [[[17]]] ⟨1⟩
- 4.3 Query optimization [15.1-15.4, 15.5, 15.7, 15.8, 15.10] [[15.1-15.4, 15.5, 15.7, 15.8, 15.10]] [[[18]]] ⟨1⟩

4.4 Transaction processing and concurrency control [17, 18] [[17, 18]] [[[19, 20]]] ⟨3⟩

4.5 Recovery [19.1-19.4, 19.7] [[19.1-19.4, 19.7]] [[[21]]] ⟨3⟩

4.6 Security and authorization [23] [[23]] [[[22]]] ⟨2⟩

5 Selected Topics

5.1 Object-oriented and object-relational models [20, 21.1-21.3, 22.1, 22.2, 22.4-22.6] [[20, 21.1-21.3, 22.1, 22.2, 22.4-22.6]] [[[4.1-4.4, 11, 12, 13, D]]] ⟨2⟩

5 Course Materials Outline

5.1 Textbook Materials Outline

The following is a list of those chapters and sections which will be covered in the course. For each chapter or section, a symbol is given which indicates the nature of coverage in the course. The meaning of these symbols is provided in the table below.

✔	Material will be covered in the course.
✣	Material will be covered partially or selectively.

Note the following:

- If an entire chapter is covered, no section-by-section breakdown is given.
- Entries have not been provided for sections entitled "Summary" or the like.
- In general, omitted items will not be covered. However, the possibility that some covered material may appear in an omitted chapter or section remains. In all cases, the lectures and course notes should be taken to be the definitive statement for the course material.

1 Databases and Database Users ✔

2 Database System Concepts and Architecture ✔

3 Data Modeling Using the Entity-Relationship Model ✔

5 The Relational Data Model and Relational Database Constraints ✔

6 The Relational Algebra and Relational Calculus ✔

7 Relational Database Design by ER- and EER-to-Relational Mapping

7.1 Relational Database Design Using ER-to-Relational Mapping ✔

5DV021, Syllabus, page 4

8 SQL-99: Schema Definition, Basic Constraints, and Queries ✔

9 More SQL: Assertions, Views, and Programming Techniques

- 9.1 Database Programming: Issues and Techniques ✔
- 9.2 Embedded SQL, Dynamic SQL, and SQLJ ✣(SQLJ not covered)
- 9.3 Database Programming with Function Calls: SQL/CLI and JDBC ✣(SQL/CLI only)

10 Functional Dependencies and Normalization for Relational Databases ✔

11 Relational Database Design Algorithms and Further Dependencies

- 11.1 Properties of Relational Decompositions ✔
- 11.2 Algorithms for Relational Database Schema Design ✔

13 Disk Storage, Basic File Structures, and Hashing ✔

14 Indexing Structures for Files ✔

15 Algorithms for Query Processing and Optimization

- 15.1 Translating SQL Queries into Relational Algebra ✔
- 15.2 Algorithms for External Sorting ✔
- 15.3 Algorithms for SELECT and JOIN Operations ✔
- 15.4 Algorithms for PROJECT and SET Operations ✔
- 15.7 Using Heuristics in Query Optimization ✣
- 15.8 Using Selectivity and Cost Estimates in Query Optimization ✣
- 15.10 Semantic Query Optimization ✣

16 Physical Database Design and Tuning ✔

17 Transaction Processing Concepts ✔

18 Concurrency Control Techniques ✔

19 Database Recovery Techniques

- 19.1 Recovery Concepts ✔
- 19.2 Recovery Techniques Based on Deferred Update ✔
- 19.3 Recovery Techniques Based on Immediate Update ✔
- 19.4 Shadow paging ✔

19.7 Database Backup and Recovery from Catastrophic Failures ✔

20 Concepts for Object-Oriented Databases ✔

21 Object Database Standards, Languages, and Design

21.1 Overview of the Object Model of ODMG ✔
21.2 The Object Definition Language ODL ✔
21.3 The Object Query Language OQL ✔

22 Object Relational and Extended Relational Database Systems

22.1 Overview of SQL and its Object-Relational Features ✔
22.2 Evolution and Current Trends of Database Technology ✔
22.4 Object-Relational Features of Oracle 8 ✣
22.5 Implementation and Related Issues for Extended Type Systems ✣
22.6 The Nested Relational Model ✣

23 Database Security and Authorization ✔

5.2 Online Materials

There web site for the course is located at

http://www.cs.umu.se/kurser/5DV021/H08/index.html.

The following materials may be found on these pages.

1. This syllabus, in both PDF and HTML.
2. The lecture slides for the course.
3. Descriptions of the obligatory exercises.
4. Information on auxiliary software, such as the Leap database system.
5. Sample programs and other aids for ODBC.
6. Miscellaneous links to database-related things.
7. Some official documents required by the Department of Computing Science.

6 Laboratory Schedule and Computer Resources

There is no official laboratory booking for the course, nor any in-laboratory instruction. In general, when not reserved by a course, the computer laboratories of the department are open for use by students for their coursework.

The relational database system *PostgreSQL* will be used in this course, both as a stand-alone query processor and via interface to the programming language *C* via *ODBC*. In addition, the relational algebra query system *Leap* may be used in some assignments.

7 Course Schedule

The table below identifies the course meeting times and places, together with the nature of the meeting. The key "L" denotes a lecture, while "E" denotes an examination booking.

For each lecture, the topics to be covered are identified via the outline header number of Section 4 of this syllabus. So, for example, on November 20 the topics of 3.3, database programming and CLI/ODBC, will be covered. This is only an approximate assignment of meeting times to topics, and it may be altered as the course progresses.

Rooms whose identifiers begin with the letter *M* are located in MIT-huset, while rooms whose identifiers begin with an *N* are located in Naturvetarhuset. The examination rooms (skrivsalar) are in the building known as Östra paviljongen.

Week	Type	Date	Time	Room	Topics
45	L	Nov 06	0815-1000	MC313	1, 2.1
46	L	Nov 10	0815-1000	MC313	2.1, 2.2
46	L	Nov 12	0815-1000	MC313	3.1.1
46	L	Nov 13	0815-1000	MC313	3.1.2
47	L	Nov 17	0815-1000	MC313	3.1.3
47	L	Nov 19	0815-1000	MC313	3.2, 3.3
47	L	Nov 20	0815-1000	MC313	3.3
48	L	Nov 24	0815-1000	MC313	3.3, 3.4
48	L	Nov 26	0815-1000	MC313	3.4
48	L	Nov 27	0815-1000	MC313	3.4. 4.1
49	L	Dec 01	0815-1000	MC313	4.1
49	L	Dec 03	0815-1000	MC333	4.2, 4.3
49	L	Dec 04	0815-1000	MC313	4.4
50	L	Dec 08	0815-1000	MC313	4.4, 4.5
50	L	Dec 10	0815-1000	MC313	4.5
50	L	Dec 11	0815-1000	MC313	5.1
02	E	Jan 12	0900-1500	Skrivsal TBA	Final examination
13	E	Apr 15	0900-1500	Skrivsal TBA	Final examination
23	E	Jun 08	0900-1500	Skrivsal TBA	Final examination

8 Prerequisites

The formal requirements are listed in the course plan, which may be found at the following . They include the following.

1. A knowledge of programming in C in the Unix/Linux environment. This requirement is met by the formal prerequisite of the course *Systemprogrammering* (Systems Programming).

2. A thorough knowledge of data structures and algorithms, as presented in the course *Datastrukturer och algoritmer* (Data Structures and Algorithms), which is a prerequisite for the course in systems programming.

3. A knowledge of discrete mathematics and the formal foundations of computer science. This requirement is met by the courses *Datavetenskaps grunder* (Foundations of Computer Science) and *Diskret matematik* (Discrete Mathematics).

4. A knowledge of first-order predicated logic, as presented in the course *Logik för datavetare* (Logic for computer scientists).

This requirements should be met by students who are following a normal path of study in one of the programs of the Department of Computing Science. However, students from other disciplines who are considering this course should understand that the level of sophistication required in these topics is relatively high, and often not met by persons who work in other disciplines, even if they have a fair amount of practical programming experience.

9 Grading System

This course has only one part (*moment* in Swedish). There are not separate 'theory' and "programming" grades. Both the final examination and the obligatory exercises carry points, which will be added together to determine the final grade. Specifically, there is a total of 1000 points in the course grading system. These points are distributed as follows.

Obligatory exercises (5 @ 20 points each):	100 points
Obligatory software project:	100 points
Final examination:	800 points

Grade boundaries are as follows:

Number p of points	Grade
$p \geq 800$	5 (med beröm godkänd – excellent)
$650 \leq p < 800$	4 (icke utan beröm godkänd – very good)
$500 \leq p < 650$	3 (godkänd – satisfactory)
$p < 500$	U (underkänd – unsatisfactory)

In addition, to receive a passing grade (3 or better), it is necessary to obtain at least 400 points (50%) on the final examination.

10 Obligatory Work

10.1 Weekly Obligatory Work

The course includes five obligatory written and short computer exercises. The rules governing these submissions are as follows.

- Points will be assigned to the submitted solutions for each exercises. These points will be based upon the quality of the solutions, as well as their timeliness.
- Exercises will furthermore be graded as satisfactory or unsatisfactory. To receive a passing grade in the course, the student must have submitted a satisfactory solution to each exercise.
- Each exercise will have a due date. For each working day or fraction thereof that the submission is late, four points will be subtracted from the grade. (The grade may never be less than zero, of course.)
- Exercises marked as unsatisfactory may be resubmitted, in order that a satisfactory evaluation be obtained. However, resubmitted work will never receive any additional points.
- Each exercise will be worth 20 points. At the end of the course, the point totals on the obligatory work will be summed, resulting in a number between 0 and 100 inclusive. This number will constitute 10% of the final examination points.

10.2 Obligatory Programming Project

In addition to the weekly obligatory exercises, there will be one larger programming project, involving interfacing to a relational database using CLI/ODBC (Call-Level Interface / Open Database Connectivity) and the C programming language. The rules governing submission are as follows.

- The project will be given quality points, as well as marked as satisfactory or unsatisfactory. To receive a passing grade in the course, a student must have submitted a satisfactory solution for the project.
- Exercises marked as unsatisfactory may be resubmitted, in order that a satisfactory evaluation be obtained. However, resubmitted work will never receive any additional points.
- For each working day or fraction thereof that the submission is late, ten points will be subtracted from the grade. (The grade may never be less than zero, of course.)

10.3 General Remarks on the Obligatory Work

- The obligatory exercises may be completed in groups, and collaboration is permitted on the software exercises, roughly as described in the documents *Riktlinjer vid labgenomförande (Policy for Obligatory Exercises)* and *Hederskodex (Honor Code)*. More details will be provided later, when the descriptions of these exercises are distributed.

- The written exercises, as well as the programming project, may be submitted individually, or two or three persons may submit one solution. However, once a solution is submitted, only those named on the submission will receive credit for it. Partners in solution may not be added after the initial submission.

- Grading resources are limited. Therefore, work submitted well after the deadline may not be graded at all. If you must submit work late, discuss this with the course assistant.

- All obligatory exercises must be submitted for grading on or before the third and final examination (on June 08, 2009). No exercises will be accepted after that date.

10.4 Obligatory Work Completed in Previous Years

- Points for obligatory exercises do not carry over from previous years.

- If all of the obligatory exercises from a previous year were completed, then credit for completing the obligatory exercises, with zero points, will be awarded for the current year upon explicit request on the part of the student. This will not be done automatically; the student must make an explicit request.

- Submissions from different years cannot be mixed. Either credit is awarded for all exercises completed in a previous year, or else all exercises for the current year must be completed.

- To the extent that the problems are similar, work from solutions developed in previous years may be used in solutions for the current year. Such re-use must be explicitly acknowledged, and may not be shared with partners who were not part of development of the original solutions.