TDBC86 Database Concepts Fall 2001

Links to Sections of this Syllabus:

Course Staff
Course Language
Course Literature
Course Outline
Course Schedule
Computer Resources
On-Line Resources
Grading System
Weekly Obligatory Work
Obligatory Programming Project
General Remarks Regarding Obligatory Work
Obligatory Work from PreviousYears

1. Course Staff:

Instructor:	Name:	Stephen J. Hegner
	Office:	C444 MIT-huset
	Telephone:	090 786 5760
	E-mail:	hegner@cs.umu.se
Assistant:	Name:	Michael Minock
	Office:	D424 MIT-huset
	Telephone:	090 786 6398
	E-mail:	mjm@cs.umu.se
	Office hours:	Tuesday, Thursday 1500-1700
Assistant:	Name:	Jonas Westling
	Office:	D420 MIT-huset
	Telephone:	090 786 5452
	E-mail:	westling@cs.umu.se
	Office hours:	Wednesday, Thursday 1000-1200

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:

Main textbook:	Ramez Elmasri and Shamkant B. Navathe, *Fundamentals of Database Systems, Third Edition, Addison-Wesley,* 2000; ISBN: 0-8053-1755-1 (cloth), 0-201-52463-3 (paper).

4. Course Outline

The numbers shown in the rectangular brackets (*i.e.*, [..]) identify chapters and sections in the textbook. The numbers in angle brackets (*i.e.*, 〈..〉) indicate the approximate number of 45-minute lecture periods which will be devoted to the topic. Please note the following.

- The authoritative source for relevant (*i.e.,* possible examination material) is the course lectures. Material not covered in the textbook may nonetheless be covered in lecture presentations.
- The number of lecture periods to be devoted to each topic is approximate. Adjustments may be made as the course progresses.

1. Introduction [1, 2] 〈2〉
2. Knowledge Representation for Database Systems
   2.1 Entity-relationship modelling [3, 4.7] 〈2〉
   2.2 The relational model [7.1-7.3, 7.7, 9.1] 〈2〉
3. Query Processing and Constraints
   3.1 Query Languages
      3.1.1 The relational algebra [7.4-7.6] 〈2〉
      3.1.2 The relational calculus [9.3-9.4] 〈2〉
      3.1.3 SQL [8.1-8.4] 〈4〉
   3.2 Views [8.5] 〈2〉
   3.3 Database programming and CLI/ODBC 〈2〉
   3.4 Dependencies and normalization [14, 15] 〈4〉
4. Implementation Issues
   4.1 Physical database design [5, 6] 〈4〉
   4.2 Database system architectures [17] 〈1〉
   4.3 Query optimization [18] 〈2〉
   4.4 Transaction processing and concurrency control [19, 20] 〈4〉
   4.5 Recovery [21] 〈2〉
   4.6 Security and authorization [22] 〈1〉
5. Selected Topics
   5.1 Object-oriented and object-relational models [4.1-4,4, 11, 12, 13, D] 〈4〉

5. 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; "E" an examination. 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 7 the topics of 2.1, that is, entity-relationship modelling, will be covered. This is only an approximate assignment of meeting times to topics, and it may be altered as the course progresses. Classrooms with names beginning with N are located in Naturvetarhuset, the Natural Sciences Building.

Week	Type	Date	Time	Room	Topic
45	L	Nov. 5	0815-1000	N200	1
	L	Nov. 7	0815-1000	N200	2.1
	L	Nov. 8	0815-1000	N200	2.2
46	L	Nov. 12	0815-1000	N200	3.1.1
	L	Nov. 14	0815-1000	N200	3.1.2
	L	Nov. 15	0815-1000	N200	3.1.3
47	L	Nov. 19	0815-1000	N200	3.1.3
	L	Nov. 21	0815-1000	N200	3.2
	L	Nov. 22	0815-1000	N200	3.3
48	L	Nov. 26	0815-1000	N200	3.4
	L	Nov. 28	0815-1000	N200	3.4
	L	Nov. 29	0815-1000	N200	4.1
49	L	Dec. 3	0815-1000	N200	4.1
	L	Dec. 5	0815-1000	MA121	4.2, 4.3
	L	Dec. 6	0815-1000	N200	4.3, 4.4
50	L	Dec. 10	0815-1000	N200	4.4
	L	Dec. 12	0815-1000	MA121	4.4, 4.5
	L	Dec. 13	0815-1000	N200	4.5, 4.6
51	L	Dec. 17	0815-1000	N200	5.1
	L	Dec. 19	0815-1000	MA121	5.1
	E	Dec. 21	0900-1500	Skrivsal 4	Final Examination
04	E	Jan. 26	0900-1500	Skrivsal 6	Final Examination Re-test

6. Computer Resources

The specific software which will be used includes the relational database systems *Leap and Microsoft Access.* Further information will be presented as the course progresses.

7. On-Line Resources

Whenever feasible, images of overhead transparencies and the like will be made available on-line, at the web page for the course, which is

http://www.cs.umu.se/kurser/TDBC86/H01/index.html

8. Grading System

The final grade will be based upon a total of 1000 points, allocated as follows.

Final Examination:	80% (800 points)
Weekly Obligatory Work:	10% (5 @ 20 points each)
Obligatory Programming project	10% (100 points)

Grade boundaries are as follows:

Number *p* of points	**Grade**
$p \geq 800$	5 (med beröm godkänd)
$650 \leq p < 800$	4 (icke utan beröm godkänd)
$500 \leq p < 650$	3 (godkänd)
$p < 500$	U (underkänd)

9. 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. 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). 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.)

11. General Remarks Regarding Obligatory Work

- 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.

12. 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.