Introduction to Databases, Fall 2004

IT University of Copenhagen

Lecture 12: Course overview; Exam preparation

November 19, 2004

Lecturer: Rasmus Pagh

Today's lecture

- What you should be able to do after this course.
- Suggestions on how to study for the exam.
- Facts about the exam.
- Tips on writing a good exam.
- Some minimalistic exam solutions.

Next: What you should be able to do after this course.

Desired effects of your study in databases

Concrete effects:

You are able do carry out certain **tasks** related to relational databases:

- Design using E/R modeling.
- Database programming in SQL.
- Analyze the possible behavior of transactions, create indexes, etc.

Less concrete (but not less important):

- You have improved your ability to make ideas and concepts precise.
- You have improved your ability of abstract thinking.

Tasks in database design

Some basic tasks in database design:

- Given a description in words of a data set, draw a corresponding E/R diagram.
- Given an E/R diagram, perform the conversion into relations.
- Add to an E/R diagram certain multiplicity or referential integrity constraints.
- Add key constraints or referential integrity constraints to an SQL database schema.
- Given a relation instance what are the possible keys?
- Find functional or multivalued dependencies in a relation schema.
- Determine whether a relation schema is in BCNF, 3NF, or 4NF.
- Decompose a relation into BCNF, 3NF, or 4NF.

Tasks in database programming

Some basic tasks in database programming:

- Given an SQL statement, explain in words what it does.
- Given a database schema and a query in words, write the query in SQL.
- Explain the difference between two SQL queries.
- Rewrite an SQL query such that it does not use some specific feature of SQL (e.g., with no subquery).
- Write an SQL query corresponding to a relational algebra expression.
- Given a sequence of `GRANT` and `REVOKE` statements, state the privileges of each user.
- Write SQL to grant a user the right to access certain information.

Other kinds of tasks

Some basic tasks in other areas:

- Explain the difference between two relational algebra expressions.
- Given a data warehouse design, write the corresponding star schema.
- Given the description of a transaction, state what SQL isolation level would be appropriate for it.
- Given the description of two transactions, state what undesired event could happen if they were run at isolation level `READ COMMITTED`.
- Given some SQL query, state what kind of index could be used to speed it up.
- Given the time to execute certain queries with and without an index, and the frequency of updates to the underlying relations, determine what indexes should be created to minimize the total time.

Next: Suggestions on how to study for the exam.

The exam curriculum

The exam curriculum can be found on the course home page.

In short, the exam curriculum consists of:

- All parts of GUW and supplementary material written on the course schedule.
- The lecture slides.
- The example runs corresponding to the lecture slides.

Material *not* in the exam curriculum:

- Literature in square brackets on the course schedule.
- The lecture slides on commercial DBMSs.

Focus your effort using the lecture slides

The lecture slides focus on:

- The most *important* aspects of the course material, and
- The most *difficult* aspects of the course material.

In other words, you should spend most of your preparation time on getting a full understanding of the material on the slides (the book and supplementary material will be needed for this).

Things on the lecture slides you need *not* focus on:

- Information specific to Oracle: This was included for the sake of the exercises and will not be tested at the exam.

Reading the book

If there are parts of the curriculum that you haven't read, do so.

Also, you might want to re-read material in the course book.

I recommend that you:

- Read in the order of the book,
- *not* in the order suggested by the course schedule.

How to use the book's examples

The book contains a wealth of examples. It is a good idea to read an example in the book:

- If you remember things better when they are made concrete.
- If you are not quite sure that you have properly understood the material it exemplifies.

But: The exam will not require knowledge of any particular example.

- You can safely skip an example if you have understood the material it exemplifies.

How to use the SQL example runs

There are files with SQL example runs for many lectures. They were designed to illustrate many of the points in the book and lectures.

Going through them is a good check to see if you understood all aspects properly.

Again: The exam will not require knowledge of any particular example run.

- You can safely skip an SQL example run if you have understood the material it exemplifies.

How to use the exercises

Since the exam is written, you may want to review some of the exercises, and in particular the hand-ins.

Most relevant for the exam are (the parts of) exercises that do not involve using a computer.

Note: Many exercises in the textbook have solutions available on the web.

How to ask questions

What to do if you encounter a question related to the course:

1. Write it down.
2. Discuss it with your study group (it is recommended to find someone to study for the exam with).
3. If still in doubt, ask your teacher
   - Send e-mail to `pagh@itu.dk`.
   - There will be a question answering session Wednesday January 5, 10-12 AM, where also questions asked by e-mail will be answered.

Next: Facts about the exam.

Exam format

- Written exam, 4 hours, January 7, 9.00-13.00.
- "Open book" with all written aids allowed. You should bring:
  - The course curriculum.
  - Your notes.
  - Your answers to exercises and hand-ins.
- You may not bring a computer, but you may use a calculator (not vital).
- Remember to bring your study card (i.e., ITU key card).
- Your answer may be written in English or Danish. (In the latter case it is OK to use English terms if you don't know the translation.)

The structure of the exam

Similar to the two exams you have seen:

- There will be 4-6 problems, each focusing on a specific topic.
- Each problem will be marked with a percentage, indicating its weight in the grade.
- Each problem will be split into 1-4 questions.
- Questions will have varying difficulty, with a tendency that the last questions in each problem have the highest difficulty.
- Difficult questions do not necessarily count more than easy ones.

Start with the questions you find easiest (quickest) to answer!

How the grading is done

- A problem marked with X% is worth X points.
- The maximum possible score is thus 100 points.
- The questions in a problem are worth roughly the same amount of points, depending on the "size" of the question.
- The passed/not passed boundary is 50 points.
- To get an average grade (8) you should get around 70 points.
- To get a top grade (10–13) you must get more than 80 points.

Next: Tips on writing a good exam.

Understand the question, answer the question

- Take your time to make sure you understand the question! Failing to do so is a common cause of losing points.
- Answer the question, the whole question, and nothing but the question.

It will usually not give any (extra) points to:

- Answer a more general question than what is posed, or
- Write something that seems related to the question.

Write short

Questions will (usually) be posed with relatively short answers in mind.

If you find yourself answering a question using more than one page, you may have overlooked something.

A verbose answer is not necessarily bad, but takes longer to write.

— Explain the non-obvious, state your assumptions

Any fully correct answer will receive full points. However, it is a good idea to add some explanation to show your understanding:

- It can make it clearer that the answer is correct.
- If there is some error in the answer, the explanation might show that this is not due to lack of understanding.

If you need to make any assumptions to answer the question, you should state them explicitly.

Order, please!

- Start the answer to every problem on a new sheet.
- Do not write on the back of a sheet.
- Number the sheets in an order consistent with that of the problems.