

Mandatory exercise for INF5100, Advanced Database Systems, autumn 2004

Important notes:

The mandatory exercise is due **Tuesday 3rd November 2004, 16:00 o'clock**.

Please deliver your work to [Norun Sanderson](#) by e-mail! Paper copies will not be accepted!

Please make sure that what you deliver is runnable and readable. Deliver installation/run procedures if necessary, and explain them!

Team structure:

The exercise is to be solved by teams of three students. No less, and no more. If you have problems forming a group of 3, inform us. Actually, there is one more deadline to be met in addition to the delivery date: Send an e-mail to [Norun Sanderson](#) with the full names and e-mail addresses of the three team members before midnight Tuesday 14th September. Students who have strong reasons may apply to be allowed to solve the exercises alone or in groups of 2. Such applications must be in our hands one week before the team-forming deadline – i.e., before midnight Tuesday 7th September. But you should really have a very good reason. Please note that **teams of more than 3 students will not be allowed**.

The mandatory exercise:

The exercise consists of two parts, each with their own exercises. Both parts and all exercises must be solved.

PART I

- The following queries should be solved both by **using SQL** against the Oracle implementation of the database and by writing Java or C++ programs **using the ObjectStore** implementation.
- In addition to the code and answer, also report the time used by the computer to answer each query. No accurate measurements are required. You may use any technique you want, and it suffices to use an ordinary wrist-watch and report the results with whole seconds as resolution. You can also time-stamp your run in the code itself (before the start and after the completion of the run), and calculate the difference (total execution time) and print that out as well.
- Also report why you think the execution time is long or short for the two database systems and the different queries.

PART I, EXERCISE #1:

The movies are categorized. Make a list of all categories and the percentage of the movies that are in each category.

PART I, EXERCISE #2:

The sum of percentages in Q1 is different from 100. This is due to the fact that some movies are classified into several categories while others are not classified in any category. Find the maximum number, m , of categories any single movie is classified into, and make a list (table) showing the number of movies that are classified into exactly k categories for $k = 0, 1, 2, \dots, m$.

PART I, EXERCISE #3:

How many (distinct) directors are there in the database, and how many of them have themselves played an act in every movie they have directed?

PART I, EXERCISE #4:

List the full name of all directors who have made more than 5 movies and who have used one and the same actor in all their movies (which may be far more than 6).

Beware! This query is unsuitable for a "trial and error" approach. A wrongly put query could run for hours, so you should both feel confident that your query is correct before you submit it.

PART II

- The following exercises are meant to give you a better insight into the state-of-the-technology.
- They are short essays you will write as a result of your own surveys.
- They should be in a nice and readable format, with a proper title, abstract, proper references (books, papers, reports, URLs) etc, like a proper paper.
- Do not just refer to URLs, books, papers, reports etc. Explain what you found there, and what it means within the context of your survey.
- It is highly recommended that you use a known format – and preferably Lecture Notes in Computer Sciences (LNCS) format. This will also ensure that you have all that is needed in a regular paper.

LNCS has ready made templates for LaTeX, LaTeX2e, TeX, Framework and MS Word users. You can download them from the LNCS site: <http://www.springeronline.com/sgw/cda/frontpage/0,11855,5-164-2-72376-0,00.html>.

If you are using MS Word, you can also use the following three files:
§ File [sv-lncs.pdf](#) is a short manual of the MS Word template.

§ File [sv-lncs.dot](#) is the MS Word template.

§ File [typeinst.doc](#) contains short instructions and is written in the LNCS format, so that you can use it as an example and a template (just open the file, save it under the name of your choice, and start editing it to fill it with your own content, replacing what is there).

PART II, EXERCISE #1:

SQL 99 is a standard query language for what we have seen as object-relational or extended-relational DBMS. Conduct your own survey to answer the following question: How much of the SQL 99 is implemented (by few major DBMS vendors)?

Note that this essay can turn into a book, so take the most relevant few (say 3) elements of deviation from the standard (or compliance to the standard if there are no deviations). Explain why you think they are the most relevant elements. Keep it to 3 to 5 pages (i.e., the size of a short paper).

Use the following **title** for this essay: [The State of Implementation of the SQL 99 Standard](#).

PART II, EXERCISE #2:

SQL 99 is only one SQL-related standard. There are several other SQL-related standards that we do not take up explicitly in the course. Conduct your own survey to answer the following question: Which other SQL and related standards exist, and what areas/needs/issues (or application/technology requirements) are these standards meant to address?

Note that this essay also can turn into a book, so list up as many standards as you can, but write only one explanatory line for each. Do not detail all of them. Pick up only 3 standards you think are important or simply interesting, and detail them only. Explain shortly (for example in conclusions or summary) why you think those 3 are important or interesting. Keep the survey to approximately 3 to 5 pages (i.e., the size of a short paper).

Use the following **title** for this essay: [A Survey of SQL and Related Standards](#).

SOME USEFUL HELP FILES:

- To get started with ObjectStore, this link (which is from earlier semesters) may be useful: [startObjectStore](#)
- A description of the Oracle version of the test database (also from earlier semesters) is found here: [OracleInINF212](#)

- The file [moviedb-oracle.java](#) contains a program skeleton for using the movie database from java.
- The following help file is from earlier semesters: [oblig-help.pdf](#). It is written by Igor V. Rafienko (now at USIT). It contains useful information like descriptions (models) of the movie database etc.

HAPPY THINKING, CODING, SURVEYING and WRITING!
PLEASE DO ASK WHENEVER YOU NEED TO!