

DD1368 VT18 Introduction to Databases

LABORATION 2: Development

RULES

This lab takes a lot of time and effort. Read the rules to make sure that your solutions are allowed, or you *will* have to redo them.

#1: The labs must be done in groups of exactly two people. No larger groups are allowed, and if you have extraordinary extenuating circumstances that force you to do the labs alone, you must obtain permission to do so from the course leader. Both students in a group must be able to present all of the lab for the group to pass.

#2: You must present correct and valid solutions to all the given problems to pass the lab.

#3: You are free to use any combination of programming language, **relational** database server and programming environment to solve the lab. It is entirely up to you. You will use your own computer, install and configure all the software you need to implement your solution.

1. Project background

You have recently graduated and got a job as a software developer at a mid-sized insurance company that has a lot of ongoing business meeting that they need to keep track of. They run a regular CRM (customer relationship management) system, but are dissatisfied with the booking capabilities and want you to design, produce and deploy a dedicated booking system for meetings.

2. Capability requirements

Your system must be able to track all people in the company, complete with their positions and the teams they belong to. It must also keep track of all the people that belong to business partners, complete with whom they represent and their positions. It must keep track of available resources in terms of meeting rooms, facilities of such, and time when they are booked. Meetings must also be tracked, with booked rooms, participants and time slots. Costs for booking rooms vary with facilities and time allocated and must be booked against the team that the person booking the room belongs to. A log of costs accrued by the various teams must be kept. The system must keep track of booked rooms and times and reject any booking attempts that would result in overlaps.

3. Functionality that the system must support

Insert new users, delete old users, insert and delete teams (all this must *not* affect the log of costs already accrued). Answer queries about what rooms are available for a given time slot. Add new meetings, but reject attempts to book meetings that would overlap with already booked ones. Delete meetings that have not occurred yet. Present occupation lists for all rooms on a given date. Show which users have booked which meetings. Show all participants of a given meeting. Show cost accrued by teams for any given time interval.

Most of these you are allowed to showcase by manually performing SQL queries against your database. *However*, you will need to produce two graphic interfaces for the tasks of booking a meeting (and possibly having your attempt to do so rejected, if it turns out the meeting would overlap with one already in the system), and to remove meetings from the system (that haven't already been held). These interfaces must use normal GUI elements (no terminal windows) and not expose the users to any coding or SQL usage. Furthermore, all choices and options that the user can select must be populated by data from the database. Your interface for booking must allow the user to choose between rooms, view their facilities, and select the people who are to attend a meeting, as well as start and end time the meeting to happen. All these choices must be provided through interface elements - no typing whatsoever required for the user. Your interface for removing bookings must offer the ability to choose which meetings to cancel and reject attempts to cancel by any user who didn't book that particular meeting. Again, there must be no typing.

4. What you must prepare before starting programming

You will design the database. Your design must be presented as an E/R-diagram and as the corresponding physical database table implementation, complete with foreign key integrity constraints and any other relevant constraints. Make sure your design meets all the capability requirements and can implement all the functionality that the system must support. It is recommended that you show your E/R-diagram to a lab assistant before embarking on programming your interfaces. In addition, you will implement your design in your relational database of choice. You will then fill the database with enough data that you can demonstrate all the functionality required by running SQL queries against it. You will prepare SQL queries for all the functionality required in (3) above, except those that are to be run from your GUI.

5. Presentation

You will bring your computer with everything ready to run to the presentation. In addition you will bring printouts of your E/R-diagram and of the table definitions. Be prepared to discuss and motivate those the choices you've made. You will have all required SQL queries on file ready to run immediately. The assistant will ask you to run queries against your database at the prompt and to demonstrate the functionality of your GUI:s. In addition, you will have to discuss your source code and answer questions about it.