CS-A1150 Databases, Spring 2023
Project, Part 1 (UML Data Model and Relational Data Model)

1. Introduction

The purpose of the project is to learn to design and implement databases in practice. The project consists of two parts. In this part, you are asked to design a database using UML and define the relations based on the diagram. In the second part, you will implement the database (using SQL) you designed in this part.

The project is designed for groups of 2-3 students. It is possible to use the course Zulip channel (stream #search_for_temmates) to find a group. **You are not allowed to do your project alone without a group.** Both project parts must be done with same group.

If something in this project description is not clear enough, you can use the course Zulip channel (stream #project) to ask questions. However, all details are not defined, because one of your tasks is to make decisions about the details.

2. Instructions

In this project you are asked to design a database that stores information about the university's courses, facilities, space reservations, employees, students, registrations and courses completed by the students.

Tasks

- Draw an UML diagram based on the information in the following sections. Use the notation taught in the course and in the course text book.
- Convert the UML diagram to the relational data model. Present the schemas of the relations and underline the attributes which form the key for each relation.
- **Provide answers (with their justifications) to the following questions**: What are the functional dependencies of the database? Are there any form of redundancy or other anomalies in the database structure? Is your database in Boyce-Codd Normal Form? If it is not, please use the decomposition algorithm to decompose the relations to Boyce-Codd Normal Form (submit both original and decomposed version).

3. Detailed Information about Database

Operating environment

- The database contains information about which courses are in the curriculumn of the university. The information about a course includes the course code, the name and number of credits .
- The same course can be organized several times either in the same semester or during different semesters.
- The course can have lectures, exercise groups and exams. Lectures and exercise groups are related to one course instance (the specified implementation of the course), but the exams to the course itself.
- The lecture times of the course are not necessarily the same every week, but may vary.

- A certain exercise group can meet several times and the meeting times are not necessarily the same every week. The group can also meet several times during the same week.
- A certain course may have several possibilities to take an exam possibilities (exam immediately after the course implementation and re-take exams).
- The student registers for the course by signing up for one of its exercise groups. For the sake of simplicity, it is assumed that all courses have at least one exercise group for each course instance. Exercise groups can have a maximum number of participants, and a student can only register for an exercise group where there is room.
- The student must register separately for the exam in which they want to take part. The date of registration and the language choice of the exam questions are stored for each exam registration.
- For each student, the information includes at least the name, student ID, date of birth, degree program, when the student has been enrolled at the university, and when their study right ends.
- The university has several buildings with spaces intended for teaching use. These spaces are called halls below. For each building, the information contains at least the name and street address of the building and which halls are located in the building.
- For each hall, the information includes how many seats there are for students and how many examinees can fit in the hall during the exam (this number can be smaller, because during the exam students cannot sit too close to each other). Different halls have different equipment (e.g. a video projector, two video projectors, a teacher's computer, a document camera, computers for students, etc.) The system must also be able to store information about other equipment in the halls than those listed above.
- Information about all reservations of the halls is stored in the system. In the same hall, there can be an exam for several courses at the same time, but only one lecture or exercise group at a time. The reservation may be longer than the occasion for which the hall is reserved (for example, in exams, extra time is reserved before the start of the exam and after it ends).
- Every reservation has a unique reservation id. In addition, the information about a reservation at least contains the occasion the reservation is for, when the reservation was made and which employee made the reservation.
- For each employee of the university, the system stores at least a unique identifier, job title, name, address, phone number and the start and possible end date of the employment relationship. The data of only one employment relationship has been recorded for one employee (so there is no need to take into account the possibility that the same employee may have several fixed-term employment relationships.) In this project, the employees' data is not used for anything other than keeping records of the makers of the reservations.
- The student completes the course by taking the course exam. (To simplify the task, it is assumed that there are no other ways to complete the courses.) The system maintains information about which exams the student has participated in and which grade they have received from each exam (including those exams where the student's grade has been failed). Participating in the exam is not the same as registering for the exam, because the student may not come to the exam for which they have registered.
- The student can increase the approved exam grade at most once. If the student already has two approved exams of the same course, the student can no longer register for the exercise group or the exam of this course. There can be an arbitrary number of failed exams and they do not affect whether the student can register for the exam or the course. (The mentioned requirement cannot be expressed directly in the UML diagram, but the structure of the database must be designed so that it is possible to check this.)
- Students can get points from exercises. The points are always related to a certain course instance. A student can therefore have a certain number of points from the autumn 2021 course session and another number of points from the autumn 2022 course instance. In order to simplify the task, it is not necessary to store this more precisely, e.g. from which week or

round some points come from, but it is enough to store information about the total number of points of the student's exercises for each course instance and to be able to update these points. With the help of exercise points, the student can get bonus points for some exams, but the awarding of bonus points is done outside the database.

For example, the following database operations must be possible (in addition, your system can support more operations). These operations are not presented in the UML diagram and in the relational model, but you must design them such that the operations are possible:

- Store information about courses and course instances, students, course lectures, exercise groups and their times, exams and student registrations, exam grades and exercise points.
- Store information about the buildings of the university, its halls and hall reservations. You must also be able to store historical information about the reservations, i.e. you must be able to find out, for example, which course and which exercise group the hall was reserved for at the specified time last year.
- Store information about employees of the university and the secret hall reservations made by them.
- Find out which courses are organized at the desired time period.
- Find out which exams a certain course has in the desired time period.
- Find out when a certain course has been organized or when it will be organized.
- Find out which lectures belong to the course instance you are interested in.
- Find out which exercise groups are belong to a certain course instance and when and where a certain exercise group meets.
- Find a hall which has at least the desired number of seats and which is free at the desired time.
- Find out the purpose for which a certain hall has been reserved at the desired time or who made the reservation.
- Register the student for an exam or exercise group. For this, you must be able to check that the student does not already have two appected exam results from this course.
- Find all students registered for a certain course instance, exercise group or exam.
- Find the exercise groups of the desired course instance that still have space.
- Store the student's course (exam) grade and find which the student has from a certain course.
- Store exercise points to the student for a certain course instance and find out how many exercise points the student has received in a desired course instance.

Please note that the database does **not** contain information on, for example, the following things:

- Students' registrations for semesters.
- Information about the teachers of the courses.
- More detailed information about students than the information mentioned above.
- More detailed information about employees than the information mentioned above.
- Information about the degree requirements of the degree programs, i.e. which courses the students must complete.

4. Principles of Grading

All members of the group will get the same grade of the project. The project consists of two parts. The parts are worth 20 points each.

- To pass the project, you must get at least 20 points in total and submit the solution to the both parts.
- If you get at least 30 points, your course grade will be increased by one full grade (however, you must pass the exam without this bonus!)

The project grade is valid until the first exam in Spring 2024, but not after that.

You can get up to 20 points for this project part:

- UML modeling 8 p
- Converting the UML diagram to relational data model 4 p
- Functional dependencies, BCNF normal form, etc. 4 p
- The quality of documentation 4 p

5. Submitting the Project

The project will be submitted to A+-system (Module *Project Part 1*). The deadline of the first part is Apr 4th 2023 at 8.00pm. If your submission is late at most 7 days, your submission will be graded, but you will lose 5 points, if the delay is 3-7 days and 3 points, if the delay is under 3 days. If the delay is over 7 days, your project will not be graded except if you have an exceptional reason (for example, a long sick leave signed by a doctor).

The only acceptable file format is PDF. You may write in English, Finnish or Swedish.

The report must consist of the following parts:

- Cover page, which includes students' names and e-mail addresses.
- UML diagram
- Relation schemas converted from the UML diagram.
- Explanation of the solution (recommended length 3-4 pages). For example, you can explain how the functions described in Section 3 can be performed in your system and explain unusual solutions.
- Answers to the questions given in Section 2 about normal form, functional dependencies, anomalies and the decomposition into Boyce-Codd Normal Form when needed.