Seminar 2

Edvin Frosterud

November 2022

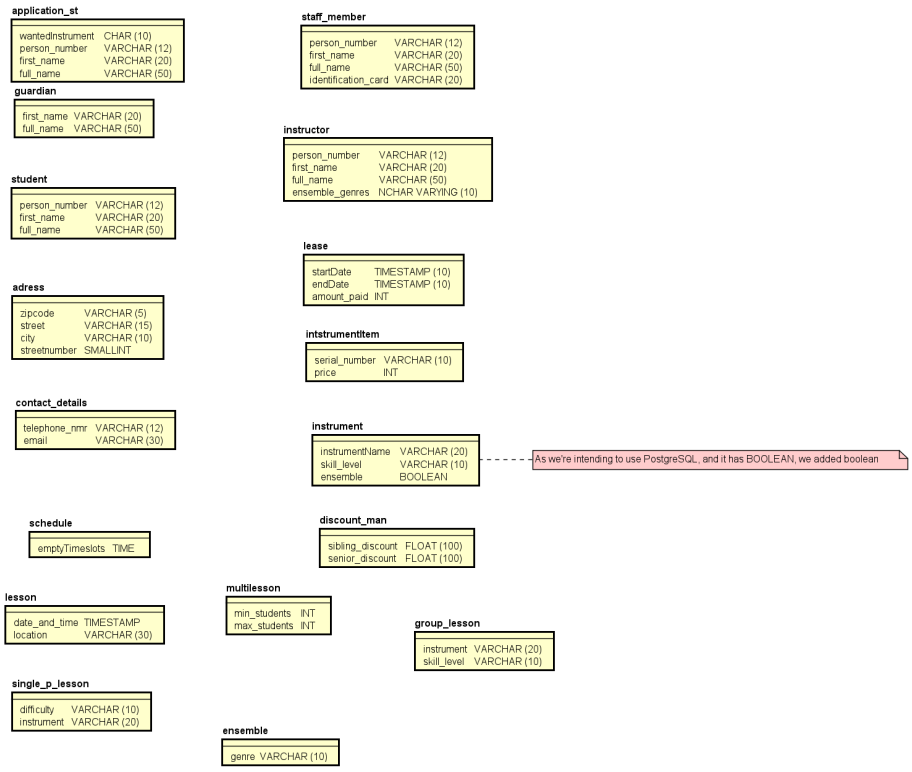
Changes for seminar 5 will be in italics. Changes follow the instructions given to us by Niharika. In this particular instannce, she had a problem of not being able to run our fill-database, which we didn't have. I'm unsure if this will happen again, but will include prints to make sure it is clear that it does work on all three systems I've tested it on + my partners system, so it could be a difference in PostgreSQL version? At this point I'm still unsure as to that comment.

1 Introduction

Seminar 2 was a mess for me and my partner, Yas Asghari. Our other course is extremely heavy, and I alone put 60+ hours into it this past week, so sadly we couldn't check all the boxes I wanted to. The idea of this weeks assignment was to transform our previously made Seminar 1 Conceptual Model into an model that can actually be used for a database. So, since we tried making the Inheritance last Seminar we had the choice of using that, slightly faulty inheriting model or a working non-inheriting. The decision was made to use the inheriting model, which I think I sorely regretted later down the line, due to both time and the lack of inheritage instructions.

2 Method

A decision was made early that the project would use Astah as the primary editor for the logical model. The second step was simply keeping up with the lectures, and after making a lot of mistakes, we managed to make a diagram, which then led us to be able to try to make a SQL database. The first step was cleaning up our previous logical model, removing and adding some changes according to the feedback that was previously given in Seminar 1. We then removed all connections, and followed the youtube lectures order of steps. First declaring a bunch of tables with equivalent names to our old entities.



2.1 Columns with column types

After adding tables, adding columns, or properties was the next relevant steps. Most tables were taken directly from the previous properties in Seminar 1. After I did this a few nights ago, this is the result after adding tables. Adding table types was pretty easy, as most of them are varchars with a limitation that is logical, for example 12 for personal number or a limit in phone numbers. We had to add text, boolean, and later the type for generating ints for primary keys.

2.2 Primary keys for strong entities

Primary keys were added to all strong entitites, but that was leading us to a lot of inherited keys, so some keys were trimmed and connections were changed in order to account for that. Primary keys were originally all called id or id_key which I later had to fix, due to the existance of foreign keys. Adding connections was also done in this stage, but these connections were always iterated on by discussion with my partner and other peers. Sadly I don't have a picture for this specific part of the project, but we'll see the later iterations how it turned out.

2.3 SQL conversion

After making our model, the next step was converting it to an SQL file to act as the first task, which in this assignment is making a script that generates the necessary tables and relations between tables and their columns. We took the SQL out of astah, and mostly through help from out peers and googling strange error messages we were able to scratch by without errors. At this stage some of the parts had to be omitted as we realised they were interfering with eachother, which we then reflected and moved to our astah file too. All of the tables we named with "UNIQUE" had to be made unique with the unique modifier at the end of column declaration.
After handin, we got criticism, and as this course requires a very heavy constant workflow we fell behind on the next seminars due to this seminar needing fixes. It would be nice to see the seminars go further into the exam period or similar, and seperate them with a more chill week where rehandins for the previous seminar would be. Anyhow, the method required was pretty extensively described in our criticism. Removing booking-manager, as that didn't make sense anymore as theres no data that cant be stored elsewhere there. We decided to move that data to schedule. It just doesn't make sense for a logical model, since that managing logic will be handled by the computer integration layer anyhow. Removing the leftover connection from student-lesson was also fixed. This one was a bit weird, as in Astah, and in the file that connection doesn't show up, and it only appears when the export function is used. This is probably a visual glitch, so I fixed that by simply removing it in the left sidebar where the actual connections are. The third problem was the connection between Guardian(s) and Student(s). To fix this, we simply removed the many to many relation and changed the layout to accomplish a similar result. Each student now only has one guardian allowed, and a Guardian can have many children allocated. This was mostly because we didn't want to make a cross reference table. We exchanged the studentID in InstrumentItem to be a leaseID, giving lease an ID and making the InstrumentItems non-identifying. We should've probably also put a studentID here, as this would be easier later for seminar 4, but nonetheless we didn't. .

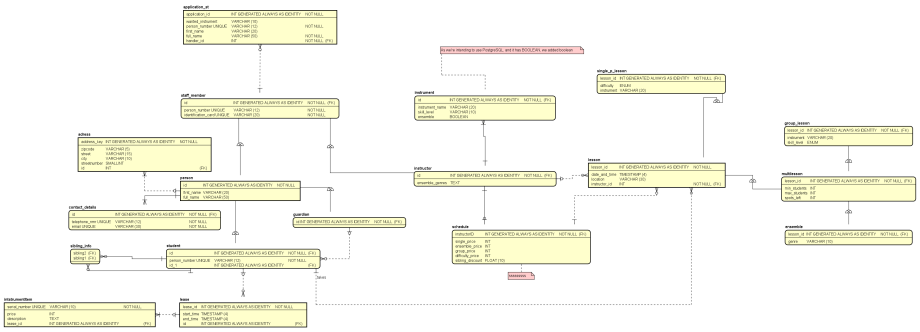
3 Result

3.1 New Final model

3.2 Creating a database

https://github.com/Froosty11/IV1351/tree/main/Seminar2

Output, to show that we don't have any issues except minor warnings can be found in the output.txt file on github. We do have warnings, but that is not the problem that was described by Niharika. *Both the fill-database and create-database were restructured from scratch and have been tested lots of times. The new model is very similar to the old one, but Fill has been improved significantly.*



Output, to show that we don't have any issues except minor warnings can be found in the output.txt file on github. We do have warnings, but that is not the problem that was described by Niharika.

4 Discussion

See previous discussion below. I do believe this model and creation/fill scripts are significantly better. I've made sure we're following naming conventions with fixing some camelcase to all lowercase. Crow foot notation is correctly followed. Third normal form is followed correctly, and back when I wrote the original one I wasn't completely sure, but I've since then increased my knowledge. All our tables are relevant. I do think there is some discussion to be had about the structure of the subtyped lessons, specifically the existance of Multi-lesson. In practice, the only usecase of having this parent type is to simplify the child types, but in practice it hasn't done anything for us except that we can select from the Multilessons to select both, which is nice, but this could be easily handled with a more advanced query or a view. We do have columns for all information that I think is necessary and everything either has foreign keys or primary keys. Primary keys are all unique, and I do think our logical choices for keys are very good, perhaps the worst one being the key solution for instrument_item and its leases as that was quite a trouble in Sem4. All relations are correct and cardinality is correct. It might actually be slightly different in code as we've changed stuff back and forward and kind of lost track of keeping changes on both SQL and in code. It is possible to do all required functions and we even improved functionality since the previous handin by increasing the usability of the sibling system. I do still think we should improve our use of notes, as the current system has a limited amount of notes that improve readability, but perhaps not actual function. Didn't get a complaint on my complaint of this, so I assume it's okay to assume we're all knowledgeable about the assignment. Yes, values in theory should be handled using other values, but there is no logic to do this in this assignment, as that is, in our humble opinion better handled in a programmatic layer, as that puts less constraints on the database in case of special access/cases. Enums are used, we did also use them for the instrumenttypes later, but in this

sem2 iteration it's only difficulty.

- *Are naming conventions followed? Are all names sufficiently explaining? In my opinion, yes it is.*
- *Is the crow foot notation correctly followed? Yes, we didn't change it from last time.*
- *Is the model in 3NF? If not, is there a good reason why not? I am actually unsure about this one. It was late, I really didn't get the last parts of the lectures.*
- *Are all tables relevant? Is some table missing? Yes. Our tables are relevant.*
- *Are there columns for all data that shall be stored? Are all relevant column constraints and foreign key constraints specified? Can all column types be motivated? Yes, columns exist for everything. I think we should've made the keys better.*
- *Can the choice of primary keys be motivated? Are primary keys unique? Yes, and they're unique or sometimes passed.*
- *Are all relations relevant? Is some relation missing? Is the cardinality correct? I believe so.*
- *Is it possible to perform all tasks listed in the project description? Sibling check note is actually not there currently, but in my opinion, the solution to that, even though our last seminar also complained about this is just looking at the childs parents children, which allows you to access possible children.*
- *Are all business rules and constraints that are not visible in the diagram explained in plain text? Why do we need to do this here again? Yeah, I should've had more notes, but I must've missed this point when originally reading and summarizing the assignment. To me this seems unneccesary.*
- *Are there attributes which are calculated from other attributes and then written back to the database (derived attributes)? If so, why? Records of student fees and instructor payments might be examples of such attributes. I don't think this currently is working- some help would be nice on this but as previously stated time constraints.*
- *Are tables (or ENUMs) always used instead of free text for constants such as the skill levels (beginner, intermediate and advanced)? Yes, only because I read this right now.*

5 Sources

Youtube channel Leif Lindbäck(Link: https://www.youtube.com/channel/UCs3aBf$_5$KdxjKnNVFk$_H$rSQ)