Task 2

Data Storage Paradigms, IV1351

Leo Halfar lhalfar@ug.kth.se

23-11-23

1 Introduction

In this report the solution to a logical model with physical aspects based on a conceptual model from the previous task will be discussed and explained. The translation from the logical model to SQL code to actually create the database will be explained and discussed as well. This assignment was done in cooperation with Otto Svenberg.

2 Literature Study

In order to solve the task a number of thing where studied and analyzed. This included the video on normalisation, logical and physical models, the tips and tricks pdf as well as the website https://www.postgresqltutorial.com. This gave the author enough knowledge about hoe to construct the model as well as how to use pgadmin and sql shell in order to create and input data into the database.

3 Method

The logical and physical model was created in Astah in an ER diagram using IE notation. From the videos the steps to achieve this was the following:

- 1. Create a table for each entity
- 2. Create a column for each attribute with at most one value
- 3. Create a new table for each column with a higher cardinality
- 4. Specify type for each column
- 5. Consider column constraints
- 6. Assign primary keys to all strong entities. Prefer surrogate keys
- 7. assign foreign and or primary keys for one-to-one and one-to-many relations

- 8. create a cross-reference table for each many-to-many relation
- 9. Assign FK to tables representing multi-valued attributes.
- 10. Verify that the model is normalized
- 11. Verify that it is possible to perform all planned operations on the data.

First all entities from the conceptual model where put as entities, all attributes with as most value 1 from the conceptual model where put as columns in the logical and physical model. The columns with a higher cardinality which was sibling and instructor instrument where made as new tables. Subsequently the type for each column was specified, Primary keys where assigned to all strong entities. Here the decision to remove address and put its columns in instructor and student was made to prohibit address from being a strong entity. Thereafter foreign and primary keys where added depending on the type of relation between entities and if the author thought a weak entity had meaning on its own or not. A cross reference table was made for the relation between instrument and instructor since they had a many to many relationship. FK was assigned to the sibling table since it represented multi-valued attributes. Lastly the author verified that the model was normalised and that all planned operations could be performed.

The script for creating the database as well as inserting data into it was written in pg admin and to make sure the scripts worked as intended sql shell. The code for creating the database was written by simply looking at the model and adding all the tables and columns in the code. Thereafter the code for inserting data was written, at least one insertion was written for all entities and the script was then verified in sql shell.

4 Result

IN figure 1 the solution to the logical model can be found. As seen in the figure the model has a similar layout as the conceptual model apart from a few changes. These changes where made according to the feedback from the last task as well as the method. For example the instrumentKnown attribute from the conceptual model is now part of a croos-reference table.

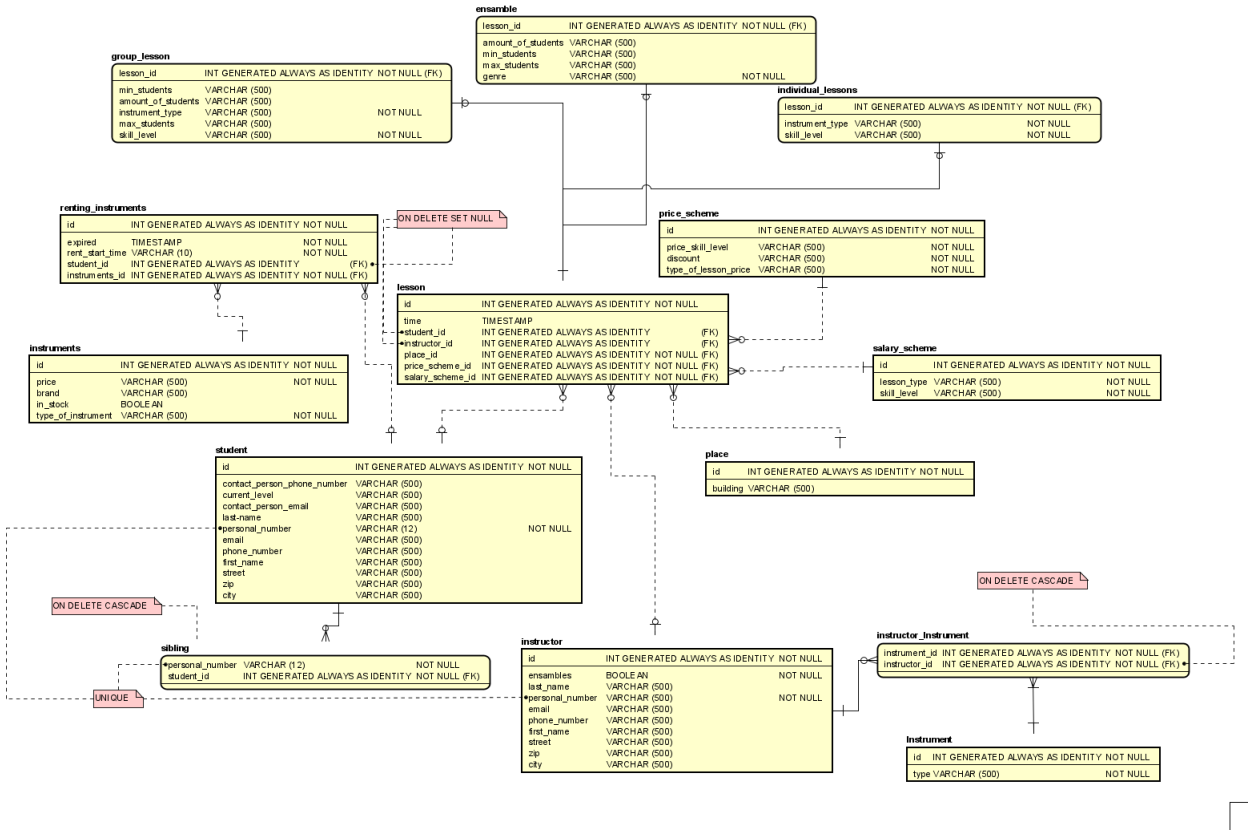

Figure 1: Logical and Physical model.

The SQL script for data input and creating the database can be found in the github repository below: https://github.com/LeoHalfar/IV1351

5 Discussion

The model includes everything needed for Sound good, the SQL script for creating the database is an accurate translation of the model. All naming conventions are followed, the names are sufficiently explaining and the IE notation is followed correctly. All choice for primary keys can be motivated either by following the method or by logical reasoning, for example the renting instruments table isn't a strong entity but one can reason that the renting has meaning even without a student. As mentioned previously, the address table was removed. This was because it was a strong entity and even if you could have addresses not connected to an instructor or student this is not wanted in the database since it doesn't add any value. The model only allows for the removal of strong entities(now tables) for instructor and student. This can be discussed as to if this is the correct approached, however the author reasoned that instructors and students

come and go while lessons shouldn't be removed and tables like place are very unlikely to need a warranted removal.

6 Comments about the course

Estimated total time spent: 20 hours