Task 1 Conceptual Modeling

IV1351 Data Storage Paradigms

Klara Lindemalm, klindema@kth.se

Fall 2022

Introduction

The purpose for the first task of the project was to understand and build a conceptual model for a specific case. The case given for this project was to build a Soundgood music school. Rasmuss Per Brodin was a collaborator for this project.

Literature Study

In order to prepare for the task, course material was provided on the canvas page including a complex description of the project. This was read later on when we had achieved a better understanding of the concept of conceptual modeling. For the initial preparation, video lectures on conceptual modeling along with recommended readings, chapter 3 from the main book and section 6.1 to 6.4 in the additional course book was read thoroughly along with a document of tips and tricks. The videos instructed a structural approach to conceptual modeling and provided the necessary terminology used. The books provided a more comprehensive description of conceptual modeling and provided great example diagrams that where used as inspiration for the design. The document of tips and tricks helped us with misinterpretations of attributes and derived data. Additionally we attended the lectures and the tutorial occasion on site for better understanding and hand on help from the teachers and instructors. For the specialized task to implement an inheritance relationship we went through section 4.1 and 4.3 in the main book to gain a better understanding of the relation and how to implement it correctly in a diagram.

Method

For the conceptual model we choose to implement the diagram with the diagram editor Astah Professionals and created an Entity Relationship diagram (ER diagram) with IE notation for the Soundgood music school using the following six steps.

1. Noun identification
2. Category list
3. Removal of unnecessary entities
4. Change of suitable entities to attributes
5. Add associations
6. Review of the setup

For the first step we did a noun identification of the project description provided in the project page in canvas, meaning all nouns in the text where sorted out and added as entities in the diagram software. We added suitable entities from the category list, provided in the course material, some additional entities where added in a few of the cases in the list even though many of these became irrelevant in the final diagram. For the third step we removed most of the redundant entities and started to structure the diagram. Additionally relationships where added between the entities, so that one could follow the associations between all entities, including cardinality and description of each. Finally for the review, we truly evaluated the diagram to make sure that nothing was unnecessary and most important of all that nothing from the initial project description was missing. For the last part not mentioned in the step by step method we created an inheritance relationship as a specialisation for the model and evaluated how beneficial this would be for the actual database and storage model.

Result

Our conceptual model displayed in figure. 1 was build with the Booking entity centered in the middle to maintain an keep track of every lesson available for a student to enrol and an instructor to teach, this was set with a primary key of a booking ID. These booking are related to Lessons of the three different types, ensemble, group and individual lesson. The idea was that some of the attributes are null when not applicable for the specific type lesson, such as an individual lesson does not require a genre. The lesson price entity relates to lesson so that one could derive a price for the lessons provided for each student and furthermore set a salary for the instructor who held the entity. The instructor and the student entity inherit properties from the Person entity with identifiable attributes, social security number, name, phone number, email address and skill level, where the skill level contains information about what instrument and at what level that instrument is played at. Furthermore the student relates to a contact person so that one could find emergency contact details for each registered student in the system. This entity could have been related to the person entity such as instructor and student but we found that all information would be unnecessary for the contact person such as skill level, therefore the contact person did not inherit the person. Lastly we set up the a leasing entity, InstrumentLease with a leasing ID, relating to Instruments where one could check for availability when setting up a leasing agreement for a student. This entity holds a date so that one could derive the time of rental and how long the student can keep the leasing agreement for a specific instrument.

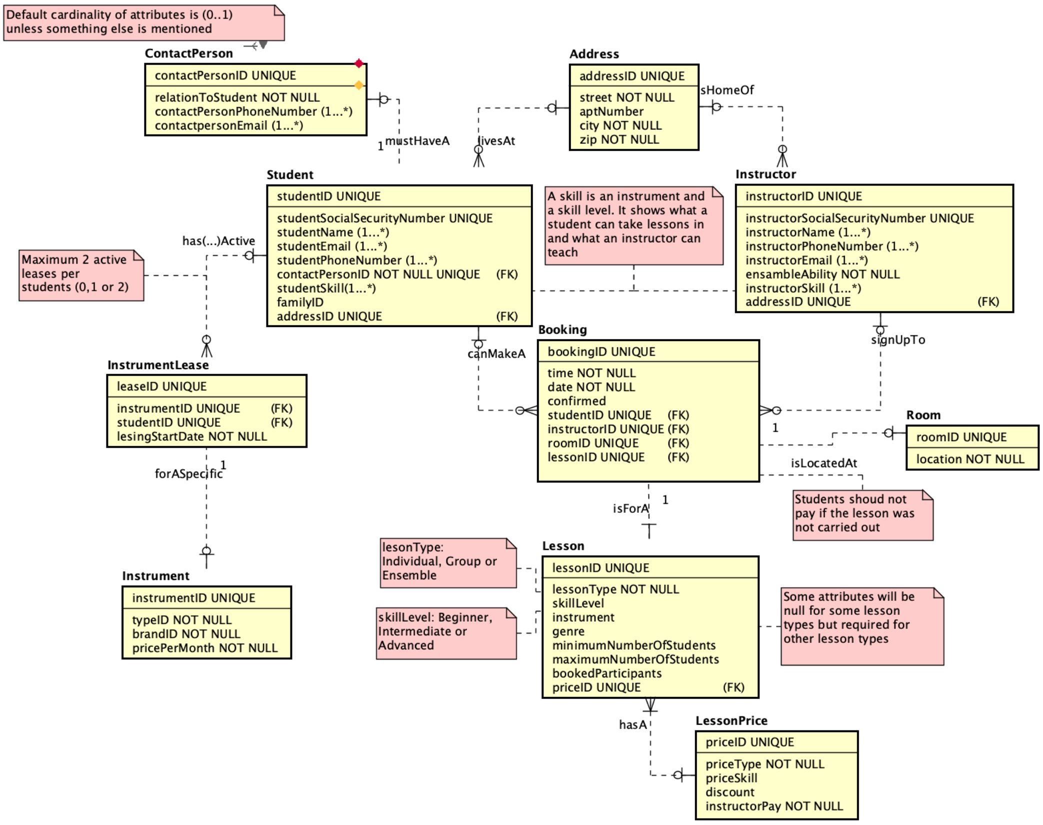

Figure 1: Our modeled ER diagram of the Soundgood music school without inheritance

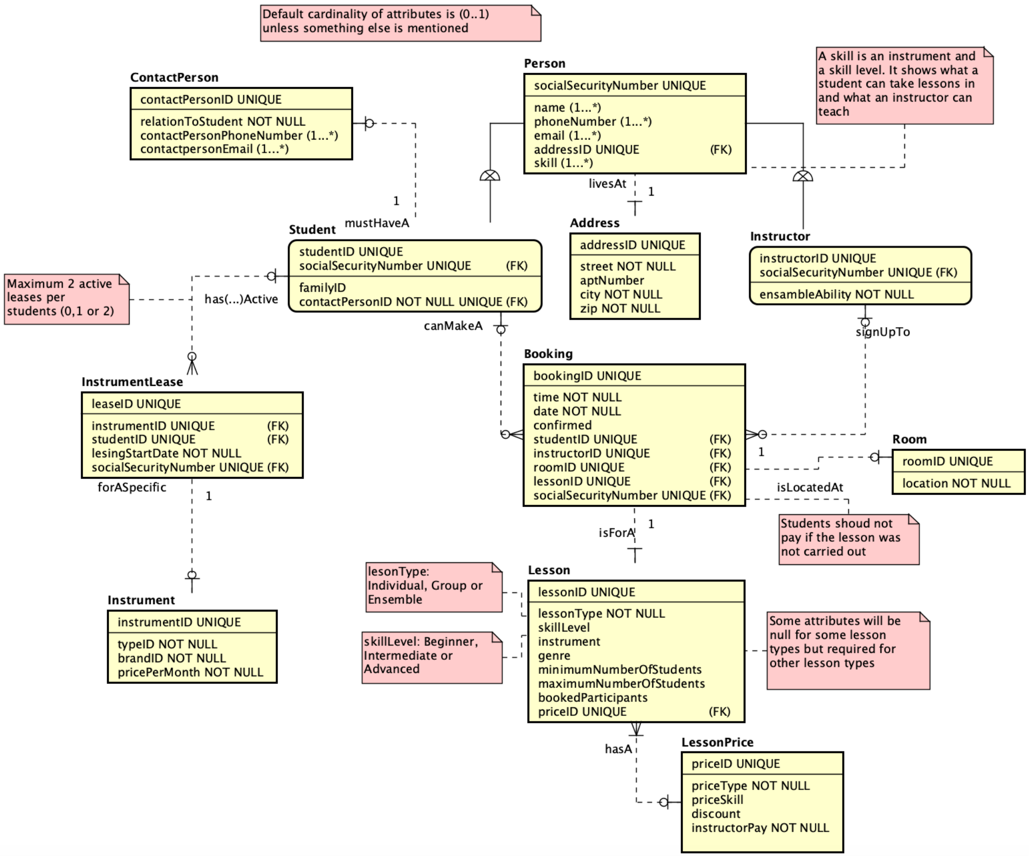

Figure 2: Our modeled ER diagram of the Soundgood music school with inheritance

Discussion

We found it very difficult to understand and really make the implementation work with attributes and the associations between entities. A lot of the images from both books came in hand when implementing the actual model and it took a good hand full of attempts followed by plenty of help from tutorial sessions before we got a model we where happy and confident to present. A lot of the attributes where removes and re-added to the diagram. Such as the booking entity, we removed it initially and drew the relation straight from a lesson, however we did not find that the structure got well represented and decided from a tutorial session to re add the booking entity and from there relate to lesson including a confirm attribute if an ensemble our group lesson would take place. Furthermore the associations between entities where redrawn plenty of times to fully understand their cardinality and the concept of relations.

Inheritance

Advantages

An example of inheritance was added to the conceptual model, displayed in Figure 2. Inheritance is a tool used in database design to base a new object of an already existing object. This helps maintain a structured diagram without repetitions of the same attributes in different entities. In our diagram without inheritance both instructor and student had their own attribute for social security number, phone, email etc. With the usage of inheritance we choose to add all shared attributes to a new person entity that could then be inherited by the instructor and student. With inheritance all shared attributes are kept in one table and identifiable attributes are added to the instructor and student entity separately, maintaining the diagram structured without to many repetitions. This is an advantage as the same attributes are now stored in the one table instead of different tables, for example emails for both student and instructor can be found in the person entity.

Disadvantages

A child entity in an inheritance relationship can only exist with the the parent class present. Since all attributes of the parent class are inherited by the child any wrongful updates on the parent entity will be inherited to all child entities related to that person, therefor one has to be careful when updating the database and be aware of how the relationships are modeled.

We evaluated if the contact person should also inherit the person entity as some attributes in this table are shared with the person entity. However this implementation was discarded as we found that two many attributes in the person entity weren't relevant for the contact person and would create null values in the table, if for example a contact person did not enter their email. This is a disadvantage as memory space would be wasted with null values attributes in the table.

Another candidate for inheritance that we evaluated when creating the diagram was the tree different lesson types individual, group and ensemble and to set up parent class lesson with shared attributes of all lesson types. Two of the lesson types shared many attributes being good candidates to set up an inheritance relationship however the third lesson type did, witch would alter duplicate values in the tables. For this implementation we decided to only create a lesson entity and allow for null values in the table and reasoned that this was more efficient than duplicate values.