Seminar 1 - Conceptual Model

Data Storage Paradigms, IV1351

Date

Project members:
[Adrian Boström, adrbos@kth.se]
[Viktor Sandström, visand@kth.se]
[Lucas Rimfrost, rimfrost@kth.se]

1 Introduction

Soundgood Music School is working to establish a comprehensive database in order to streamline operations. The increased need to better handle intricate business activities and information flow is the driving force behind this project. Strong data organization and validation frameworks are ensured by a ground-up approach that begins with conceptual modelling. The meticulous procedure used to build the conceptual model, which serves as the basis for improved operational decision-making, is described in the report that follows.

2 Literature Study

The development process started with a thorough examination of the fundamentals of conceptual modelling, comparing and contrasting domain and conceptual modelling methodologies. Conceptual models lay the groundwork for database architecture, whereas domain models act as blueprints for programming realities. A number of crucial ideas became apparent as the cornerstones of successful database design through a combination of scholarly resources, such as course materials, video tutorials, and pertinent literature. Important Results and Perspectives: Value of Conceptual Modelling in Strategy A well-designed conceptual model is a strategic tool that facilitates the creation of later reference models and greatly improves decision-making in the future. Throughout the course of the project, this foundation shows to be vital.

It was essential to comprehend the subtle differences between identifying and non-identifying relationships, which required multiple model adjustments as understanding grew. The significance of relationship precision in database architecture was brought to light by this iterative approach. Considerations for Cardinality The importance of entity and attribute cardinality mapping was demonstrated by the study. By establishing how different components interact and rely on one another, these linkages define the database's structural integrity. Important Database Architecture Concepts Although not shown directly in conceptual models, knowing primary and foreign keys was crucial for:

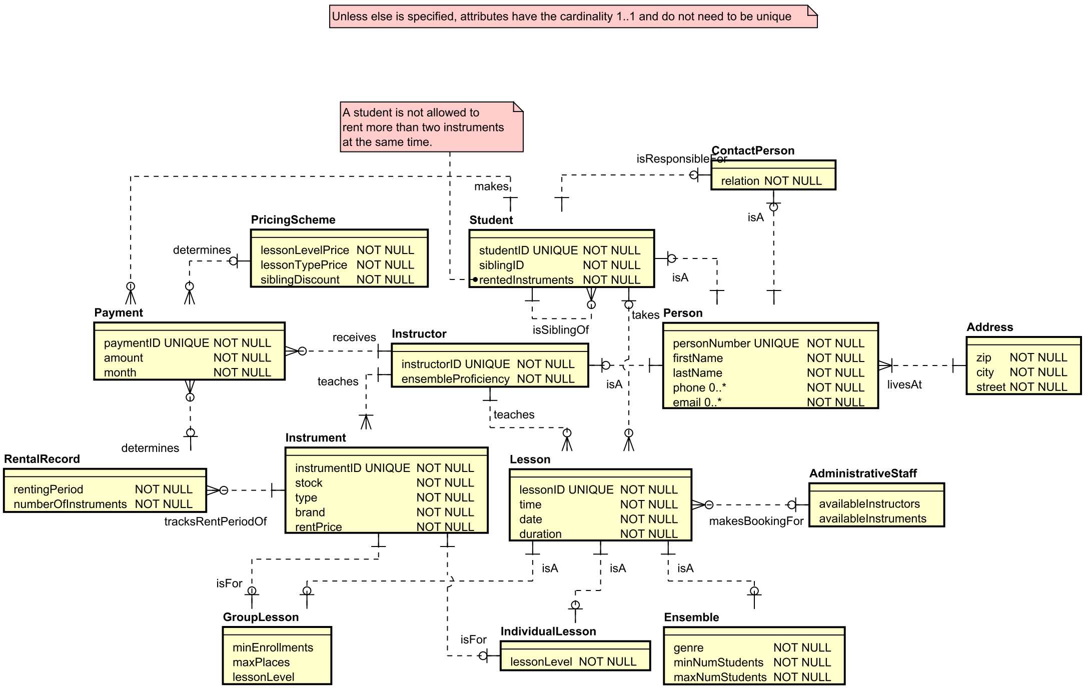
- Establishing proper entity relationships
- Defining data dependencies
- Structuring identifying relationships effectively

3 Method

The conceptual model's development followed a methodical process, meticulously incorporating Soundgood Music School's operational needs at each unique phase. In order to construct the basic database structure elements, initial efforts concentrated on the crucial task of entity and attribute definition, which required a detailed investigation of the school's operating framework. The most time-consuming part of development turned out to be relationship mapping, which required extensive study and numerous revisions. In order to handle complicated interactions, this method required careful consideration of entity interconnections, attribute placement, and the possible formation of new entities. For the best entity connectivity, extensive study of inheritance structures, identifying versus non-identifying relationships, and important implementations was necessary.

The process then moved on to a thorough analysis of attribute constraints, in which every component was carefully assessed to identify the necessary constraints and suitable cardinality requirements. Robust data integrity is ensured throughout the system by this careful attention to constraint implementation. The last methodological element focused on recording hierarchical relationships, especially those between parents and children. Clear structural definitions were produced during this documentation phase, resulting in a thorough framework for comprehending the system's dependencies and data flow.

4 Result



The resulting model is comprehensive and compact yet easy to follow. Overlapping concepts such as person and instrument where placed towards the centre with many attributes since many other entities where dependant on them such as instructor, person and contact person for the person entity. There where a great number of loops with entities connecting over and around other ones, this made it difficult to organize the diagram. Multiple versions where created and reorganized but the resulting connections stayed mostly the same.

An inheritance model was chosen for the Conceptual model where certain entities inherited for a "parent" entity. This can be seen for the Student and Instructor Entities, which inherited from Person, as both students and instructors are persons. The same was done for the different lessons that inherited from the Lesson entity.

5 Discussion

There where some points in the assignment that where rather subjective whether they were a part of the implementation. Such as the applications to the school, since it could be its own separate branch of the diagram or exactly what we input to this diagram, making parts of the diagram what said application would be. And the information will be registered into the diagram. There are also potential improvements with duplicate information and which class should hold that information. For example the person class holds information about the person, but there is also the address of which the person resides at. This could be a class called personal information but that would eliminate the person class as it would no longer hold any arguments.

To get the simplest information about the core classes there are few jumps to read (usually just one per connection). But since many depend on each other to get the complete picture, there needs to be more jumps that result in a cascading effect that requires almost the entire graph to be read. This however should not be a problem since this graph does not contain an abundance of entities, and there are no entries without attributes.