Project Report

Data Storage Paradigms, IV1351

Yuhui Gan

1st.Jan.2025

Declaration:

By submitting this assignment, it is hereby declared that all group members listed above have contributed to the solution. It is also declared that all project members fully understand all parts of the final solution and can explain it upon request.

It is furthermore declared that the solution below is a contribution by the project members only, and specifically that no part of the solution has been copied from any other source (except for lecture slides at course IV1351), no part of the solution has been provided by someone not listed as a project member above, and no part of the solution has been generated by a system.

1 Introduction

In this task, I created a conceptual model for the Soundgood Music School with an Entity-Relationship (ER) diagram to model its data requirements and support its operations. The school offers music lessons, both individual and group, for various instruments and skill levels. It also offers instrument rentals and sibling discounts. My goal was to design a comprehensive data model that captures all these aspects as described in the project documentation.

To begin, I identified the key entities, attributes, and relationships based on the school's operations. I then organized them into a clear ER diagram using standard notation. Where details were unclear, I explained them in plain text.

I worked independently throughout this task, ensuring the solution reflected my own understanding and effort. In the end of this report, I discussed about some design thought, learning outcomes and future thinking.

2 Literature Study

To prepare for this task, I began by watching lectures by Leif Lindbäck, which provide step-by-step guide on how to create conceptual models. Then I read the "Tips and

Tricks for Task 1" document which in my opinion is similar to the lecture. The lectures emphasized the following five steps:

- Identify potential entities by analyzing nouns in the requirements.
- Expand the list of entities using a predefined category list.
- Eliminate unnecessary entities, such as those without attributes.
- Define attributes, their cardinality, and whether they can hold null values.
- Establish associations and refine the model iteratively.

This method helped me focus on the data and relationships within the system as required for the conceptual model (CM). The lecture also introduced several categories to consider, such as transactions, roles, places, events, physical objects, and resources, ensuring a thorough analysis of the system's data requirements. There is also some additional knowledge I searched online which helped me a lot when it comes to adding notations to the diagram: Identifying and Non-Identifying Relationships An identifying relationship occurs when a child entity cannot exist without a parent entity. On the other hand, a non-identifying relationship exists when the child entity can exist independently of the parent.

Many-to-Many Relationships

A many-to-many relationship occurs when multiple records in one entity are associated with multiple records in another. **I**nheritance Relationships An inheritance relationship is used to represent hierarchies. This helps avoid redundancy and simplifies the model. **C**ardinality Cardinality specifies the number of relationships between entities: 1 or 0: Indicates that an entity may or may not have a related entity (optional relationship). 0 or more: Allows an entity to have no or multiple related entities. 1 or more: Requires an entity to have at least one related entity. Parent and Child Entities In a relationship, the parent entity is the one on which the child entity depends. The child's existence depends on the parent, while in non-identifying relationships, this dependency may not be strict. Through these resources, I gained a deeper understanding of how to create a conceptual model.

3 Method

As described earlier, I followed Leif Lindbäck's guidance to create the ER diagram. I began by thoroughly reviewing the instruction document and identifying all nouns that could potentially represent entities. These were added to Astah, the graphical tool I used for this task. I iteratively refined this list, repeating the process 2–3 times until I felt confident about the entities to include. Next, I added attributes to each entity. This step required careful consideration to ensure the attributes were logical and relevant for the entity's purpose. During this process, I eliminated unnecessary entities—such as duplicates or those better represented as attributes within another entity—to streamline the model. Once the attributes were defined, I reviewed the ER

diagram systematically, moving from left to right, to establish relationships and assign cardinalities based on logical connections and functionality between entities. I aimed for correctness and optimization in the notations. To finish up, I labelled attributes with constraints such as "UNIQUE" and "NOT NULL" where appropriate, ensuring clarity and adherence to the requirements. I also included explanatory notes in plain text within the diagram to clarify elements that could not be directly represented graphically. Throughout the process, I iteratively reviewed and refined the diagram to improve its accuracy and clarity.

4 Result

I worked independently to create the conceptual model for the Soundgood Music School, which is presented in the ER diagram. This diagram reflects the school's requirements for managing data related to lessons, rentals, billing, and administrative tasks. The model is designed using crow's foot notation and adheres to the course guidelines. **I**n the model: n this chapter, I explain the final version of my conceptual model for the Soundgood Music School, shown in Figure 1. I used Astah to create the diagram and focused on making sure that all the information described in the assignment could be represented clearly in the model. I tried to avoid unnecessary complexity while still making it complete.

The model is centered around the core activities of the school: students, instructors, lessons, instruments, and payments. I structured it in a way that reflects how these parts interact in real life.

Each student has personal and contact information, and they can optionally be connected to a guardian (for underage students). I also included a siblingDiscount entity which allows students with siblings to get discounts. Siblings are linked using the hasSibling relation.

One of the main improvements in this version was clearly linking instructors to lessons. I created a lesson entity that represents both individual and group activities, and connects to a specific student, instructor, and course. From lesson, it's possible to see who was involved, what kind of lesson it was, and when it happened. To handle different kinds of lessons, I used inheritance to create subtypes like individualLesson, nonIndividualLesson, and ensemble. This way, each type of lesson has only the attributes relevant to it. For example, nonIndividualLesson includes group size and instruments taught, which don't apply to individual lessons. This structure avoids having lots of empty fields and makes the data cleaner.

For instrument rentals, I made sure that instrumentLease is clearly connected to instrumentInventory, so it's easy to track which student rented which instrument and when. I also included availability checking, shown in a note, since that logic isn't always easy to model directly.

I created a priceList entity that holds pricing information based on course level, type, and instructor. This is linked to the invoice system, where studentInvoice collects all relevant charges, such as, lesson fees, instrument rental, and applies any available discounts.

There is also a lessonInvoiceBreakdown entity to explain how each lesson contributed to the final invoice. Instructors are paid through the instructorSalary entity, which uses lesson data to calculate how much each instructor should be paid. The connection is made through their participation in lessons.

Same with instrument rental, student need to check the inventory first and see if their desired instrument is in stock or not then make a request. The administrative staff will decide if the student is granted a lease based on whether they filled both rental instrument attributes in the student entity. All the personal info is stored in the contact list entity as everyone can decide to be a student and learn something new. All the "IDs" are "UNIQUE AND NOT NULL" as they should be easily identified to reduce confusion.

In addition, I put the relation between address and contact list as Inheritance, as I believe the address is a subset under the contact list.

Throughout the process, I refined the model iteratively, ensuring proper relationships, cardinality, and constraints. This ER diagram provides a solid foundation for the database, capturing all required information while maintaining simplicity and usability.

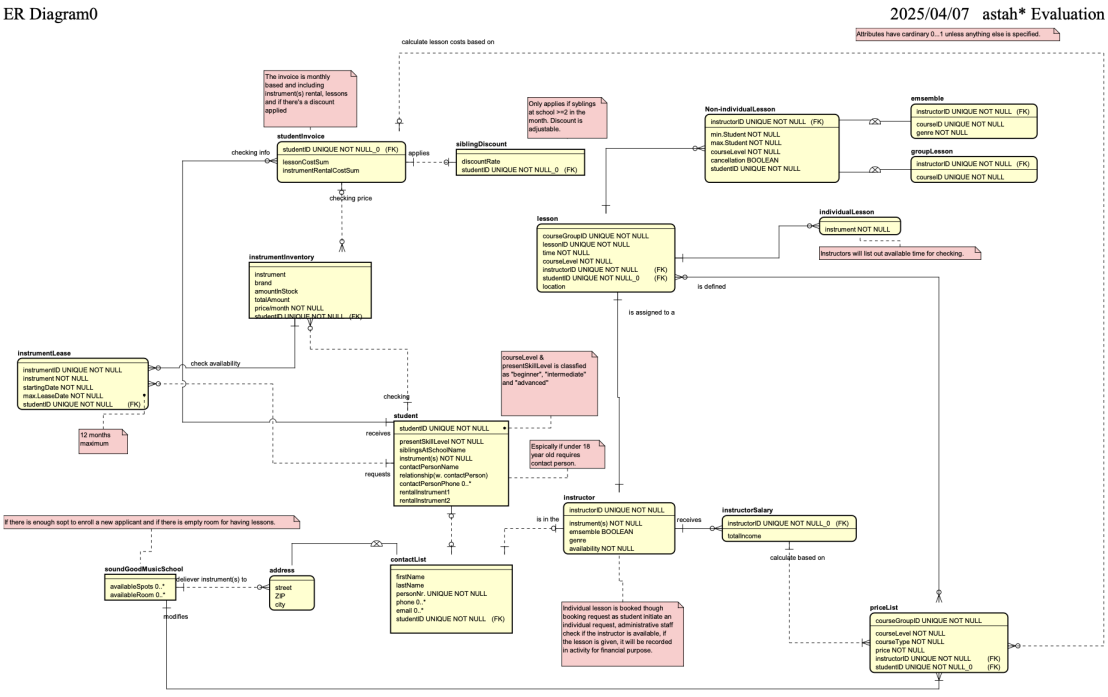

Figure 1: Conceptual Model for Soundgood Music School

Link to Github: `https://github.com/GerminalG/IV1351.git`

5 Discussion

I believe the conceptual model I designed meets the requirements outlined for the Soundgood Music School. It captures all the necessary data for managing students, lessons, rentals, and billing. The relationships between entities are logical and efficient, ensuring data can be accessed with minimal effort. I included all key entities without redundancy, and any unnecessary or duplicate entities were removed during the design process. Each entity has meaningful attributes with appropriate constraints, such as marking IDs as UNIQUE and NOT NULL, to maintain data integrity and clarity.

The relationships in the model are clearly defined with proper cardinalities, ensuring they accurately represent the interactions between different parts of the system. For example, billing entities are connected to pricing and activity records to facilitate instructor salary calculations and student invoices. I used descriptive naming conventions throughout to make the model intuitive and easy to understand. Additionally, crow's foot notation was applied consistently, ensuring the diagram was professional and visually clear. Some business rules and constraints, like sibling discounts and the two-instrument rental limit, were included in the description where they could not be represented directly in the diagram. I documented my methodology and results clearly in the report, and I refined the model iteratively to ensure it aligned with Soundgood's requirements. Overall, I am confident the model is robust, comprehensive, and effectively supports the school's operations.

While working on this model, I learned that lines between entities don't just mean "relationship" in a general sense, they only exist when one entity uses data from another. If two entities are related in real life, but don't share data directly, they don't necessarily need a line in the conceptual model. This helped me avoid over-complicating the diagram.

6 Comments About the Course

Getting the software working could be tedious which took me hours of effort, I spent 2 hours trying to skip the free trial but failed due to it never asked if I wanted a free trial or not, and it didn't allow me to place the license document, so I ended up with using the free version.