

Project Report - Task 1

Data Storage Paradigms, IV1351

Reza Hosseini rezahos@kth.se

November 2022

1 Introduction

The purpose of this task and this project report is to facilitate information handling and business transactions for the Soundgood music school company, by designing a model that meets all the requirements. The requirements of the project are described below:

A student can take individual lessons, group lessons or join ensembles. An individual lesson is taught by an instructor that is available at the time that suits the student. A group lesson has scheduled time slots and it needs to have a minimum number of students and it also have a limit on how many students can enroll. Both individual and group lessons have specified instruments and level skills where level skill is one of the following: "beginner", "intermediate" or "advanced". An ensemble is a mixture of different instruments and it has a specified genre.

A student, in order to be able to attend, needs to give person number, name, contact detail and address. It can also give contact detail of a contact person. If the student has a sibling who also is taking lessons, they can give the information of that too so they get discount. A student can also hire maximum of two instruments for up to 12 months.

An Instructor also needs to give person number, name, address and contact detail as well as if he/she can teach ensemble, which instruments he/she can and at when can be available for individual lessons.

A student pays monthly and the amount depends on which lessons the student has taken, what skill level and how many lessons they are. The instruments that have been hired are also charged monthly. Similarly, an instructor is also paid monthly, based on number of lessons, the skill level type of lessons.

The model we use is a conceptual model that is based on reality and covers all the necessary data that the Soundgood music school company has in the real world. It shows all the required entities, their corresponding attributes and the relationships between them. Since the model mainly focuses on data, we only have entities that store data in some way and we also only have the relationships between the different data.

This task was done entirely by myself but I had multiple discussions with some of my classmates to get inspirations and see how you generally solve this kind of task.

2 Literature Study

In order to succeed with the task I first went through the YouTube videos from Leif Lindbäck about how to implement a conceptual model properly; The videos taught me about what a model is, how to do modelling and specifically how to design a conceptual model. It also explained how to use IE notations, create diagrams and add entities or attributes:

- Conceptual Model, Part1 (Youtube video by Leif Lindbäck)
- Conceptual Model, Part2 (Youtube video by Leif Lindbäck)

In addition, when designing the conceptual model, I followed the steps that were recommended in the YouTube videos which were originally from the book below:

- Object Oriented Development - A Hands-On Approach (Leif Lindbäck April 13, 2022)

The step-by-step method taught me how to think when trying to design a conceptual model and it helped me to spot some wrongs implementations I had. A part from that, I took advantage of both **Tips and Tricks** and **Assessment Criteria** documents to update and fix the model I had designed.

3 Method

I used Astah as the diagram editor to do the modelling; In order to learn the software, I watched Leif Lindbcäks introduction videos which did use the Astah software, but I also watched some other YouTube tutorials to make myself familiar with Astah.

To get started, I firstly read the project detail very carefully and made sure I understand how the whole system is supposed to work. After understanding the project and also watching the previously mentioned videos about conceptual models, I followed the 4-step procedure recommended by Leif Lindbäcks to build the model, which are the following steps:

3.1 Noun Identification

The first step to design the conceptual model was to find all the nouns from the project description that could either be entities or attributes. At this stage, I did not pay so much attention in if the noun should be an entity or an attribute and I just wrote them down on a paper:

Lesson	IndividualLesson	GroupLesson
minimumEnrolledStudents	lessonPlace	placeMaxSize
Ensemble	Genre	maximumEnrolledStudents
lessonPlaceWaitlist	scheduledTime	appointment
instrument	brand	quantityInStock
student	contactDetail	personNumber
Admin	Instructor	availability
name	address	hasSibling
studentPayment	receipt	instructorPayment
period	rentedInstruments	fee
enrolledLessons	skill	beginner
intermediate	advanced	contactPerson
instrumentsThatCanTeach	lessonsGiven	RentingInstruments

Table 1: Nouns taken from project description

3.2 Category List

The next step was to follow the list below to ensure that I have not missed any important nouns:

- Transactions, selling or buying a product or service
- Products or services, what is sold or bought in the transaction
- Roles of peoples and organizations involved in the transaction

- Places, maybe where a transaction is performed
- Records of a transaction, for example contract, receipt
- Events, often with a time and place
- Physical objects
- Devices, are probably physical objects
- Descriptions of things
- Catalogs, where the descriptions are stored
- Systems, software or hardware that is collaborating with the system for which we are creating the DM
- Quantities and units, for example length, meter, currency, fee
- Resources, for example time, information, work force

3.3 Remove Unnecessary Entities

After adding all the nouns, it was time to clear them and remove the unnecessary ones from the list.

3.4 Find Attributes

Finally the last step was to distinguish attributes from entities and create those entities with their corresponding attributes in the Astah software.

When I was comfortable with the entities and attributes, I started to draw relationships between them and add cardinality to both relationships and all the attributes.

As mentioned earlier, I also used the **Tips and Tricks** and **Assessment Criteria** documents to ensure that my solution is valid and meets the requirements.

4 Result

The final result is shown in figure 1:

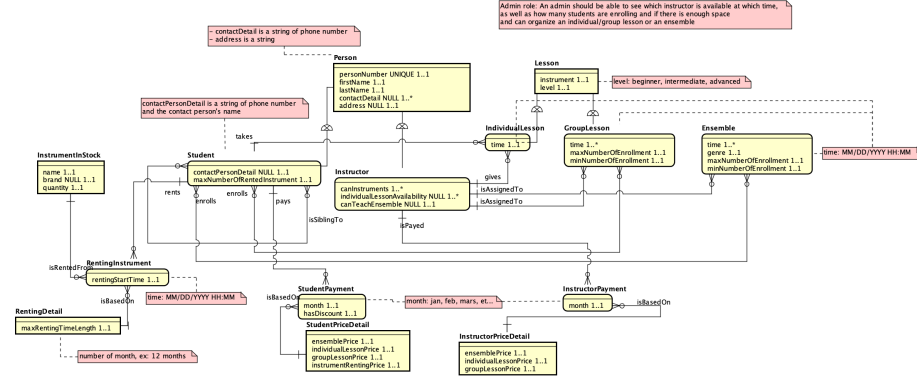


Figure 1: The Conceptual Model

The model consists of multiple entities, attributes and relations. Every entity is explained below:

Person:

A person has a unique person number, first name, last name, contact detail and address. Contact detail here is just a phone number in a string and address is also a string.

Lesson:

A lesson is specified in a certain instrument and has a specified skill level. The instrument is a string and the skill level is one of the following: "beginner", "intermediate", "advanced".

IndividualLesson:

An individual lesson extends a lesson (so it basically is a lesson) and has a specified time that is chosen by a student and is approved by the admin (which means an instructor is available and can give that lesson).

GroupLesson:

A group lesson extends a lesson and has several scheduled time slots (chosen by the admin). A group lesson has to have a minimum number of enrolled students, otherwise it gets cancelled or gets scheduled again by admin. And a group lesson can also have a maximum number of students.

Ensemble:

An ensemble has similar attributes to group lesson and also works similarly in the model except that an ensemble has a specified genre as well.

Student:

A student can give details of a contact person which is just a phone number. A student can take individual lessons which the time of each lesson is chosen by the student and approved by the admin (which checks if any instructor is available at that time for the approval). Several students can enroll group lessons or ensembles which the time of them are scheduled by admin. A student can also rent instruments that are available in the stock. There is also a limit on how many instruments a student can rent at a time.

A student does pay monthly student payments based on current price for each lesson and rented instrument.

Instructor:

An instructor needs to be able to teach one or more instruments and can also show if he/she is available for giving individual lessons (if yes, then at which times). An instructor can also teach ensemble if possible.

An instructor can give (if any) individual lessons and also teach group lessons and ensembles. An instructor gets paid monthly based on current price for each given lesson.

InstrumentInStock:

There are different instruments in the stock of the company that can be rented by students which are differentiated by name and brand. There is also a noun number of each instrument with a name and a brand so we can see if a specific instrument is available for renting or not. It is also possible to see all available instruments and their quantities in case a student wants to rent one.

RentingDetail:

An instrument can only be rented for a limited time period which is given by number of months (ex: 12). The renting detail is predefined by the admin and its value gets used when a student rents an instrument.

RentingInstrument:

A renting instrument is rented by a student at a specific time. It comes from the instruments in the stock and it knows the maximum renting time for an instrument.

InstructorPriceDetail:

An instructor price detail shows all the pricing regarding to an instructor, that is: individual lesson pricing, group lesson pricing (which also varies depending on skill level of the group lesson) and ensemble pricing.

InstructorPayment:

An instructor payment shows how much an instructor in a specific month should be payed and the pricing information is taken from the instructor price detail table.

StudentPriceDetail:

A student price detail shows all the pricing regarding to a student, that is: individual lesson pricing, group lesson pricing (which also varies depending on skill level of the group lesson) and ensemble pricing as well as price for renting instrument.

StudentPayment:

A student payment shows how much a student in a specific month should pay and the pricing information is taken from the student price detail table.

5 Discussion

After completing the task, there were a few things that some discussions could be done about:

Number of entities and relations:

The conceptual model I came up with does have less entities compared to other solutions I happen to see, and it also has more relations. However, the way it is constructed works just fine and I think it is just another way of designing the model.

Although this implementation works just fine, I could also try to lower the amount of relations and instead have more entities. An example of that is having a dedicated entity for sibling where two students are related to each other through that sibling entity.

Time and Contact Details:

In my model, I tried to hold it simplified and not have extra data that is not necessary. For instance the contact detail of a person and the contact person detail of a student is simply a phone number in a string format. The reason is that more data is not given about the specifics of a detail, we either should contact the customer and ask for more information or just go with something that makes sense and is also possible to update in the future.

In this case, we can possibly create an extra entity in which the contact detail is specifically set.

The same goes with the attribute time that has been used multiple times in the model. There are different types of time that are used for example Ensemble and Lesson have a specific time format that you can see in figure 1. While `rentingStartTime` from `RentingInstrument` only contains the date and does not include the time. This unclarity is due to not having enough information and also not having a standardised time for the whole model. Here we can also, in the future when we get more information, create an extra entity in which the time detail is specifically set.

Inheritance:

In this model, I decided to have an entity called `Person` where `Student` and `Instructor` inherit from that. The reason is that both students and instructors are humans and they share a lot of data in our model, so it is a good idea to have a `Person` entity which has the shared detail of both `Student` and `Instructor`. The same idea and reasoning applies to the entity `Lesson` which `IndividualLesson` and `GroupLesson` inherit.

Having or not having inheritance in the model is optional and you can also model without inheritance. In that case you just move down all the attributes from the parent entity to its children which will cause duplication, meaning the children entities will have many similar attributes.

In the case of Person and Student/Instructor we could have instead something like in figure 2:

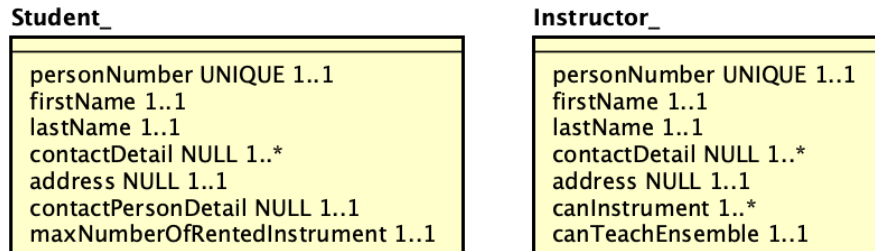


Figure 2: Student and Instructor entity without the parent entity Person

Method procedure:

The method I followed to do the modelling was quite powerful, however I did not found the step 2 of the method (where I needed to go through a category list to see if I can find new nouns) quite useful as I already had all the nouns included. It is always harmless to go through the list since you never know if you have missed anything but sometimes you might have everything ready and nothing new is found. The important thing is to always carefully go through the information and double/triple check to see if you have missed anything.

Administrative:

There is an Admin role in this model which does not have an own entity. An admin books all the lessons (individual and group lessons)/ensembles, makes sure that there are enough students enrolled in lessons, there are instructors that are available, schedules new group lessons/ensembles based on the demand and updates the payment pricing for the month if needed.