

Conceptual model

Seminar 1, IV1351

Saina Shamshirdar

2023-11-07

1 Introduction

The first task in this project is to create a conceptual model of the Soundgood music school that illustrates information handling and business transactions. A specific procedure must be followed when creating the conceptual model. The Soundgood music school offers lessons to students that can be ensemble, individual or group lessons. A lesson is given based on the instrument and the skill level. Contact details for a student must be recorded, as well as the instrument they want to play and their current skill level. Students can take as many lessons for as many number of instruments as they want. An instructor can be assigned to teach different types of lessons and specified number of instruments. The contact details for the instructors must also be stores. Students do a payment monthly for the taken lessons during the recent month. There is a discount for siblings that take lessons during the same month. Instructor salaries are payed monthly depending on the number of lessons they have had, the level and the type of the lesson. The students have the opportunity to rent up to two instruments for a period of 12 months and the fee is payed the same way as the lessons are. *In this task, I have worked with Ruth Jenbere Shewa and Lassya Desu.*

2 Literature Study

First I learned what a domain model was which was helpful to understand the principals of a conceptual model. A domain model is a model of the reality used when programming and a conceptual model is a model of the reality used when creating a database. A client comes with a requirement which can vary depending on the project or the client. From this requirement description, all the subjects are extracted. A category list is followed to make sure all the relevant and necessary subjects are included. Then it is time to eliminate the unnecessary subjects. After that attributes are added to all of this subjects that are now classes. Some classes can turn into attributes. Finally, association should be drawn between the classes and be given a name that explains the relation between those two classes.

3 Method

The diagram editor that was used in this task was Astah. To start off with the task, all the possible nouns were written down as classes in an ER diagram in Astah. Using a category list, it was checked whether all the relevant nouns were found. Based on the category list, transactions, products or services, roles, places, records of transactions, events, physical objects, devices, descriptions, catalogs, Systems, quantities and units and resources are the nouns to look for. For example a transaction means selling or buying a product or service. In this case, a transaction is a payment and the product or service that is bought is a lesson. A physical object is an instrument. Systems such as software or hardware that is collaborating with the system for which we are creating the CM should not be included in our conceptual model. Therefore, nouns such as computer is not included in the model. The next step in building the model is to remove the unnecessary nouns. The written nouns were reviewed and the nouns that couldn't have an attribute or wasn't needed were removed. After that, the search for attributes to be put in each and every class was started. For this purpose, the requirements were checked over and over again. Next step was to find the right associations between the classes. Some points to be avoided when drawing the associations were creating a spider meaning a class having many associations to other classes making the association number of every class unbalanced through the entire model. It is recommended to keep the number of associations to each class balanced in the model. The cardinality of all the attributes were determined where some had NOT NULL, some UNIQUE and the others 0...*. Also the cardinality of the associations were decided. In the end, everything was reviewed to make sure that the nouns, attributes and associations make sense according to the clients requirements.

4 Result

The existing nouns after eliminating and removing unnecessary nouns are person, student, instructor, lesson, booking, pricingScheme, contactDetails, contactPerson, address, rental and payment. In the presented model in figure one, there is a "person" class that is associated to "student" and "instructor" meaning a person can either be a student or an instructor. A person has a personID which is unique. The student has specified number of instruments he/she wants to play and a skill level. The instructor has a specified number of instruments he/she can teach.

There is an association from the student to itself which indicates if there is a sibling. Since the student has a student ID coming from the person entity and the association is many-to-many it is possible to determine if the student has any siblings. If so, according to the comment box, they both get a discount if they take lessons in the same month. Since, every person has contact details and the students have even a contact person two classes were made for this purpose. The "person" is then associated with contact details. The contact details have even an association with an address entity. The address entity was an attribute in the contact details entity in the first place. Since, address can have

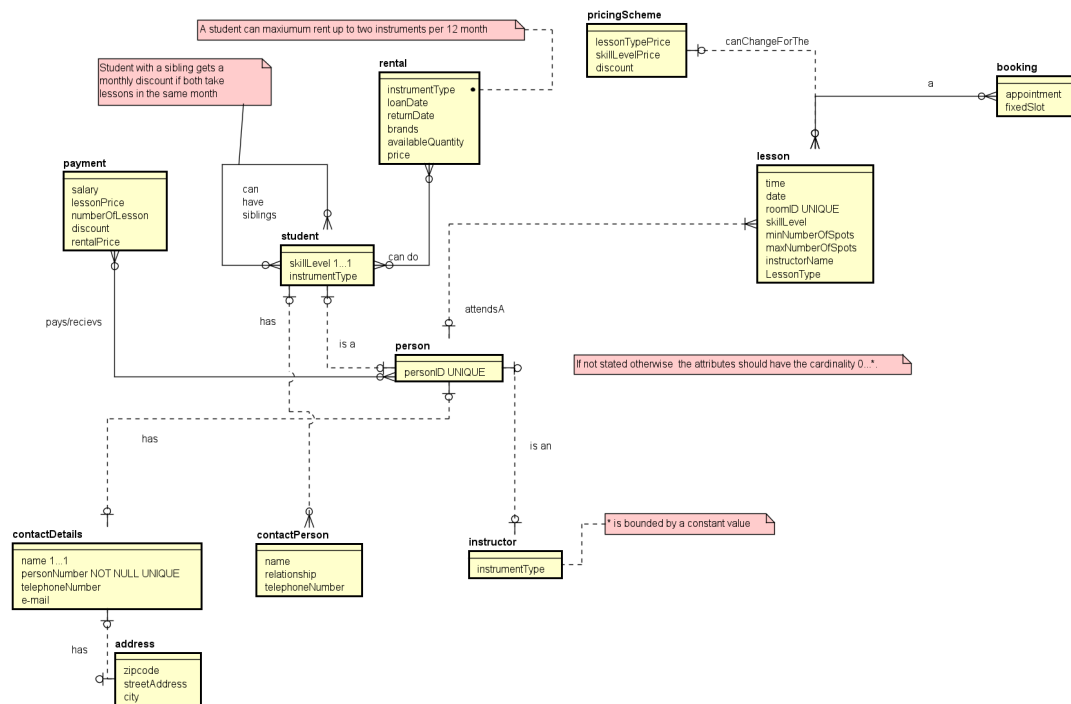


Figure 1: Conceptual model

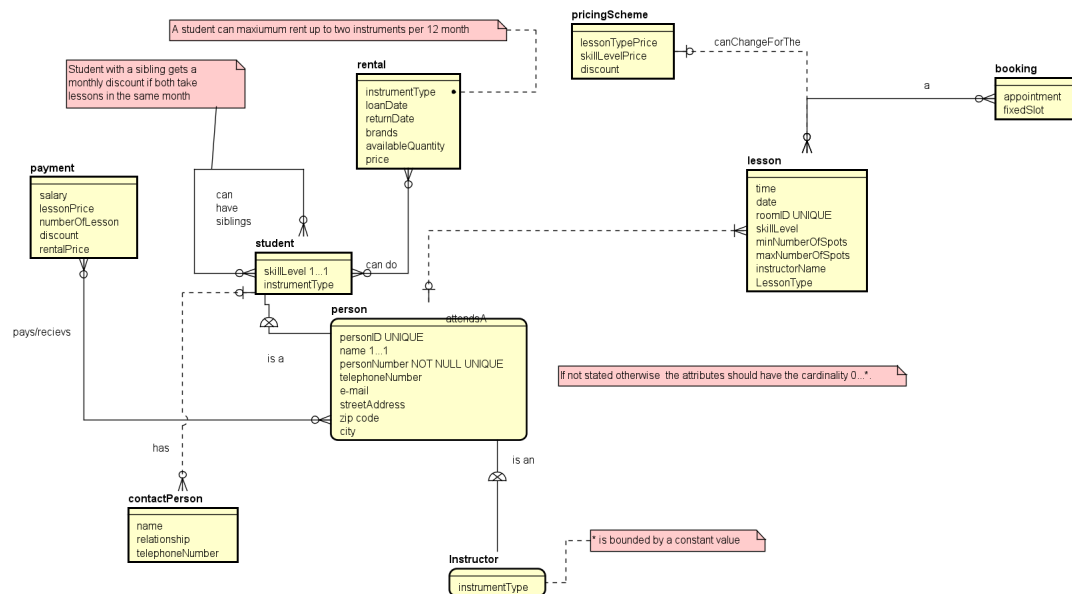


Figure 2: Conceptual model. inheritance version

attributes itself, it was taken out to be its own entity with new attributes. The student entity is associated with the contact person entity.

The students have to pay for their lessons and rentals, as well as the instructors need to get paid. A class called "payment" was created holding the number of lessons taken, the lesson price, the rental price and a salary in case it is an instructor. The "person" has an association to this class. And it is a many-to-many association since students can pay multiple payments and instructors can receive multiple salaries. Students can rent instruments, so an association was drawn from the student to the "rental" class which holds some attributes relevant for this matter. To book a lesson, there are two choices of time slots, either its and appointment or its a fixed slot which is determined by the lesson and the lesson type. Therefore, an association is drawn from "booking" to "lesson". The prices for the lessons can change, and this change is determined by the "pricingScheme" class.

Another version of the diagram with inheritance would be to put the contact details attributes in the person entity. This way, the student and instructor entities can inherit the attributes from person, since both of them need all those attributes. The inheritance associations are shown in figure 2. The student is a person, so the is-a criteria is met for both of them. The super entity which is person is generalized and the sub-entities are specializations of the super entity. Everything in the super entity is relevant in the sub-entities.

5 Discussion

All the entities have attributes, the important classes are included and it is easy to collect data about all of the major entities. The crow foot notation is followed where the cardinality of the associations are indicated. For example, the student association to itself is many-to-many which is shown by the crow foot notations at both ends meaning different students can have different number of siblings. The cardinality of the attributes are stated where attributes like person number and different ID:s are NOT NULL and UNIQUE while the rest of the attributes have the cardinality zero to many if anything else is not mentioned.

The advantage of having the inheritance is that one less entity is used which makes the model more simple and easy to understand. The advantage of not having the inheritance is that it is easier to keep track of the attributes. An entity with many attributes might seem complicated and make the model confusing. So it is better that the attributes are distributed in a smart way.