

Lecture 7 — Modelling Diagrams

Jeff Zarnett

Modelling Diagrams

We have talked at some length about the tools for defining the database, but we really should spend significant time on good design. Knowing how hammers work is not enough; we also need to understand what good architectural plans look like.

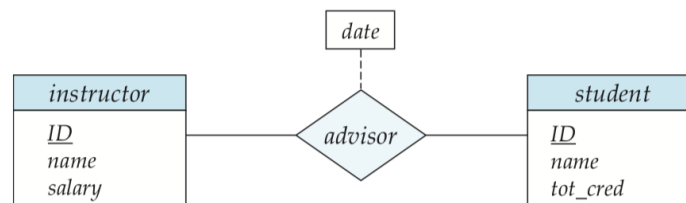
The first thing we would like to watch out for is redundancy in the database and eliminate it wherever possible. In short, if there is redundant data then there is the potential for data to get out of date or be in some other inconsistent state.

When we have some data that we would like to represent we need to turn it into some database tables. Going directly from some ideas about what data we need to the tables is sometimes difficult and may produce undesirable results. Instead, we should devise a plan for how it should be implemented and for that we will create diagrams. Diagrams are done on paper or a whiteboard or similar and it is very easy to visualize, understand, and change the design at that stage.

We are going to use Entity-Relationship diagrams (E-R diagrams) as our method for modelling the entities in the database as well as their relations. Entity-relationship diagrams have the following symbols and their meanings [SKS11]:

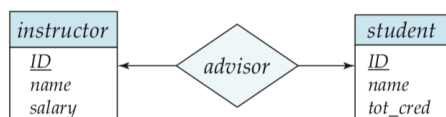
- **Divided Rectangles** represent an entity; an entity has a title (the top part) and a list of the attributes of that entity (the bottom part).
- **Diamonds** represent a relationship set.
- **Undivided Rectangles** represent the attributes of a relationship set.
- **Lines** link entity sets to relationship sets.
- **Dashed Lines** linked the attributes of a relationship set to its relationship set.
- **Double lines** indicate total participation of an entity in a relationship set.
- **Double Diamonds** represent identifying relationship sets linked to weak entity sets.

We'll come back to those last two items later. Note also that in rectangles, attributes that are or form part of the primary key are underlined. Consider an example below:

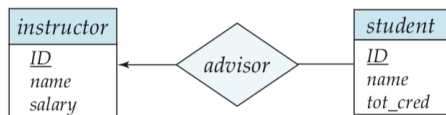


E-R diagram showing relationship between instructors and students, with attributes [SKS11].

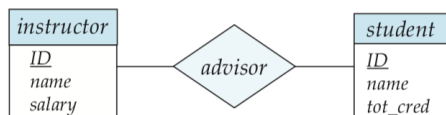
In this case we have a relationship between instructor and student called “advisor” and that relationship has an associated date. What sort of relationship is shown here? This is a many-to-many relationship. To decide as to whether that makes sense, let’s think about real life: a professor can advise multiple graduate students, and a graduate student can have more than one advisor (for example, two of them). The diagrams below show the other possibilities:



(a)



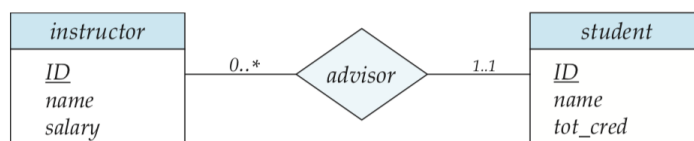
(b)



(c)

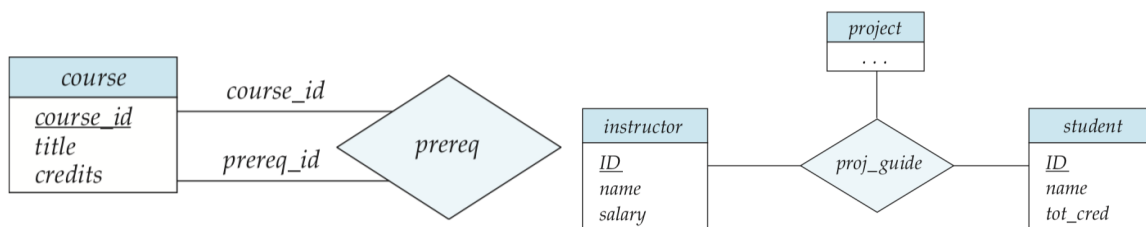
E-R diagram with (a) one-to-one, (b) one-to-many, and (c) many-to-many relationships [SKS11].

If we want to show some more specific rules that allow us to be more specific than just one vs. many, we can apply some numbers to the lines. In the diagram below, an instructor can have zero or more students and a student can have exactly one advisor.



E-R diagram showing cardinality limits [SKS11].

There is no reason why a relationship must be binary. A table may reference itself, as in the case of courses that have prerequisites. If that is so, then assigning names to the relations help to differentiate. Or a relation may involve many relations as in a ternary relationship:



Non-binary E-R diagrams; unary and ternary respectively [SKS11].

Weak Entity Sets. Weak entity sets are not entities that do not even lift. Instead, they are entities that are dependent on another entity, namely, one that cannot be identified uniquely based on its own attributes. The

example that builds on this “university” model is a section: a section of a course, such as 001, is dependent on a course in a particular term.

Suppose that a course has an id, name, and code. A section has a section number, year, and term. A particular course, ECE 356, will have one or more sections, e.g. 001. But the attributes of the section on their own (001, 2018, Winter) are not enough to uniquely identify a single course. Because at least one other course, such as ECE 459, also has a section that matches (001, 2018, Winter). To differentiate between two sections that match on those same criteria we need a course number, but the course number isn’t a part of the section entity. That is part of the course entity. And this is what makes the section entity a weak entity.

More formally, an entity that on its own does not possess sufficient attributes to form a primary key is a weak entity; one that is not weak is a strong entity. A weak entity set is associated with another entity, called its identifying set or its owner entity set [SKS11].

In a practical sense the weak entity will probably have some attribute added to it that identifies which strong entity the weak entity belongs to. That is called the *discriminator*. That would mean some attribute added to the section entity that references a particular course. Still, that is an implementation detail that plays no role in the E-R diagram, which looks like the diagram below:



E-R diagram with weak entity set [SKS11].

It is worth noting that the weak entity shown in the diagram has some slight differences versus a strong entity. The attributes are underlined with a dashed line and the relationship is modelled with a double diamond. It also uses double lines between section and sec_course that indicates total participation: every section MUST be associated with a single course.

We could choose a different option, of course, and add some sort of unique identifier to the section that would promote the weak entity to a strong one. But that doesn’t necessarily make logical sense in the context of the application; a section doesn’t make sense on its own and giving it a unique identifier does not really correspond with reality: a section is logically dependent on the course, isn’t it?

Weak entity sets can exist in other configurations than just the identifying relationship. A weak entity can be the owner of another weak entity, or a weak entity can belong to more than one identifying set.

References

[SKS11] Abraham Silberschatz, Henry F. Korth, and S. Sudarshan. *Database System Concepts, 6th Edition*. McGraw Hill, 2011.