Exercise 4-1

Figure A-12 shows a first draft of modeling the situation where a publishing company wants to keep information about authors and books. Consider the possible optionalities at each end of the relationships `writes`, and so determine some possible definitions for a book and an author.

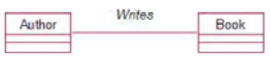


Figure A-12. *Consider possible optionalities for authors writing books*

At first we might think that an author will always have at least one book he has written and a book will always have at least one author (even if we might not know who it is). This may be true for *actual* books and authors, but here we are concerned with *information* about books and authors. A publishing company might often see an opportunity for a book on a particular topic and record that information while they search for an author. Similarly, a publisher might retain a potential author and store information even though no books have yet been written for the publisher by that person.

Possible definitions might include: *a book is a work that has been written or is planned to be written; an author is a person who has or might in the future write a book.*

Exercise 4-2

Figure A-13 shows a possible data model for cocktail recipes. What is missing?

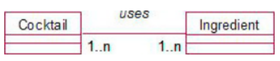
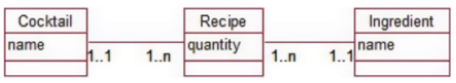
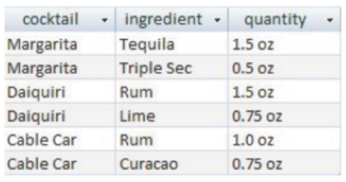


Figure A-13. *Cocktails and their ingredients; what is missing?*

Each cocktail may have a number of ingredients (Manhattan: Vermouth, Whisky; Margarita: Tequila, Triple Sec, Lime). What are missing are the quantities. As is often the case with a Many–Many relationship, an intermediate class is required. The quantities depend on a particular pairing of `Cocktail` and `Ingredient`. A better model is shown in Figure A-14, along with some possible data. The inclusion of the `Recipe` class allows us to keep information such as how much Rum is required for a Daiquiri as opposed to a Cable Car.





cocktail	ingredient	quantity
Margarita	Tequila	1.5 oz
Margarita	Triple Sec	0.5 oz
Daiquiri	Rum	1.5 oz
Daiquiri	Lime	0.75 oz
Cable Car	Rum	1.0 oz
Cable Car	Curacao	0.75 oz

Recipe Table

igure A-14. *An intermediate class, Recipe, can record quantities for each pairing of cocktail and ingredient.*

Exercise 4-3

Part of the data model about guests at a hostel is shown in Figure A-15. How could the model be amended to keep historical information about room occupancy?

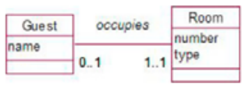


Figure A-15. *How could this be amended to keep historical information about room occupancy?*

The data model indicates that, for a hostel with single occupancy rooms, a room might be empty or have at most one occupant. Each current guest occupies one room. Over time, however, a room will have many different guests, and guests may return and occupy different rooms. This needs to be modeled as a Many–Many relationship as in Figure A-16. (As an aside, you can deduce from the optionality of 1 for a guest being associated with a room, that our definition of a guest is someone who has been assigned a room at some stage—not just any person who might or might not come to the hostel.)

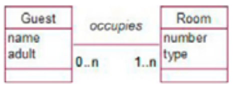


Figure A-16. *Guests and rooms modeled with a Many–Many relationship*

Now that we have a Many–Many relationship we need to ask the question: is anything missing? Clearly what is missing is information about when a particular guest occupied a particular room. This requires an intermediate class as in Figure A-17.

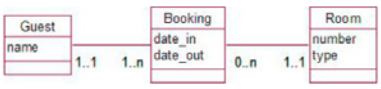


Figure A-17. *Including a Booking class to keep information about the dates that guests occupy rooms*

Each guest can have several bookings over time, as can a room. Each booking is for one guest in a particular room. A word of caution here though—our original data model (Figure A-16) indicated that a room could only have a single guest. Now that we have allowed many guests in a room over time, we have lost the information that at any one time a room can have only one guest. Our model in Figure A-17 would not prevent several people all having a simultaneous booking for one room. These sorts of problems are never simple! One way to record a business rule about simultaneous bookings would be to describe it in the use case for adding a booking for a room. It could say something such as: *no booking can be added to a room where an existing booking has overlapping dates*. A data model gives us a huge amount of insight, but on its own it is not a complete description of a problem.