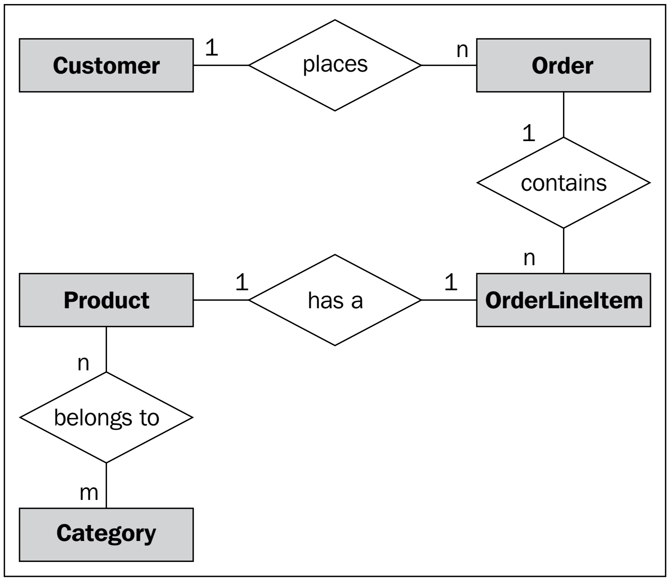
Once we have created the first drafts of both the ER diagram and the domain model, we evolve them in parallel by making changes in one and reflecting these updates in the other. It is not common to completely finalize an ER diagram and only then work on a domain model.

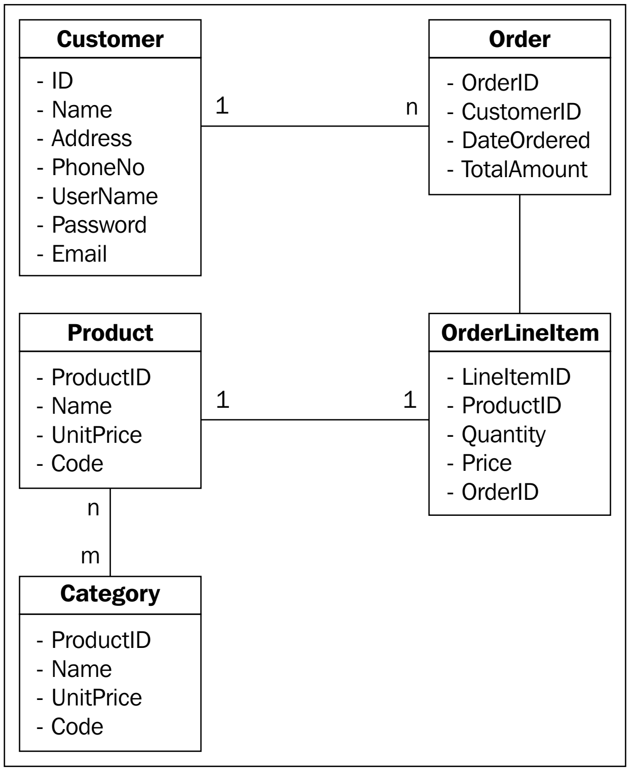
An important point to be noted here is that an object model entity is different from a data model entity. For example, both `Employee` and `Manager` are different entities in the Object Model, but they can be persisted in the same physical data table. Hence the data model entity would be the same for both the `Employee` and the `Manager`. A logical data model is independent of the database. So the same logical model can be used to design a physical model for Oracle, or MS SQL Server, or any other database.

An Example of a Logical Model

Let us revisit the ER diagram that we created for our Order Management System in Chapter 3.



Although we have not shown any attributes of the entities here, an ER diagram can have attributes for each entity listed in the diagram. But I usually refer to the ER diagram for understanding entity relationships from a high level (without getting into the details of the attributes involved), and use the logical data model for a more complete picture of the database. The ER diagram here is basically a foundation for the logical data model for our database. Using an ER diagram, we can quickly see how different entities are related to each other, and this will help us in creating the right relationships in the Relational Database Management System. Let us create a logical data model for our database using the same ER diagram:



The above logical model has attributes and also depicts the relationships between the different entities, giving us a "plan" of the actual database that should be developed. It is very important to make sure that the logical data model is developed with a detailed understanding of the system, as this forms the foundation on which the final database will be developed. Changes to the logical data model should be avoided during the later stages of application development as they can be costly in terms of both time and money.