Cardinality in Data Modeling

A quick guide to understanding relationship types in database design

Adam Alalouf

Types of Relationships

Cardinality describes a fundamental characteristic of the relationship between two entities.

Let’s start with the basics:

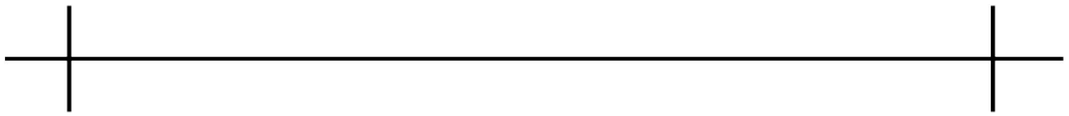
- 1:1 – a <u>one to one</u> relationship
- 1:m – a <u>one to many</u> relationship
- m:m – a <u>many to many</u> relationship

What does this mean?

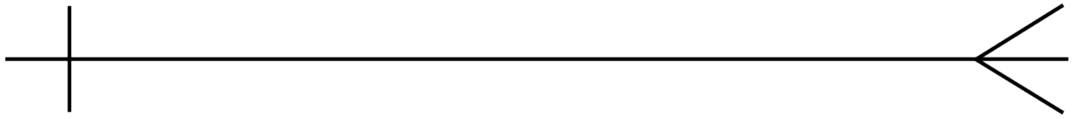
- When we say there is a 1:1 relationship between two entities, it means that for each occurrence of one entity there is exactly one occurrence of a related entity.
- When we say there is a 1:m relationship between two entities, it means that for each occurrence of one entity there is one or many occurrences of a related entity.
- When we say there is a m:m relationship between two related entities, it means that for each occurrence of either entity, there is one or many occurrences of the other entity.

Crow’s Foot Notation

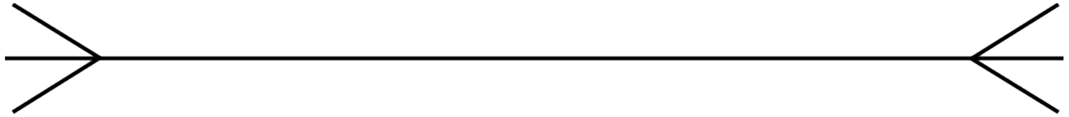
Crow’s foot notation is used to signify relationship types between entities in entity relationship diagrams (ERD). They are drawn as lines with terminal markings.



1:1 *one to one*



1:m *one to many*

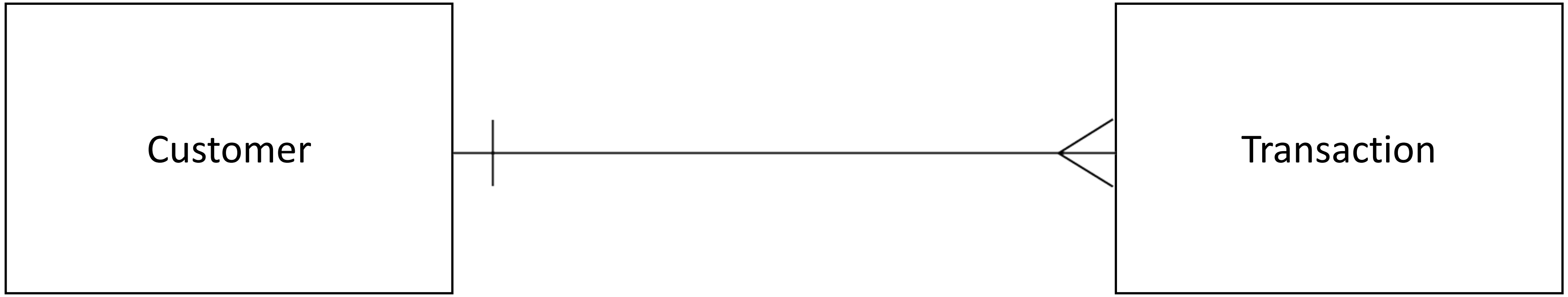


m:m *many to many*

Example

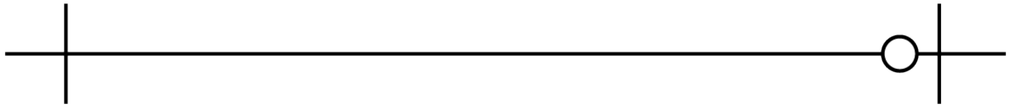
Below is an example of a 1:m relationship between *customer* and *transaction*.

These entities have a 1:m relationship because a customer can book multiple transactions but a transaction belongs to one and only one customer.

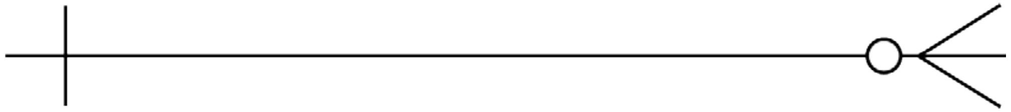


Optional Entities and Null Values

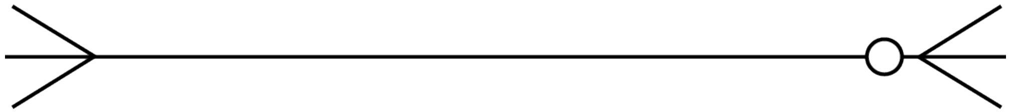
Crow’s foot notation can include a little circle, indicating a null value. This means that the related entity is not mandatory.



1:1 *one to zero or one*



1:m *one to zero or many*

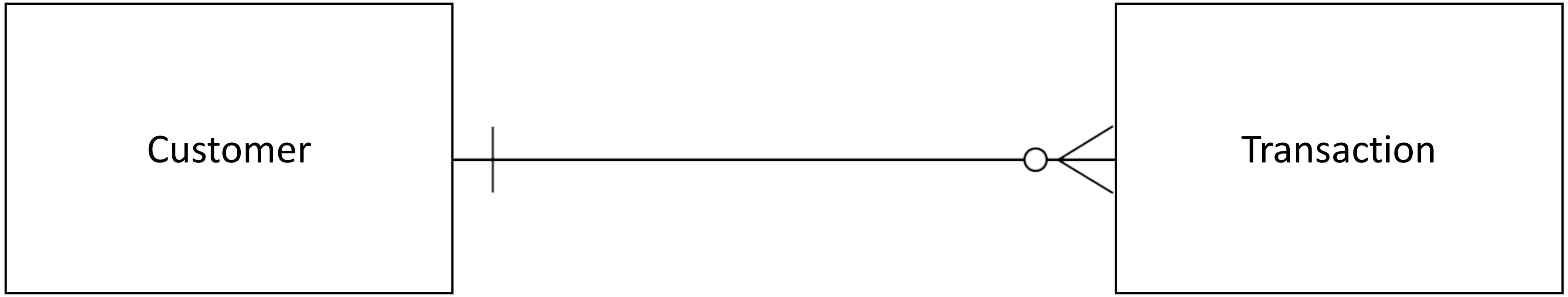


m:m *many to zero or many*

Example

Let’s review our example of the relationship between *customer* and *transaction*.

These entities now have a 1:0m relationship because a customer can book one, many, or zero transactions, but a transaction still belongs to one and only one customer.

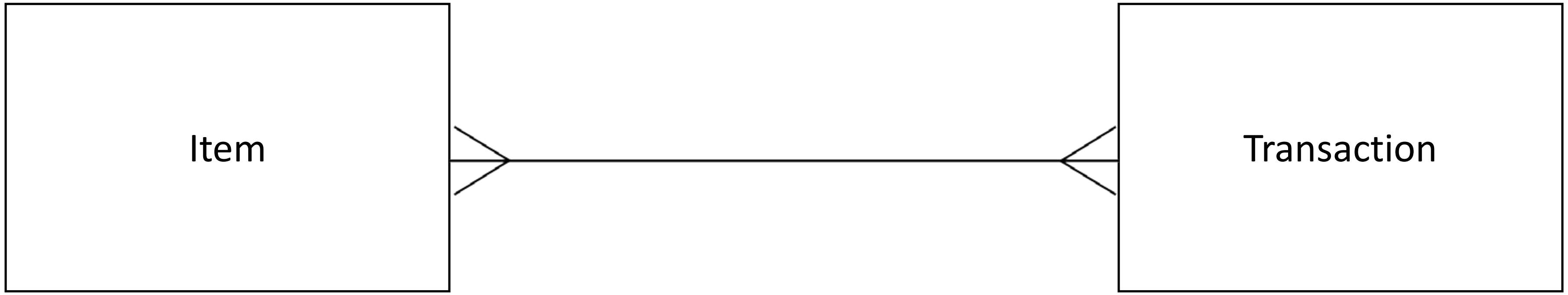


So Which Relationship Type is Correct?

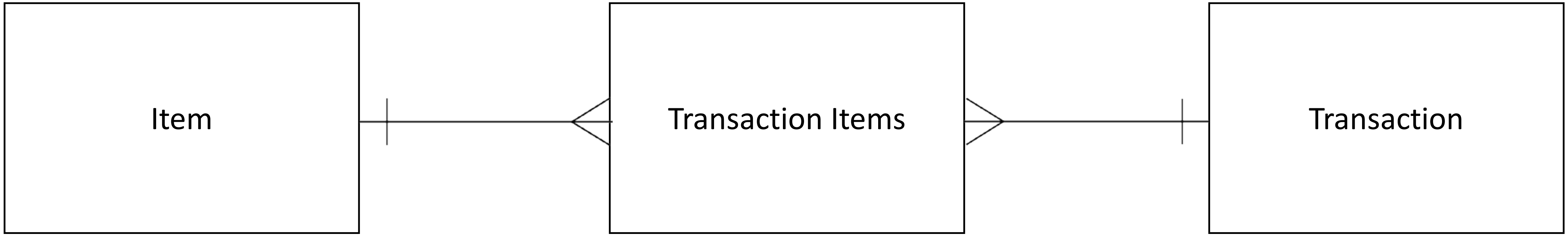
- In our customer order example, the second diagram makes more sense. A customer could exist in the database but could have zero transactions.
- However, the correct relationship type ultimately depends on the business logic of the organization.
- Best practices can help inform the most appropriate cardinality in the structure of a database model.

A note about m:m relationships

- Conceptually, a many to many relationship can be represented using crow’s foot notation.
- For example, an item can appear on several transactions, and a transaction can contain several items.



- However, this is strictly conceptual. In actual design, a many to many relationship requires a separate table, called an associative table, to capture the relationship between the two entities.
- One convention is to name the table by concatenating the names of the parent and child tables. In our example, this would create a *transaction items* table. You could also call it *transaction detail* or give it another meaningful name.
- Notice that the many to many relationship has been resolved to two one to many relationships.



A note about 1:1 relationships

- One to one relationships are rather rare in data models.
- Often, this type of relationship can be captured sufficiently within a single entity. We split such values to separate tables with a 1:1 relationship when a performance advantage is obtained by utilizing the separate table.
- For example, *employee* and *employeeHealthInsurance* have a 1:1 relationship because an employee has one enrollment, and a policy belongs to one and only one employee. Since the latter has potentially high null rate, it is better stored in a dedicated table.