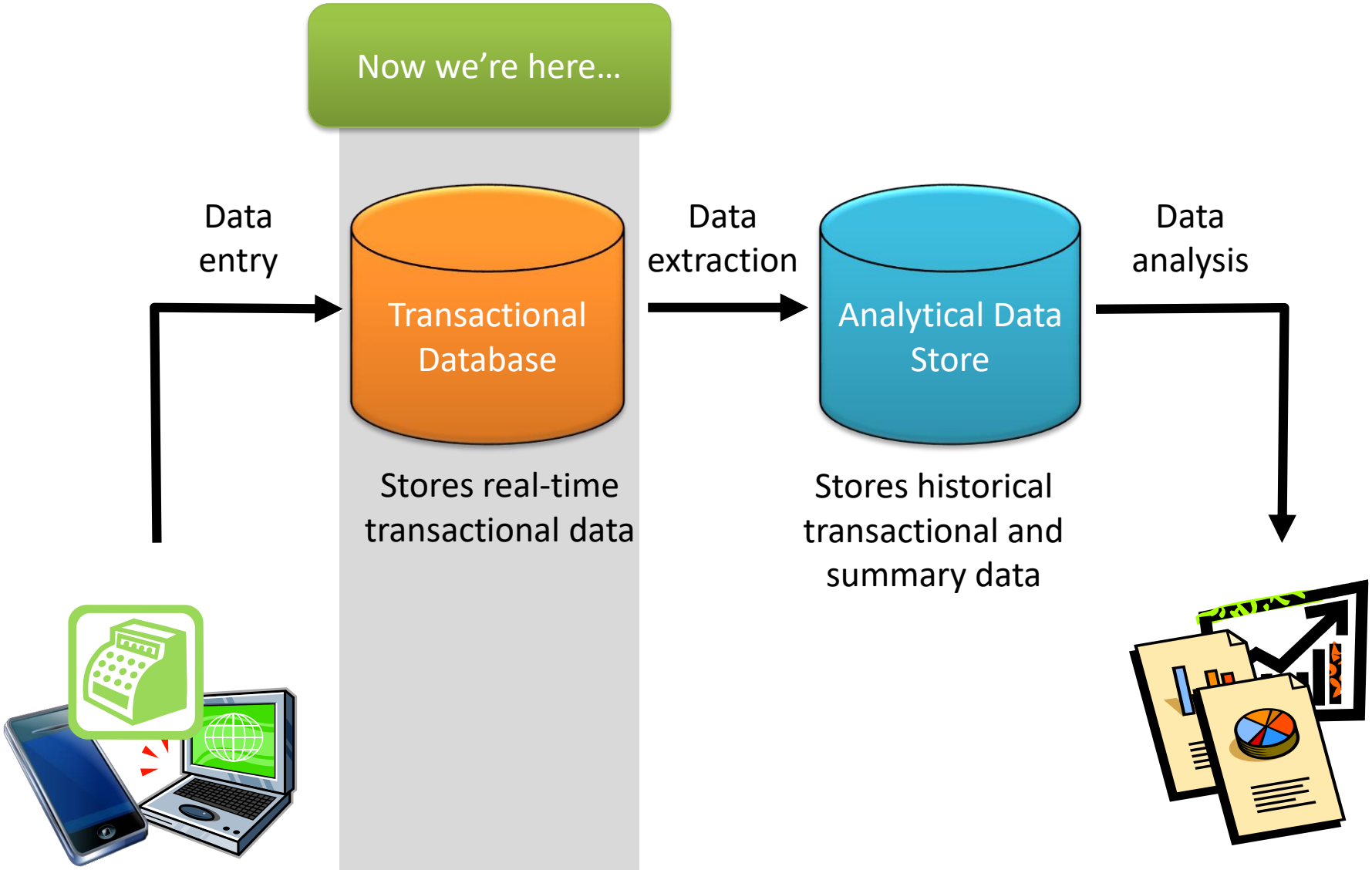
MIS2502:
Data Analytics
Relational Data Modeling (1)

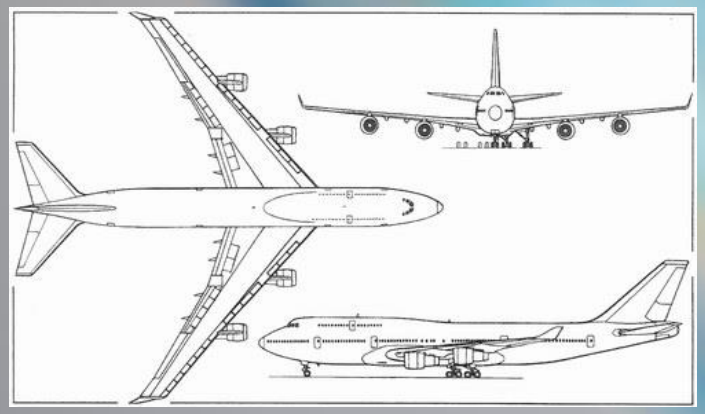
Alvin Zuyin Zheng
zheng@temple.edu
http://community.mis.temple.edu/zuyinzheng/

Where we are…


Now we're here…
Data entry
Transactional Database
Data extraction
Analytical Data Store
Data analysis
Stores real-time transactional data
Stores historical transactional and summary data

What is a model?

Representation of something in the real world



Modeling a database

- A representation of the information to be captured
- Describes the data contained in the database
- Explains how the data interrelates

Why bother modeling?

- Creates a blueprint before you start building the database
- Gets the story straight: easy for non-technical people to understand
- Minimize having to go back and make changes in the implementation stage

Start with a problem statement

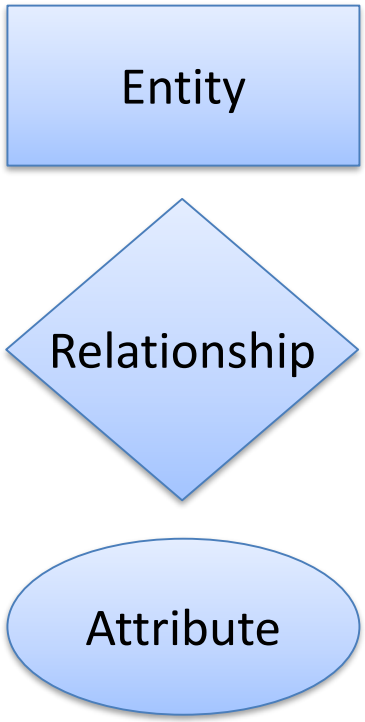
Design a database to track orders for a store. A customer places an order for a product. People can place an order for multiple products.

Record first name, last name, city, state, and zip code for customers. We also want to know the date an order was placed.

Finally, we want to track the name and price of products and the quantity of each product for each order.

The Entity Relationship Diagram (ERD)

- The primary way of modeling a relational database
- Three main diagrammatic elements



A uniquely identifiable thing (i.e., person, order, cell phone,)

Describes how two entities relate to one another (i.e., makes)

A characteristic of an entity or relationship (i.e., first name, order number)

Begin with Identifying the Entities

This is what your database is about.

1. List the nouns in the problem statement.
2. When nouns are synonyms for other nouns, choose the best one.
3. Make a note of nouns that describe other nouns. These will be your entities' attributes.
4. Rule out the nouns that don't relate to the process to be captured.

What's left are your entities!

So here are the nouns…

Design a **database** to track orders for a **store**. A **customer** places an **order** for a **product**. **People** can place an order for multiple products.

Record **first name**, **last name**, **city**, **state**, and **zip code** for customers. We also want to know the **date** an order was placed.

Finally, we want to track the **name** and **price** of products and the **quantity** of each product for each order.

Which nouns are entities

Which nouns are attributes?

Which nouns are irrelevant?

Here's where it gets tricky…

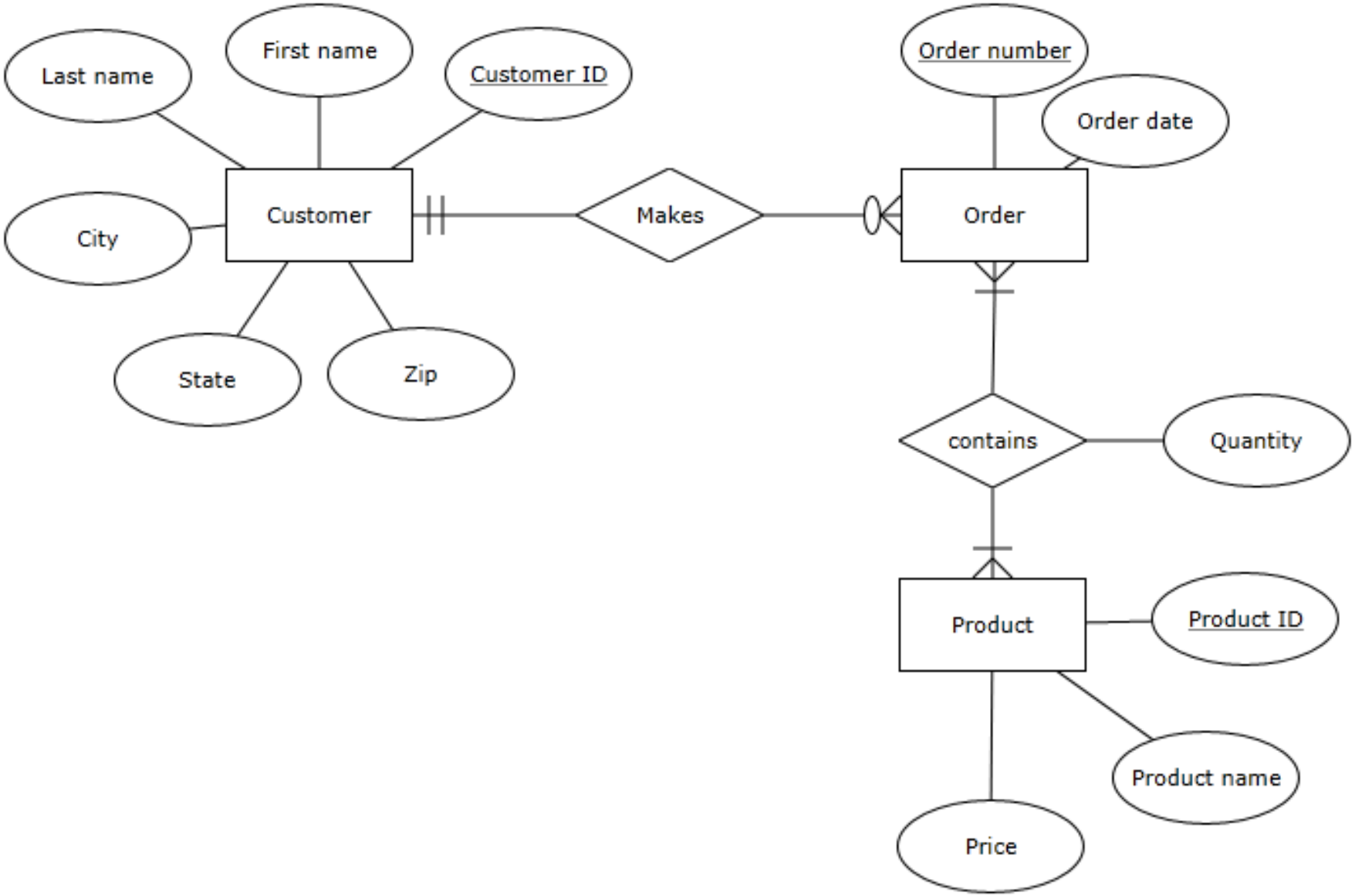
store is not an entity because we are not tracking specific information about the store (i.e., store location)

In this case, "store" is the context

BUT…if there were many stores and we wanted to track sales by store, then store would be an entity!

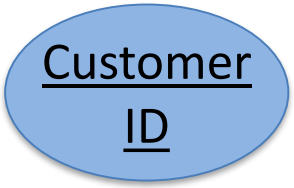
But that isn't part of the problem statement….

The ERD Based on the Problem Statement

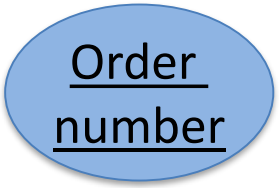


The primary key

- Entities need to be uniquely identifiable
  - So you can tell them apart
  - They may not be explicitly part of the problem statement, but you need them!
- Use a primary key
  - One or more attributes that uniquely identifies an entity



Uniquely identifies a customer



Uniquely identifies an order

How about these as primary keys for Customer:

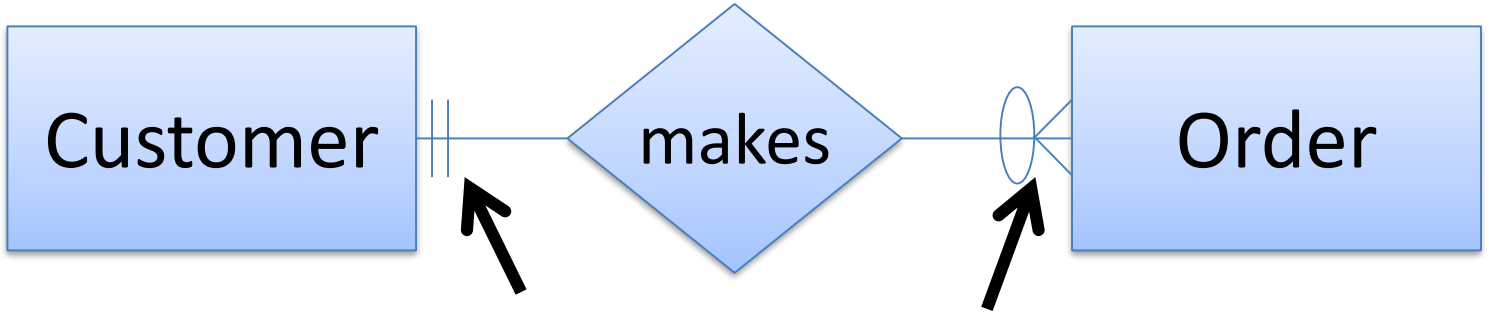
First name and/or last name?

Social security number?

IN CLASS ACTIVITIES

Last component: Cardinality

- Defines the rules of the association between entities



Minimum cardinality: **at least** – one

Maximum cardinality: **at most** - one

at least – zero (optional)

at most - many

This is a one-to-many (1:m) relationship:

- One customer can have **many** orders.
- One order can only belong to **one** customer.

Additionally

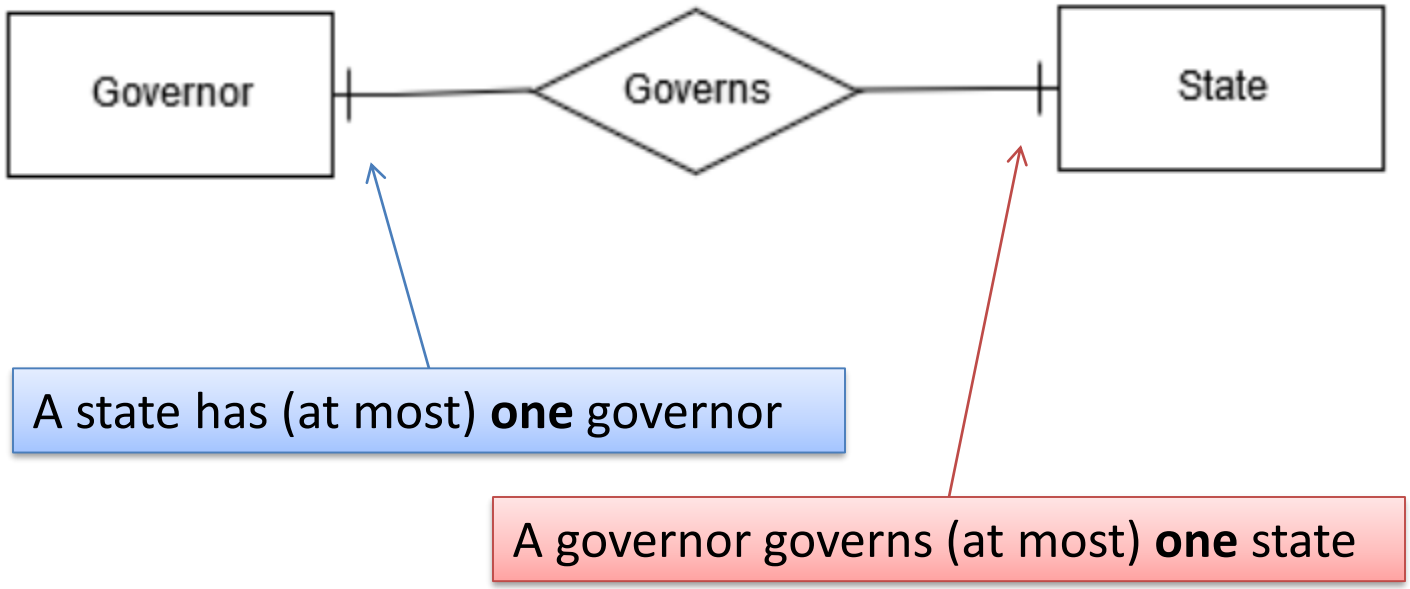
- A customer could have **no orders**.
- An order has to belong to **at least one** customer.

Maximum and Minimum Cardinality

- Maximum cardinality (type of relationship)
  - Describes the maximum number of entity instances that participate in a relationship
    - One-to-one
    - One-to-many
    - Many-to-one
    - Many-to-many
- Minimum cardinality
  - Describes the minimum number of entity instances that must participate in a relationship

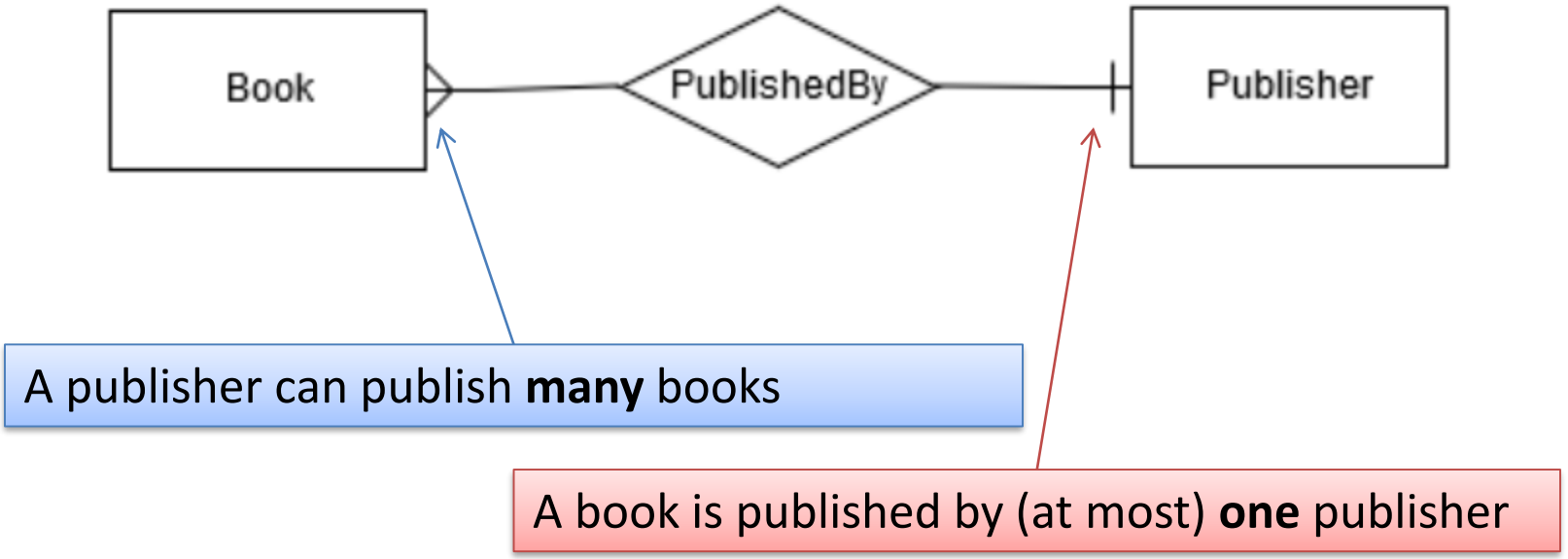
One-to-One Relationship

- One-to-One (1:1)
  - A single instance of one entity is related to a single instance of another entity



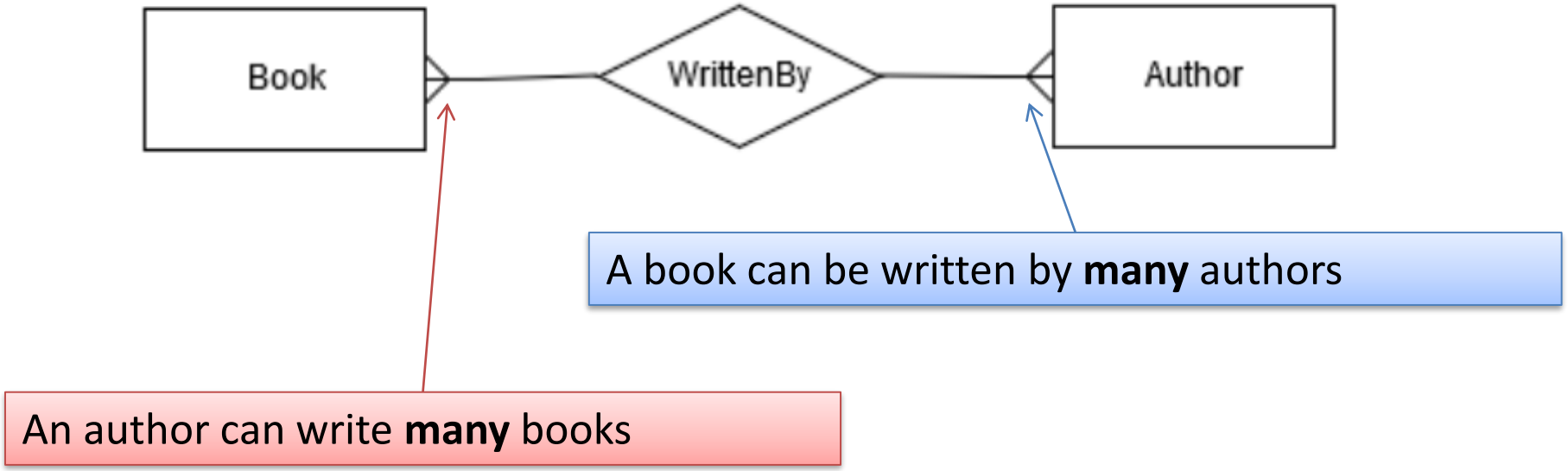
One-to-Many Relationship

- One-to-Many (1:n or 1:m)
  - A single instance of one entity is related to multiple instances of another entity



Many-to-Many Relationship

- Many-to-Many (n:n or m:m)
  - Each instance of one entity is related to multiple instances of another entity, and vice versa

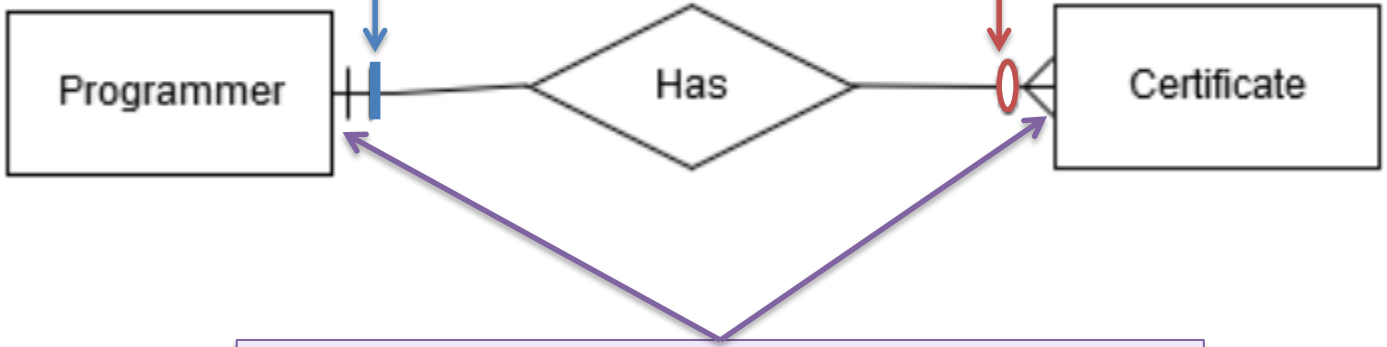


Minimum Cardinality

- Minimums are generally stated as either zero or one:
  - 0 (optional): participation in the relationship by the entity is optional.
  - 1 (mandatory): participation in the relationship by the entity is mandatory.

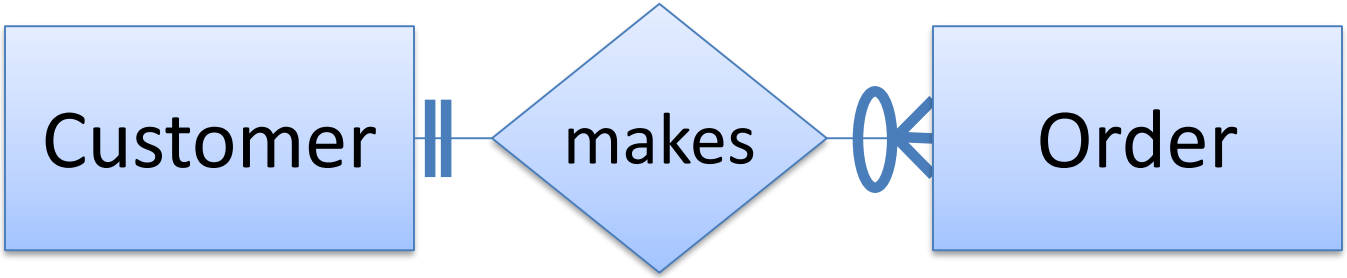
A programmer is **mandatory** for a certificate); or a certificate has to be issued to (at least) one programmer.

A certificate is **optional** for a programmer; or a programmer may not have any certificates



1:m maximum cardinality: a programmer can have **many** certificates; a certificate is issued to (at most) **one** programmer

Crows Feet Notation



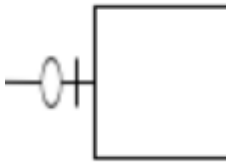
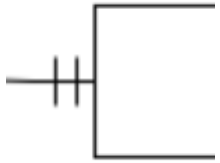
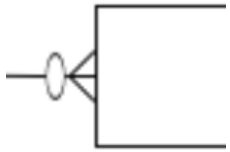
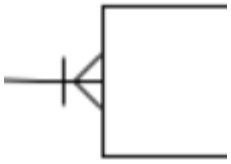
So called because this…

…looks something like this

There are other ways of denoting cardinality, but this one is pretty standard.

There are also variations of the crows feet notion!

Crow's Foot Notation Summary

Symbol	Minimum Cardinality + Maximum Cardinality
	**Optional + One**
	**Mandatory + One**
	**Optional + Many**
	**Mandatory + Many**

The Order-Product Example: A Many-to-Many (m:m) Relationship

An order can be composed of **many** products.

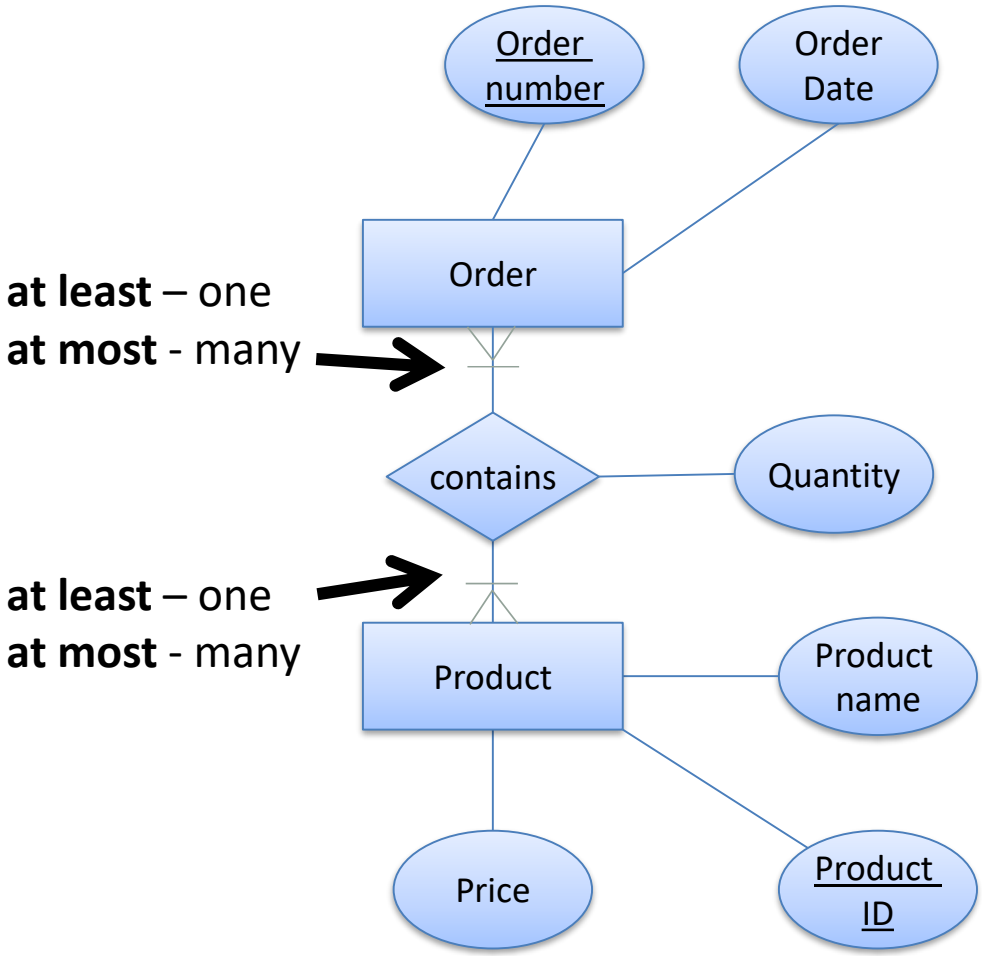
An order has to have **at least one** product.

A product can be a part of **many** orders.

A product has to be associated with **at least one** order.

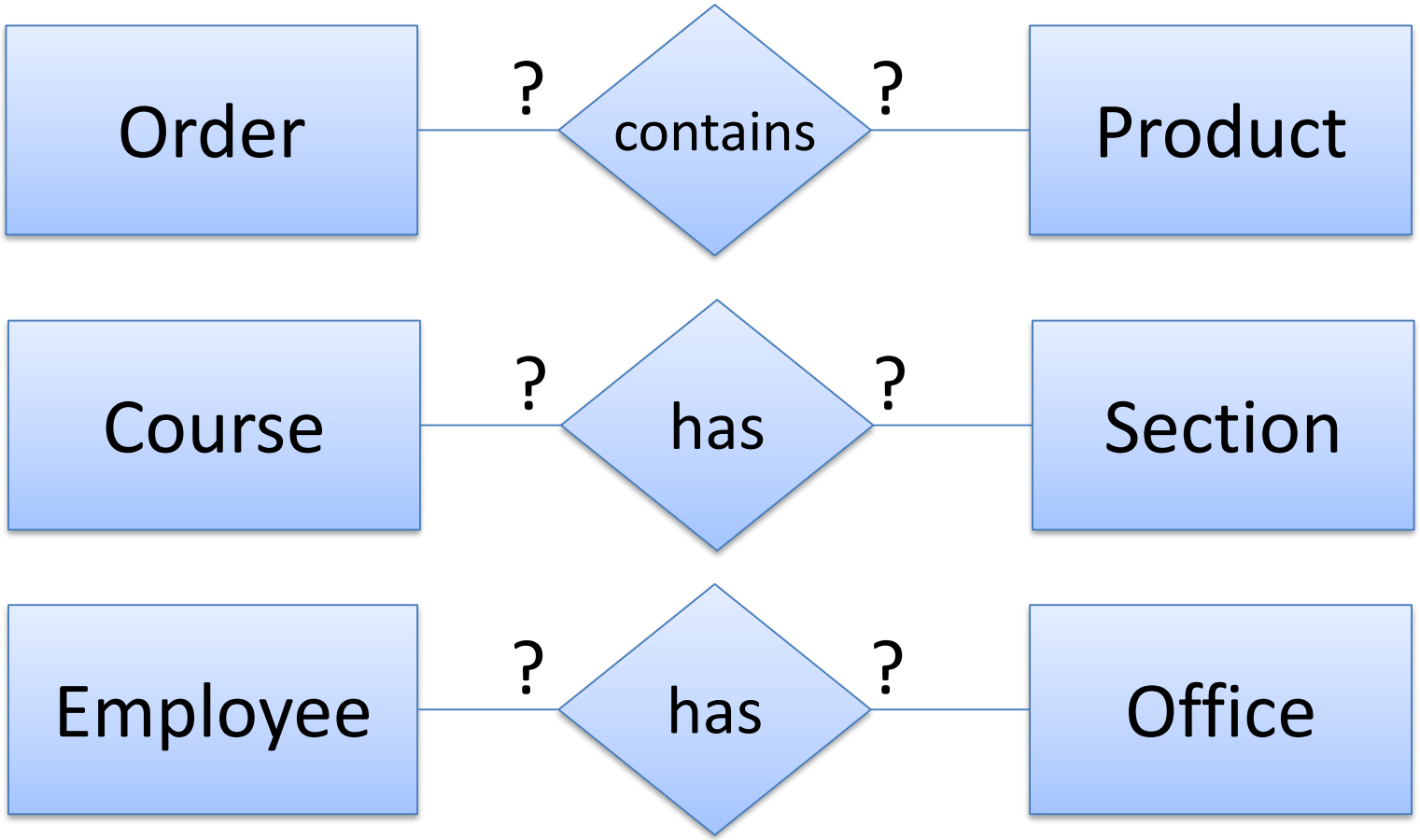
Does it make sense for the maximum cardinality to be 1 for either entity?

Does it make sense for the minimum cardinality to be 0 (optional) for either entity?

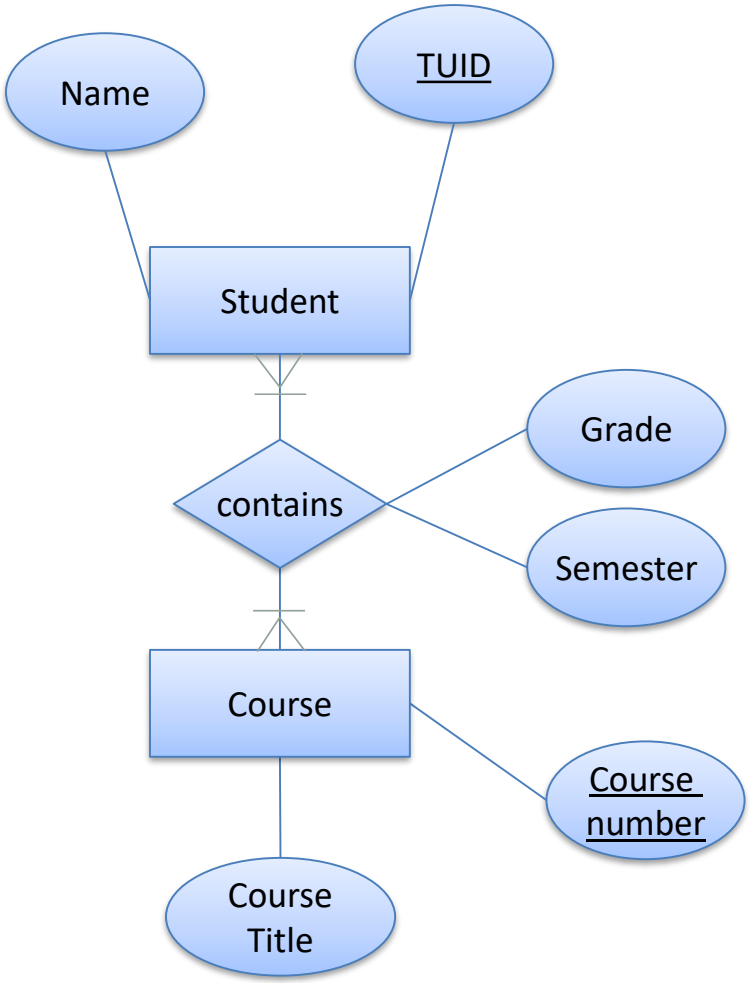


Cardinality is defined by business rules

- What would the cardinality be in these situations?



Relationship Attributes



The grade and semester describes the combination of student and course

(i.e., Bob takes MIS2502 in Fall 2011 and receives a B; Sue takes MIS2502 in Fall 2012 and receives an A)

A scenario: The auto repair shop

Each **transaction** is associated with a **repair**, a **car**, and a **mechanic**.

Cars, **repairs**, and **mechanics** can all be part of multiple **transactions**.

Many **transactions** can make up an **invoice**.

A **transaction** can only belong to one **invoice**.

A **car** is described by a VIN, make, and model.

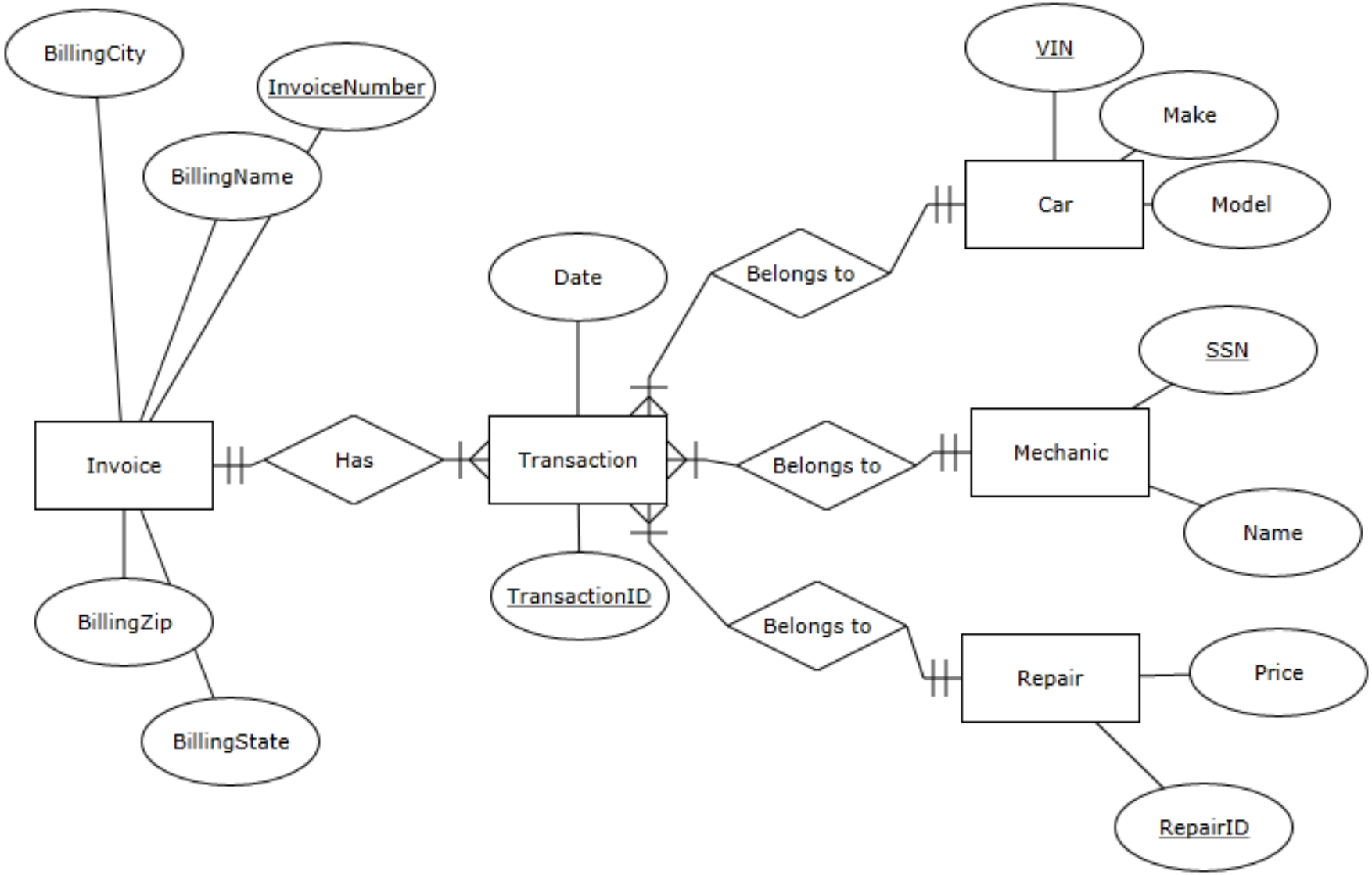
A **mechanic** is described by a name and SSN.

A **repair** is described by a repair id and a price.

A **transaction** occurs on a particular date and has a transaction ID.

An **invoice** has an invoice number and a billing name, city, state, and zip code.

Solution


BillingCity
InvoiceNumber
BillingName
VIN
Make
Car
Model
Date
Belongs to
SSN
Invoice
Has
Transaction
Belongs to
Mechanic
Name
TransactionID
Belongs to
BillingZip
Repair
Price
BillingState
RepairID

Normalization

- Organizing data to minimize redundancy (repeated data)

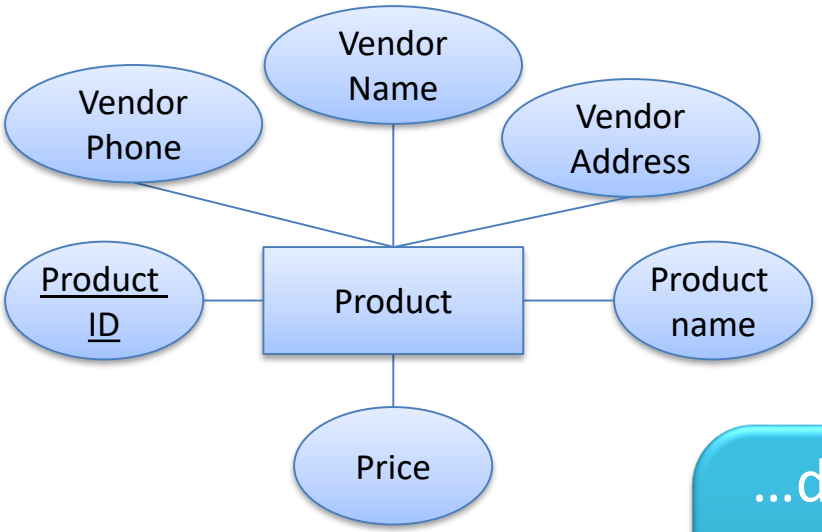
- This is good for several reasons
  - The database takes up less space
  - Fewer inconsistencies in your data
  - Easier to search and navigate the data

- It's easier to make changes to the data
  - The relationships take care of the rest

Normalizing your ER Model

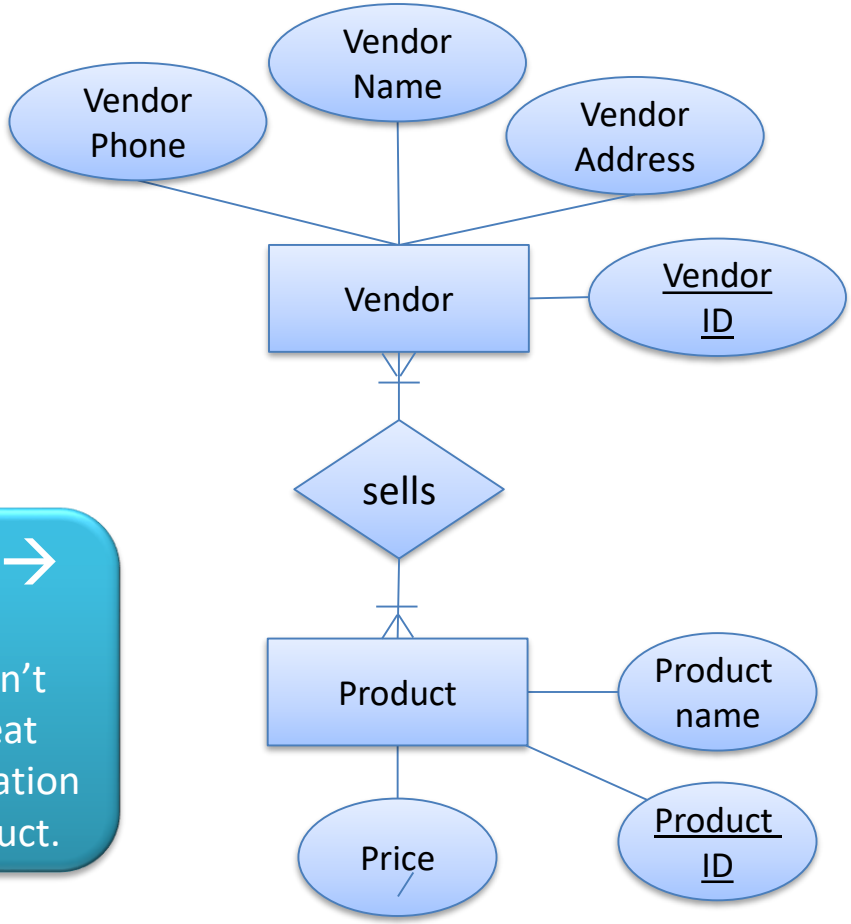
If an entity has multiple sets of related attributes, split them up into separate entities

Don't do this…



…do this →

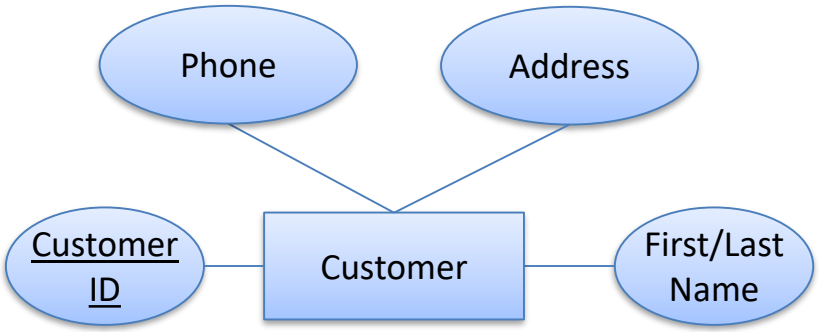
Then you won't have to repeat vendor information for each product.



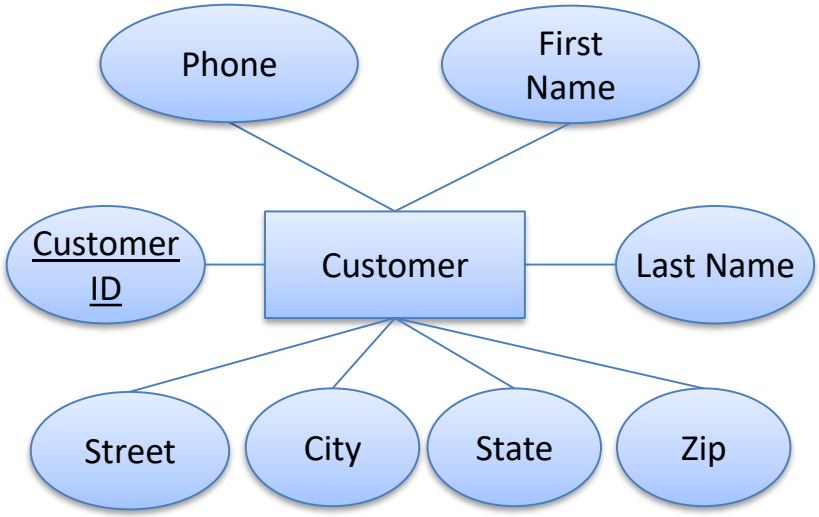
Normalizing your ER Model

Each attribute should be **atomic** – you can't (logically) break it up any further.

Don't do this…



…do this



This way you can search or sort by last name OR first name, and by city, state, or zip code.

Summary of ERD

- **Key concepts**
  - Entity
  - Relationship
  - Cardinality
    - Minimum cardinality: 1:1, 1:m, m:m
    - Maximum cardinality: optional or mandatory (i.e., 0 or 1)
    - Crow's foot notation
  - Attributes
    - Entity attributes: primary key vs. non-key
    - Relationship attributes

- **Key skills**
  - Interpret simple ERDs
  - Draw an ERD based on a scenario description

Drawing ERD: A Checklist

- Entities
- Entity attributes
  - ✓Primary key
  - ✓Non-key attributes
- Relationships
  - ✓Minimum cardinality
  - ✓Maximum cardinality
- Relationship attributes