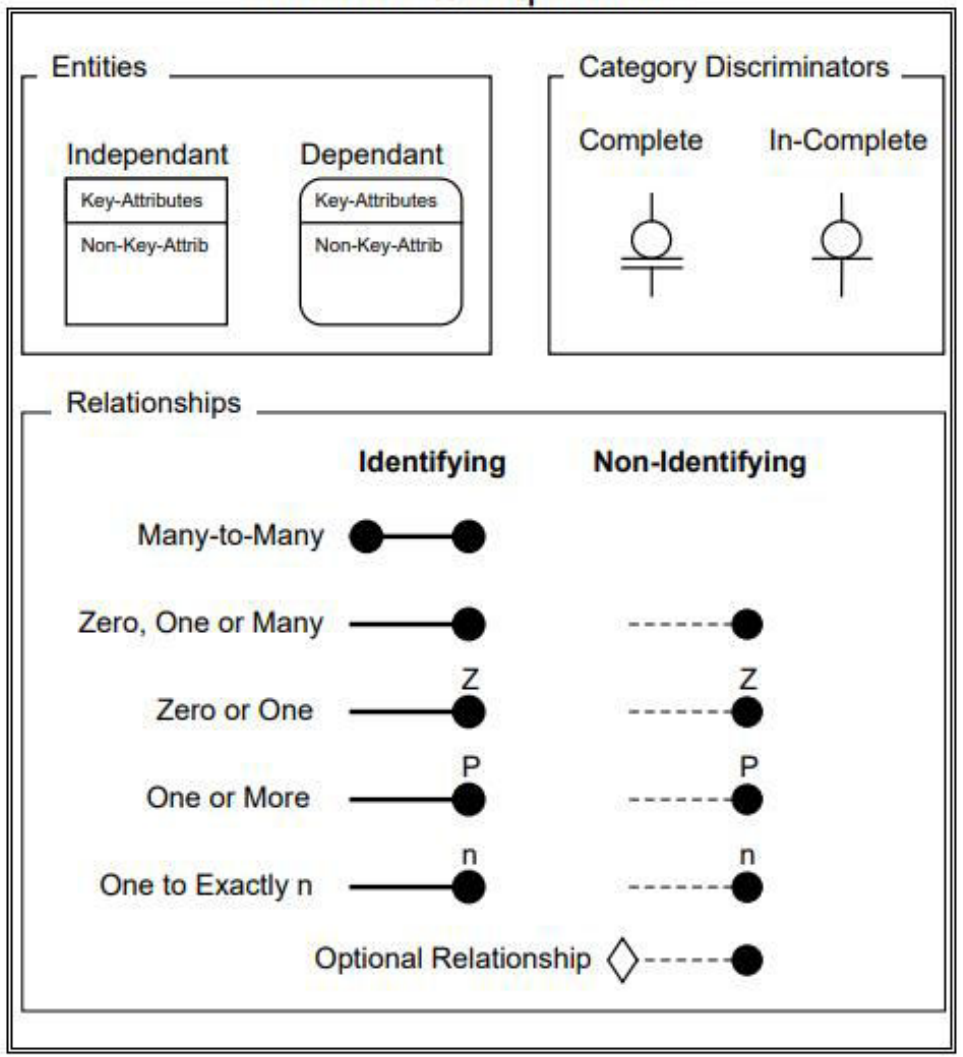
Outline

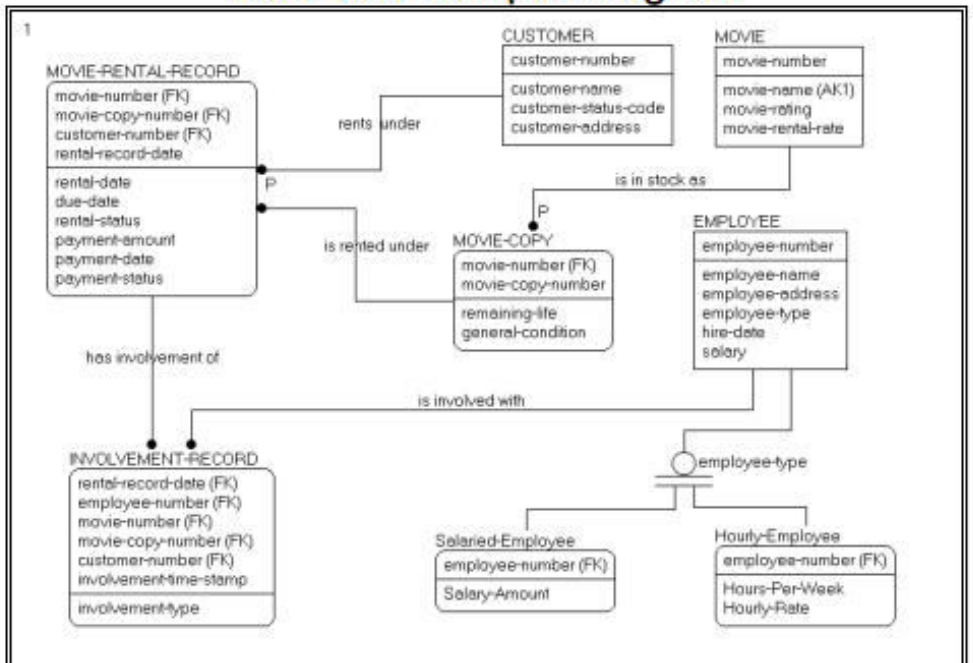
- **ER modeling**
- **Elements of an ER model**
  - Entity types
  - Attributes
  - Relationships
    - Regular (non-identifying)
    - Identifying
    - Special relationships
      - One-to-one
      - Many-to-many
      - Recursive
      - Higher arity
  - Subtypes

IDEF1X Components



Entity: any person, place, thing, event or concept about which information is kept.
Attributes: Information about a certain property of an entity.
Key-Attributes: Information which is used to uniquely identify an instance of an entity.
Non-Key-Attributes: Other information about an entity.
Independent Entities: An entity which does not depend upon any other entity in a model for its identification.
Dependent Entities: Depend both for their existence and their identification upon other entities in the model.
Category Entity: A special type of entity which is the "*subtype*" of a parent entity.
Category Discriminators: Are attributes which indicate how we can tell one ***Category Entity*** from another.
Complete SubCategory: Indicate that the current subcategory entities form the complete range of values for the Category Discriminator.
In-Complete SubCategory: Indicate that the current subcategory entities are only a partial list of the values the Category Discriminator can have.
Relationships: Represent connections, links, or associations between entities.
Parent Entities: Entities that originate (side without the dot) a relationship.
Child Entities: Entities that are the target of the relationship (Side with the dot).
Cardinality: The number of instances of a child entity that are described as participating in the relationship.
Identifying Relationship: The keys from the parent form part of the key for the child entity.
Non-Identifying Relationship: The keys of the parent entity become Foreign Keys in the child entity.

IDEF1X Example Diagram



ER modeling

- **Many methods, many notations**
  - Entity-Relationship (ER) modeling
  - This course uses the IDEF1X notation
- **Benefits**
  - Relatively easy to understand
  - Hide/expose details when zooming in or out
  - Maps to relational database design
- **Risks**
  - Wrong responsibility
  - Incomplete
  - Different notations

Entity types

- **Entity**
  - Objects, persons, events, or abstractions
  - Relevant in the context of the data application
  - Also called "entity instance" (or "instance")

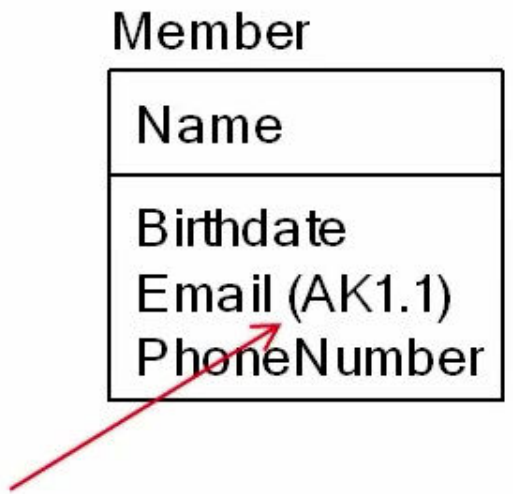
- **Entity type**
  - Class of objects
  - Same characteristics
  - Also called "entity"
    - (And yes, that is indeed confusing!)

Entity types

Entity instances	Entity type
Mary Dave	Club member
Table 1 Table 3	Table
2012 Christmas Tournament 2013 Midsummer Tournament	Tournament
A League B League C League Junior League	League

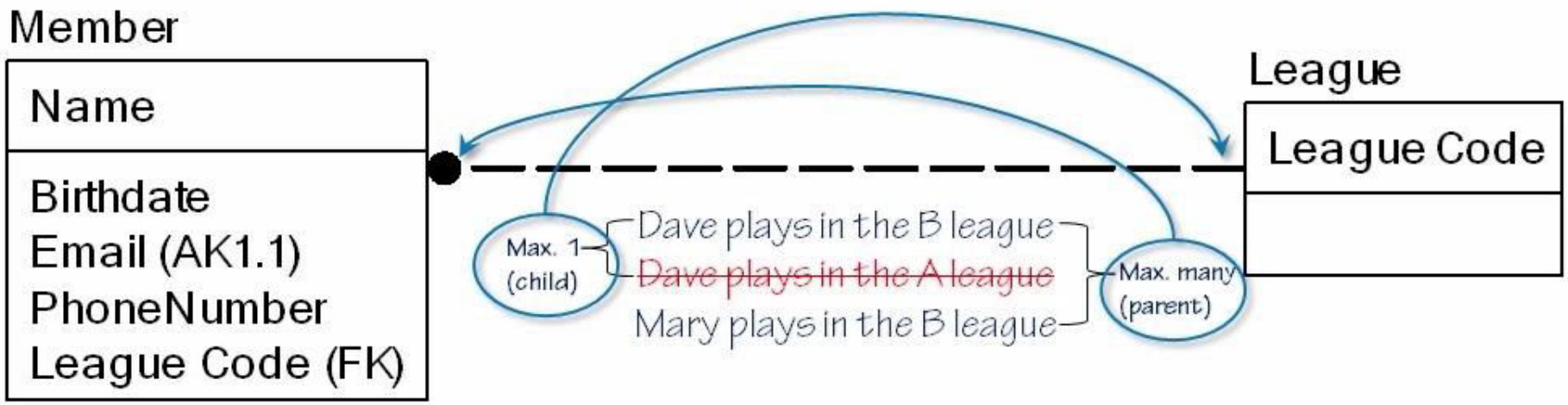
Attributes

- **Instance level: A fact about an entity occurrence**
- **Abstract level: A class of facts about instances of an entity type**
- **Key attributes**
  - Composite key
- **Candidate keys**
  - One primary key
  - Zero or more alternate keys



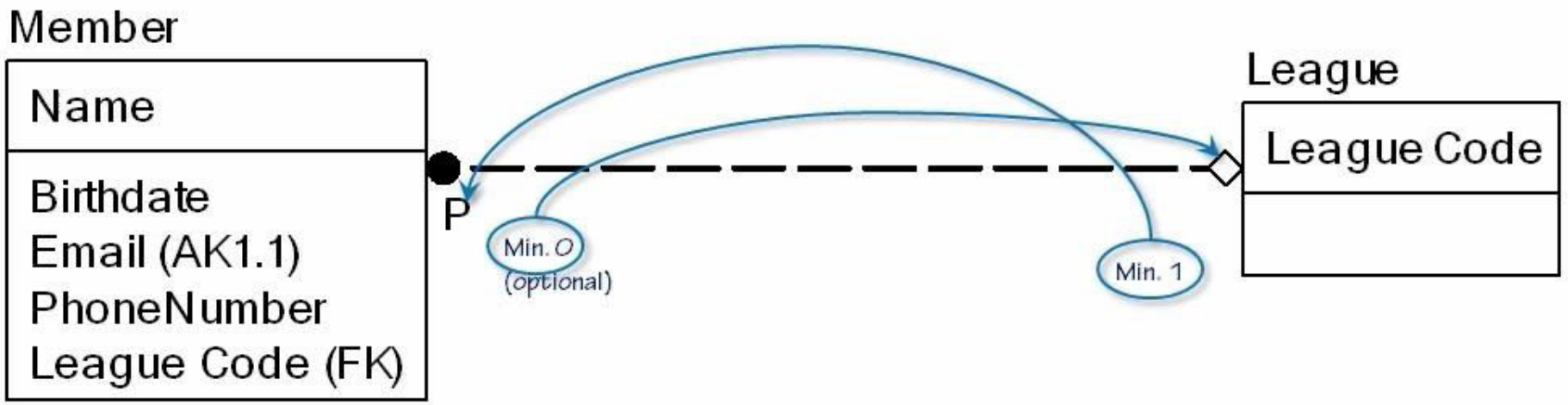
Relationships

- **Class of facts that associate an instance of an entity type with another instance of an entity type**
- **Cardinality**
  - One-to-many: one "parent" may associate with multiple "children"



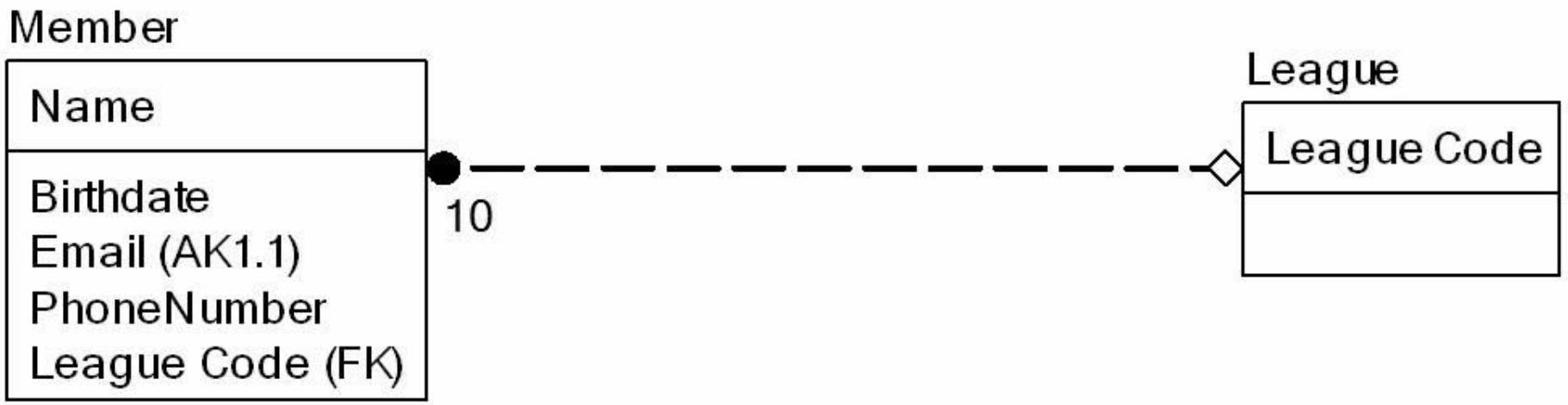
Relationships

- **Class of facts that associate an instance of an entity type with another instance of an entity type**
- **Cardinality**
  - One-to-many: one "parent" may associate with multiple "children"
  - Minimum cardinality: zero or one (optional or mandatory)



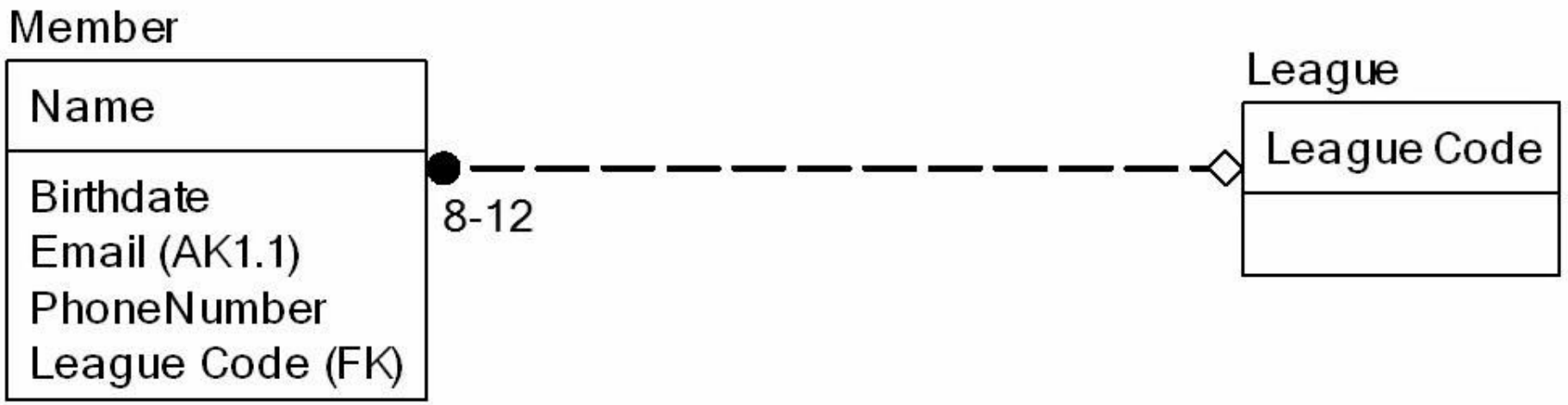
Relationships

- **Class of facts that associate an instance of an entity type with another instance of an entity type**
- **Cardinality**
  - One-to-many: one "parent" may associate with multiple "children"
  - Minimum cardinality: zero or one (optional or mandatory)



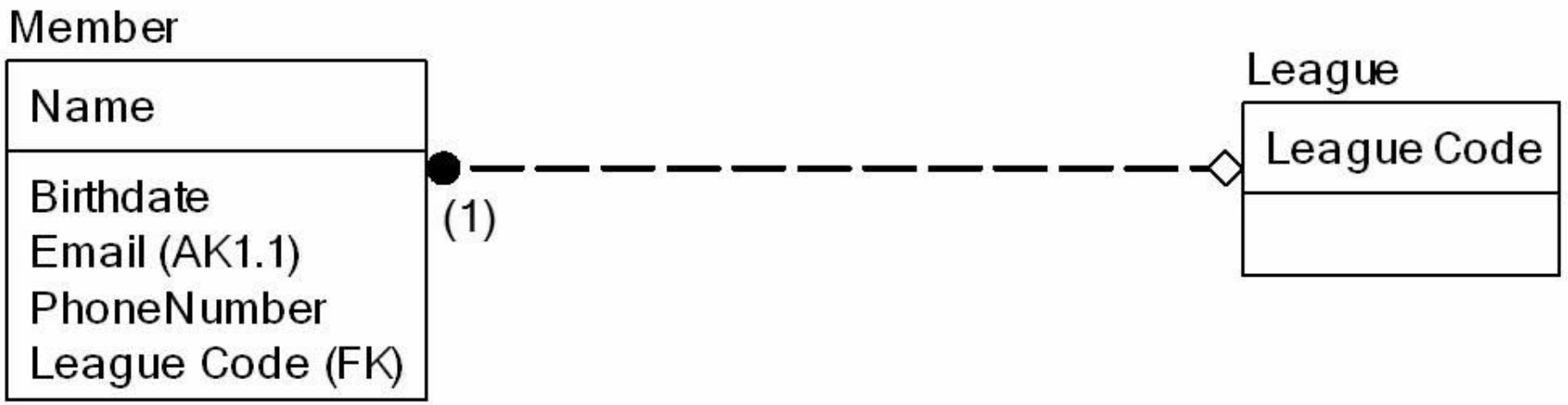
Relationships

- **Class of facts that associate an instance of an entity type with another instance of an entity type**
- **Cardinality**
  - One-to-many: one "parent" may associate with multiple "children"
  - Minimum cardinality: zero or one (optional or mandatory)



Relationships

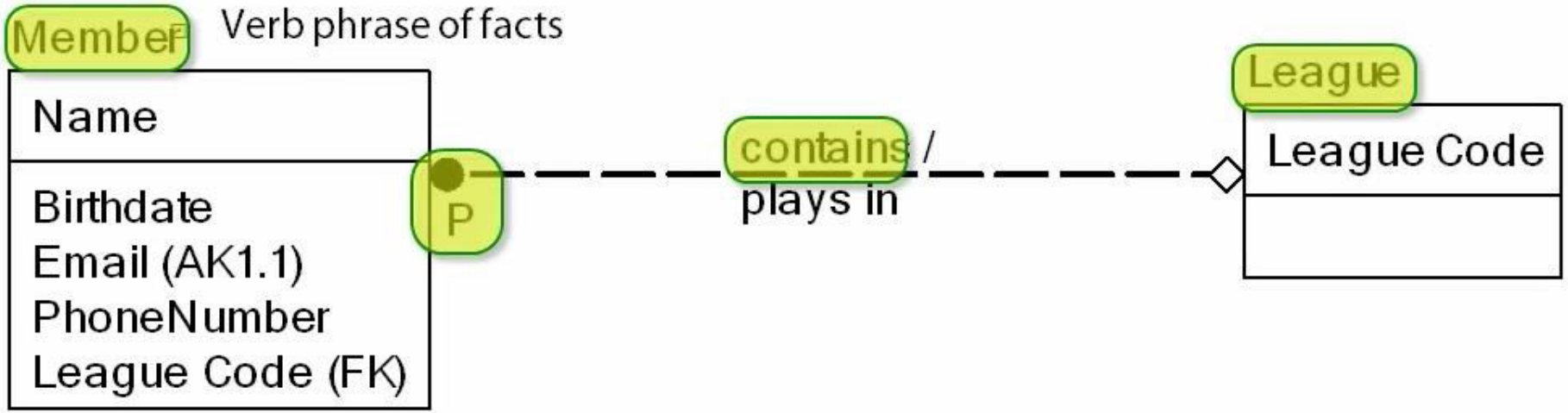
- **Class of facts that associate an instance of an entity type with another instance of an entity type**
- **Cardinality**
  - One-to-many: one "parent" may associate with multiple "children"
  - Minimum cardinality: zero or one (optional or mandatory)



(1) Each league contains an even number of members

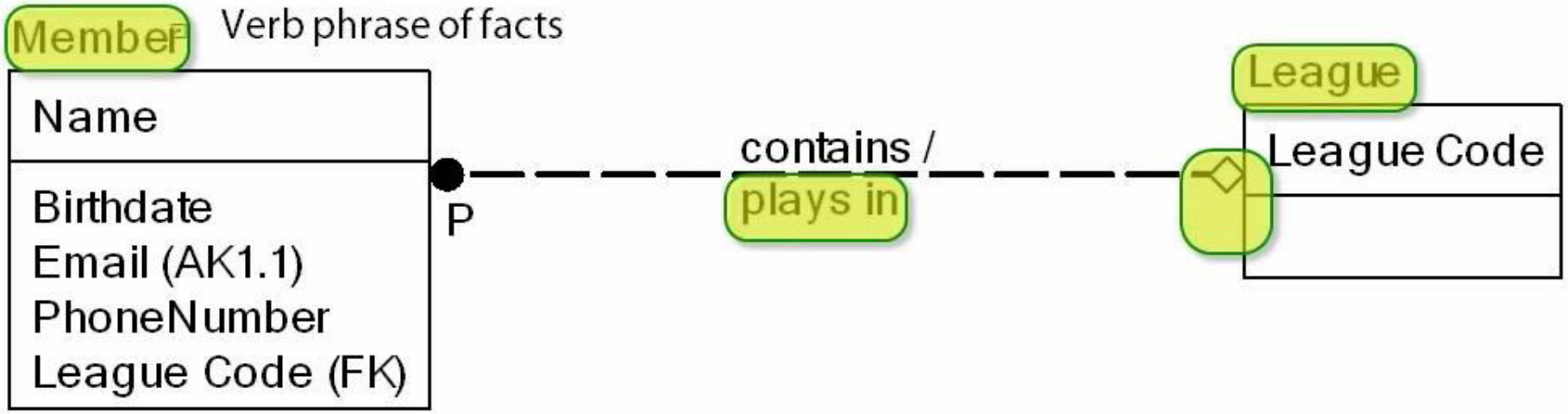
Relationships

- **Class of facts that associate an instance of an entity type with another instance of an entity type**
- **Cardinality**
  - One-to-many: one "parent" may associate with multiple "children"
  - Minimum cardinality: zero or one (optional or mandatory)
- **Relationship readings**
  - Verb phrase of facts



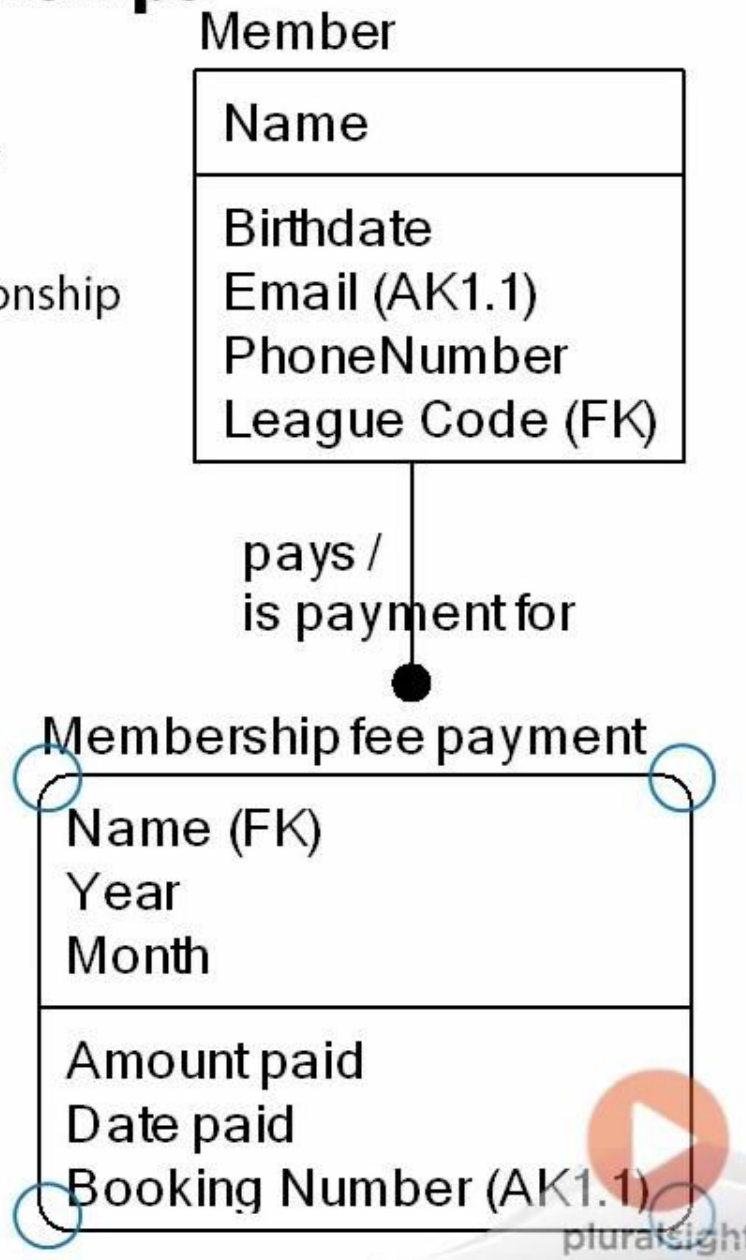
Relationships

- **Class of facts that associate an instance of an entity type with another instance of an entity type**
- **Cardinality**
  - One-to-many: one "parent" may associate with multiple "children"
  - Minimum cardinality: zero or one (optional or mandatory)
- **Relationship readings**
  - Verb phrase of facts



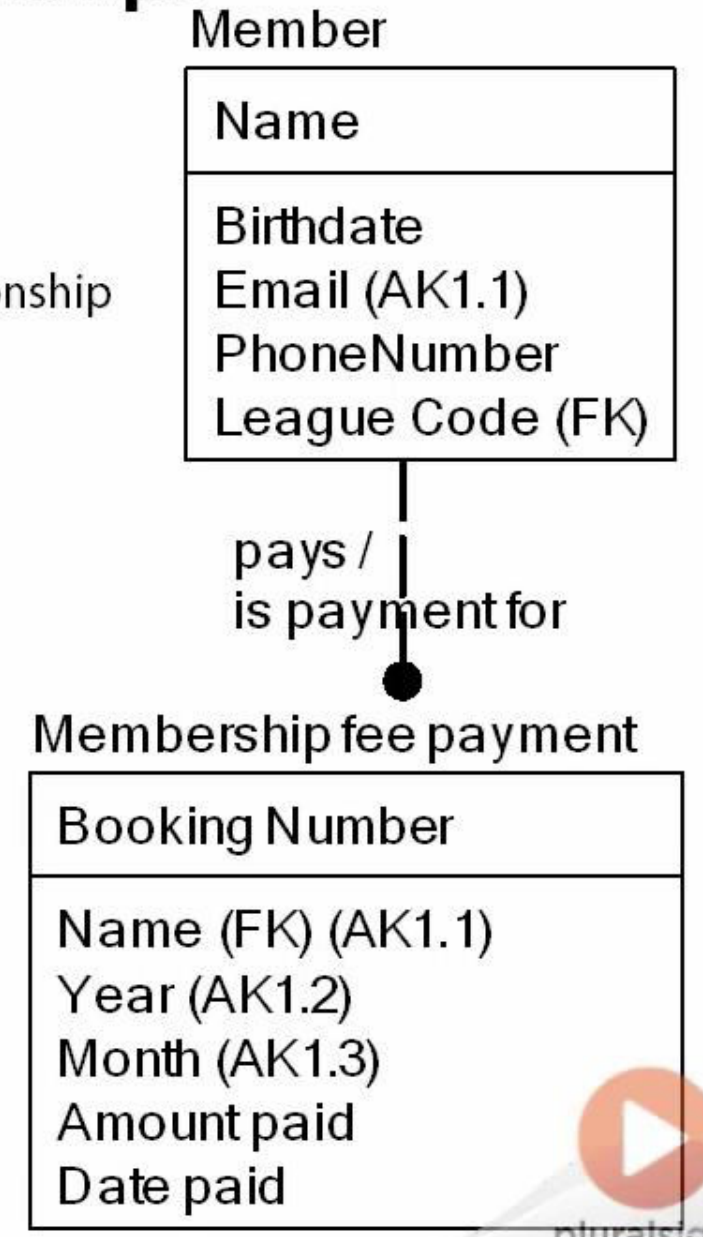
Identifying relationships

- **Foreign key attribute(s) part of child entity types key**
  - Cardinality for parent: same as normal relationship
  - Cardinality for child: never optional
- **Entity types:**
  - Child in identifying relationship: *weak*
  - Others: *strong*



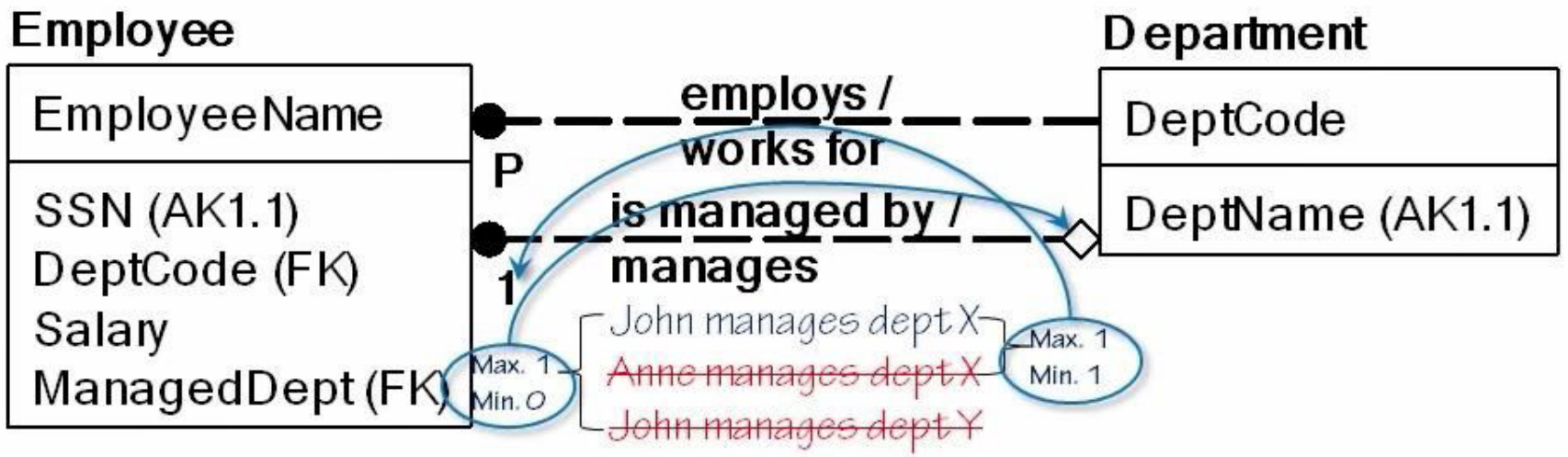
Identifying relationships

- **Foreign key attribute(s) part of child entity types key**
  - Cardinality for parent: same as normal relationship
  - Cardinality for child: never optional
- **Entity types:**
  - Child in identifying relationship: *weak*
  - Others: *strong*



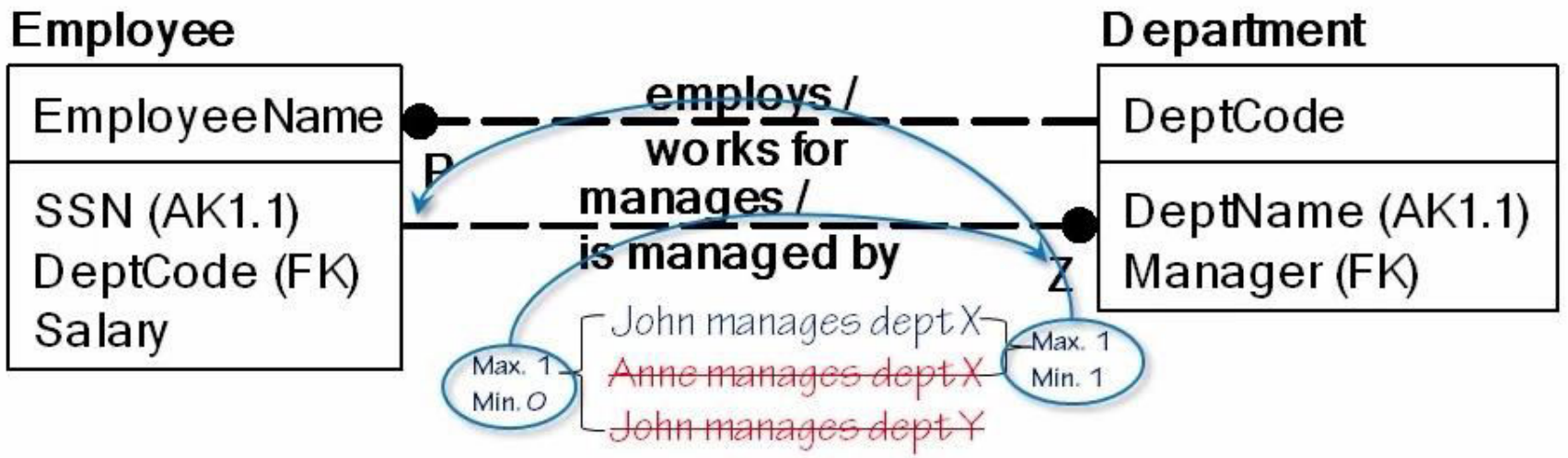
Special relationships

- **One to one**
  - Designate one of the entity types as parent, one as child
  - In physical model, the choice is based on storage and performance
  - In logical model, either choice is valid



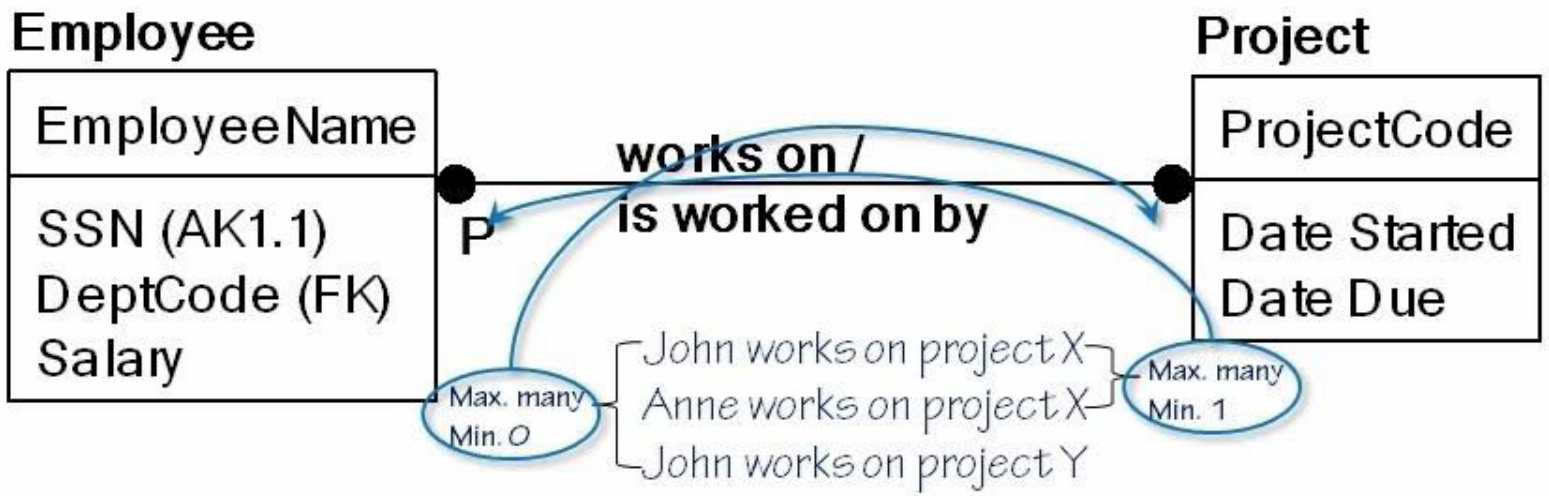
Special relationships

- **One to one**
  - Designate one of the entity types as parent, one as child
  - In physical model, the choice is based on storage and performance
  - In logical model, either choice is valid



Special relationships

- One to one
- Many to many
  - Both sides have maximum cardinality *many*
  - No parent or child in the relationship

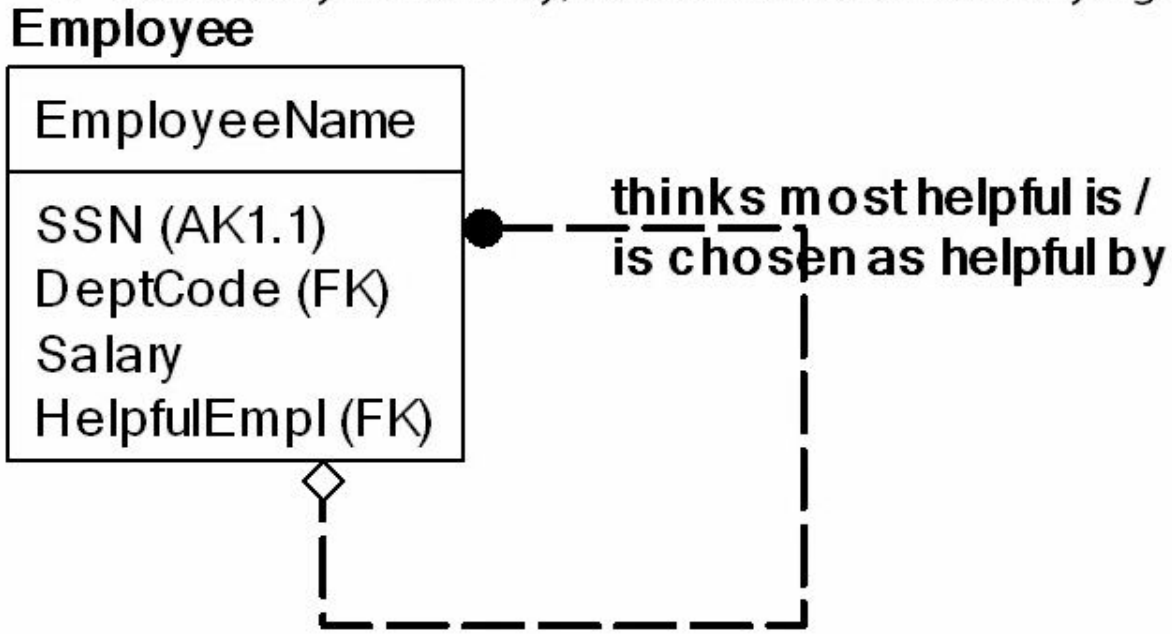


Special relationships

- One to one
- Many to many
- Recursive
  - Relates instances of an entity type to (usually different) instances of **the same** entity type

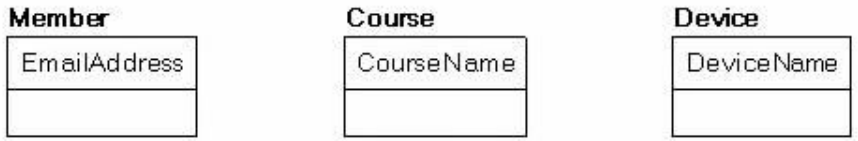
Special relationships

- One to one
- Many to many
- Recursive
  - Relates instances of an entity type to (usually different) instances of **the same** entity type
  - Can be any cardinality, but can not be an identifying relationship



Special relationships

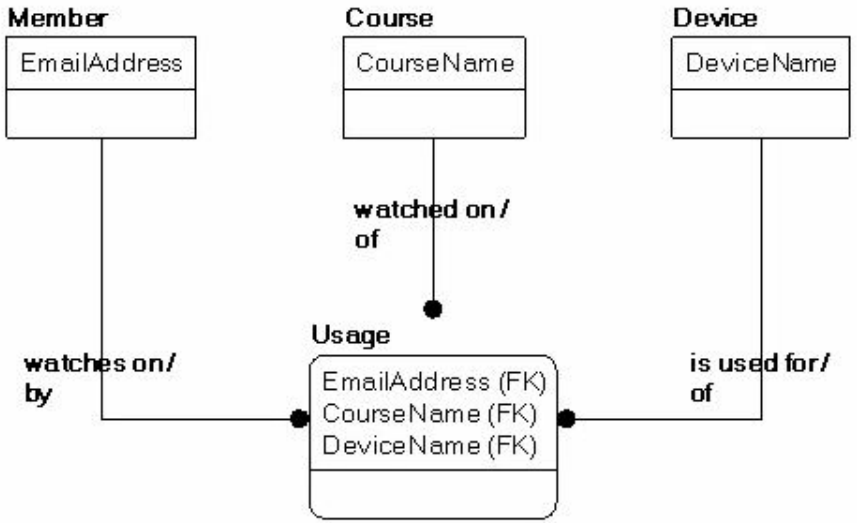
- One to one
- Many to many
- Recursive
- Higher arity
  - Binary (arity 2) relationships: between **two** entity types
  - Ternary (arity 3) relationships: between **three** entity types
  - Nominalize: transform relationship into entity type



Member a@b.c's use of a tablet to watch Database design
Member d@e.f's use of a smartphone to watch Introduction to BI

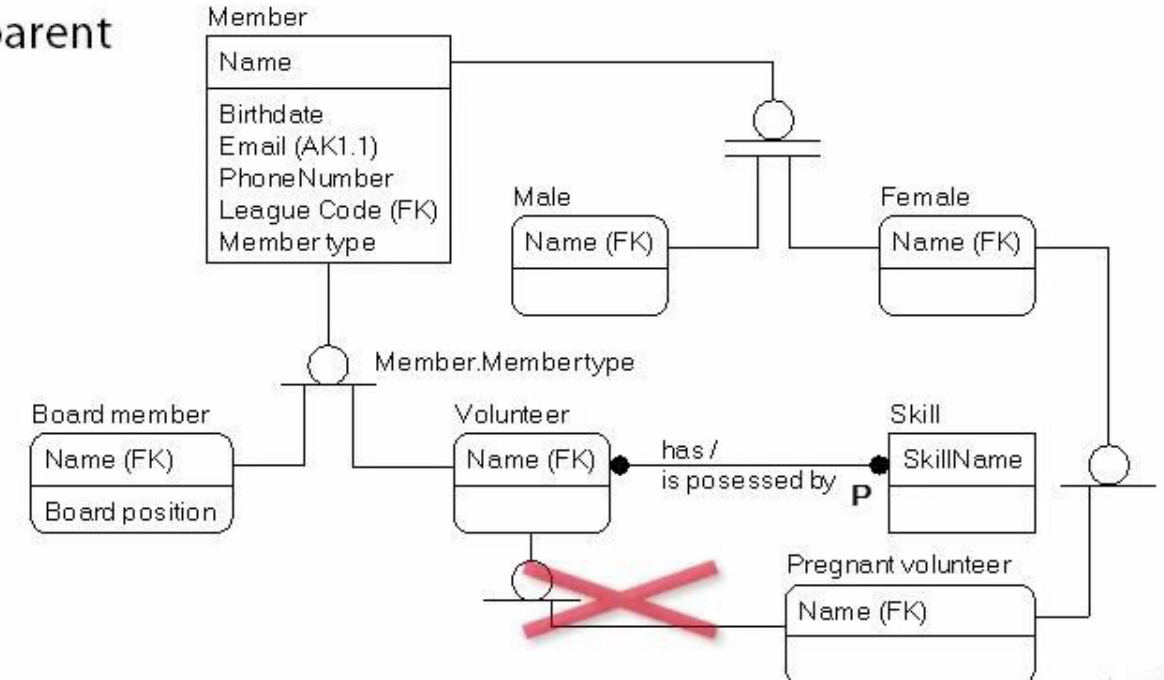
Special relationships

- One to one
- Many to many
- Recursive
- Higher arity
  - Binary (arity 2) relationships: between **two** entity types
  - Ternary (arity 3) relationships: between **three** entity types
  - Nominalize: transform relationship into entity type

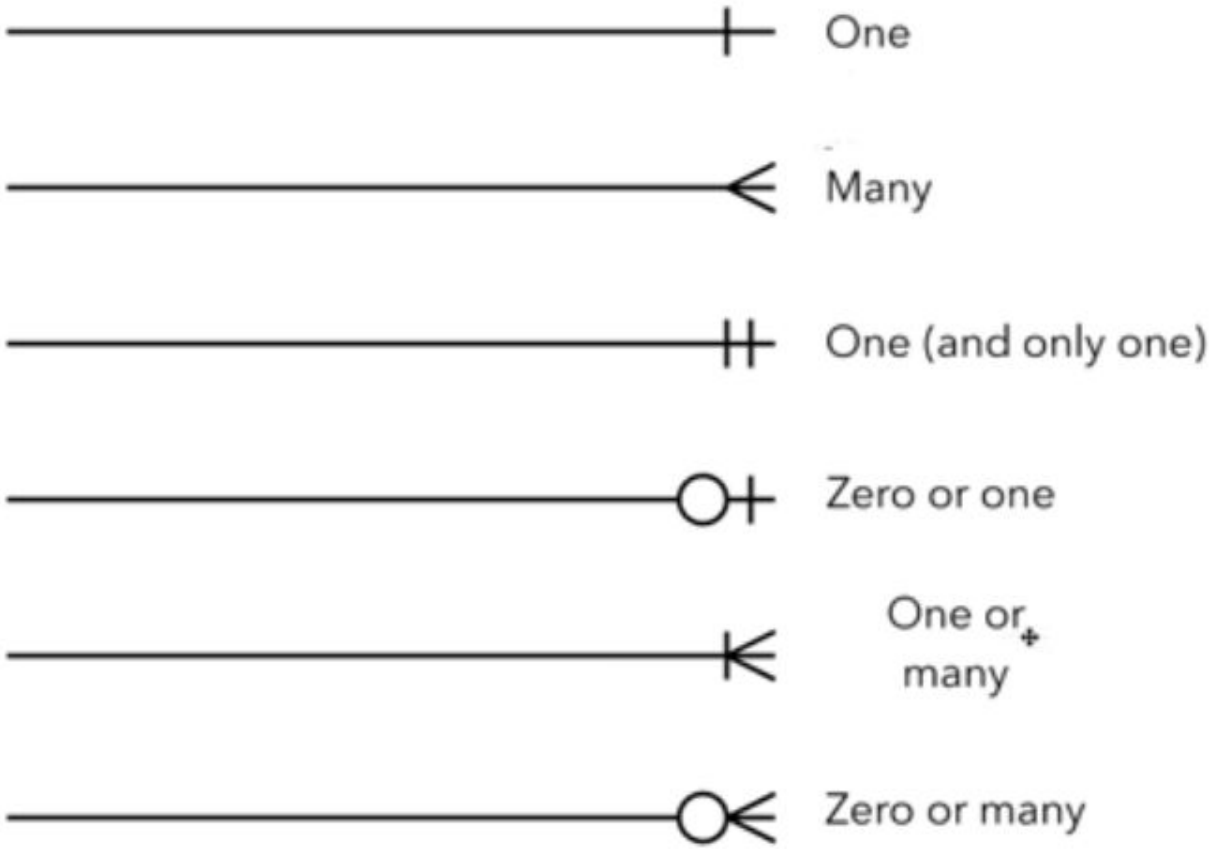


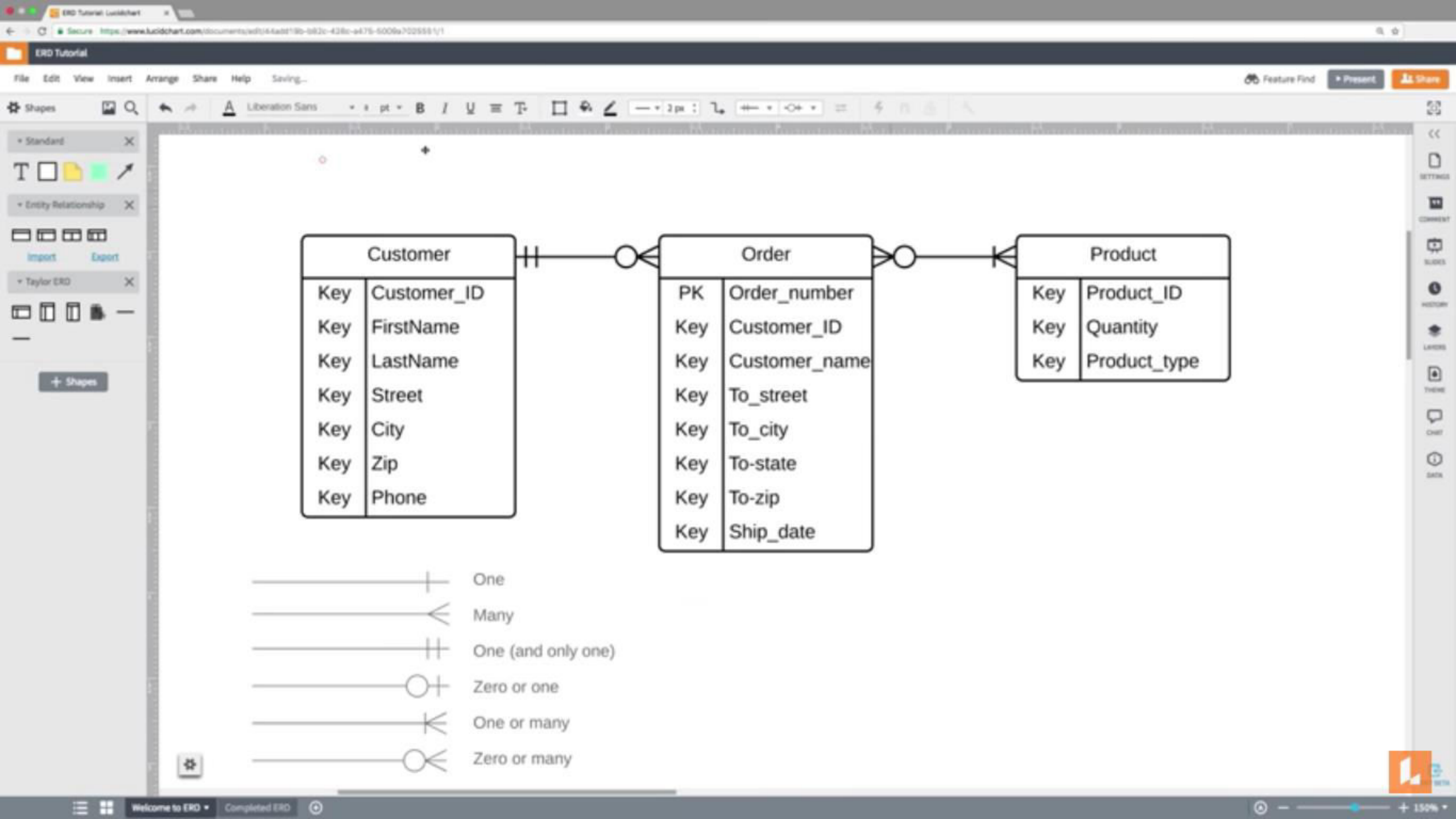
Subtypes

- **Subtype / category / specialization**
  - Well-defined subset of the occurrences of another entity type
    - That other entity type is called supertype, generic entity, or generalization
  - Use same symbol if:
    - Mutually exclusive
    - Same discriminator
  - Double bar = complete set of categories
  - Single parent



ERD Cardinality




ERD Tutorial
Customer
Key Customer_ID
Key FirstName
Key LastName
Key Street
Key City
Key Zip
Key Phone
Order
PK Order_number
Key Customer_ID
Key Customer_name
Key To_street
Key To_city
Key To-state
Key To-zip
Key Ship_date
Product
Key Product_ID
Key Quantity
Key Product_type
One
Many
One (and only one)
Zero or one
One or many
Zero or many

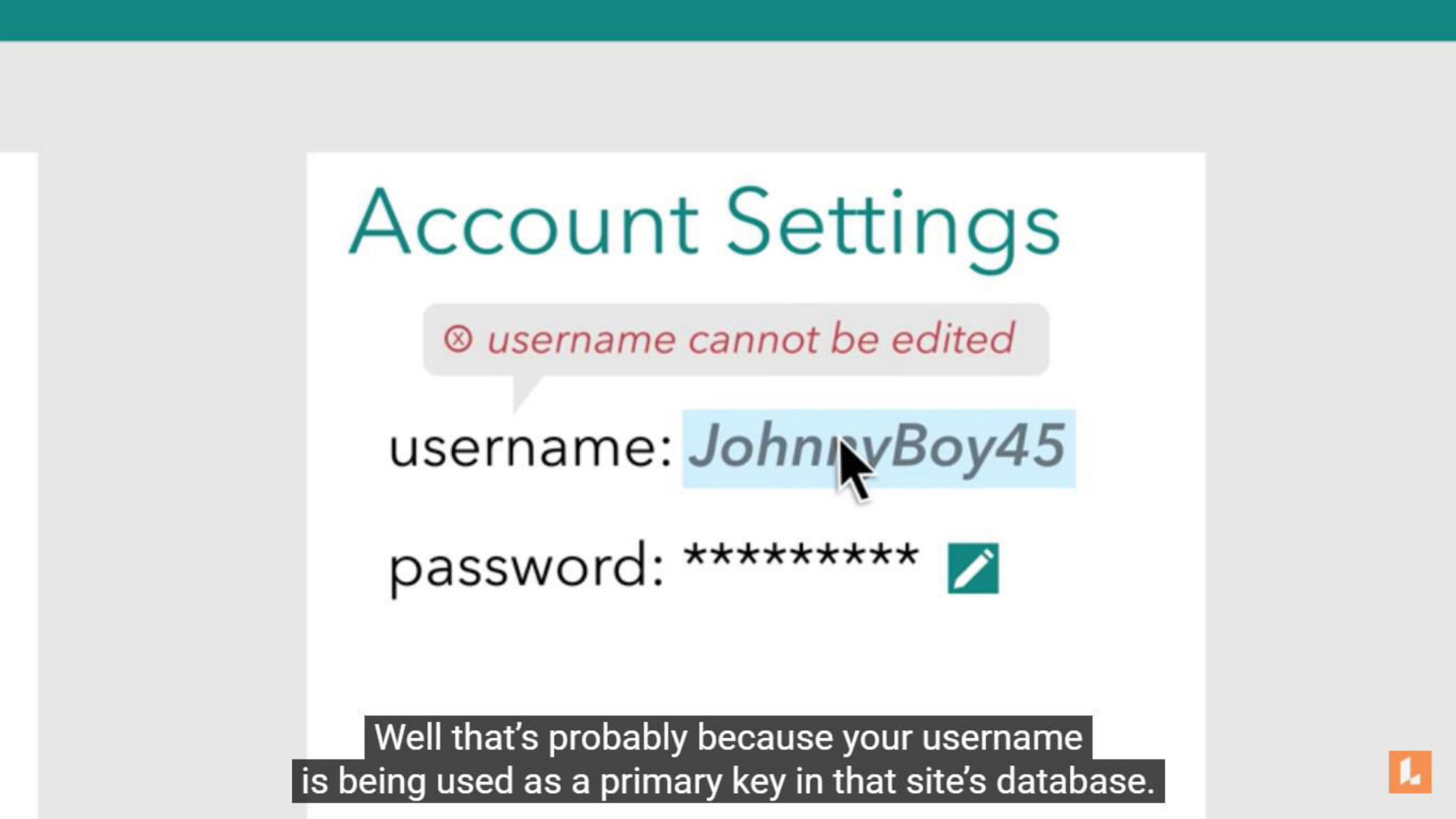
Customer Table

1	Customer_ID	FirstName	LastName	Street	City	Zip	Phone
12	30011	Linda	McGrath	7249 N. Bow Ridge St.	Ft Mitchell	23358	903-296-6663
13	30012	Iris	Edmunds	7135 North Rocky River Court	Yorktown	55720	728-849-9825
14	30013	Chandra	Parsons	847 Tanglewood Dr.	Calhoun	27759	821-271-9463
15	30014	Ranee	Peters	696 Fawn Court	Albany	97083	614-522-4822
16	30015	Steven	Langdon	64 Pennington Ave.	Jacksonville	33490	545-041-1643
17	30016	John	Smith	7411 Shirley Street	Springfield	41437	522-287-2538
18	30017	Ben	Chapman	6 James Ave.	Hopkinsville	30476	172-245-1141
19	30018	Jeremy	Nash	76 Strawberry Court	Billerica	70728	111-267-2814
20	30019	Rhett	Buckland	243 Mayflower St.	Watertown	97924	147-612-1745
21	30020	Carmen	Jones	8318 Mammoth Ave.	Lorton	48852	648-246-5531
22	30021	Marylynn	Smith	7411 Shirley Street	Springfield	41437	522-287-2538
23	30022	Dorothy	Taylor	845 South Bay Meadows Dr.	Trumbull	34485	544-720-3697
24	30023	Lena	Clarkson	50 Westport Rd.	Valparaiso	75622	487-800-7382
25	30024	Catheryn	Terry	921 Cardinal Court	Norwich	87483	632-495-0457
26	30025	Henry	Mackay	9927 Morris Ave.	Edison	62761	737-342-1771
27	30026	John	Smith	14 Lakewood Ave.	Centerville	31740	371-452-5023
28	30027	Irene	Ferguson	9100 N. Sleepy Hollow Street	Harlingen	47207	324-598-9629
29	30028	Erika	Knox	967 Summer Street	South Lyon	87143	
30	30029	Marvin	James	7536 Fairground Ave.	Pottstown	20063	
31	30030	Inez	Morrison	9 Shore St.	Wyandotte	74177	
32	30031	Mariann	[illegible]	[illegible]	[illegible]	[illegible]36	
33	30032	Sam	Duncan	9892 Dogwood Road	Sanford	58368	

John's Customer ID will completely identify him as a particular instance in our database,

Primary Key Rules

1. Unique
2. Never changing
3. Never null



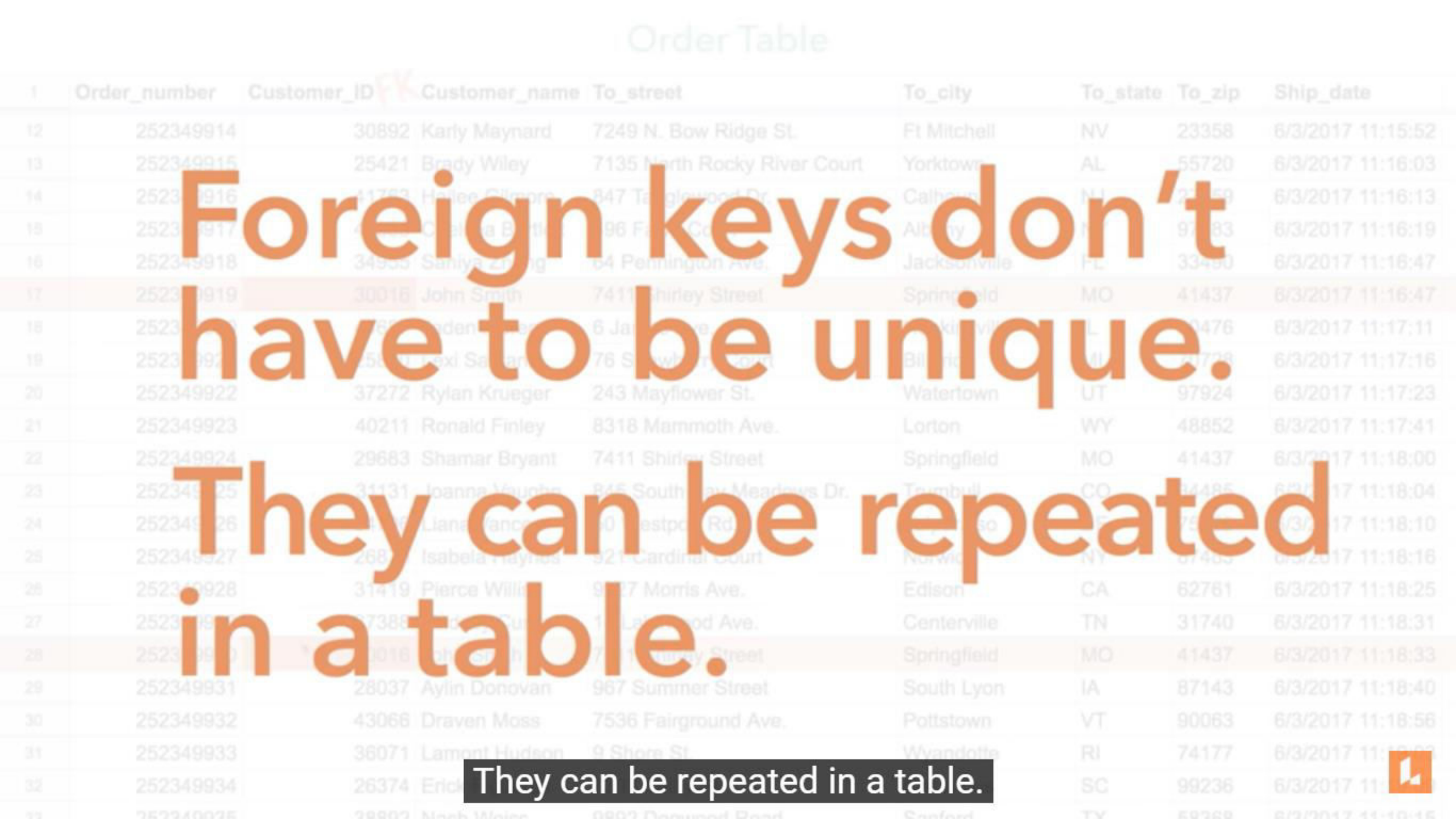
Well that's probably because your username
is being used as a primary key in that site's database.

Create Username

JohnSmith1

⊗ username is already in use

Again, this could be happening because the username is being used as the primary key,


Foreign keys don't have to be unique.
They can be repeated in a table.
They can be repeated in a table.

Composite Primary Key

Shipment Table

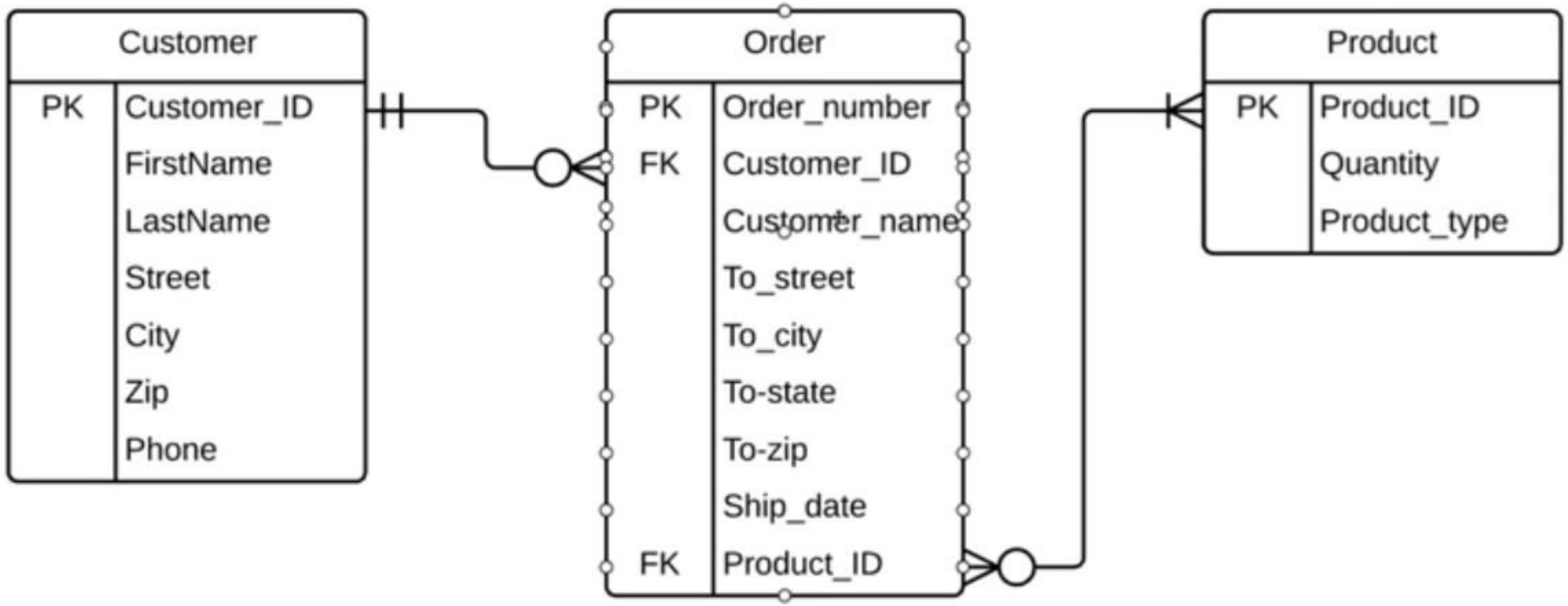
1	Product_ID	Order_number	ChargeCardTime	PackingTime	ShipOrderDate
13	49225	252349915	6/1/2017 9:13:34	6/2/2017 10:14:46	6/3/2017 11:15:52
14	40807	252349916	6/1/2017 9:14:16	6/2/2017 10:15:02	6/3/2017 11:16:03
15	76342	252349917	6/1/2017 9:14:01	6/2/2017 10:15:26	6/3/2017 11:16:13
16	96893	252349918	6/1/2017 9:14:21	6/2/2017 10:15:39	6/3/2017 11:16:19
17	69246	252349919	6/1/2017 9:14:34	6/2/2017 10:15:41	6/3/2017 11:16:47
18	69253	252349919	6/1/2017 9:14:34	6/2/2017 10:15:45	6/3/2017 11:16:47
19	99002	252349920	6/1/2017 9:15:07	6/2/2017 10:16:07	6/3/2017 11:17:11
20	64382	252349921	6/1/2017 9:15:14	6/2/2017 10:16:07	6/3/2017 11:17:23
21	91514	252349922	6/1/2017 9:15:33	6/2/2017 10:16:28	6/3/2017 11:17:23
22	64244	252349923	6/1/2017 9:15:33	6/2/2017 10:16:50	6/3/2017 11:17:41
23	94251	252349924	6/1/2017 9:16:17	6/2/2017 10:17:05	6/3/2017 11:18:00
24	69253	252349925	6/1/2017 9:16:21	6/2/2017 10:17:17	6/3/2017 11:18:04
25	94166	252349926	6/1/2017 9:16:38	6/2/2017 10:17:29	6/3/2017 11:18:10
26	44199	252349927	6/1/2017 9:16:41	6/2/2017 10:17:30	6/3/2017 11:18:16
27	40759	252349928	6/1/2017 9:16:49	6/2/2017 10:17:44	6/3/2017 11:18:25
28	39668	25			8:31
29	71292	25			18:43

You could technically call this a compound key because we're using two foreign keys,

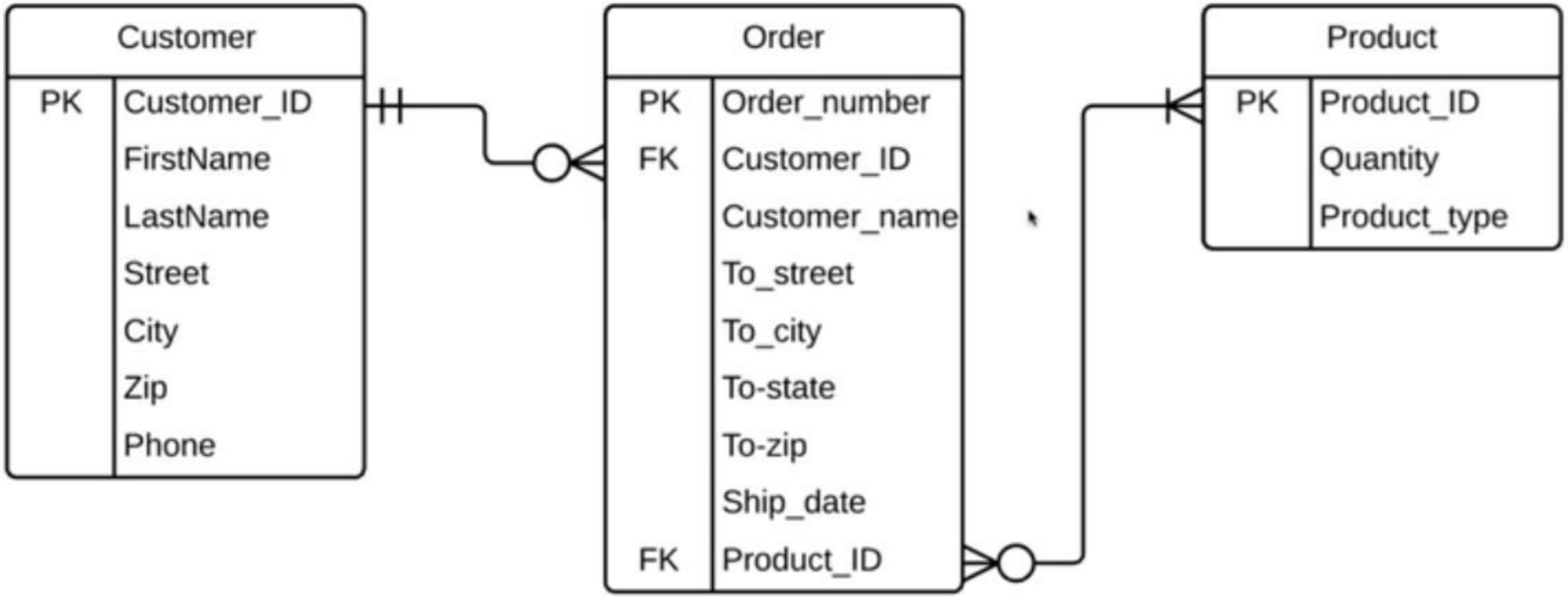
Composite Primary Key Rules

1. Use the fewest number of attributes possible.
2. Don’t use attributes that are apt to change.

Two, don’t use attributes that are apt to change because that can make things messy.



purchases a product, we're going to have
a record of that interaction in our Order table.



getting all necessary information and use bridge tables to capture that data.