

CSC 261/461 – Database Systems

Lecture 7

Spring 2018

Announcement

- Project 1 Milestone 1
 - is due Tonight
- Project 2 Part 1:
 - will be out tonight!
- You should definitely go to workshops this week
 - Workshop on Project 2
 - You will be 50% done with project 2 just by visiting the workshop
 - Bring your own laptop!
 - It will also cover queries on the quiz.

The ER (Entity-Relationship) Model

Agenda

1. High-level motivation for the E/R model
2. Entities
3. Relations
4. E/R Model

Database Design

- Database design: Why do we need it?
 - Agree on structure of the database before deciding on a particular implementation
- Consider issues such as:
 - What entities to model
 - How entities are related
 - What constraints exist in the domain
 - How to achieve good designs
- Several formalisms exist
 - We discuss one flavor of E/R diagrams

Database Design Process

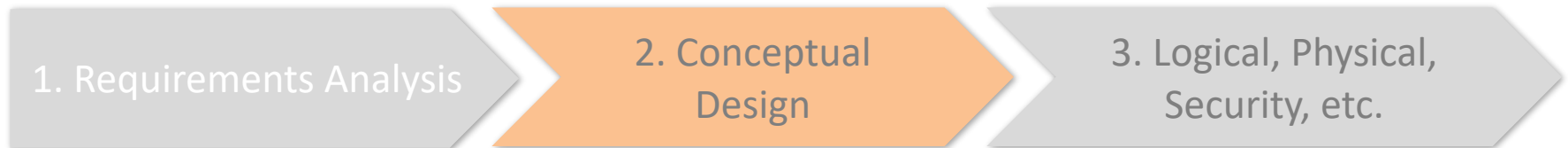


1. Requirements analysis

- What is going to be stored?
- How is it going to be used?
- What are we going to do with the data?
- Who should access the data?

Technical and non-technical people are involved

Database Design Process

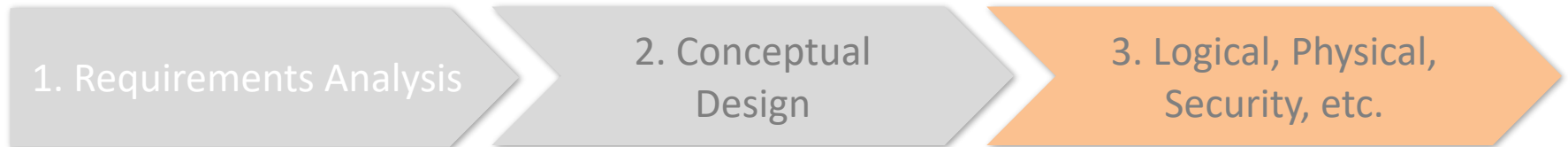


2. Conceptual Design

- A high-level description of the database
- Sufficiently precise that technical people can understand it
- But, not so precise that non-technical people can't participate

This is where E/R fits in.

Database Design Process



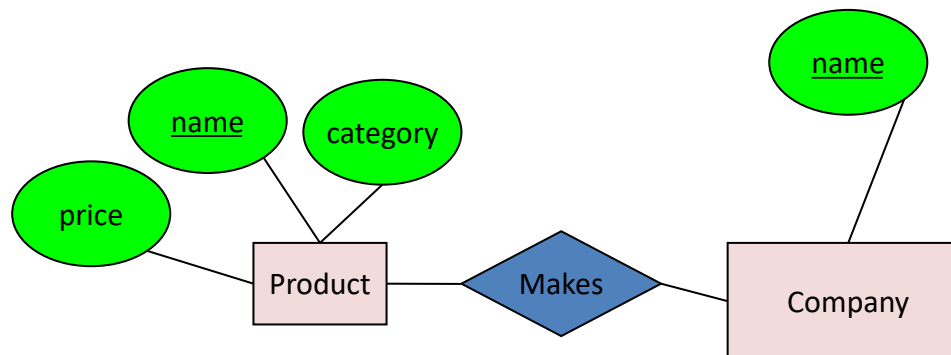
3. Implementation:

- Logical Database Design
- Physical Database Design
- Security Design

Database Design Process



E/R Model & Diagrams used



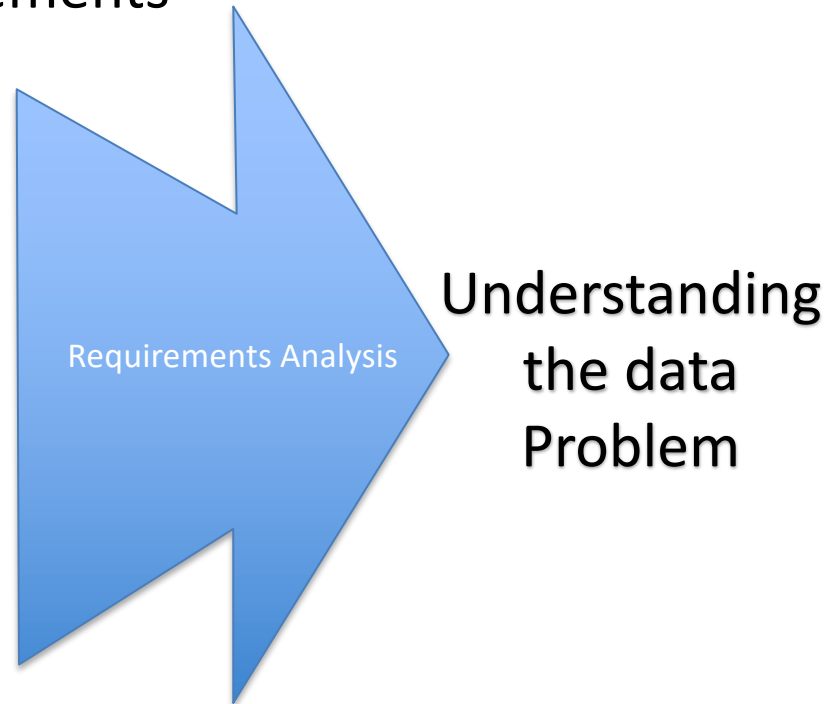
This process is iterated **many** times

E/R is a *visual syntax* for DB design which is ***precise enough*** for technical people, but ***abstracted enough*** for non-technical people

The Requirements Analysis Stage

- Sources of requirements

- User Interviews
- Forms
- Reports
- Queries
- Use Cases
- Observation



Requirements Become the E-R Data Model

- After the requirements have been gathered, they are transformed into an Entity Relationship (E-R) Data Model
- E-R Models consist of
 1. Entities
 2. Attributes
 - a) Identifiers (Keys)
 - b) Non-key attributes
 3. Relationships

1. E/R BASICS: ENTITIES & RELATIONS

Interlude: Impact of the ER model

- The E/R model is one of the most cited articles in Computer Science
 - Author: Peter Pin-Shan Chen

The entity-relationship model—toward a unified view of data

PPS Chen - ACM Transactions on Database Systems (TODS), 1976 - dl.acm.org

Abstract A data model, called the entity-relationship model, is proposed. This model incorporates some of the important semantic information about the real world. A special diagrammatic technique is introduced as a tool for database design. An example of

[Cited by 10395](#) [Related articles](#) [All 97 versions](#) [Cite](#) [Save](#) [More](#)



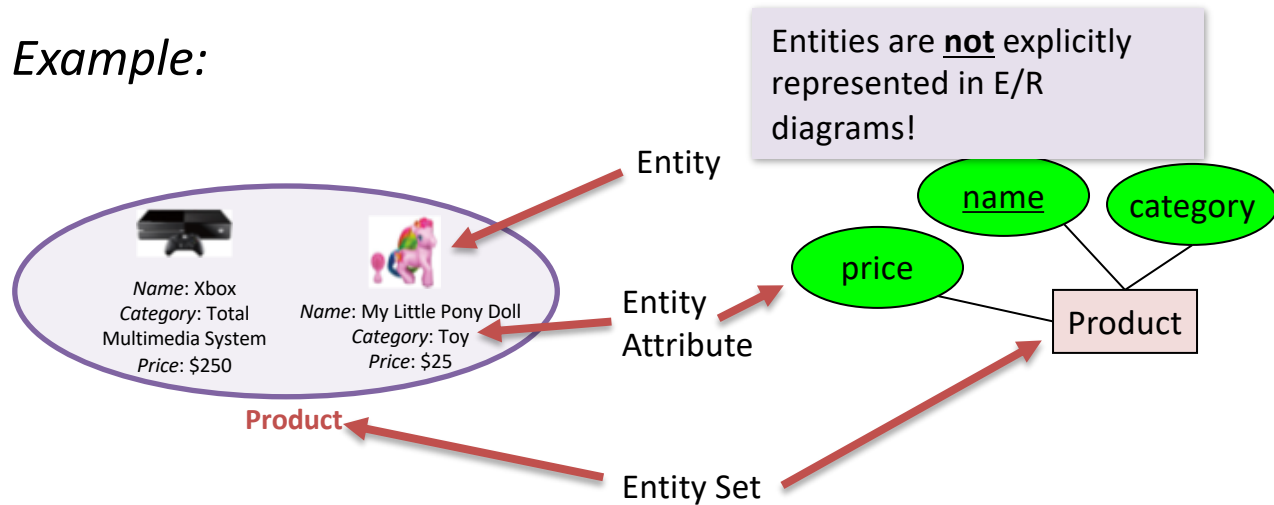
From Google Scholar

Entities and Entity Sets

- **Entities** & **entity sets** are the primitive unit of the E/R model
 - Entities or (entity instances) are the individual objects, which are members of entity sets
 - Ex: A specific person or Item
 - Entity sets or (entity classes) are the *classes* or *types* of objects in our model
 - Ex: Person, Item
 - *These are what is shown in E/R diagrams - as rectangles*
 - *Entity sets represent the sets of all possible entities*

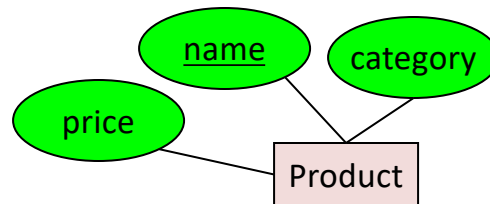
Entities vs. Entity Sets

Example:



Entities and Entity Sets

- An entity set has **attributes**
 - Represented by ovals
attached to an entity set

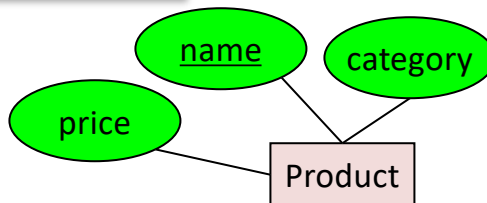


Shapes are important. Colors are not.

Keys

- A key is a **minimal** set of attributes that uniquely identifies an entity.

Denote elements of the primary key by underlining.



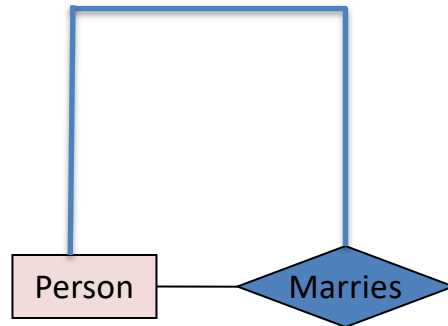
Here, {name, category} is not a key (it is not *minimal*).

The E/R model forces us to designate a single **primary** key, though there may be multiple candidate keys

Relationships

- A **relationship** connects two or more entity sets.
- It is represented by a **diamond**, with lines to each of the entity sets involved.
- The **degree** of the relationship defines the number of entity classes that participate in the relationship
 - Degree 1 is a **unary** relationship
 - Degree 2 is a **binary** relationship
 - Degree 3 is a **ternary** relationship

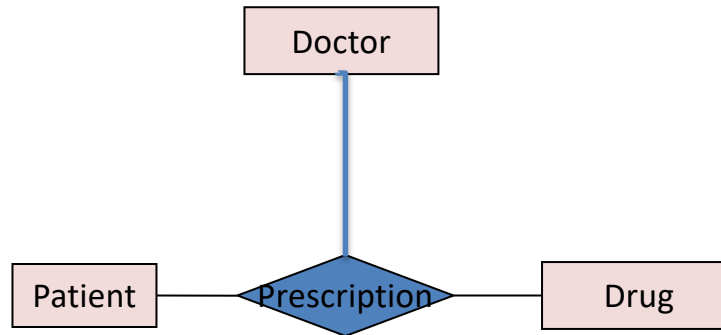
Conceptual Unary Relationship



Conceptual Binary Relationship

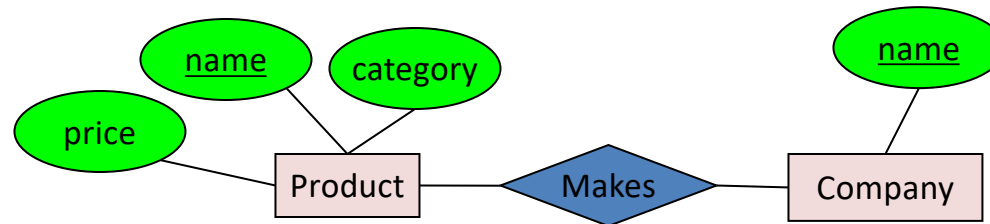


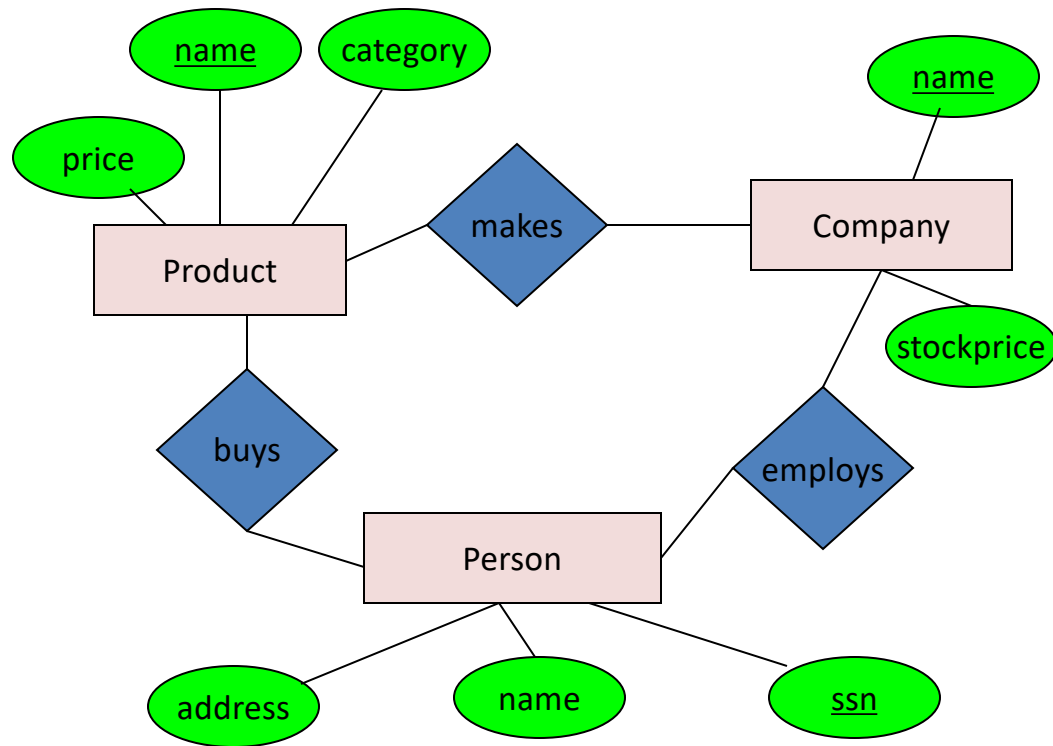
Conceptual Ternary Relationship



The R in E/R: Relationships

- E, R and A together:



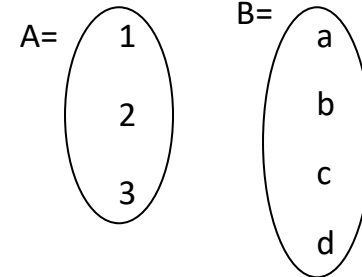


What is a Relationship?

- ***A mathematical definition:***

- Let A, B be sets

- $A=\{1,2,3\}, \quad B=\{a,b,c,d\}$



What is a Relationship?

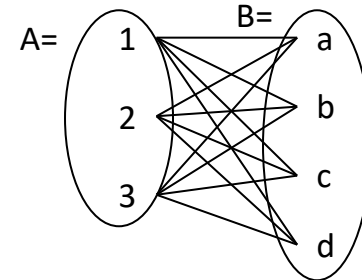
- ***A mathematical definition:***

- Let A, B be sets

- $A=\{1,2,3\}$, $B=\{a,b,c,d\}$

- $A \times B$ (the ***cross-product***) is the set of all pairs (a,b)

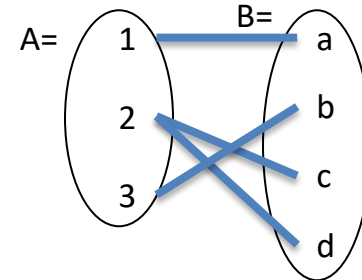
- $A \times B = \{(1,a), (1,b), (1,c), (1,d), (2,a), (2,b), (2,c), (2,d), (3,a), (3,b), (3,c), (3,d)\}$



What is a Relationship?

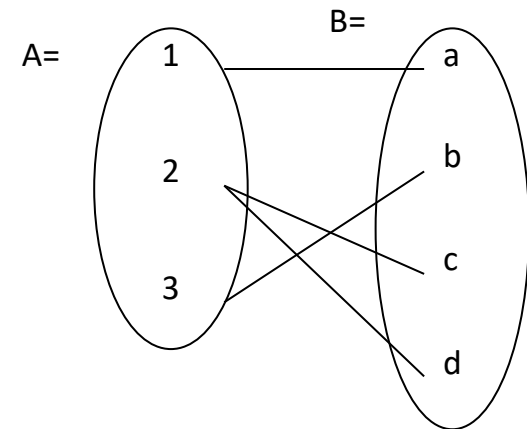
- ***A mathematical definition:***

- Let A, B be sets
 - $A=\{1,2,3\}$, $B=\{a,b,c,d\}$,
- $A \times B$ (the ***cross-product***) is the set of all pairs (a,b)
 - $A \times B = \{(1,a), (1,b), (1,c), (1,d), (2,a), (2,b), (2,c), (2,d), (3,a), (3,b), (3,c), (3,d)\}$
- We define a relationship to be a subset of $A \times B$
 - $R = \{(1,a), (2,c), (2,d), (3,b)\}$

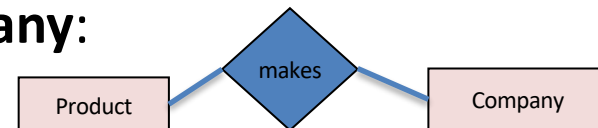


What is a Relationship?

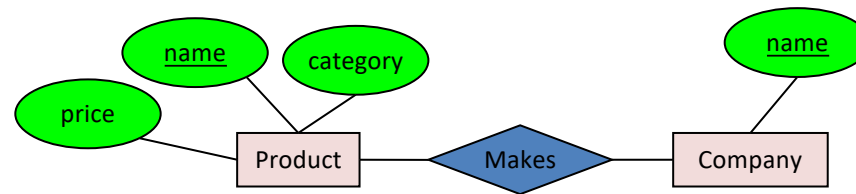
- ***A mathematical definition:***
 - Let A, B be sets
 - $A \times B$ (the ***cross-product***) is the set of all pairs
 - A relationship is a subset of $A \times B$



- **Makes** is relationship- it is a ***subset*** of **Product** \times **Company**:



What is a Relationship?



A **relationship** between entity sets **P** and **C** is a ***subset of all possible pairs of entities in P and C,*** with tuples uniquely identified by ***P and C's keys***

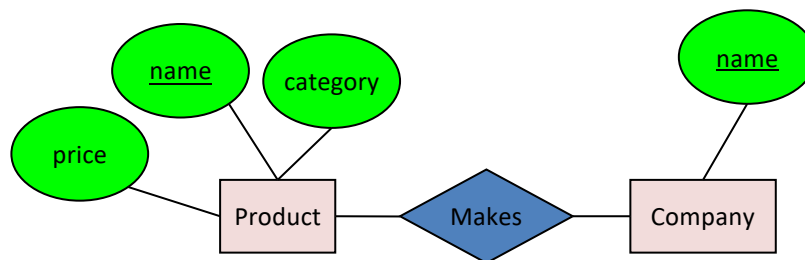
What is a Relationship?

Company

<u>name</u>
GizmoWorks
GadgetCorp

Product

<u>name</u>	category	price
Gizmo	Electronics	\$9.99
GizmoLite	Electronics	\$7.50
Gadget	Toys	\$5.50



A **relationship** between entity sets **P** and **C** is a **subset of all possible pairs of entities in P and C**, with tuples uniquely identified by **P and C's keys**

What is a Relationship?

Company

<u>name</u>
GizmoWorks
GadgetCorp

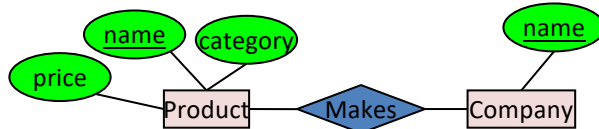
Product

<u>name</u>	category	price
Gizmo	Electronics	\$9.99
GizmoLite	Electronics	\$7.50
Gadget	Toys	\$5.50



Company C × Product P

<u>C.name</u>	<u>P.name</u>	P.category	P.price
GizmoWorks	Gizmo	Electronics	\$9.99
GizmoWorks	GizmoLite	Electronics	\$7.50
GizmoWorks	Gadget	Toys	\$5.50
GadgetCorp	Gizmo	Electronics	\$9.99
GadgetCorp	GizmoLite	Electronics	\$7.50
GadgetCorp	Gadget	Toys	\$5.50



A **relationship** between entity sets **P** and **C** is a **subset of all possible pairs of entities in P and C**, with tuples uniquely identified by **P and C's keys**

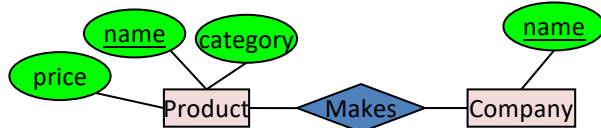
What is a Relationship?

Company

<u>name</u>
GizmoWorks
GadgetCorp

Product

<u>name</u>	category	price
Gizmo	Electronics	\$9.99
GizmoLite	Electronics	\$7.50
Gadget	Toys	\$5.50



A **relationship** between entity sets **P** and **C** is a **subset of all possible pairs of entities in P and C**, with tuples uniquely identified by **P and C's keys**



Company C × Product P

<u>C.name</u>	<u>P.name</u>	P.category	P.price
GizmoWorks	Gizmo	Electronics	\$9.99
GizmoWorks	GizmoLite	Electronics	\$7.50
GizmoWorks	Gadget	Toys	\$5.50
GadgetCorp	Gizmo	Electronics	\$9.99
GadgetCorp	GizmoLite	Electronics	\$7.50
GadgetCorp	Gadget	Toys	\$5.50

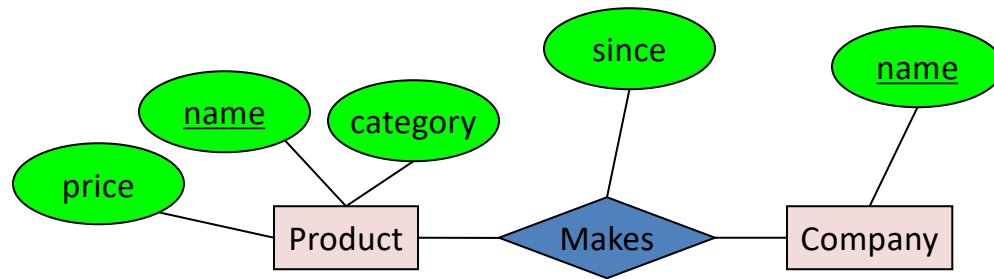


Makes

<u>C.name</u>	<u>P.name</u>
GizmoWorks	Gizmo
GizmoWorks	GizmoLite
GadgetCorp	Gadget

Relationships and Attributes

- Relationships may have attributes as well.



For example: “since” records when company started making a product

Note: “*since*” is implicitly unique per pair here! Why?

TYPES OF RELATIONSHIP

Conceptual Crow's Foot Relationship Symbols

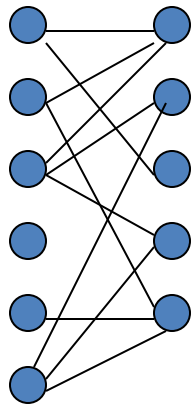
One —————

Many —————>

Many-Many Relationships

- Focus: **binary** relationships, such as **Sells** between **Seller** and **Buyer**.
- In a **many-many relationship**, an entity of either set can be connected to many entities of the other set.
 - E.g., a seller sells many items; a buyer can buy many items too.

In Pictures:



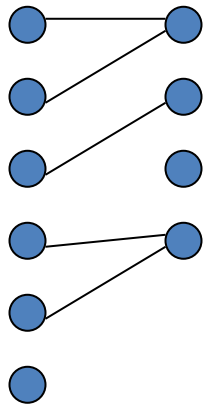
many-many



Many-One Relationships

- Some binary relationships are *many -one* from one entity set to another.
- Each entity of the first set is connected to at most one entity of the second set.
- But an entity of the second set can be connected to zero, one, or many entities of the first set.
- E.g.: One buyer can have *multiple order* number, but one order can be bought by *only one buyer*.

In Pictures:



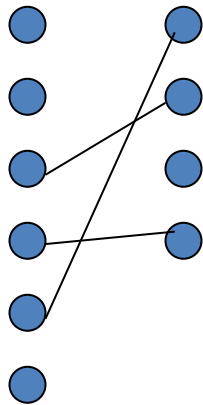
many-one



One-One Relationships

- In a *one-one relationship*, each entity of either entity set is related to at most one entity of the other set.
- **Example:** Relationship *president* between entity *country* and *person*.
 - A person can be the president of only one country.
One country can have only one president.

In Pictures



Maximum Cardinality

- Relationships are named and classified by their **cardinalities**, which is a word that means *count* (as in the number of items in a set)
- Each of the three types of binary relationship shown previously has a different *maximum cardinality*
- Maximum cardinality is the **maximum** number of entity instances that can participate in a relationship instance
 - One, many, or some other positive fixed number

Minimum Cardinality

- **Minimum cardinality** is the minimum number of entity instances that must participate in a relationship instance
- These values typically assume a value of zero (optional) or one (mandatory)

Crow's Foot Symbols with Cardinalities

One-and-Only-One —||—

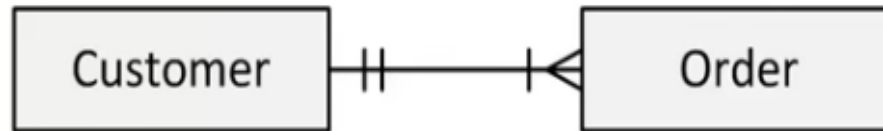
One-to-Many —|—<

Zero-to-One —○|—

Zero-to-Many —○—<

Cardinality Example

- Maximum cardinality is many for Order and one for Customer
- Minimum cardinality is one for both Customer and Order
 - Each customer can place one or more orders
 - Each order is associated with one and only one customer



Entity-Relationship Diagrams

- The diagrams in previous slides are called entity-relationship diagrams
 - Entities represented by rectangles
 - Relationships represented by lines
 - Cardinalities represented by Crow's Foot symbols

HAS-A Relationships

- The relationships in the previous slides are called HAS-A relationships
- The term is used because each entity instance *has a* relationship to a second entity instance
 - An employee has a locker
 - A locker has an employee
- There are also IS-A relationships

DRAW AN E/R DIAGRAM FOR FOOTBALL

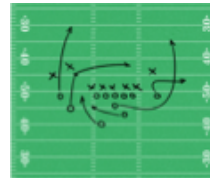
Use the following simplified model of a football season
(concepts to include are underlined):



Teams play each other in Games.
Each pair of teams can play each other multiple times



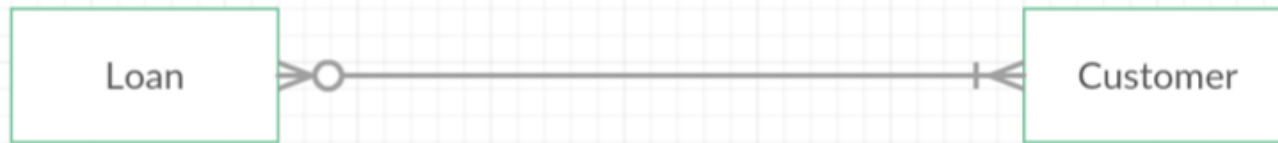
Players belong to Teams



A Game is made up of Plays that result in a yardage gain/loss, and potentially a touchdown



A Play will contain either a Pass from one player to another, or a Run by one player



Acknowledgement

- Some of the slides in this presentation are taken from the slides provided by the authors.
- Many of these slides are taken from cs145 course offered by Stanford University.