L08: ER modeling

CS3200 Database design (sp18 s2)

2/5/2018

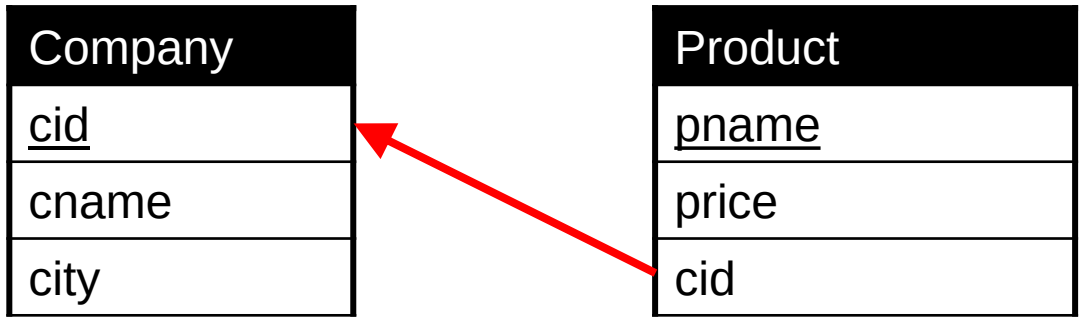
Announcements!

- Jupyter: setup on your laptops, we start using next week
- HW3 solutions: posted earlier (no FMs)
- Confidential or anonymous questions on HW?
  - please post on Piazza "visible to instructors only"
- Anonymous question to instructor only: Google feedback form
- Exam1 (Mon Feb 12): Laptop, BlackBoard, Postgres, SQL only. Vote:
  - Variant 1: closed book, one letter cheatsheet allowed
  - Variant 2: open book, more time-constrained, graded very carefully

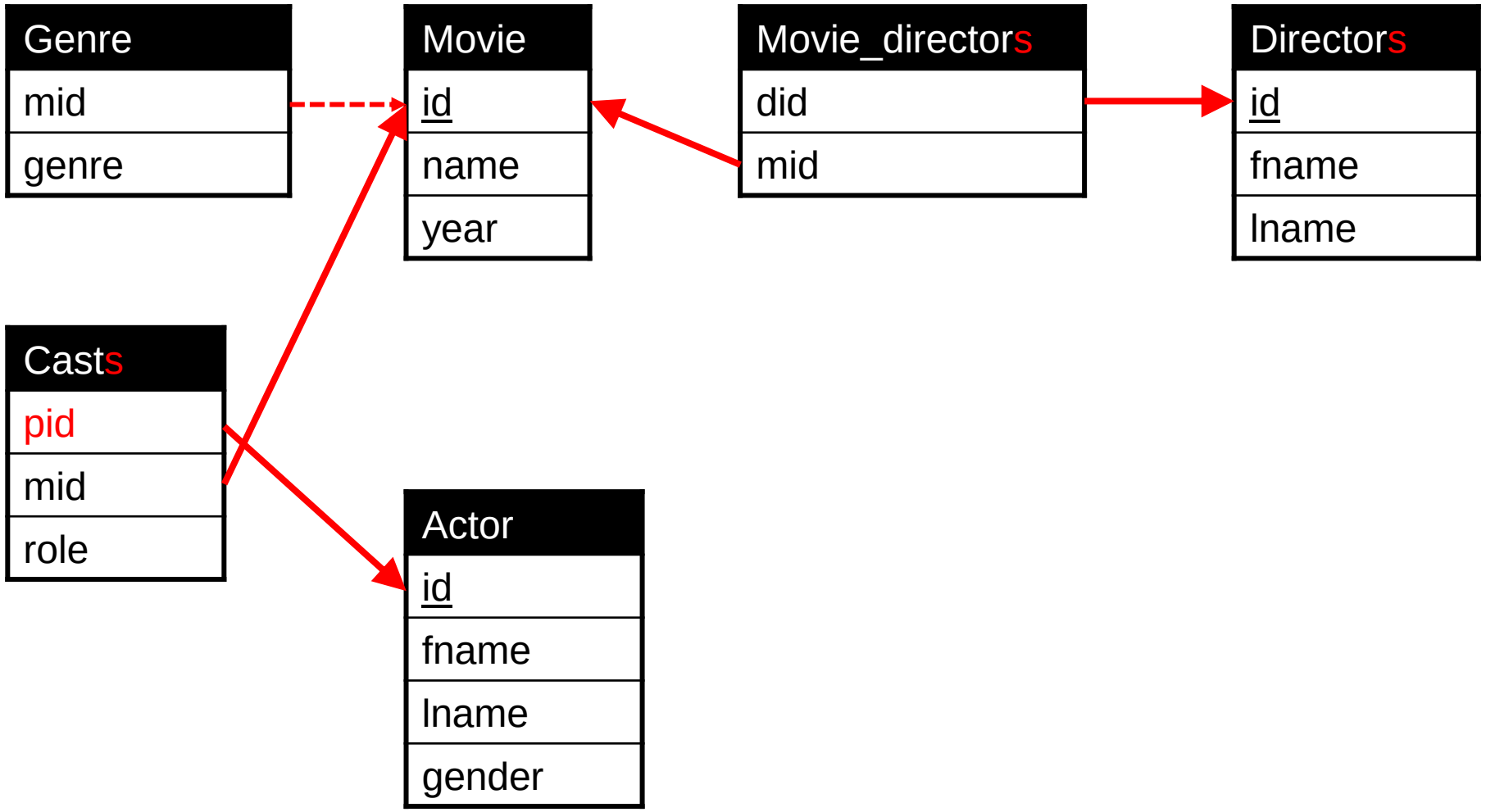
- Outline
  - HW2
  - ER modeling

HW2

Company/Product



Big IMDB schema (Postgres)

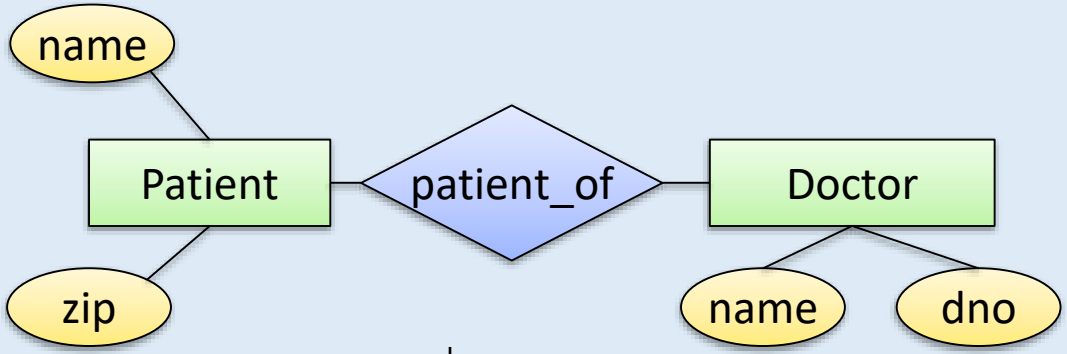


ER modeling

Data modeling and Database Design Process

1. ER Diagram

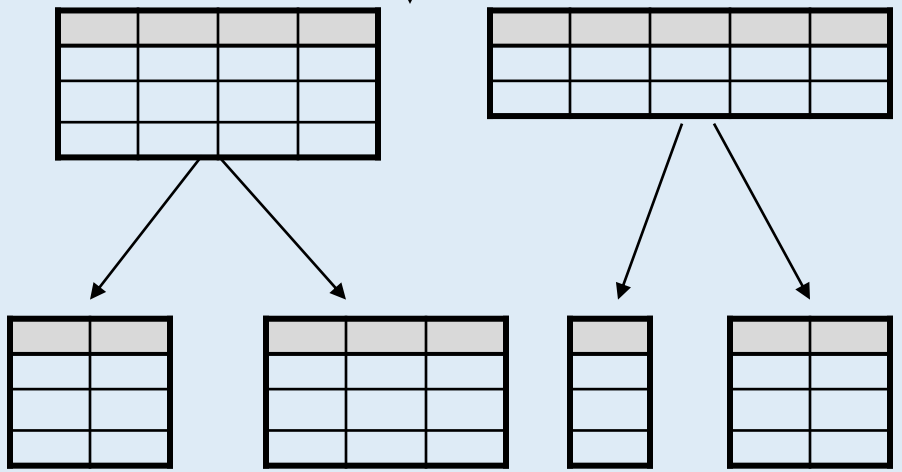
Conceptual Model:
("technology independent")
describe main data items



2. Relational Database Design

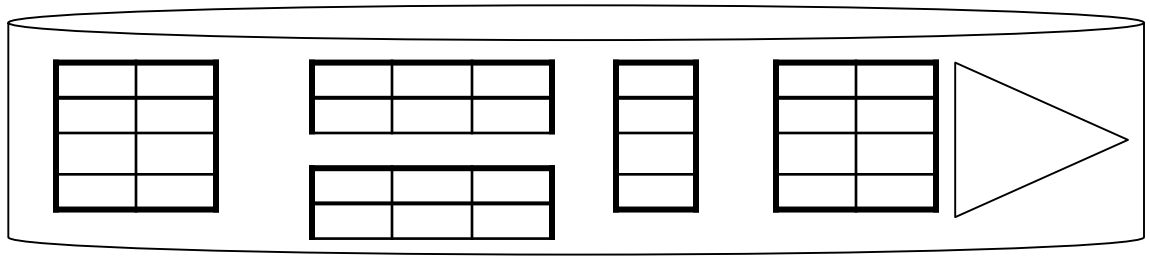
Logical Model
("for relational databases"):
Tables, Constraints
Functional Dependencies

Normalization:
Eliminates anomalies



3. Database Implementation

Physical Model
Physical storage details
Result: Physical Schema



Database Design

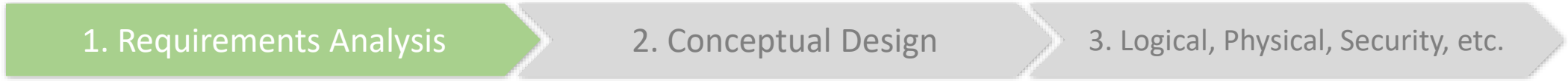
- Database design: Why do we need it?
  - Agree on structure of the database before deciding on a particular implementation

- Consider issues such as:
  - What entities to model
  - How entities are related
  - What constraints exist in the domain
  - How to achieve good designs

This the first project

- Several formalisms exist
  - We discuss two flavors of E/R diagrams
    - Chen notation: Stanford GUW book
    - Crow feet notation

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

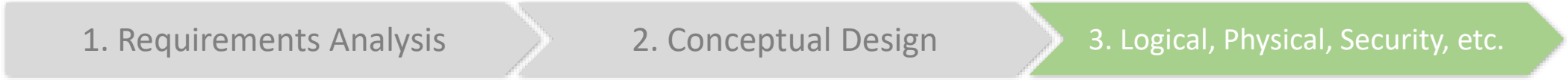
1. Requirements Analysis → 2. Conceptual Design → 3. Logical, Physical, Security, etc.

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

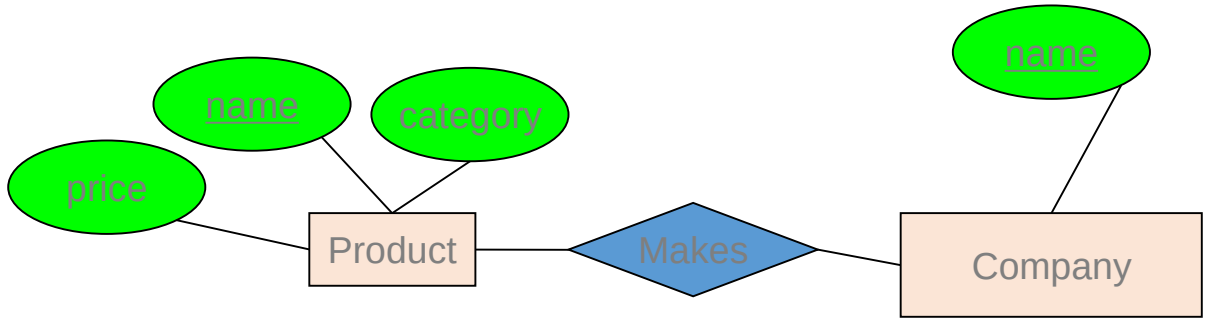


3. More:
   - Logical Database Design
   - Physical Database Design
   - Security Design

Database Design Process

1. Requirements Analysis → 2. Conceptual Design → 3. Logical, Physical, Security, etc.

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 points, but ***abstracted enough*** for non-technical people

Interlude: Impact of the ER model

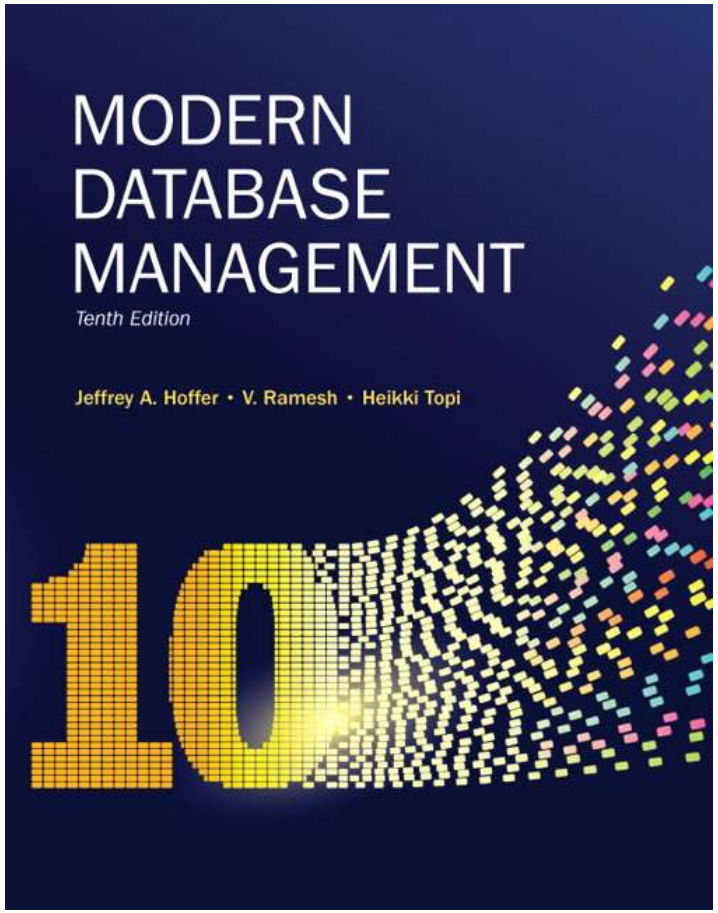
- The E/R model is one of the most cited articles in Computer Science
  - “The Entity-Relationship model – toward a unified view of data” Peter Chen, 1976

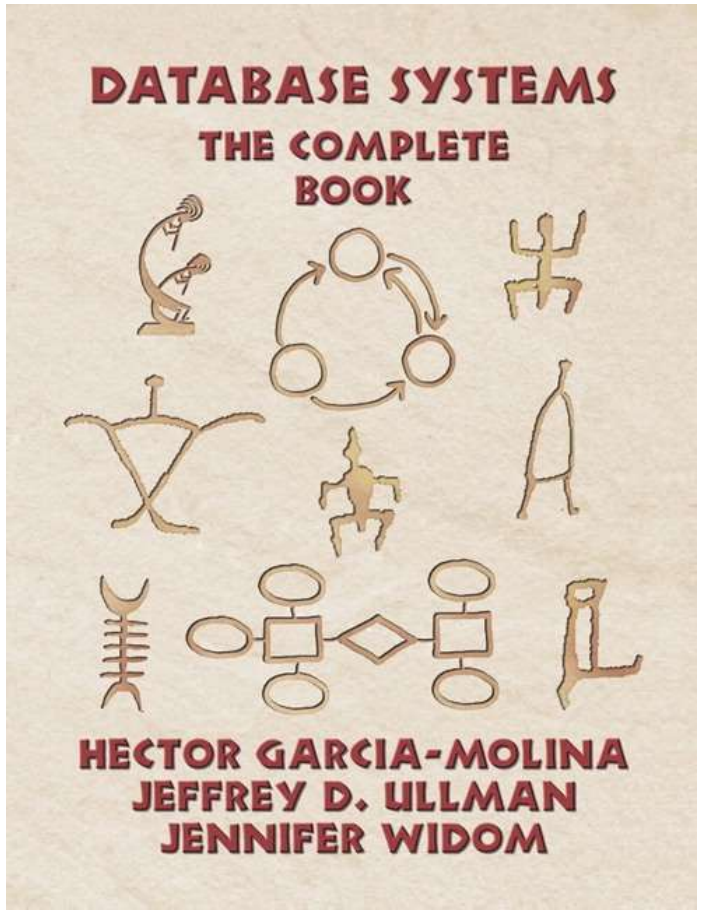
- Used by companies big and small
  - You’ll know it soon enough

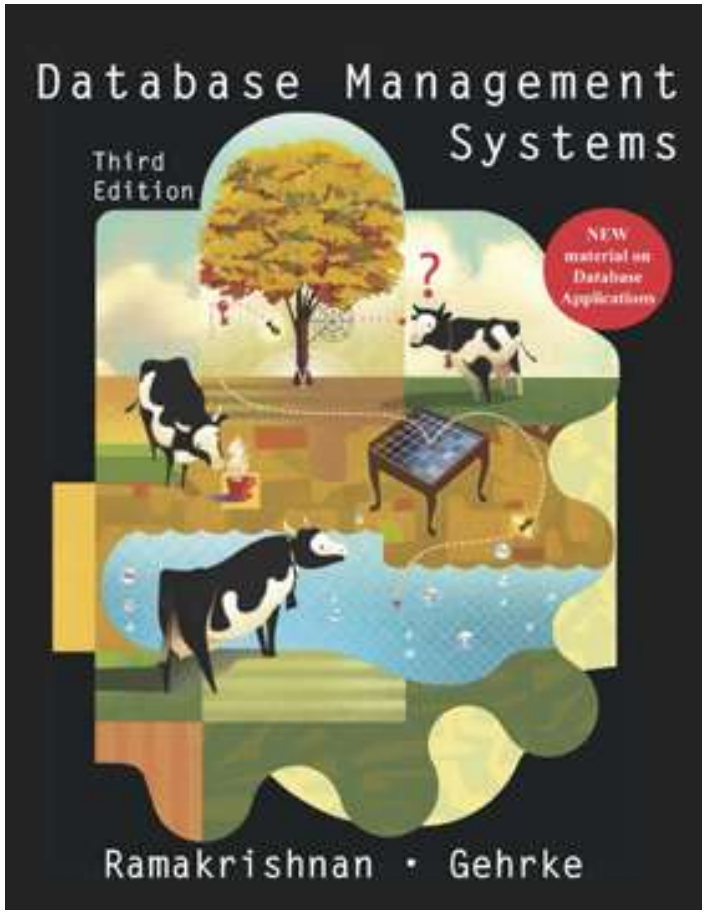
- "Chen notation": different from "UML"

Some comments on Notations

Different sources, different notations

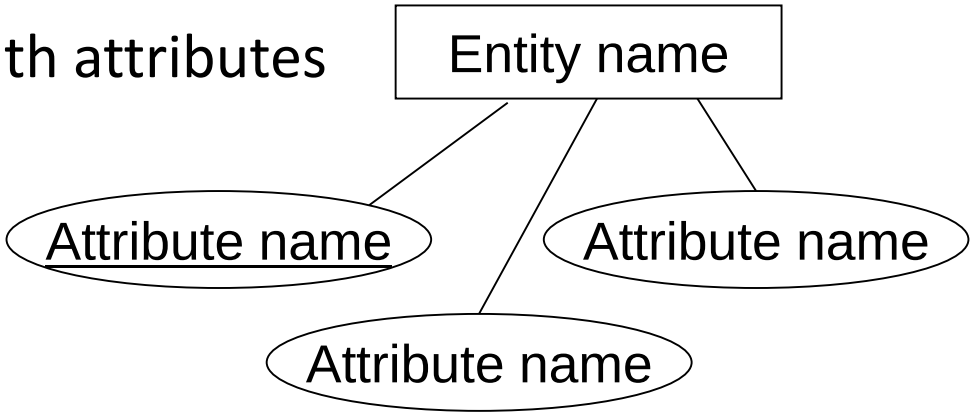
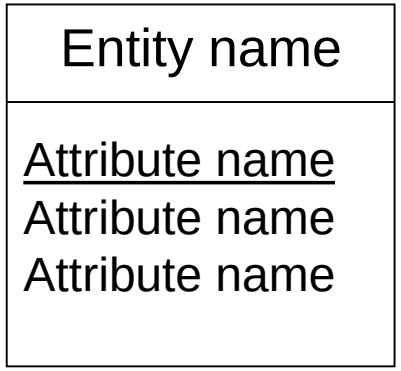






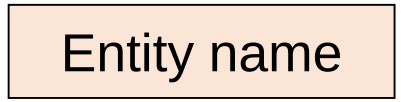
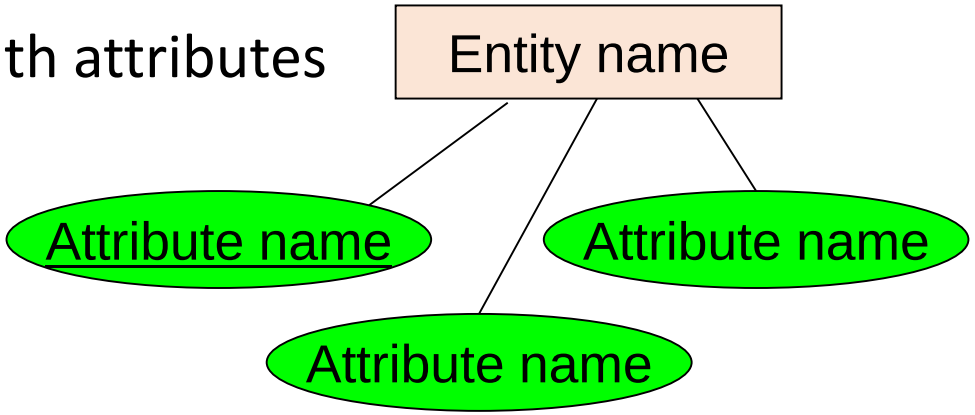
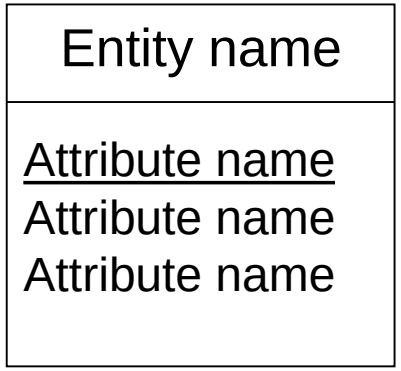
Comparison of ERD frameworks

A variant of "UML"

	Chen's	Crow's Feet
Strong entity		
Weak entity		
Entity with attributes		

Comparison of ERD frameworks

A variant of "UML"

	Chen's	Crow's Feet
Strong entity		
Weak entity		
Entity with attributes		

Color is not part of the standard…

Attributes

Chen's

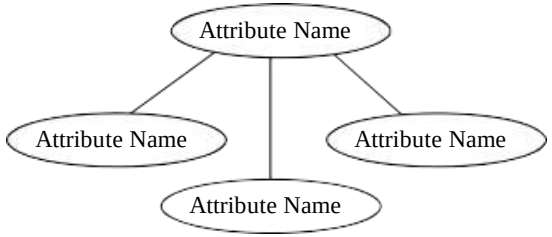
Attribute

PK attribute

Derived attribute

Multi-valued attribute

Composite attribute



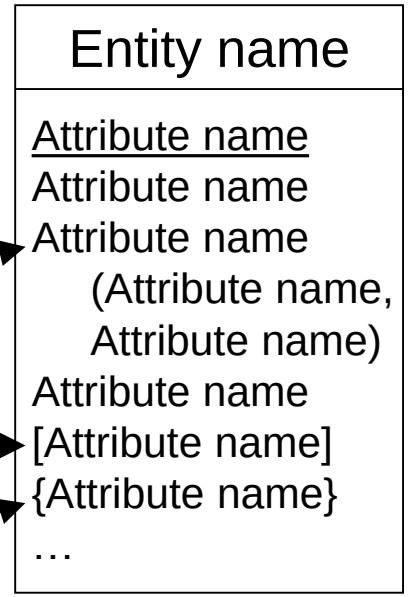
Crow's Feet

PK attribute

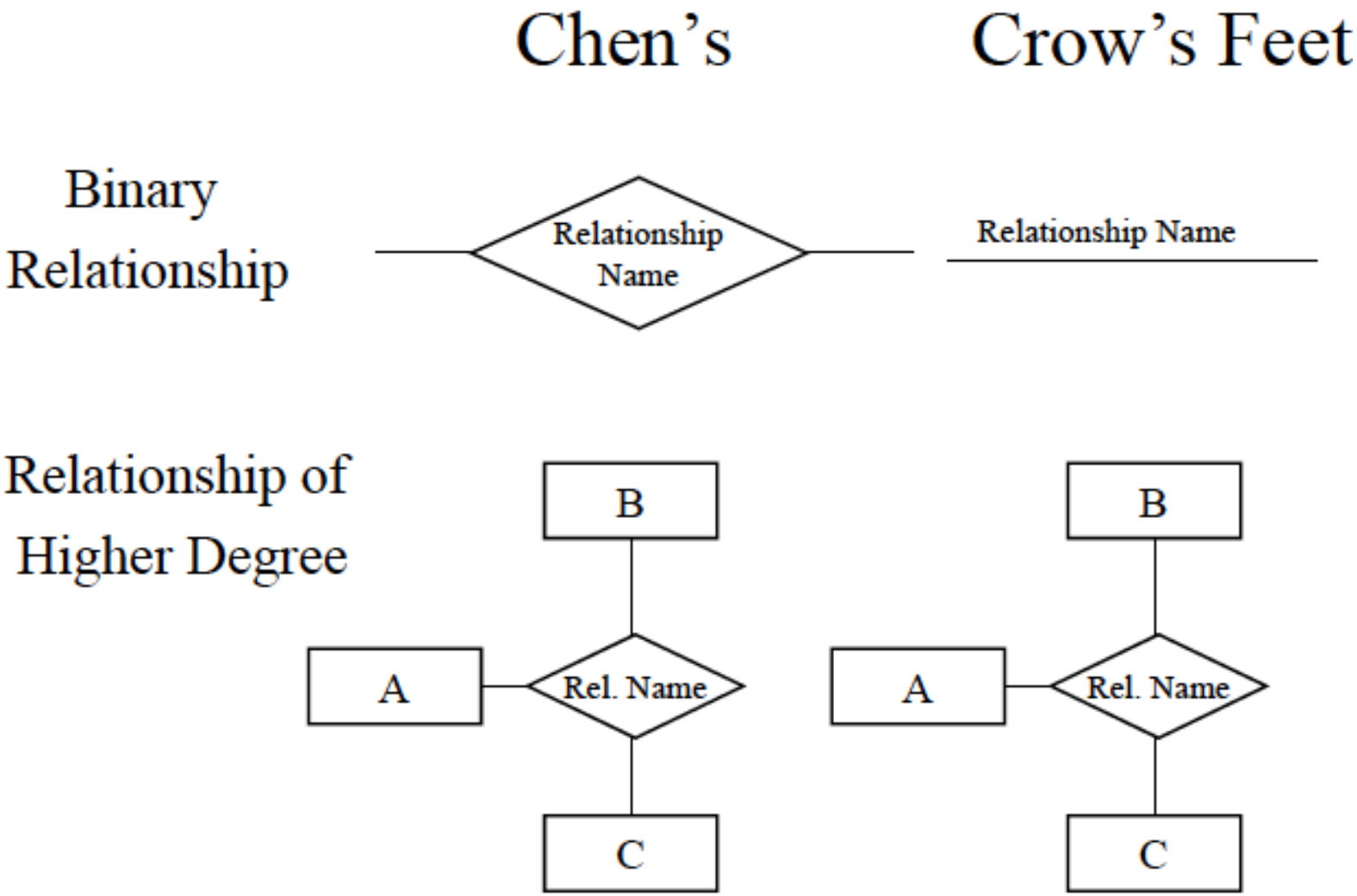
Composite attribute

Derived attribute

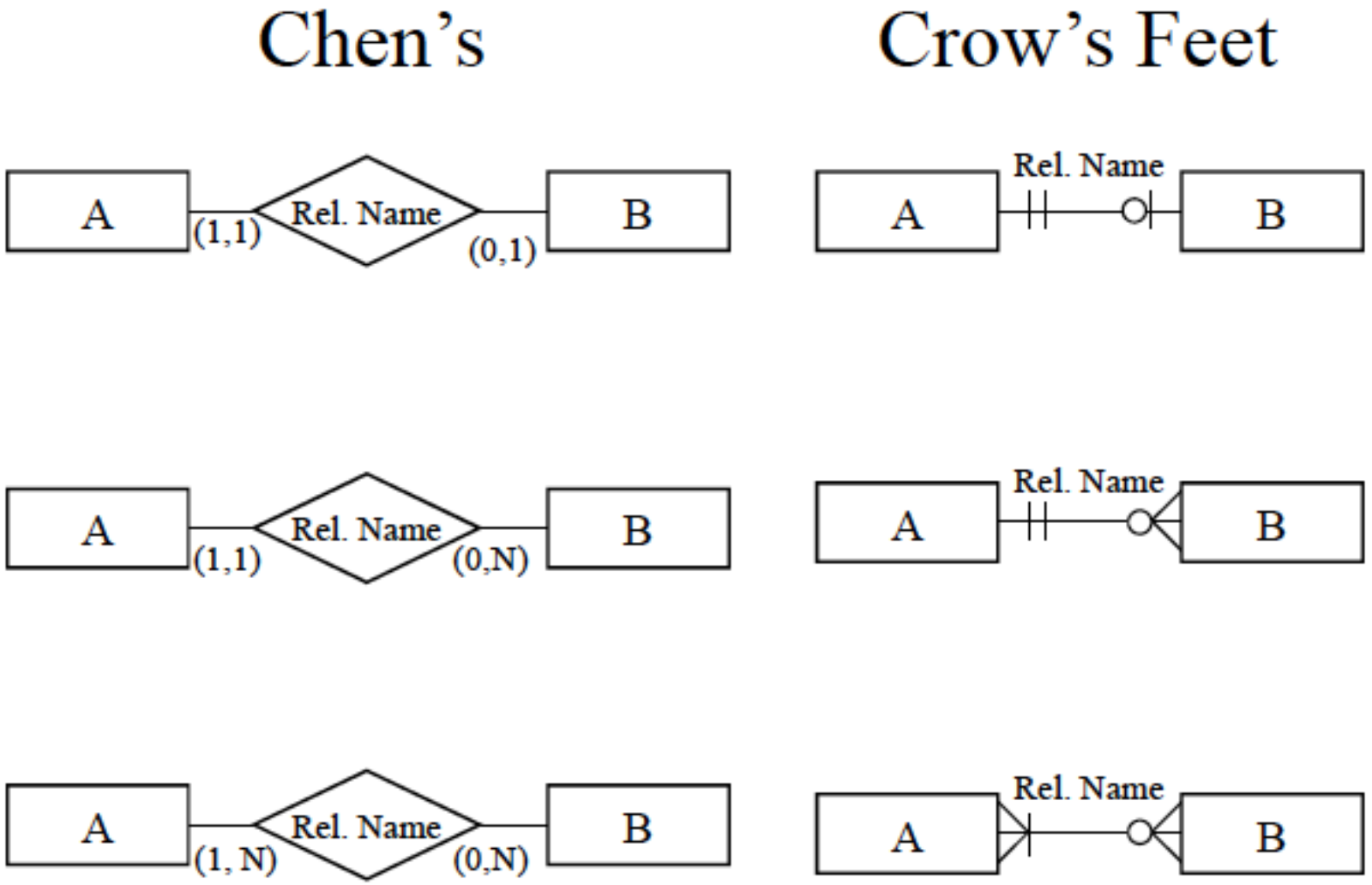
Multi-valued attribute



Relationships

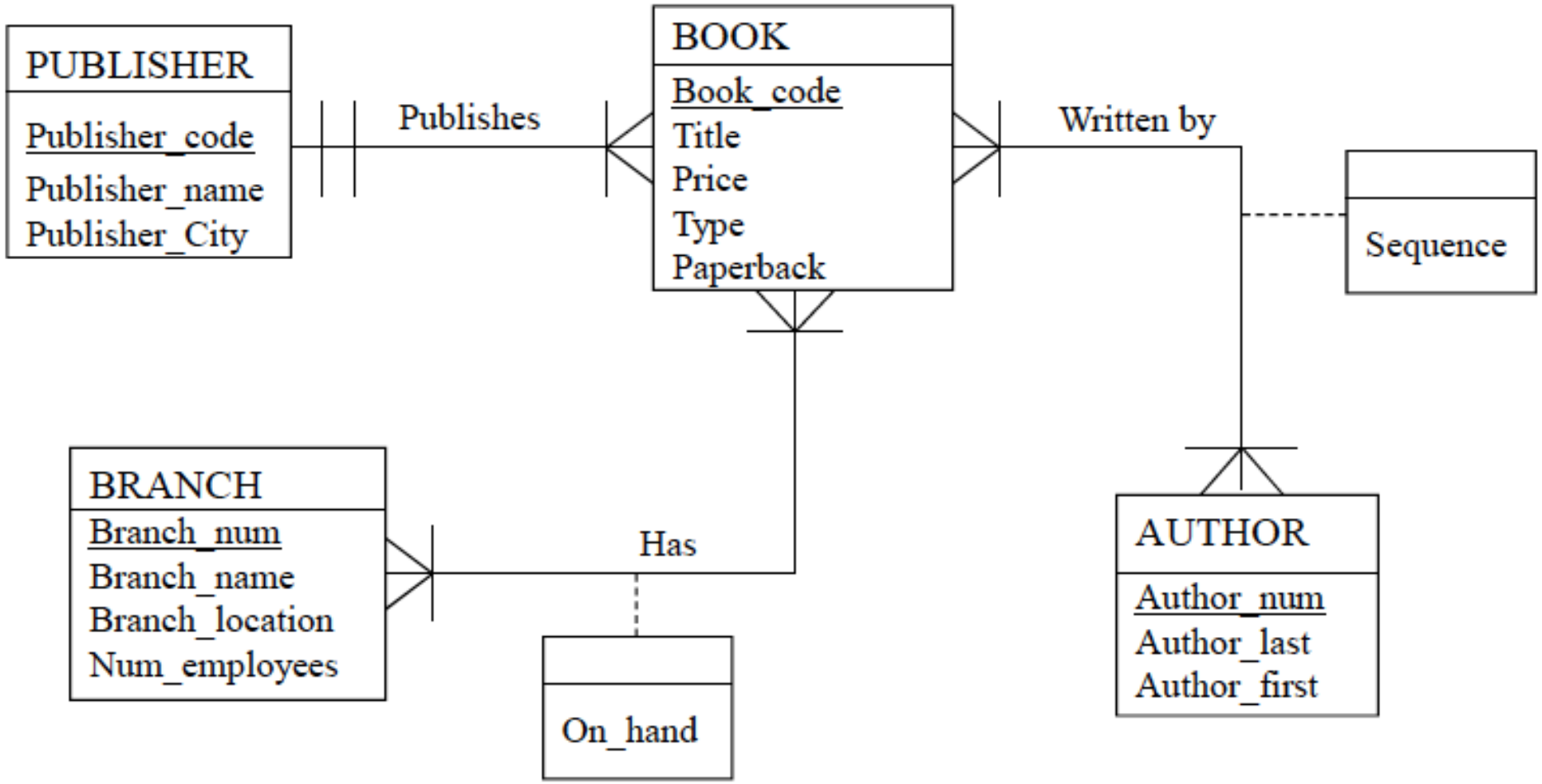


Types of Binary Relationships



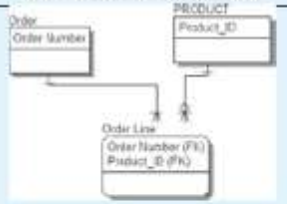
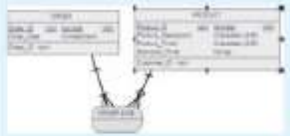
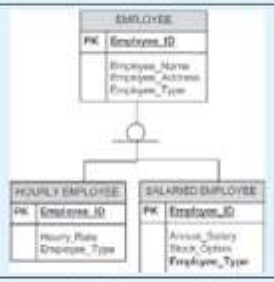
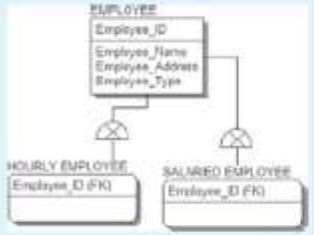
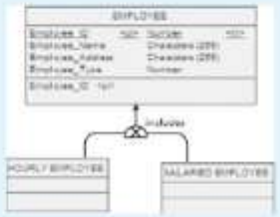
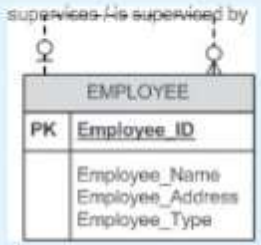
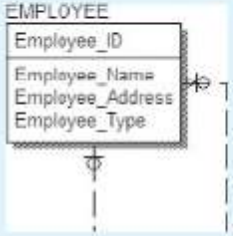
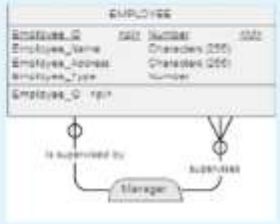
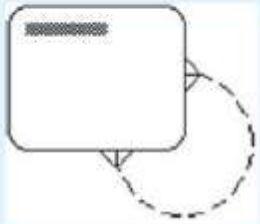
Redo this ER diagram with Crow's feet notation

Source: Janusz Szczypula, 95703 Database Management, 2011.



Source: Janusz Szczypula, 95703 Database Management, 2011.

Modeling Notation

	Hoffer-Ramesh-Topi Notation	Visio PRO 2003	CA ERWin Data Modeler r7.3	Sybase PowerDesigner 15	Oracle Designer 10g
Basic Entity	Strong / Weak	EMPLOYEE	EMPLOYEE	EMPLOYEE	PRODUCT LINE
Associative Entity	Associative				(No special symbol. Uses regular Entity symbol.)
Subtypes	EMPLOYEE; HOURLY EMPLOYEE; SALARIED EMPLOYEE				SUPERTYPE; SUBTYPE A; SUBTYPE B
Recursive Relationship	Manages; EMPLOYEE				
Attributes	ENTITY NAME; Identifier; Partial Identifier; Optional; [Derived]; {Multivalued}; Composite(,)	EMPLOYEE; PK Employee_ID; Employee_Name; Employee_Address; Employee_Type	EMPLOYEE; Employee_ID; Employee_Address; Employee_Name		PRODUCT LINE; # PRODUCT_LINE_ID; * PRODUCT_LINE_NAME

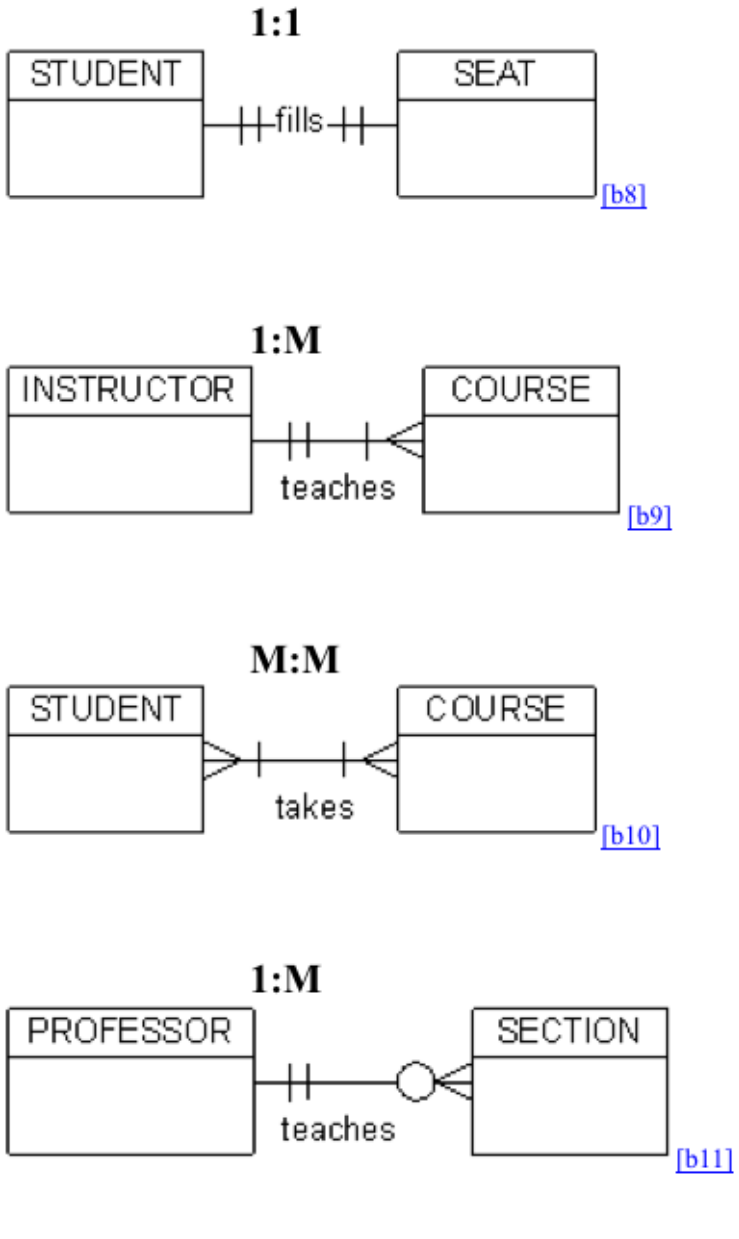
Source: Table A-1, Hoffer et al, 10th ed, 2010.

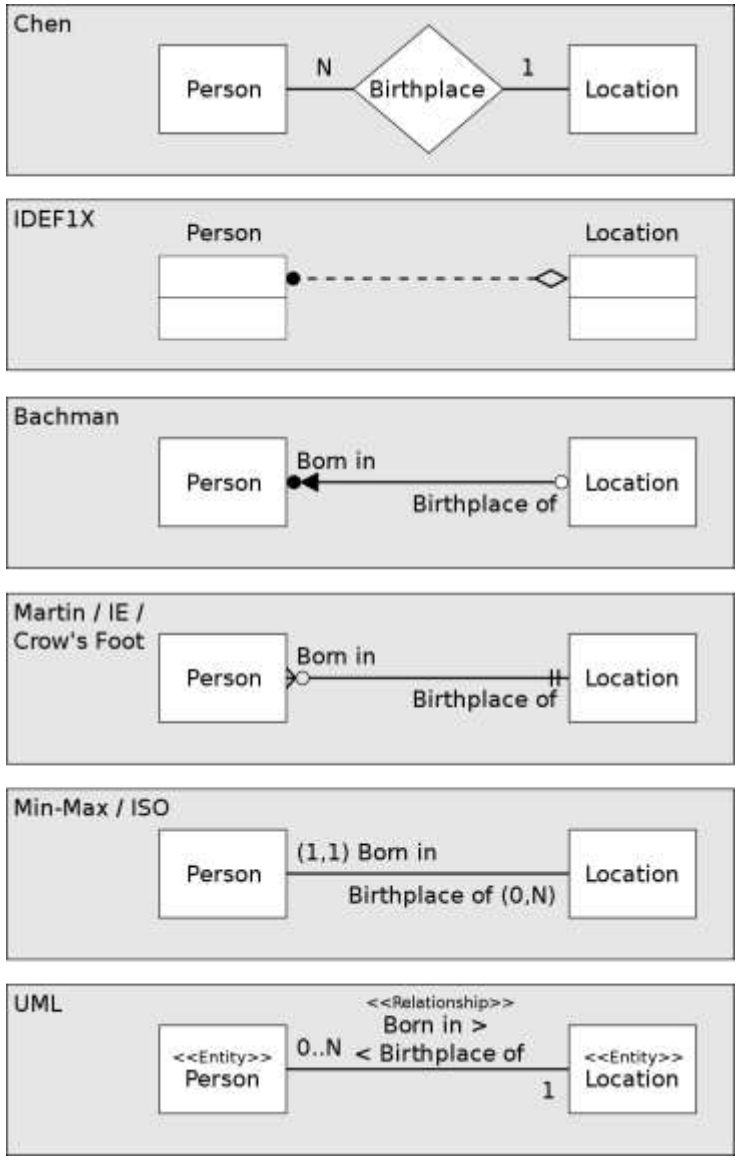
Modeling Cardinality/Optionality Notations

	Hoffer-Ramesh-Topi Notation	Visio PRO 2003	CA ERWin Data Modeler r7.3	Sybase PowerDesigner 15	Oracle Designer 10g
1:1		(Not available without cardinality)	(Not available without cardinality)	0,1 0,1	
1:M		(Not available without cardinality)	(Not available without cardinality)	0,1 0,n	
M:N		(Not allowed)		0,n 0,n	
Mandatory 1:1			1		
Mandatory 1:M			P		
Optional 1:M				0,1 0,n	

Source: Table A-1, Hoffer et al, 10th ed, 2010.

Crow's feet notation and alternatives

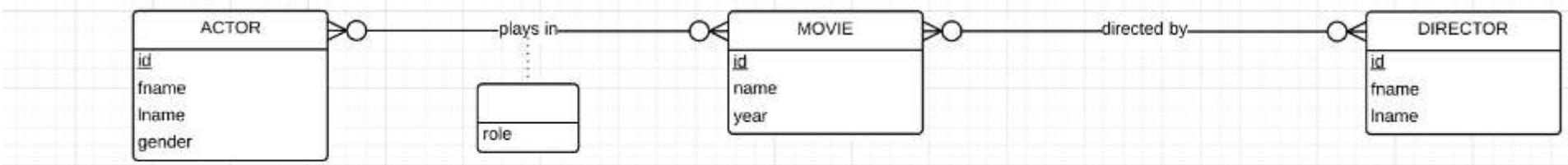




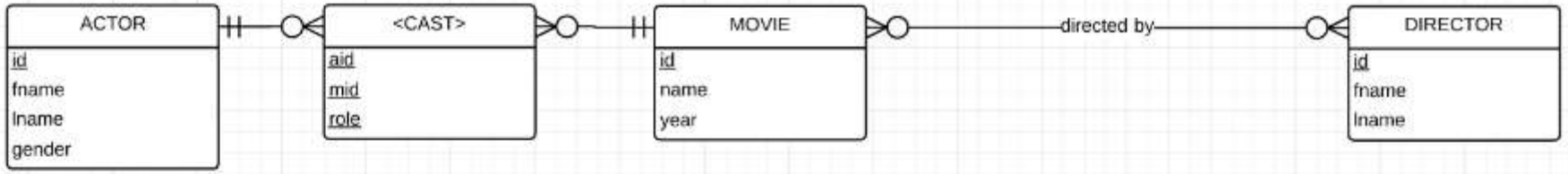
Source: http://en.wikipedia.org/wiki/Entity-relationship_model

IMDB movie database in Lucidchart

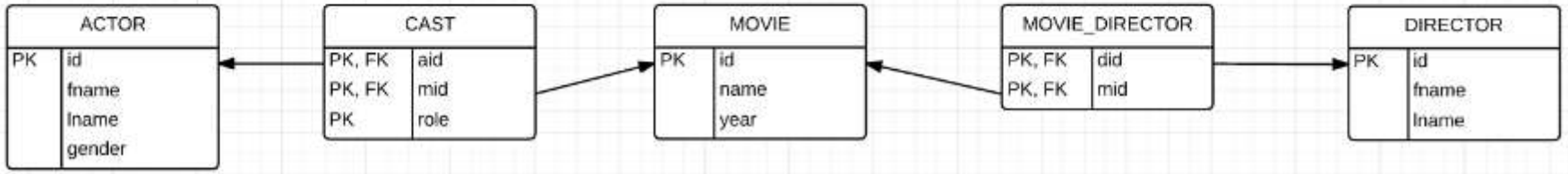
ER diagram: don't forget identifiers, but no FKs



ER diagram: CAST as associative entity can be justified



Relational schema: don't forget PKs and FKs



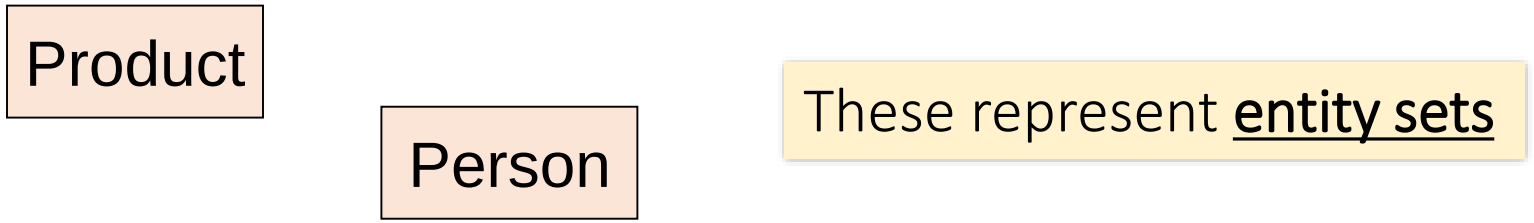
Entities

Entities and Entity Sets

- Entities & entity sets are the primitive unit of the E/R model

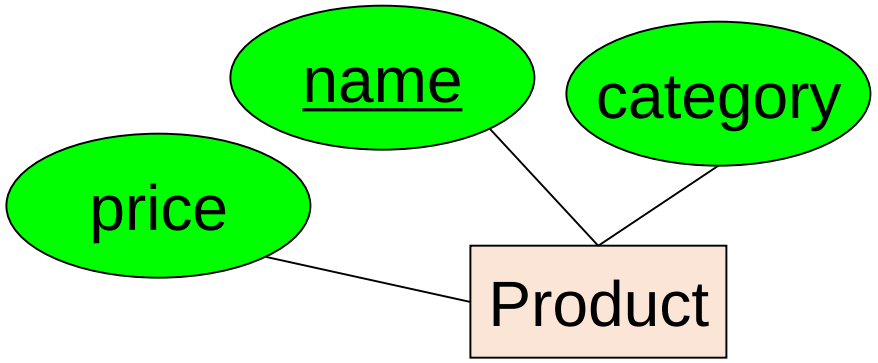
  - Entities: the individual objects, which are members of entity sets
    - Ex: A specific person or product

  - Entity sets: the classes or types of objects in our model
    - Ex: Person, Product
    - These are what is shown in E/R diagrams - as rectangles
    - Entity sets represent the sets of all possible entities



Entities and Entity Sets

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

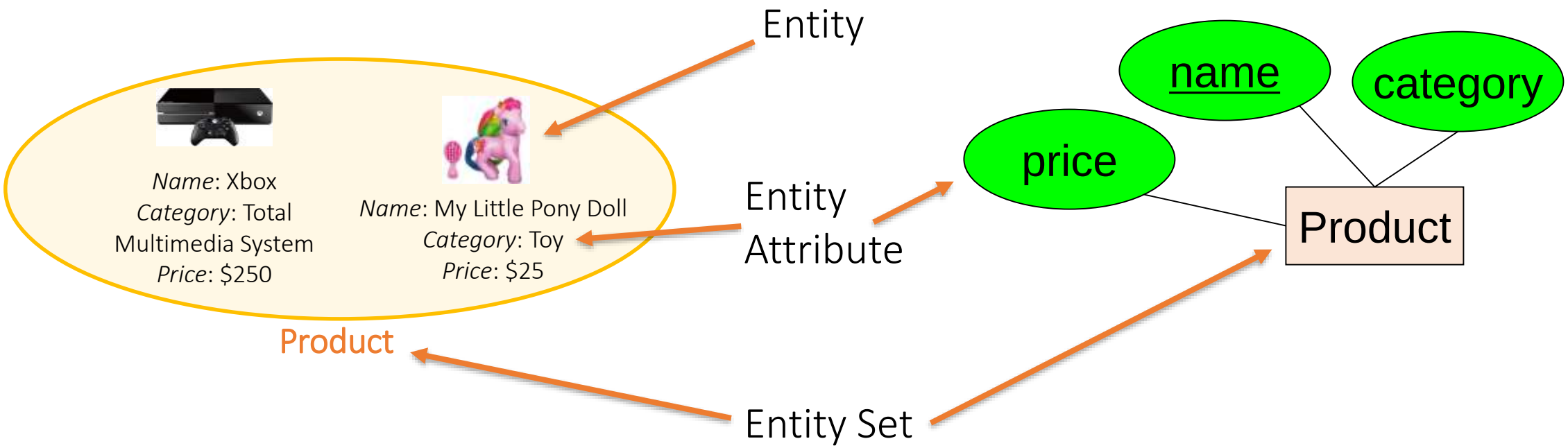


Shapes **are** important.
Colors **are not**.

Entities vs. Entity Sets

- Example:

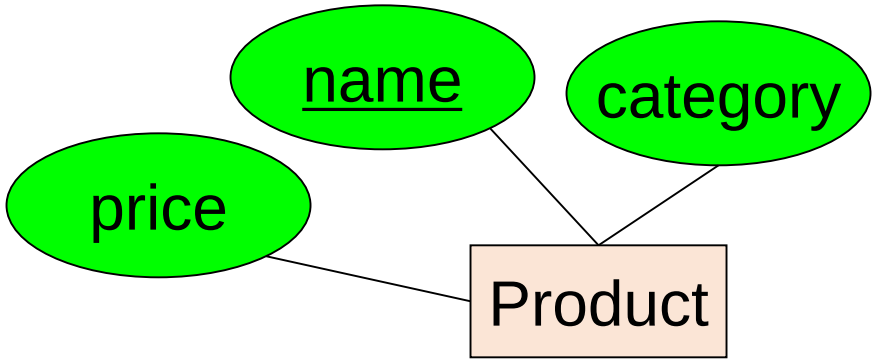
Entities are **not** explicitly represented in E/R diagrams!



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*).

If it were, what would it mean?

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

Identifiers (Keys)

- Identifier (Key): An attribute (or combination of attributes) that uniquely identifies individual instances of an entity type
  - Can be simple or composite
  - Will not be null
  - Will not change in value

    SURROGATE
    - e.g., family name, or telephone number, or street address, if those can change over time (say through marriage...)
  - Substitute new, simple keys for long, composite keys
- Candidate Key: an attribute (or set of) that could be a key…satisfies the requirements for being a key
- Primary Key

Naming Entities

Poor Examples	Good Examples
FormerStudentFromIowa	Student
Customers	Customer
ClientsWhoCameToBigEvent	Employee
ObscureRecmdForFrtherAction	Invoice
Order	Purchase Order
	Flight

- Guidelines for naming entity types:
  - Use singular nouns
  - Names should be specific to the organization
  - Be concise
  - Abbreviations are ok, as long as they are standardized
  - Event entity types should be named for the result of the event (e.g., "Order")
  - Be consistent

Exercise (Part I): Entities / Attributes

- Identify the entities that appear on the report card
- Identify the attributes of each previously identified entity

MILLENNIUM COLLEGE
GRADE REPORT
FALL SEMESTER 200X

NAME:	Emily Williams	ID: 268300458
CAMPUS ADDRESS:	208 Brooks Hall	
MAJOR:	Information Systems	

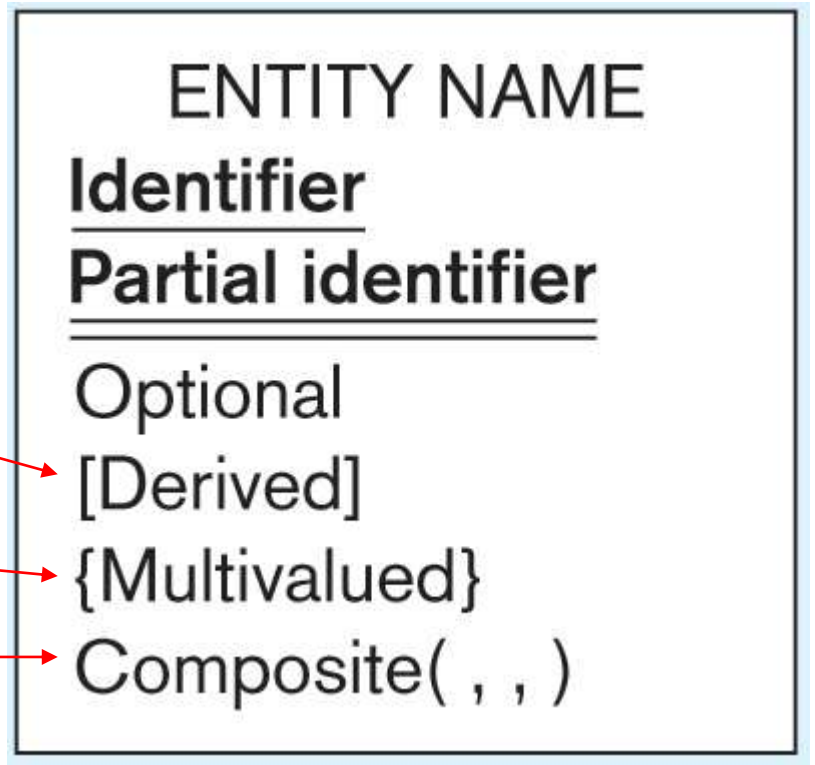
COURSE ID	TITLE	INSTRUCTOR NAME	INSTRUCTOR LOCATION	GRADE
IS 350	Database Mgt.	Codd	B104	A
IS 465	System Analysis	Parsons	B317	B

Source: Hoffer, Ramesh, Topi, "Modern database management," 10th ed, 2010.

Attributes

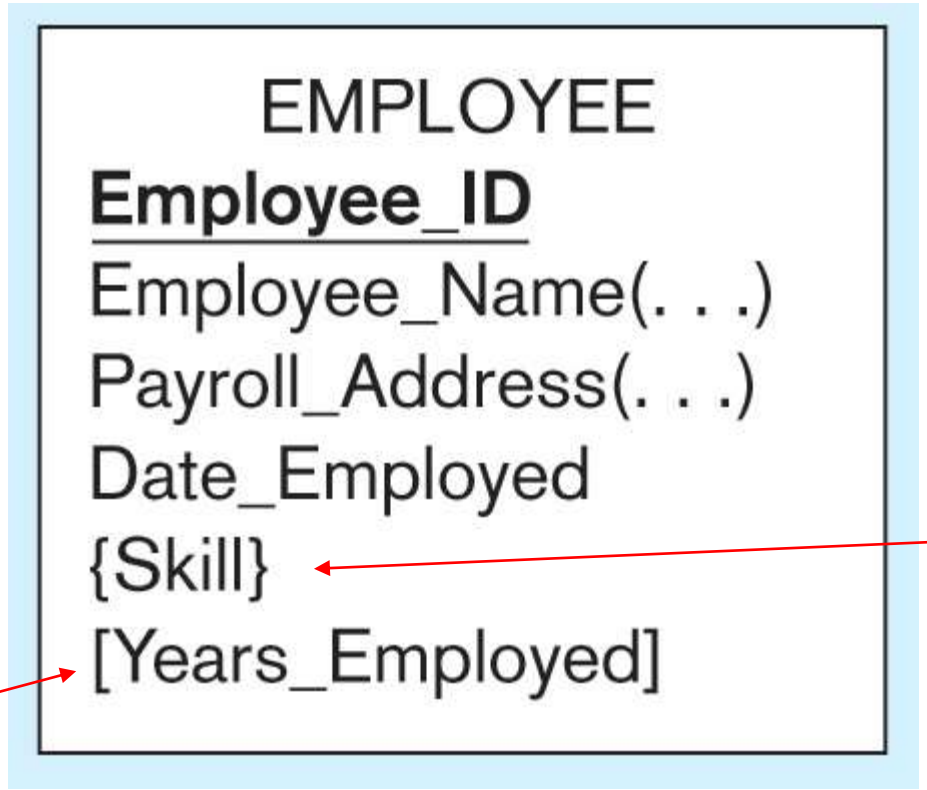
- A property or characteristic of an entity type

- Classifications of attributes:
  - Identifier Attributes
  - Required versus Optional
  - Stored versus Derived
  - Single-Valued versus Multivalued Attribute
  - Simple versus Composite



Source: Hoffer, Ramesh, Topi, "Modern database management," 10th ed, 2010.

Example: Describe the Attributes



Multivalued Attribute (e.g., SQL, Python, ...)

Derived Attribute

Source: Hoffer, Ramesh, Topi, "Modern database management," 10th ed, 2010.

Naming Attributes

Poor Examples	Good Examples
TheDayThatThisPersonEnrolled	Date
	Birth_Date
NumEnrollInSpecificClass	NumberEnrolled
Student_Names	StudentName
ClientLastName	CourseID
	Employee_ID

- Guidelines for naming attributes:
  - Be concise
  - Use singular nouns or noun phrases
  - Names should be unique (at least within an entity type)
  - Follow a standard format (e.g., either Camelcase or "_")
  - Similar attributes should use the same qualifiers and classes (e.g., CustomerID, ProductID)

Example: modeling flights

- Assume you want to model "flights"
- Attributes: FlightNumber, Date, NumberOfPassengers
- What would be the key / identifier?

Example: modeling flights

- Assume you want to model "flights"
- Attributes: FlightNumber, Date, NumberOfPassengers
- What would be the key / identifier?

FLIGHT

Flight_ID

(Flight_Number, Date)

Number_of_Passengers

. . .

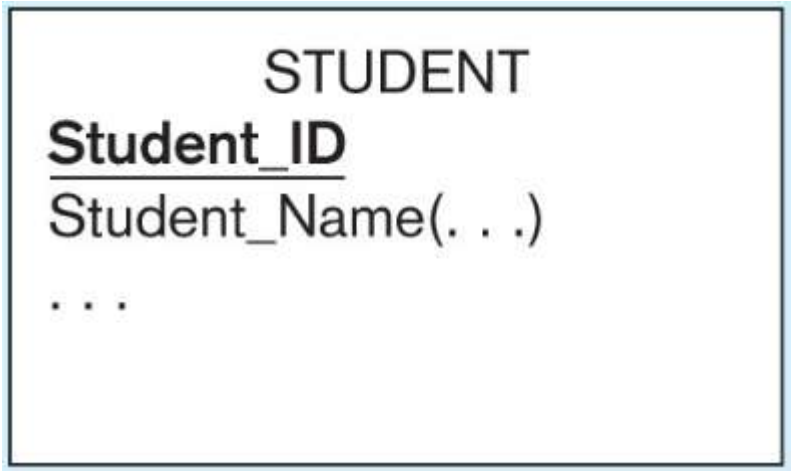
US Airways Flight 1549

The downed US Airways Flight 1549 floating on the Hudson River

Accident summary	
Date	January 15, 2009

Identifier Examples: Simple and Composite

- Simple identifiers:
  - Single attribute uniquely identifies each entity instance
  - Identifier attribute underlined



- Composite identifiers:
  - Multiple attributes required to uniquely identifies each entity instance
  - Identifier attribute underlined and composite attributes listed below in (parentheses)

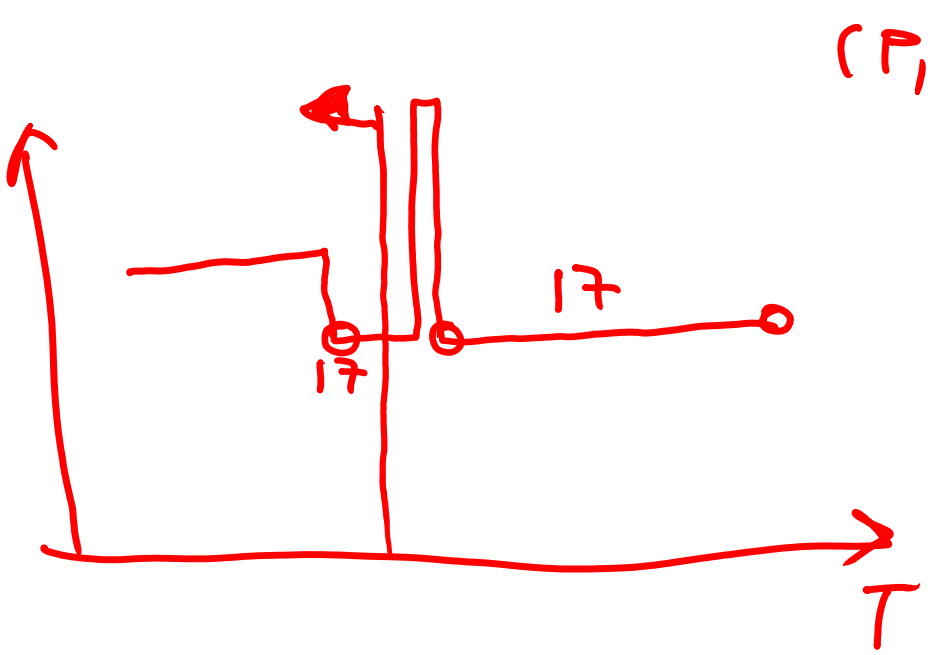
Source: Hoffer, Ramesh, Topi, "Modern database management," 10th ed, 2010.

Example: modeling time-dependent data

- Assume you have an entity "product"
- The price can change over time
- You would like to preserve the history of prices and the time period

Example: modeling time-dependent data

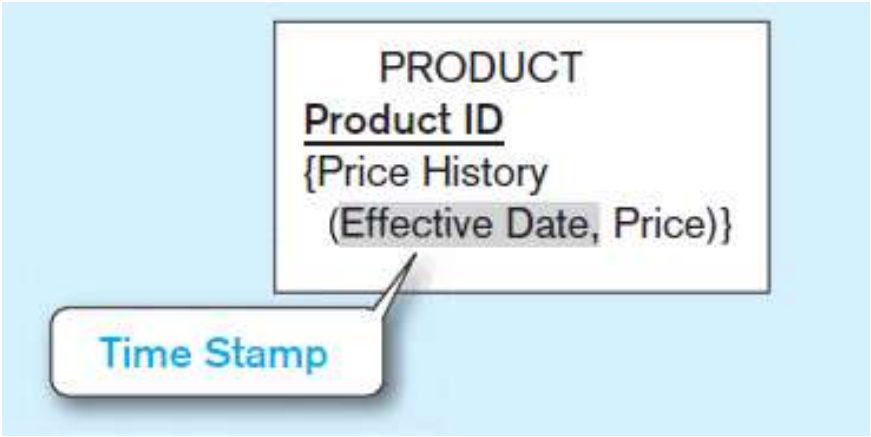
- Assume you have an entity "product"
- The price can change over time
- You would like to preserve the history of prices and the time period



Example: modeling time-dependent data

- Assume you have an entity "product"
- The price can change over time
- You would like to preserve the history of prices and the time period

101, {(Jan, '18), (Jul 9, ...}



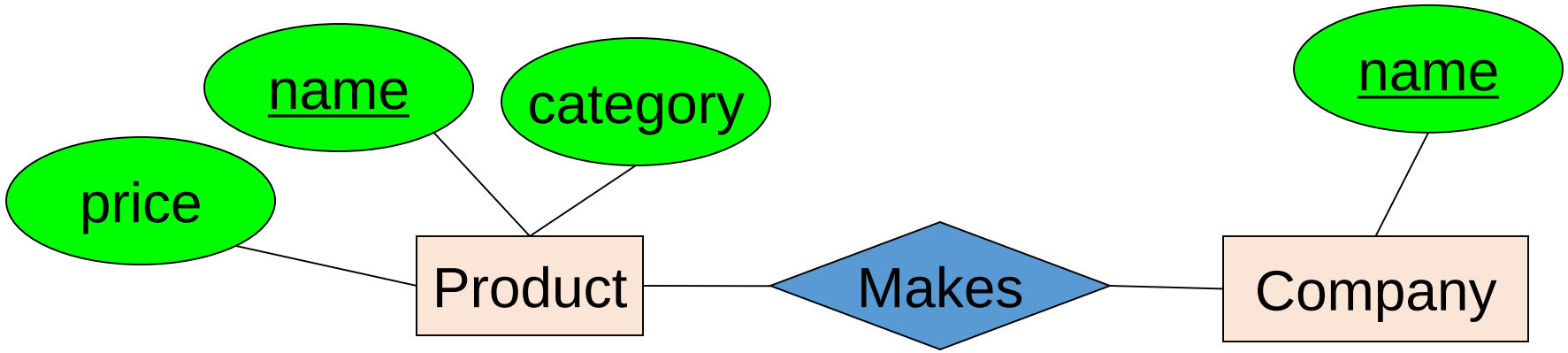
Time-stamping is commonly done with a multi-valued and composite attribute (or associative entities: see later)

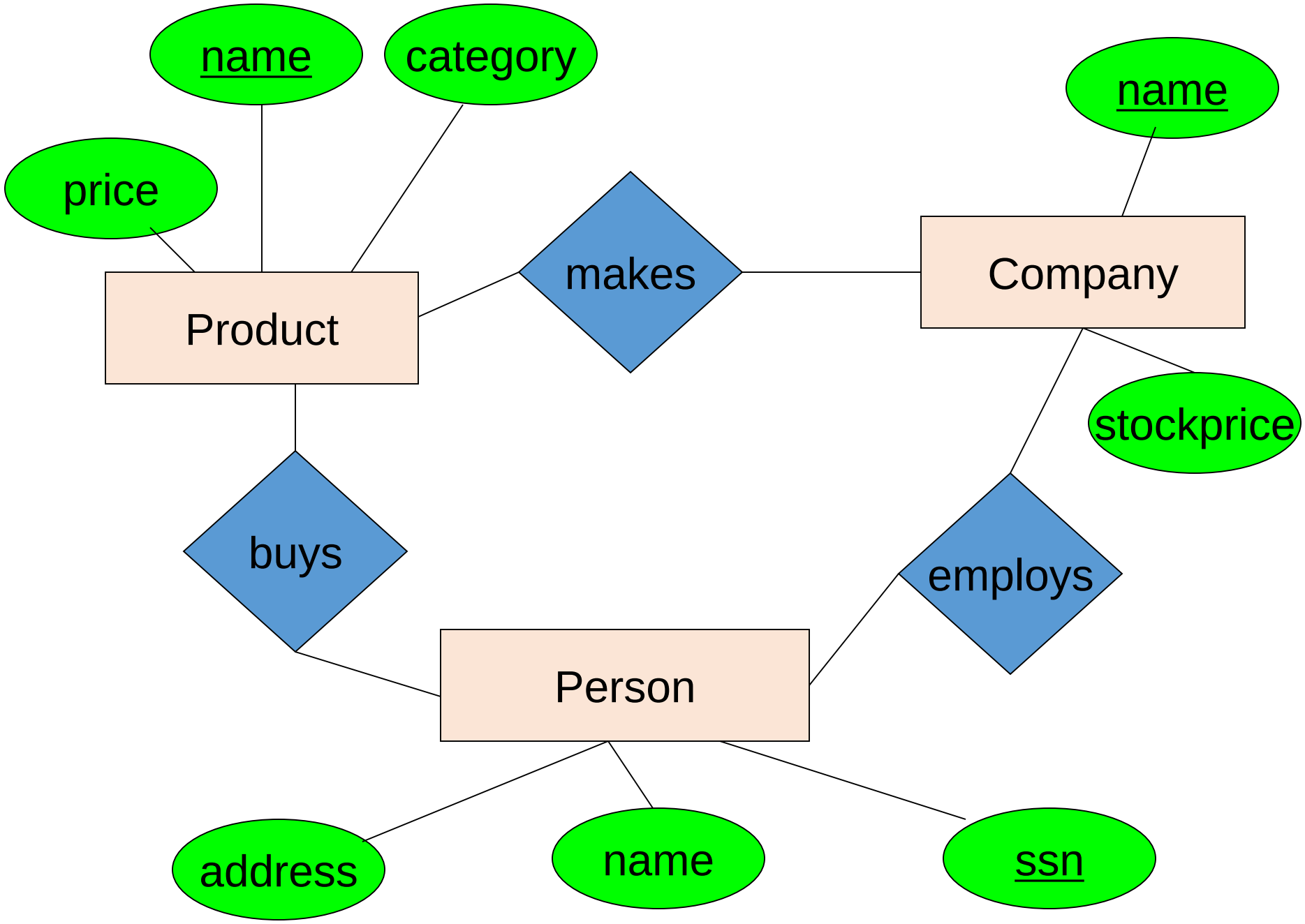
Source: Hoffer, Ramesh, Topi, "Modern database management," 10th ed, 2010.

Relationships

The R in E/R: Relationships

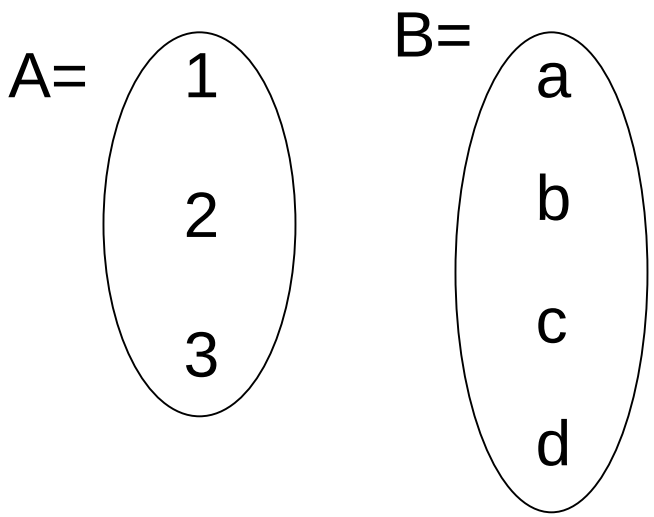
- A relationship is between two or more entities




name
category
price
Product
makes
name
Company
stockprice
buys
employs
Person
address
name
ssn

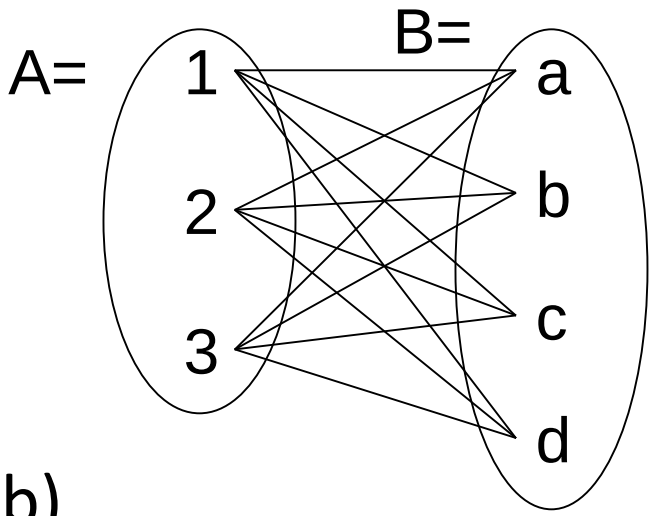
What is a Relationship?

- A mathematical definition:
  - Let A, B be sets
    - A={1,2,3}, B={a,b,c,d}



What is a Relationship?

- A mathematical definition:

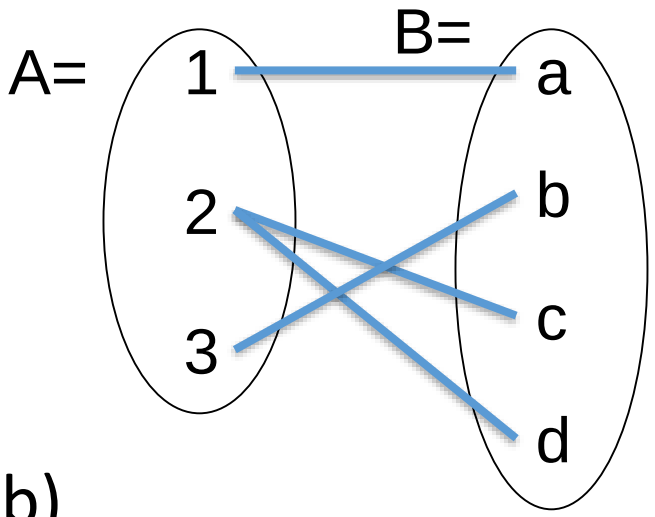


  - Let A, B be sets
    - A={1,2,3}, B={a,b,c,d}

  - A x 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?

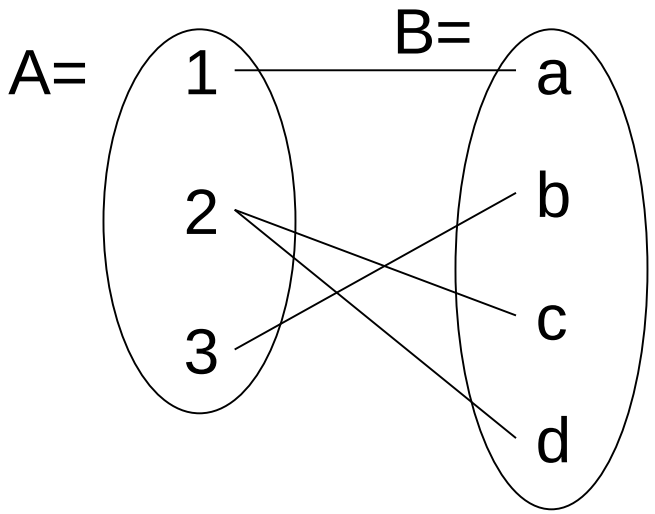
- A mathematical definition:



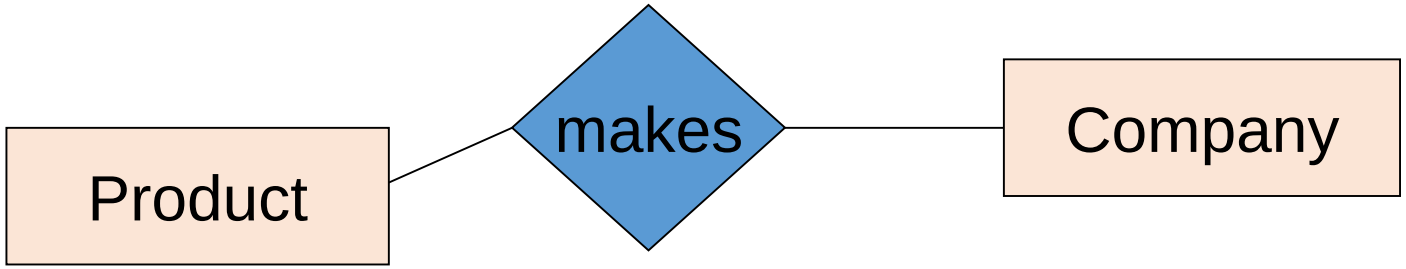
  - Let A, B be sets
    - A={1,2,3}, B={a,b,c,d},
  - A x B (the cross-product) is the set of all pairs (a,b)
    - $A \times B = \{(1,a), (1,b), (1,c), (1,d), (2,a), (2,b), (2,c), (2,d), (3,a), (3,b), (3,c), (3,d)\}$
  - We define a relationship to be a subset of A x B
    - R = {(1,a), (2,c), (2,d),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

What is a Relationship?

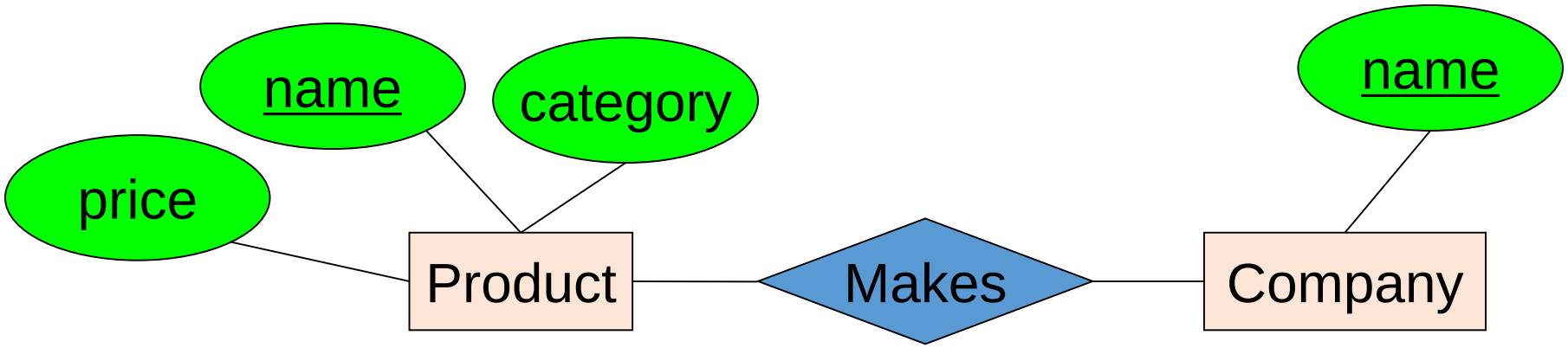
- A mathematical definition:
  - Let A, B be sets
  - A x B (the cross-product) is the set of all pairs
  - A relationship is a subset of A x B



- Makes is a relationship: it is a subset of Product × 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***

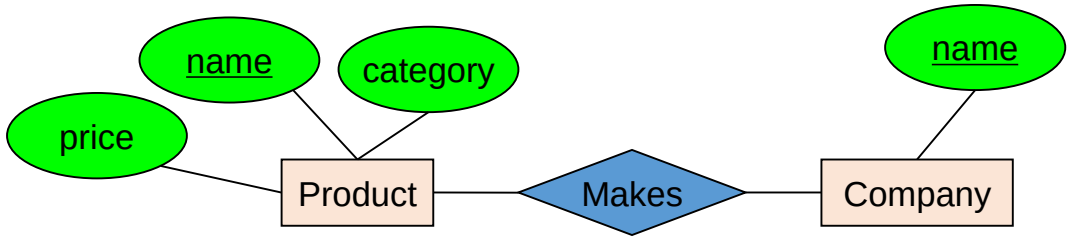
What is a Relationship?

Company

name
GizmoWorks
GadgetCorp

Product

name	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

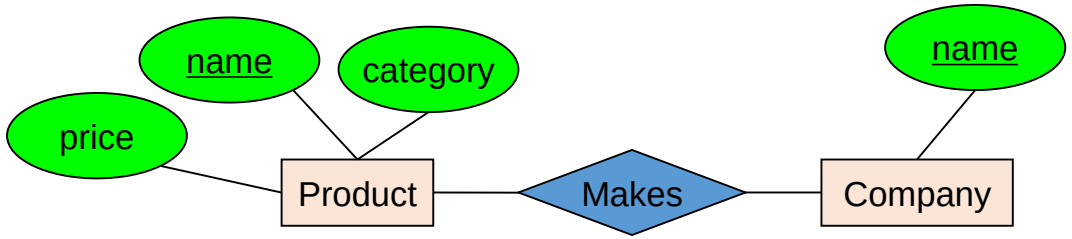
name
GizmoWorks
GadgetCorp

Product

name	category	price
Gizmo	Electronics	$9.99
GizmoLite	Electronics	$7.50
Gadget	Toys	$5.50

Company C × Product P

C.name	P.name	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

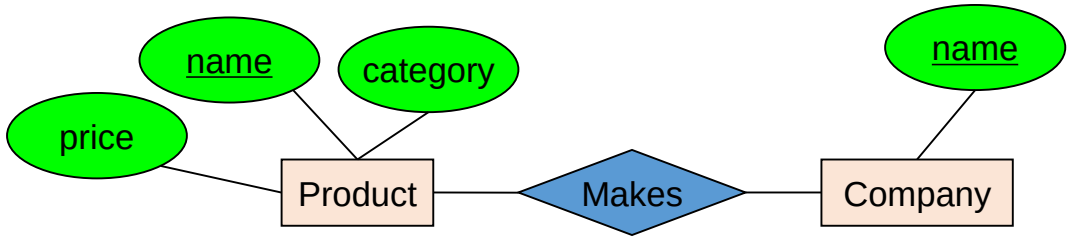
name
GizmoWorks
GadgetCorp

Product

name	category	price
Gizmo	Electronics	$9.99
GizmoLite	Electronics	$7.50
Gadget	Toys	$5.50

Company C × Product P

C.name	P.name	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

C.name	P.name
GizmoWorks	Gizmo
GizmoWorks	GizmoLite
GadgetCorp	Gadget

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***

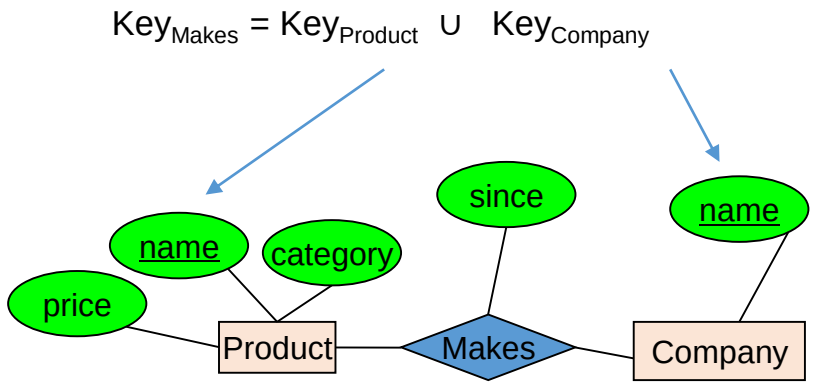
What is a Relationship?

- There can only be one relationship (instance) for every unique combination of entities

This follows from our mathematical definition of a relationship- it's a SET!

- This also means that the relationship is uniquely determined by the keys of its entities

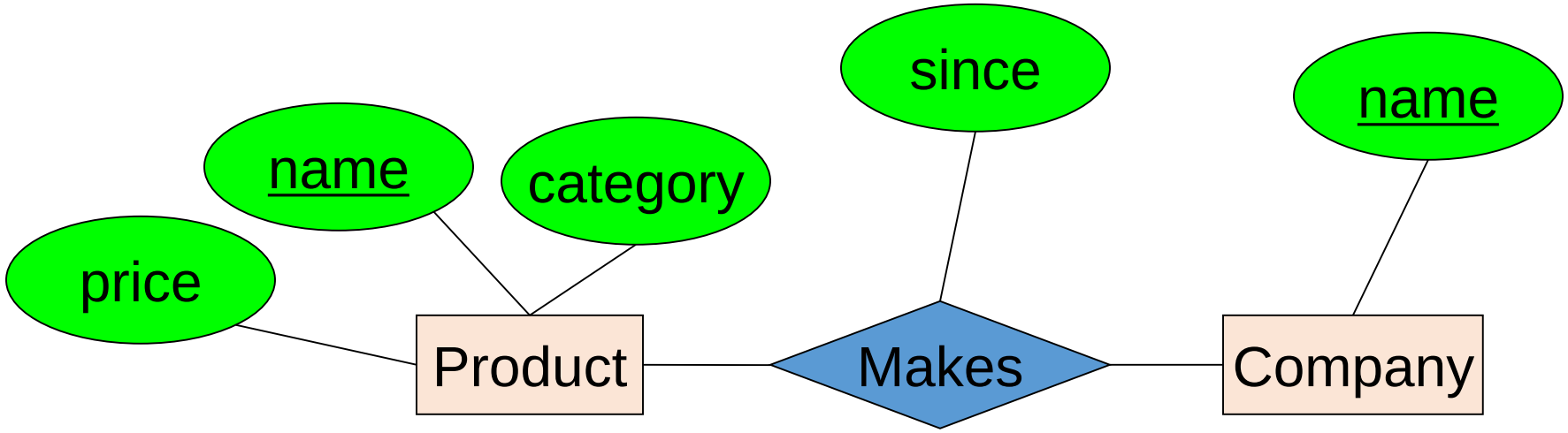
- Example: the "key" for Makes (to right) is
  {Product.name, Company.name}



Why does this make sense?

Relationships and Attributes

- Relationships may have attributes as well.



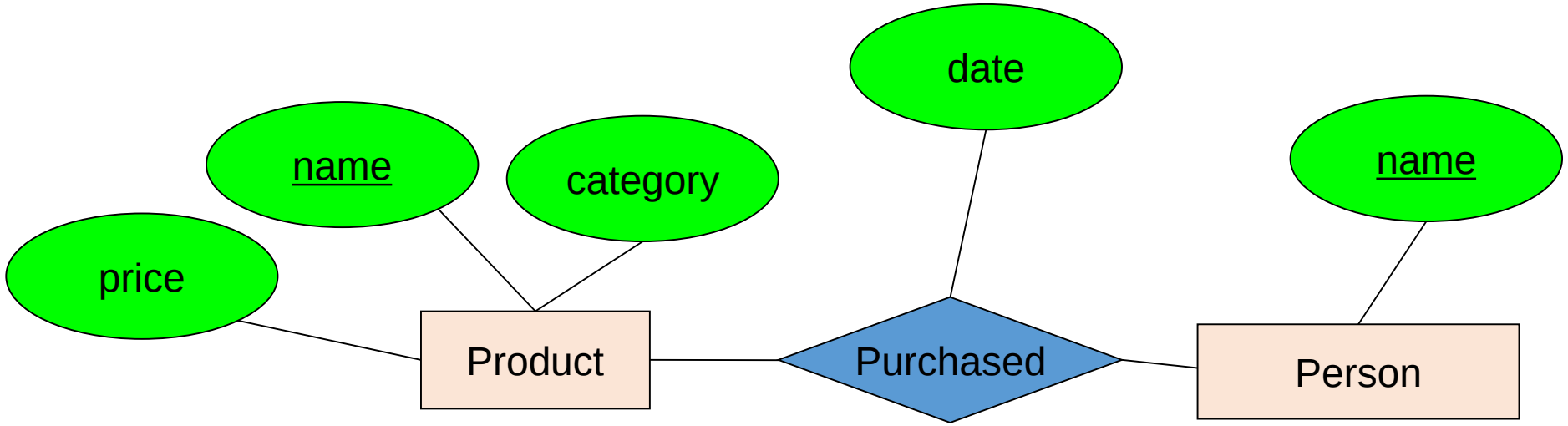
For example: "since" records when company started making a product

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

Note #2: Why not "how long"?

Decision: Relationship vs. Entity?

- Q: What does this say?

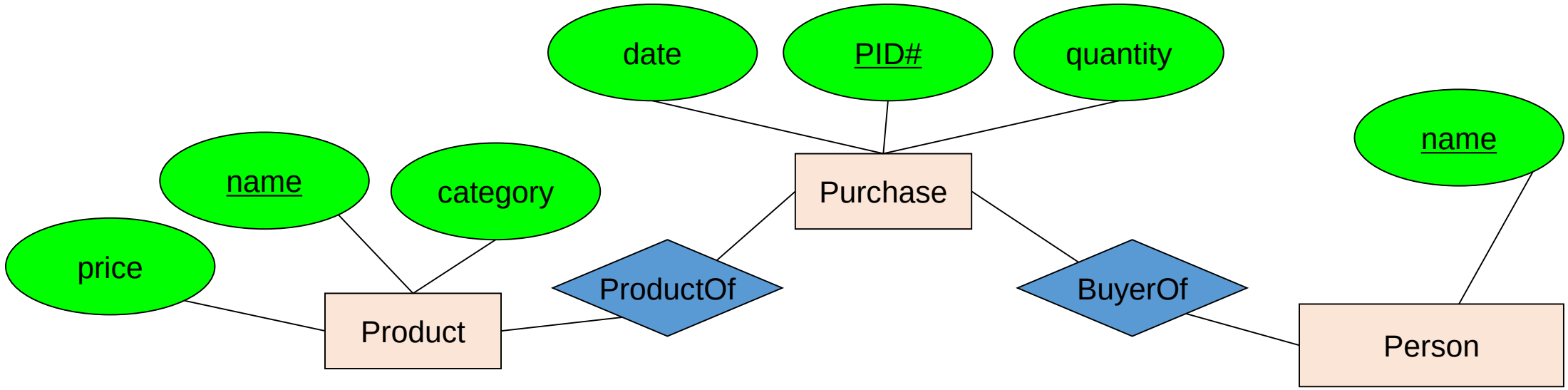


- A: A person can only buy a specific product once (on one date)

Modeling something as a relationship makes it unique; what if not appropriate?

Decision: Relationship vs. Entity?

- What about this way?

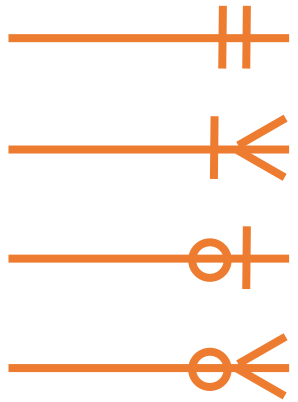


- Now we can have multiple purchases per product, person pair!

We can always use **a new entity** instead of a relationship. For example, to permit multiple instances of each entity combination!

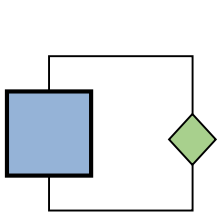
Overview: 3 important concepts for relationships

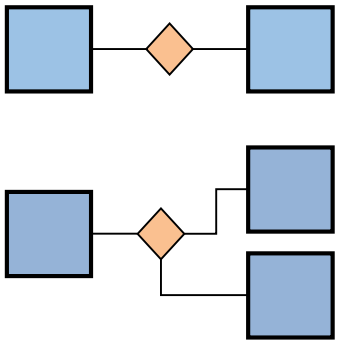
- **Cardinality** ("arity"): number of entity instances that participate (~mainly max)



- **Participation constraints**: mandatory or optional (equivalent to minimum cardinality 0 or 1)

- **Degree**: number of entity types that participate





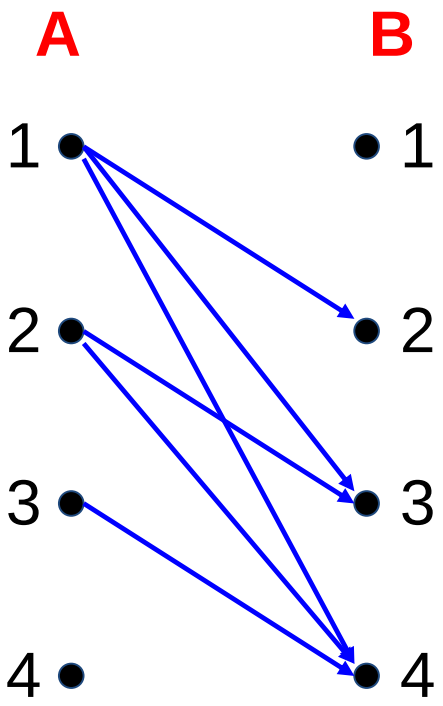
Random question: Why "relational model"?

Relations

Definition: Let A and B be sets. A binary relation from A to B is a subset of A×B.

Example: R = {(1, 2), (1, 3), (1, 4), (2, 3), (2,4), (3,4)}

(R = {(a, b) | a < b} with a, b from A=B={1, 2, 3, 4})



B					
	4	X	X	X	
	3	X	X		
	2	X			
	1				
	R	1	2	3	4
			A		

Definition of Function: For nonempty sets *A* and *B*, a function *f* from *A* to *B*, denoted *f:A→B*, is a relation from *A* to *B* in which every element of *A* appears exactly once as the first component of an ordered pair in the relation.