# L08-L12: Database design and normalization

CS3200 Database design (fa18 s2)

https://northeastern-datalab.github.io/cs3200/

Version 10/1/2018

# L08: ER modeling

CS3200 Database design (fa18 s2)

https://northeastern-datalab.github.io/cs3200/

Version 10/1/2018

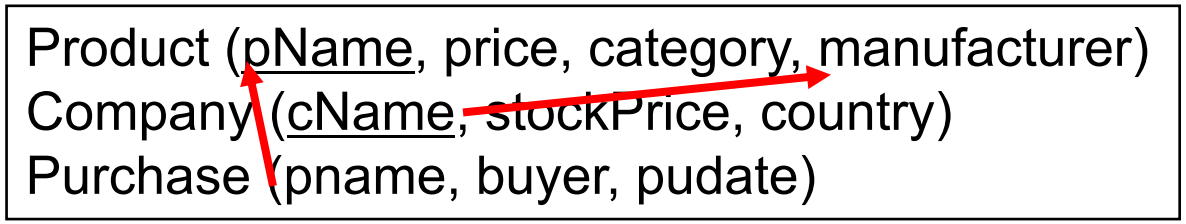
# Announcements!

- Exam1 (THU Oct 4): Laptop, BlackBoard, Postgres, SQL only.
  - Closed book, yet one page cheatsheet required (see template on website)
  - Sign and hand-back "class honor code" (on website or today in class)
  - Please watch the "character is more important" from our website!
  - We will have hand-outs (e.g. with the database schema) and a text file to be filled out
  - Practice exam today: just modalities (Niklas will be around to help)
- Updates on homeworks (Niklas)
  - HW solutions posted by tomorrow night (thus no extensions)
- Lucidcharts: recommended but not required
  - feel free to post alternatives on Piazza
- Confidential or anonymous questions on HW?
  - please post on Piazza "visible to instructors only"
  - Anonymous question to instructor only: Google feedback form

# HW2

# Something tricky about Nested Queries






Are these queries equivalent?

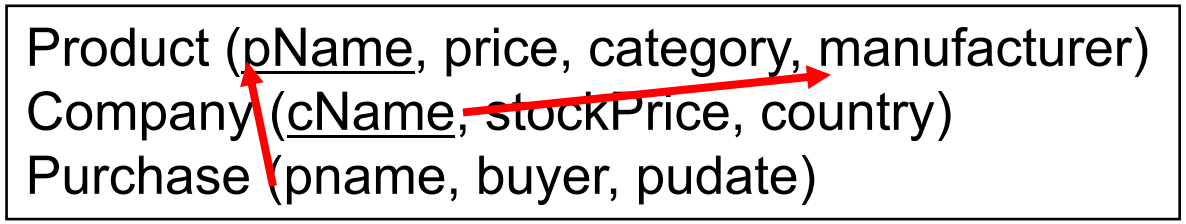
```
SELECT C.country
FROM   Company C
WHERE  C.cname  IN (
SELECT P.manufacturer
FROM   Purchase PU, Product P
WHERE  P.pname = PU.pname
  AND  PU.buyer = 'Joe B')
```

```
SELECT C.country
FROM   Company C,
       Product P,
       Purchase PU
WHERE  C.cname = P.manufacturer
  AND  P.pname = PU.pname
  AND  PU.buyer = 'Joe B'
```

Beware of duplicates!

# Something tricky about Nested Queries






Are they now equivalent?

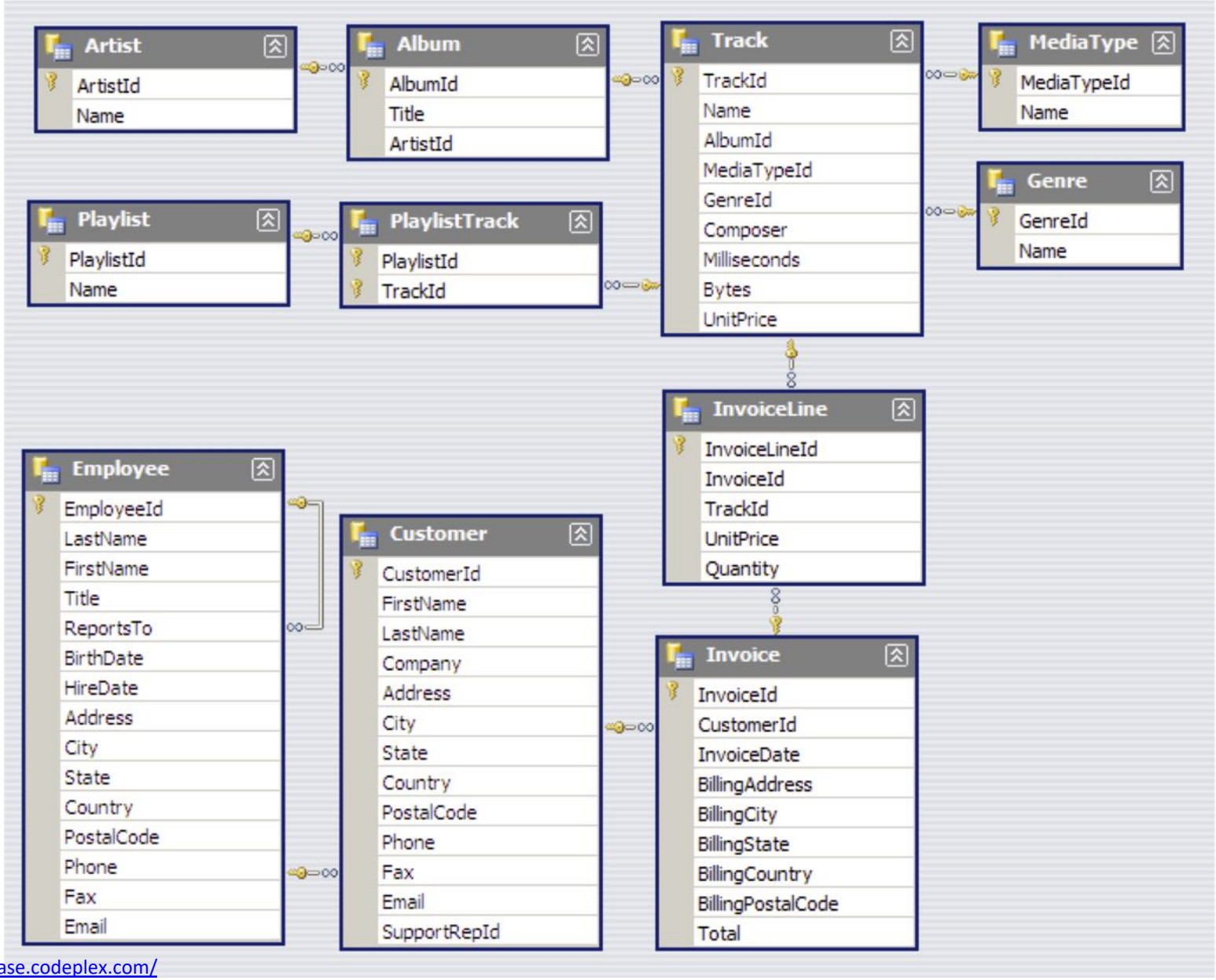
```
SELECT C.country
FROM   Company C
WHERE  C.cname  IN (
SELECT P.manufacturer
FROM   Purchase PU, Product P
WHERE  P.pname = PU.pname
  AND  PU.buyer = 'Joe B')
```

```
SELECT DISTINCT C.country
FROM   Company C,
       Product P,
       Purchase PU
WHERE  C.cname = P.manufacturer
  AND  P.pname = PU.pname
  AND  PU.buyer = 'Joe B'
```

Beware of duplicates!

# Chinook database





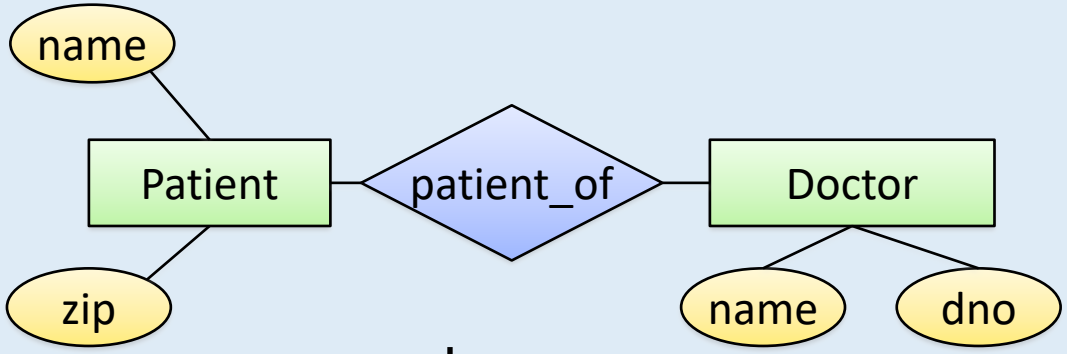

Source: https://chinookdatabase.codeplex.com/

# ER modeling

# Data modeling and Database Design Process

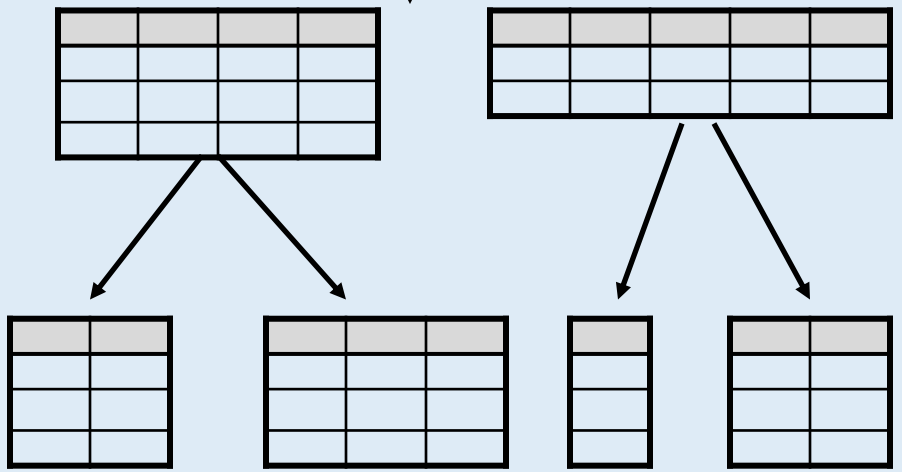
## 1. ER Diagram

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



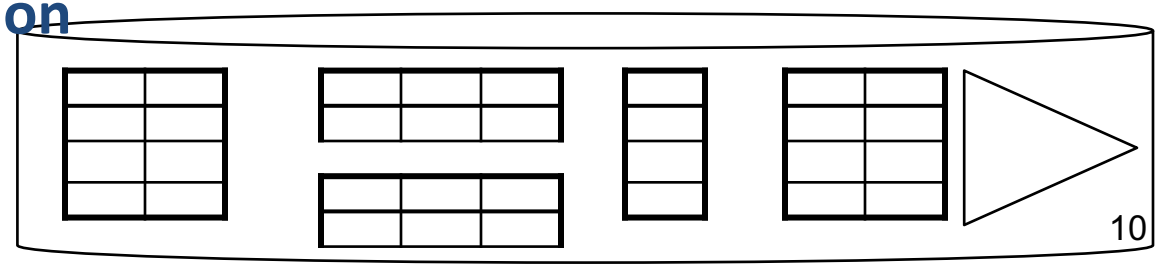

## 2. Relational Database Design

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



## 3. Database Implementation

Physical Model
Physical storage details
Result: Physical Schema



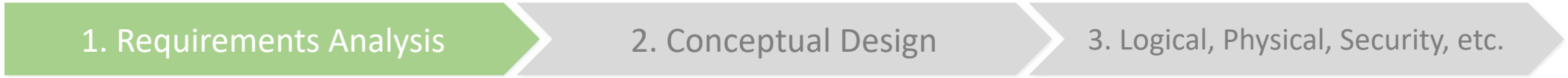
# Database Design

- 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

- 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

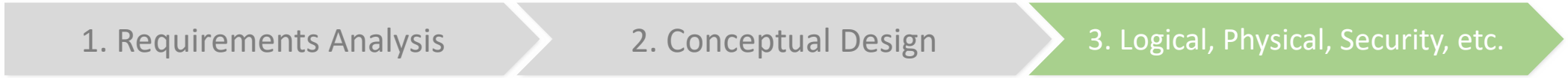
# Database Design Process


## 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



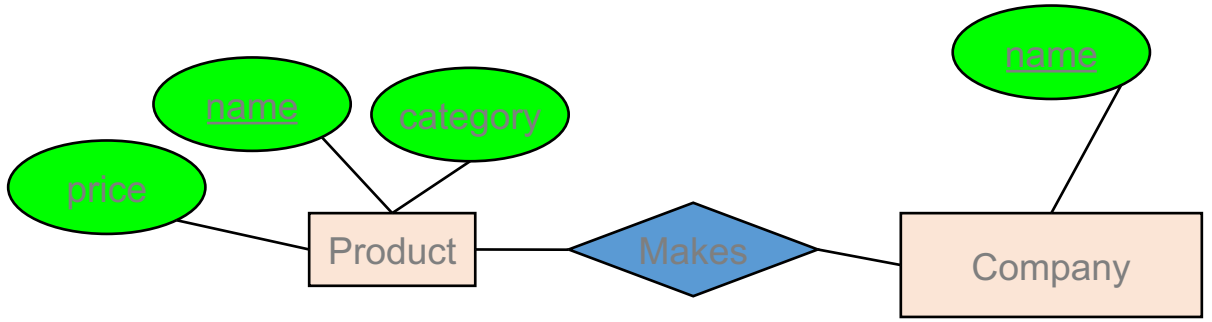

3. More:
   - Logical Database Design
   - Physical Database Design
   - Security Design

# 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

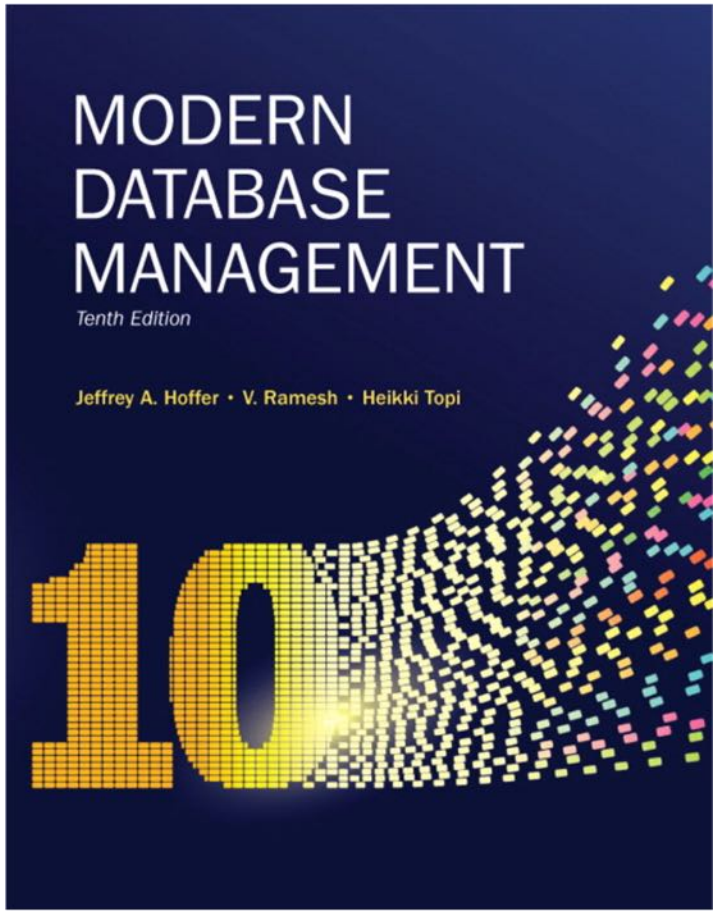
- The E/R model is one of the most cited articles in Computer Science
  - “The Entity-Relationship model – toward a unified view of data” Peter Chen, 1976
  - Compare to "business model canvas", Alexander Osterwalder 2008
    https://en.wikipedia.org/wiki/Business_Model_Canvas

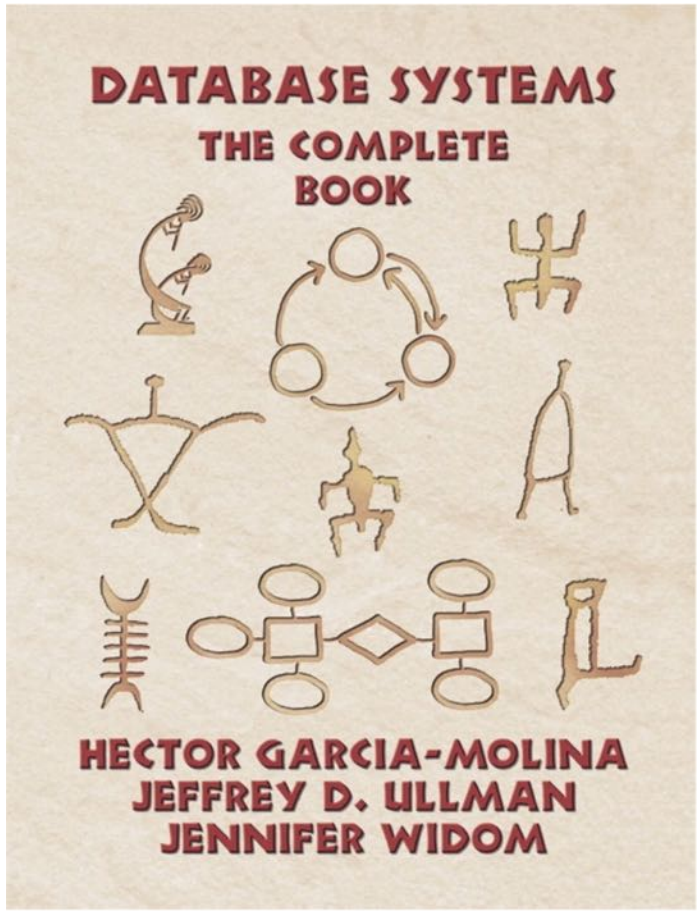
- Used by companies big and small
  - You’ll know it soon enough

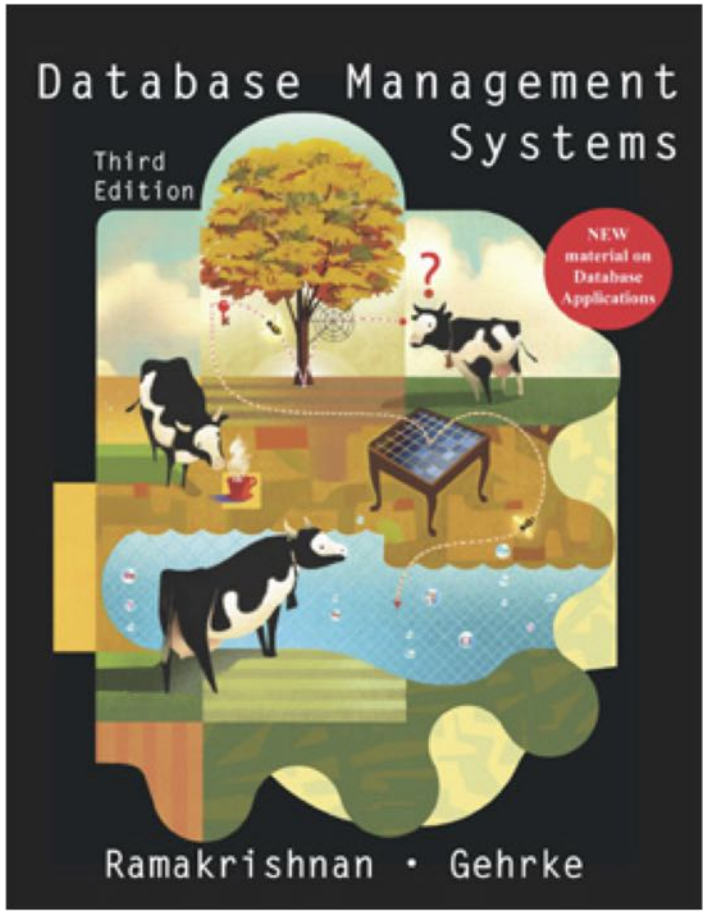
- "Chen notation": different from "UML"

# Some comments on Notations

# Different sources, different notations



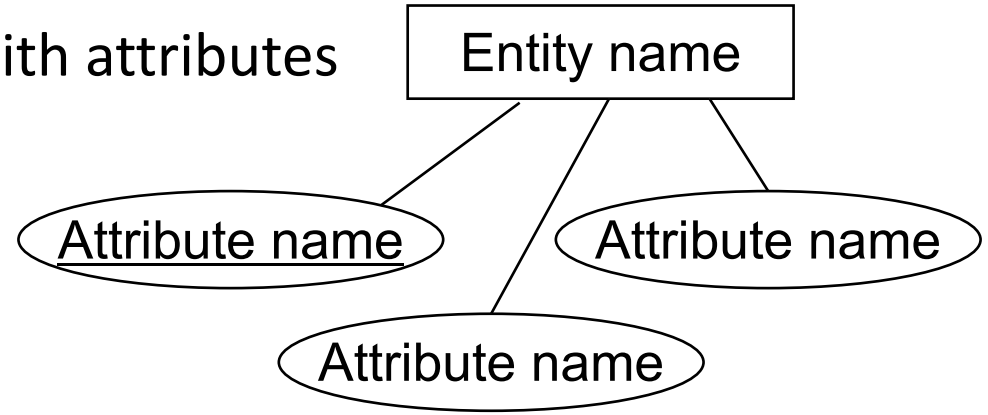
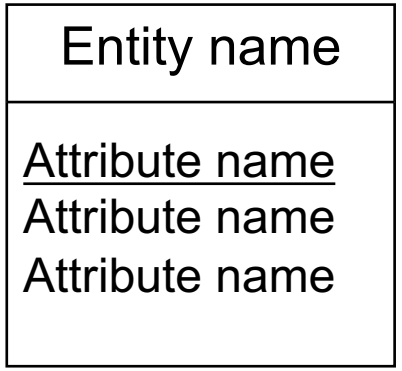




Craw foot

Chen / Stanford arrow notation

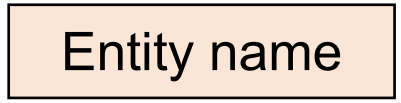
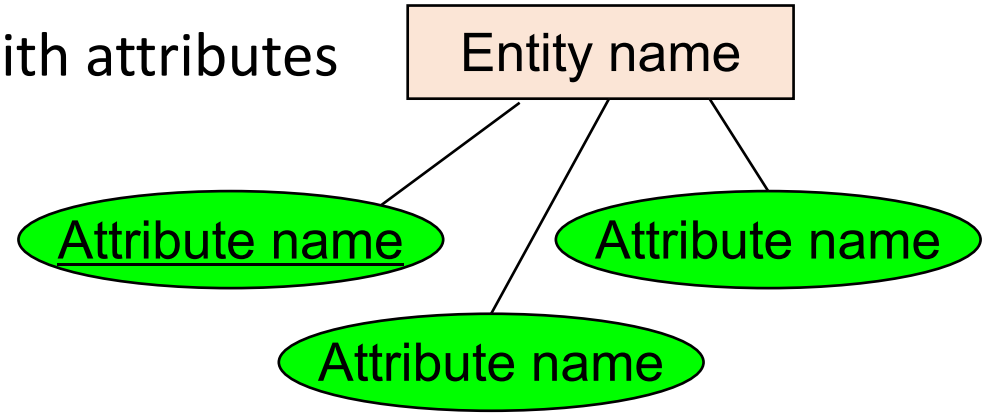
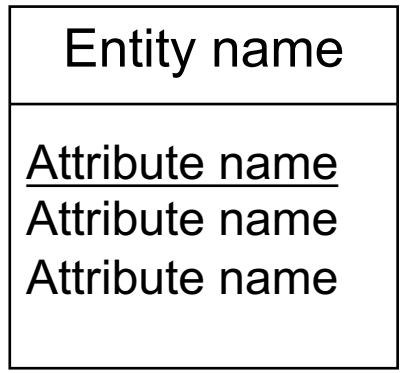
# Comparison of ERD frameworks

A variant of "UML"

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

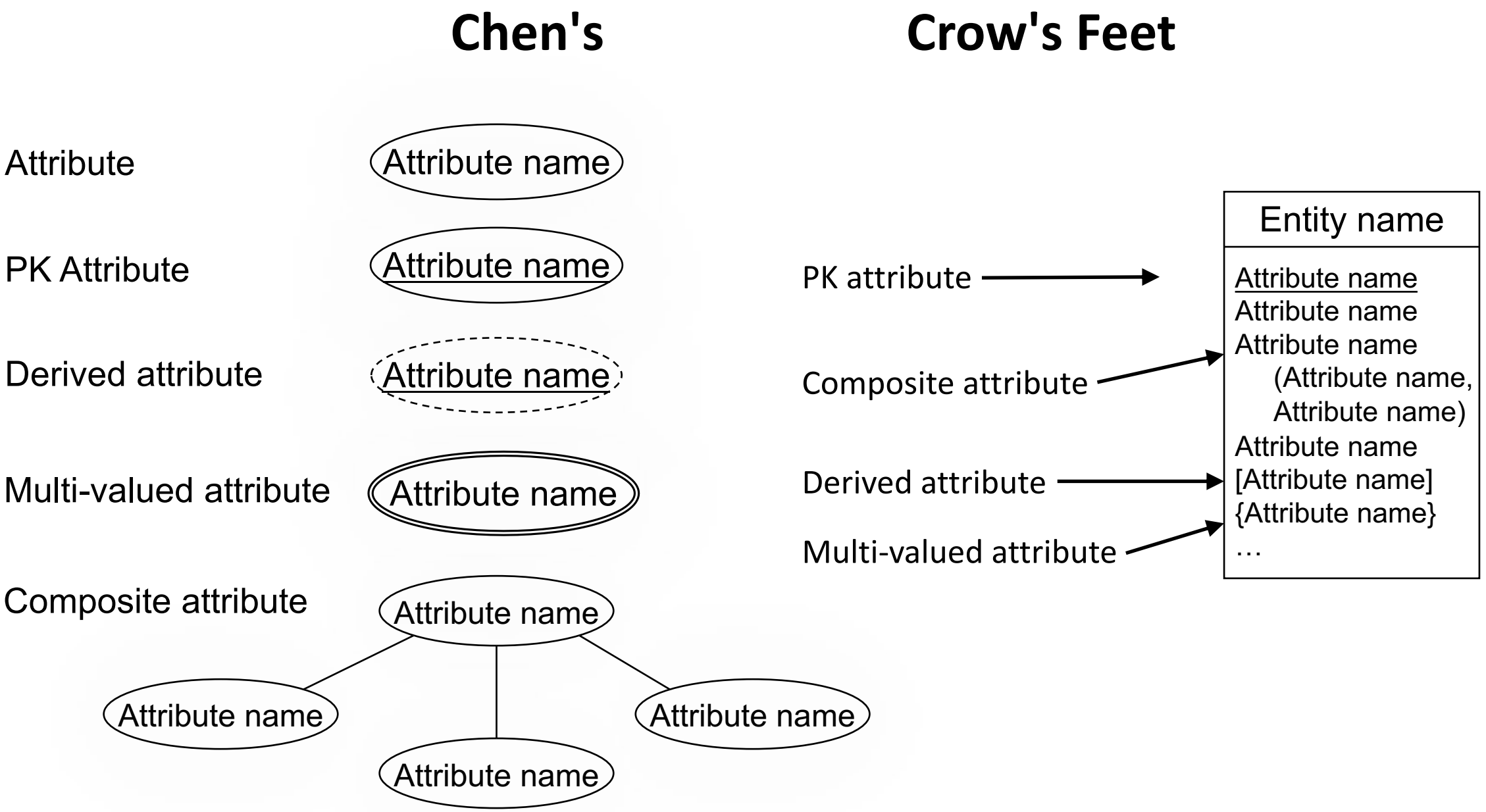
# Comparison of ERD frameworks

A variant of "UML"

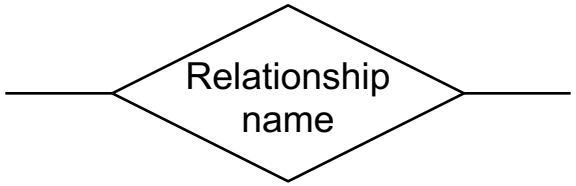
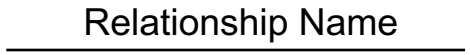
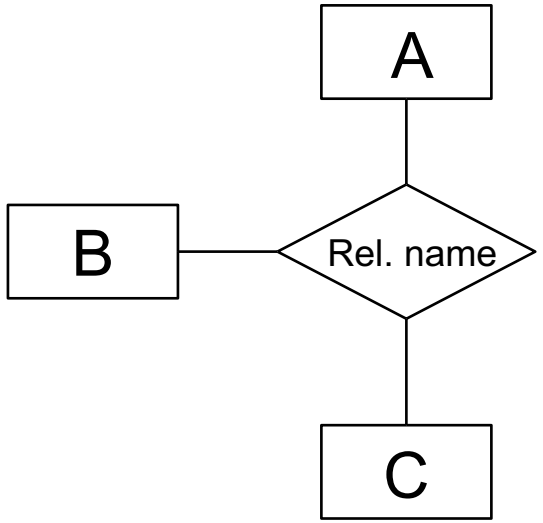
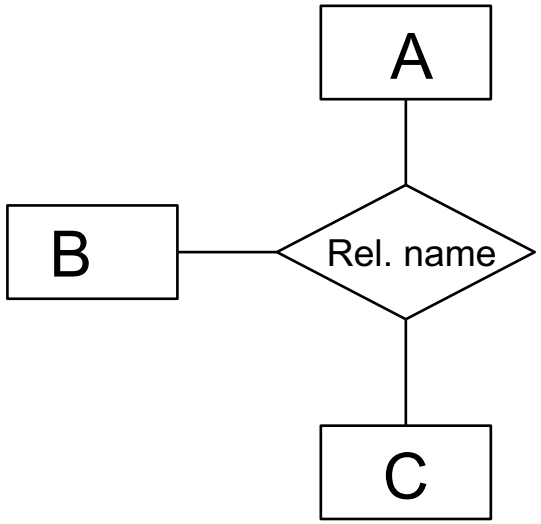
| | Chen's | Crow's Feet |
|---|---|---|
| Strong entity |   |   |
| Weak entity |   |   |
| Entity with attributes |   |   |

Color is not part of the standard…

# Attributes

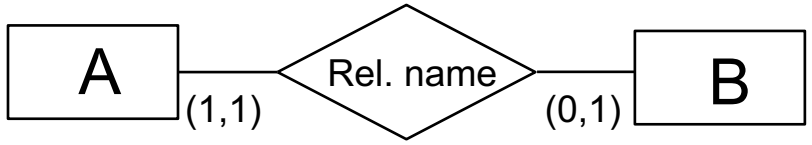

Chen's
Crow's Feet
Attribute
Attribute name
PK Attribute
Attribute name
Derived attribute
Attribute name
Multi-valued attribute
Attribute name
Composite attribute
Attribute name
Attribute name
Attribute name
Attribute name
PK attribute
Composite attribute
Derived attribute
Multi-valued attribute
Entity name
Attribute name
Attribute name
Attribute name
(Attribute name,
Attribute name)
Attribute name
[Attribute name]
{Attribute name}
…

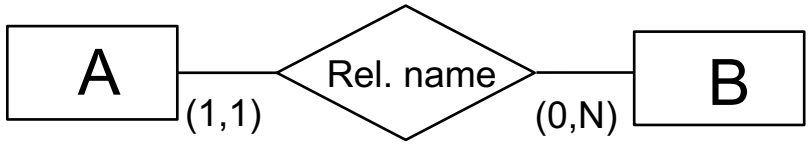
# Relationships

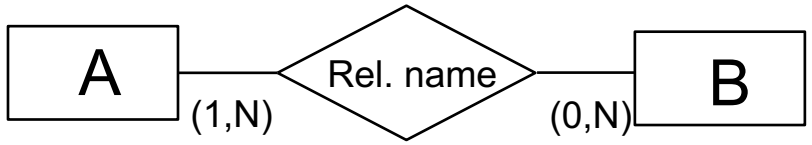
| | Chen's | Crow's Feet |
|---|---|---|
| Binary Relationship |   |   |
| Relationship of Higher Degree |   |   |

# Types of Binary Relationships

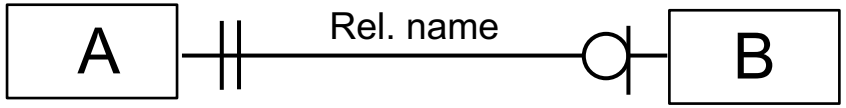
**Chen's**

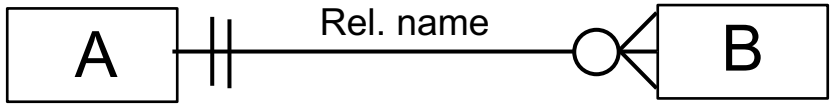


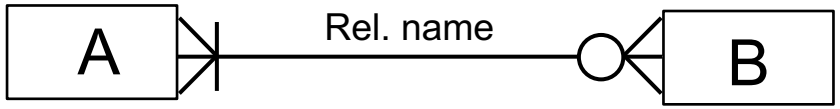



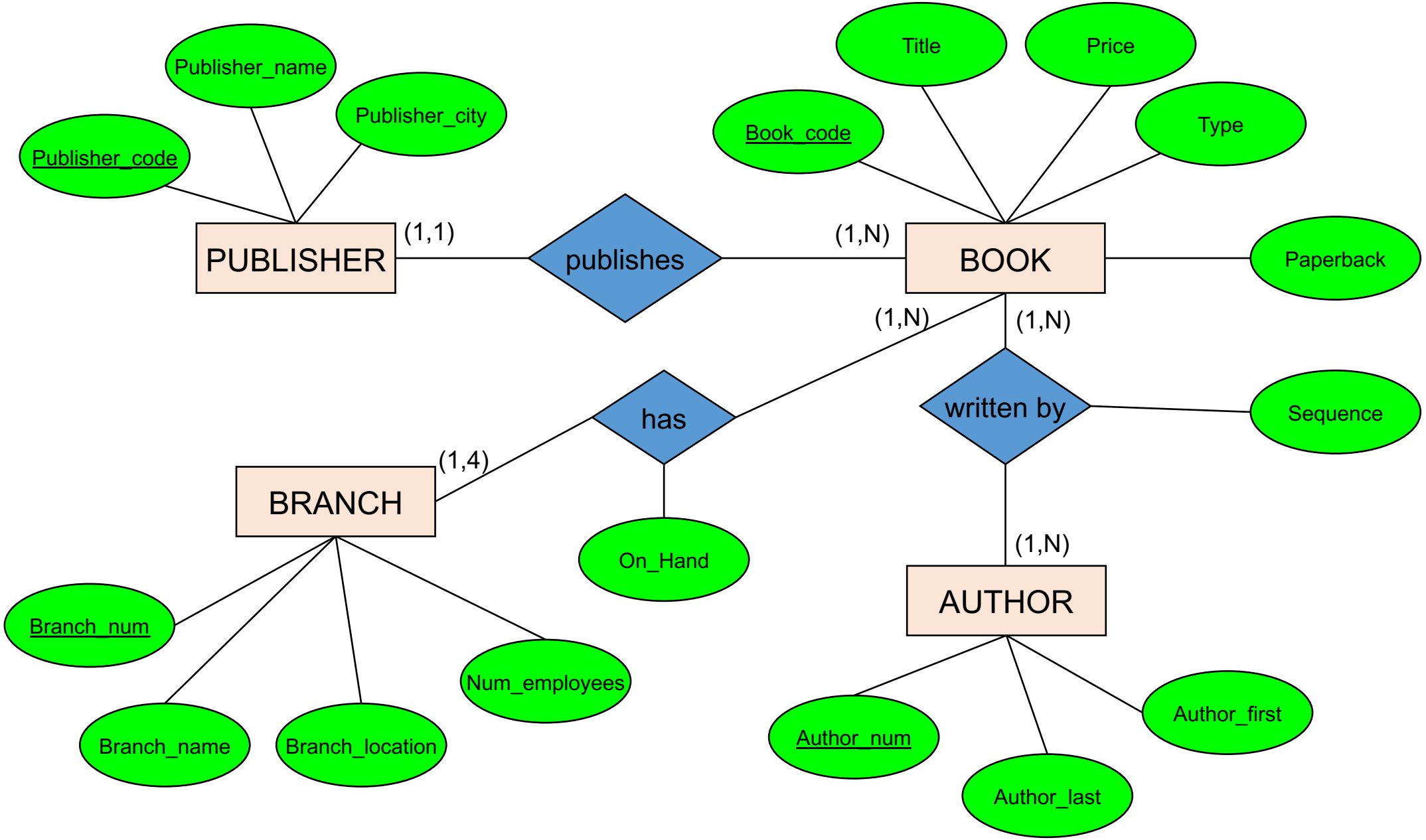

**Crow's Feet**

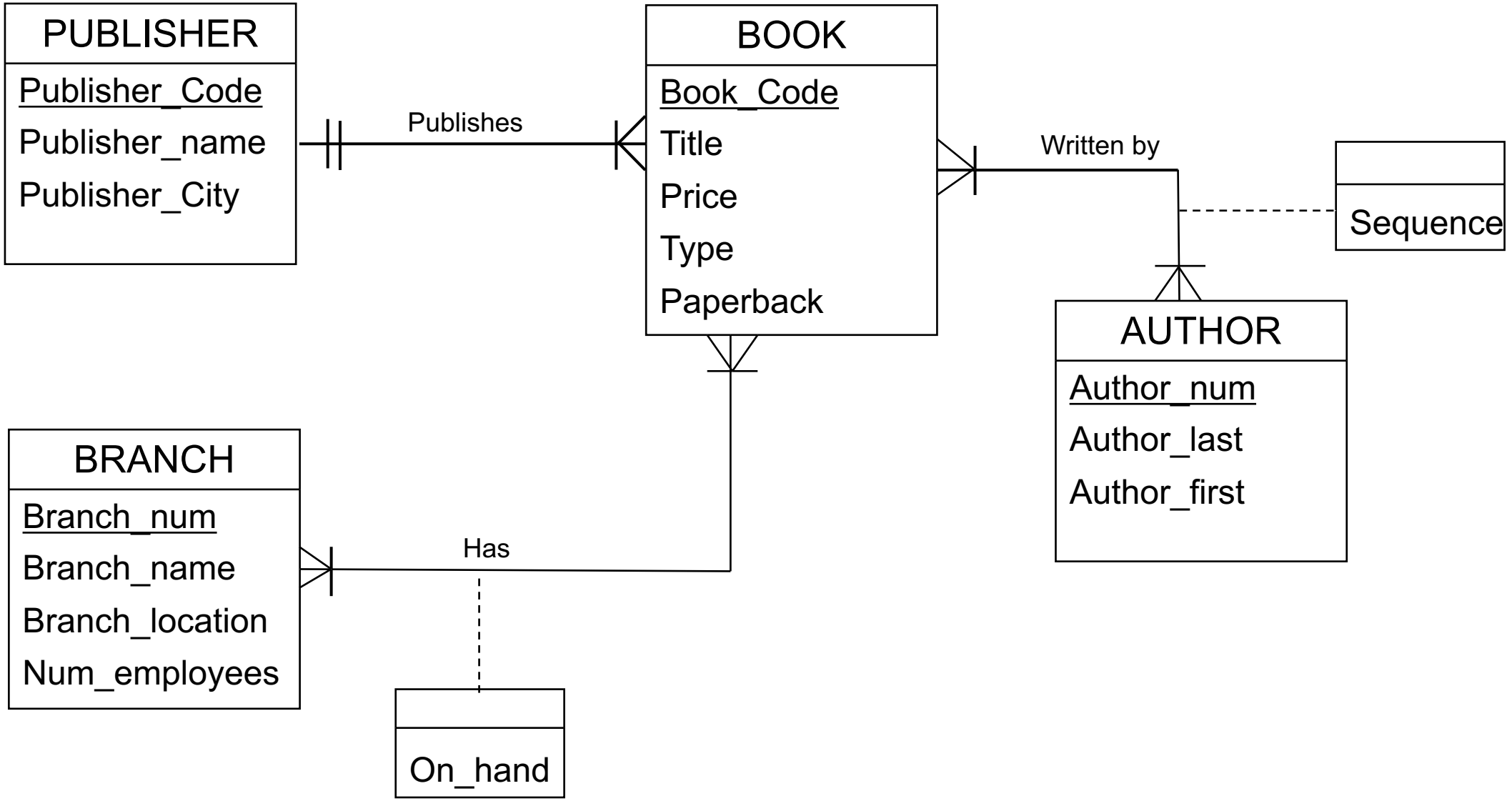


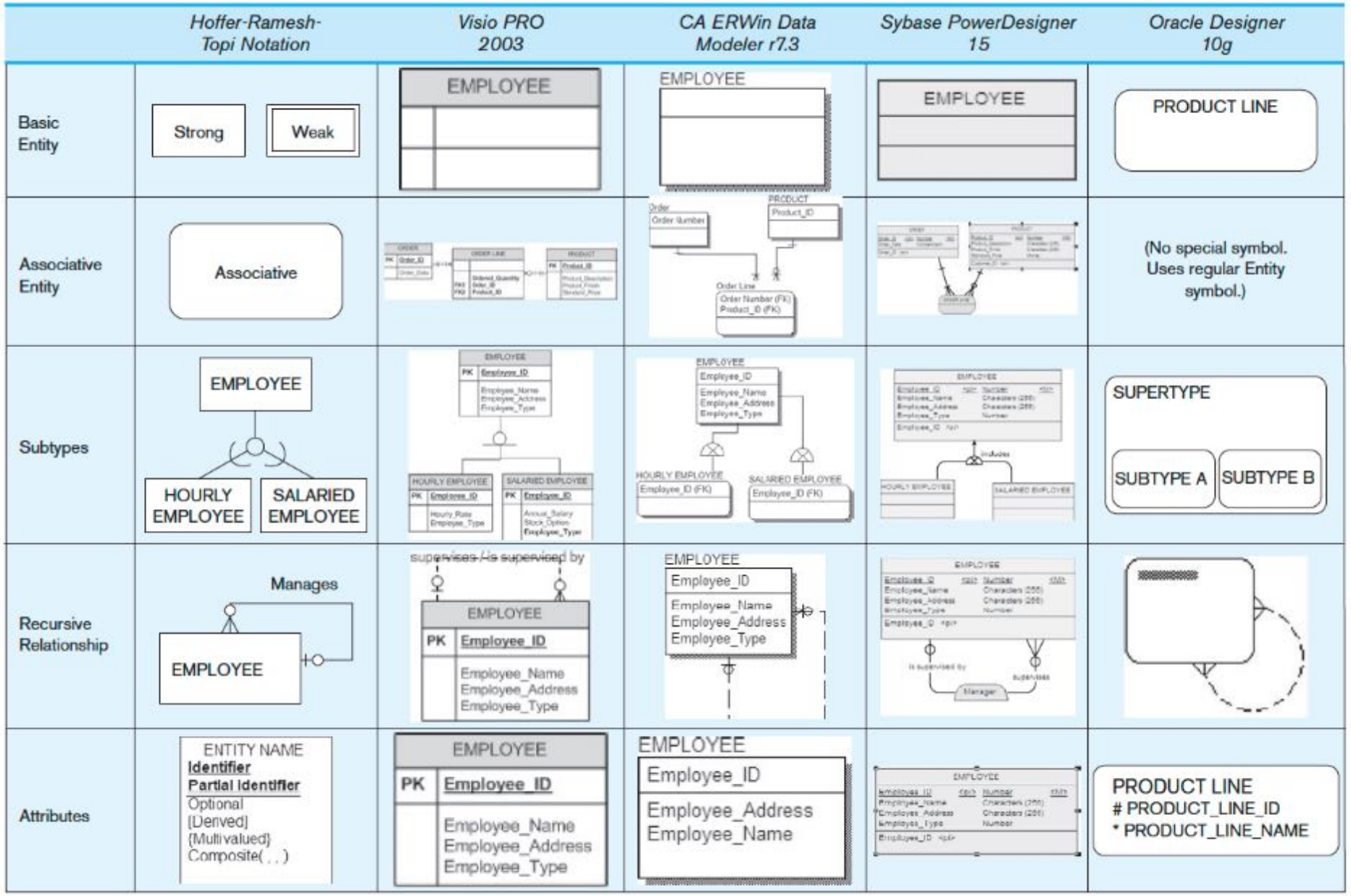
# Redo this ER diagram with Crow's feet notation






Source: Janusz Szczypula, 95703 Database Management, 2011.

# Modeling Notation ("Consistency beats brilliance")



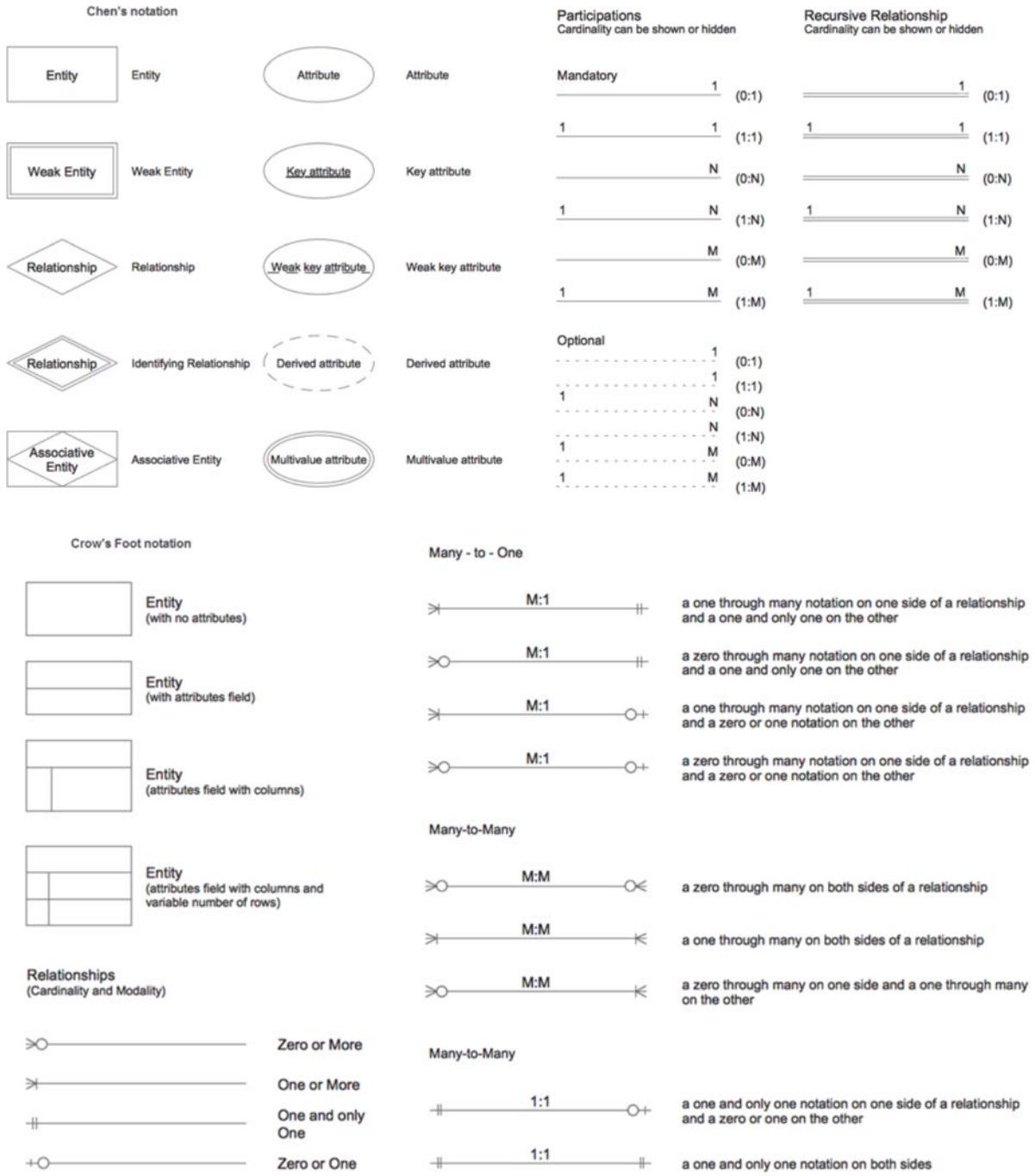

Source: Table A-1, Hoffer et al, 10th ed, 2010.

# 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 | |

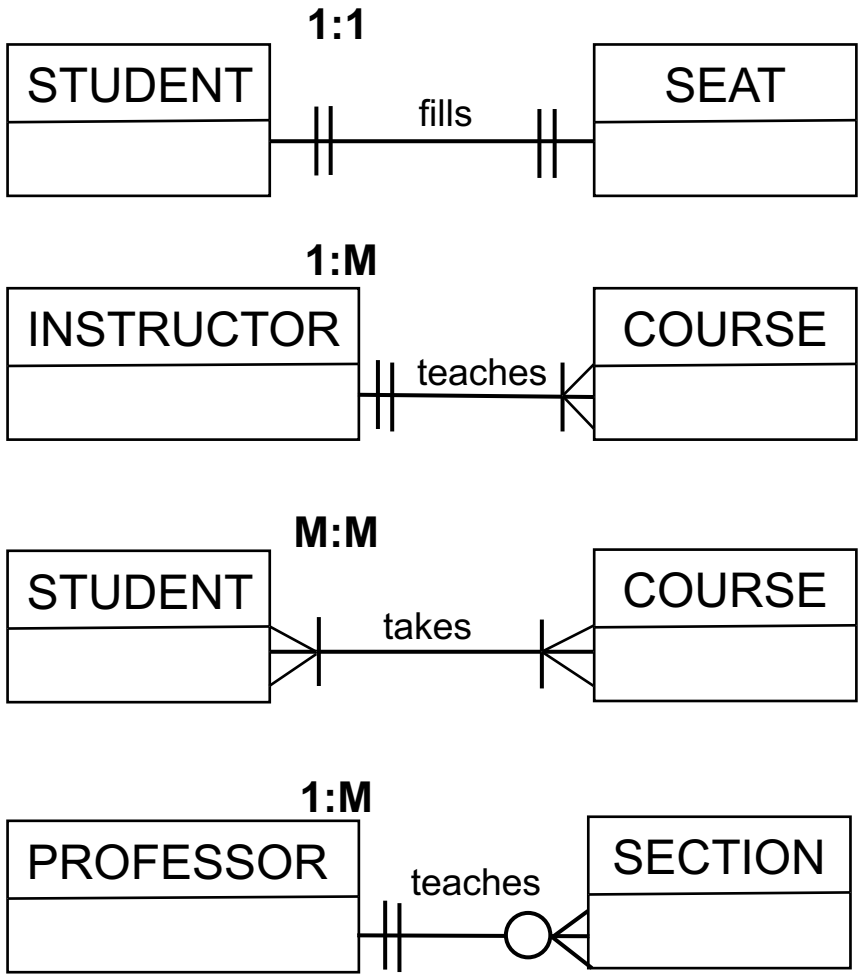
Source: Table A-1, Hoffer et al, 10th ed, 2010.

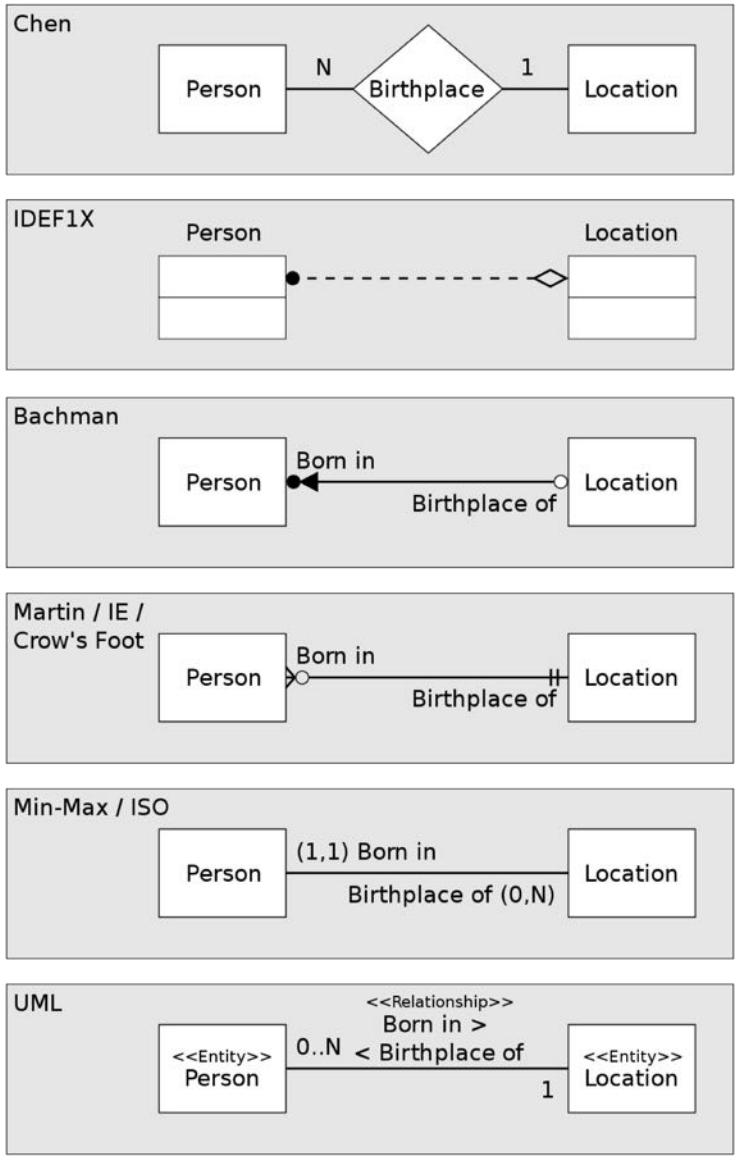
# Various Notations




Source: http://www.conceptdraw.com/How-To-Guide/erd-entity-relationship-diagram-symbols

# Crow's feet notation and alternatives

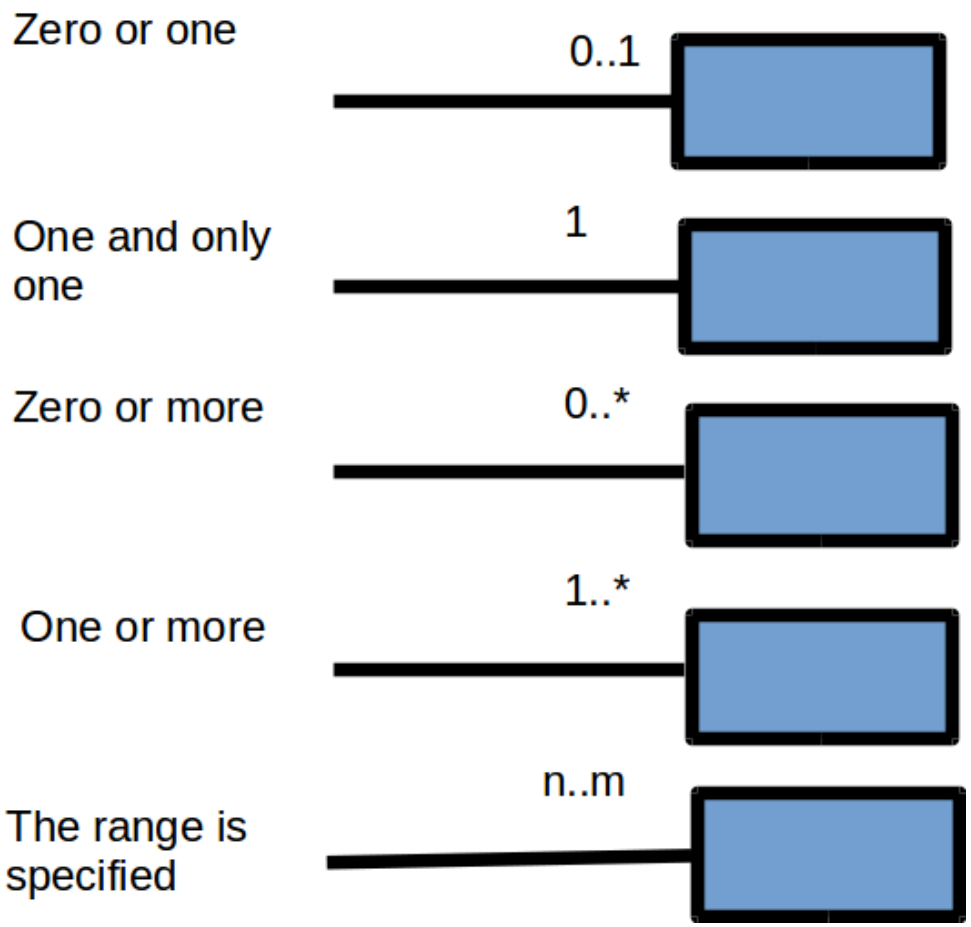


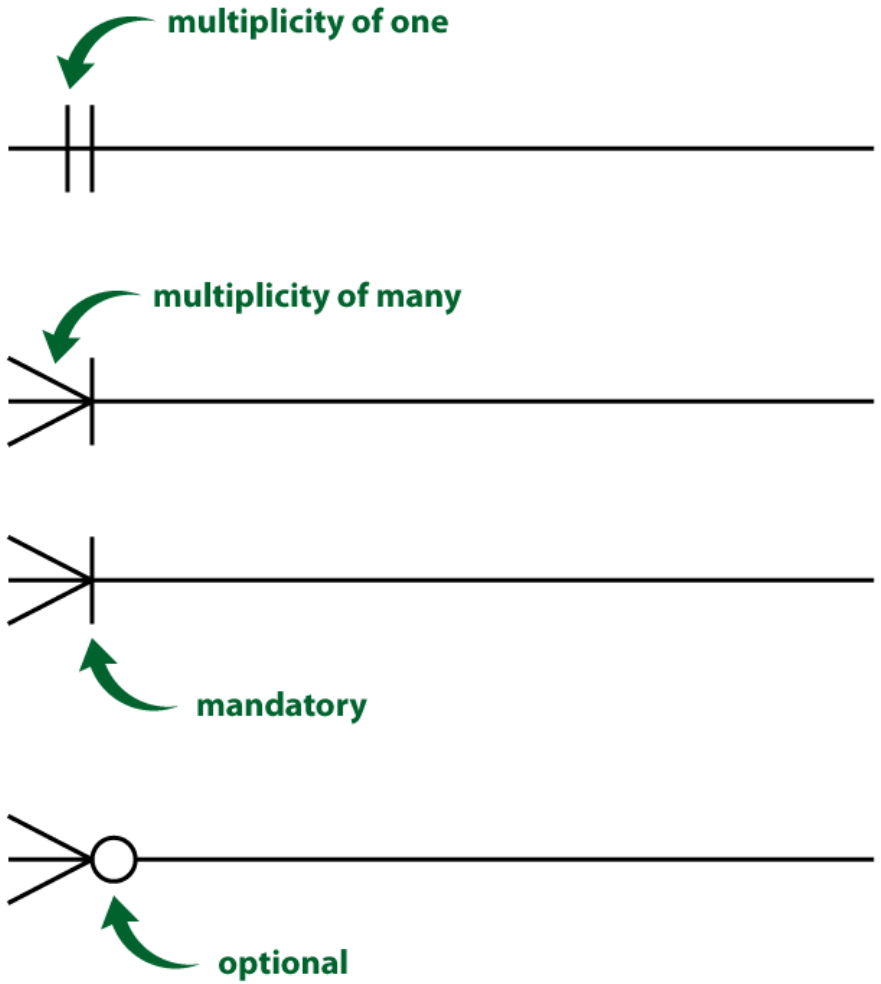

Source: http://en.wikipedia.org/wiki/Entity-relationship_model

# Relationships with specified cardinalities

## UML notation

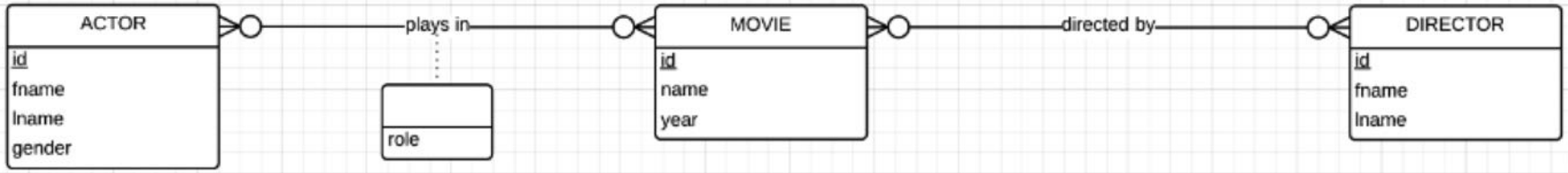



## Crow's Feet

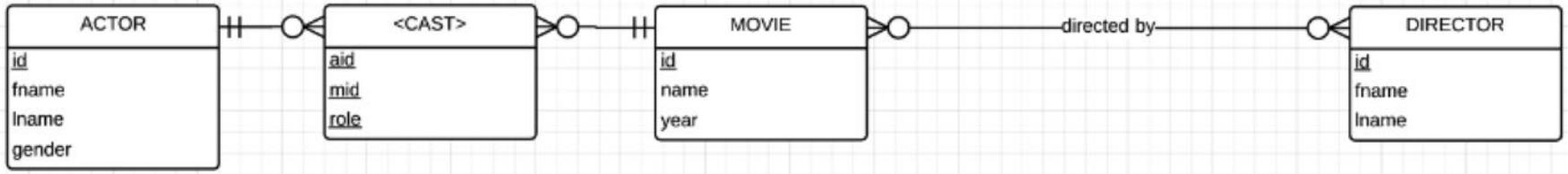


# IMDB movie database in Lucidchart

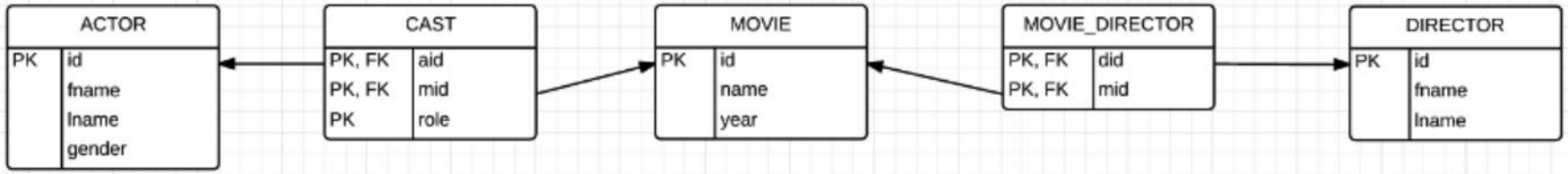
ER diagram: don't forget identifiers, but no FKs




ER diagram: CAST as associative entity can be justified



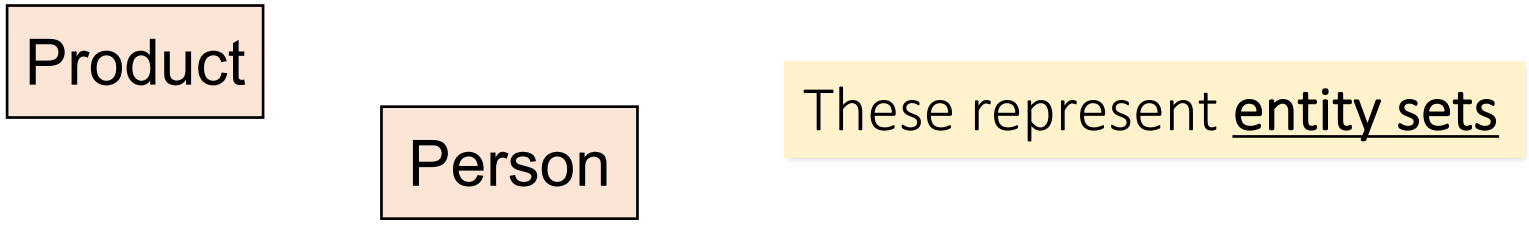

Relational schema: don't forget PKs and FKs



# Entities

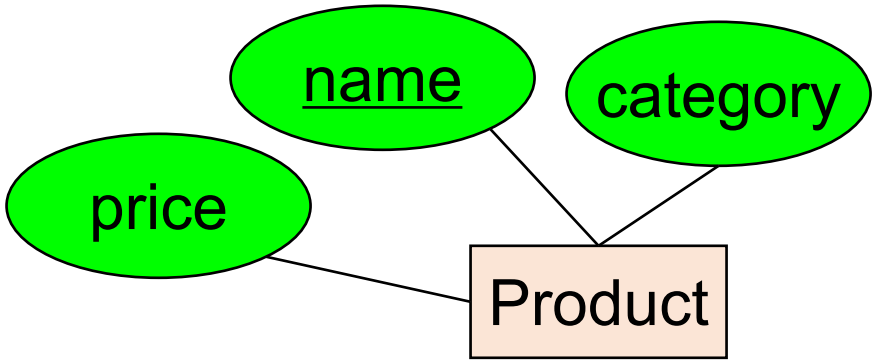
# Entities and Entity Sets

- 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

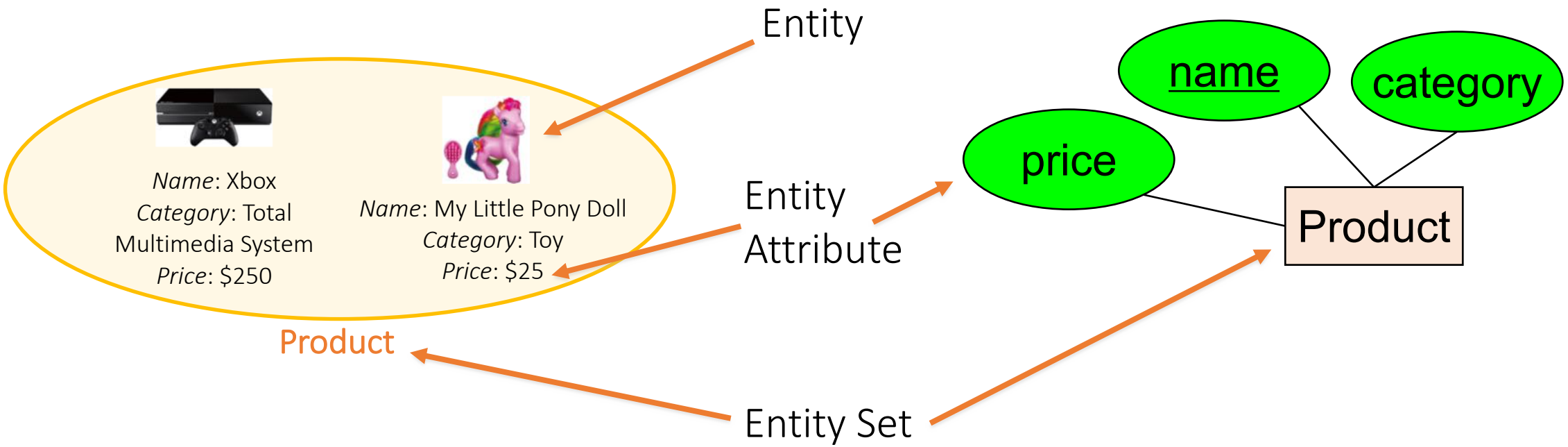



Shapes **are** important.
Colors **are not**.

# Entities vs. Entity Sets

- Example:

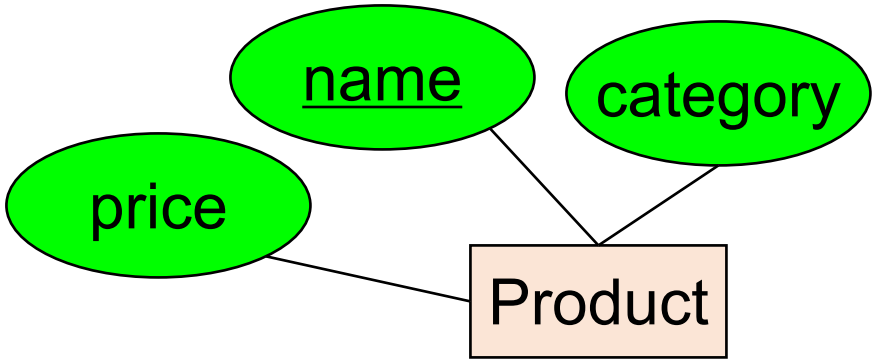
Entities are **not** explicitly represented in E/R diagrams!



# 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
    - 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 ("surrogate key")
- 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 | |

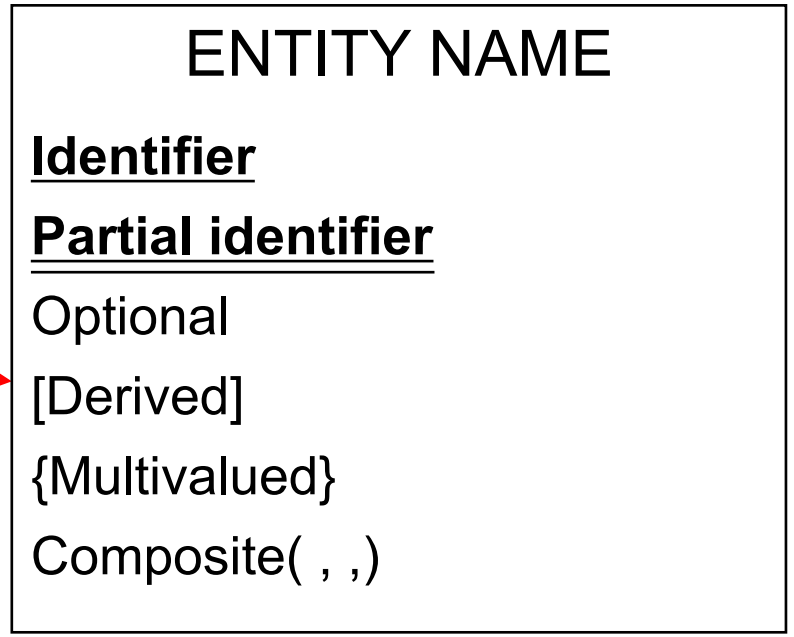
| COURSE ID | TITLE | INSTRUCTOR NAME | INSTRUCTOR LOCATION | GRADE |
|---|---|---|---|---|
| IS 350 | Database Mgt. | Codd | B104 | A |
| IS 465 | System Analysis | Parsons | B317 | B |

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

# Attributes

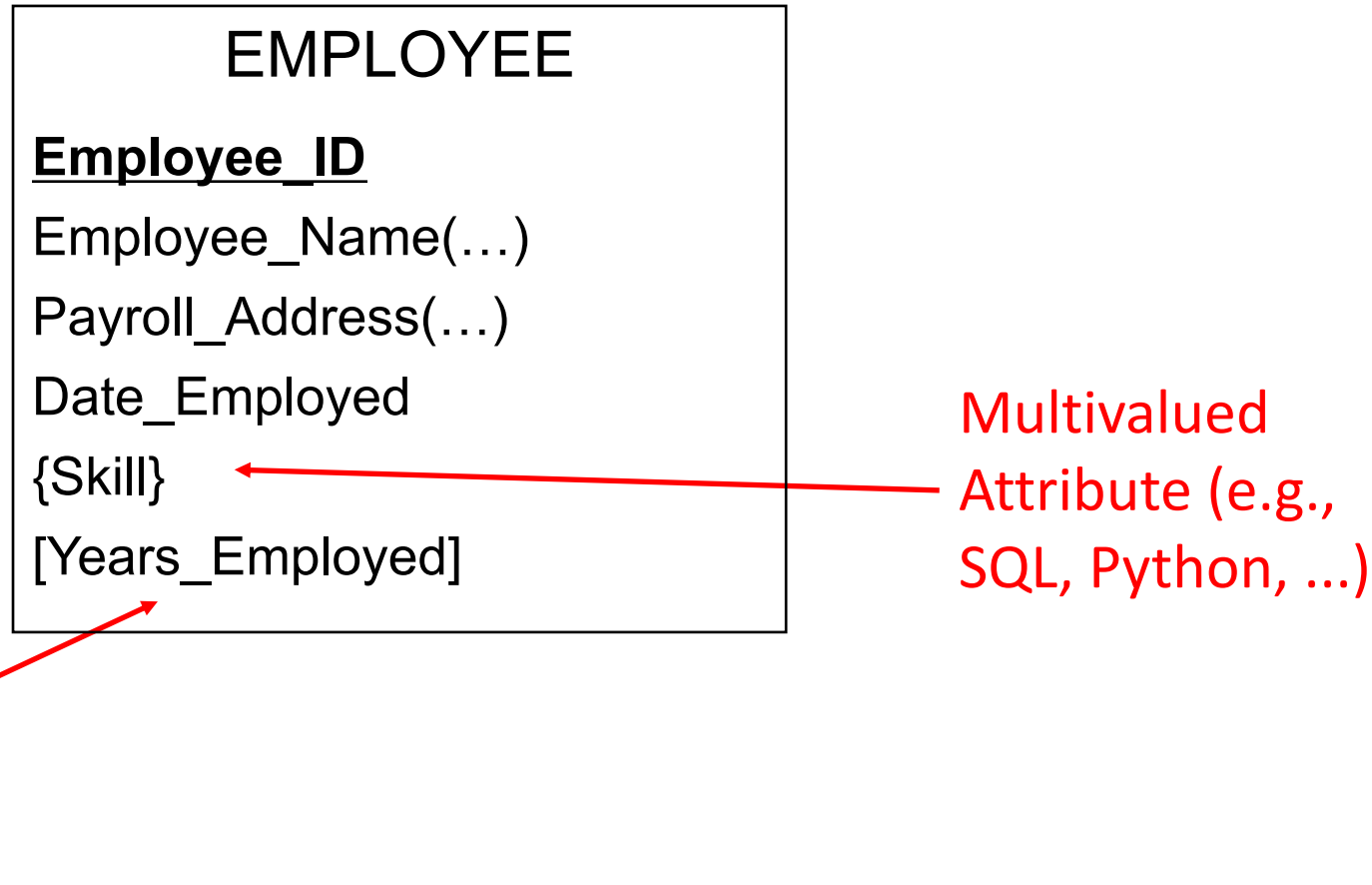
- A property or characteristic of an entity type

- 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




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**<br>**(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 |

# 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

. . .

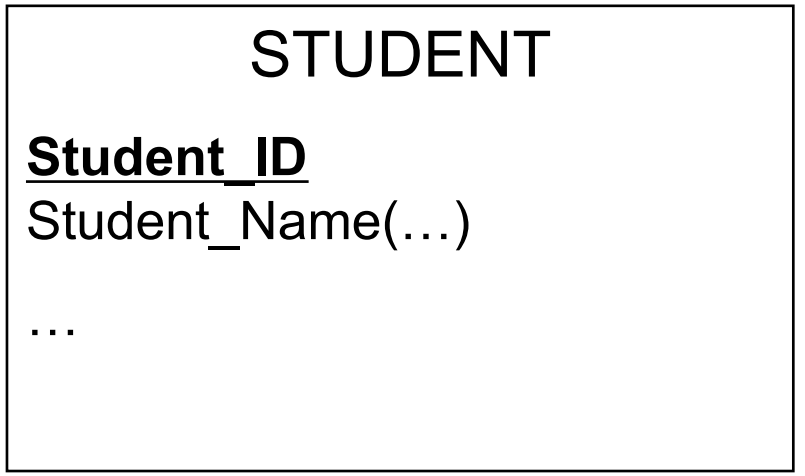
**US Airways Flight 1549**

The downed US Airways Flight 1549 floating on the Hudson River

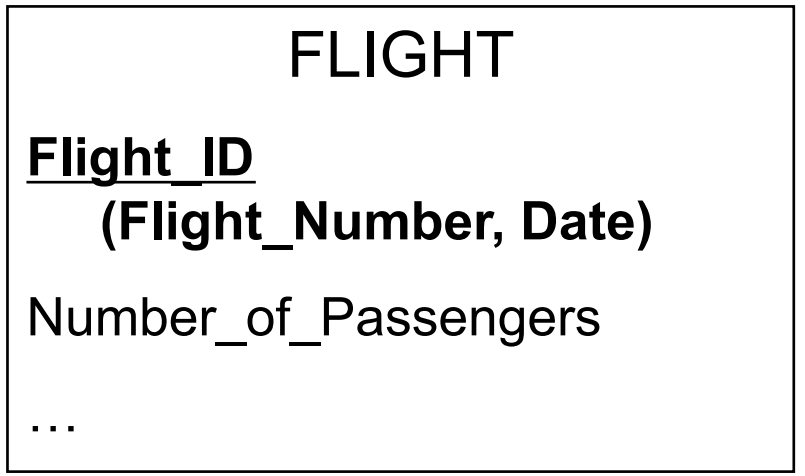
| Accident summary | |
|---|---|
| **Date** | January 15, 2009 |

# Identifier Examples: Simple and Composite

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



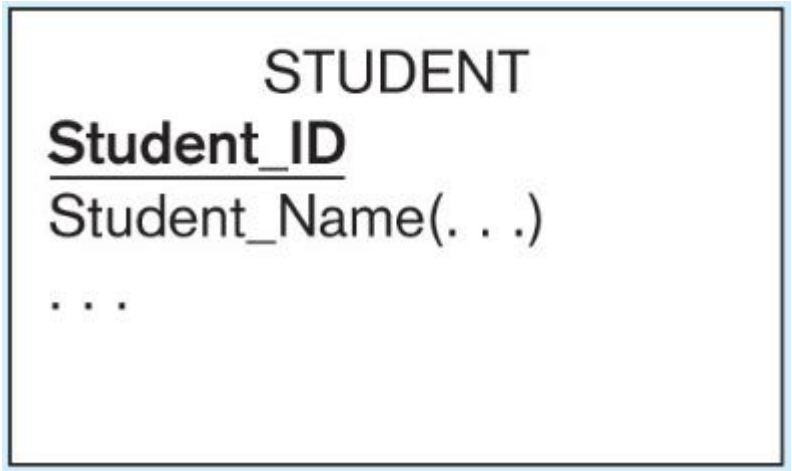

- Composite identifiers:
  - Multiple attributes required to uniquely identifies each entity instance
  - Identifier attribute underlined and composite attributes listed below in (parentheses)




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

# 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)

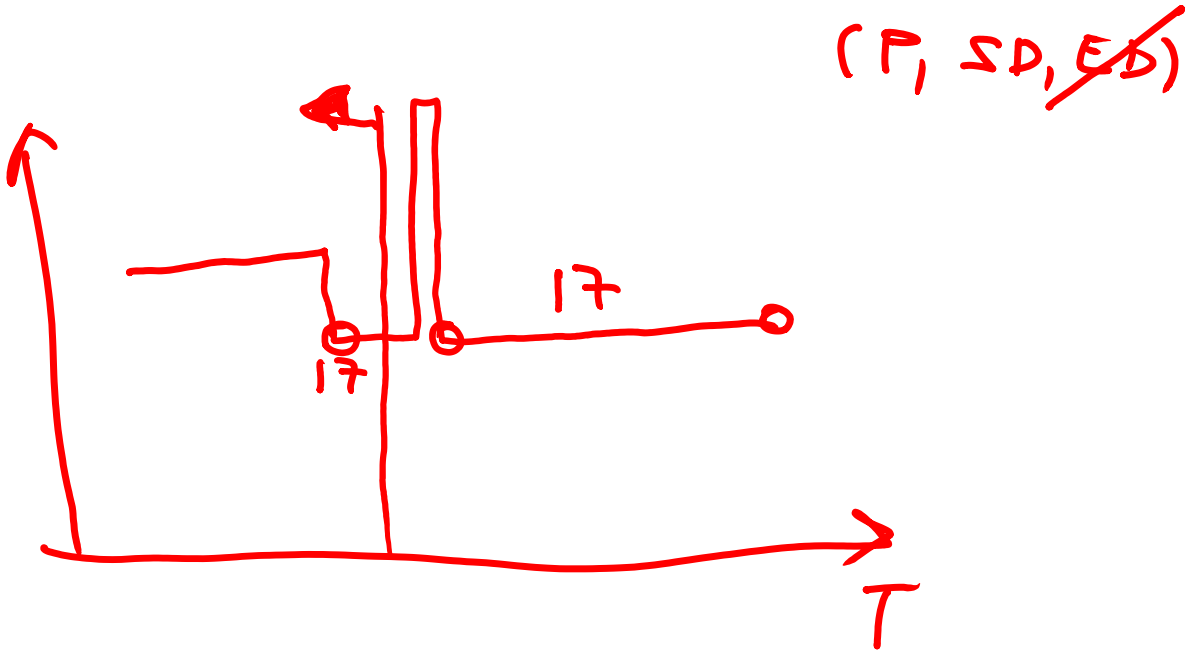

Source: Hoffer, Ramesh, Topi, "Modern database management," 10th ed, 2010.

# 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

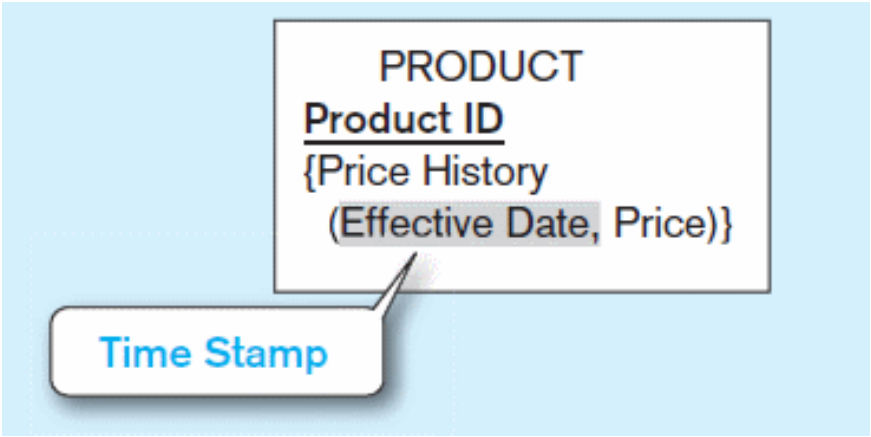
- Assume you have an entity "product"
- The price can change over time
- You would like to preserve the history of prices and the time period



# 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, {(No, '8), (Tu, 9, ...}



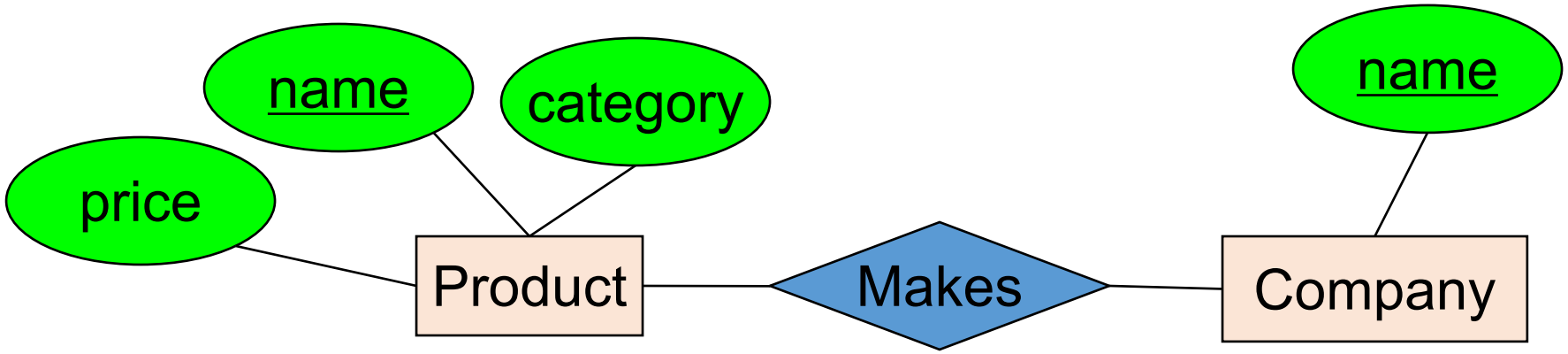

Time-stamping is commonly done with a multi-valued and composite attribute (or associative entities: see later)

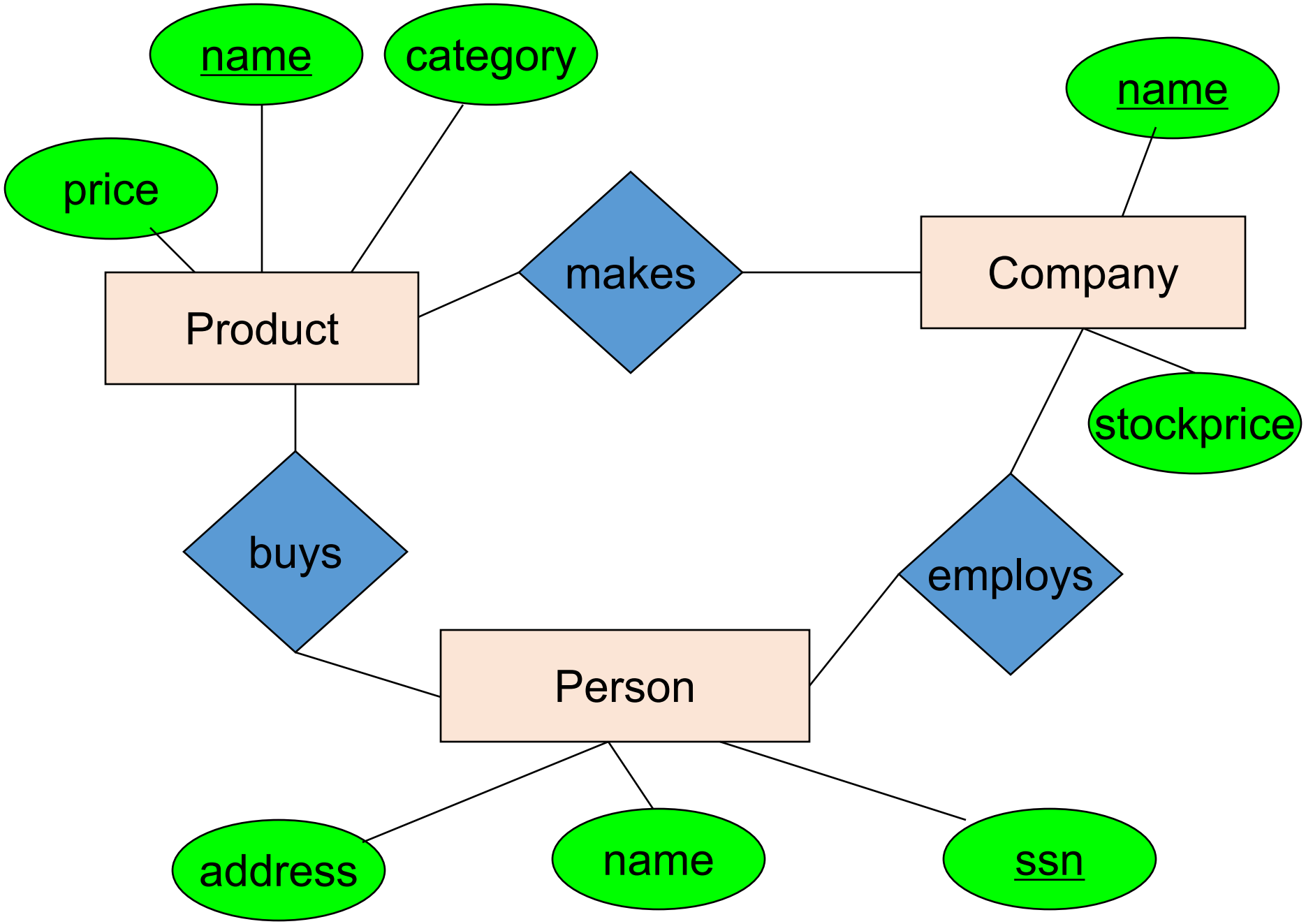
Source: Hoffer, Ramesh, Topi, "Modern database management," 10th ed, 2010.

# Relationships

# The R in E/R: Relationships

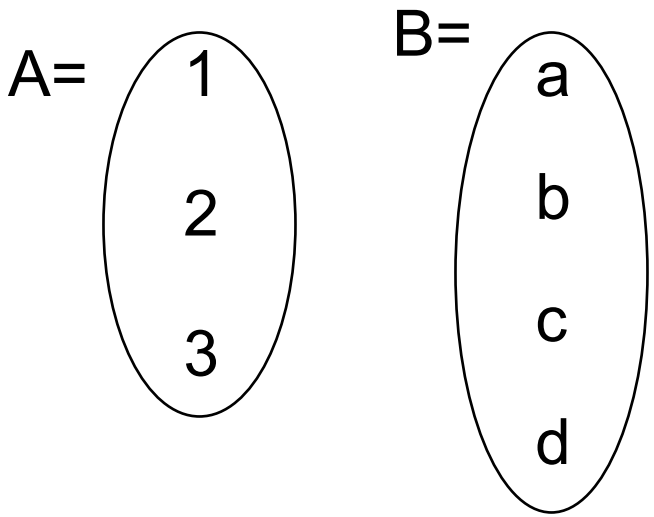
- A relationship is between two or more entities




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

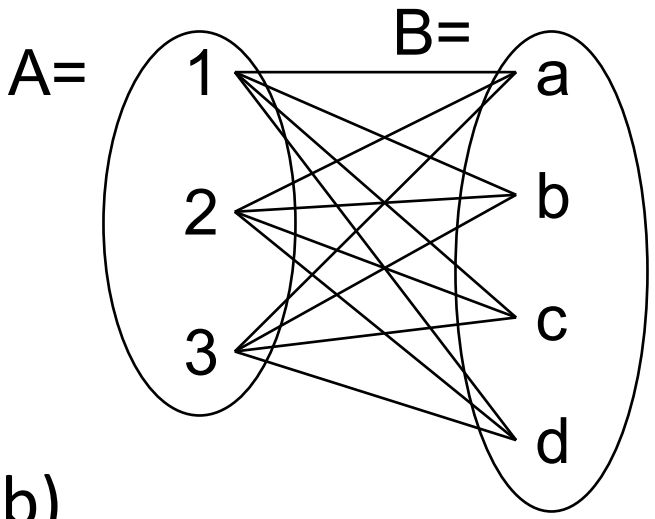
# What is a Relationship?

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



# What is a Relationship?

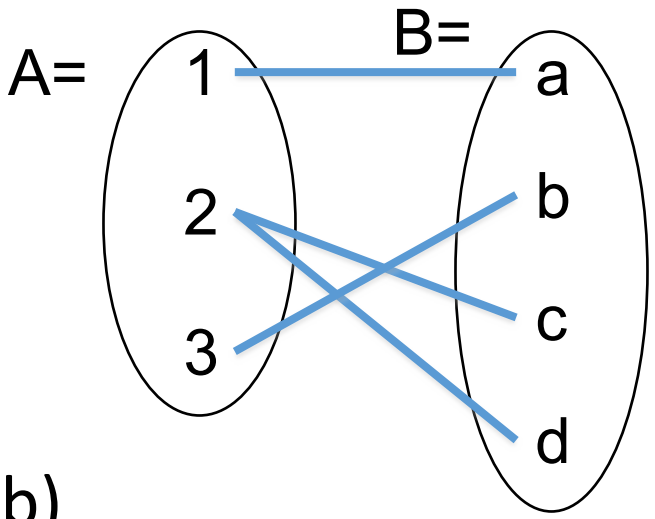
- A mathematical definition:
  - Let A, B be sets
    - A={1,2,3},  B={a,b,c,d}




  - 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?

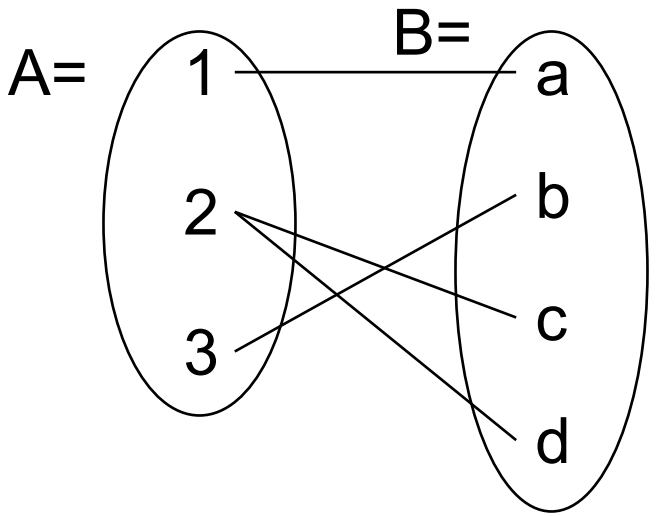
- A mathematical definition:




  - 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: