ER Model

CS411 Database Systems

Kevin C. Chang

The New Contract on Lecture: Students

- Students:
  - Please attend class and participate.
  - Please sit in the front rows so we are together.
  - Please interact with instructor (signal, ask, answer).
  - Please do not fall asleep or …

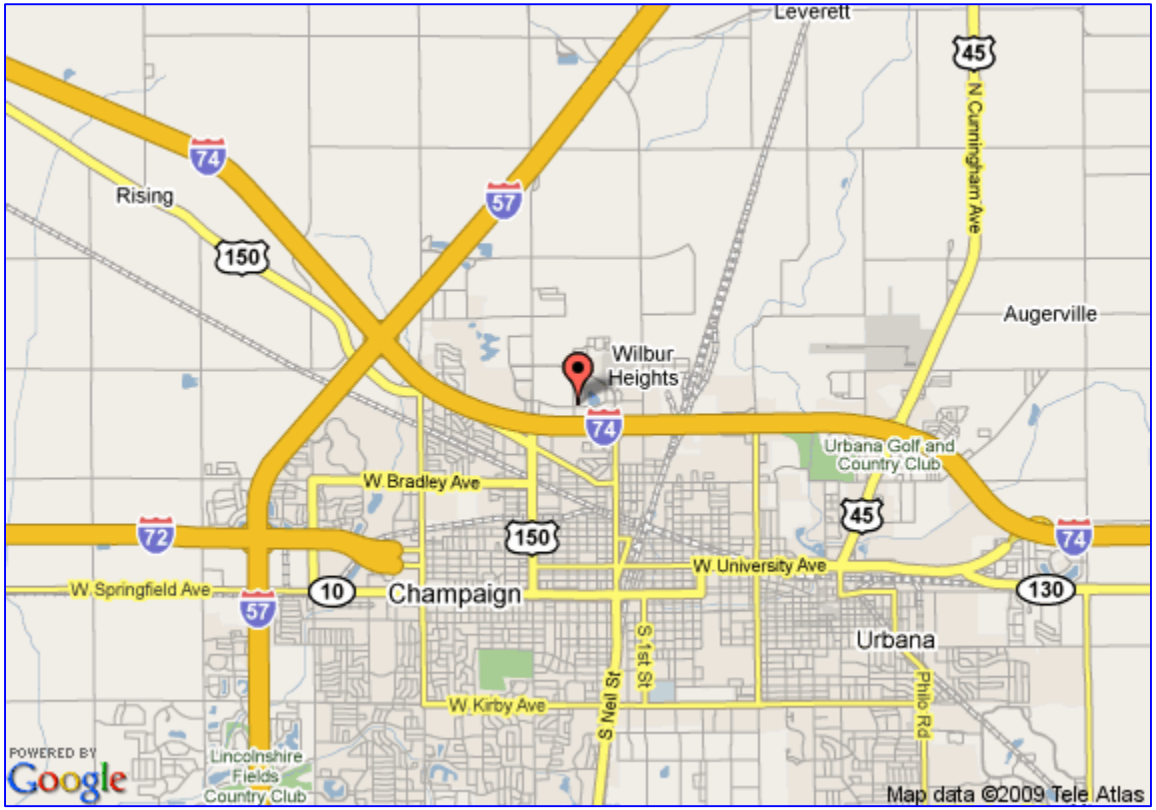
The New Contract on Lecture: Instructor

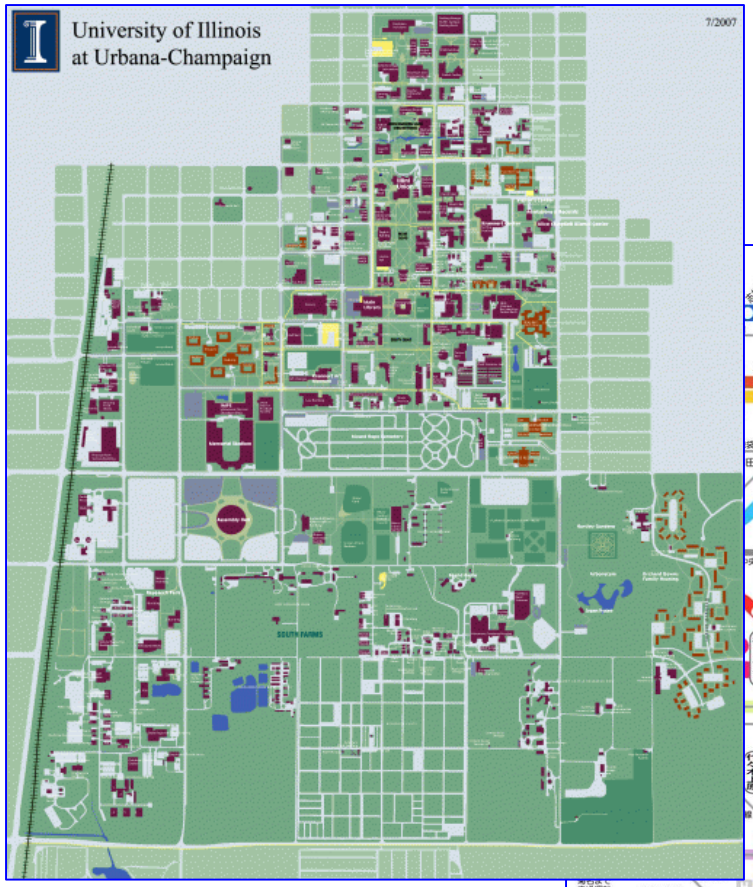
- Instructor:
  - Will be do my best to prepare.
  - Will respect each question.
  - Will not rush to cover all the materials.
  - Will make sure online students hear well.
  - Will not fall asleep or …

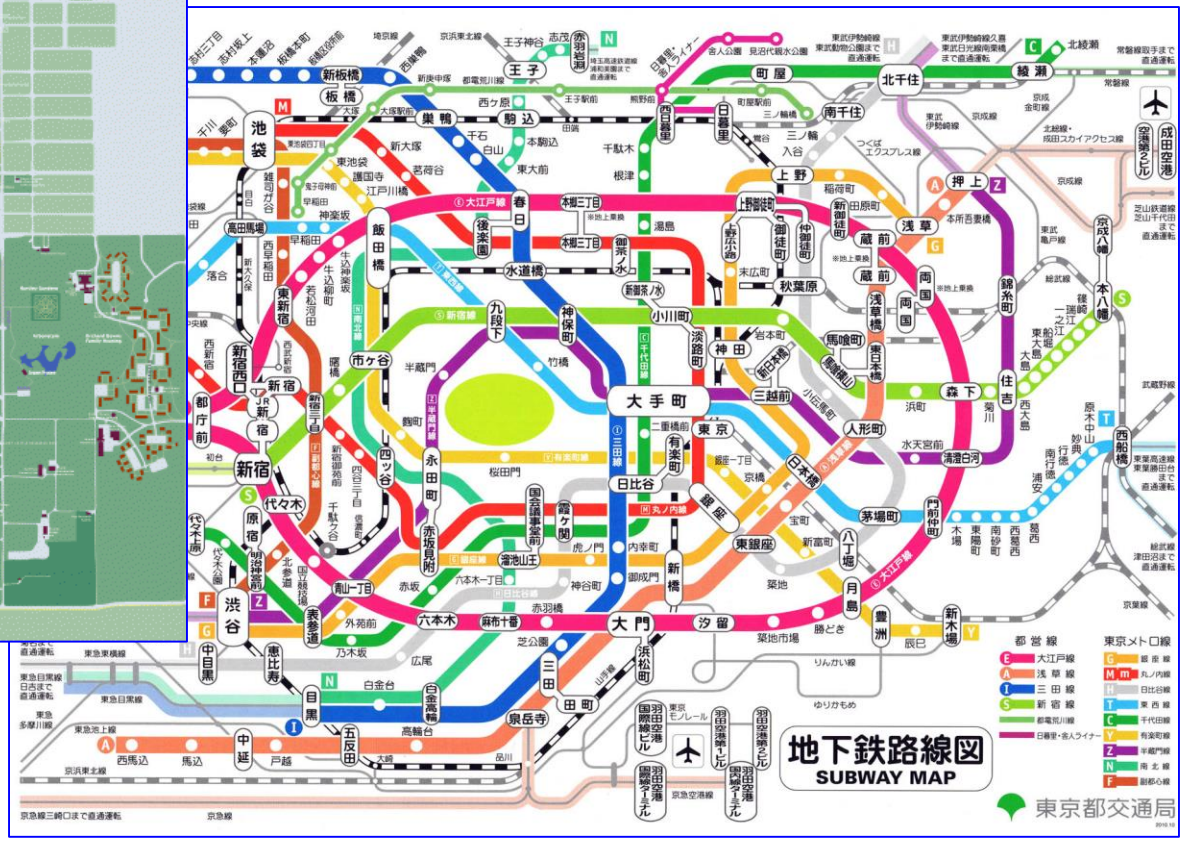
Why Do We Learn This?

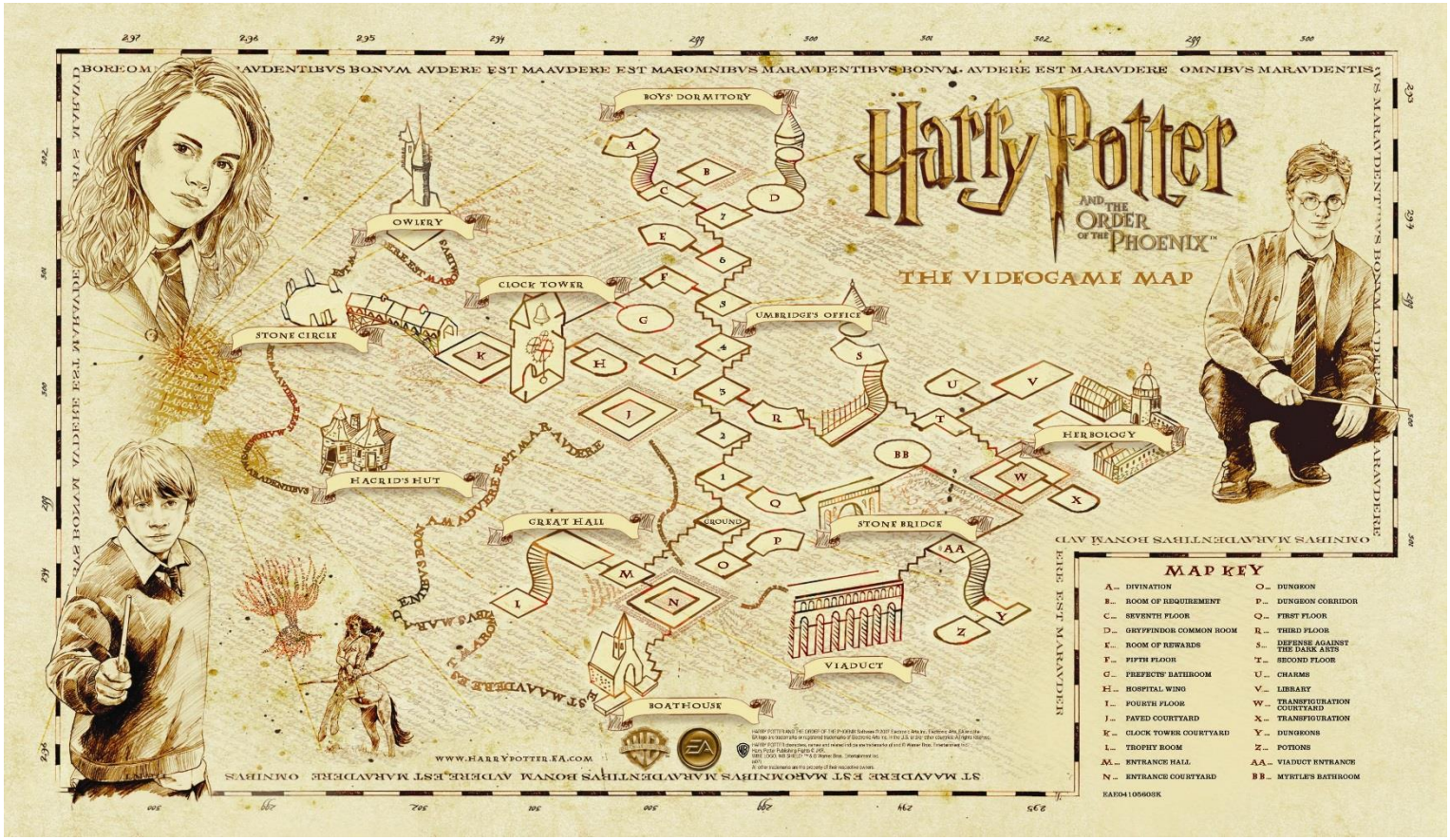
Why Do We Learn This?

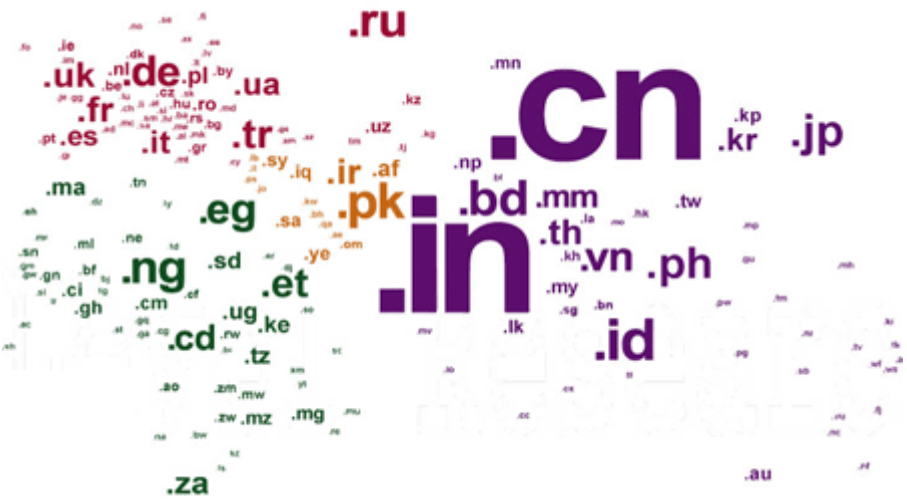
Why Do We Learn This?

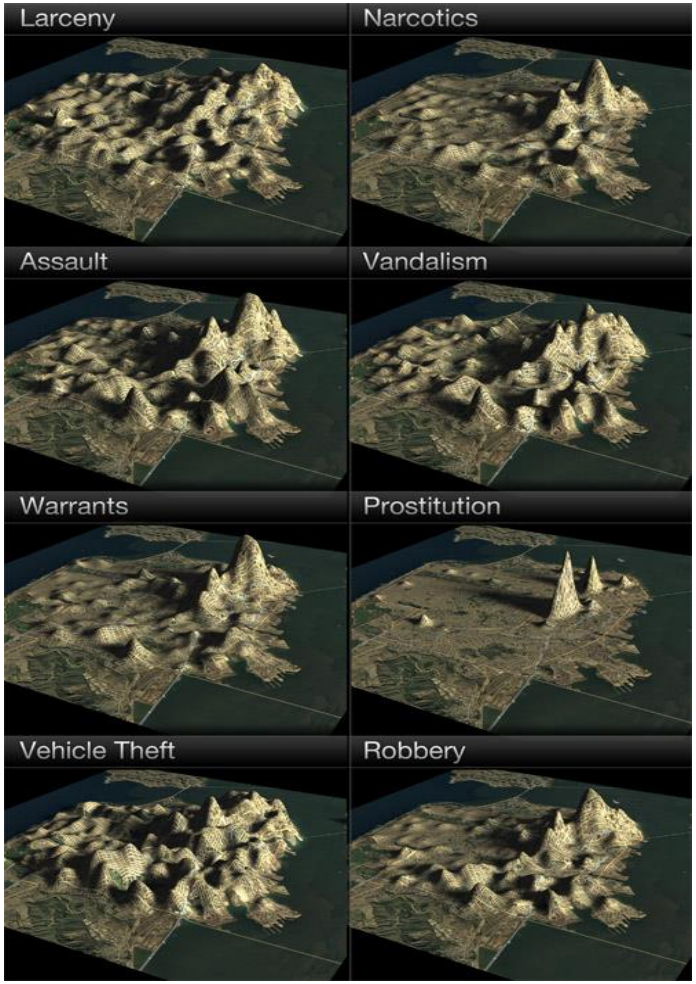



University of Illinois
at Urbana-Champaign
7/2007


地下鉄路線図
SUBWAY MAP
東京都交通局


Harry Potter
AND THE ORDER OF THE PHOENIX™
THE VIDEOGAME MAP
BOYS' DORMITORY
OWLERY
CLOCK TOWER
UMBRIDGE'S OFFICE
STONE CIRCLE
HERBOLOGY
HAGRID'S HUT
GREAT HALL
STONE BRIDGE
VIADUCT
BOATHOUSE
MAP KEY
A... DIVINATION
B... ROOM OF REQUIREMENT
C... SEVENTH FLOOR
D... GRYFFINDOR COMMON ROOM
E... ROOM OF REWARDS
F... FIFTH FLOOR
G... PREFECTS' BATHROOM
H... HOSPITAL WING
I... FOURTH FLOOR
J... PAVED COURTYARD
K... CLOCK TOWER COURTYARD
L... TROPHY ROOM
M... ENTRANCE HALL
N... ENTRANCE COURTYARD
O... DUNGEON
P... DUNGEON CORRIDOR
Q... FIRST FLOOR
R... THIRD FLOOR
S... DEFENSE AGAINST THE DARK ARTS
T... SECOND FLOOR
U... CHARMS
V... LIBRARY
W... TRANSFIGURATION COURTYARD
X... TRANSFIGURATION
Y... DUNGEONS
Z... POTIONS
AA... VIADUCT ENTRANCE
BB... MYRTLE'S BATHROOM
EAE041056038
WWW.HARRYPOTTER.EA.COM


.ru
.cn
.in
.de
.uk
.fr
.es
.it
.pl
.ua
.tr
.jp
.kr
.kp
.pk
.ir
.bd
.mm
.th
.vn
.ph
.eg
.ng
.et
.id
.cd
.ug
.ke
.tz
.za
.au


Larceny
Narcotics
Assault
Vandalism
Warrants
Prostitution
Vehicle Theft
Robbery

Steps in Building a DB Application

- Suppose you are working on CS411 project
- Step 0: pick an application domain
  - we will talk about this later
- Step 1: conceptual design
  - discuss with your team mates what to model in the application domain
  - need a modeling language to express what you want
  - ER model is the most popular such language
  - output: an ER diagram of the app. domain

Steps in Building a DB Application

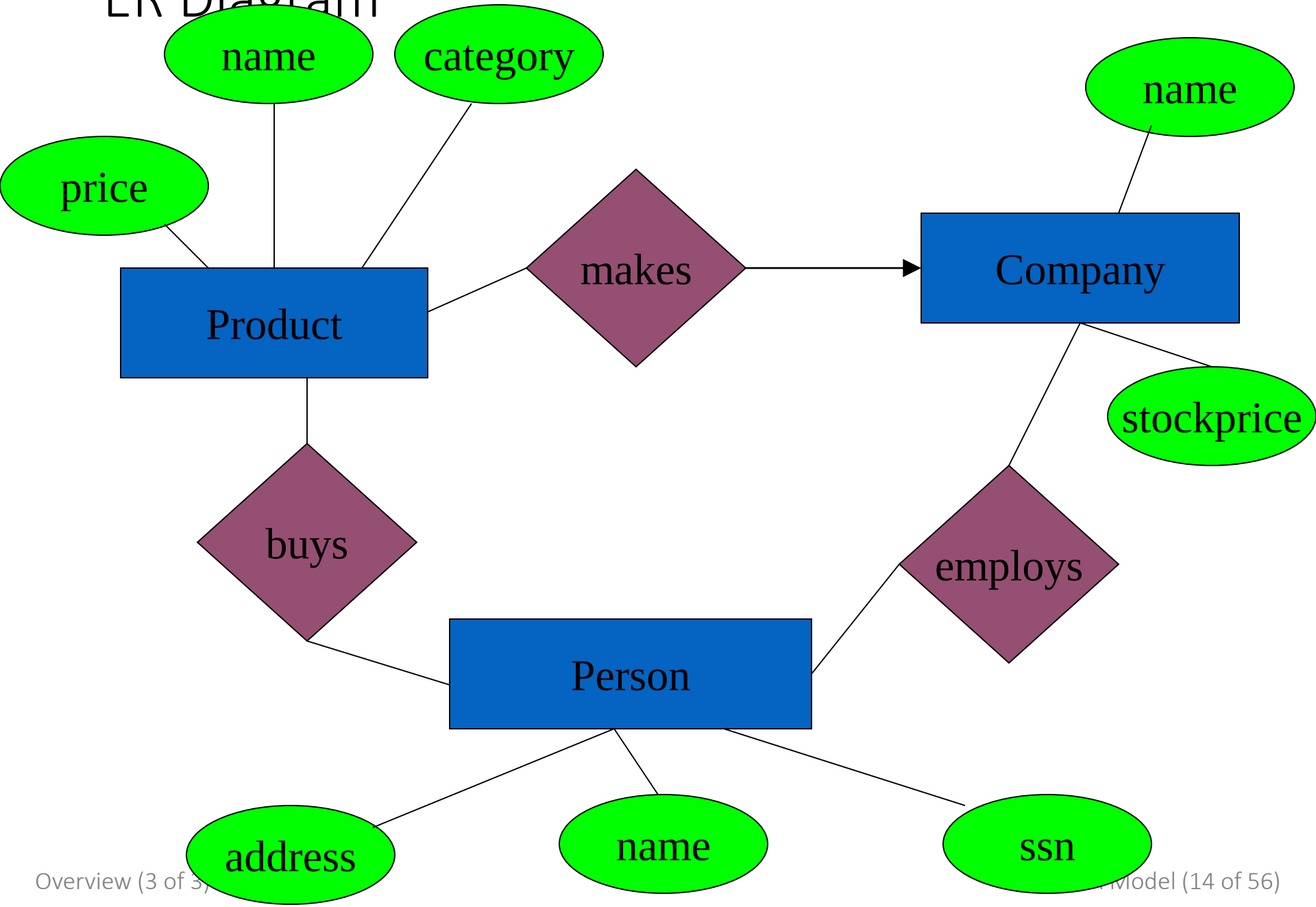
- Step 2: pick a type of DBMS
  - relational DBMS is most popular and is our focus
- Step 3: translate ER design to a relational schema
  - use a set of rules to translate from ER to rel. schema
  - use a set of schema refinement rules to transform the above rel. schema into a good rel. schema
- At this point
  - you have a good relational schema on paper
- And then …

ER Model

ER Model

- Gives us a language to specify
  - what information the db must hold
  - what are the relationships among components of that information
- Proposed by Peter Chen in 1976
- What we will cover
  - basic stuff
  - constraints
  - weak entity sets
  - design principles

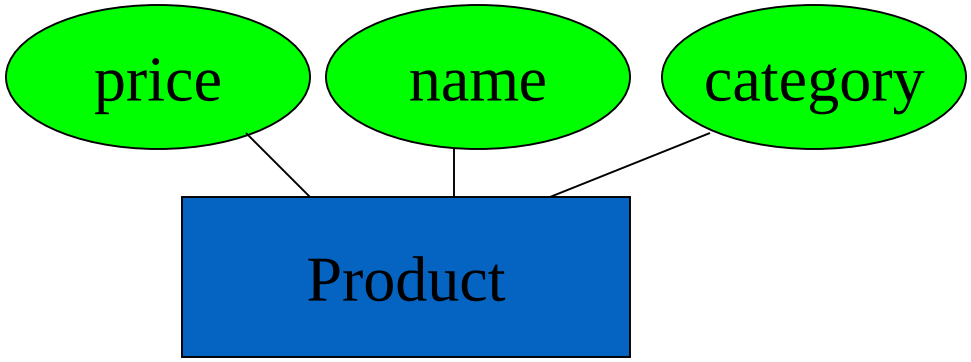
ER Diagram


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

Entities, Attributes, Relationships

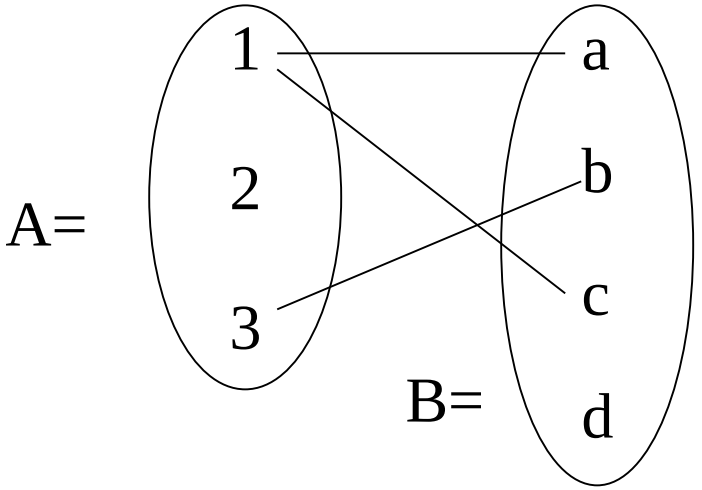
Entities and Attributes

- Entities
  - real-world objects distinguishable from other objects
  - described using a set of attributes
- Attributes
  - each has an atomic domain: string, integers, reals, etc.
- Entity set: a collection of similar entities

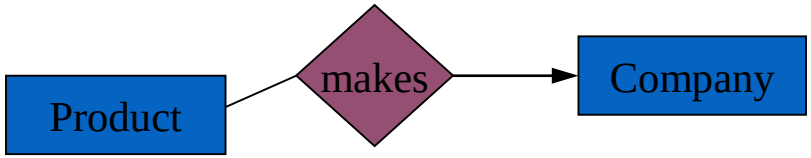


Relationships

- A mathematical definition:
  - if A, B are sets, then a relation R is a subset of A x B
- A={1,2,3}, B={a,b,c,d},
  - R = {(1,a), (1,c), (3,b)}

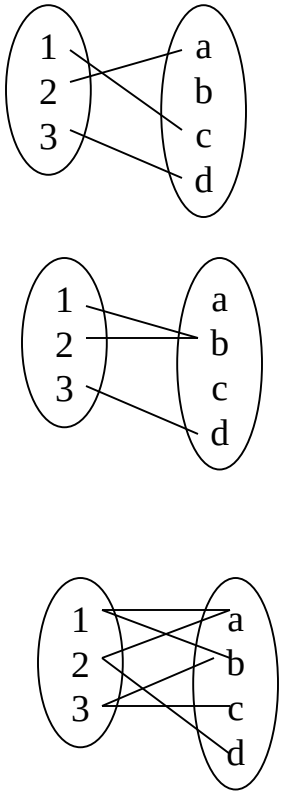


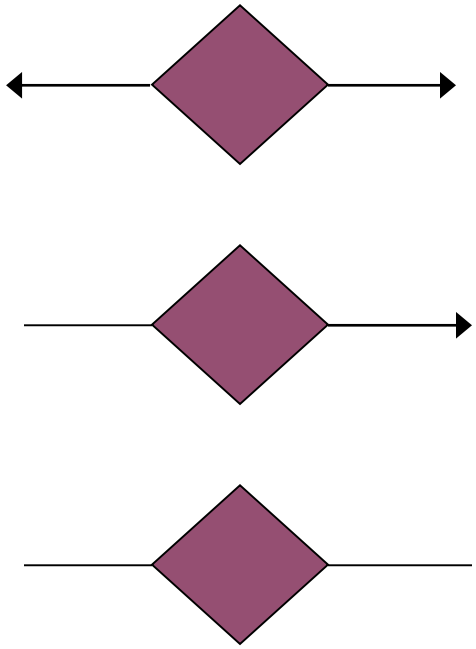
- makes is a subset of Product x Company:



Multiplicity of E/R Relationships

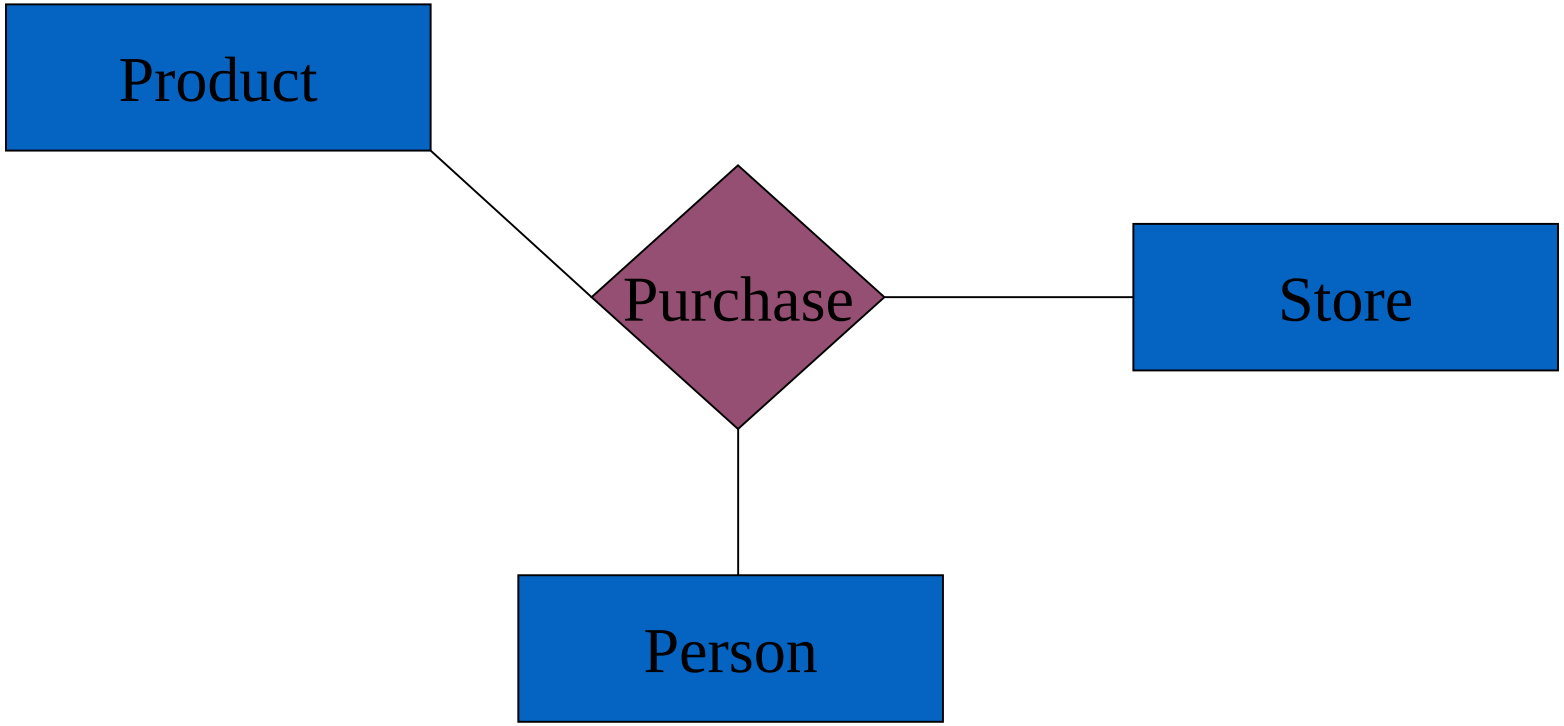
- one-one:
- many-one
- many-many





Multiway Relationships

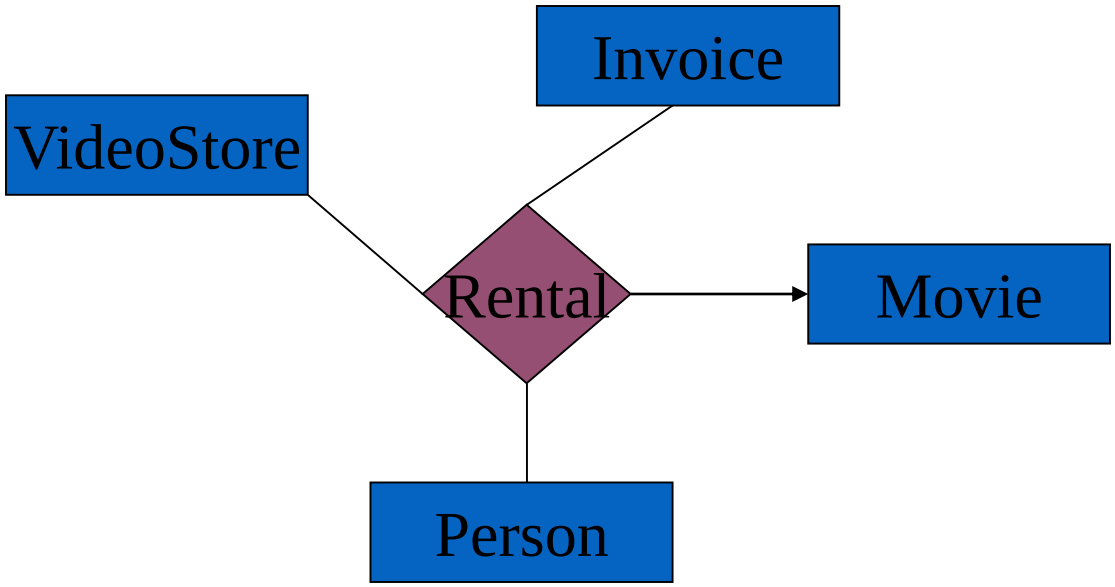
How do we model a purchase relationship between buyers, products and stores?



Can still model as a mathematical set (how ?)

Arrows in Multiway Relationships

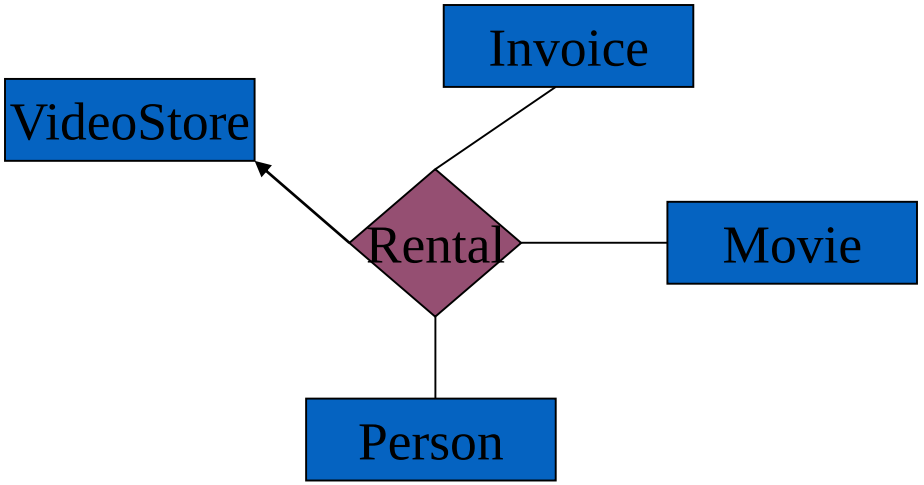
- Q: what does the arrow mean ?



- A:

Arrows in Multiway Relationships

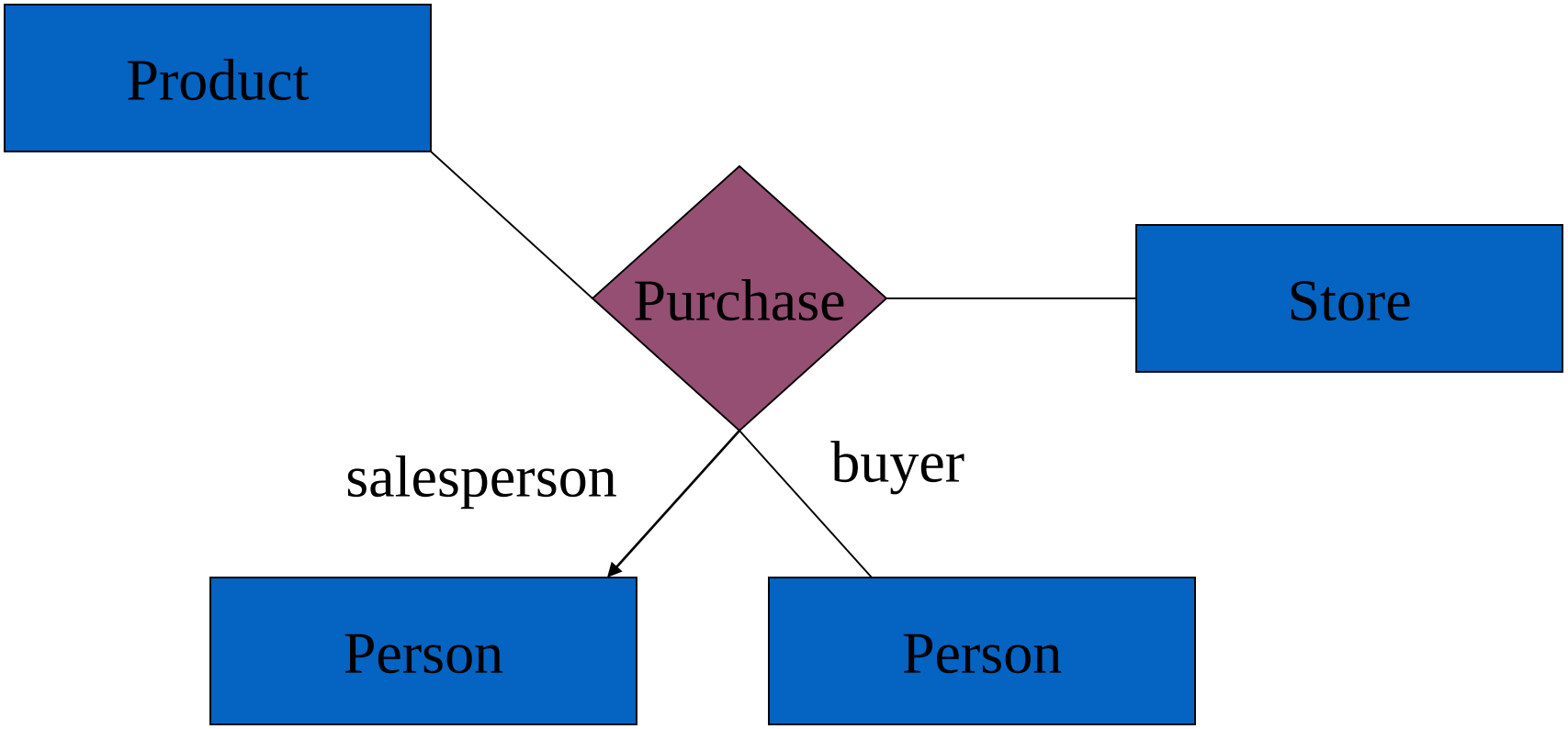
- Q: how do I say: “invoice determines store” ?
- A: no good way; best approximation:



- Q: Why is this incomplete ?

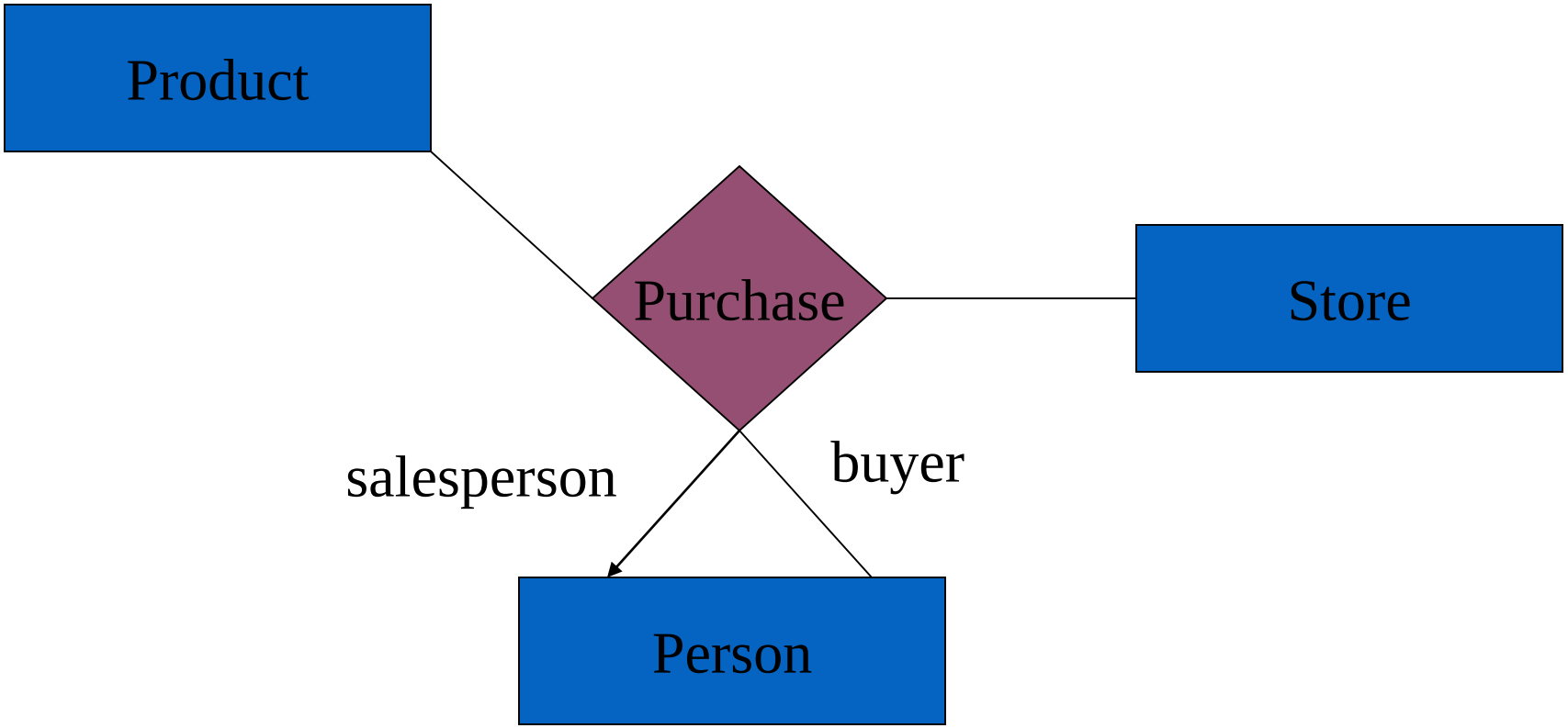
Roles in Relationships

What if we need an entity set twice in one relationship?

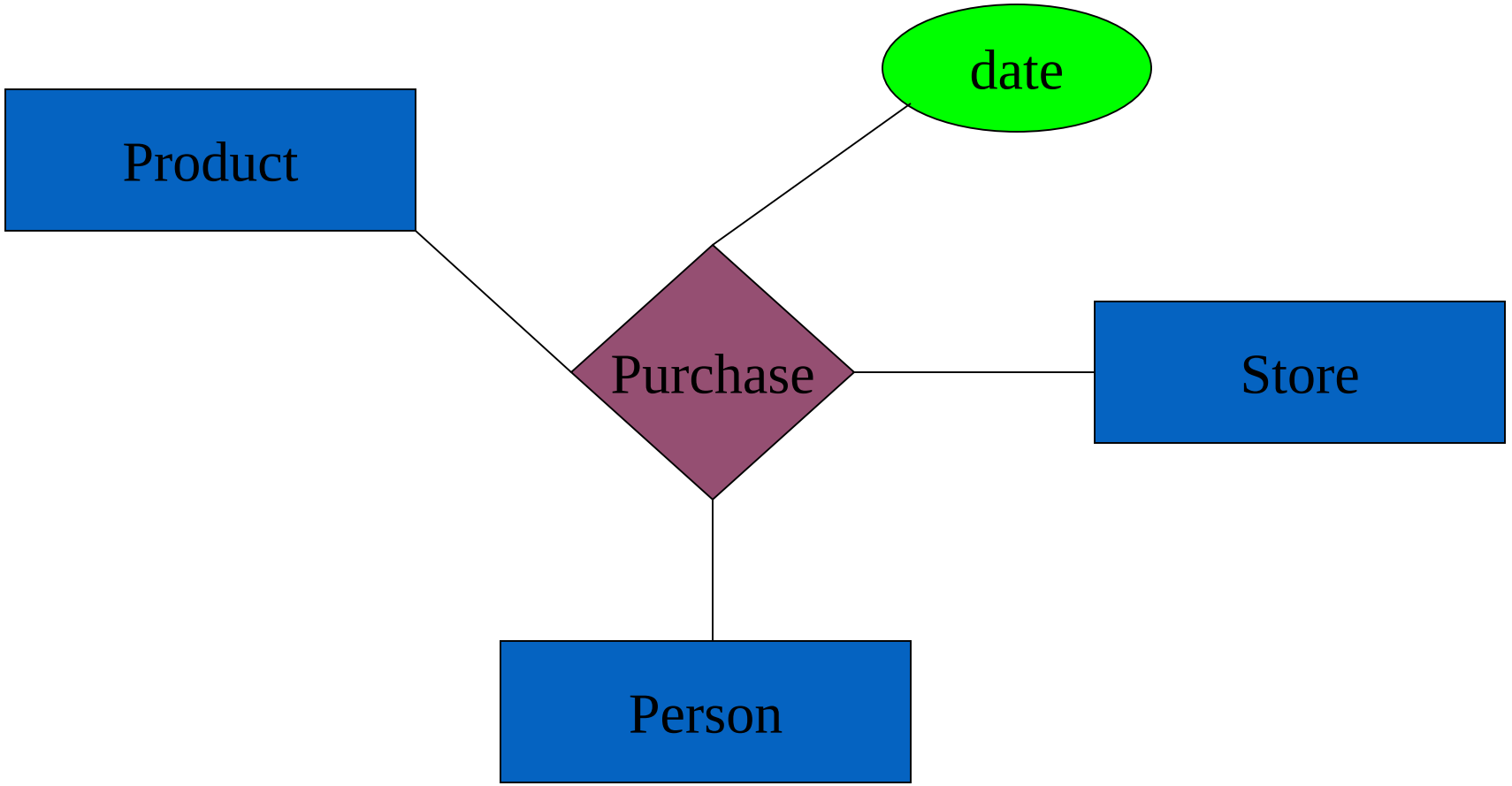


Roles in Relationships

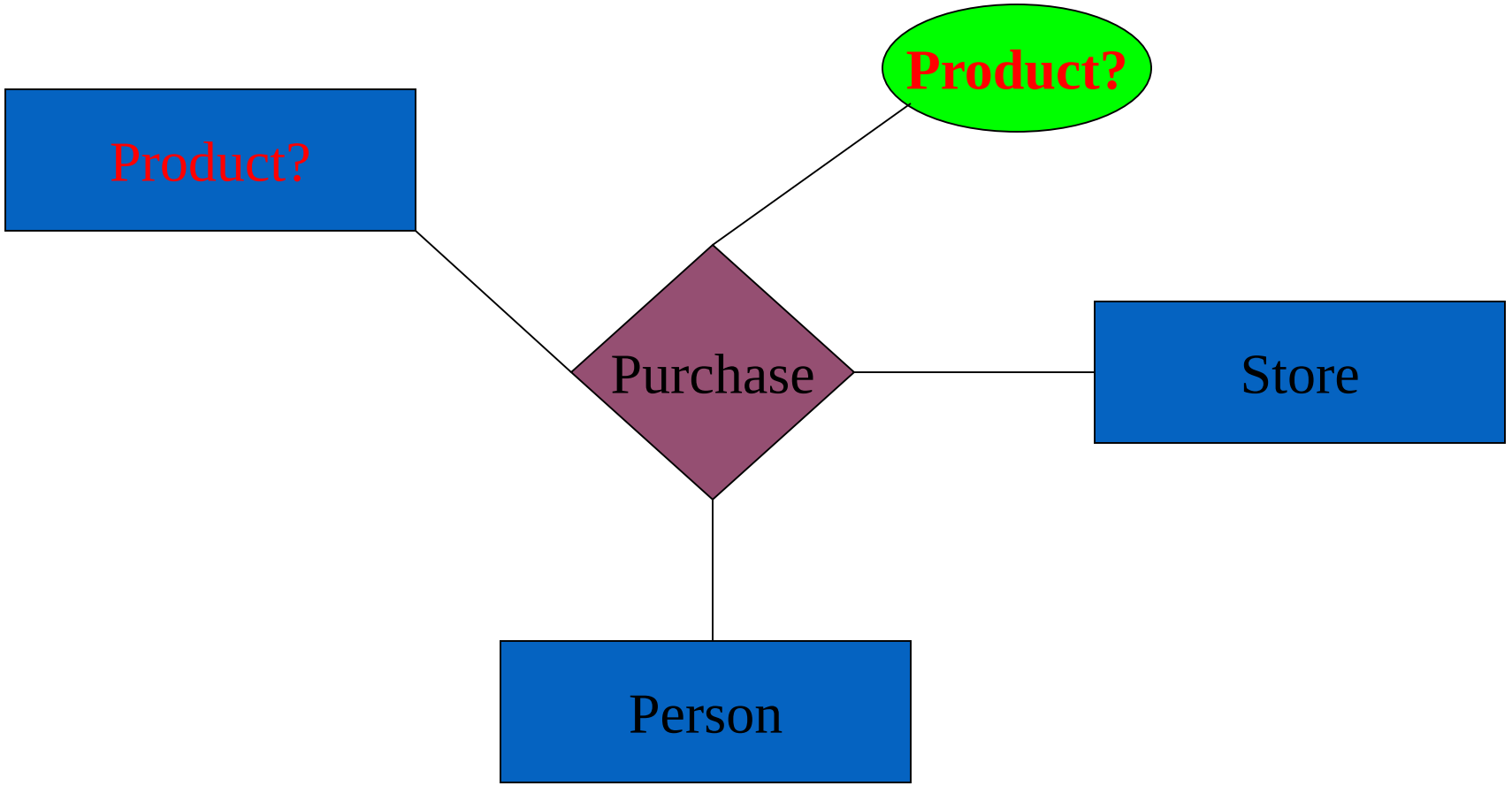
What if we need an entity set twice in one relationship?



Attributes on Relationships

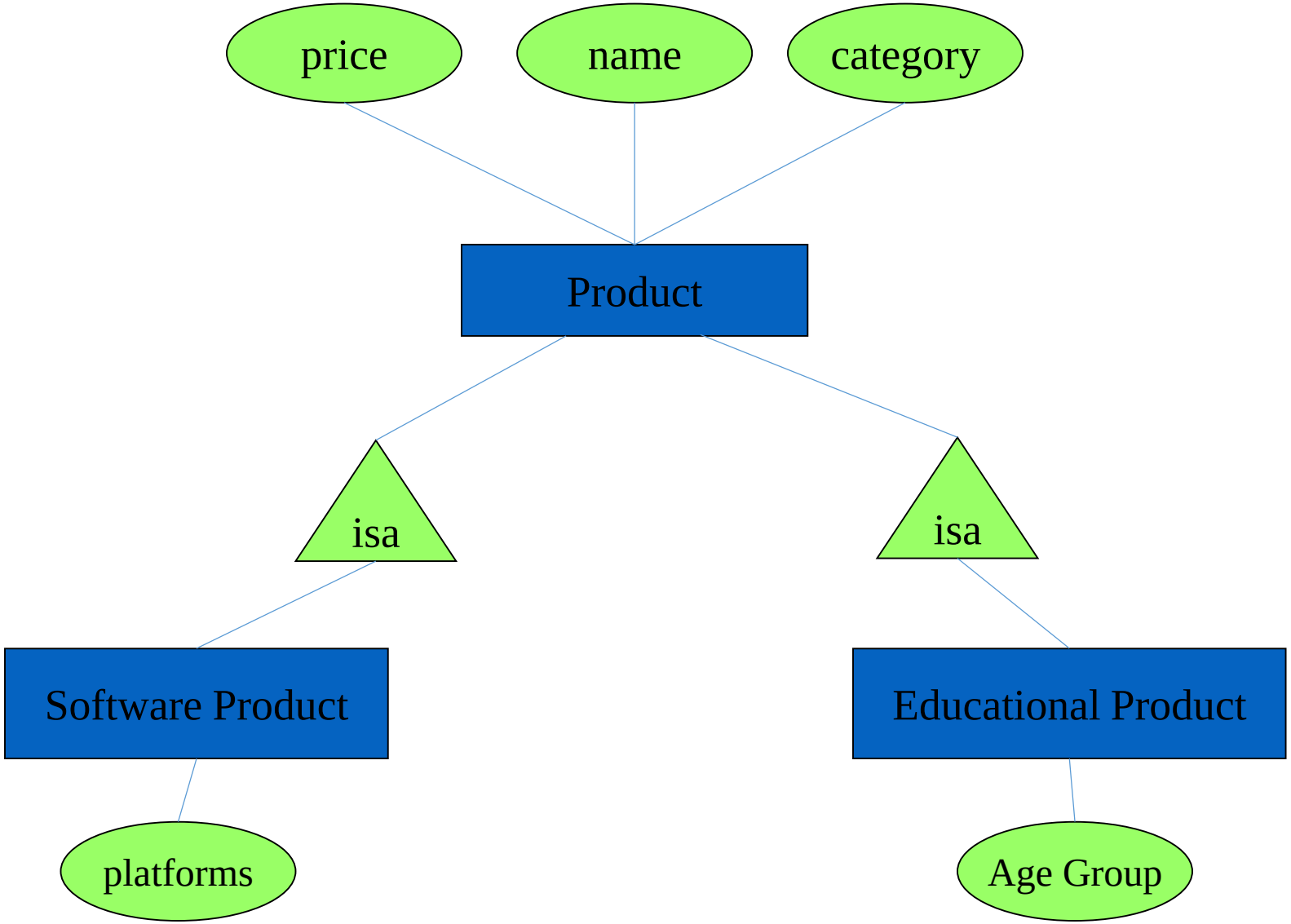


Q: Attributes vs. Entities on Relationships?



Subclasses

Subclasses in ER Diagrams



Warning: Viewers' Discretion Please

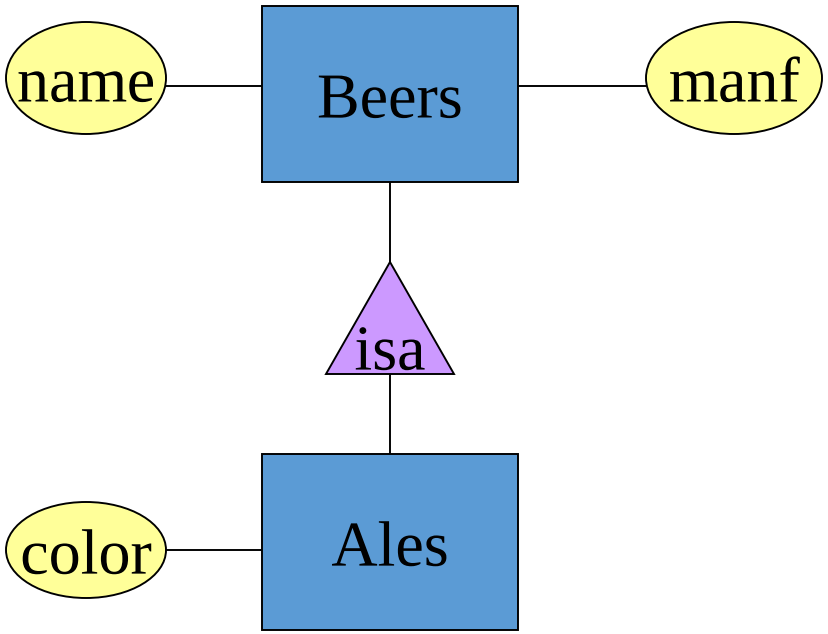
Subclasses

- Subclass = special case = fewer entities = more properties.
- Example: Ales are a kind of beer.
  - Not every beer is an ale, but some are.
  - Let us suppose that in addition to all the properties (attributes and relationships) of beers, ales also have the attribute color.

Subclasses in ER Diagrams

- Assume subclasses form a tree.
  - I.e., no multiple inheritance.
- Isa triangles indicate the subclass relationship.
  - Point to the superclass.

Example



Behind the Scene: Peter Chen

- Born in Taichung, Taiwan Peter Chen received a B.S. in electrical engineering in 1968 at the National Taiwan University, and a Ph.D. in computer science/applied mathematics at the Harvard University in 1973. From 1974 to 1978 Chen as Assistant Professor at MIT Sloan School of Management. From 1978 to 1984 he was Professor at the University of California, Los Angeles (UCLA Management School). Since 1983 Chen has held the position of M. J. Foster Distinguished Chair Professor of Computer Science at Louisiana State University.

- 1976. "The Entity-Relationship Model--Toward a Unified View of Data". In: ACM Transactions on Database Systems 1/1/1976 ACM-Press ISSN 0362-5915, S. 9–36

- One of the most cited CS papers.

- http://en.wikipedia.org/wiki/Peter_Chen

Constraints

Constraints

- A constraint = an assertion about the database that must be true at all times
- Part of the database schema
- Very important in database design

Modeling Constraints:
Part of the modeling process.

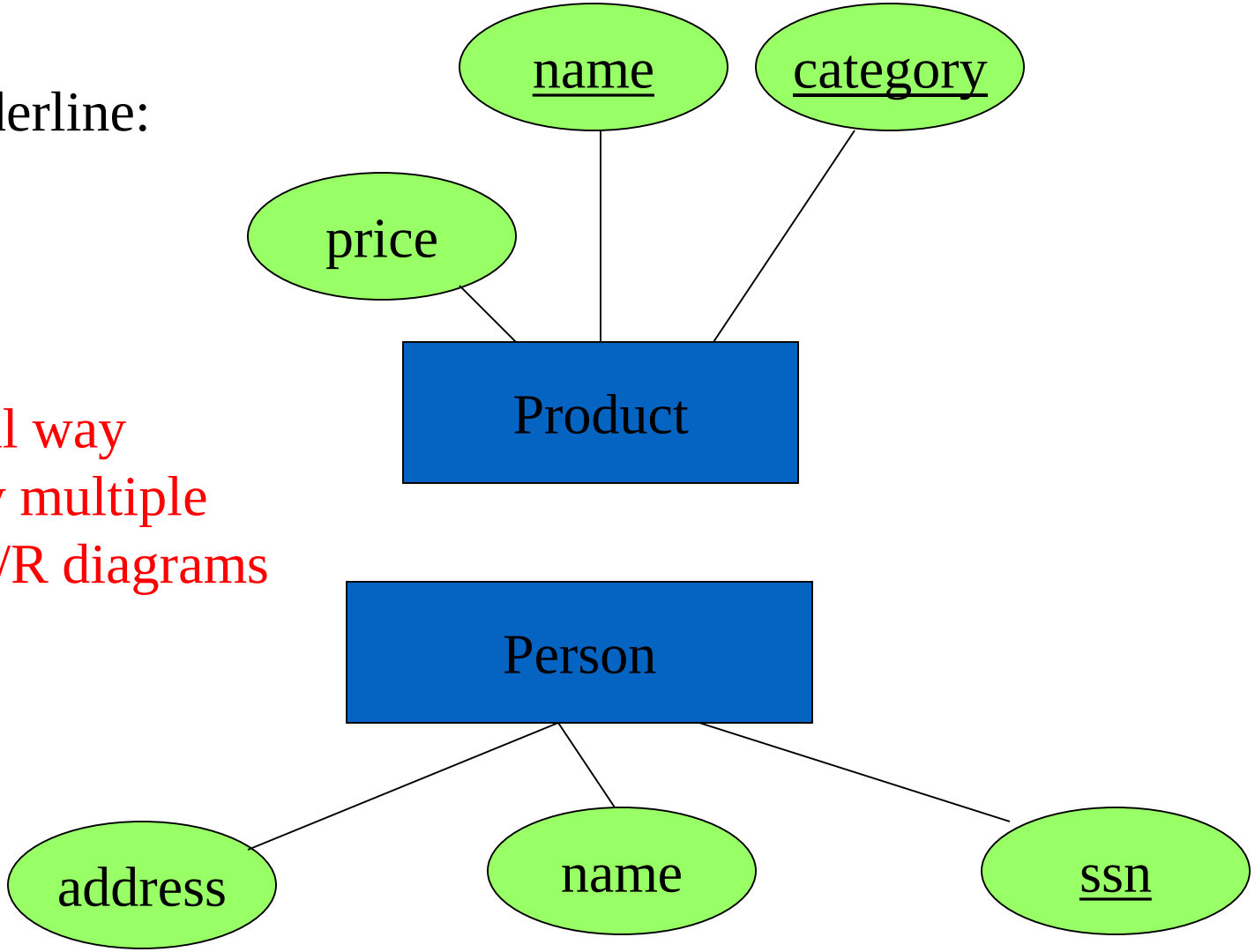
- Keys: social security number uniquely identifies a person.
- Single-value constraints: a person can have only one father.
- Referential integrity constraints: if you work for a company, it must exist in the database.
- Domain constraints: peoples' ages are between 0 and 150.
- General constraints: all others (at most 50 students enroll in a class)

Why Constraints are Important?

- Give more semantics to the data
  - help us better understand it
- Allow us to refer to entities (e.g., using keys)
- Enable efficient storage, data lookup, etc.

Keys in E/R Diagrams

Underline:



No formal way
to specify multiple
keys in E/R diagrams

More about Keys

- Every entity set must have a key
  - why?
- A key can consist of more than one attribute
- There can be more than one key for an entity set
  - one key will be designated as primary key
- Requirement for key in an isa hierarchy
  - see book

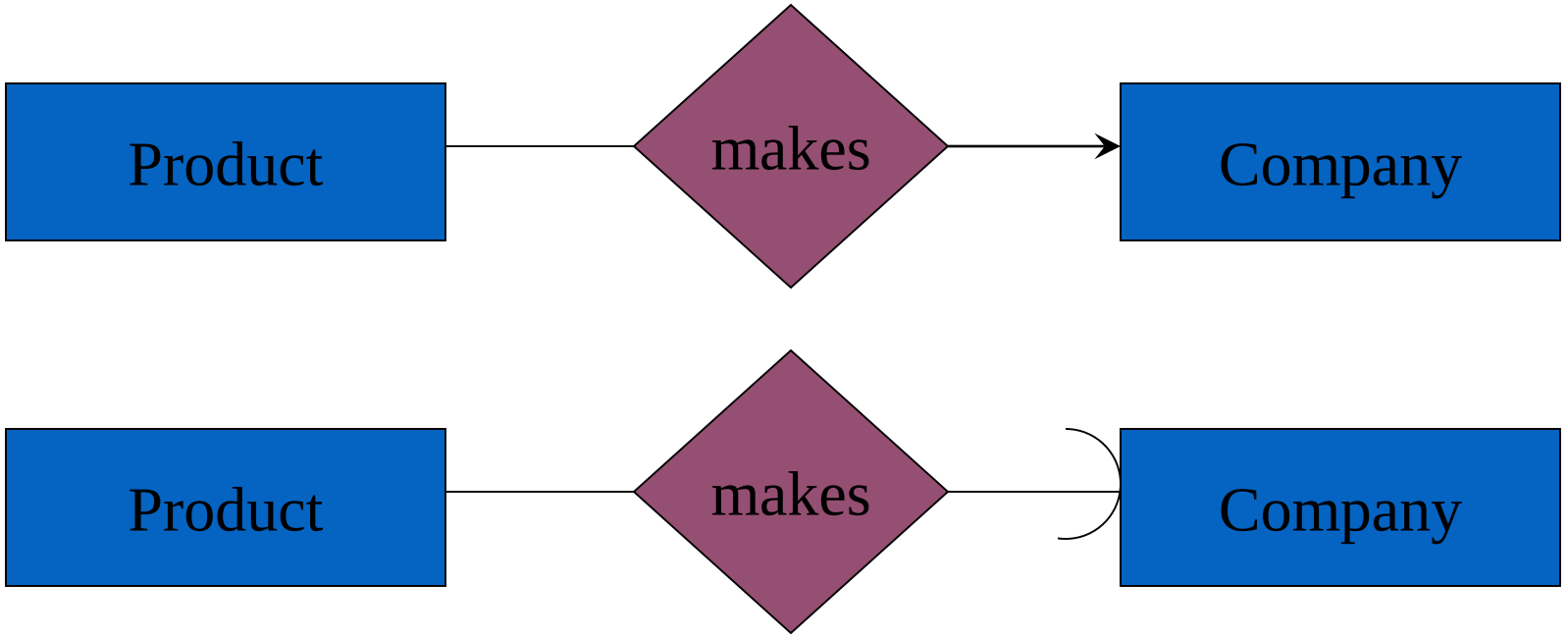
Referential Integrity Constraint

- Ref. int. constraint: exactly one value exists in a given role
- An attribute has a non-null, single value
  - this can be considered a kind of ref. int. constraint
- However, we more commonly use such constraints to refer to relationships

Referential Integrity Constraints

- In some formalisms we may refer to other object but get garbage instead
  - e.g. a dangling pointer in C/C++
- The Referential Integrity Constraint on relationships explicitly requires a reference to exist

Referential Integrity Constraints



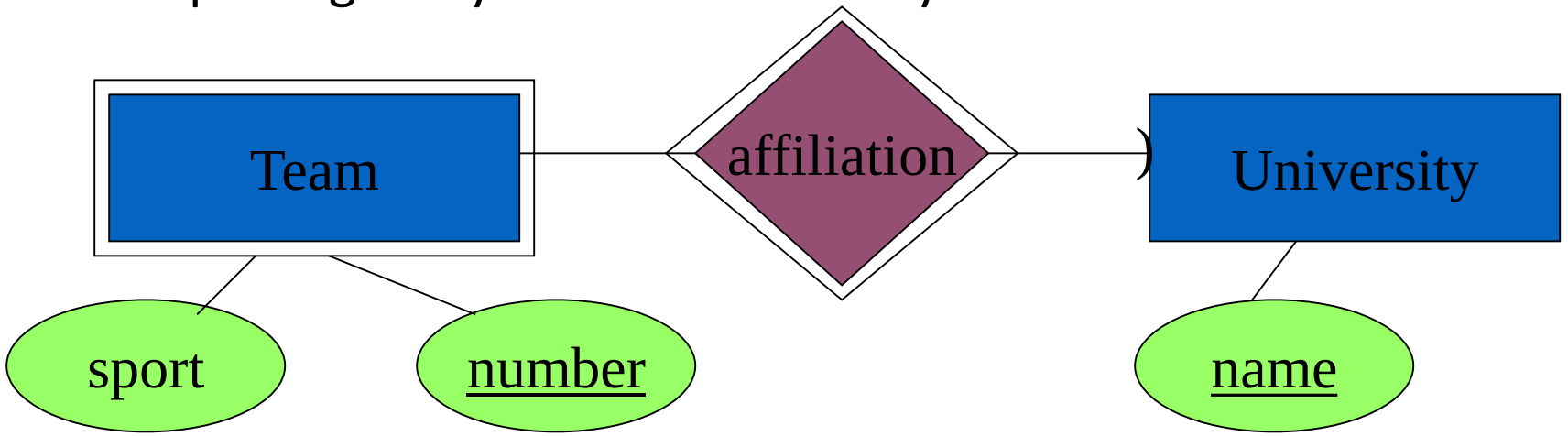
- This will be even clearer once we get to relational databases

Weak Entity Sets

Entity sets are weak when their key attributes come from other classes to which they are related.

This happens if:

- part-of hierarchies
- splitting n-ary relations to binary.

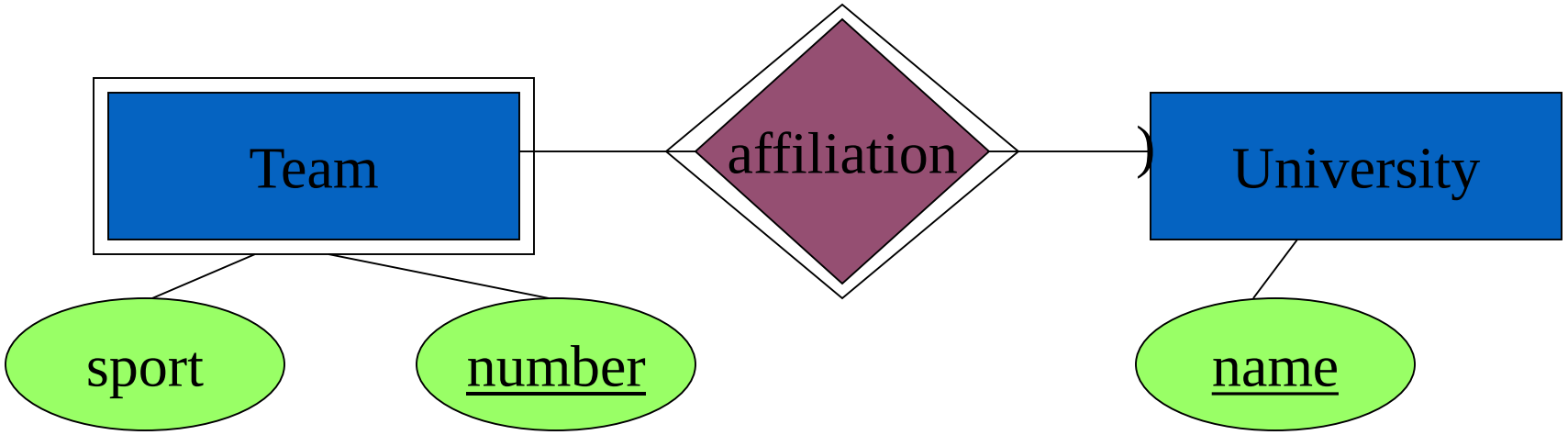


Weak Entity Sets

- Occasionally, entities of an entity set need “help” to identify them uniquely.

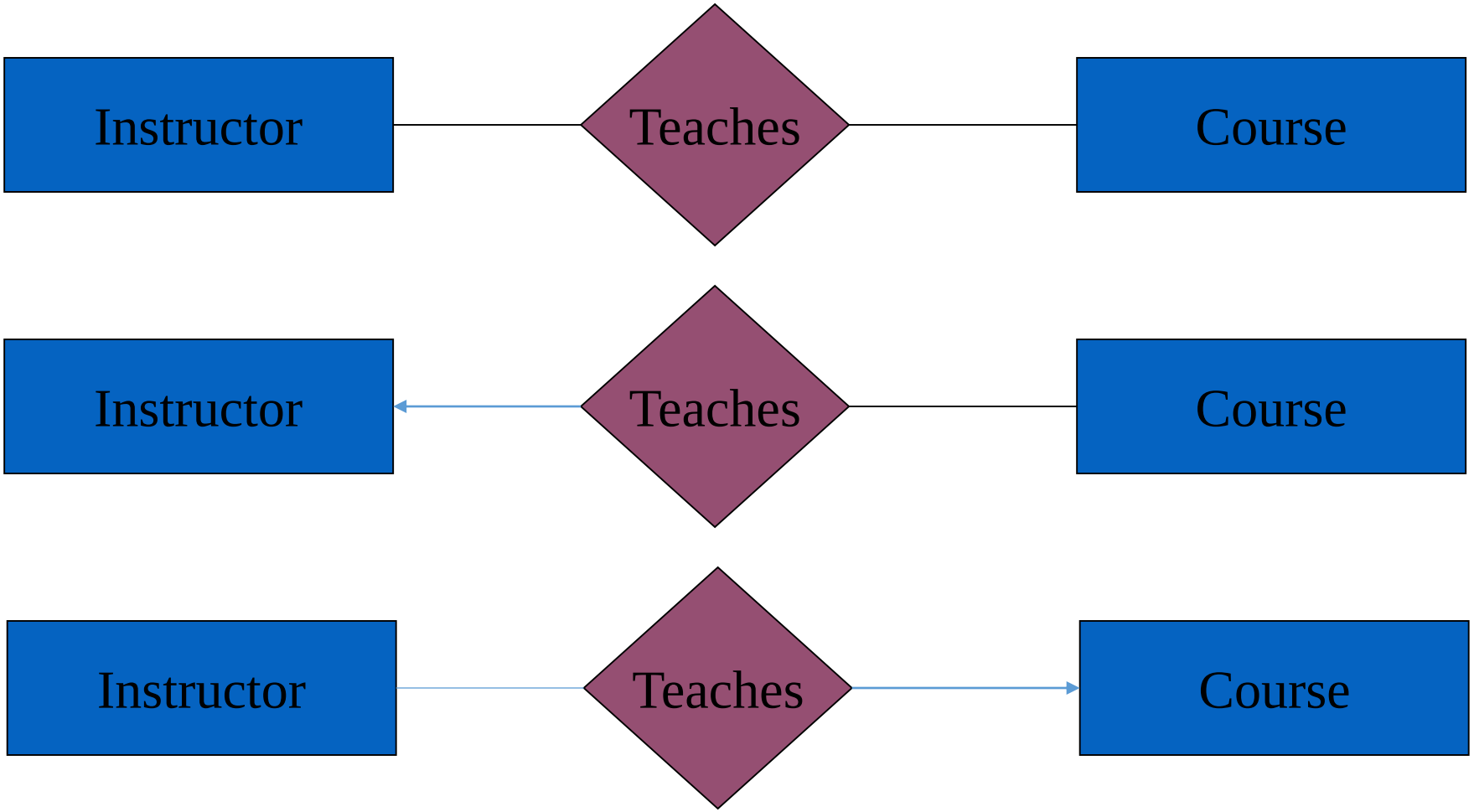
- Entity set E is said to be weak if in order to identify entities of E uniquely, we need to follow one or more many-one relationships from E and include the key of the related entities from the connected entity sets.

Q: Is this subclassing? Similarity? Difference?



Design Principles

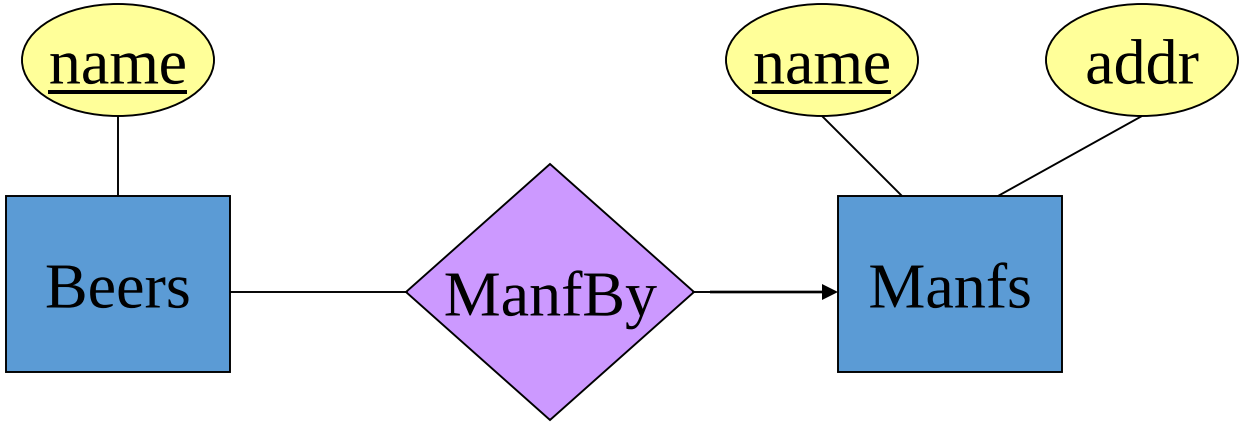
Design Principles: Be Faithful



Avoid Redundancy

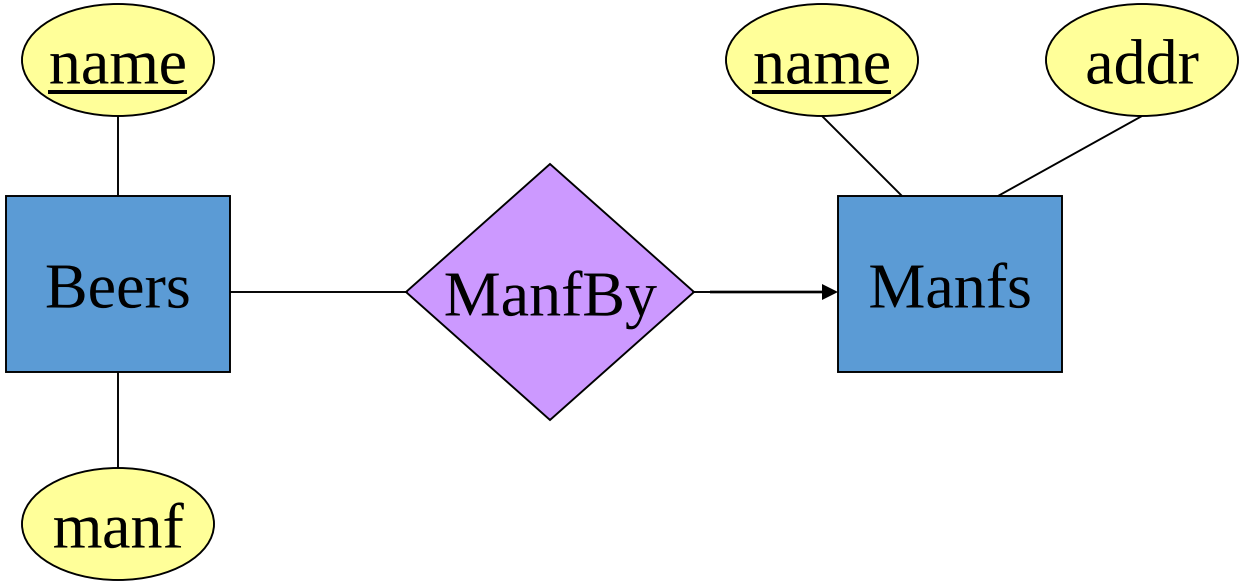
- Redundancy occurs when we say the same thing in two different ways.
- Redundancy wastes space and (more importantly) encourages inconsistency.
  - The two instances of the same fact may become inconsistent if we change one and forget to change the other, related version.

Example: Good



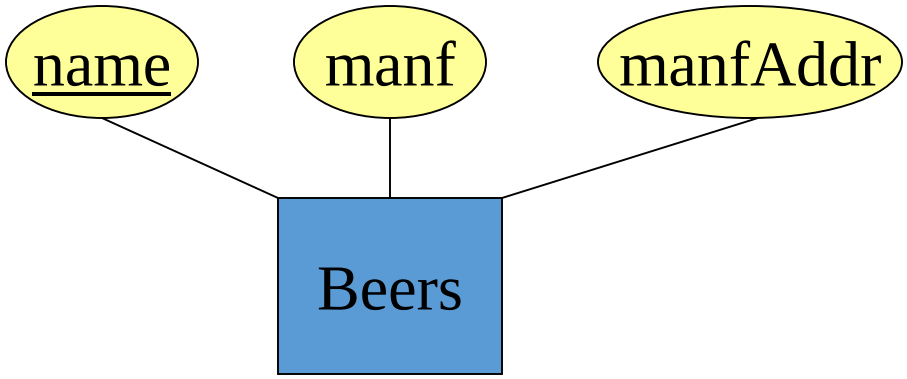
This design gives the address of each manufacturer exactly once.

Example: Bad



This design states the manufacturer of a beer twice: as an attribute and as a related entity.

Example: Bad



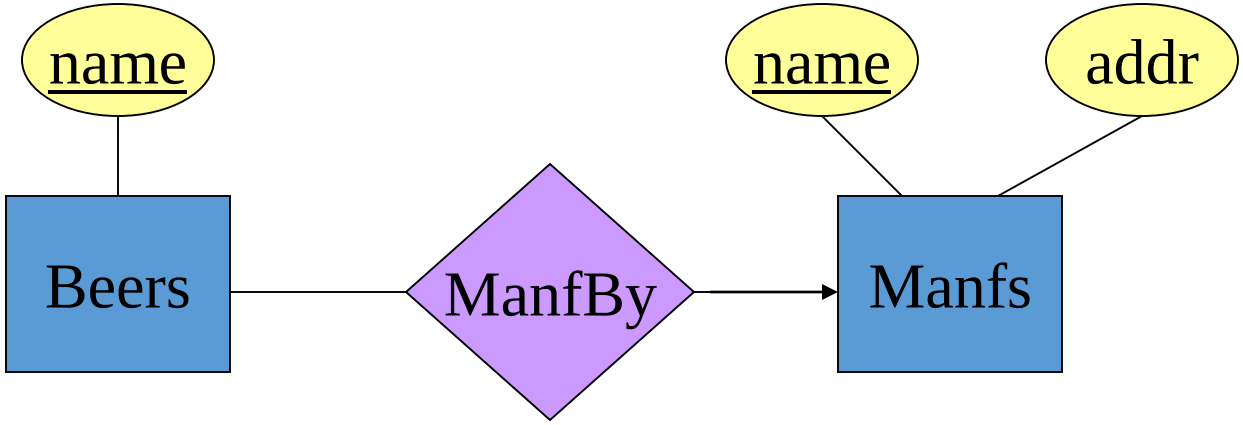
This design repeats the manufacturer's address once for each beer; loses the address if there are temporarily no beers for a manufacturer.

Entity Sets vs. Attributes

- An entity set should satisfy at least one of the following conditions:
- It is more than the name of something; it has at least one non-key attribute.

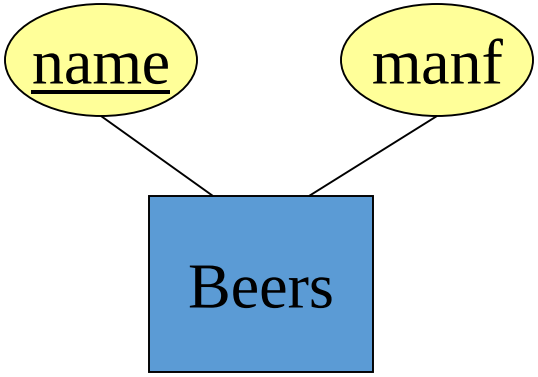
- It is the “many” in a many-one or many-many relationship.

Example: Good



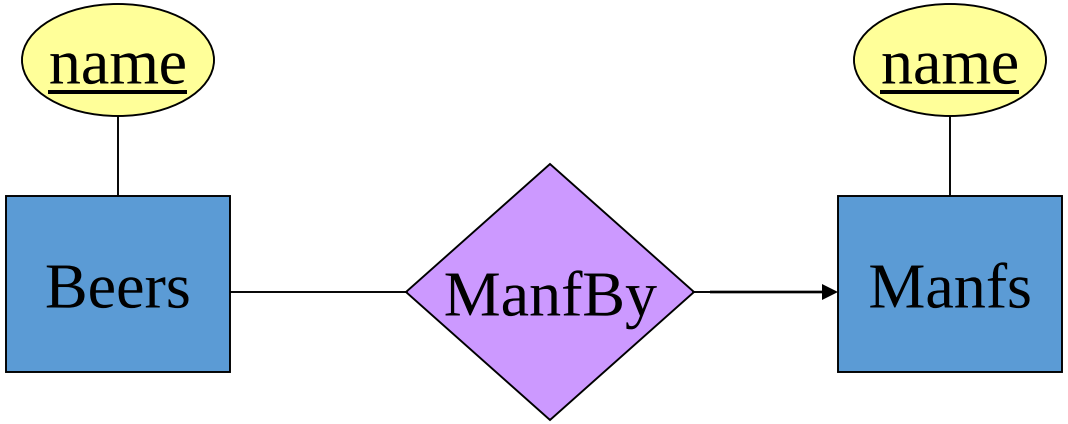
- Manfs deserves to be an entity set because of the non-key attribute addr.
- Beers deserves to be an entity set because it is the "many" of the many-one relationship ManfBy.

Example: Good



There is no need to make the manufacturer an entity set, because we record nothing about manufacturers besides their name.

Example: Bad



Since the manufacturer is nothing but a name, and is not at the "many" end of any relationship, it should not be an entity set.

Don’t Overuse Weak Entity Sets

- Beginning db designers often doubt that anything could be a key by itself.
  - They make all entity sets weak, supported by all other entity sets to which they are linked.
- In reality, we usually create unique ID’s for entity sets.
  - Examples include social-security numbers, automobile VIN’s etc.

When Do We Need Weak Entity Sets?

- The usual reason is that there is no global authority capable of creating unique ID's.
- Example: it is unlikely that there could be an agreement to assign unique player numbers across all football teams in the world.