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*.

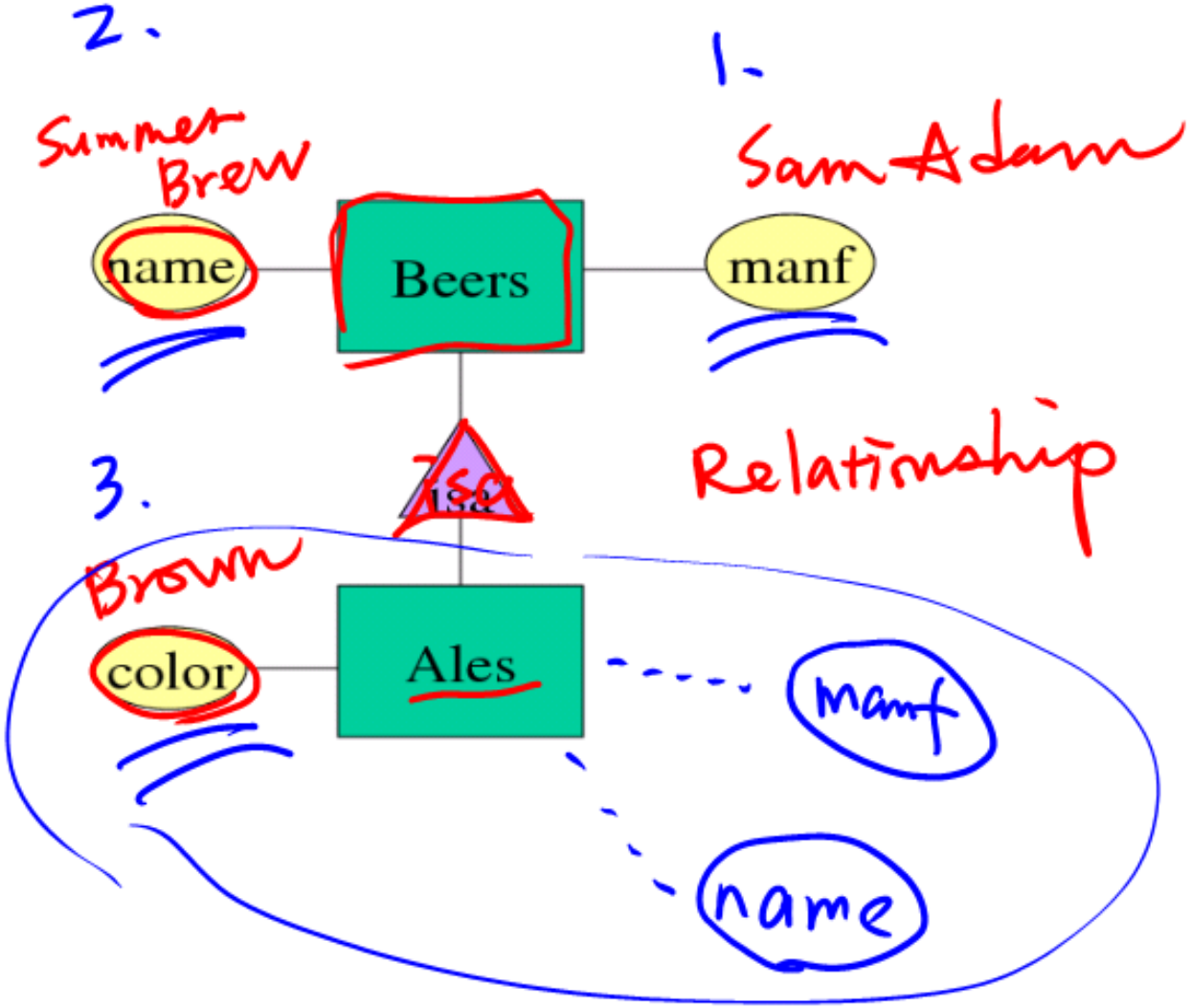
Subclasses in ER Diagrams

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

"storage"

Example

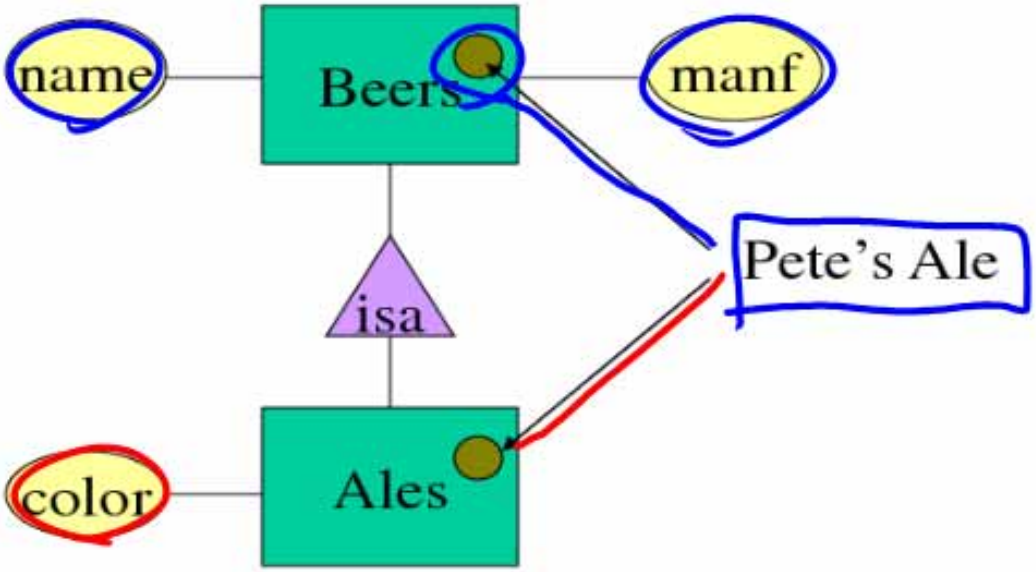
which table do I store attrs?



ER Vs. Object Oriented Subclasses

- In the object-oriented world, objects are in one class only.
  - Subclasses inherit properties from superclasses.
- In contrast, E/R entities have components in all subclasses to which they belong.
  - Matters when we convert to relations.

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 Assistent 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

Java Assert

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

Finding constraints is part of the modeling process.
Commonly used constraints:

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?

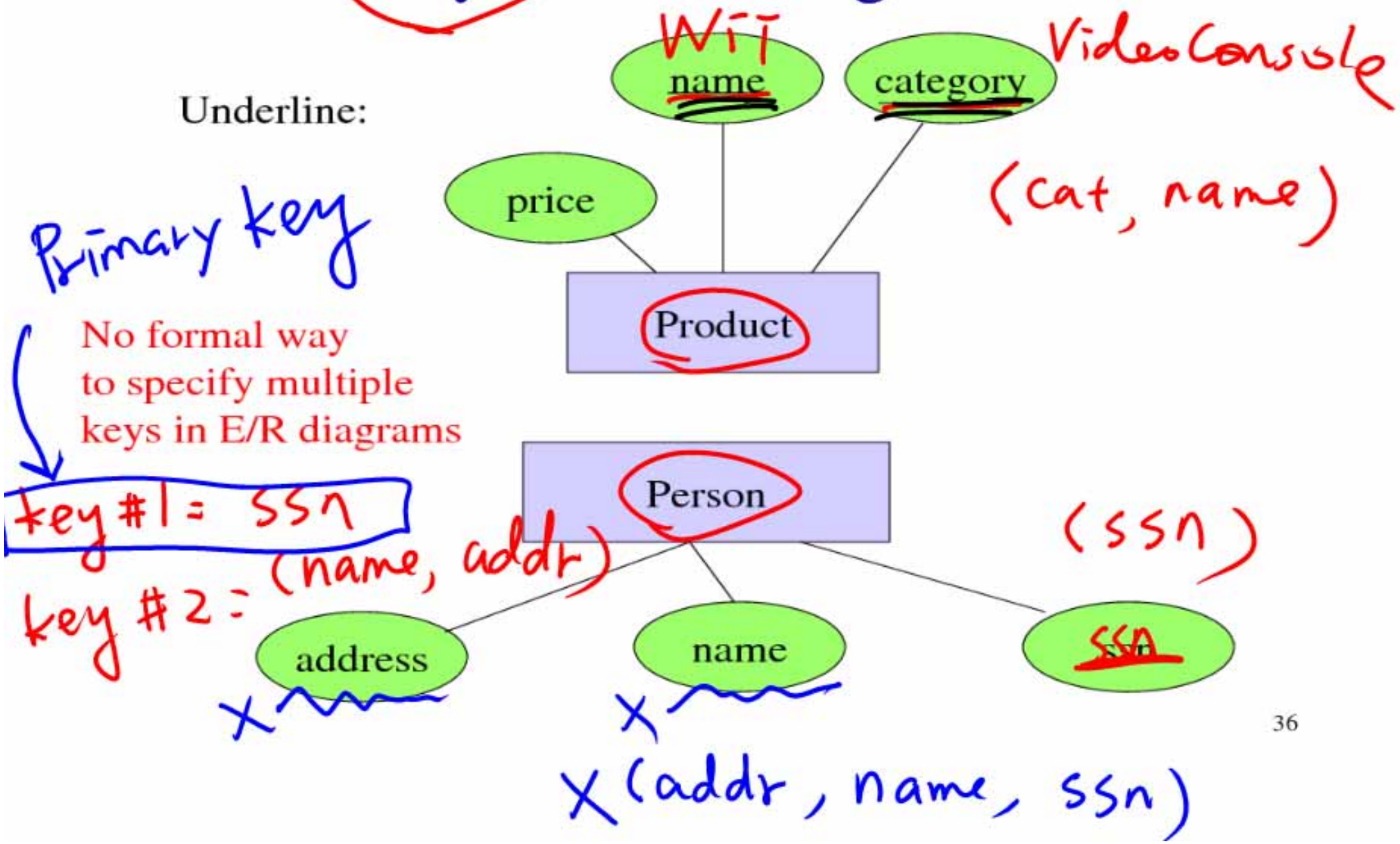
- Give more semantics to the data
  - help us better understand it
- Allow us to refer to entities (e.g., using keys)
- Enable efficient storage, data lookup, etc.

emp_date : Date
∨
birth_date : Date

every person can take ≤ 5 courses

netid is unique

Keys in E/R Diagrams



Entity set ⊆ no duplicates.

More about Keys

- Every entity set must have a key = All attr
  - 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

2. I don't want seg fault

Referential Integrity Constraint

pointer no fault.

- Ref. int. constraint: exactly one value exists in a given role

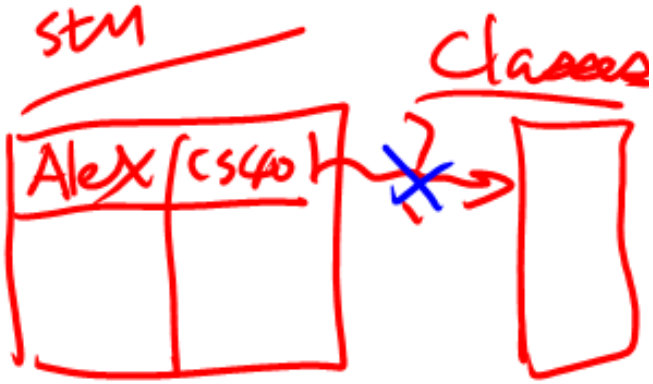
a) • An attribute has a non-null, single value
  - this can be considered a kind of ref. int. constraint

b) • However, we more commonly use such constraints to refer to relationships

a) Students

Name	gen.	Age
John/Alex	M	26
Sally	F	?? Null

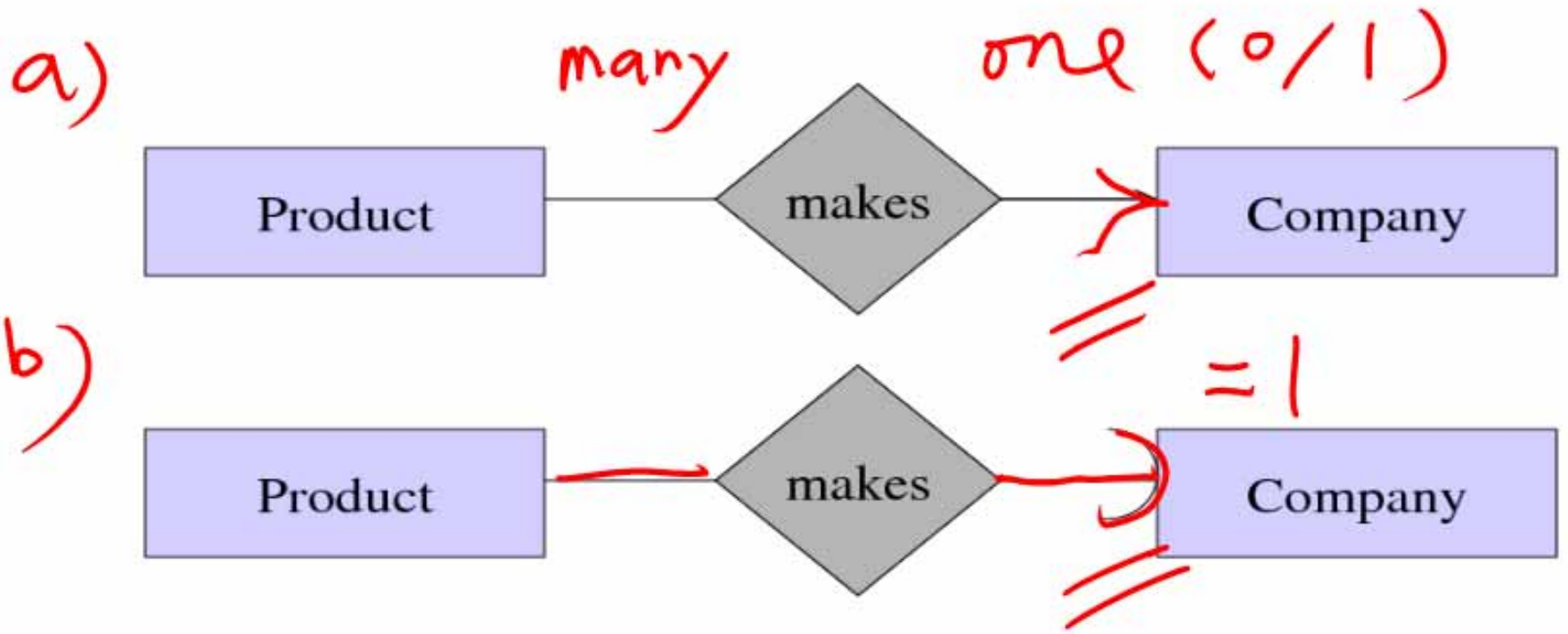
b)



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



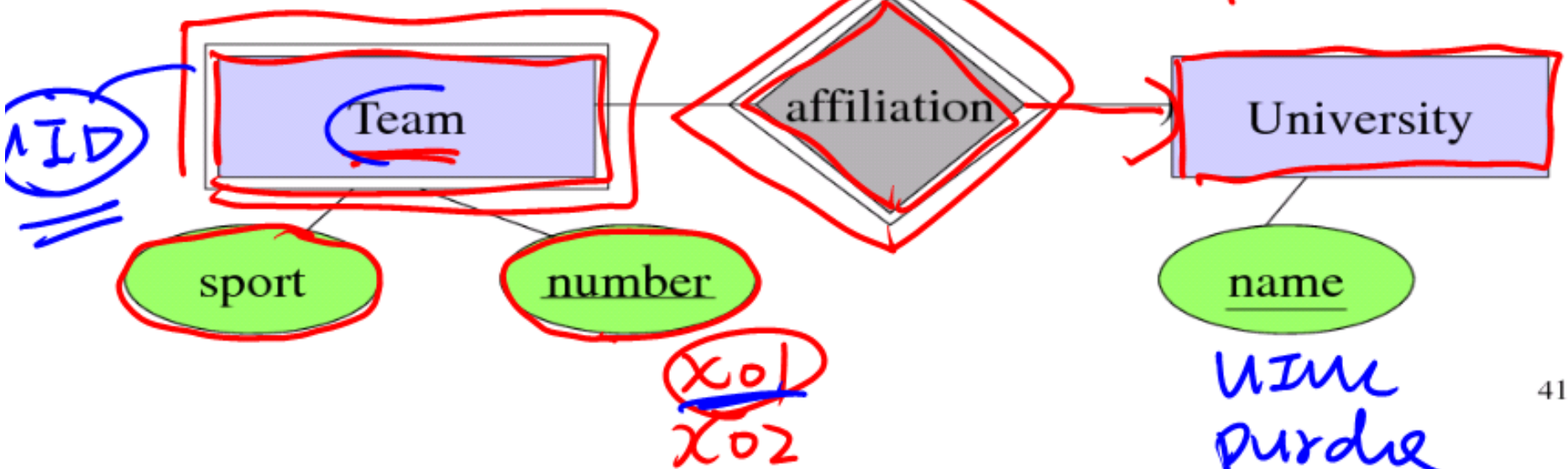
- This will be even clearer once we get to relational databases

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.

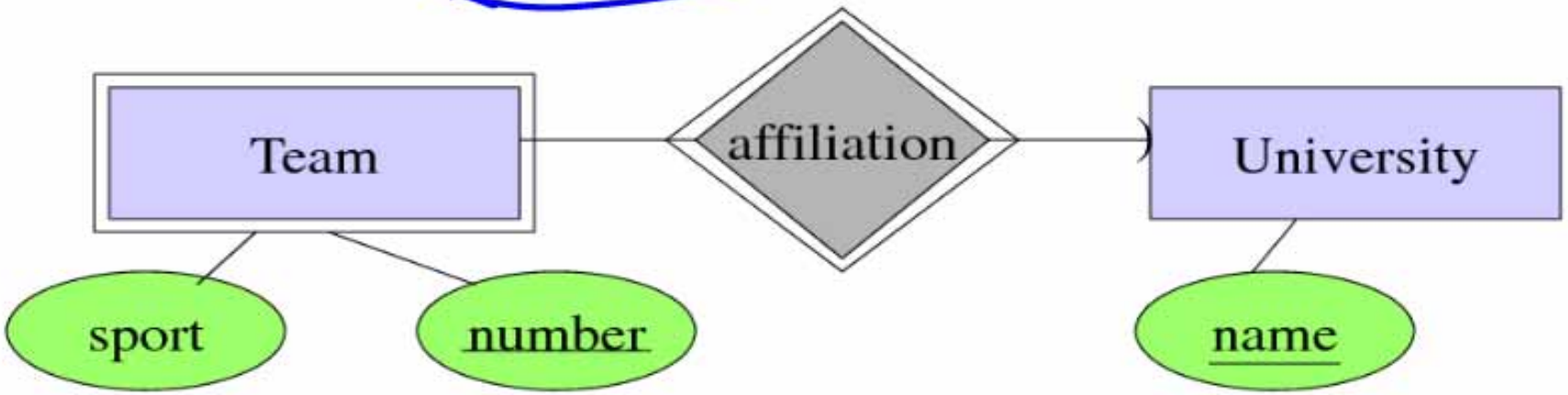


Weak Entity Sets

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

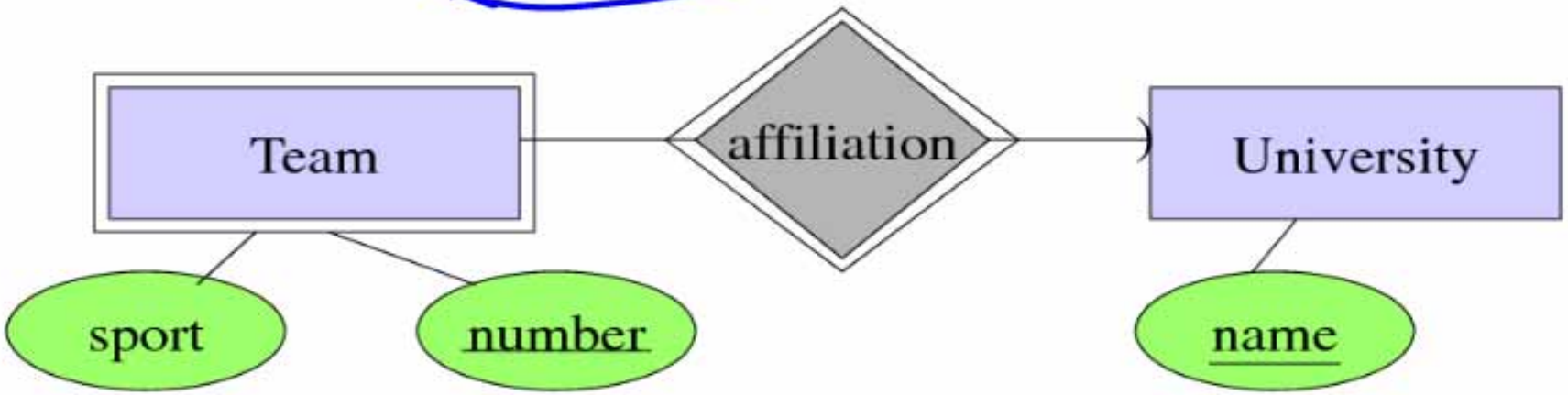
- Entity set E is said to be *weak* if in order to identify entities of E uniquely, we need to follow one or more many-one relationships from E and include the key of the related entities from the connected entity sets.

Q: Is this subclassing? Similarity? Difference?



Think about it.

Q: Is this subclassing? Similarity? Difference?

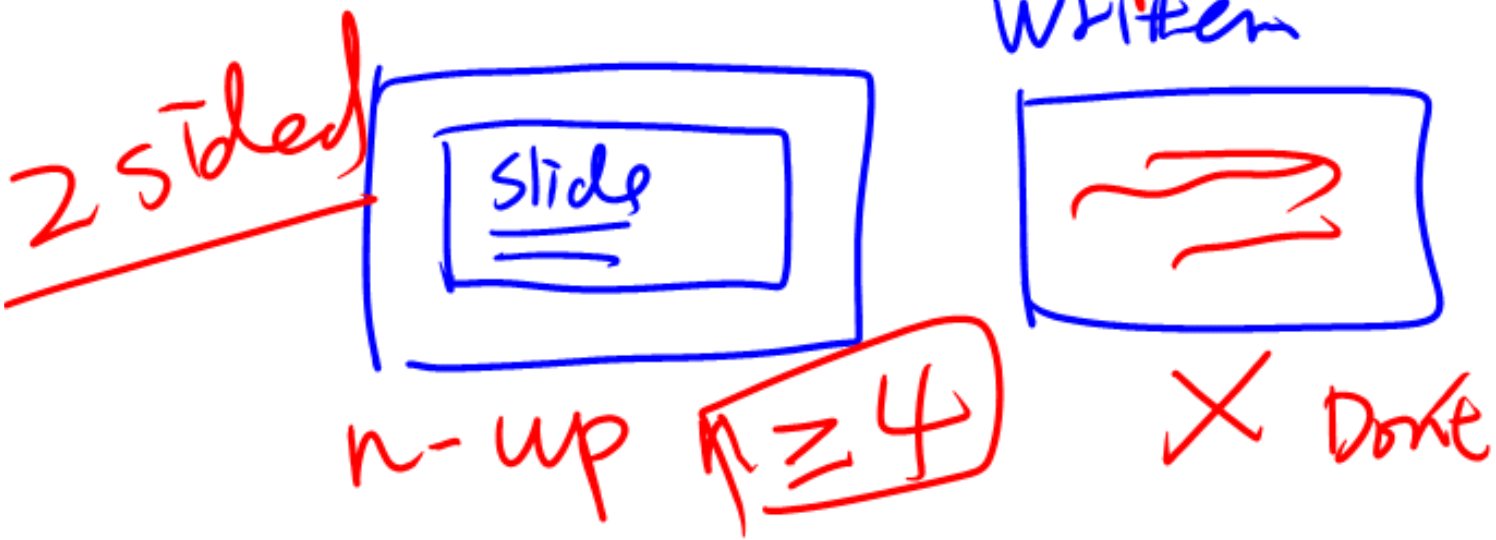


Think about it.

Announcements

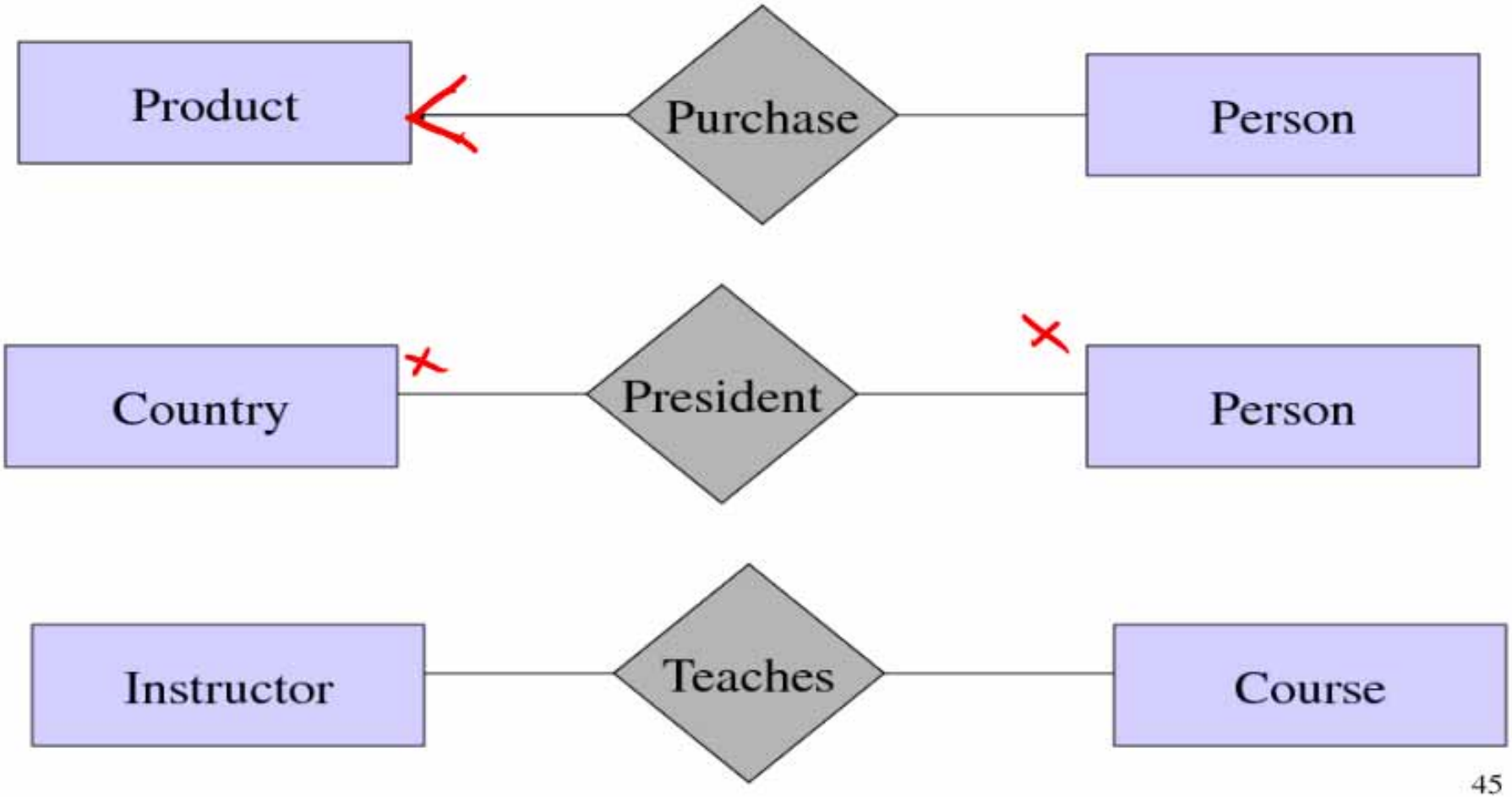
1. Vote for Final Exam time

2. Print n-up h.o.



Now, about design techniques ...

Design Principles: Be Faithful

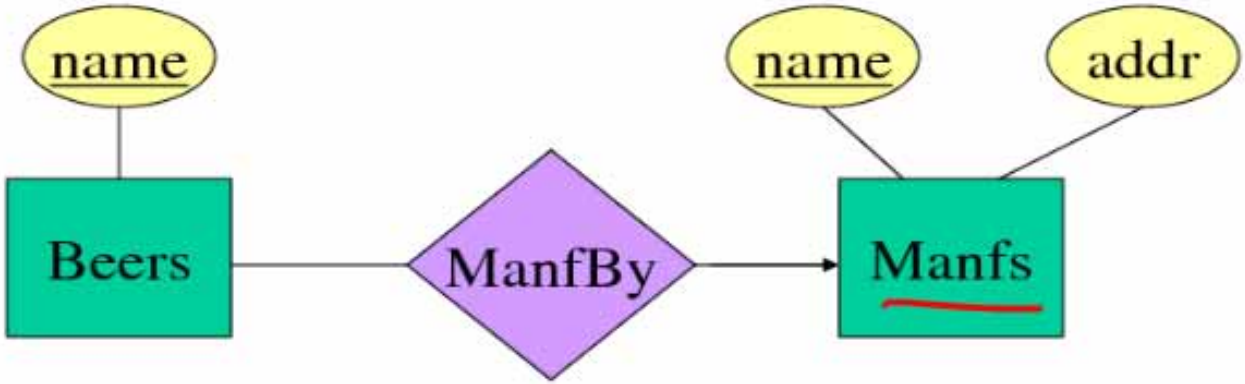


Say Just Once.

Avoiding Redundancy

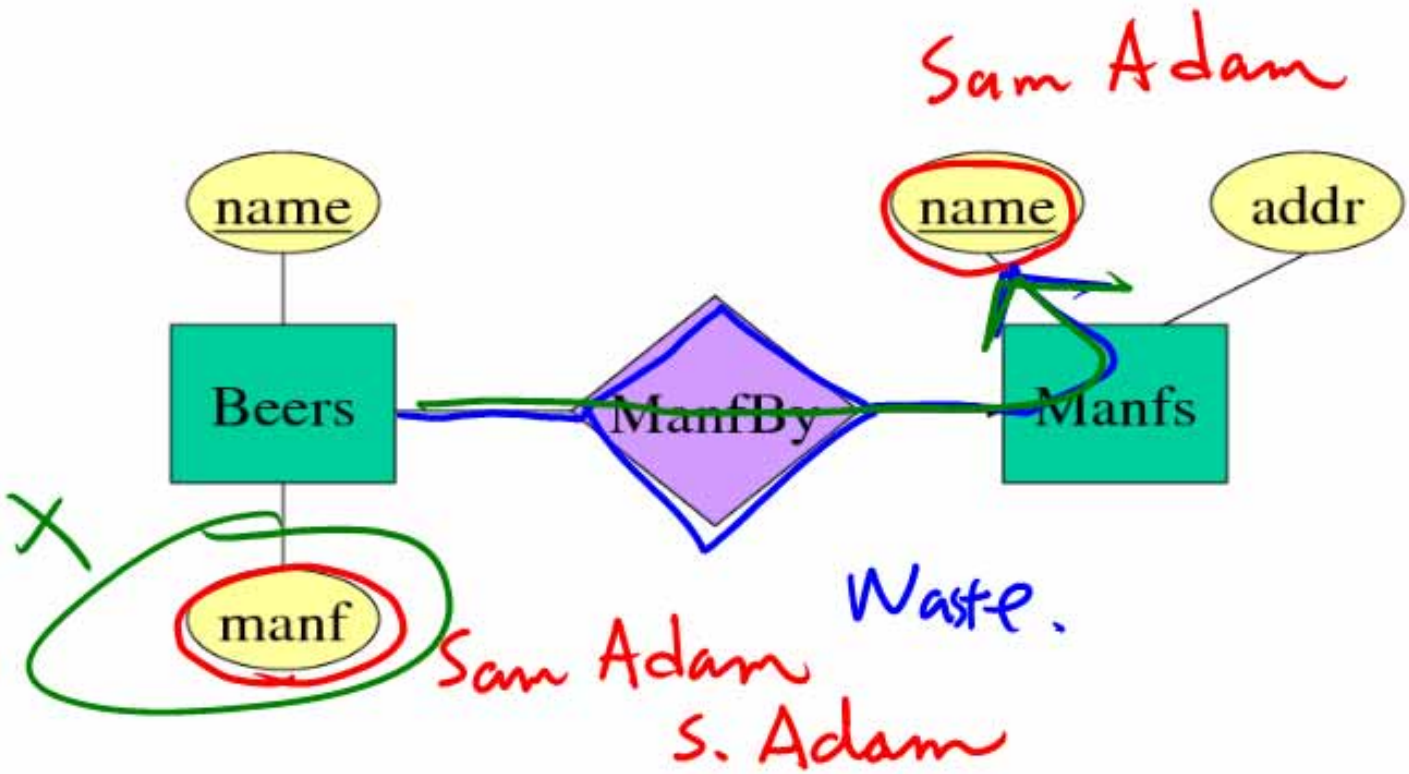
- Redundancy occurs when we say the same thing in two different ways.
- Redundancy wastes space and (more importantly) encourages inconsistency.
  - The two instances of the same fact may become inconsistent if we change one and forget to change the other, related version.

Example: Good



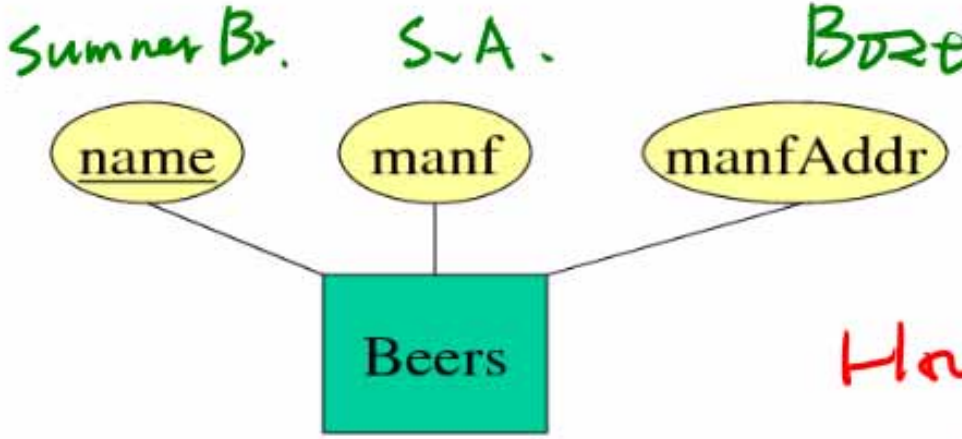
This design gives the address of each manufacturer exactly once.

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.

Obj w/ attr.

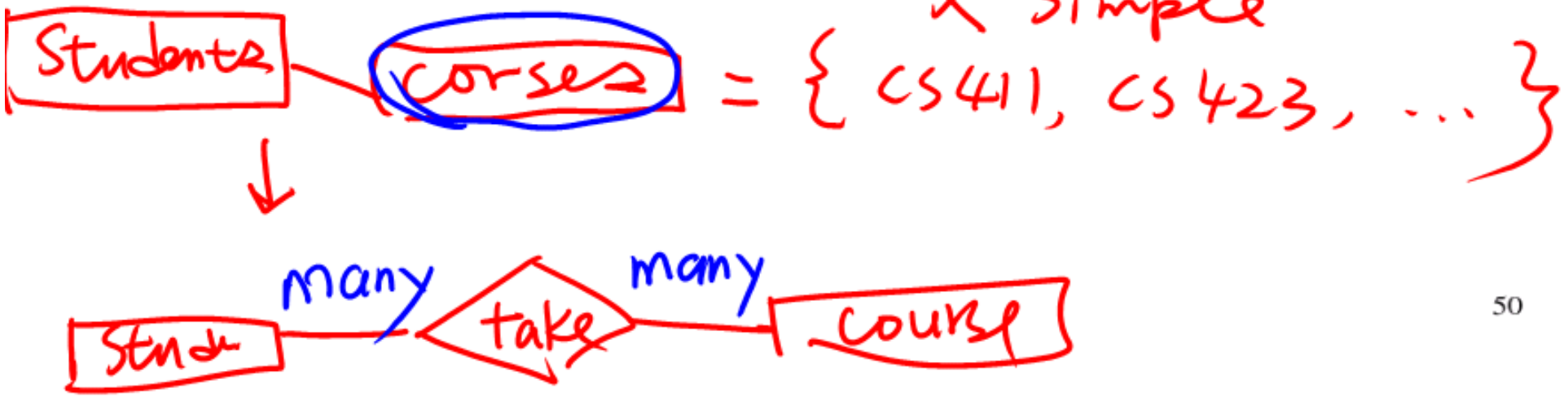
Simple

Entity Sets Versus Attributes

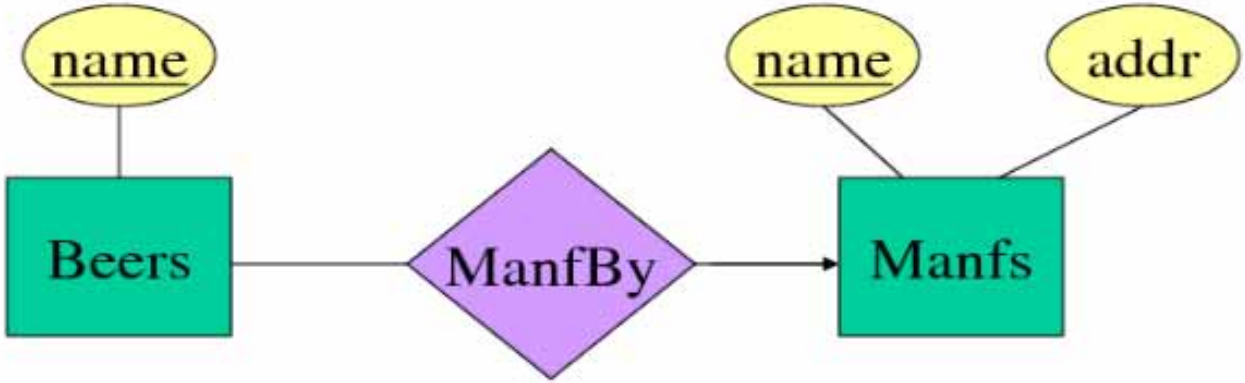
- An entity set should satisfy at least one of the following conditions:
  - (1) It is more than the name of something; it has at least one nonkey attribute.

    or

  - (2) It is the "many" in a many-one or many-many relationship.



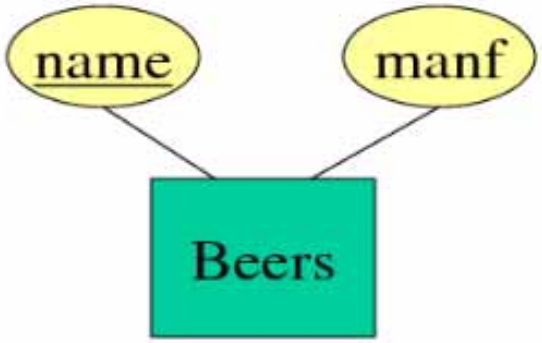
Example: Good



- *Manfs* deserves to be an entity set because of the nonkey attribute *addr*.

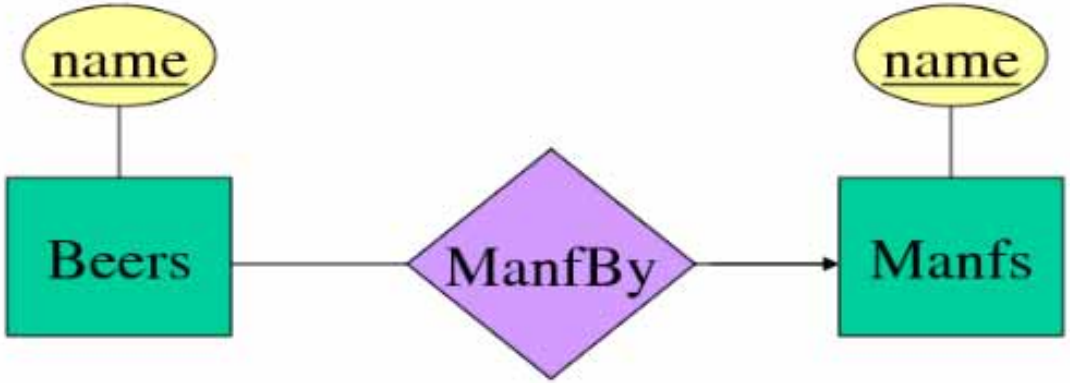
- *Beers* deserves to be an entity set because it is the "many" of the many-one relationship *ManfBy*.

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 database 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.

ER Review

- Basic stuff
  - entity, attribute, entity set
  - relation: binary, multiway, converting from multiway
  - relationship roles, attributes on relationships
  - subclasses (is-a)
- Constraints
  - on relations
    - many-one, one-one, many-many
    - limitations of arrows
  - keys, single-valued, ref integrity, domain & general constraints

Hierarchical (Network) → "Ims"

Obj Oriented DBMS

XML

CS411
Database Systems

Doc. DB

the choice

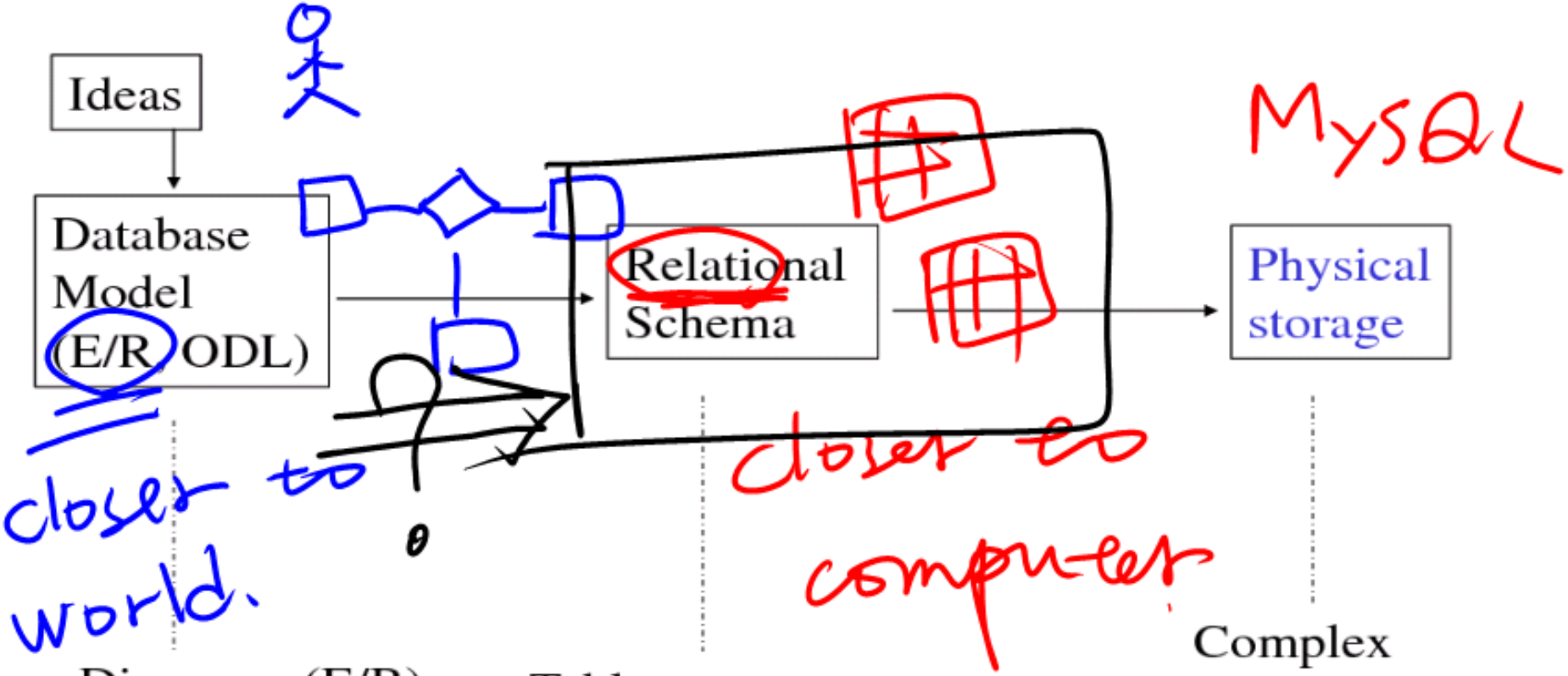
03: Relational Model

Why Do We Learn This?

- What is "relation"? Why such a simple notion can model Data?
- How to fit data onto our computer?

Motivations & Comparison

Database Modeling & Implementation



Diagrams (E/R)

Tables:
column names: attributes
rows: tuples

Complex file organization and index structures.

ER Model vs. Relational Model

- Both are used to model data
- ER model has many concepts
  - entities, relations, attributes, etc.
  - well-suited for capturing the app. requirements
  - not well-suited for computer implementation
  - (does not even have operations on its structures)
- Relational model
  - has just a single concept: relation
  - world is represented with a collection of tables
  - well-suited for efficient manipulations on computers

The Basics

T ⊆ Name × Price × Category × Manufacturer

An Example of a Relation

Table name

Attribute names

Columns

Products:

Name	Price	Category	Manufacturer
gizmo	$19.99	gadgets	GizmoWorks
Power gizmo	$29.99	gadgets	GizmoWorks
SingleTouch	$149.99	photography	Canon
MultiTouch	$203.99	household	Hitachi

row 1

row 2

4

tuples

Column Domains

- Each attribute has a type
- Must be atomic type (why? see later)
- Called *domain*
- Examples:
  - Integer
  - String
  - Real
  - …

Schemas vs. Instances

Schemas

The Schema of a Relation:

- Relation name plus attribute names
- E.g. Product(Name, Price, Category, Manufacturer)
- In practice we add the domain for each attribute

The Schema of a Database

- A set of relation schemas
- E.g. Product(Name, Price, Category, Manufacturer),
  Vendor(Name, Address, Phone),
  . . . . . .

Instances

- Relational schema = R(A1,…,Ak):
  Instance = relation with k attributes (of "type" R)
  - values of corresponding domains

- Database schema = R1(…), R2(…), …, Rn(…)
  Instance = n relations, of types R1, R2, ..., Rn

snapshots of ↑

Example

Relational schema: Product(Name, Price, Category, Manufacturer)

Instance:

Name	Price	Category	Manufacturer
gizmo	$19.99	gadgets	GizmoWorks
Power gizmo	$29.99	gadgets	GizmoWorks
SingleTouch	$149.99	photography	Canon
MultiTouch	$203.99	household	Hitachi