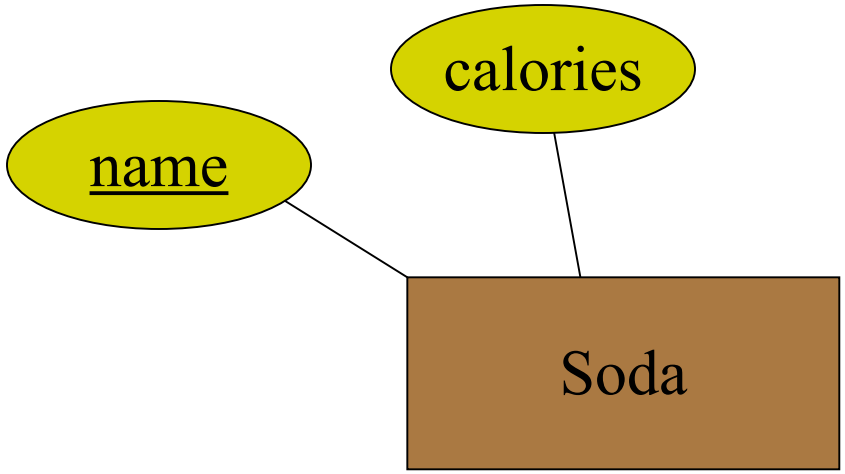
From ER Diagrams to the Relational Model

Rose-Hulman Institute of Technology

Curt Clifton

Review – Entity Sets and Attributes

- Entity set: collection of "things" in the DB
- Attribute: property of an entity

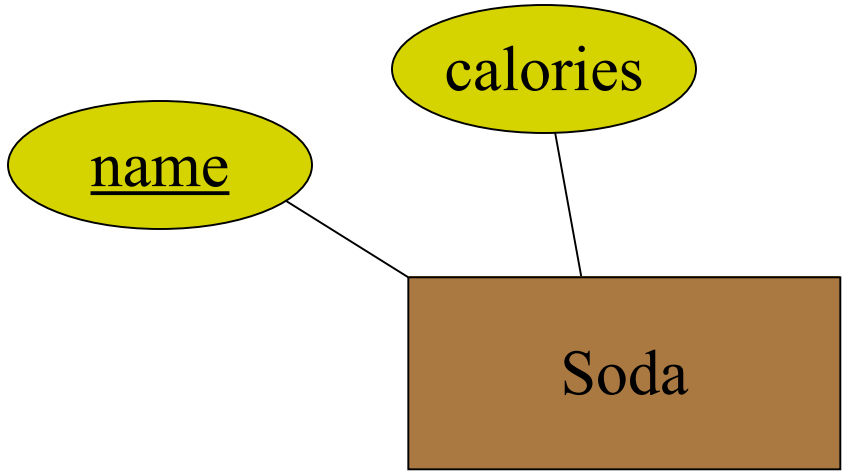


Kinds of Attributes

- Simple – single atomic value
  - Soda name, calories
- Composite – several sub-attributes
  - PersonName(First,Middle,Last)
- Multi-valued – set of values for one attribute
  - Car color, Degrees earned
  - (Somewhat rare, makes some searches harder)

Review – Keys

- Let us tell entities apart
- The key for an entity set is a **subset of the attributes** for that entity set, **such that no two entities agree** on all the attributes



Review – Relationships

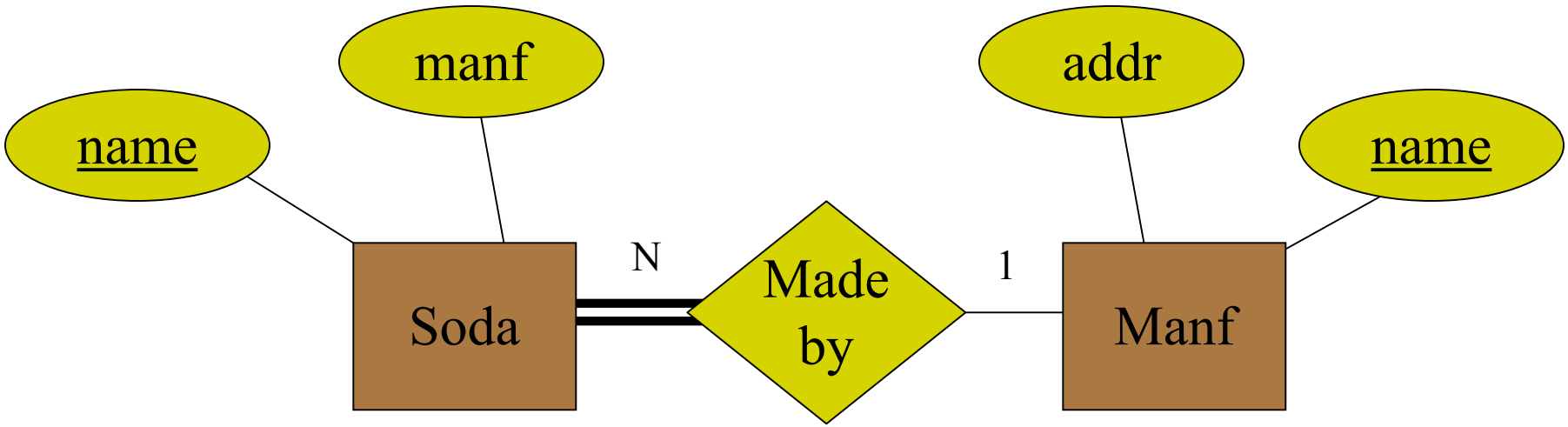
- Associate 2 or more entity sets
- Constraints
  - Maximums shown with numbers
    - Read like: a subject-verb-**number**-object
  - Participation shown with double line
    - Read like: a subject-**has to**-verb…

ER Design Techniques

- Avoid redundancy and don't duplicate data
- Don't use entity set when attribute will do
- Limit use of weak entity sets

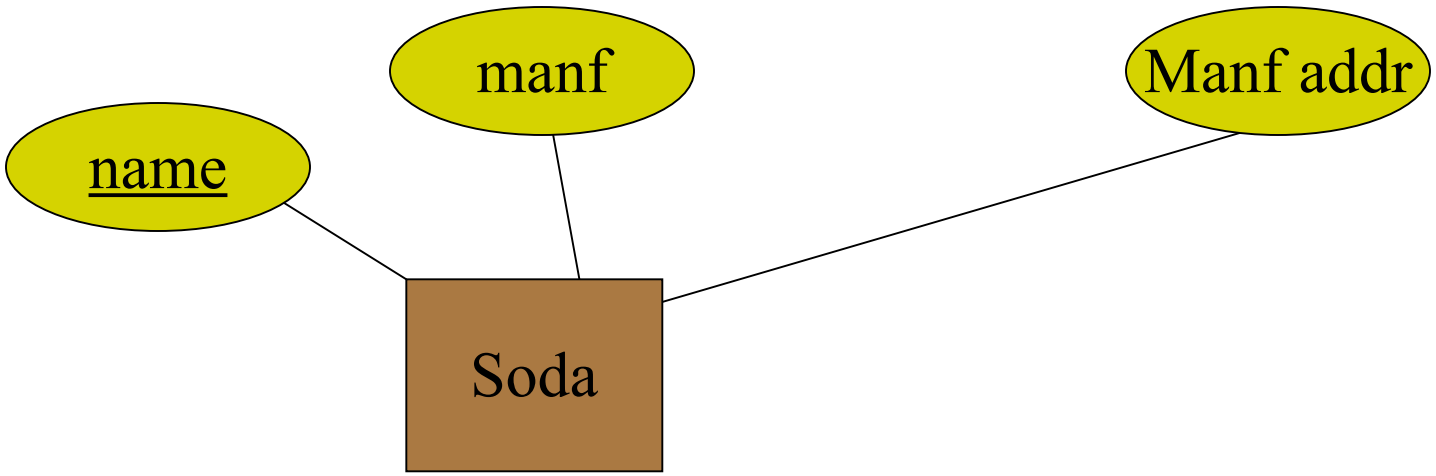
Redundancy

- Wastes space
- Leads to inconsistency
- For example:



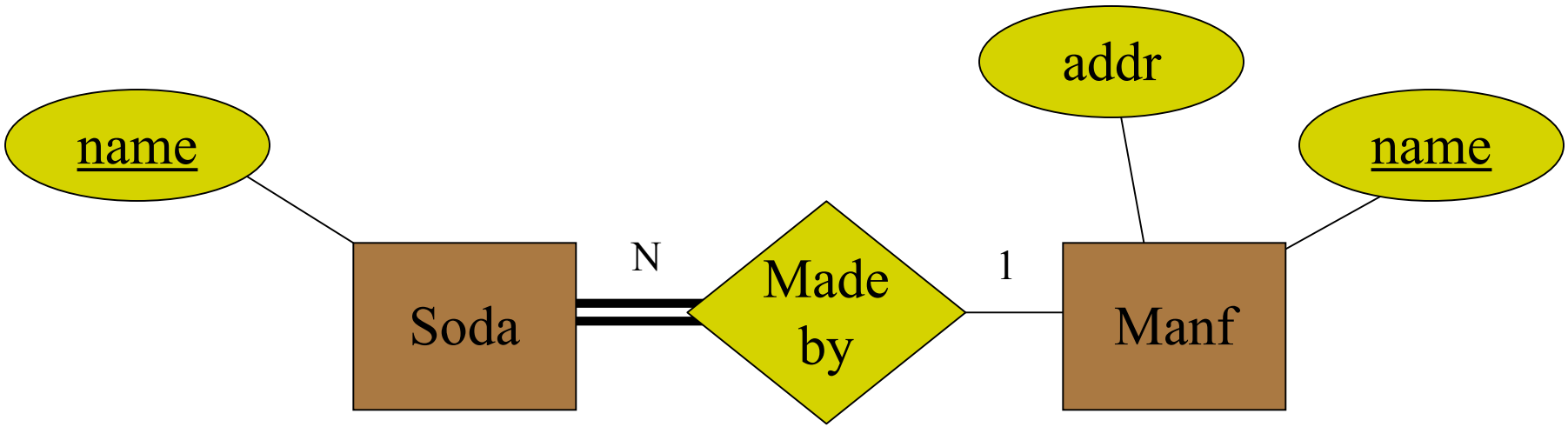
Failed Attempt At Fix

- Delete Manf entity set
- Add address to Soda



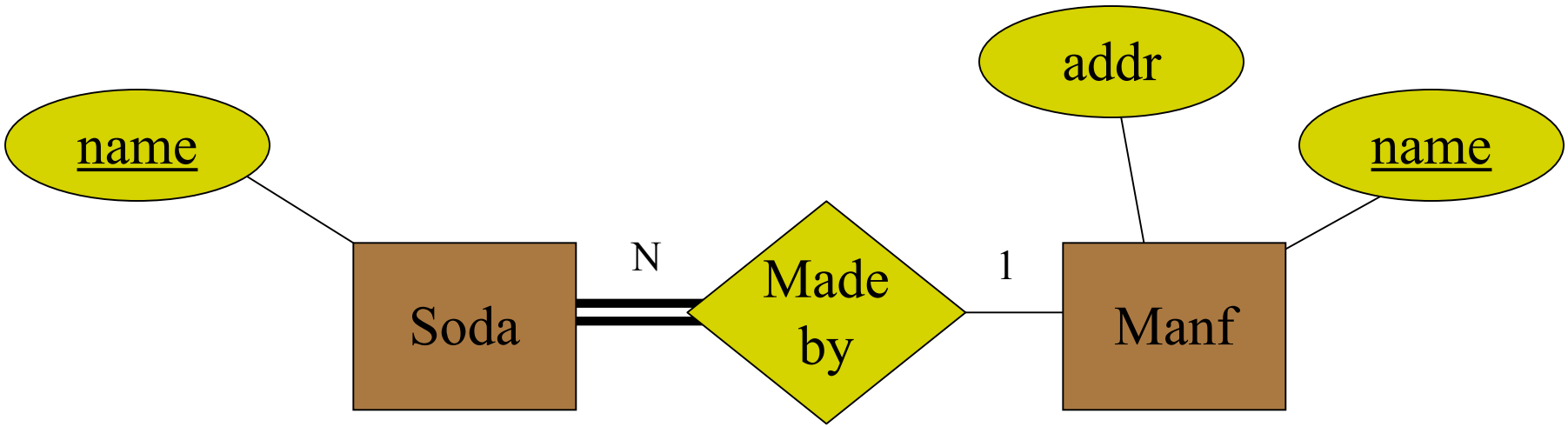
Successful Fix

- Eliminate manf attribute from Soda
- Use relationship to find manufacturer info.



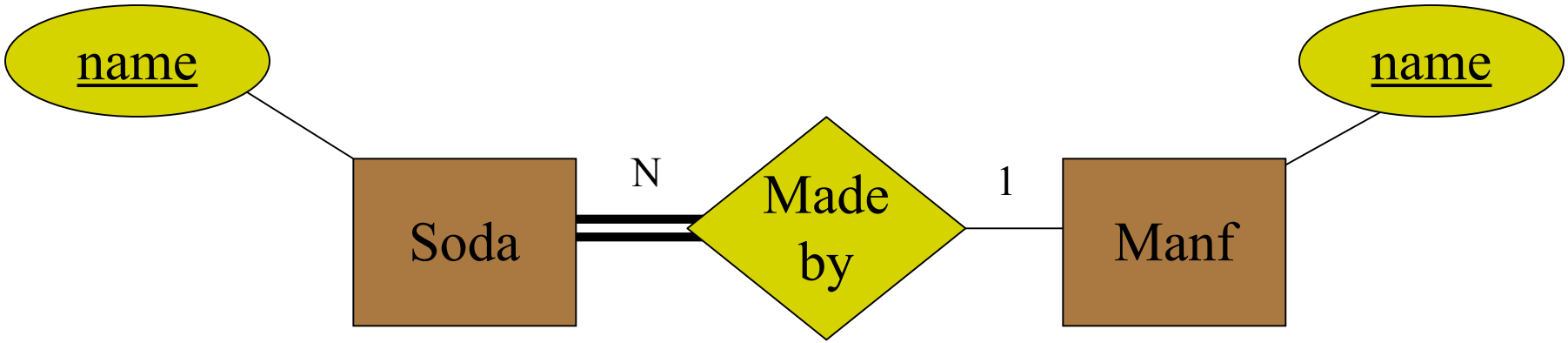
Don’t Use Unnecessary Entity Sets

- Entity set should…
  - Have at least one non-key attribute OR
  - Be the “many” in a many-one or many-many relationship



Bad Entity Set

- Suppose we didn't have manufacturer address



Avoid Weak Entity Sets

- Don't try to be too clever
- Can usually just add a unique ID
- Government has done this for their databases:
  - Social Security Numbers
  - Vehicle Identification Numbers
- But…
  - Don't trust uniqueness of IDs assigned by others

Why Use Weak Entity Sets At All?

- Federated Databases, for example…
  - All students in Indiana receiving state aid
  - All players on FIFA soccer teams
- One query sent to multiple DB
- Still want a Conceptual DB Schema
- But **no global authority** to assign unique IDs

The Relational Model

- Originated as theoretical idea
  - "A Relational Model of Data for Large Shared Data Banks", E. F. Codd, *Comm. of the ACM*, 13(6), June 1970
  - http://www.acm.org/classics/nov95/s1p3.html
- Revolutionized databases
- Led to 1981 ACM Turing Award
  - The "Nobel Prize of computing"

Some Terms

- Relation Schema
  - Relation

Relations

- (Semi-) Formally
  - Tuple: an ordered list
  - *n*-tuple: an ordered list of length *n*
  - Relation: a **set** of *n*-tuples
- Informally:
  - Relation: a table with unique rows
  - Rows = tuples; Columns = attributes;
  - Values in column = domain
- *Database*: a collection of relations

Some Other Terms

- Relation schema
  - Describes a relation
  - *RelationName (AttrName1, AttrName2,…)*
  - Or *RelationName (AttrName1:type, …)*
- Database schema
  - Set of all the relation schema for the DB's relations

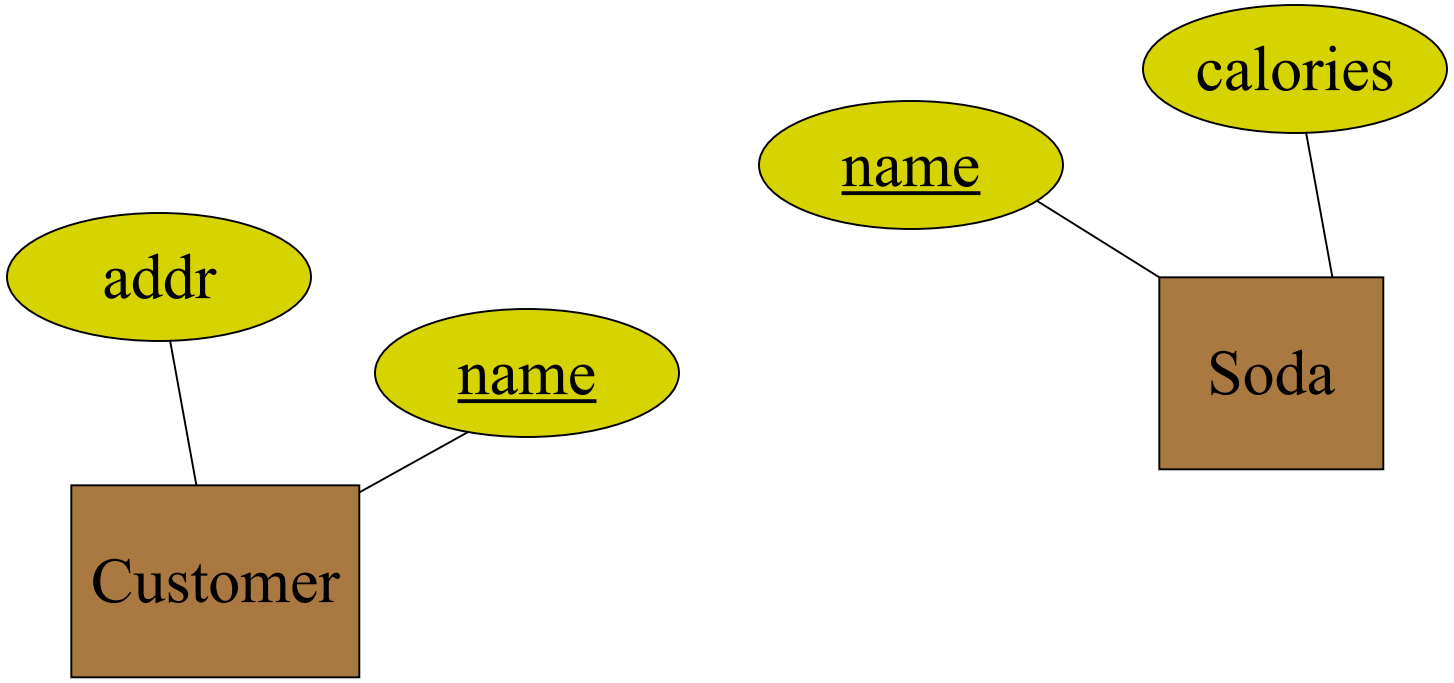
Why is the Relational Model Dominant?

- Very simple – just one data structure
- Matches a “list the items” mentality
- Easy to manipulate tables with UI
- Forms basic foundation for SQL
  - Relational model based on sets
  - SQL based on bags (a.k.a., multi-sets)

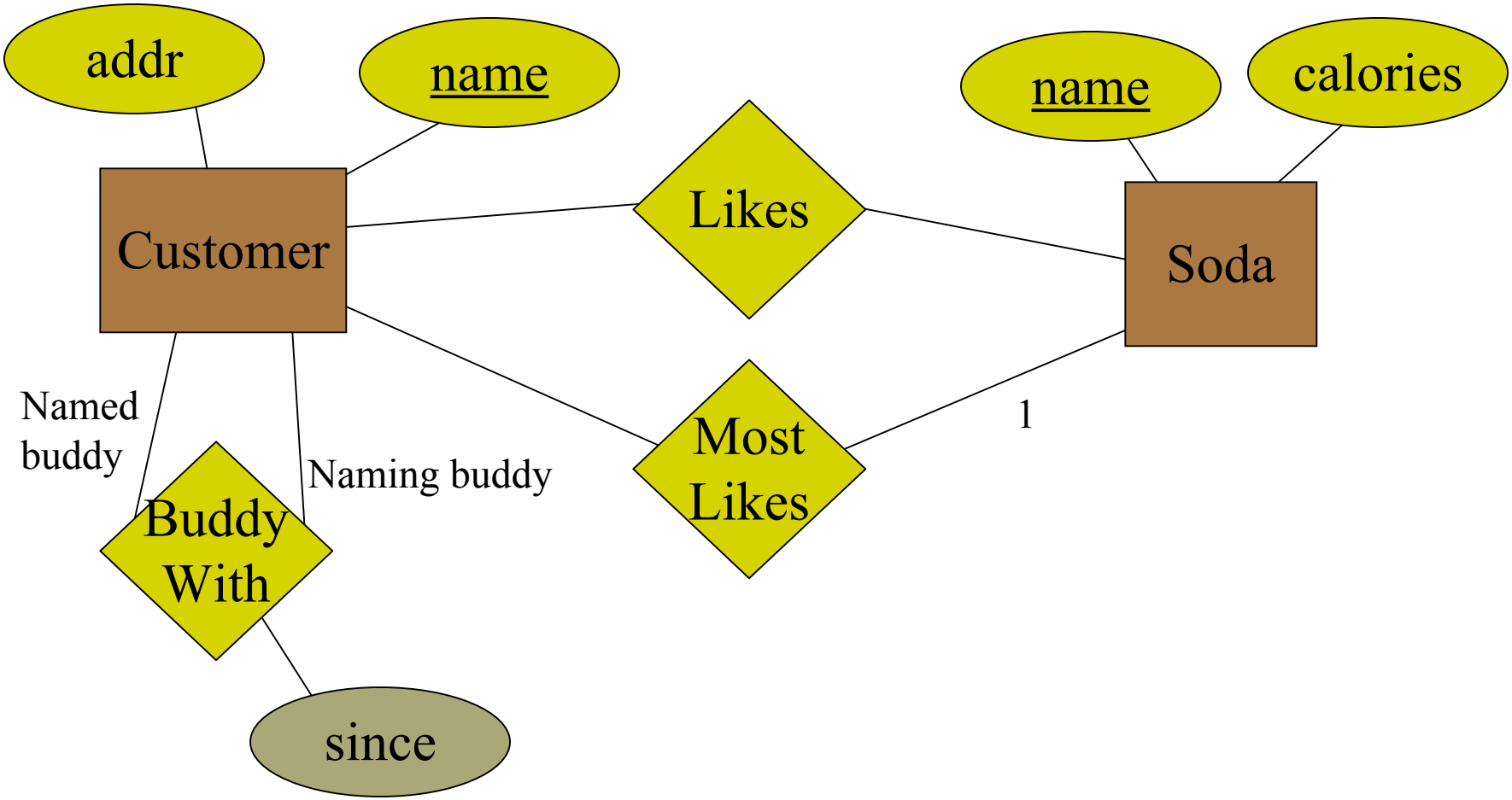
From ER Diagrams to Relations

- Entity sets become relations
  - Columns are attributes of entity set
- Relationships also become relations
  - Columns are keys of participating entity sets

Example: Basic Entity Sets

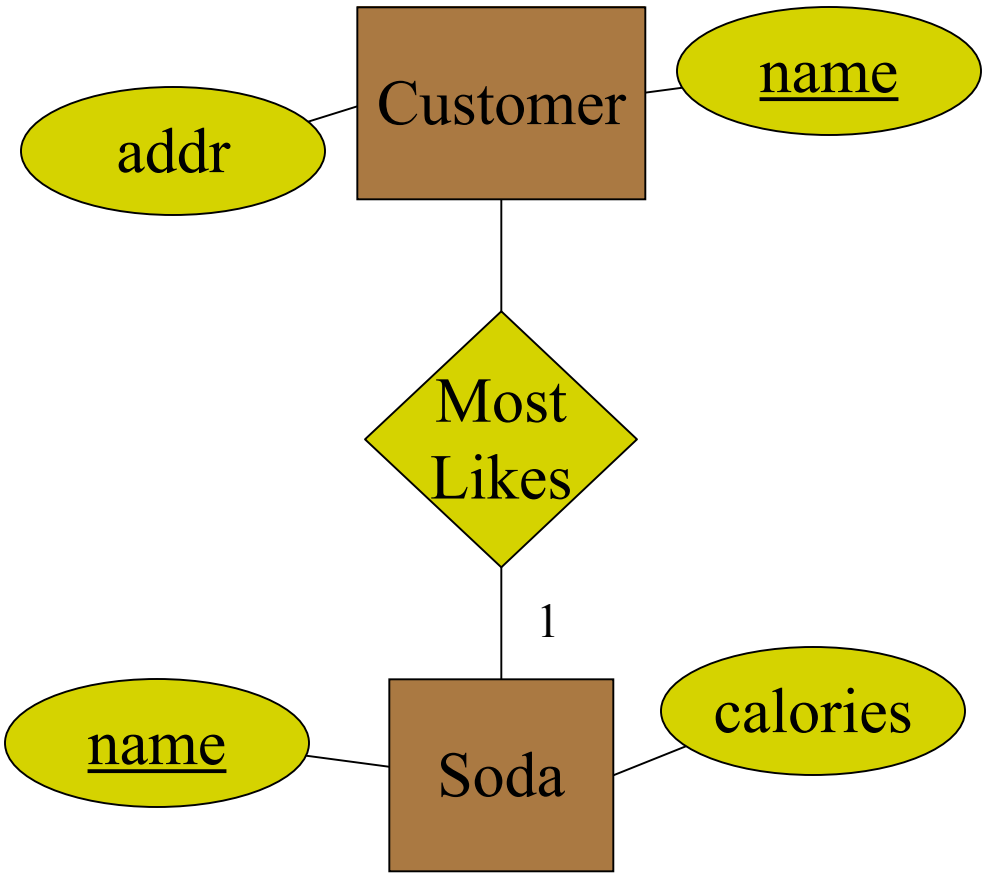


Example: Basic Relationship



Simplifying!

- Can avoid relations for **many-one** relationships
- Just add key of the **one** to the relation of the **many**

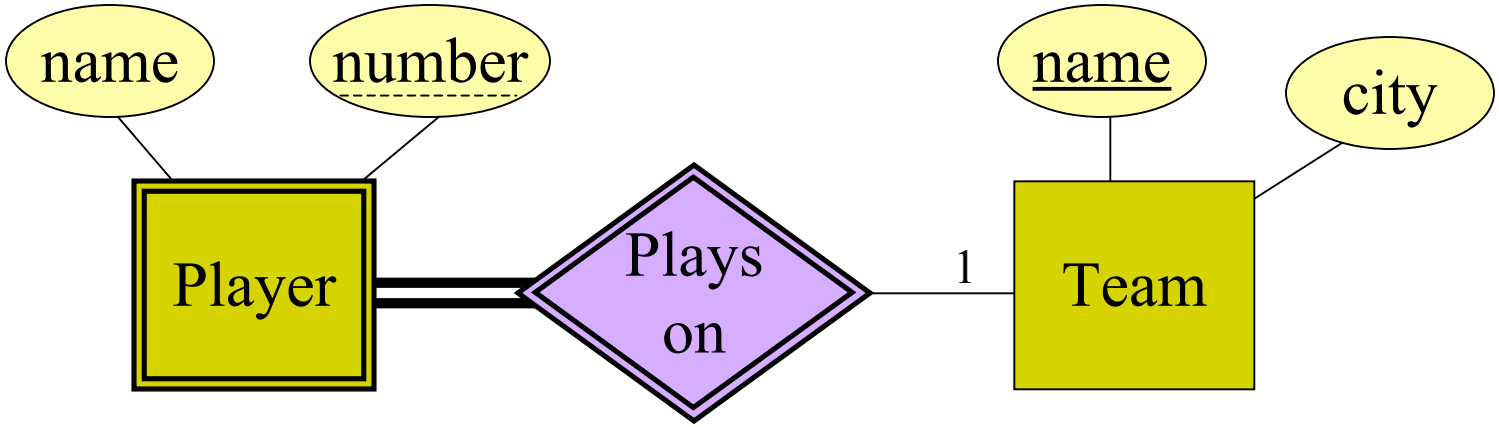


Over Simplifying!

- What happens if we try to eliminate relation for a many-many relationship?
- Consider treating Likes as we did Most Likes
  - Redundancy
  - Data loss

Weak Entity Sets

- Need enough columns to make rows unique!
- So…
  - All attributes of weak entity set
  - + Key from supporting relationship



Entity Sets with Subclasses

- Use nulls, *or*
- Use multiple relations
  - "ER Style"
- How should we choose which to use?