Database Design: ER and UML Diagrams

Abdu Alawini

University of Illinois at Urbana-Champaign

CS411: Database Systems

October 1, 2018

Annoucements

- HW 1 is due today
- HW 2 will be posted on Wednesday
- Project Track 1 – Stage 1 is due on Wednesday
- The class is full and registration is close
  - no students from the waitlist will be allowed to register ☹ ☹ ☹

Overview of Database Design

- Conceptual design: (ER & UML Models are used for this.)
  - What are the entities and relationships we need?
- Logical design:
  - Transform ER design to Relational Schema
- Schema Refinement: (Normalization)
  - Check relational schema for redundancies and related anomalies.
- Physical Database Design and Tuning:
  - Consider typical workloads; (sometimes) modify the database design; select file types and indexes.

We'll do this later

Entity-Relationship Model is a different model than the Relational Model

- Relational model has:
  - tables (relations) with attributes, keys, foreign keys, domain definitions for attributes
- Entity-Relationship model has:
  - Entities and entity sets with attributes, keys, and domain definitions for attributes
  - Relationships among entities and relationship sets with uniqueness or cardinality constraints

Entity Relationship Model
Unified Modeling Language

ER Model

- Proposed by Peter Chen in 1976
- Gives us a language to specify
  - What information the database must hold
  - How the bits of information relate to one another

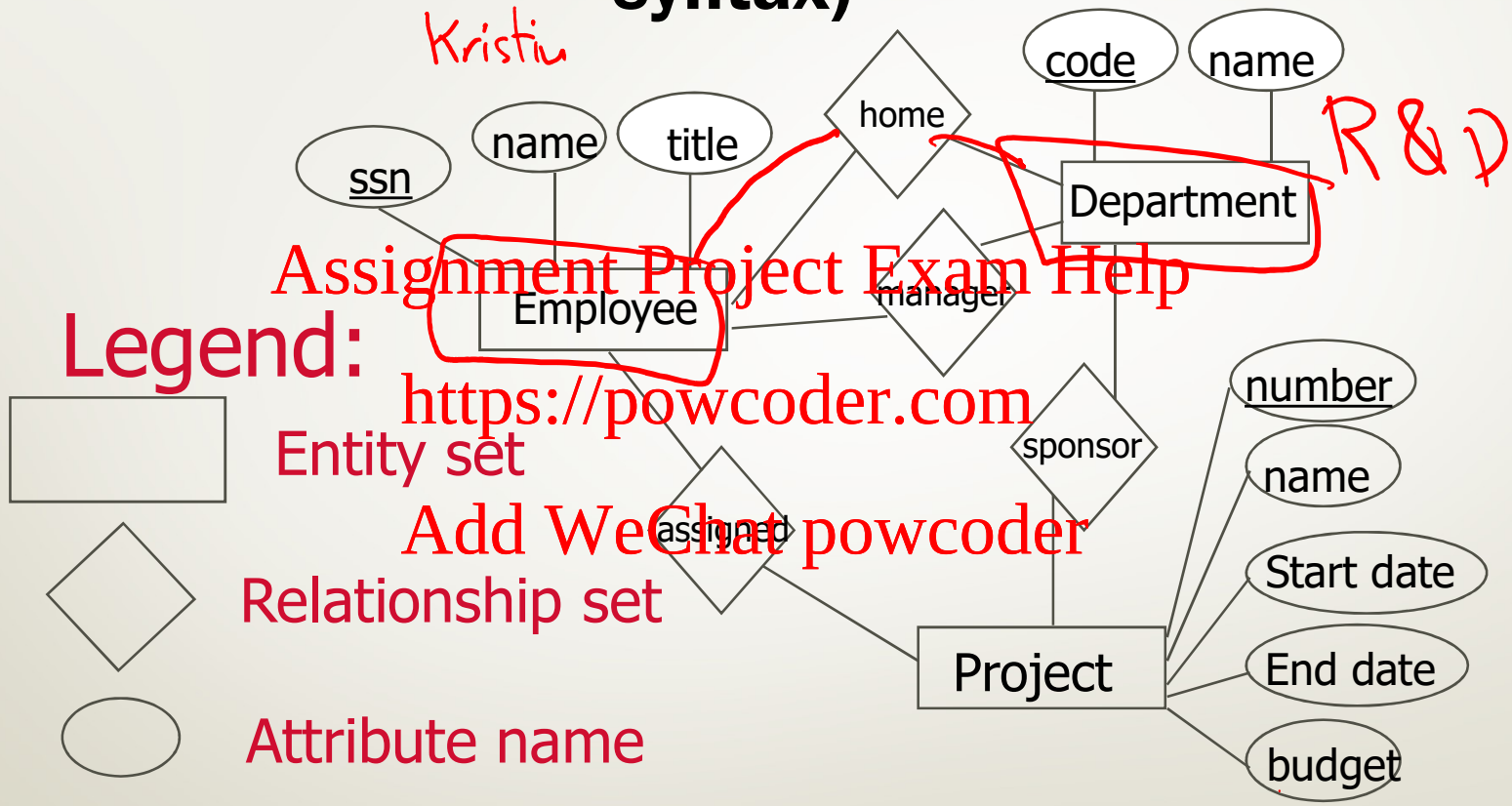
UML Model

- UML is a standard language for designing software systems
  - also used for DB design
- created by the Object Management Group (OMG)
- **UML** 1.0 specification draft was proposed to the OMG in early 1997.

Agenda

- basics of ER and UML models
- constraints
- weak entity sets

Entity-Relationship Diagram (original syntax)



Definitions

- *Entity:* Real-world object distinguishable from other objects.

  An entity is described using a set of *attributes.*

- *Entity Set*: A collection of similar entities. E.g., all employees.

  (often referred to as just entity, which blurs the distinction between type and collection)

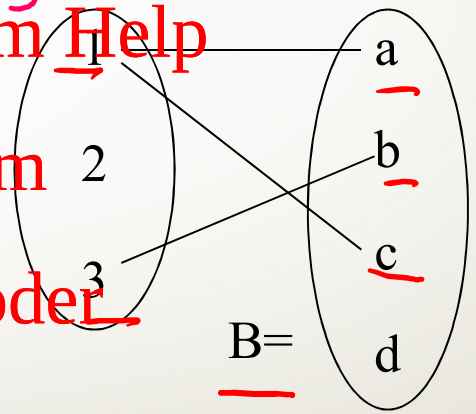
Definitions

- *Relationship*: Association among 2 or more entities. E.g., Kristin's home department is Research & Development.
- *Relationship Set*: Collection of similar relationships. E.g., Home (often referred to as just relationship).

Relationships

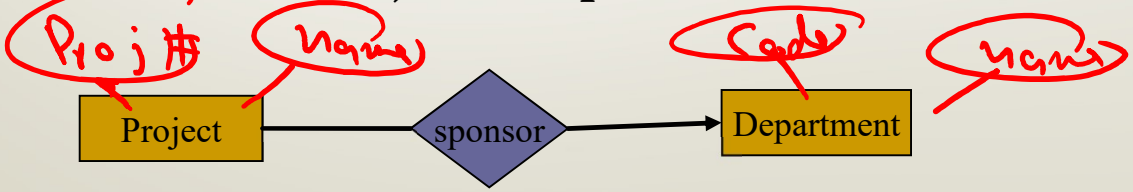
- Formal definition:
  - if A, B are sets, then a relation R is a subset of A x B
- A={1,2,3}, B={a,b,c,d},

  R = {(1,a), (1,c), (3,b)}



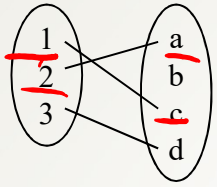
Same story w/ entity sets

sponsor is a subset of **Project** x **Department**:

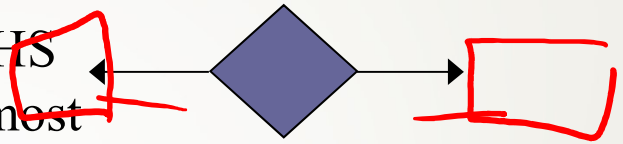


Multiplicity of E/R Relationships

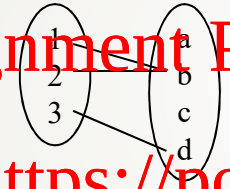
- one-one:



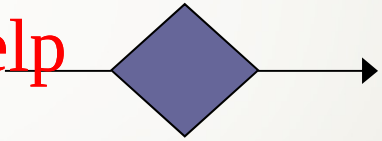
One on LHS/RHS connected to at most one on RHS/LHS



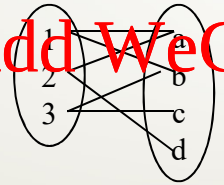
- many-one:



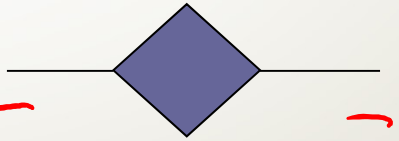
One on LHS connected to at most one on RHS



- many-many:



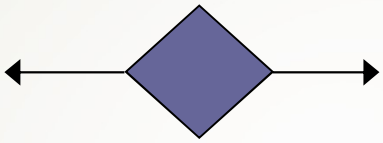
No constraints



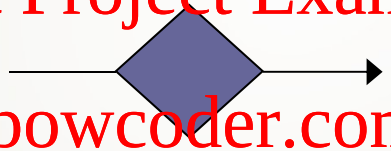
- Multiplicity can be shown with arrows
- Arrow = at most 1
- Another interpretation: "determines"

Q: Example scenarios for each case?

- one-one:



- many-one:



- many-many:

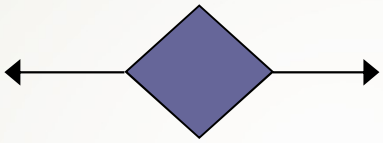
Department – head ??

Actor – play ??

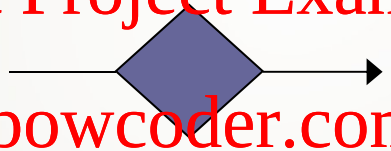
Employee – company ??

Q: Example scenarios for each case?

- one-one:



- many-one:



- many-many:

1-1

Department – head ?? *Each dep has 1 head, each head has 1 dep: one-one*

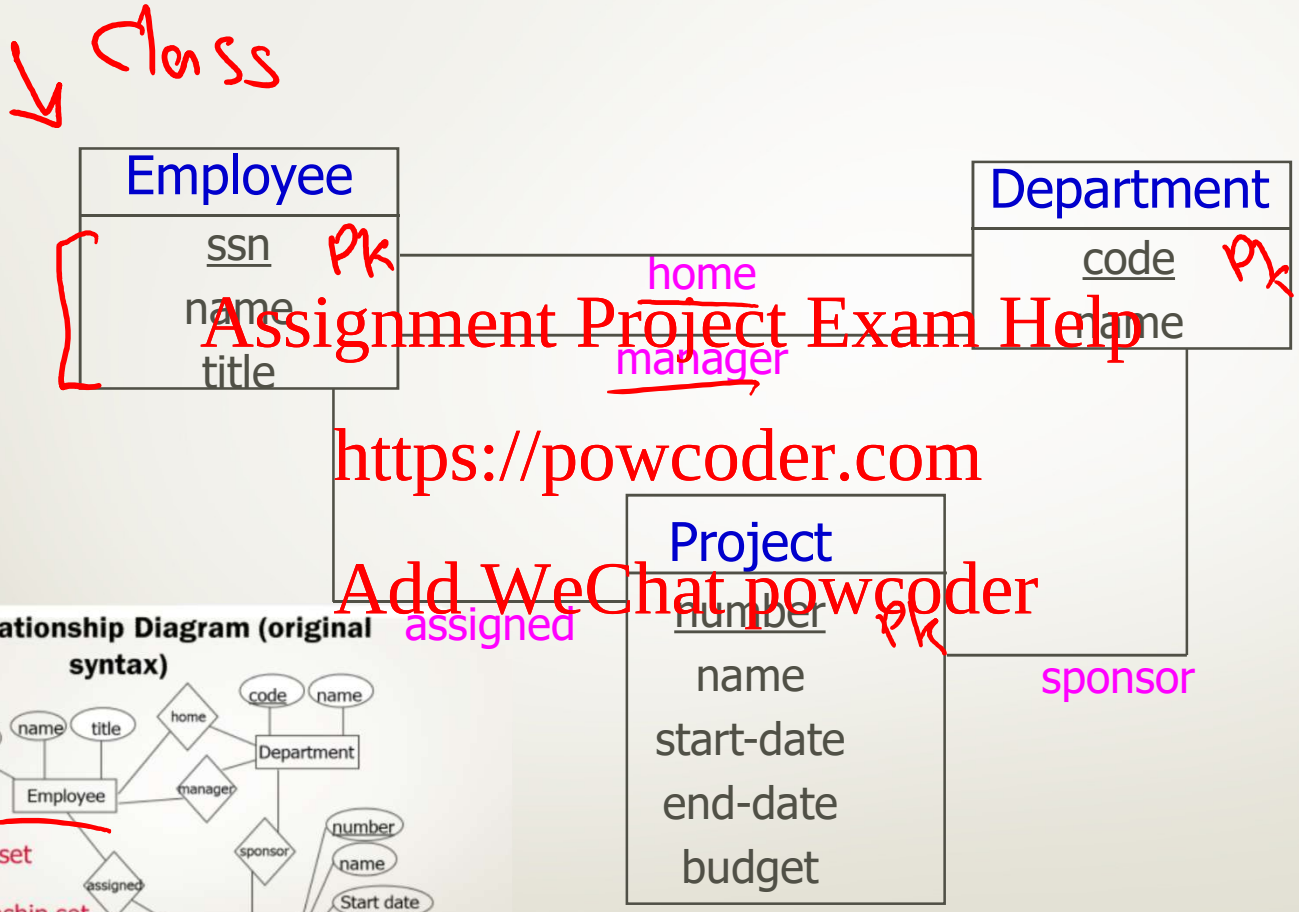
many-many

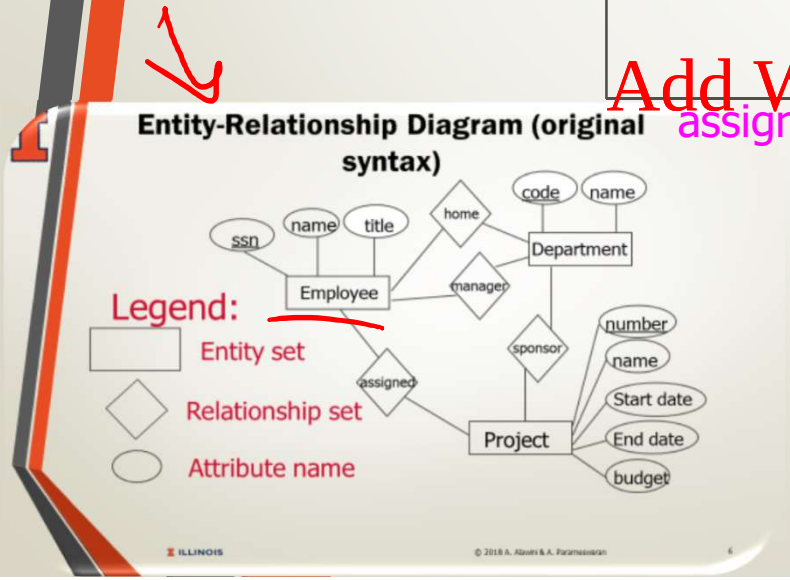
Actor – play ?? *Each actor has many plays, each play has many actors: many-many*

Employee – company ?? *Each employee has one company, each company has many employees*

many-one

UML version of the same E-R Diagram





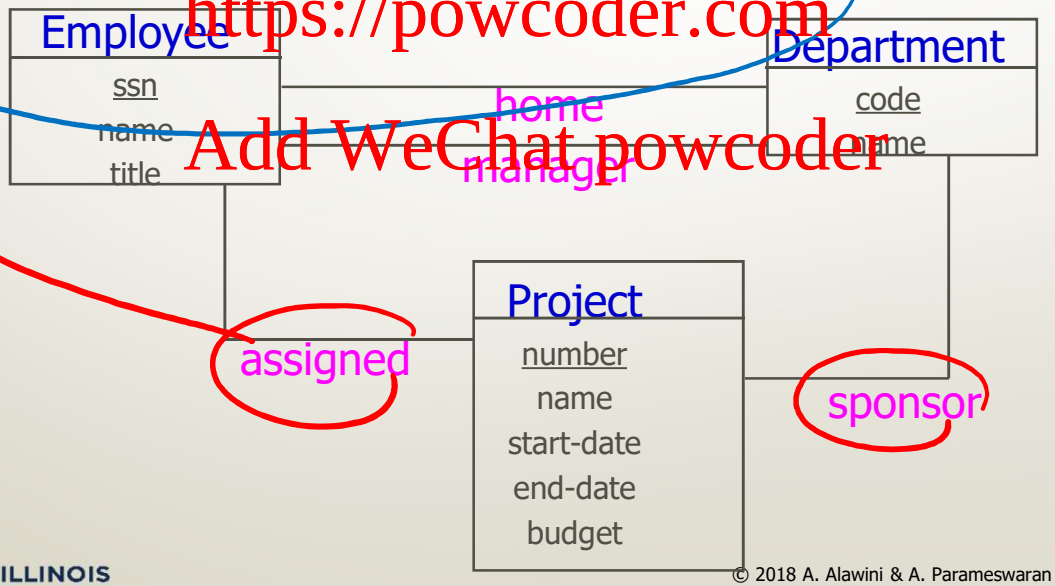
Equivalent Relational Schema

Employee (ssn, name, title, home-dept)

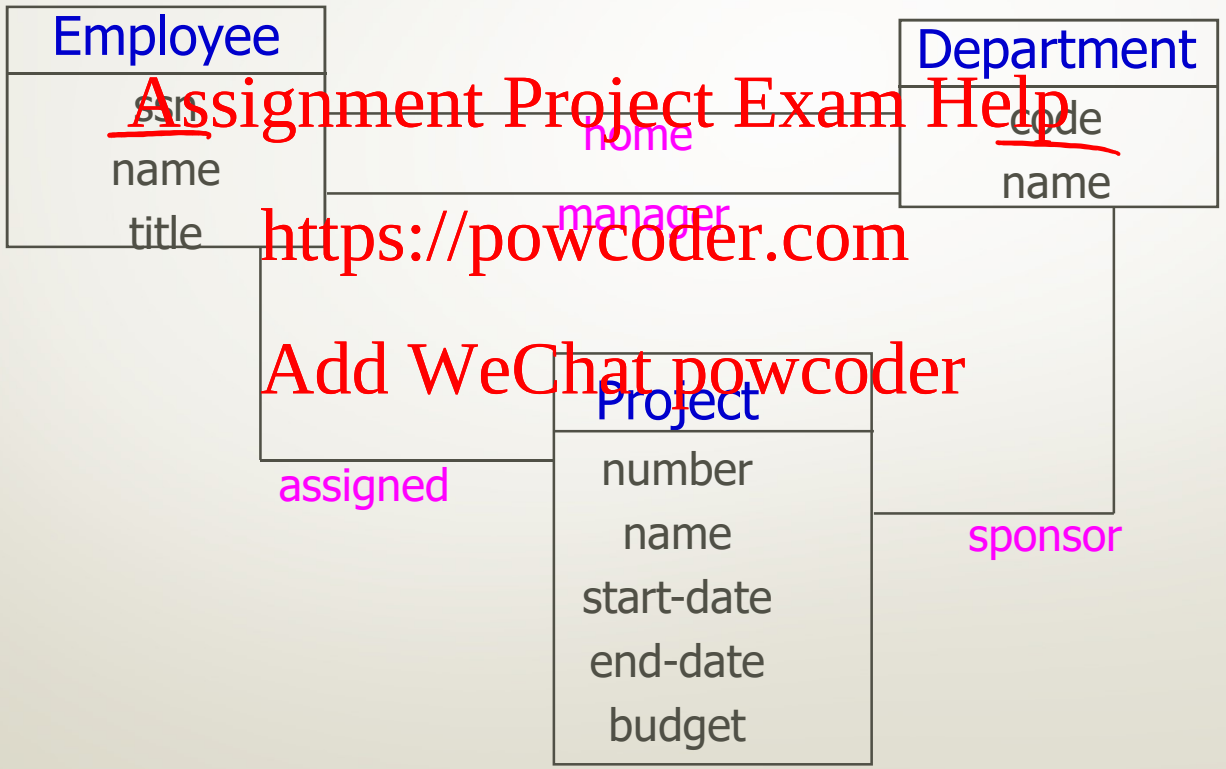
Project-team(ssn, number)

Department (code, name, manager)

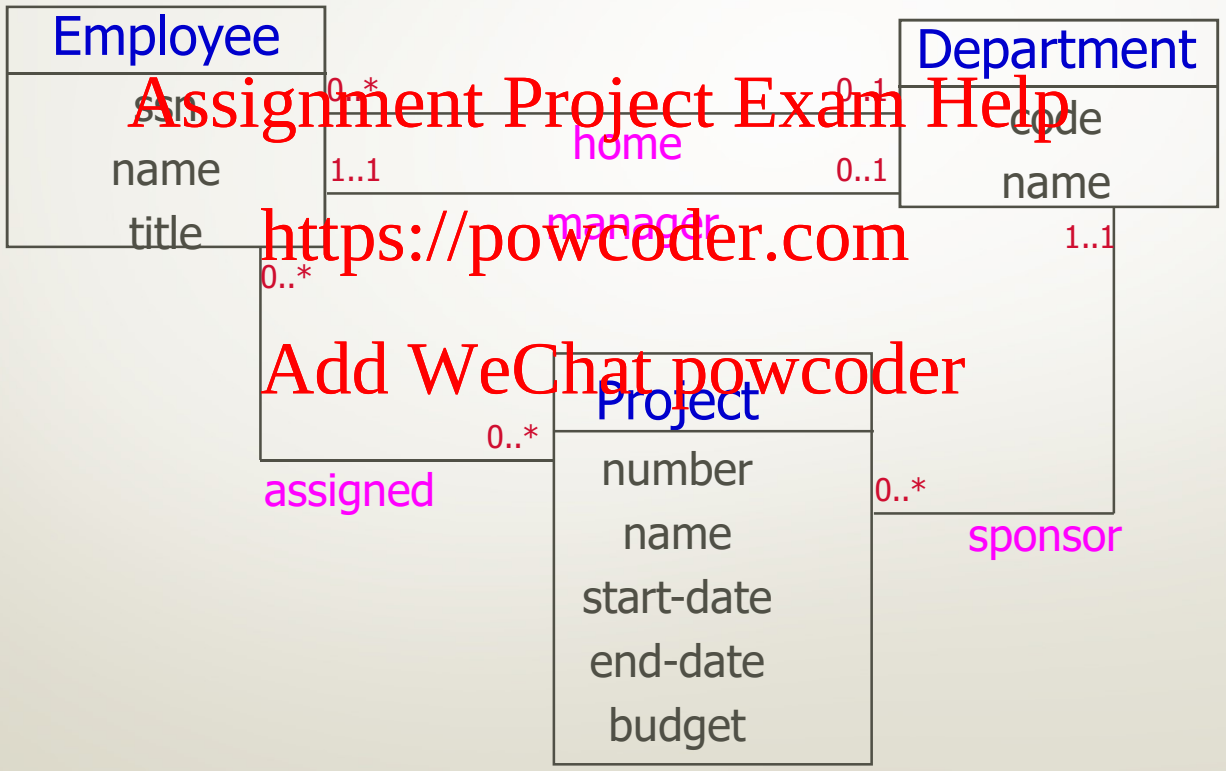
Project (number, name, start-date, end-date, budget, sponsor)

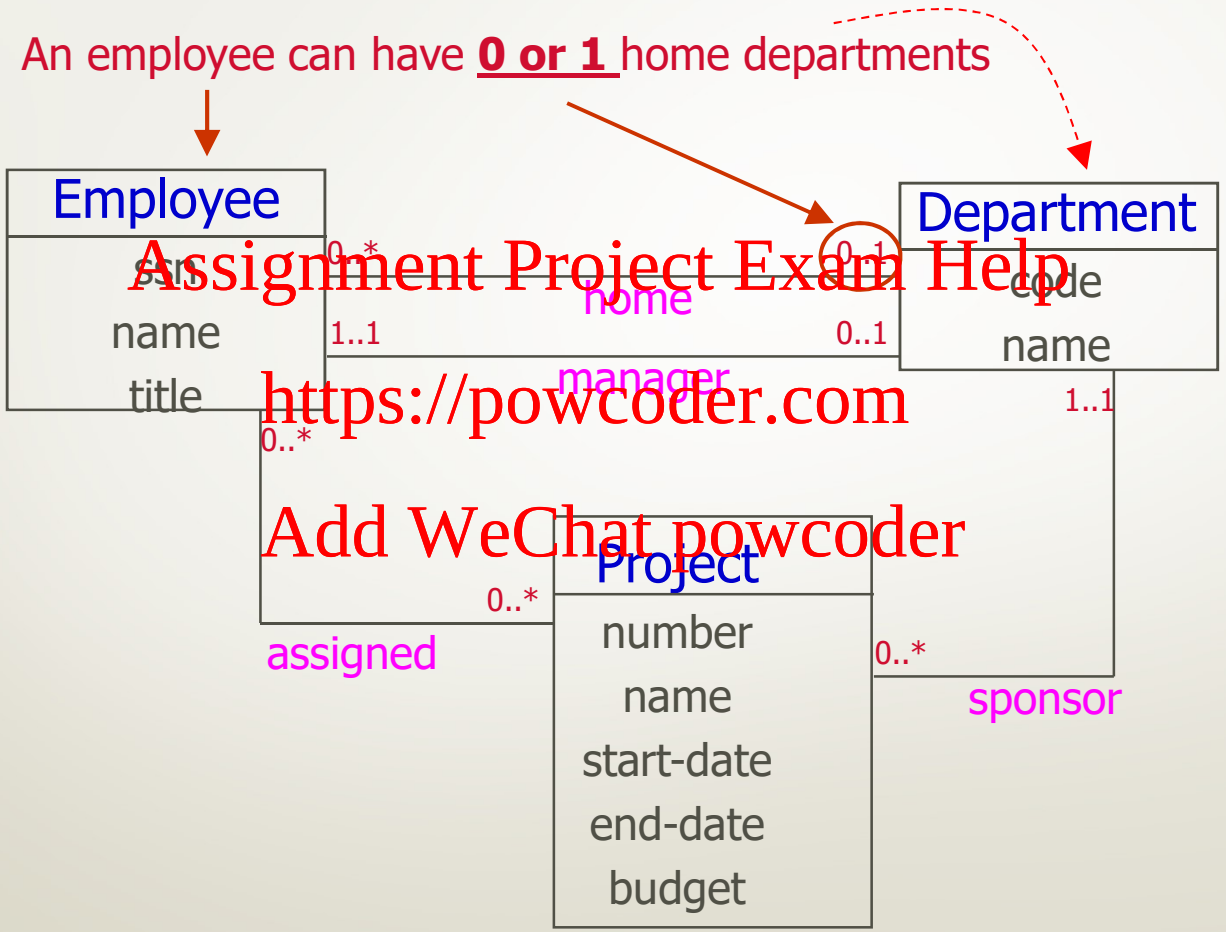


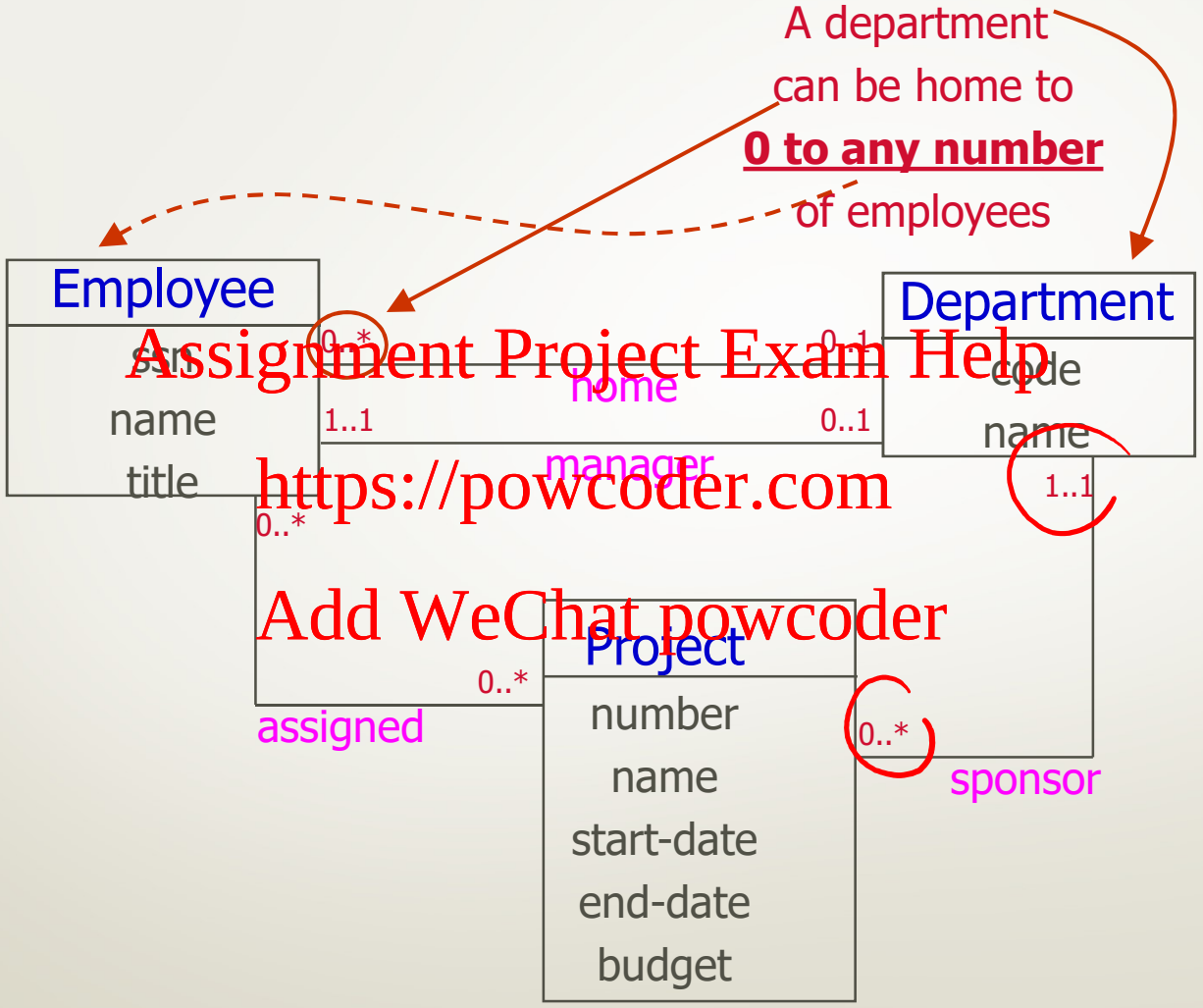
Cardinality Constraints on Relationship sets: How many entities can participate?



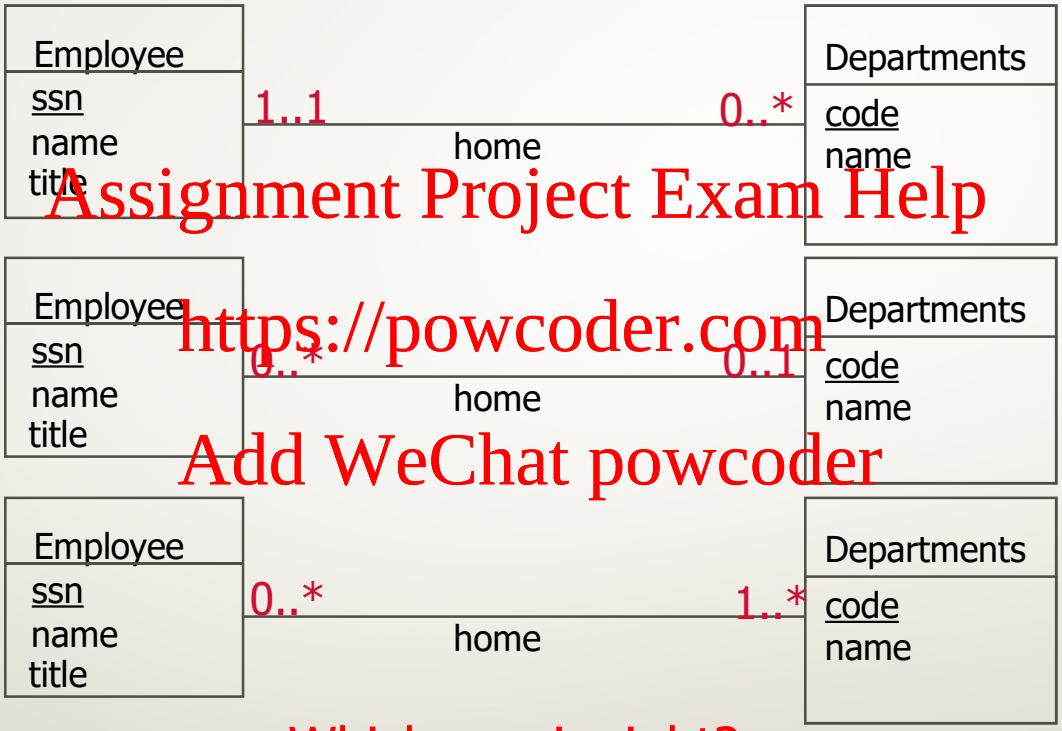
Cardinality Constraints on Relationship sets: How many entities can participate?







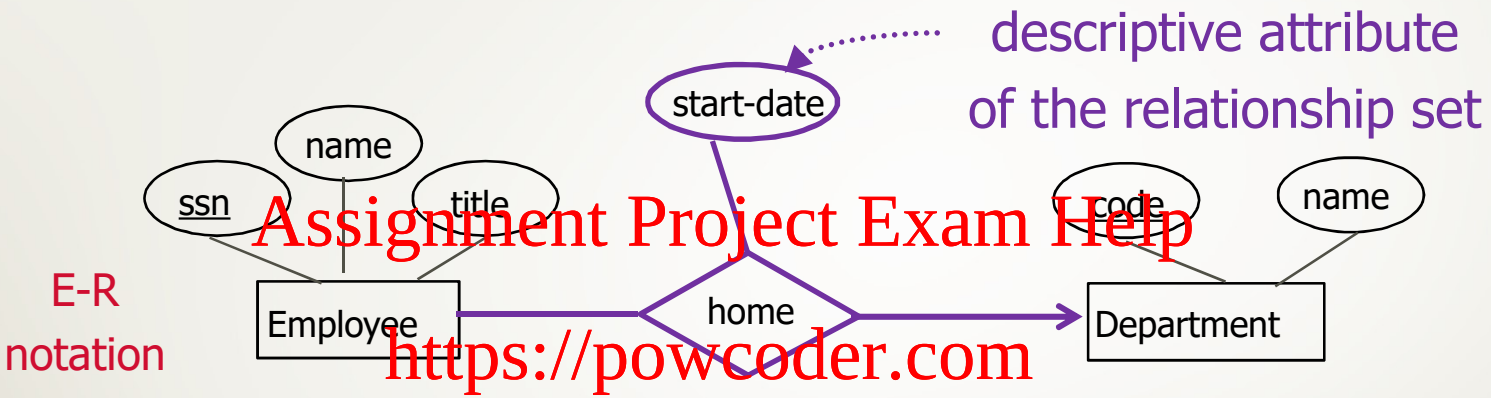
Some Alternative Constraints (in UML)

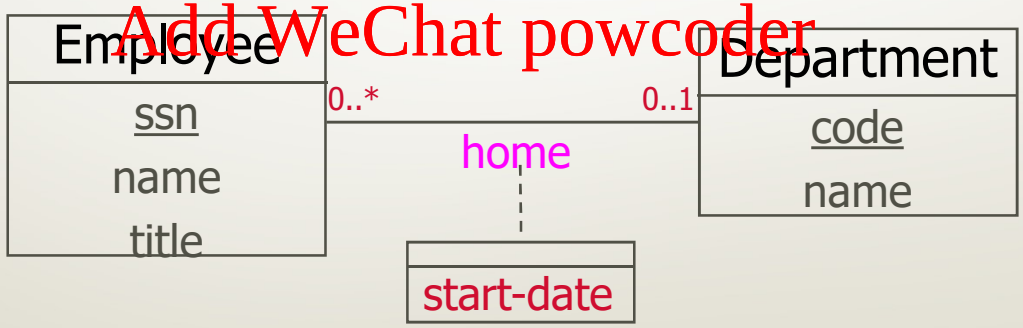


Which one is right?

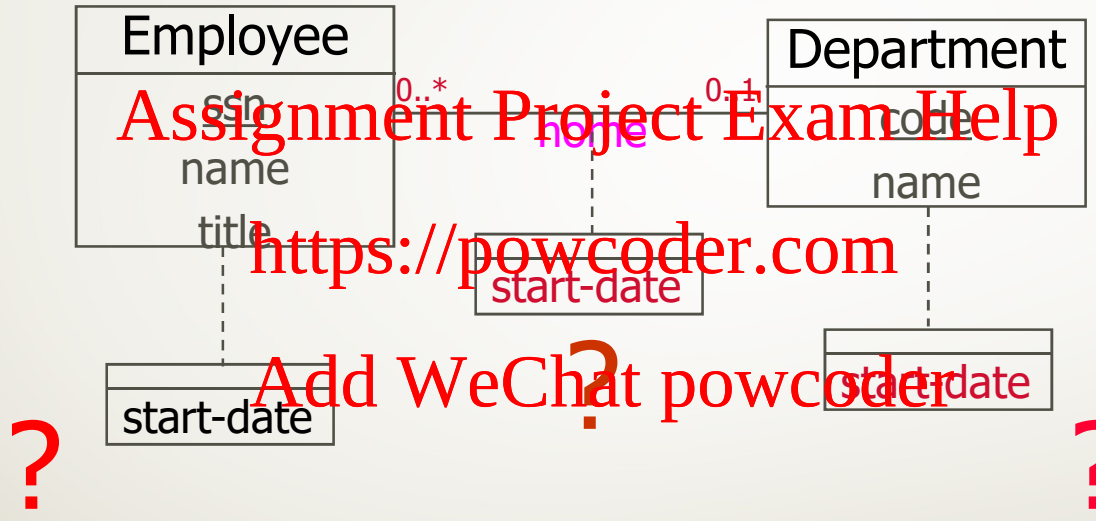
We must discover the semantics of the application!

Relationship sets can have attributes



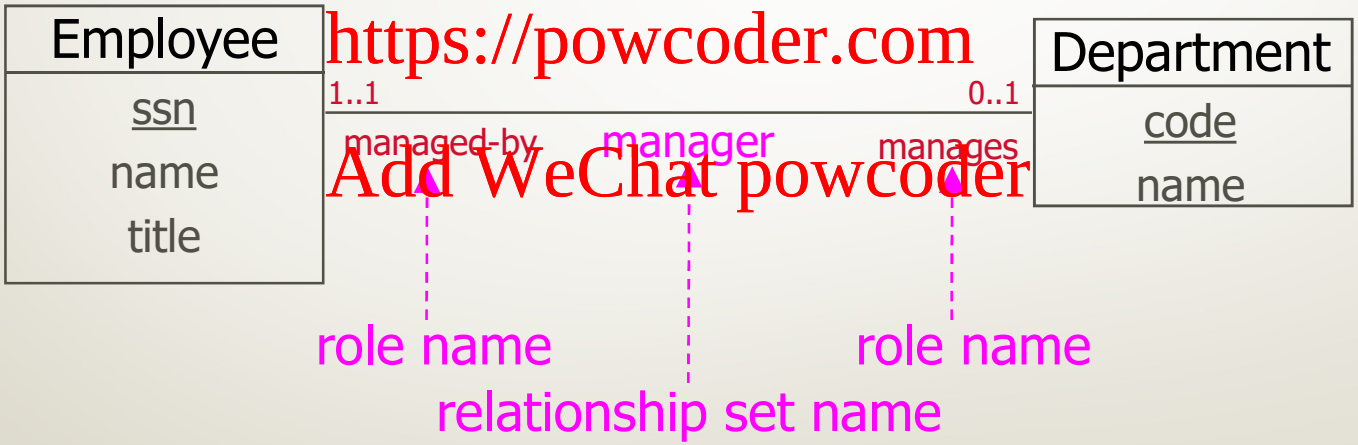


Try all three locations for the attributes: which one makes sense?

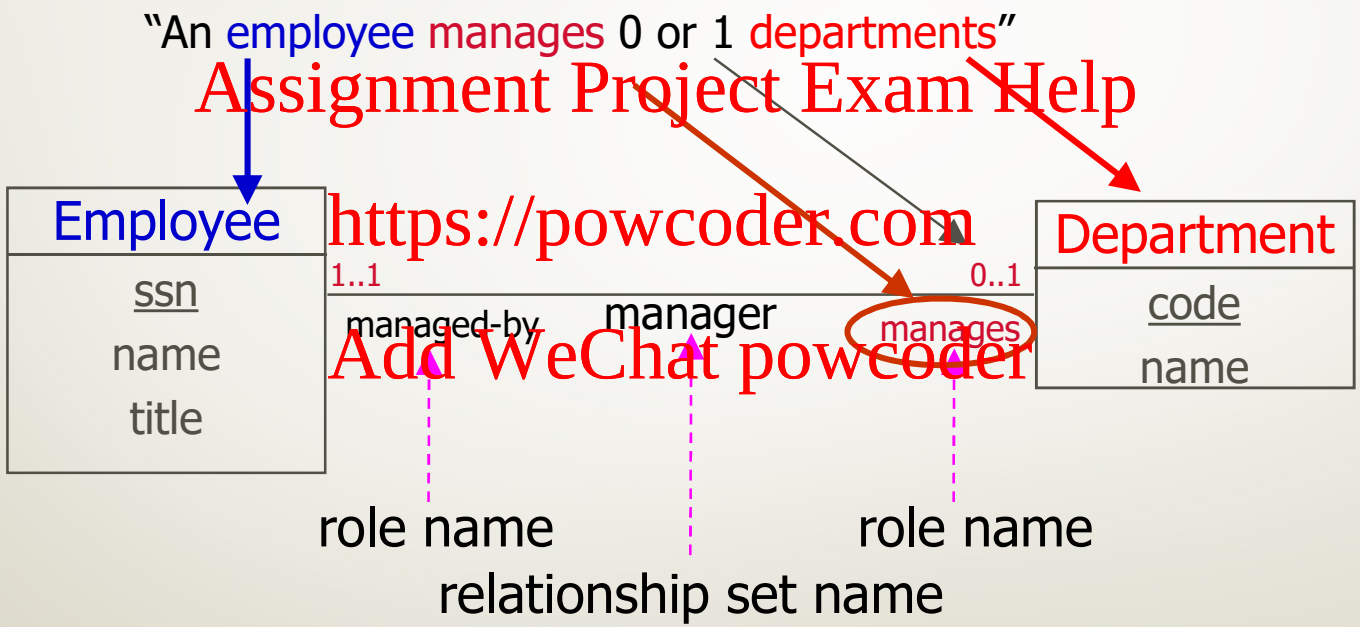


Relationship sets can have role names

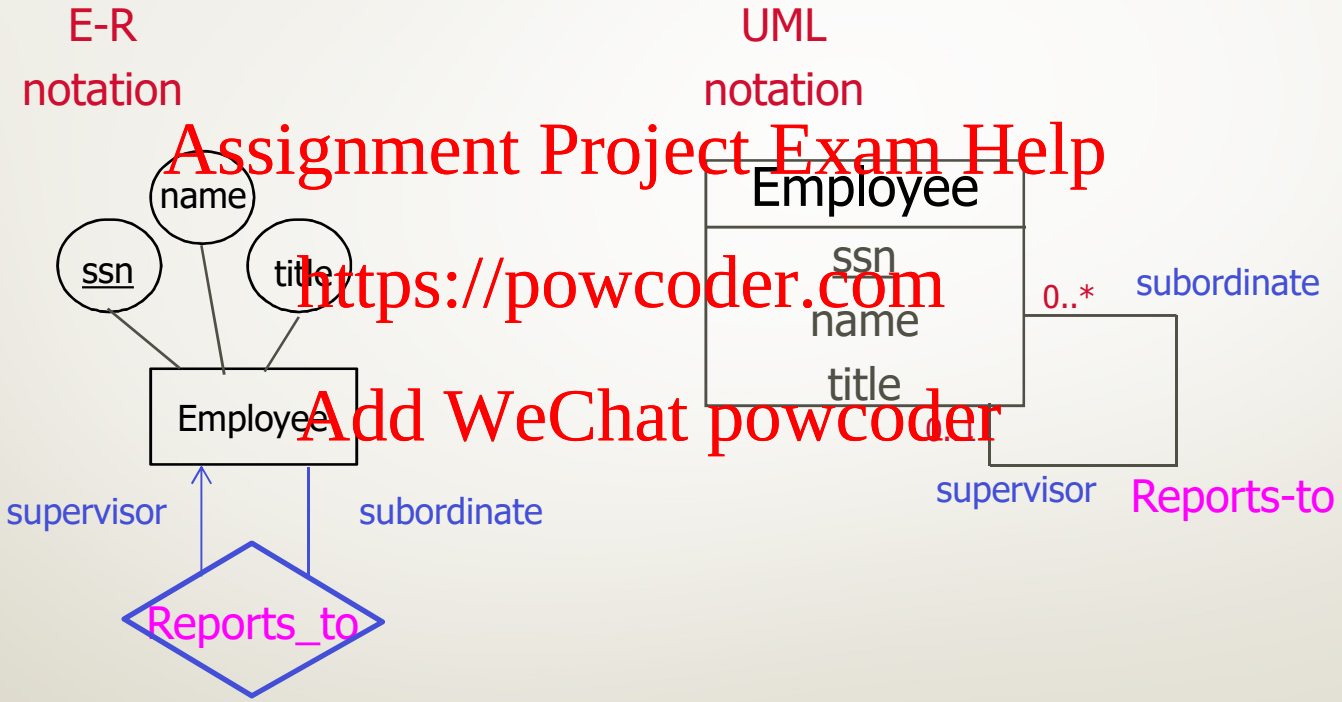
(in addition to the name of the relationship set)



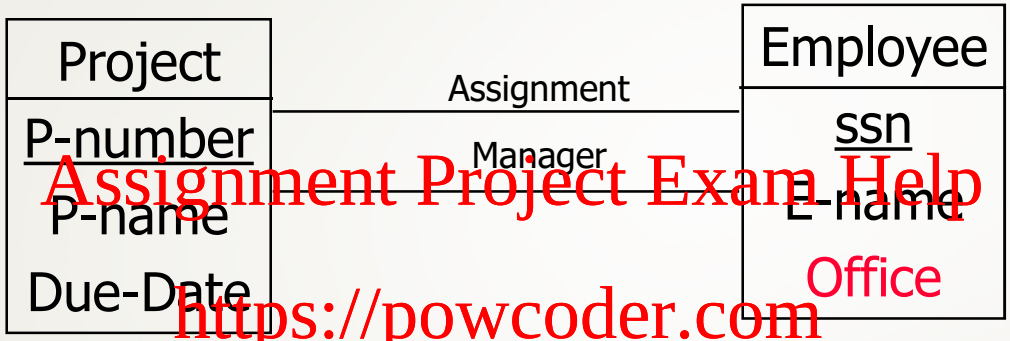
Example: reading role names



Same entity sets can participate in different "roles" for the same relationship set



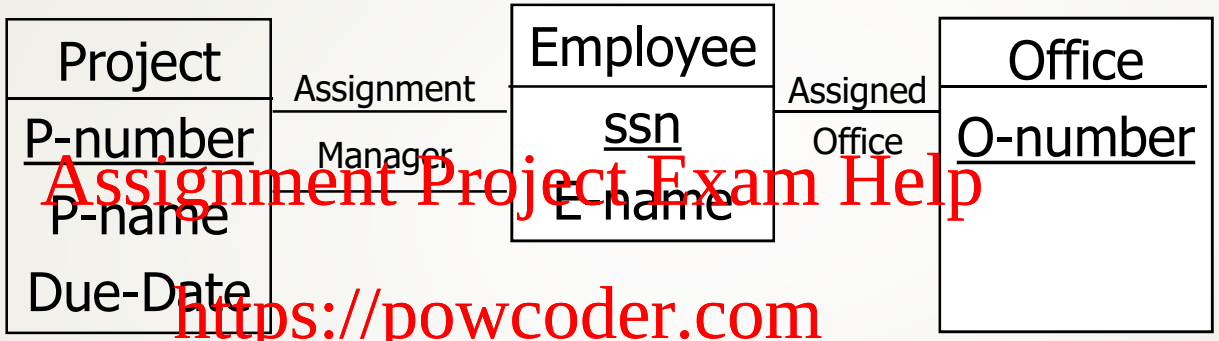
Duality: entity ⟺ value and attribute ⟺ relationship



Should Office be an attribute of Employee? Or a separate entity set? Most attributes can be "promoted" to an entity set and some entities can be "demoted" to an attribute value.

This explains why there are so many different ways to design a schema.

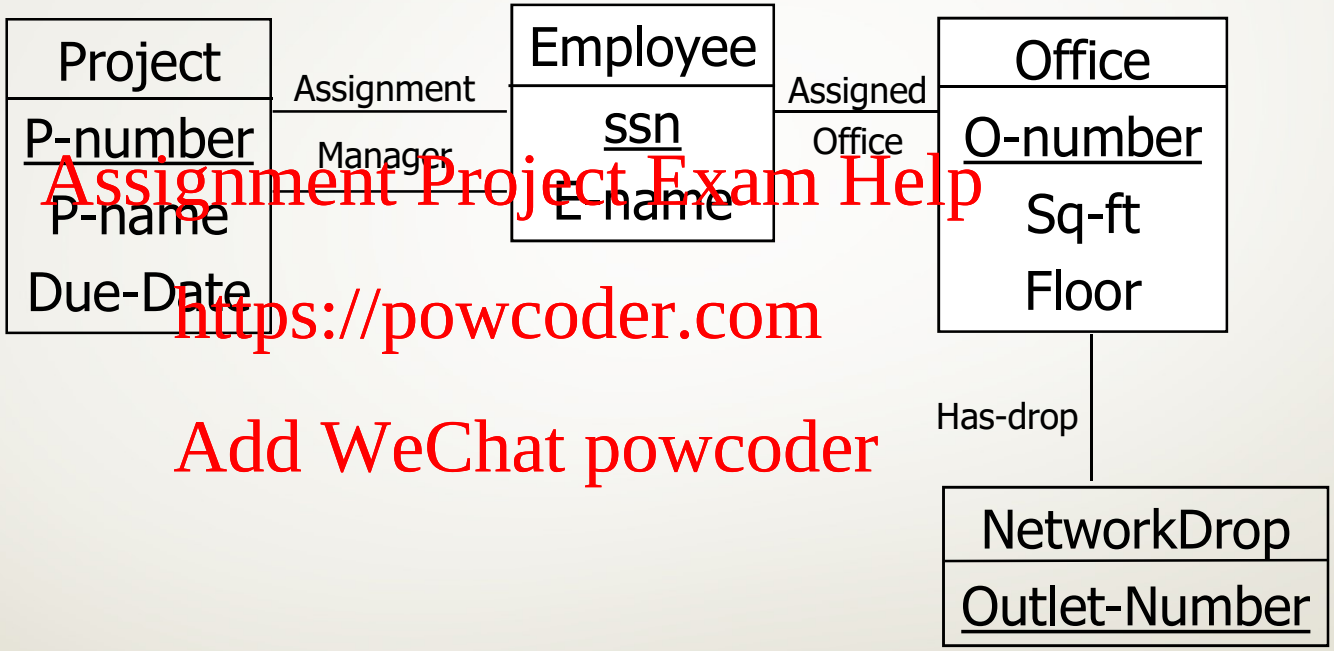
Entity vs. Value of an Attribute



What are some reasons to model Office as an entity set?

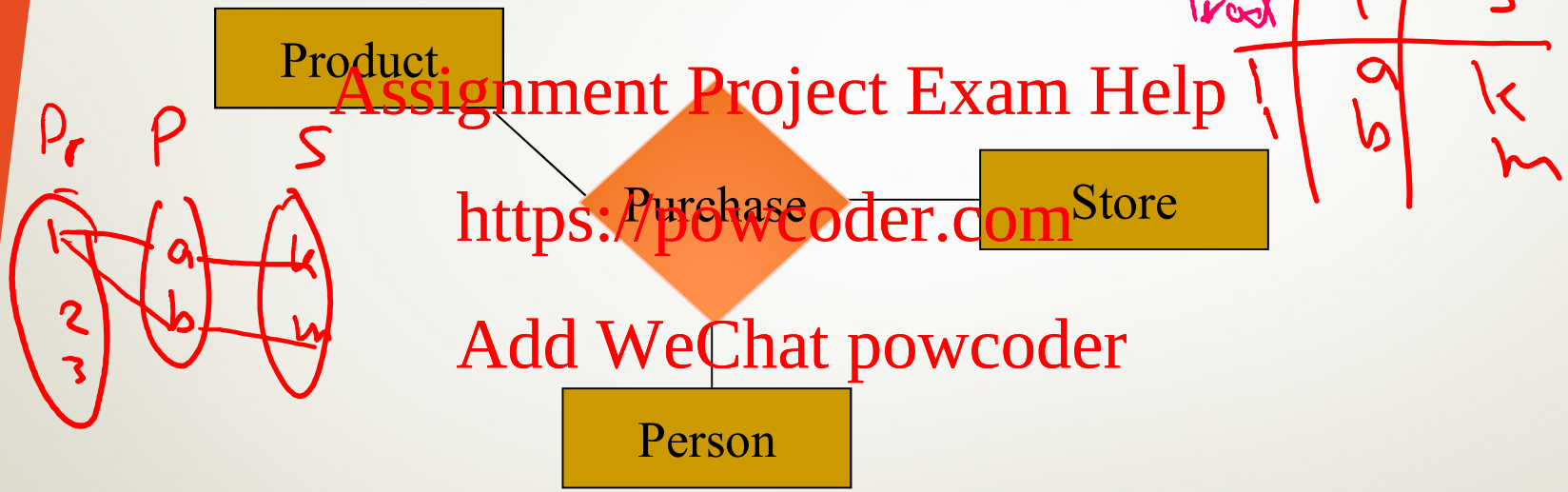
- An employee can have more than one office
- There are other attributes of Office
- Office needs to participate in other relationship sets such as a relationship set connecting to telephones jacks or network drops (located in the office)

Entity vs. Value of an Attribute



Multiway Relationships

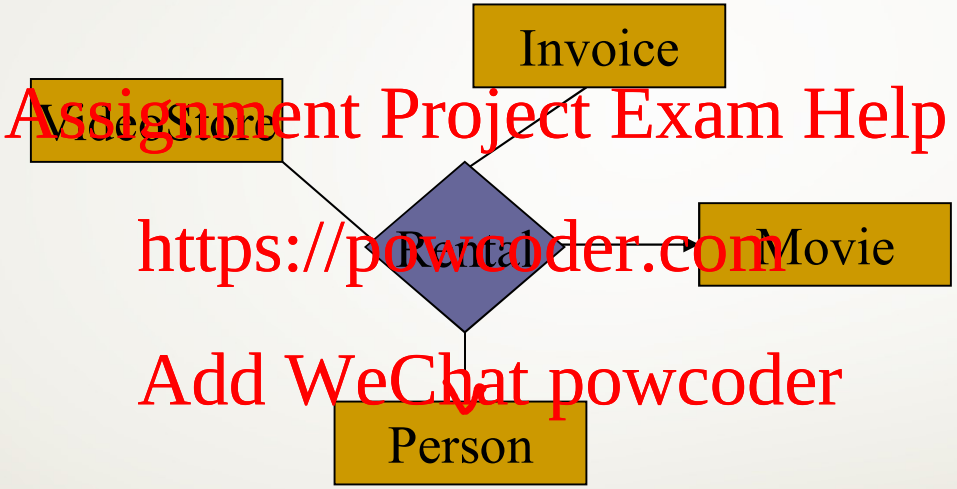
How do we model a purchase relationship between buyers, products and stores?



- Can still model as a mathematical set (how ?)
  - Yes: As a subset of product x store x person

Arrows in Multiway Relationships

Q: what does the arrow mean ?



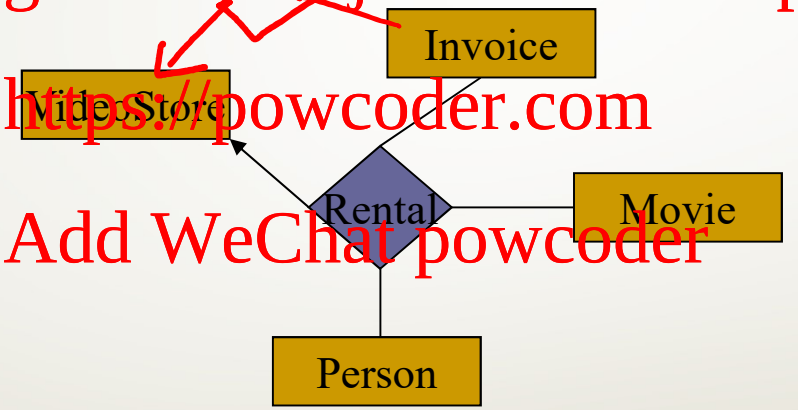
A: "At most one". That is, a specific combination of videostore, invoice and person can correspond to at most one movie.

Q: What if I had an arrow into Person?

Arrows in Multiway Relationships

Q: how do I say: "invoice determines store" ?

A: no good way; best approximation:



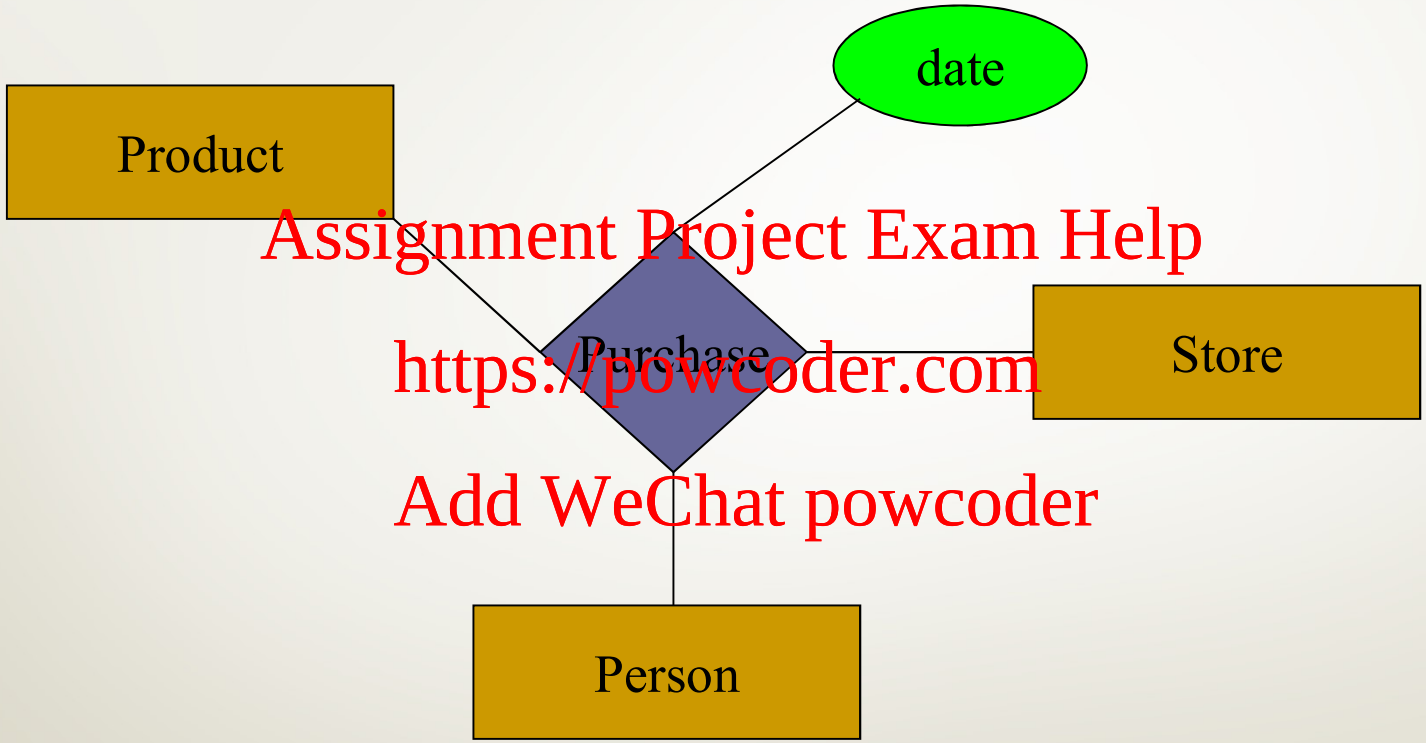
Q: Why is this bad ?

A: We aren't clarifying that the store is a function of the invoice only

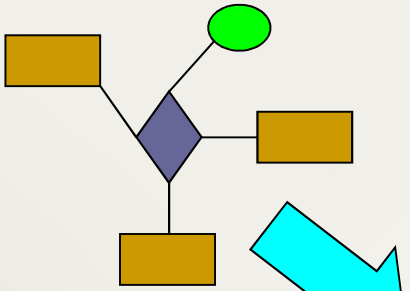
Some ER Modeling Tools Require 2-way Relationships

Do we need multi-way relationships or do 2-way (binary) relationships suffice?

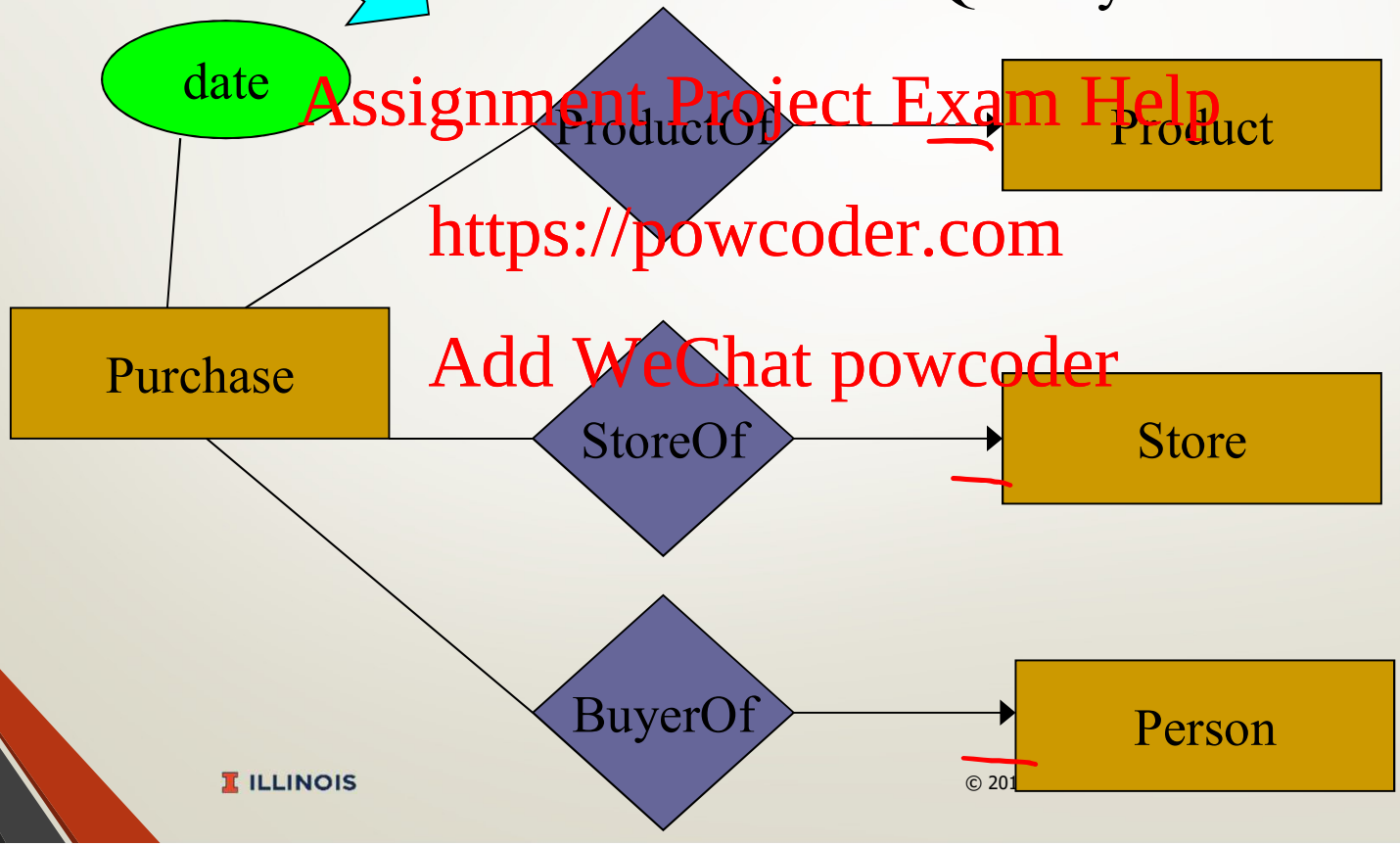
How would you convert this into binary?



Converting Multiway Relationships to Binary



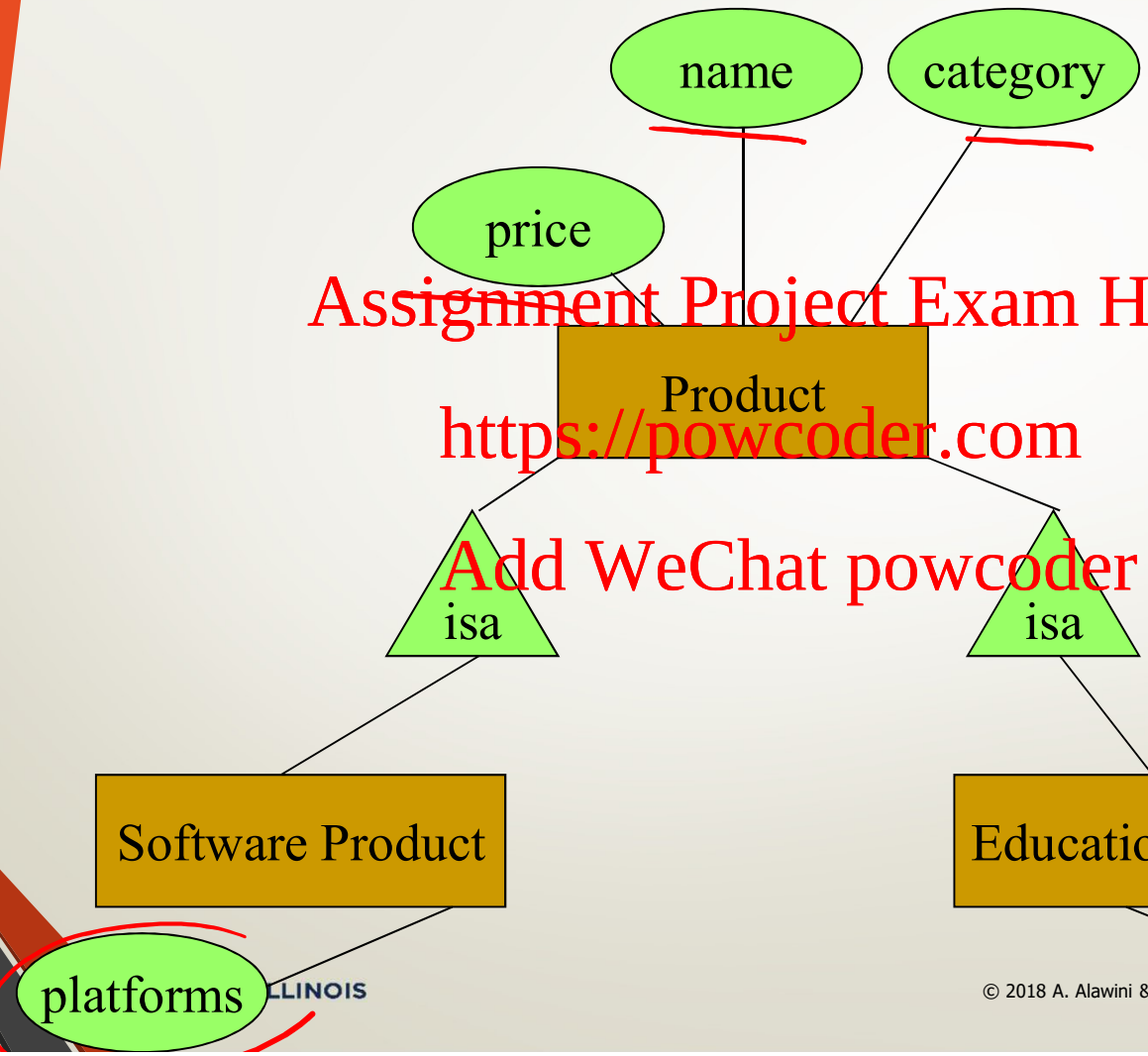
Q: Why the arrows?



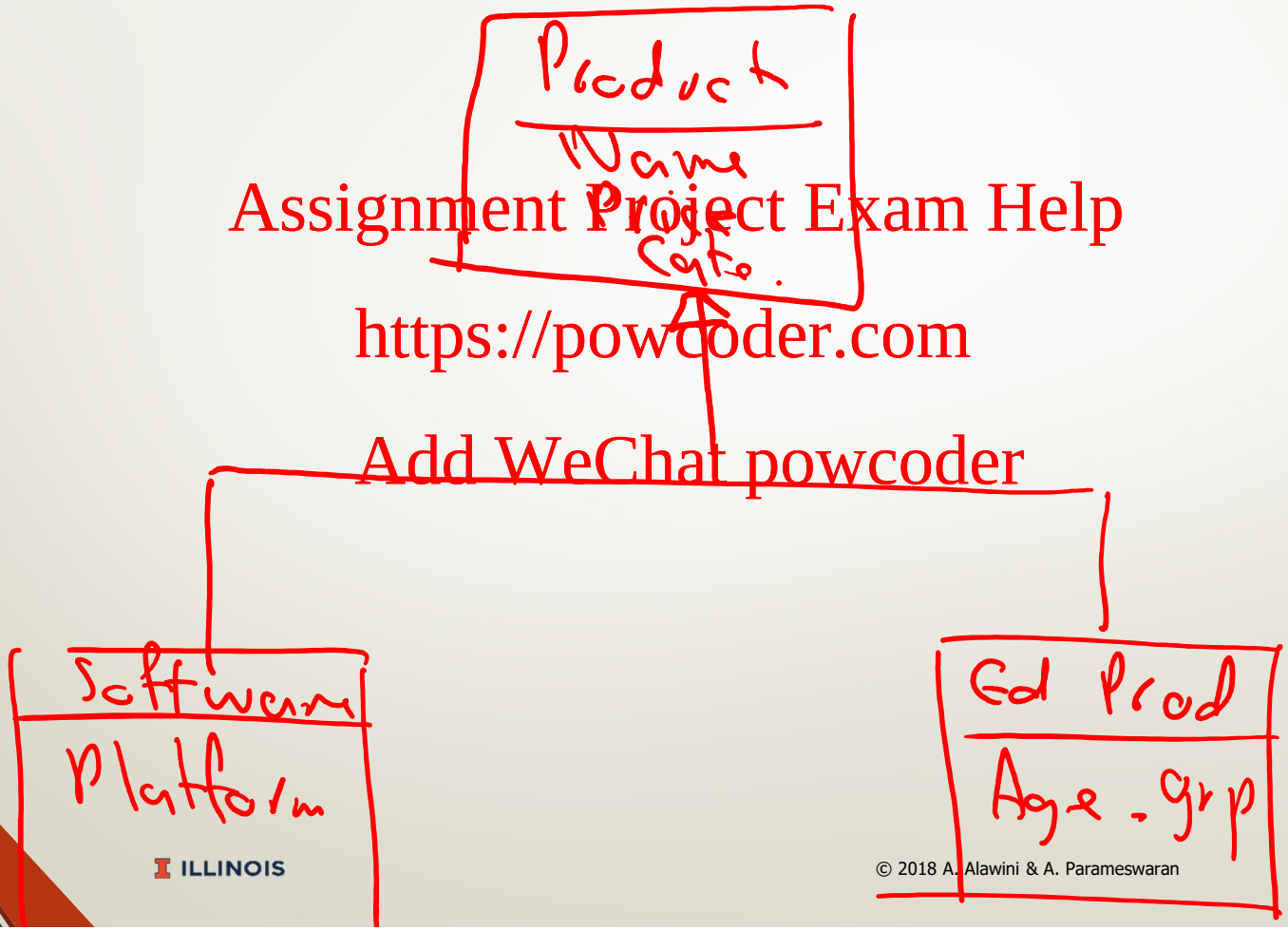
Relationships: Summary

- Modeled as a mathematical set
- Binary and multi-way relationships
- Converting a multi-way one into many binary ones
- Constraints on the degree of the relationship
  - many-one, one-one, many-many
  - limitations of arrows
- Attributes of relationships
  - not necessary, but useful

Subclasses in ER Diagrams



Subclasses in UML



Subclasses

- "Isa" triangles indicate the subclass relationship.
  - Point to the superclass.

- Subclasses form a tree.
  - I.e., no "multiple inheritance"

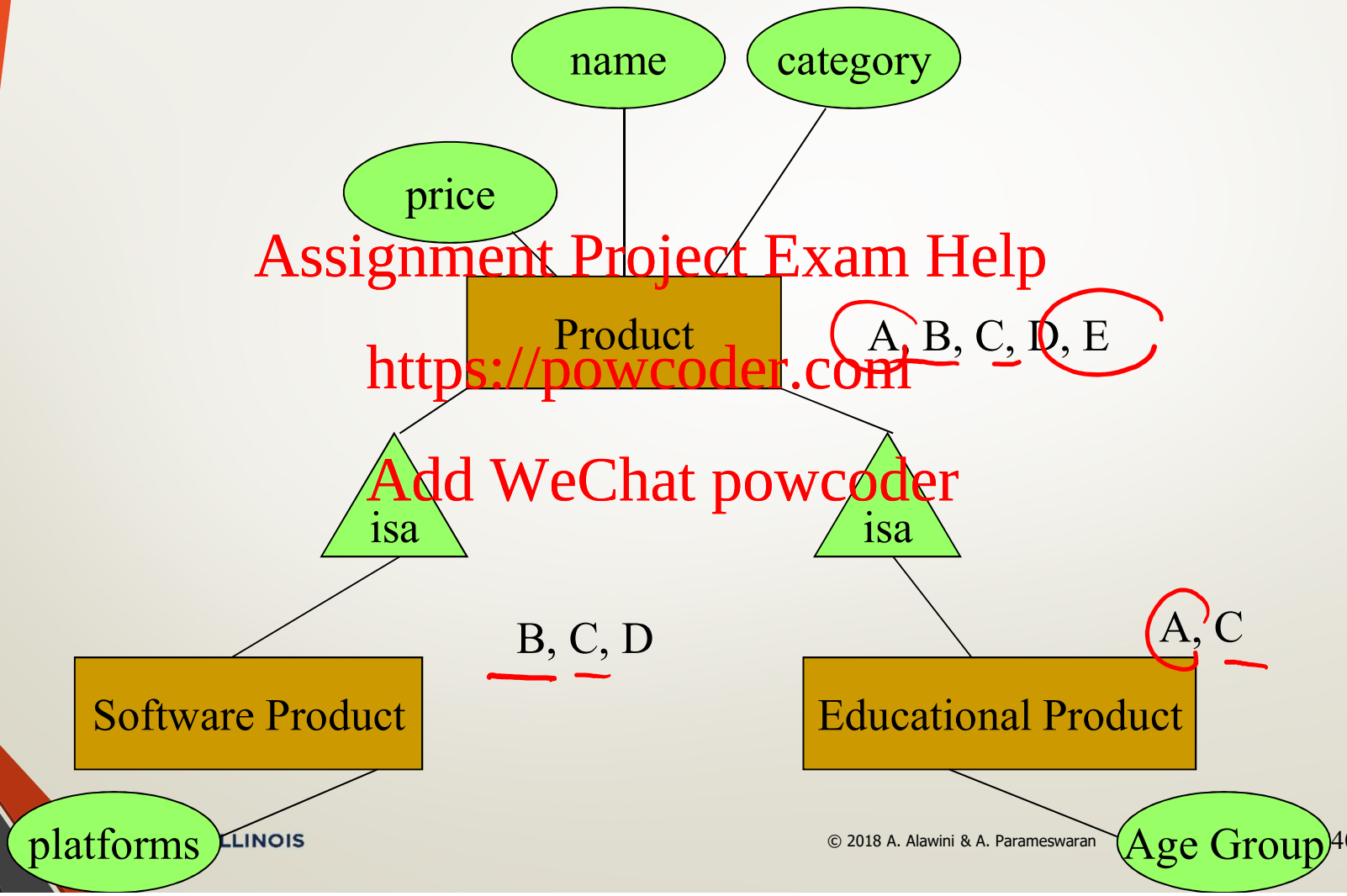
- Why subclasses?

  - Unnecessary to add redundant properties to the root entity set that don't apply to many of the entities

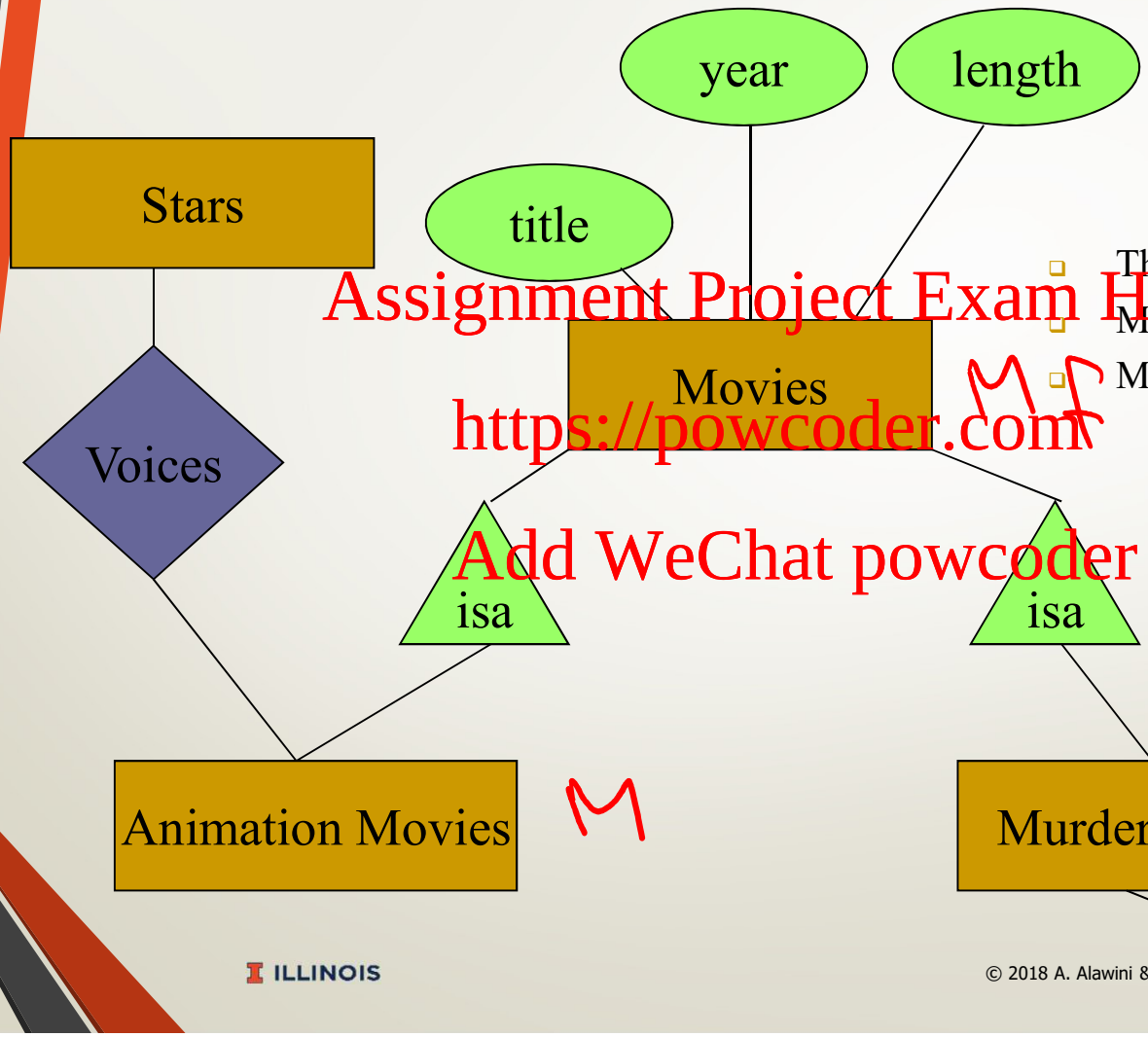
ER Vs. Object Oriented Subclasses

- 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.

Subclasses in ER Diagrams



Another Example



- The Minions?
- Mad Max: Fury Road?
- Murder on the Orient Express?

Agenda

- ✓ basics of ER and UML diagrams
- constraints
- weak entity sets

Constraints in ER diagram

- A constraint = an assertion about the database that must be true at all times
- Part of the database schema = structure
  (so it must be part of the ER diagram)
- Very important in database design

Modeling Constraints

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

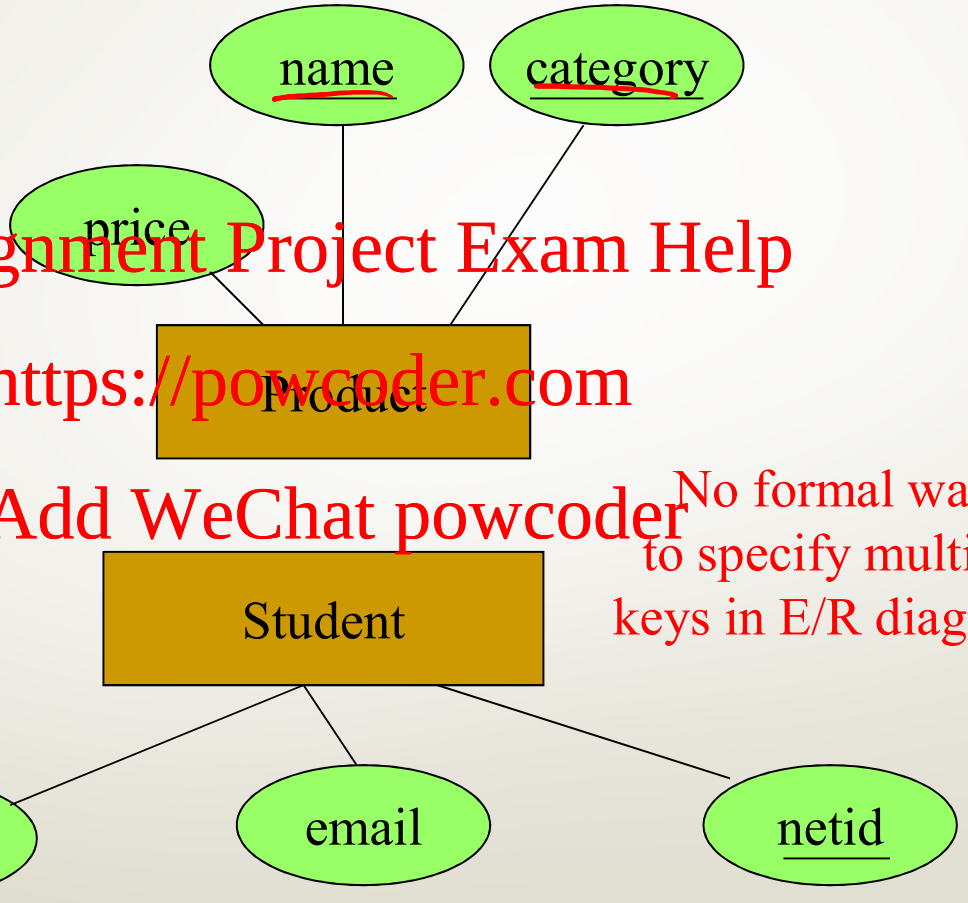
- Keys: attributes that identify entities in an entity set
  e.g., social security number uniquely identifies a person.

- Referential integrity constraints: relationship-based constraints
  e.g., 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)

Keys in E/R Diagrams

Underline:



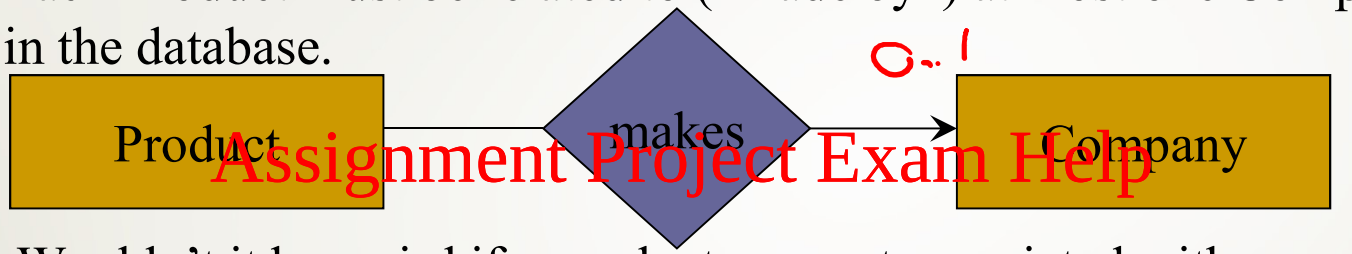
This means "name" and "category" together uniquely determine product

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

Referential Integrity Constraints

Recall: the arrow meant "at most one".

Each Product must be related to ("made by") at most one Company in the database.



Wouldn't it be weird if a product was not associated with any company?

This says "exactly one".

Each Product must be related to ("made by") exactly one Company in the database.

Arrow = at most 1

Semicircle = exactly 1

Referential Integrity Constraints

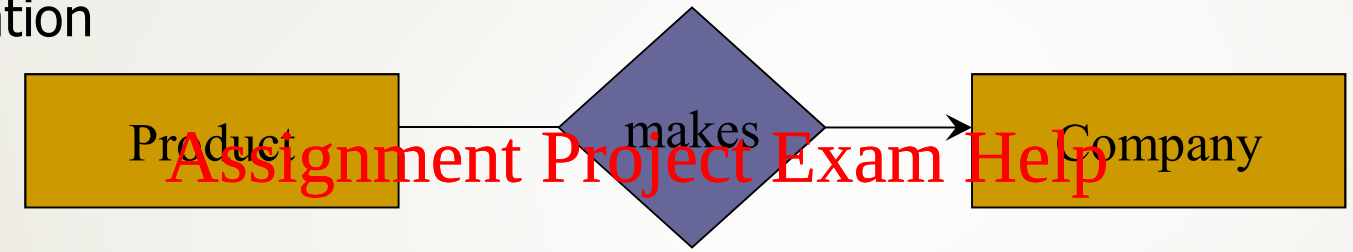
What do these two semi-circles mean?

Each movie is owned by precisely one studio, and
Each president runs exactly one studio

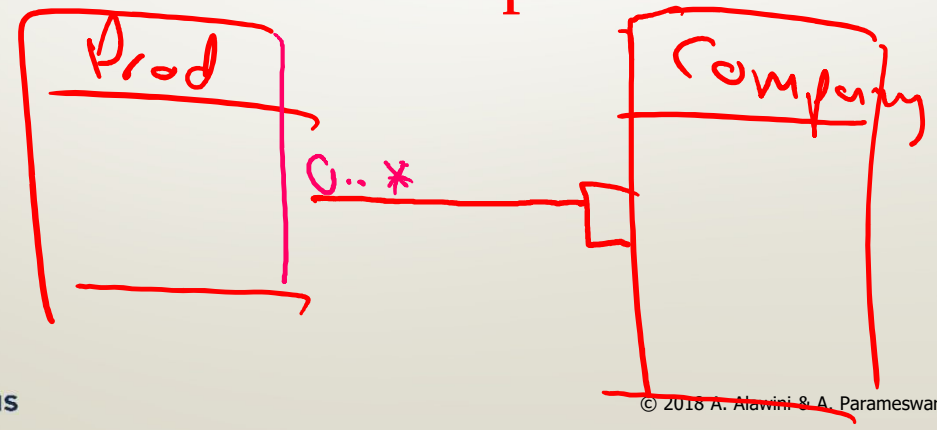
Each studio has up to one president

Referential Integrity Constraints
UML: Aggregation

ER Notation

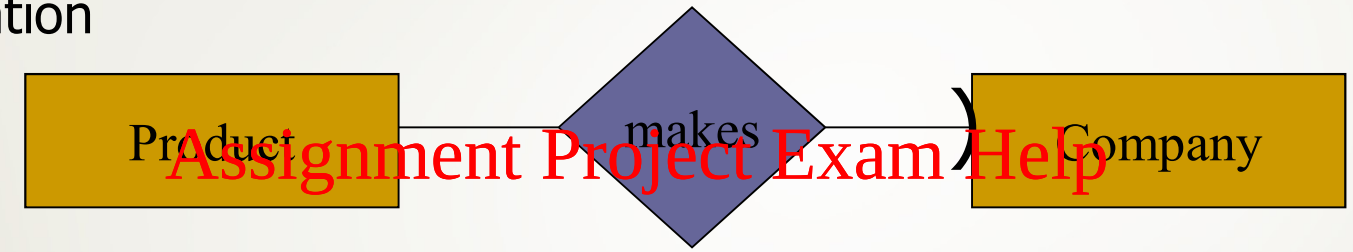


UML Notation

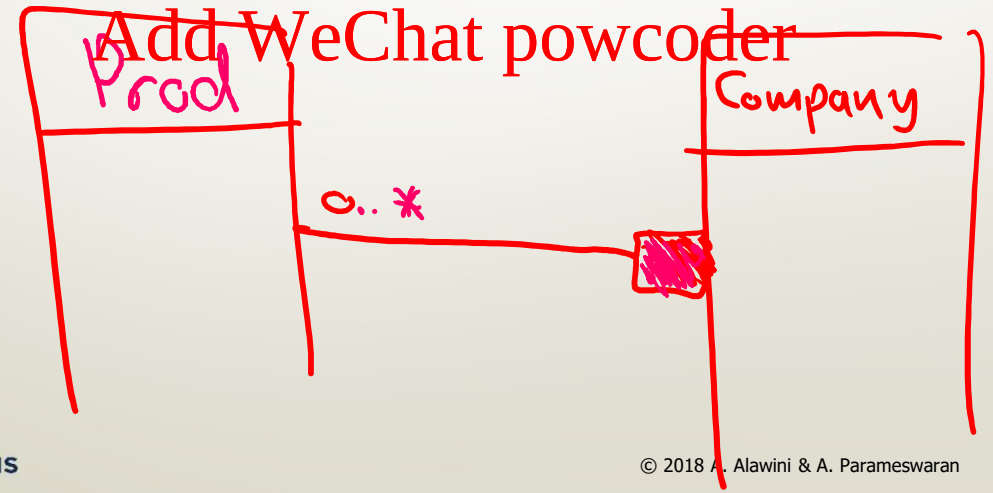


Referential Integrity Constraints
UML: Composition

ER Notation



UML Notation



Agenda

- ✓ basics of ER and UML Models
- ✓ constraints
- weak entity sets

Weak Entity Sets

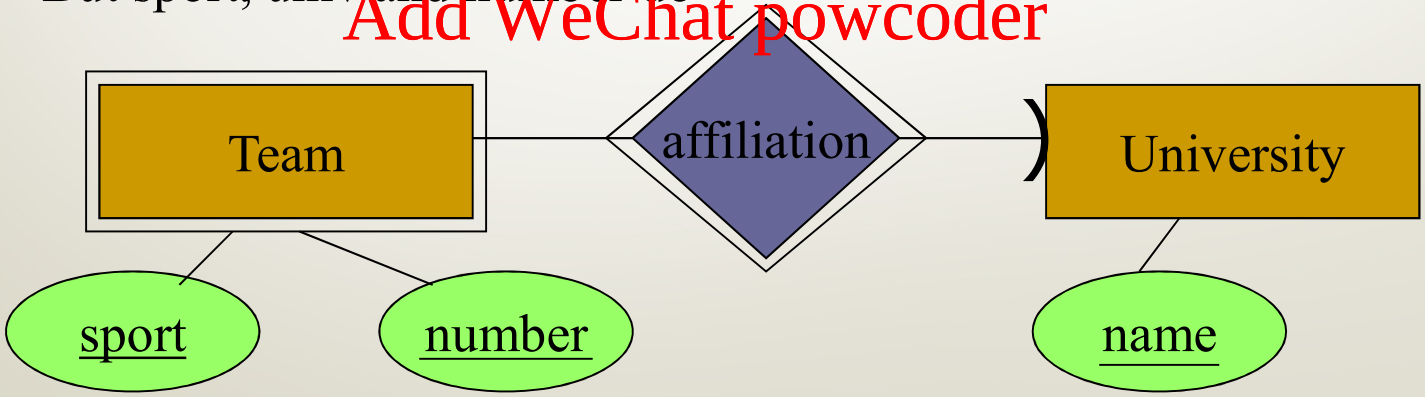
Entity sets are weak when some of their key attributes come from other classes to which they are related.

We'll see why this is important in a bit.

Sports and number don't uniquely determine the team

Football team 1 may be present multiple times

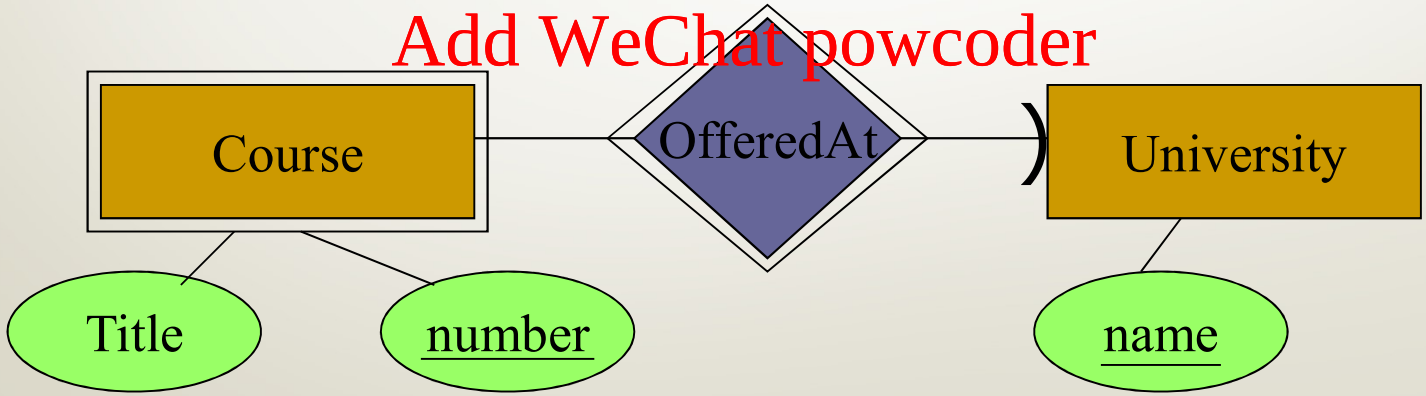
But sport, univ and number do



Weak Entity Sets

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

CS411, along with UIUC determine the rest of the attributes ("Database Systems"). Lots of CS411s otherwise!



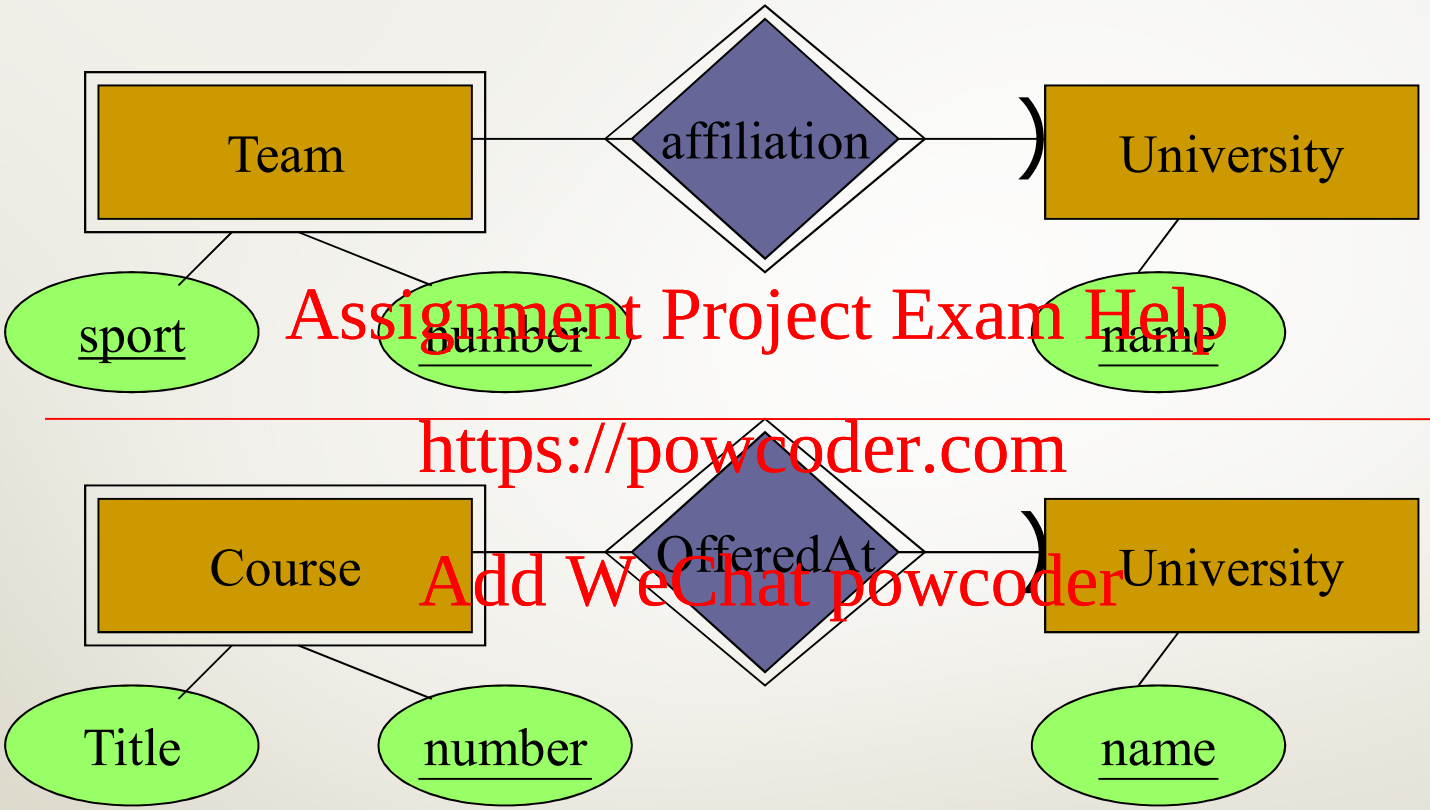
Weak Entity Sets

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

- Entity set E is *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 entity sets.

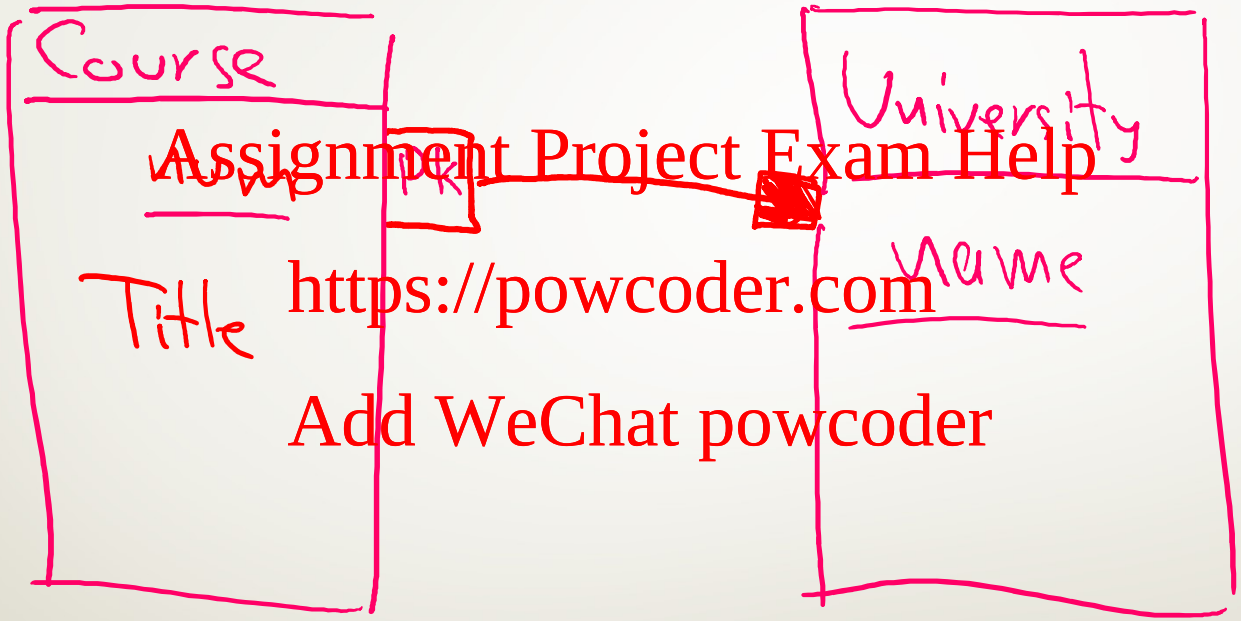
- Note: not an is-a relationship because E is not a "subclass" of F: Univ and Team

Notations for weak entity set

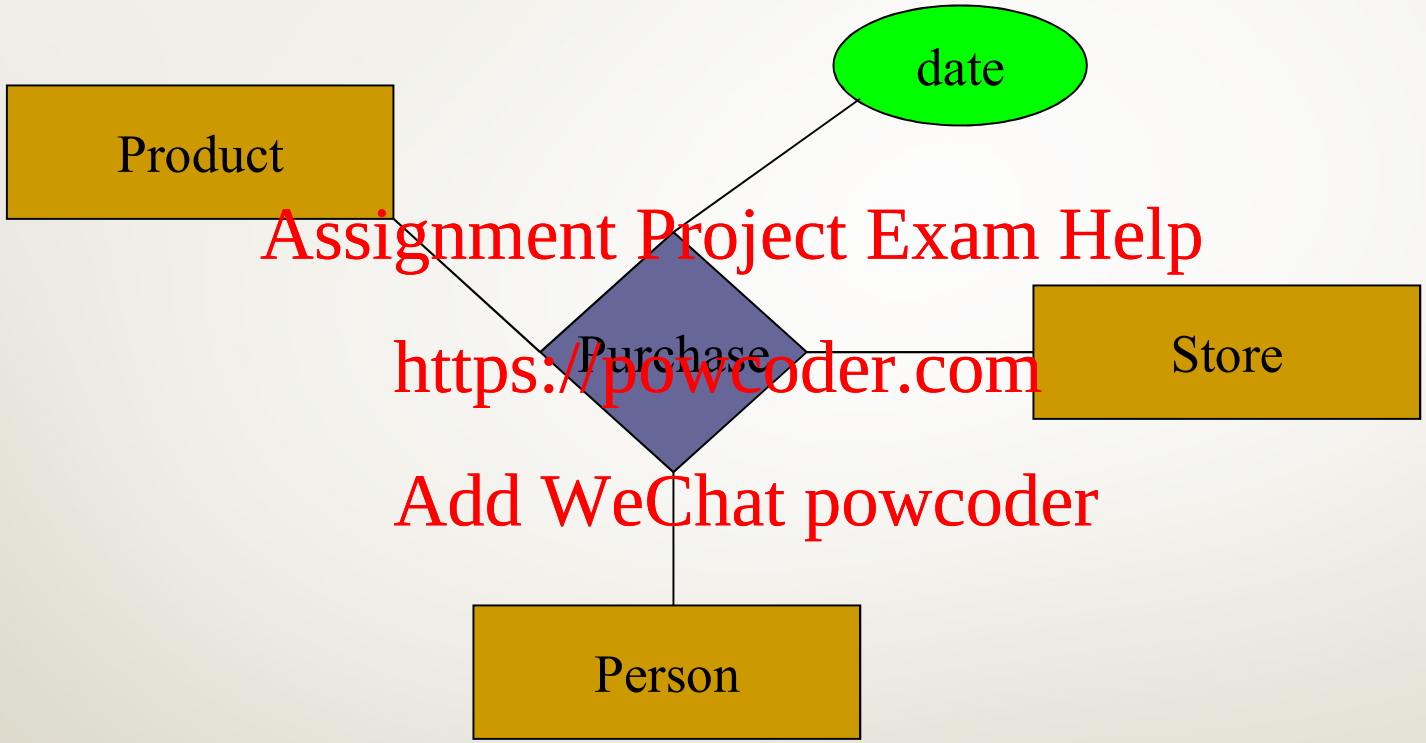


- "University" is a "supporting entity set" for "Team".
- "Affiliation" is a "supporting relationship".

Weak entity set in UML



Another scenario where weak e.s. arises



A Multi-way relationship …

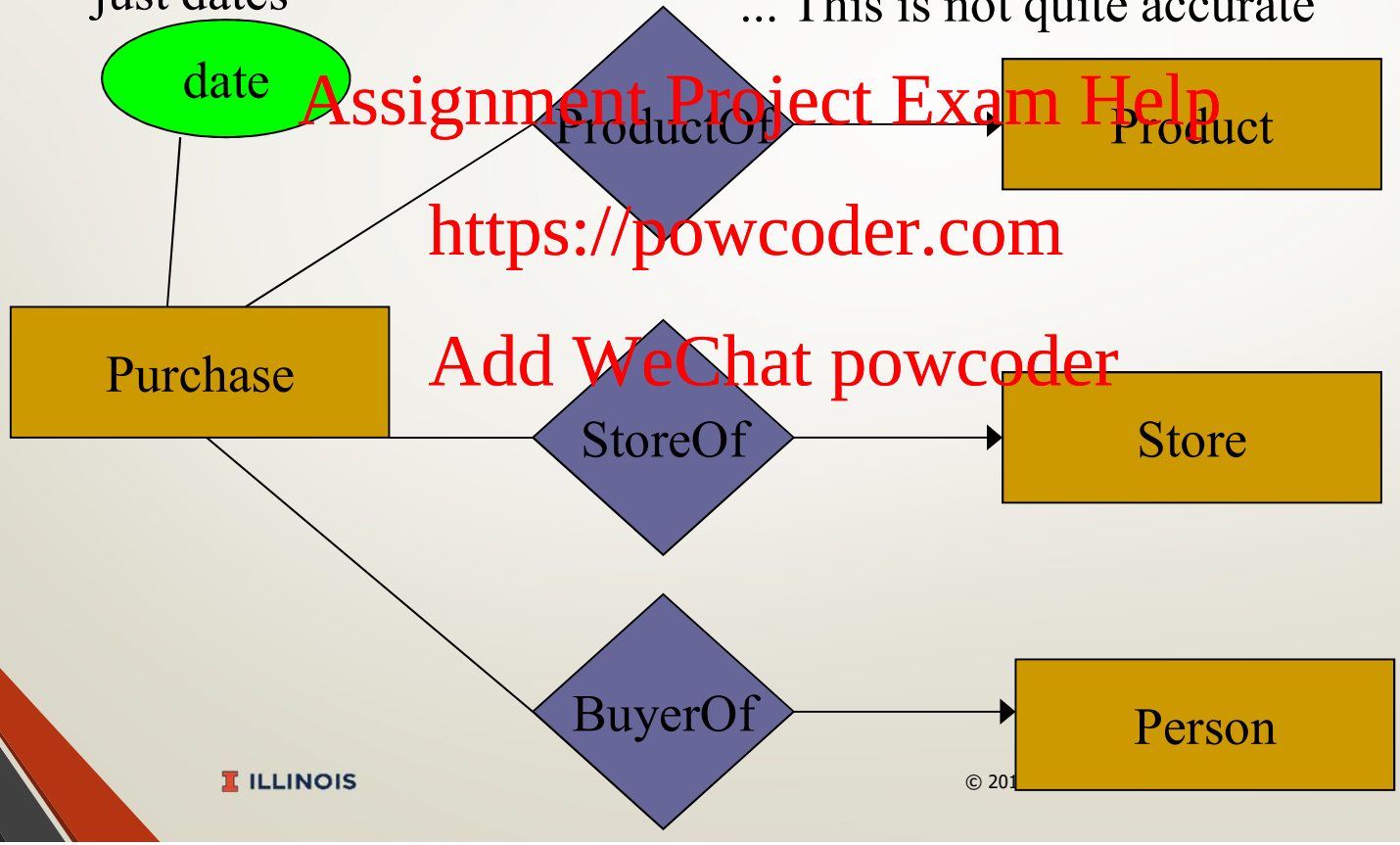
Another scenario where weak e.s. arises

Not sufficient to identify purchase via just dates

… converted to binary relationships

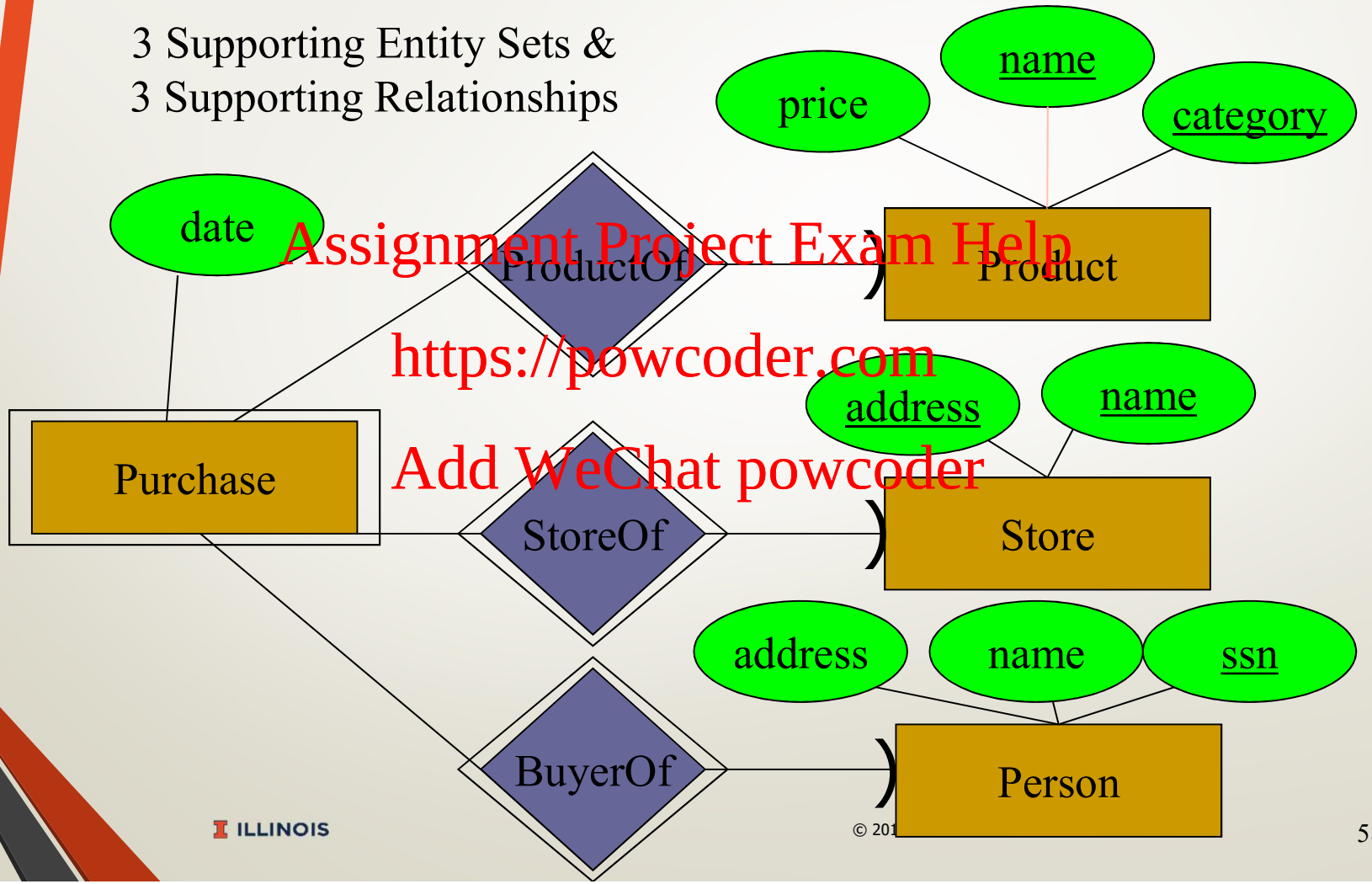
Remember this is what we had …

... This is not quite accurate



Another scenario where weak e.s. arises

3 Supporting Entity Sets &
3 Supporting Relationships



Agenda

- ✓ basics of ER and UML diagrams
- ✓ constraints
- ✓ weak entity sets

ER Review

- Basics of ER and UML
  - 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