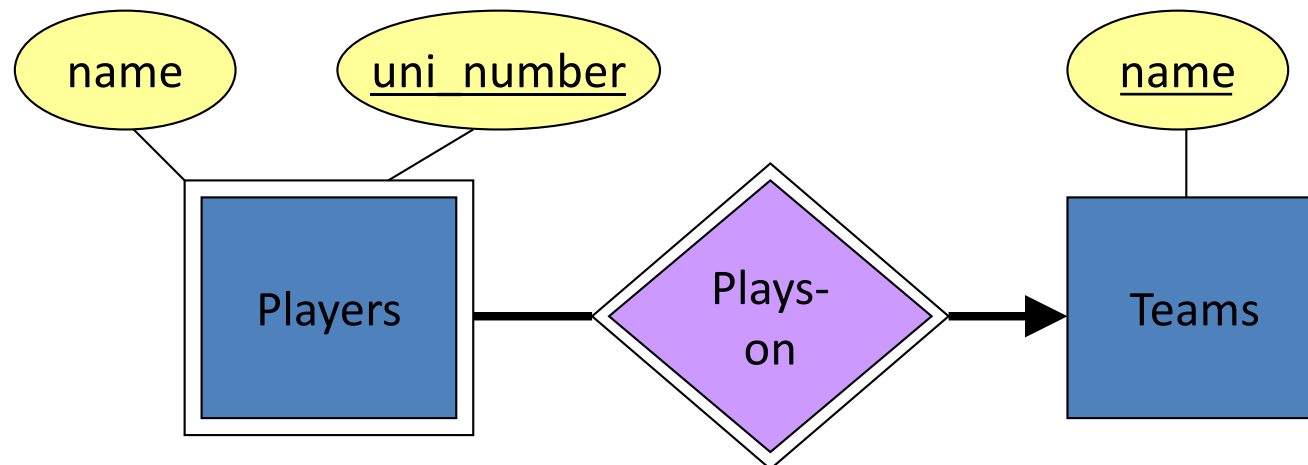


Weak entity sets (4.4)

- A weak entity set is an entity set whose (primary) key contains attributes from one or more other entity sets.
- In other words, an 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 in E 's key.
- Possible that all attributes in a weak entity set's key come from other entity sets.

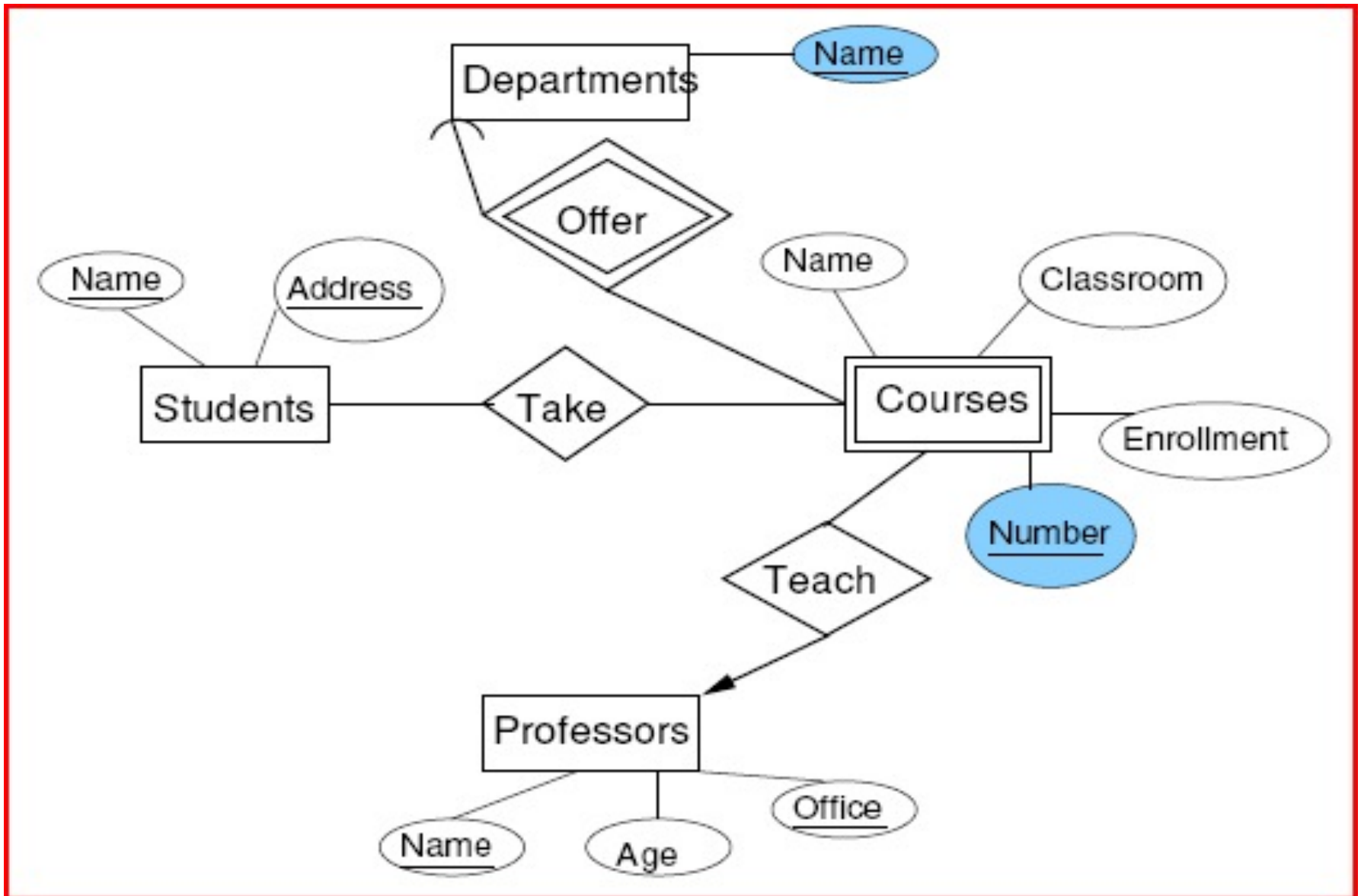
Example

- Consider players in a sports league:
 - Name is not a key (might be duplicate names)
 - Uniform number is certainly not a key (numbers will be duplicated across teams)
 - But number + team should be a key



- Use double border for weak entity sets and their supporting many-one relationships.

How about courses and departments?



Keys for a weak entity set

- A relationship R from a weak entity set E to F is *supporting* if
 - R is a binary, many-one relationship from E to F.
 - R has referential integrity from E to F (curved arrow into F).
- F supplies its key attributes to define E's key.
- If F itself is a weak entity set, then we must find F's supporting relationships and also use the keys from those supporting entity sets.

Where do weak entity sets come from?

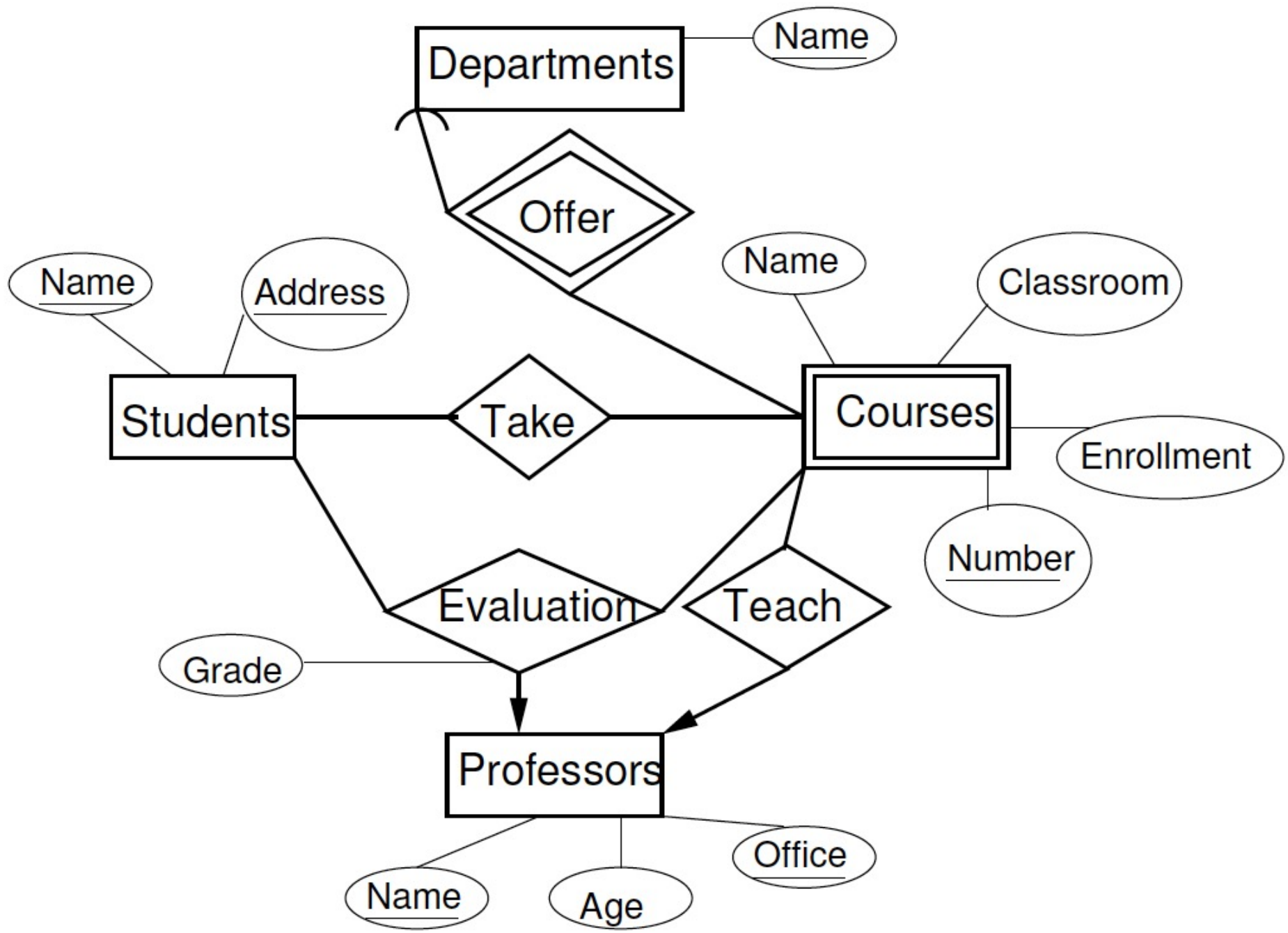
- Cause 1: Implicit hierarchies not from an "is-a" relationship.
 - A player “belongs to” a team, or a flight “is flown by” an airline.
 - Happens when a piece of a key is represented as an entity set rather than an attribute.
 - Can (technically) be solved by putting a unique ID on an entity set, but sometimes this causes more trouble than it’s worth.
 - "is-a" hierarchies seem to lead to weak entity sets (subclasses), but we don't notate them with double borders because their hierarchical relationships are always one-one.

Where do weak entity sets come from?

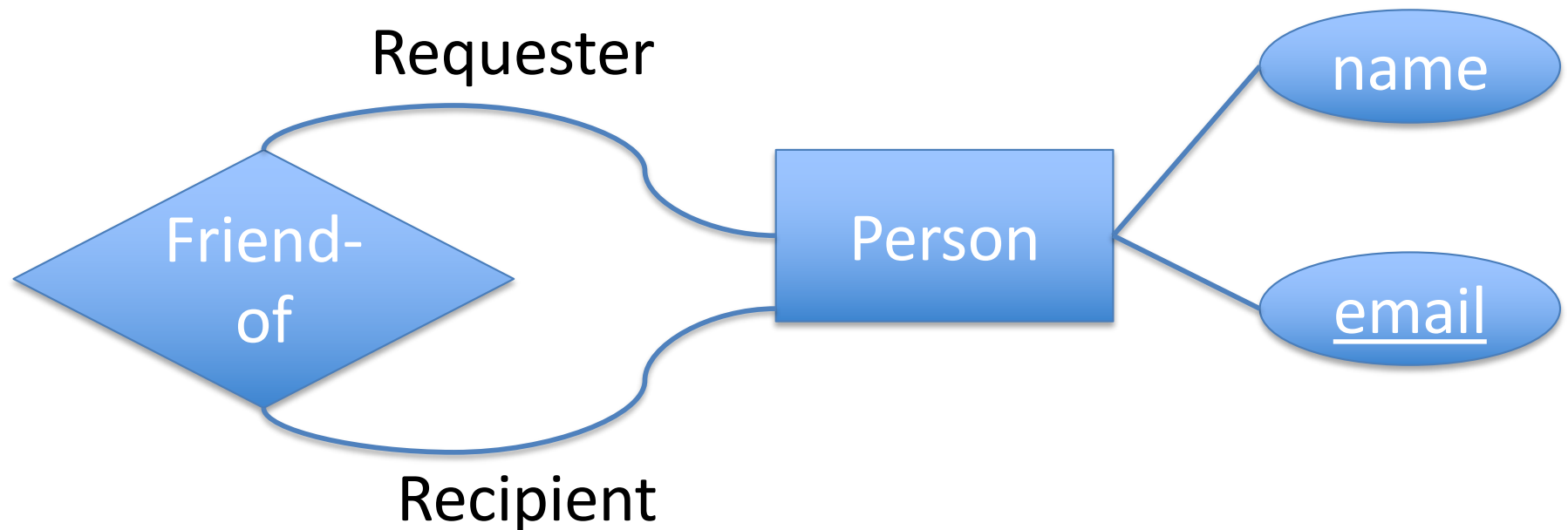
- Cause 2: Connecting entity sets created by eliminating a multi-way relationship.
 - Often, connecting entity sets have no attributes of their own; they must pick up their key attributes from the entity sets they connect.
 - Example: A CUSTOMER rents a CAR from a SALESPERSON.

Converting E/R diagrams to relational designs

- Entity set -> Relation
 - Attribute of entity set -> attribute of relation
 - Key of entity set -> primary key of relation
- Relationship -> Relation
 - Attribute of relationship -> attribute of relation
 - Key attribute of connecting entity set -> key attribute of relation
- Special cases: weak entity sets, "is-a" hierarchies, combining relations.



Handling multiple roles

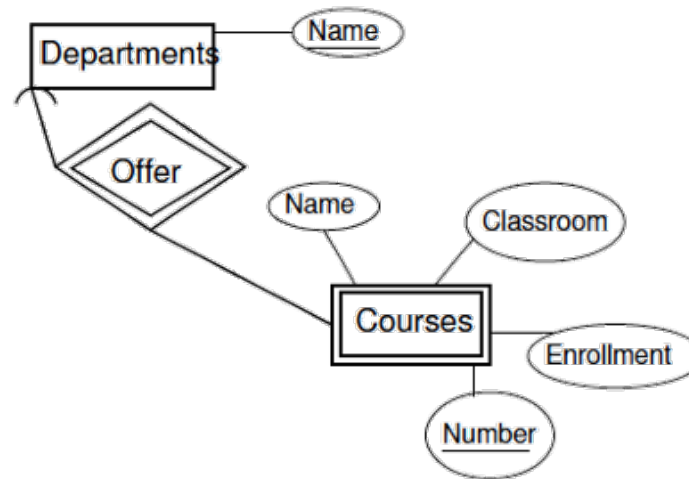


If an entity set E appears $k > 1$ times in a relationship R , then the key attributes for E appear k times in the relation for R , appropriately renamed.

Handling weak entity sets

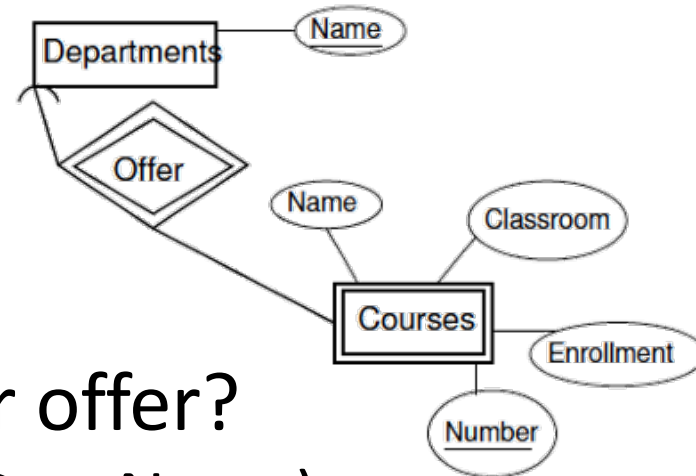
- For each weak entity set W , create a relation with attributes:
 - attributes of W
 - attributes of supporting relationships for W
 - *key* attributes of supporting entity sets for W

Supporting Relationships



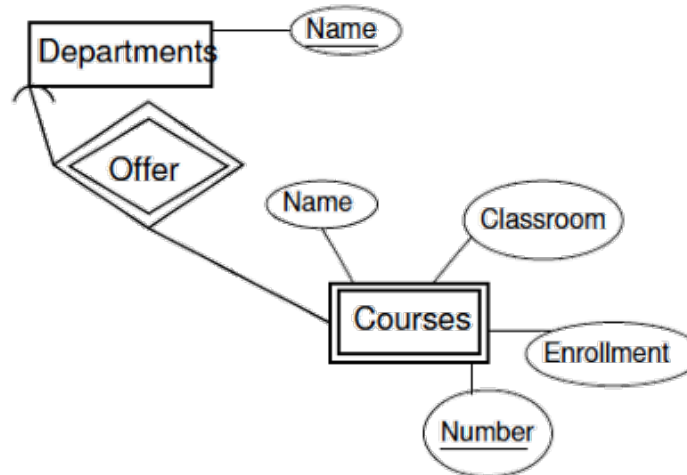
- Schema for Departments is Departments(Name)
- Schema for Courses is Courses(Number, DeptName, CourseName, Classroom, Enrollment)
- What is the schema for Offer?

Supporting Relationships



- What is the schema for offer?
 - Offer(Name, Number, DeptName)
 - But Name and DeptName are identical, so the schema for Offer is Offer(Number, DeptName)
 - The schema for Offer is a subset of the schema for the weak entity set, so ***we can dispense with the relation for Offer.***
 - ***Key point: Don't make a relation for supporting relationships.***

Summary of Weak Entity Sets



- If W is a weak entity set, the relation for W has a schema whose attributes are
 - all attributes of W
 - all attributes of supporting relationships for W
 - for each supporting relationship for W to an entity set E
 - the key attributes of E
- There is no relation for any supporting relationship for W

Combining Relations

- Consider many-one Teach relationship from Courses to Professors
- Schemas are:
 - Courses(Number, DepartmentName, CourseName, Classroom, Enrollment)
 - Professors(Name, Office, Age)
 - Teach(Number, DepartmentName, ProfessorName, Office)

Combining Relations

Courses(Number, DepartmentName, CourseName, Classroom, Enrollment)

Professors(Name, Office, Age)

Teach(Number, DepartmentName, ProfessorName, Office)

- The key for Courses uniquely determines all attributes of Teach
- We can combine the relations for Courses and Teach into a single relation whose attributes are
 - All the attributes for Courses,
 - Any attributes of Teach, and
 - The key attributes of Professors

Rules for Combining Relations

- We can combine into one relation Q
 - The relation for an entity set E
 - all many-to-one relationships R_1, R_2, \dots, R_k from E to other entity sets E_1, E_2, \dots, E_k respectively
- The attributes of Q are
 - All the attributes of E
 - Any attributes of R_1, R_2, \dots, R_k
 - The key attributes of E_1, E_2, \dots, E_k
- Combining a **many-many** relationship with one of its entity sets often leads to redundancy. You probably never want to do this!