E/R to relations: summary

Redundancy? No, just multi-attribute keys

Relationship **Stars-In** between entity sets **Movies** and **Stars** is represented by a relation with schema:

Stars-In(title, year, starName)

A sample instance is:

title	*year*	*starName*
Star Wars	1977	Carrie Fisher
Star Wars	1977	Mark Hamill
Star Wars	1977	Harrison Ford
Mighty Ducks	1991	Emilio Estevez
Wayne’s World	1992	Dana Carvey
Wayne’s World	1992	Mike Meyers

Redundancy? Yes

Relationship **Stars-In** between entity sets **Movies** and **Stars** is represented by a relation with schema:

Stars-In(<u>title</u>, <u>year</u>, duration, <u>starName</u>)

A sample instance is:

title	*year*	*duration*	*starName*
Star Wars	1977	120	Carrie Fisher
Star Wars	1977	120	Mark Hamill
Star Wars	1977	120	Harrison Ford
Mighty Ducks	1991	130	Emilio Estevez
Wayne's World	1992	90	Dana Carvey
Wayne's World	1992	90	Mike Meyers

Surrogate keys – only if we can uniquely identify any entity

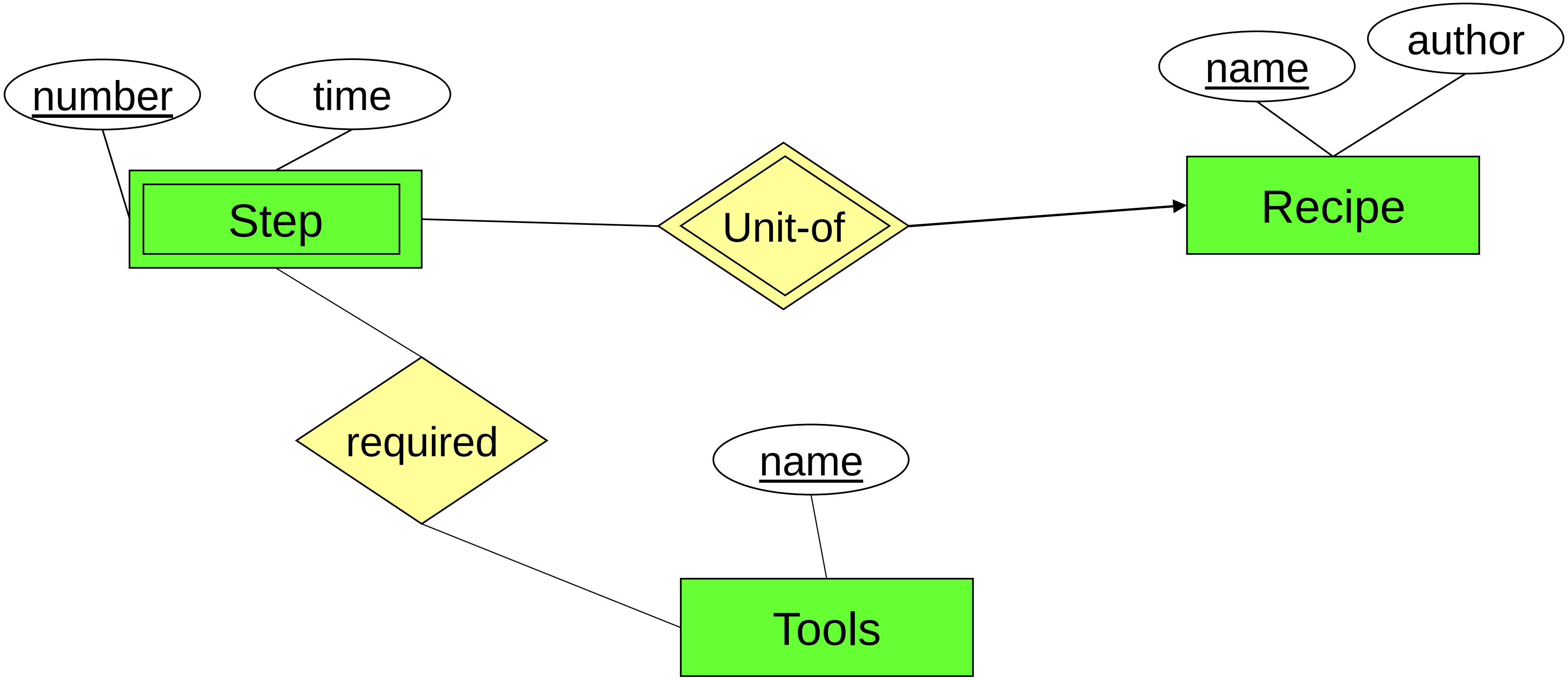
- If we want to combine data from IMDB, MovieLens, Netflix – can only identify movies by name, year
- No globally accepted movie identifier exists
- Movies, video games vs. books (International Standard Book Number)

Week entity sets

- It is possible that the key of an entity set is composed of attributes, some or all of which do not belong to this entity set
- Such an entity set is called a ***week entity set***

- We use week entity sets to identify sub-units of the main entity, rather than sub-classes

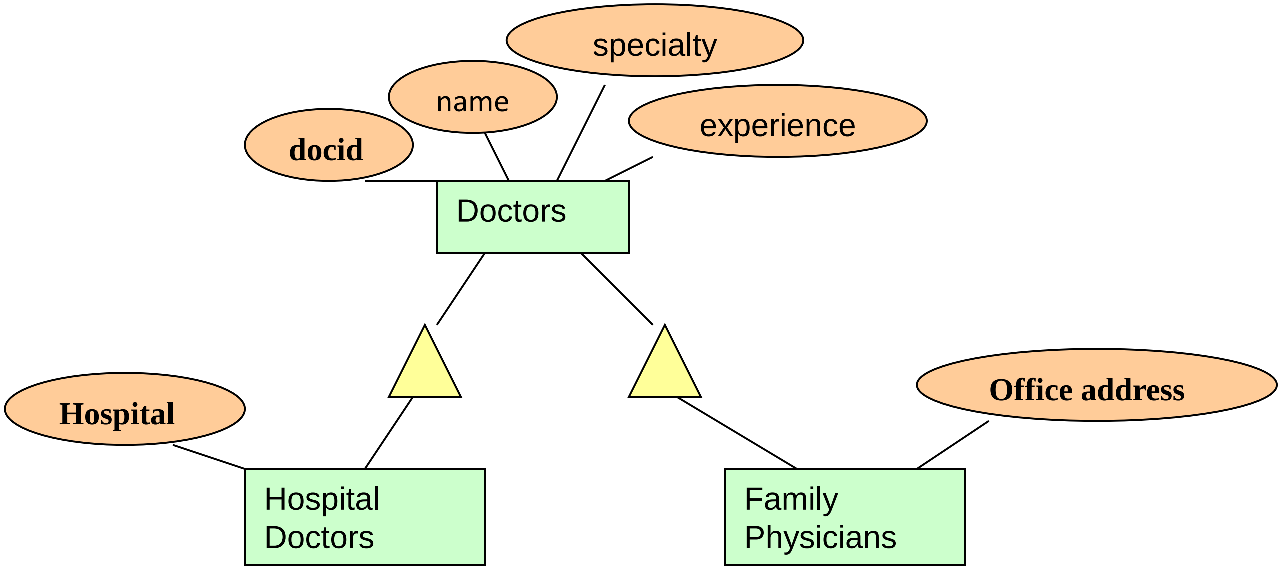
Sub-units example



Sub-classes in E/R

- Sometimes, an entity set contains certain entities that have special properties not associated with all members of this entity set.
- In this case it is useful to define special-case entity sets, or *subclasses*, each with its own attributes and relationships

Sub-classes example

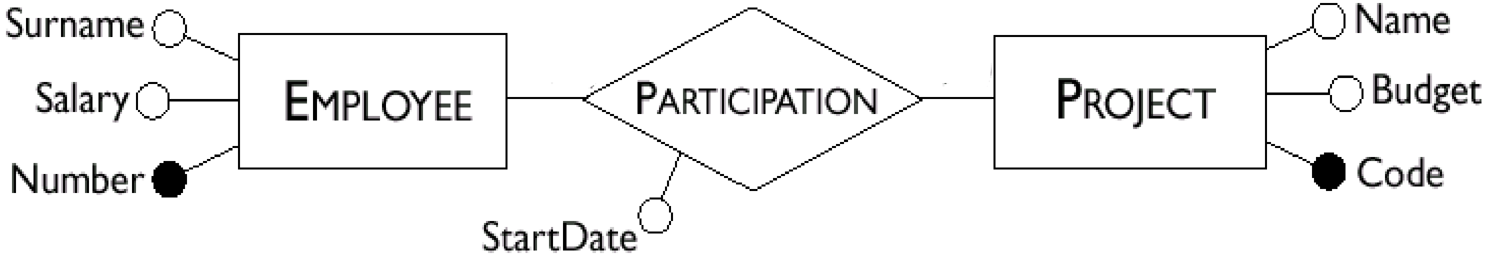


General rules:
E/R to relations

From E/R to relational schema

- Each entity set becomes a relation.
  Its attributes are
  - the attributes of the entity set.
- Each relationship becomes a relation.
  It’s attributes are
  - the keys of the entity sets that it connects, plus
  - the attributes of the relationship itself.

Many-to-Many Binary Relationships

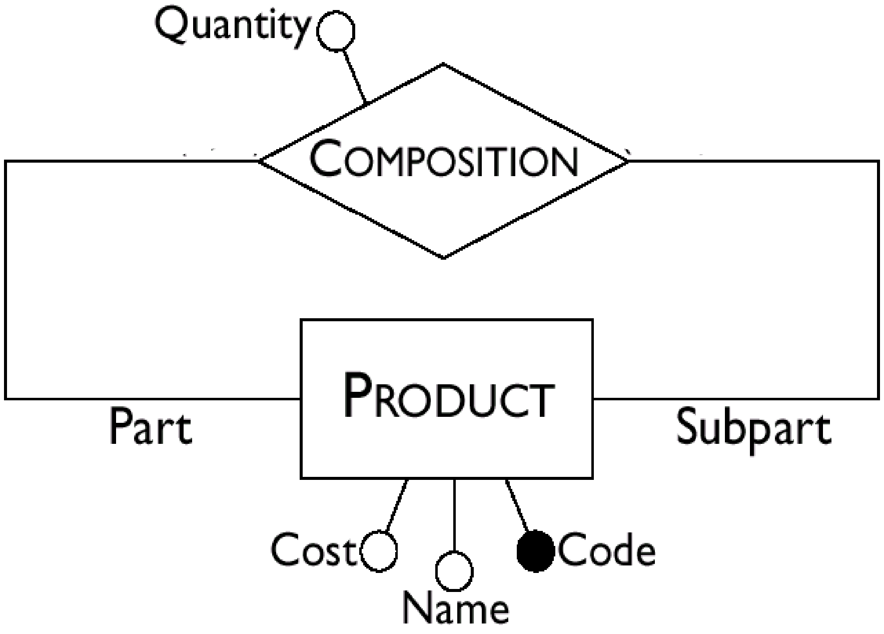


Employee(Number, Surname, Salary)

Project(Code, Name, Budget)

Participation(Number, Code, StartDate)

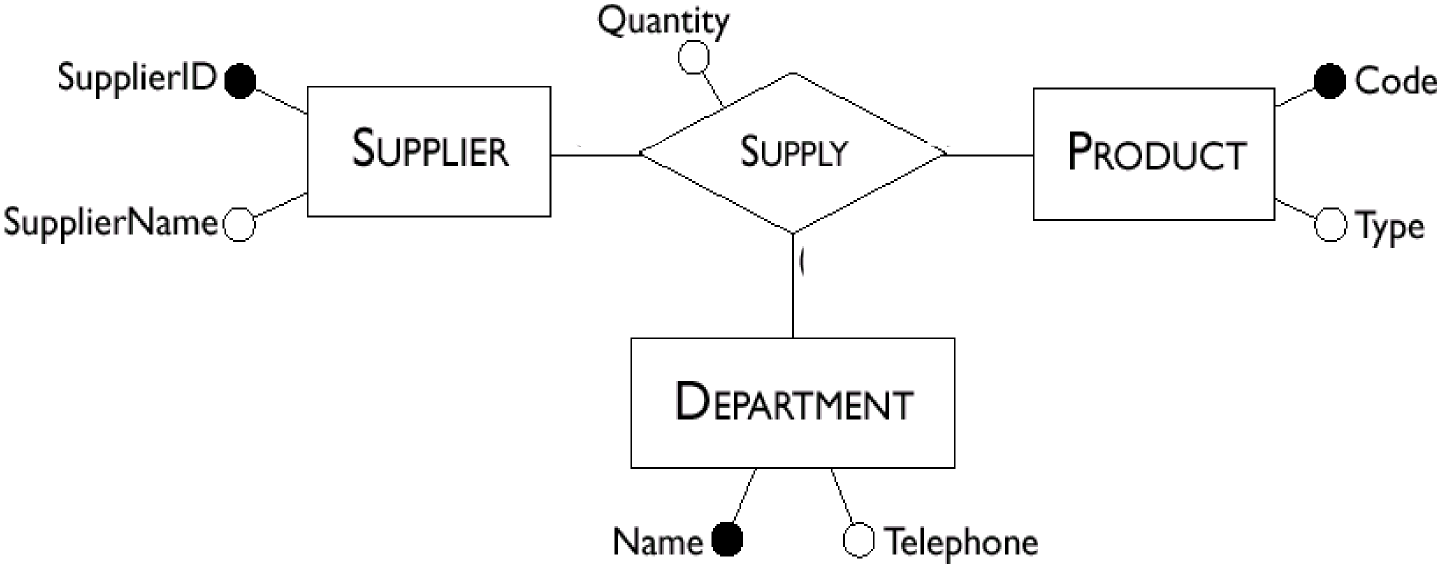
Many-to-Many Self-Relationships



Product(<u>Code</u>, Name, Cost)

Composition(<u>Part</u>, <u>SubPart</u>, Quantity)

Many-to-Many Ternary Relationships



Supplier(<u>SupplierID</u>, SupplierName)

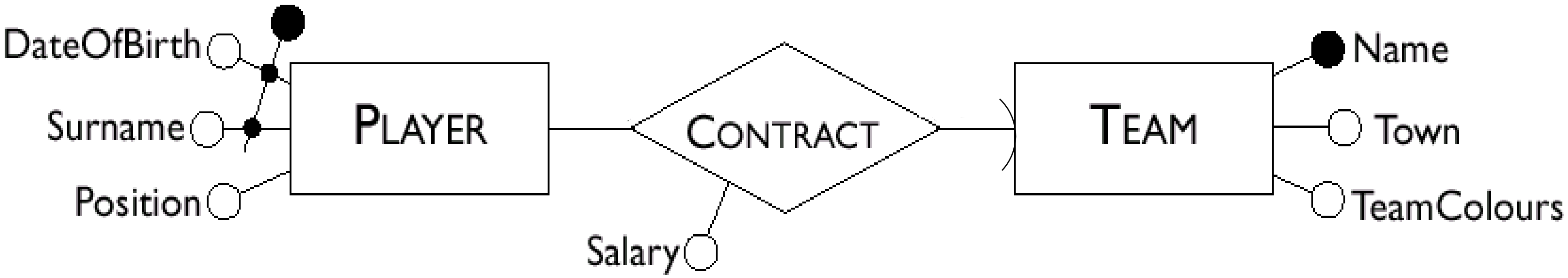
Product(<u>Code</u>, Type)

Department(<u>Name</u>, Telephone)

Supply(<u>Supplier</u>, <u>Product</u>, <u>Department</u>, Quantity)

One-to-Many Relationships

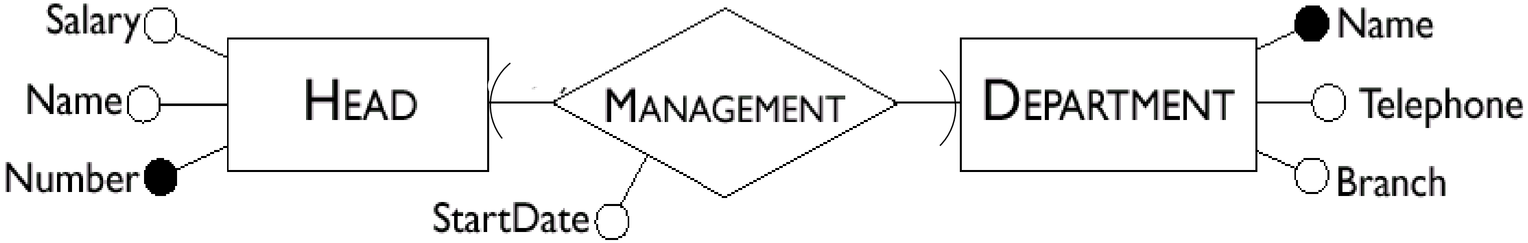
with mandatory participation for one



Player(Surname,DateOfBirth, Position)
Team(Name, Town, TeamColours)
Contract(PlayerSurname, PlayerDateOfBirth, Team, Salary)

BETTER:
Player(Surname,DateOfBirth, Position, TeamName, Salary)
Team(Name, Town, TeamColours)

One-to-One Relationships
with mandatory participation for both



Head(<u>Number</u>, Name, Salary, Department, StartDate)

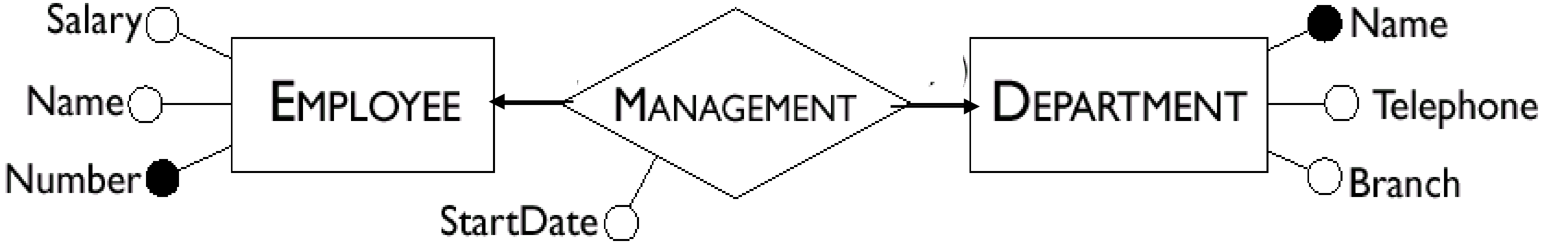
Department(<u>Name</u>, Telephone, Branch)

OR

Head(<u>Number</u>, Name, Salary, StartDate)

Department(<u>Name</u>, Telephone, HeadNumber, Branch)

One-to-One Relationships
with optional participation for one

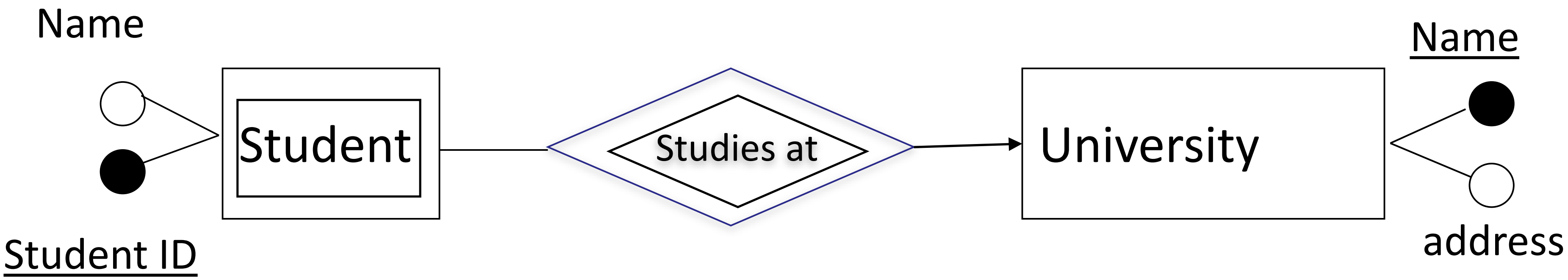


Employee(Number, Name, Salary)
Department(Name, Telephone, Branch, Head, StartDate)

Or, if both entities are optional

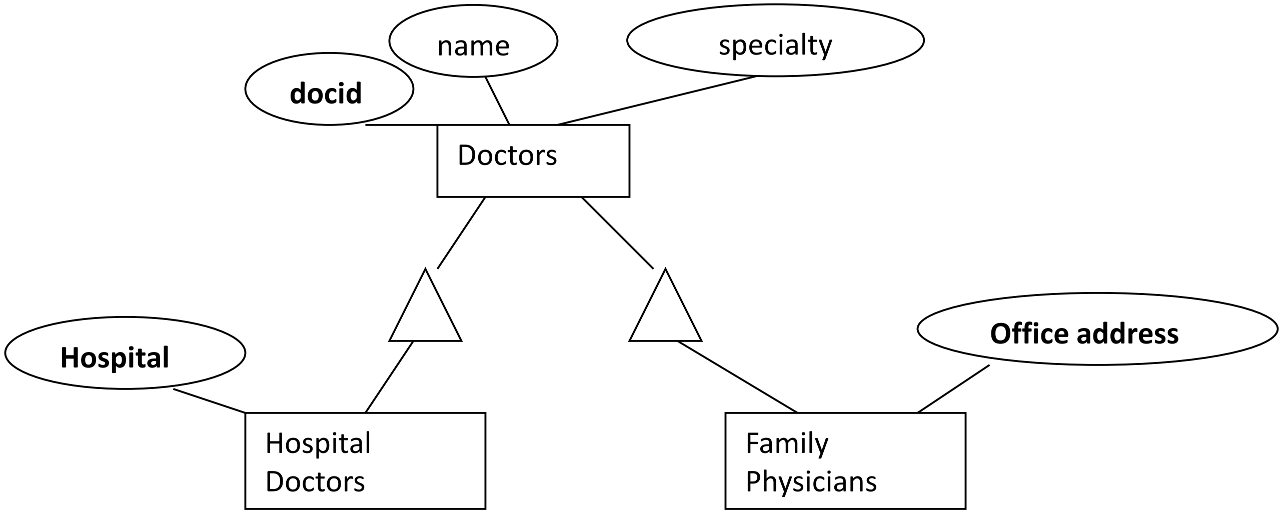
Employee(Number, Name, Salary)
Department(Name, Telephone, Branch)
Management(HeadName, Department, StartDate)

Week entity sets



Student (Student ID, University name, Student Name)
University (Name, address)

Sub-classes: OO approach



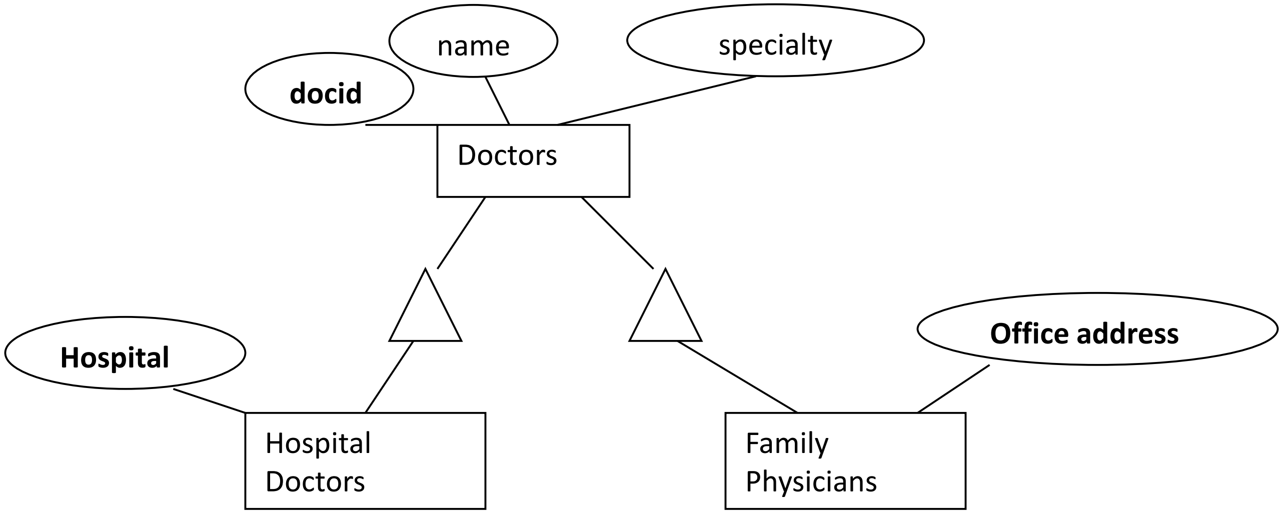
Doctors (<u>docid</u>, name, specialty)

HospitalDoctors (<u>docid</u>, name, specialty, hospital)

FamilyDoctors (<u>docid</u>, name, specialty, address)

HospitalFamilyDoctors (<u>docid</u>, name, specialty, hospital, address)

Sub-classes: E/R approach

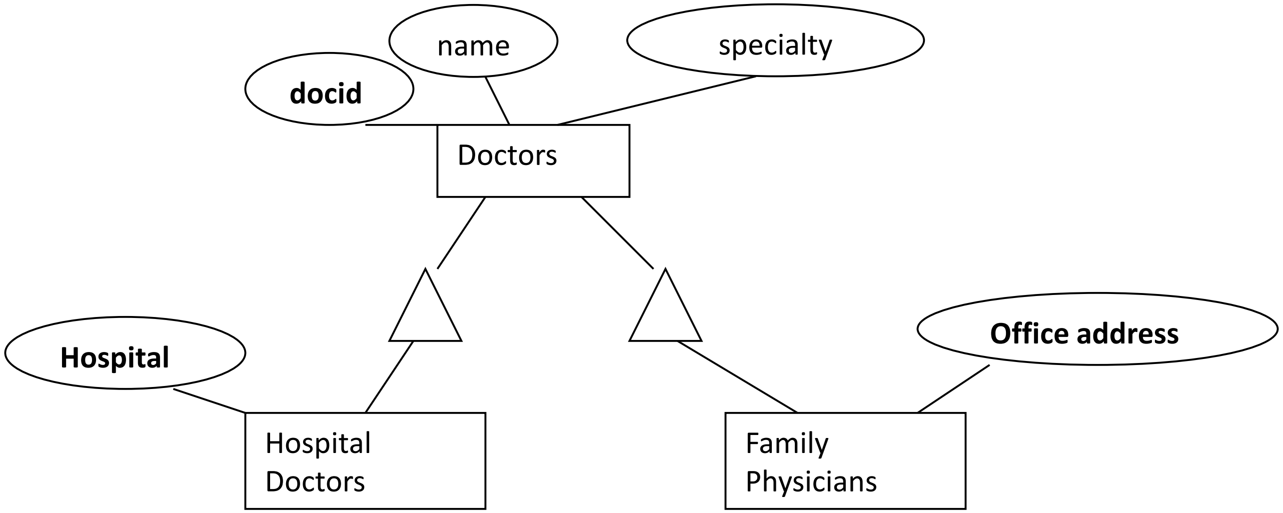


Doctors (docid, name, specialty)

HospitalDoctors (docid, hospital)

FamilyDoctors (docid, address)

Sub-classes: NULL approach



Doctors (<u>docid</u>, name, specialty, hospital, address)