



Week4_Course_part1

ER Model-Basic stuff

⟨#⟩

Database System - Nankai



ER Model

- Gives us a language to specify
 - what information the db must hold
 - what are the relationships among components of that information
- Proposed by Peter Chen in 1976
- What we will cover
 - – Basic stuff
 - Relationships
 - Constraints
 - Design principles

⟨#⟩

Database System - Nankai



Entity Sets

- *Entity* = “thing” or object.
- *Entity set* = collection of similar entities.
 - Similar to a class in object-oriented languages.
- *Attribute* = property of an entity set.
 - Generally, all entities in a set have the same properties.
 - Attributes are simple values, e.g. integers or character strings.

⟨#⟩

Database System - Nankai



E/R Diagrams

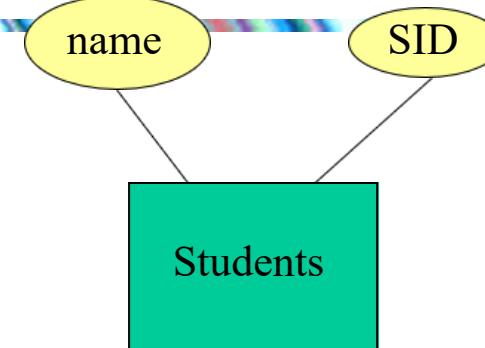
- In an entity-relationship diagram, each entity set is represented by a rectangle.
- Each attribute of an entity set is represented by an oval, with a line to the rectangle representing its entity set.

⟨#⟩

Database System - Nankai



Example



- Entity set Students has two attributes, name and SID (student ID).
- Each Student entity has values for these two attributes, e.g. (张三, 20191234)

‹#›

Database System - Nankai



Relationships

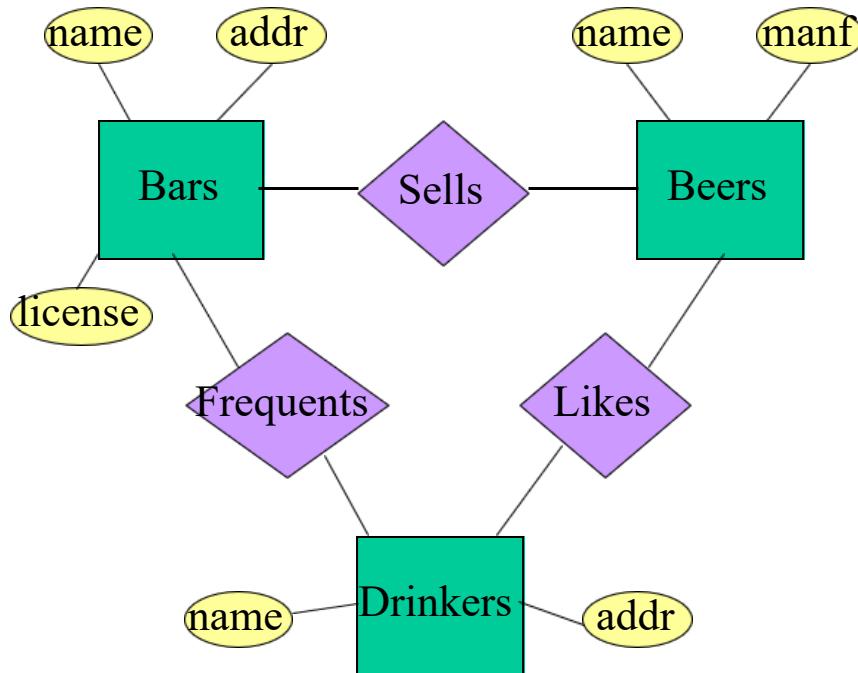
- A relationship connects two or more entity sets.
- It is represented by a diamond, with lines to each of the entity sets involved.

⟨#⟩

Database System - Nankai



Example



Bars sell some beers.

Drinkers like some beers.

Drinkers frequent some bars.

{#}

Database System - Nankai



Relationship Set

- The current “value” of an entity set is the set of entities that belong to it.
 - Example: the set of all bars in our database.
- The “value” of a relationship is a set of lists of currently related entities, one from each of the related entity sets.

⟨#⟩

Database System - Nankai



Example

- For the relationship *Sells*, we might have a relationship set like:

Bar	Beer
Joe's Bar	Bud
Joe's Bar	Miller
Sue's Bar	Bud
Sue's Bar	Pete's Ale
Sue's Bar	Bud Lite

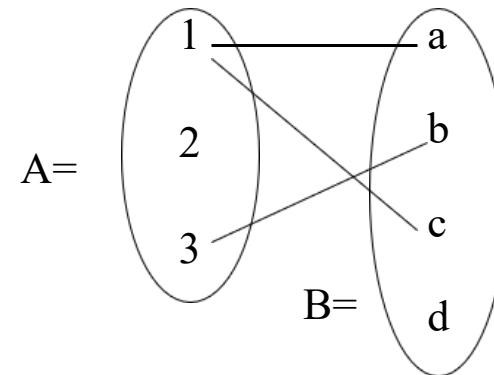
⟨#⟩

Database System - Nankai

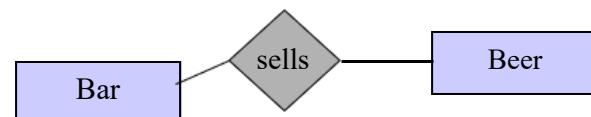


Relations

- A mathematical definition:
 - if A, B are sets, then a relation R is a subset of $A \times B$
- $A=\{1,2,3\}, B=\{a,b,c,d\}$,
 $R = \{(1,a), (1,c), (3,b)\}$



sells is a subset of **Bar** \times **Beer**:



⟨#⟩

Database System - Nankai



Basic stuff: Summary

- Entity, entity set, attribute
- Relation
- Value of an entity set
- Value of a relationship
- Mathematical definition

⟨#⟩

Database System - Nankai

单选题 1分



互动交流一

ER模型的作用是（）？

- A 概念设计
- B 逻辑模式设计
- C 物理模式设计

提交

⟨#⟩

Database System - Nankai

单选题 1分



互动交流二

以下哪个是实体集合?

- A 数据库课程
- B 电影
- C 选课
- D 南开大学

提交

{#}

Database System - Nankai

单选题 1分



互动交流三

以下不可以作为属性的取值类型的是？

- A 字符串
- B 整数数组
- C 浮点数
- D 整数

提交

⟨#⟩

Online System - Nankai

单选题 1分



互动交流四

两个实体集合之间可以有多种不同关系吗？

- A 可以
- B 不可以

提交

⟨#⟩

Database System - Nankai

填空题 1分



互动交流五

从数学定义上来看，关系可以看作是关系所关联的实体集合的 [填空1]

提交

⟨#⟩

Course System - Nankai



Entity -Relationship Model



Why do database systems need modeling tools ?



What is the core technique of conceptual model?



How to design conceptual model of DBs ?

⟨#⟩

Database System - Nankai

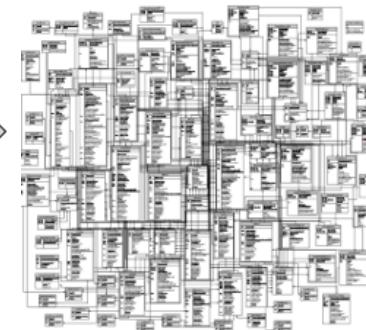


Why do database systems need modeling tools ?

现实世界



机器世界
DBMS支持的数据模型



$R = \text{set of } R_1, R_2, \dots$



领域信息系统



数据准确表达现实信息?

{#}

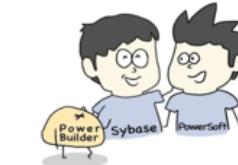
Database System - Nankai



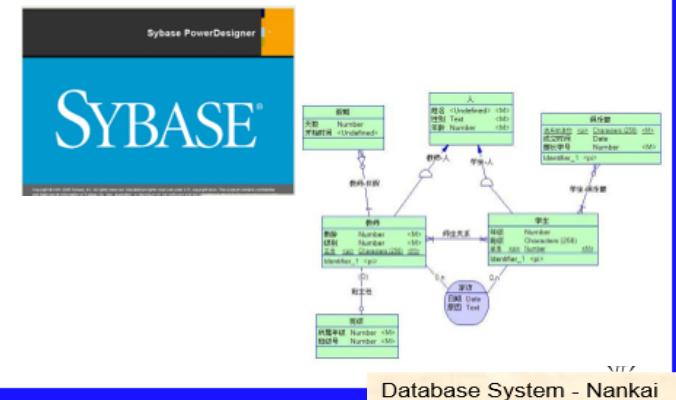
What is the core technique of conceptual model?

- 概念数据模型是现实世界到信息世界的第
一层抽象，它面向用户、面向现实世界，
与DBMS无关
- 概念数据模型应具有的特点：1) 较强的
语义表达能力；2) 简单、清晰、易于用
户理解；3) 可以方便地映射到DBMS的
逻辑模型（例如：ER Model）

到了90年代，Sybase收购PowerSoft，把它的
旗舰产品PowerBuilder纳入麾下，1995年，
Sybase又通过收购获得了PowerDesigner



至此三件神器：设计（PowerDesigner），开发
(PowerBuilder)，数据库（Sybase）集齐了。





How to design ER model of DBs ?

- Basic stuff 预习视频
 - Relationships
 - Constraints
 - Design principles
- 
- 本周课程

⟨#⟩

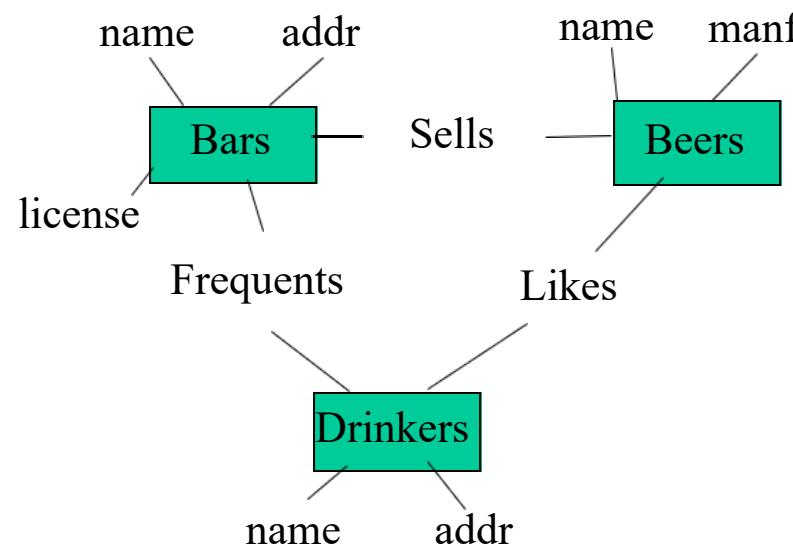
Database System - Nankai

投票 最多可选1项



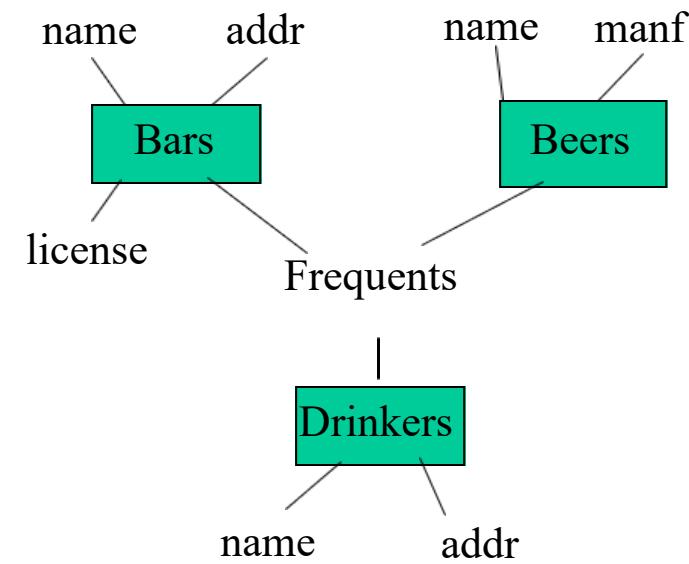
思考题一

以下两个ER模型是否是相同的？



A

相同



B

不相同

提交

⟨#⟩

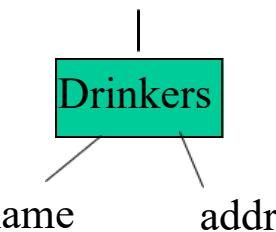
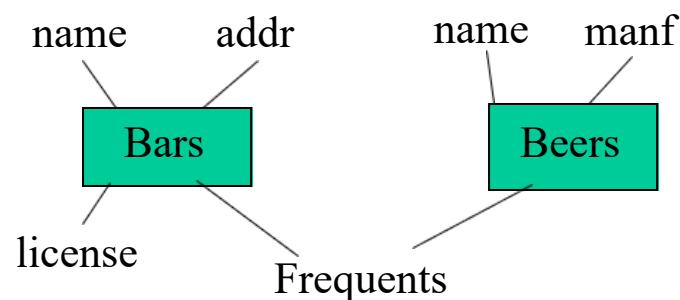
Database System - Nankai

投票 最多可选1项



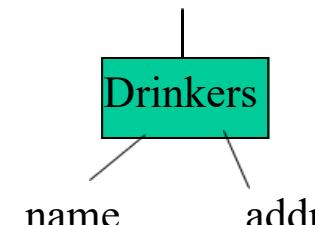
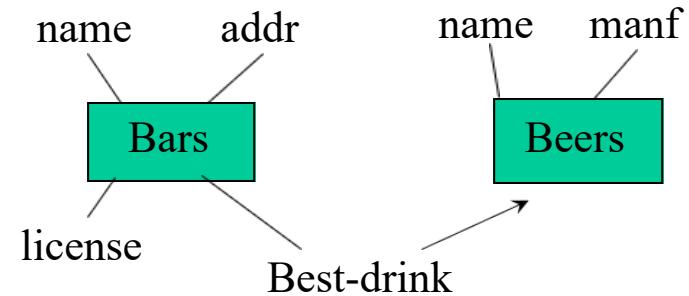
思考题二

以下两个ER模型是否是相同的？



A

相同



B

不相同

提交

⟨#⟩

Database System - Nankai



Week4_Course_part2

ER Model-Relationships

⟨#⟩

Database System - Nankai



ER Model

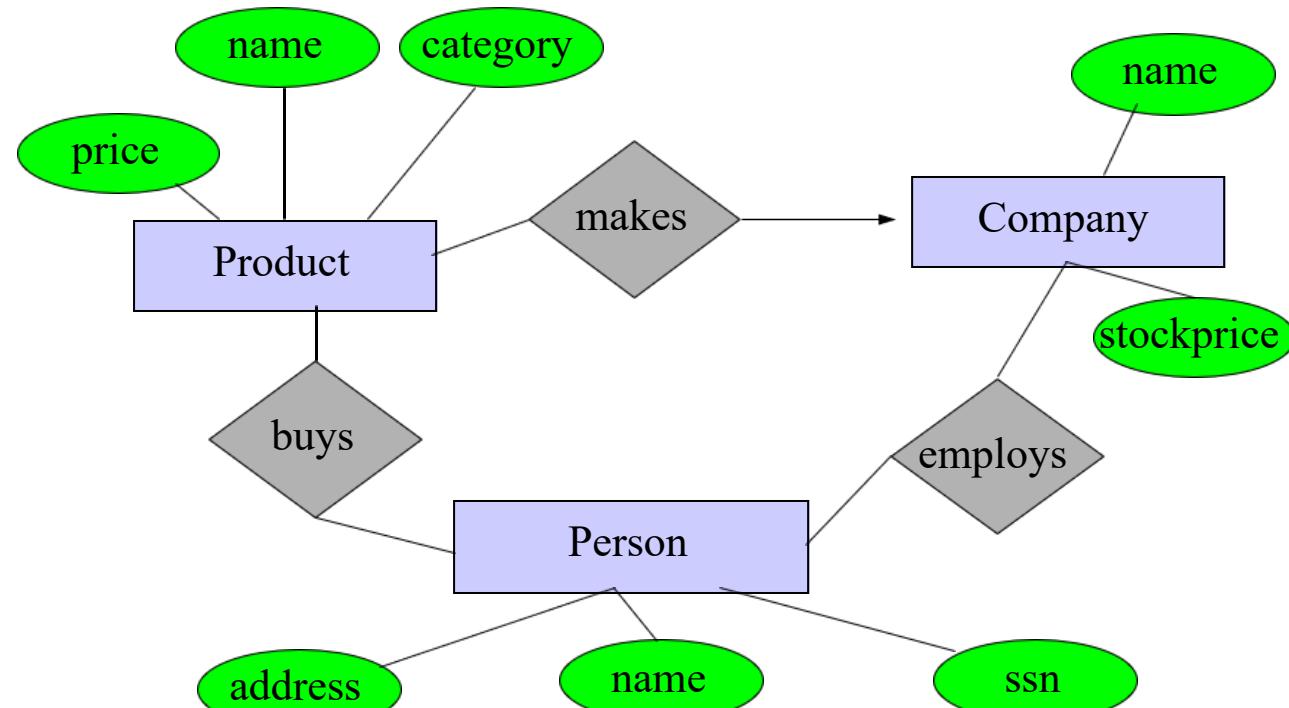
- Basic stuff
- • Relationships
- Constraints
- Design principles

⟨#⟩

Database System - Nankai



ER Diagram



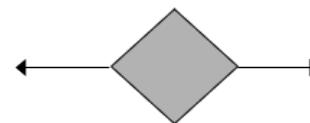
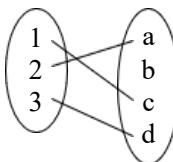
{#}

Database System - Nankai

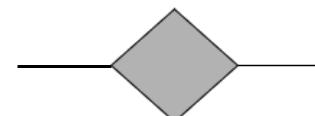
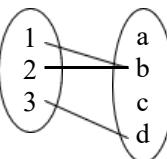


Multiplicity of E/R Relations

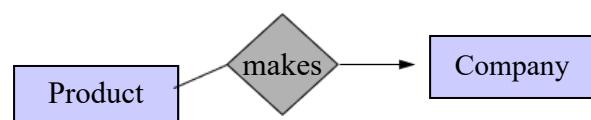
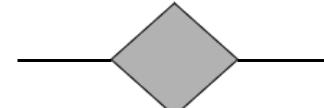
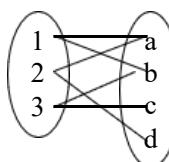
- one-one:



- many-one



- many-many



⟨#⟩

Database System - Nankai



Representing “Multiplicity”

- Show a many-one relationship by an arrow entering the “one” side.
- Show a one-one relationship by arrows entering both entity sets.
- In some situations, we can also assert “exactly one,” i.e., each entity of one set must be related to exactly one entity of the other set. To do so, we use a rounded arrow.

⟨#⟩

Database System - Nankai



Example

- Consider *Best-seller* between *Manfs* and *Beers*.
- Some beers are not the best-seller of any manufacturer, so a rounded arrow to *Manfs* would be inappropriate.
- But a manufacturer has to have a best-seller (we assume they are beer manufacturers).



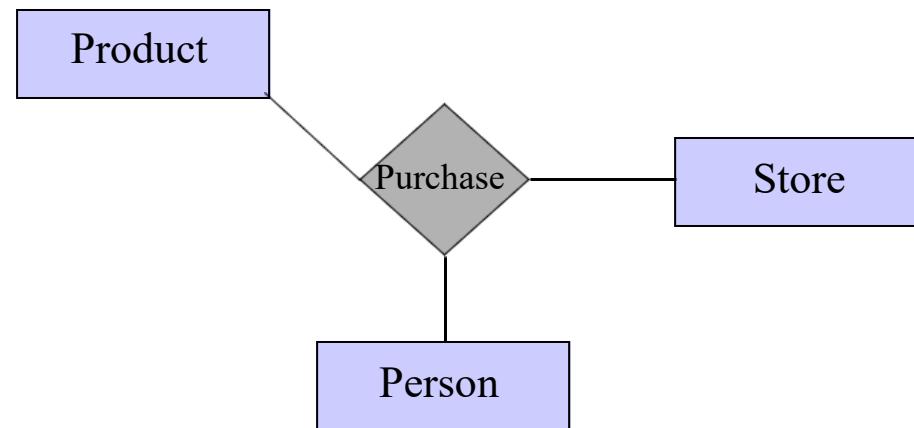
⟨#⟩

Database System - Nankai



Multiway Relationships

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



Can still model as a mathematical set: (Joe, Macy's, iPhone8)

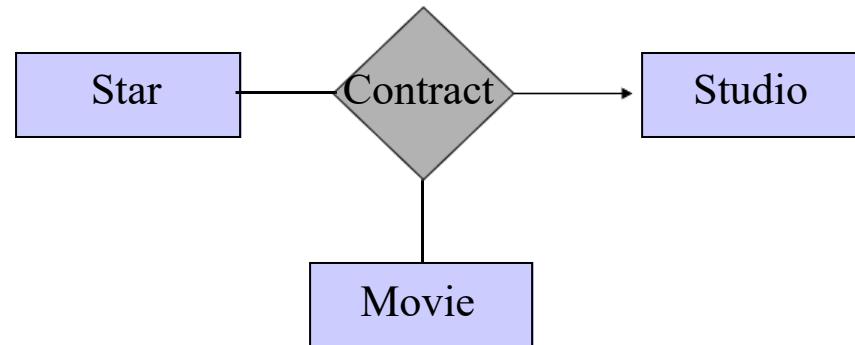
⟨#⟩

Database System - Nankai



Arrows in Multiway Relationships

Q: what does the arrow mean ?



A: if I know the movie and star, I know the studio too

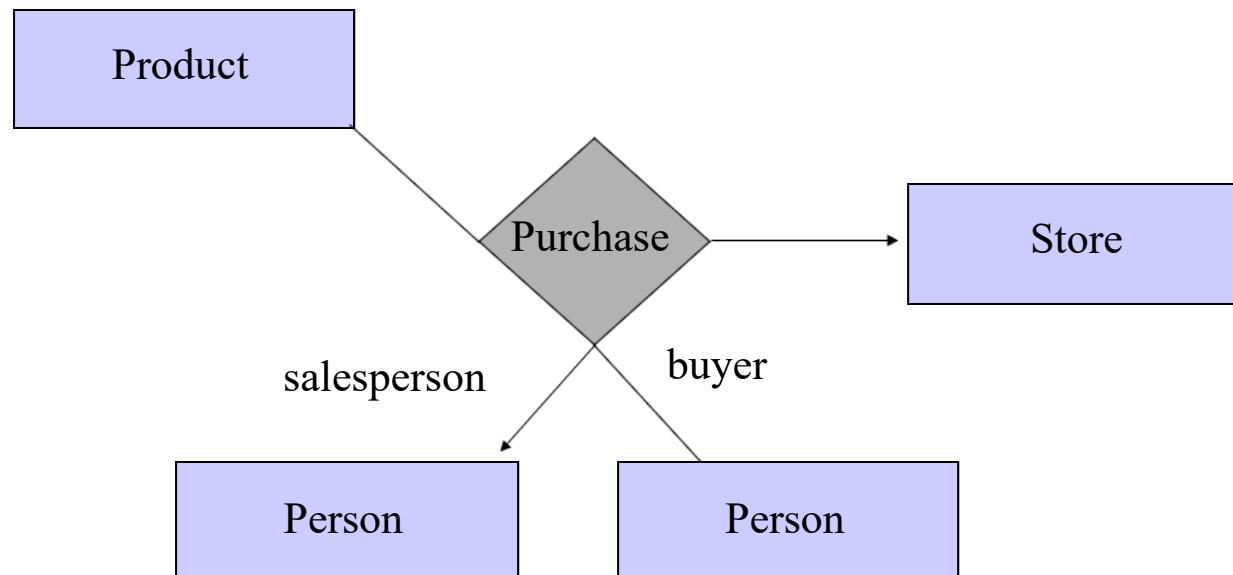
{#}

Database System - Nankai



Roles in Relationships

What if we need an entity set twice in one relationship?



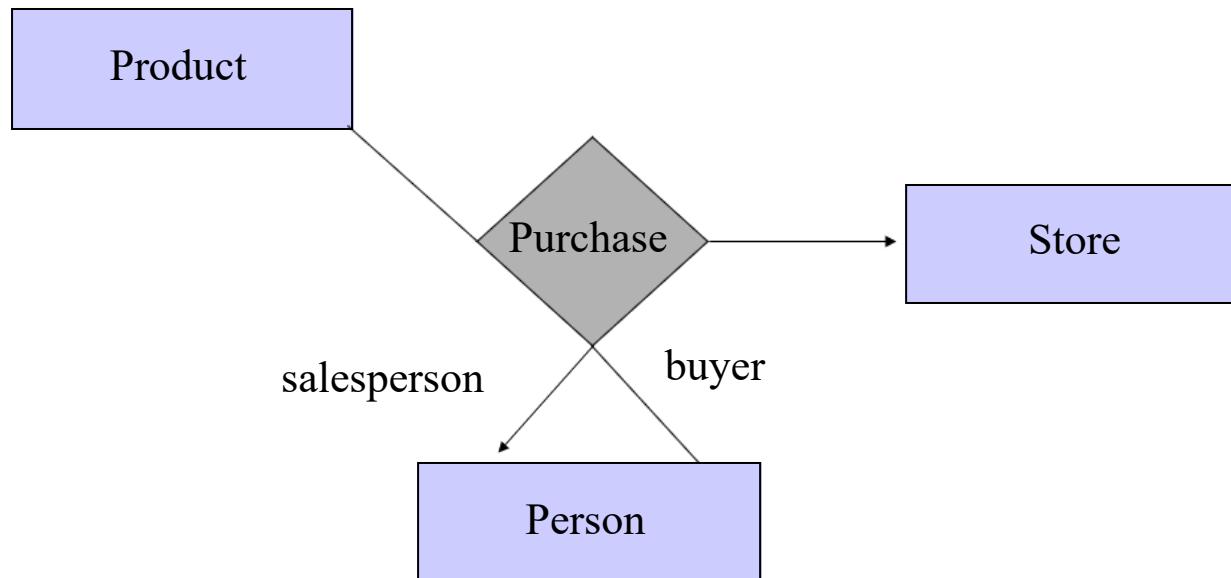
{#}

Database System - Nankai



Roles in Relationships

What if we need an entity set twice in one relationship?

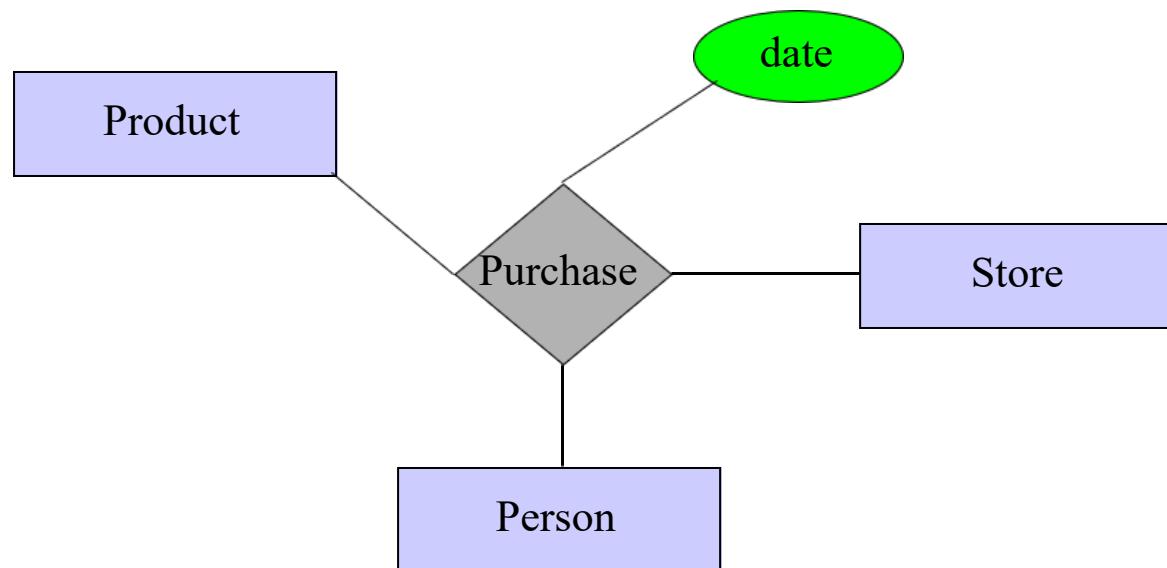


⟨#⟩

Database System - Nankai



Attributes on Relationships

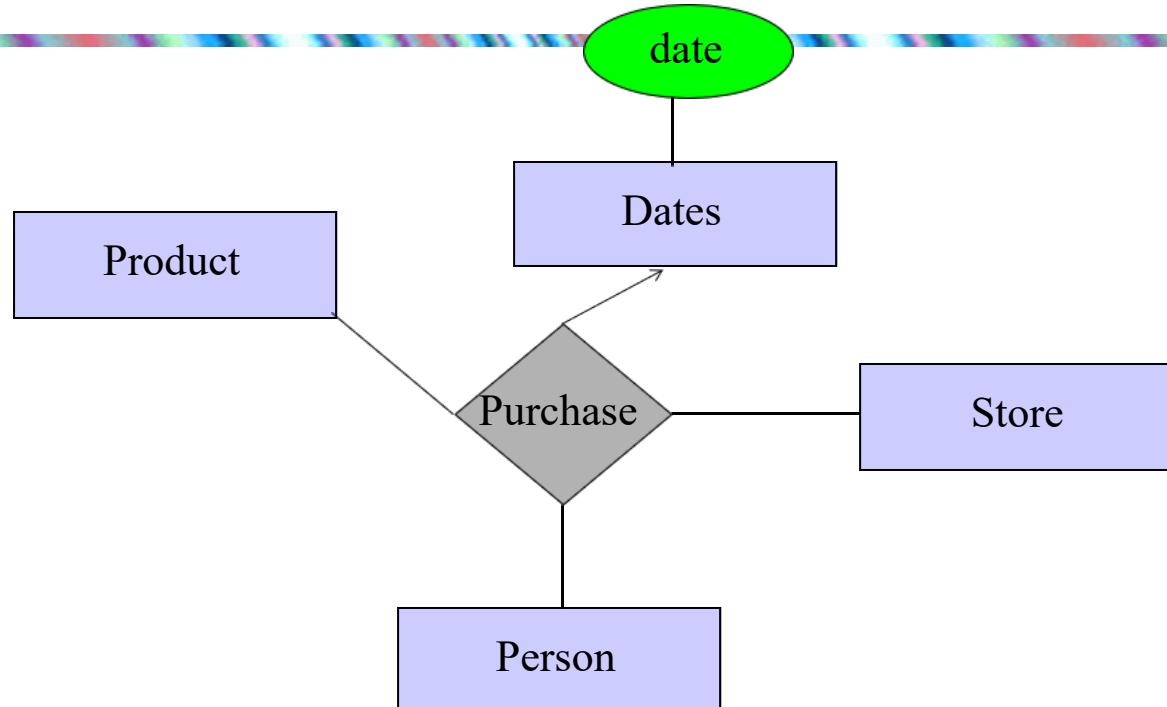


⟨#⟩

Database System - Nankai



Attributes on Relationships



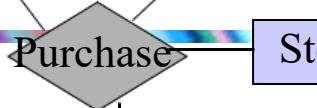
⟨#⟩

Database System - Nankai



Product

date



Store

Person

date

Converting Multiway Relationships to Binary

Purchase

(connecting entity set)



Product



Store



Person

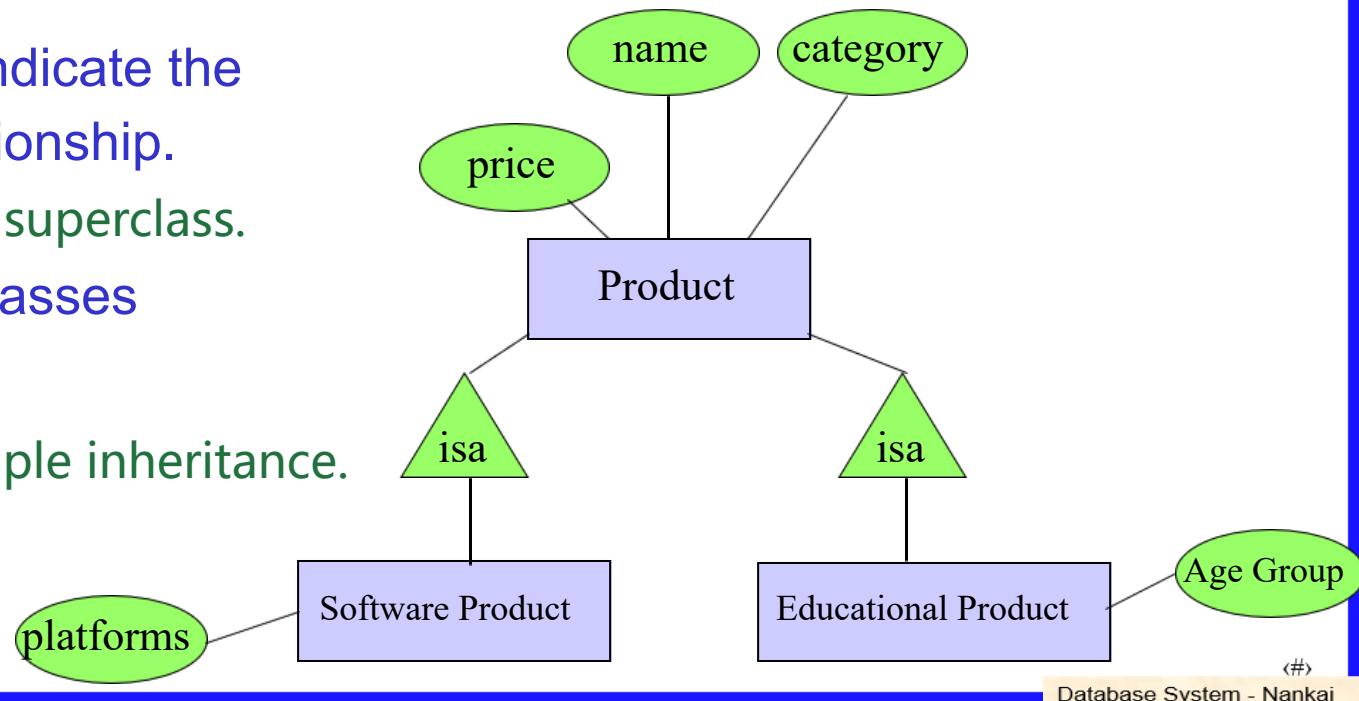
{#}

Database System - Nankai



Subclasses in ER Diagrams

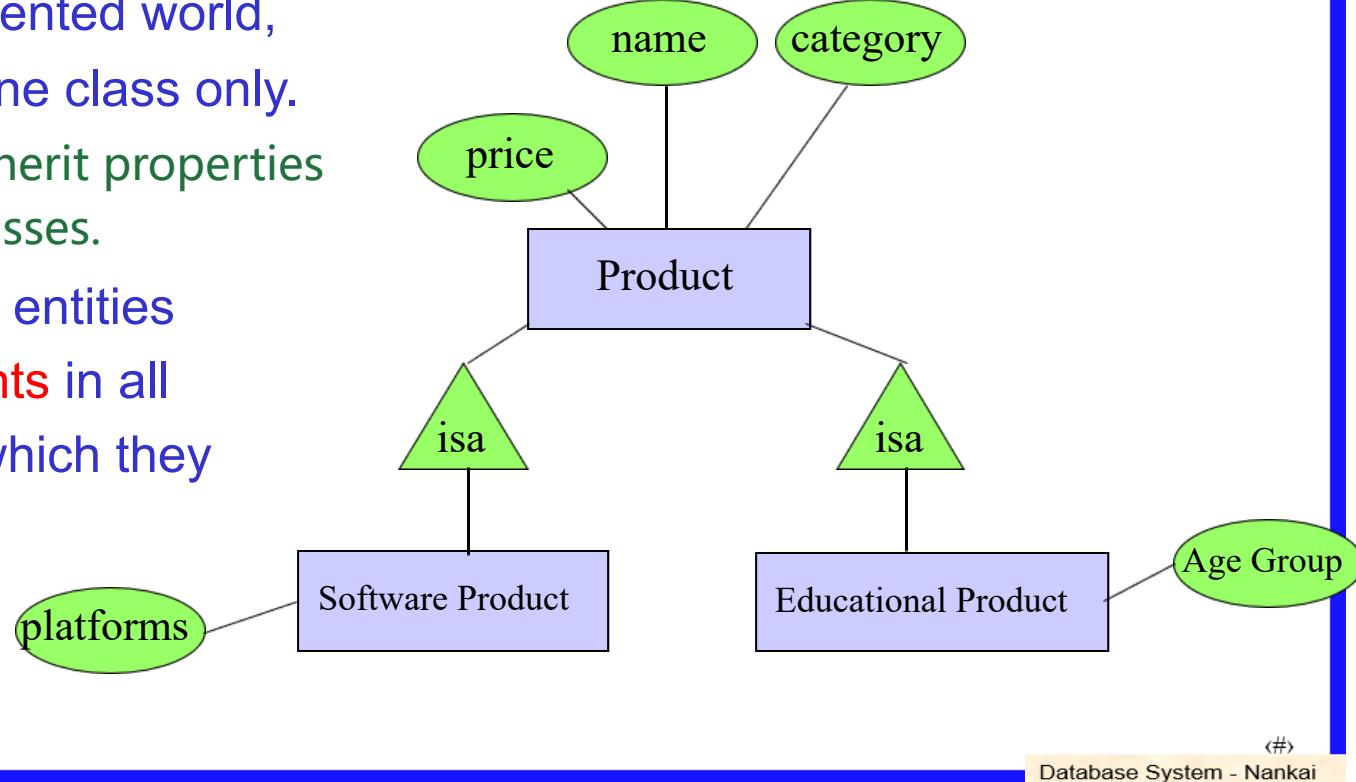
- Subclass = special case = fewer entities = more properties.
- Isa triangles indicate the subclass relationship.
 - Point to the superclass.
- Assume subclasses form a tree.
 - I.e., no multiple inheritance.





E/R Vs. Object-Oriented Subclasses

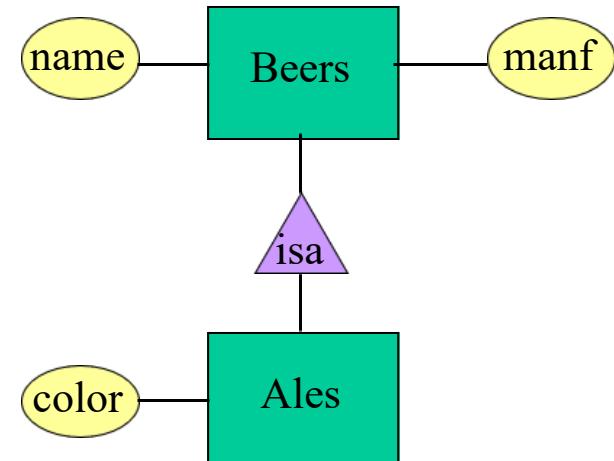
- In the object-oriented world, objects are in one class only.
 - Subclasses inherit properties from superclasses.
- In contrast, E/R entities have **components** in all subclasses to which they belong.
 - E.g., 雨课堂





Example

- 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 attributes of beers, ales also have the attribute *color*.



⟨#⟩

Database System - Nankai



Relationships: Summary

- Relation Multiplicity
 - many-one, one-one, many-many
- Multiway Relationships
 - Converting a multiway one into many binary ones
- Roles in Relationships
- Attributes on Relationships
- Subclasses

⟨#⟩

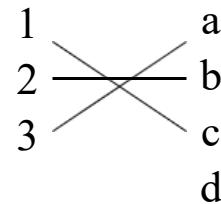
Database System - Nankai

单选题 1分



互动交流一

以下两个实体集合的联系可能是ER模型中的哪种关系？



- A many-one
- B one-one
- C many-many

提交

⟨#⟩

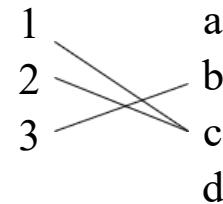
Database System - Nankai

单选题 1分



互动交流二

以下两个实体集合的联系可能是ER模型中的哪种关系？



A

many-one

B

one-one

C

many-many

提交

⟨#⟩

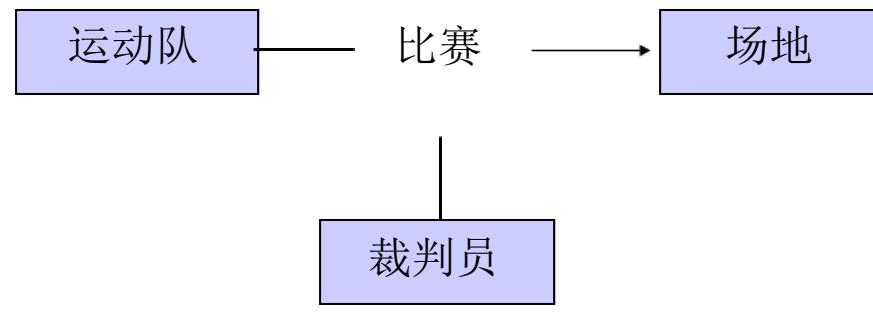
Database System - Nankai

单选题 1分



互动交流三

以下多路联系中的箭头是否合理?



A 合理

B 不合理

提交

{#}

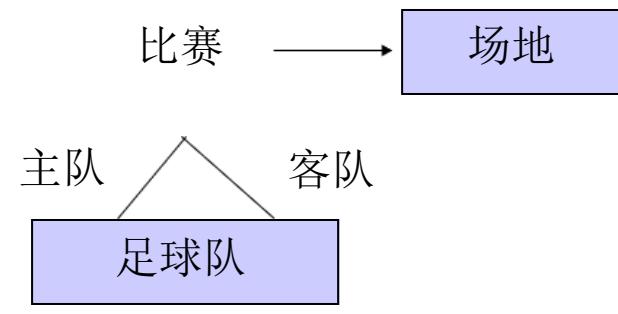
Database System - Nankai

单选题 1分



互动交流四

以下ER图的背景是中国足球超级联赛，该多路联系中的箭头是否合理？



A 合理

B 不合理

提交

{#}

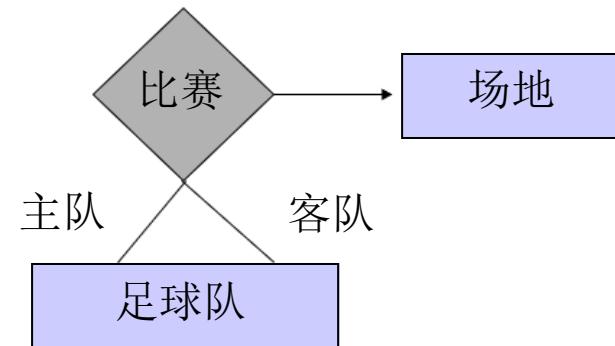
Database System - Nankai

填空题 1分



互动交流五

以下ER图的背景是中国足球超级联赛，如果想表达比赛时间，通常做法应该怎么修改ER图？ [填空1]



提交

⟨#⟩

Course System - Nankai

单选题 1分



互动交流六

如果实体集合A中当前拥有100个实体，那么实体集合B所拥有的实体数量是多少？

- I. 1
- II. 100
- III. 200



- A I or II
- B II or III
- C II only
- D I,II or III

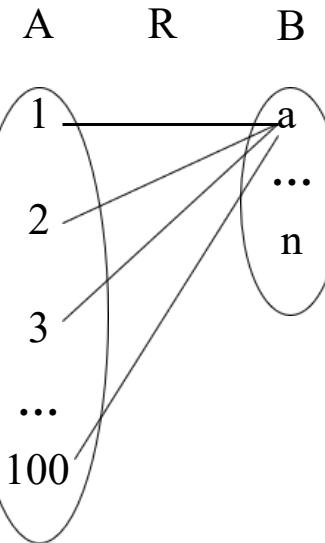
提交

⟨#⟩

Database System - Nankai



讲解



<#>

Database System - Nankai



思考题

我们需要为学校设计一个学生信息管理系统，在设计该系统时，本科生、研究生是否需要作为两个不同的实体集合？为什么？

(请用弹幕回答)

怎么设计ER模型才能体现他们很相似（具有很多共同属性，如学号、姓名、性别、出生日期、身份证号等），但又不完全相同（他们拥有各自属性，如本科生具有班导师，研究生具有研究方向、导师）的特性？

(请用弹幕回答)

⟨#⟩

Database System - Nankai



Week4_Course_part3

ER Model-Constraints

⟨#⟩

Database System - Nankai



ER Model

- Basic stuff
- Relationships
- Constraints
- Design principles

⟨#⟩

Database System - Nankai



Modeling Constraints

Finding constraints is part of the modeling process.

Commonly used constraints:

Keys: social security number uniquely identifies a person.

Referential integrity constraints: if you work for a company, it must exist in the database.

Domain constraints: peoples' ages are between 0 and 200.

General constraints: all others (at most 50 students enroll in a class)

⟨#⟩

Database System - Nankai



Keys

- We must designate a key for every entity set.
- A *key* is a set of attributes for one entity set such that no two entities in this set agree on all the attributes of the key.
 - It is allowed for two entities to agree on some, but not all, of the key attributes.

⟨#⟩

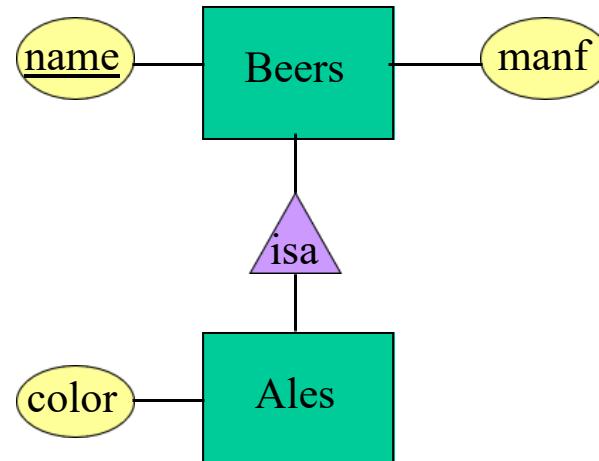
Database System - Nankai



Keys in E/R Diagrams

- Underline the key attribute(s).
- In an Isa hierarchy, only the root entity set has a key, and it must serve as the key for all entities in the hierarchy.

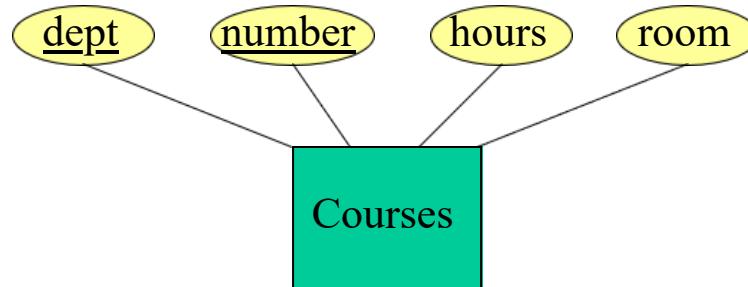
Example:
name is Key
for *Beers*



Database System - Nankai



Example: a Multi-attribute Key



- Note that *hours* and *room* could also serve as a key, but we must select only one key.

⟨#⟩

Database System - Nankai



More about Keys

- Every entity set must have a key
- 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
 - only the root entity set has a key

⟨#⟩

Database System - Nankai



Weak Entity Sets

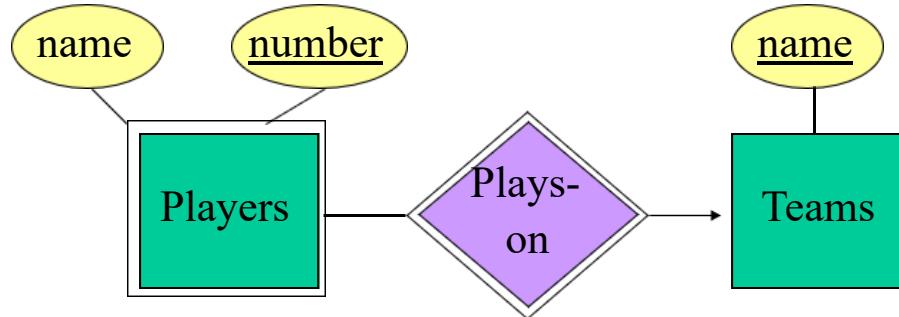
- Occasionally, entities of an entity set need “help” to identify them uniquely.
- Entity set E is said to be *weak* if in order to identify entities of E uniquely, we need to follow one or more many-one relationships from E and include the key of the related entities from the connected entity sets.

⟨#⟩

Database System - Nankai



In E/R Diagrams



- *name* is almost a key for players, but there might be two with the same name.
- *number* is certainly not a key, since players on two teams could have the same number.
- But *number*, together with the *Team* related to the player by *Plays-on* should be unique.
- **Double diamond for many-one supporting relationship.**
- **Double rectangle for the weak entity set.**

⟨#⟩

Database System - Nankai



Weak Entity Set Rules

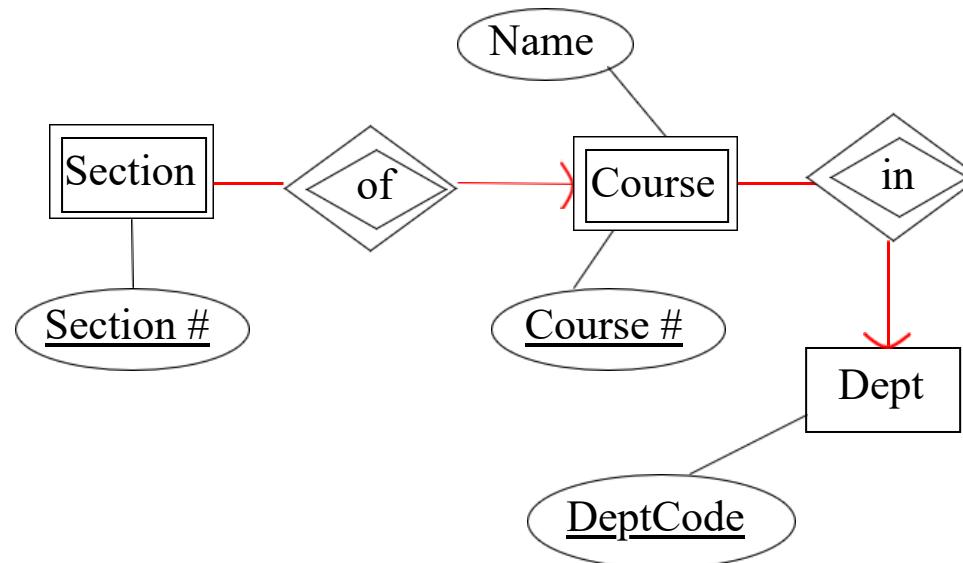
- A weak entity set has one or more many-one relationships to other (supporting) entity sets.
 - Not every many-one relationship from a weak entity set need be supporting.
- The key for a weak entity set is its own underlined attributes and the keys for the supporting entity sets.
 - E.g., *player-number* and *team-name* is a key for *Players* in the previous example.

⟨#⟩

Database System - Nankai



Chain of Weakness



Dept' s Key: (DeptCode)

Course' s Key: (Course #, DeptCode)

Section' s key: (Section #, Course #, DeptCode)

{#}

Database System - Nankai



Constraints: Summary

- Key
- Weak entity set

⟨#⟩

Database System - Nankai



互动交流一

关于键约束的描述，错误的是：

- A 一个实体集合可以有多个键
- B 一个键只能由一个属性构成
- C 同一实体集合中的实体在键属性上可以部分相同
- D 在isa层次结构中，只需为根节点的实体集合设置键

提交

⟨#⟩

Database System - Nankai



互动交流二

关于弱实体集合的描述，错误的是：

- A 支持联系可以是多对多的联系
- B 从弱实体集合出发的联系可能是支持联系，也可能不是支持联系
- C 支持联系用双边菱形表示
- D 弱实体的键由自己的键属性加上它所有支持实体集合的键属性构成

提交

⟨#⟩

Database System - Nankai

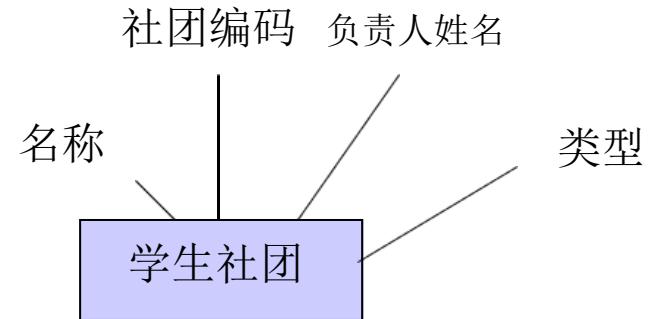
多选题 1分



互动交流三

以下ER模型的背景是南开大学需要一个社团管理系统，指出该实体集合的键可以是哪些情况？

- A 名称
- B 社团编码
- C 负责人姓名
- D 负责人姓名，类型



提交

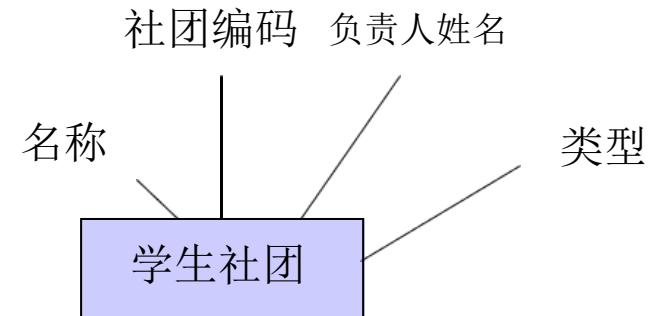
Database System - Nankai



互动交流四

以下ER模型的背景变为教育部需要一个高校社团管理系统，
指出该实体集合的键可以是哪些情况？

- A 名称
- B 社团编码
- C 负责人姓名
- D 弱实体集合



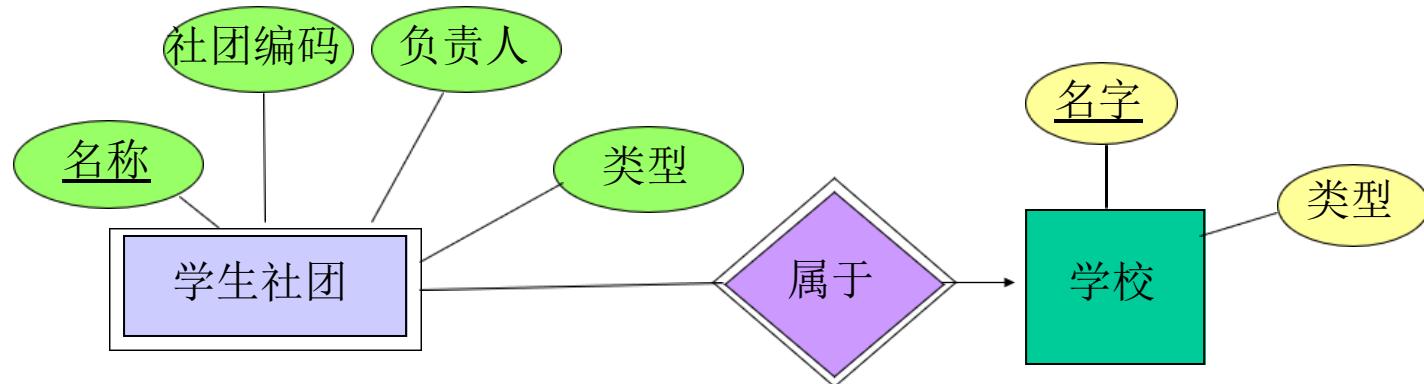
提交

⟨#⟩

Course System - Nankai



讲解



<#>

Database System - Nankai



Week4_Course_part4

ER Model-Design principles

⟨#⟩

Database System - Nankai



ER Model

- Basic stuff
 - Relationships
 - Constraints
 - Design principles
- 

⟨#⟩

Database System - Nankai



Design Techniques

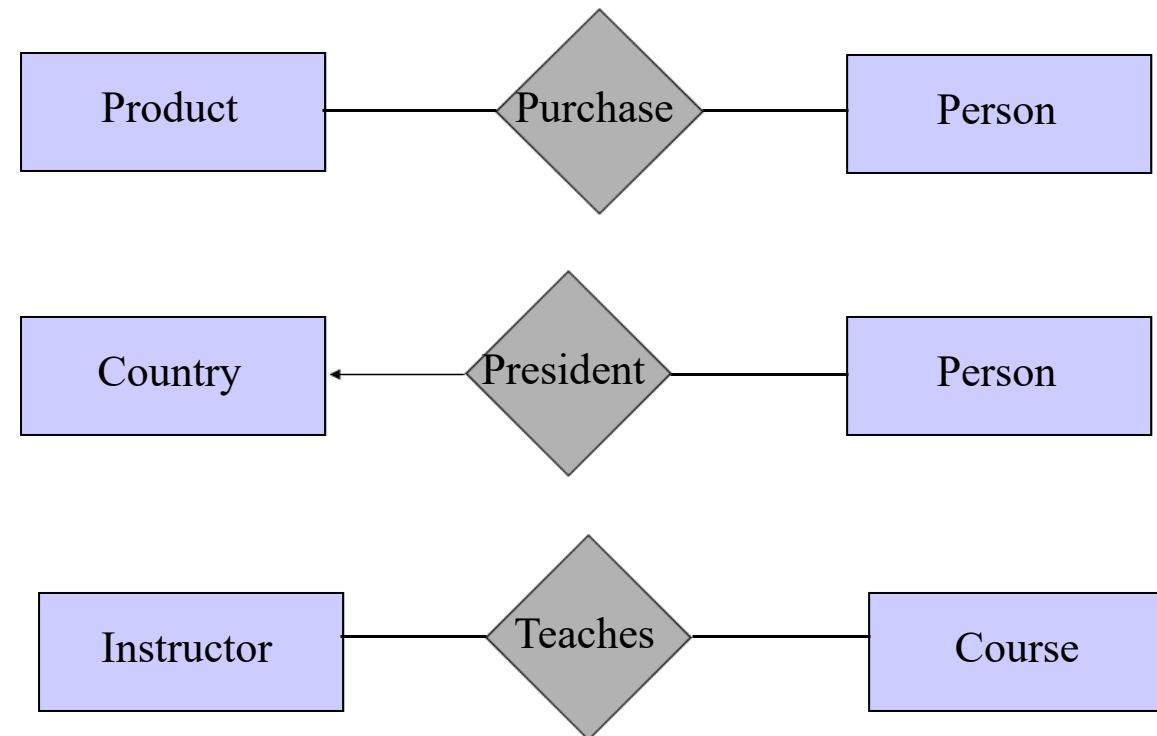
1. Be Faithful
 - Design should reflect your (possibly vague) ideas of the data.
2. Avoid redundancy
3. Don't use an entity set when an attribute will do
4. Limit the use of weak entity sets

⟨#⟩

Database System - Nankai



Design Principles 1: Be Faithful



{#}

Database System - Nankai



Avoiding Redundancy

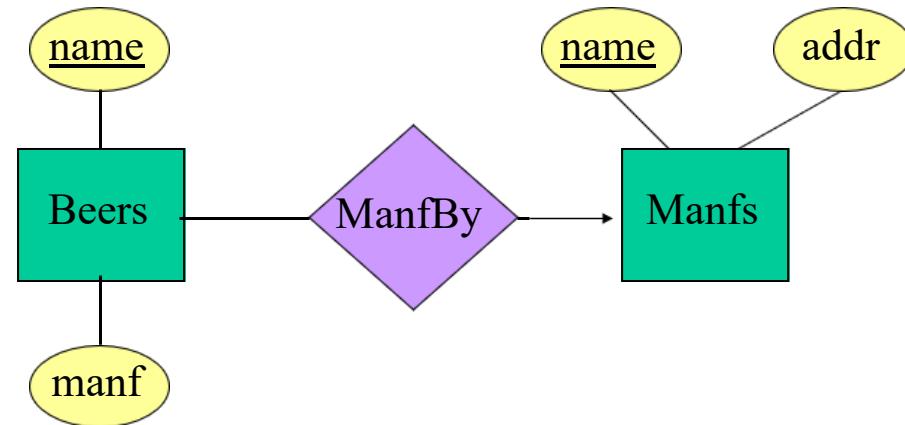
- Redundancy occurs when we say the same thing in two different ways.
- Redundancy wastes space and (more importantly) encourages inconsistency.
 - The two instances of the same fact may become inconsistent if we change one and forget to change the other, related version.

⟨#⟩

Database System - Nankai



Example: Bad



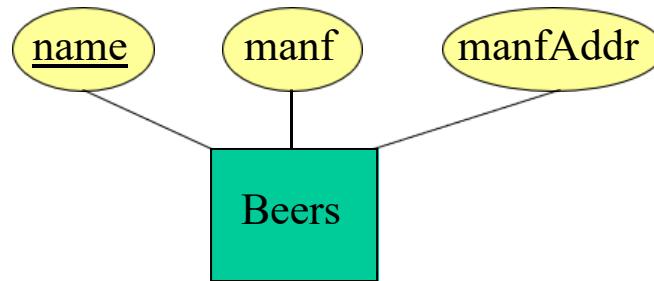
This design states the manufacturer of a beer twice: as an attribute and as a related entity.

{#}

Database System - Nankai



Example: Bad



name	manf	manfAddr
1	A	A-Address
2	A	A-Address
3	A	A-Address
4	B	B-Address

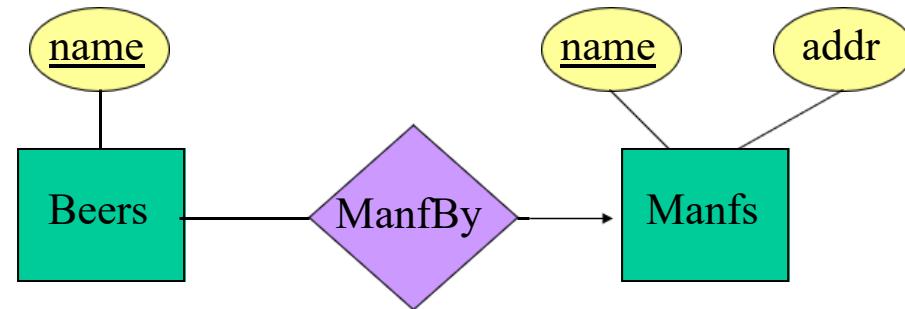
This design repeats the manufacturer's address once for each beer; loses the address if there are temporarily no beers for a manufacturer.

⟨#⟩

Database System - Nankai



Example: Good



This design stores the information of each manufacturer exactly once.

{#}

Database System - Nankai



Entity Sets Versus Attributes

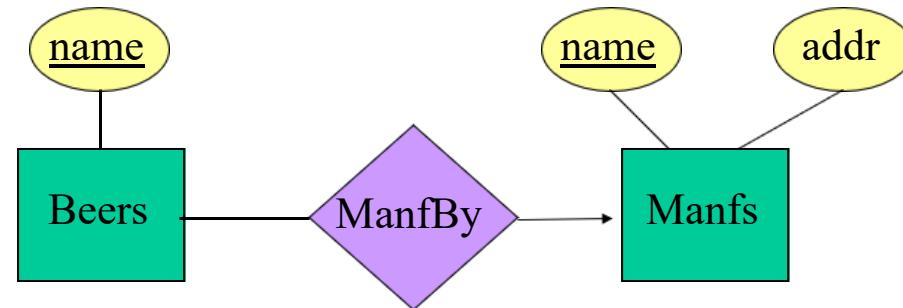
- An entity set should satisfy at least one of the following conditions:
 - It is more than the name of something; it has at least one nonkey attribute.
or
 - It is the “many” in a many-one or many-many relationship.

⟨#⟩

Database System - Nankai



Example: Good



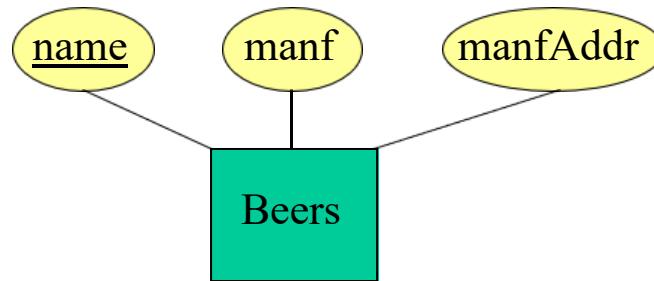
- *Manfs* deserves to be an entity set because of the nonkey attribute *addr*.
- *Beers* deserves to be an entity set because it is the “many” of the many-one relationship *ManfBy*.

⟨#⟩

Database System - Nankai



Example: Bad

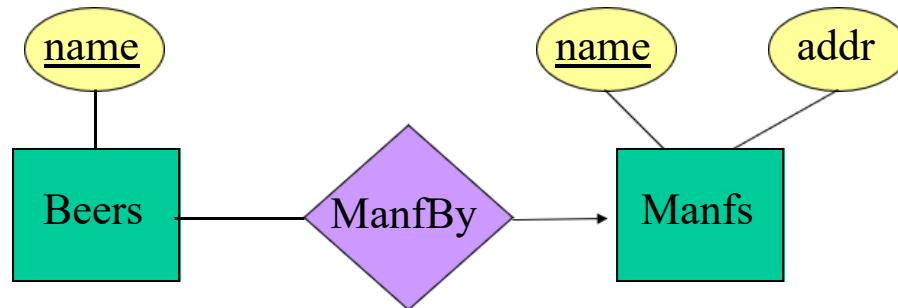


⟨#⟩

Database System - Nankai



Example: Good



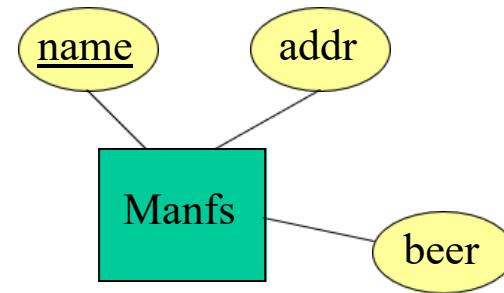
- *Manfs* deserves to be an entity set because of the nonkey attribute *addr*.
- *Beers* deserves to be an entity set because it is the “many” of the many-one relationship *ManfBy*.

⟨#⟩

Database System - Nankai



Example: Bad

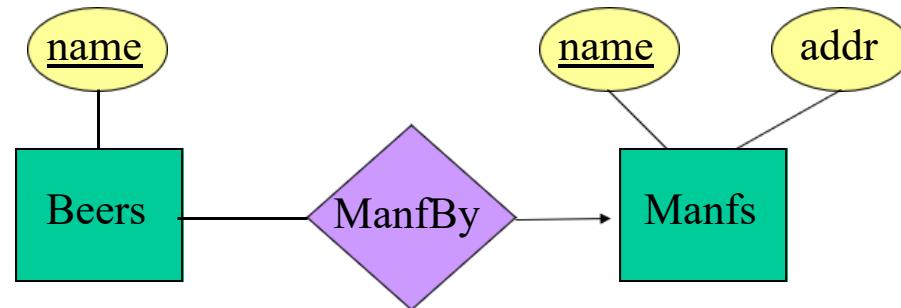


⟨#⟩

Database System - Nankai



Example: Good



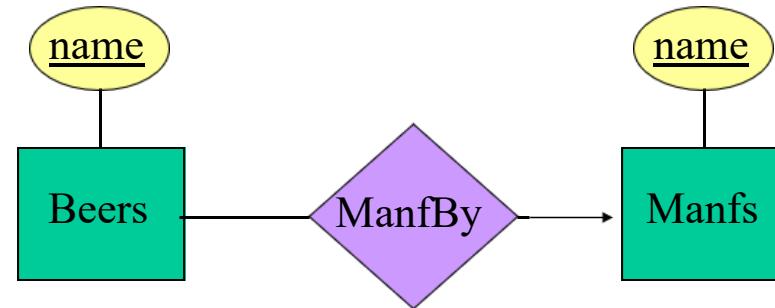
- *Manfs* deserves to be an entity set because of the nonkey attribute *addr*.
- *Beers* deserves to be an entity set because it is the “many” of the many-one relationship *ManfBy*.

⟨#⟩

Database System - Nankai



Example: Bad



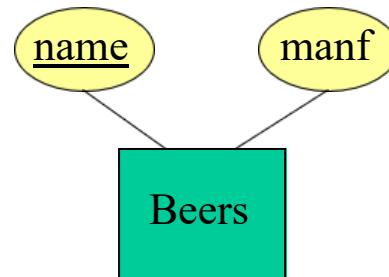
Since the manufacturer is nothing but a name,
and is not at the “many” end of any relationship,
it should not be an entity set.

⟨#⟩

Database System - Nankai



Example: Good



There is no need to make the manufacturer an entity set, because we record nothing about manufacturers besides their name.

⟨#⟩

Database System - Nankai



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.

⟨#⟩

Database System - Nankai



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.

⟨#⟩

Database System - Nankai



Design principles: Summary

- Be faithful
- Avoid redundancy
- Don't use an entity set when an attribute will do
- Limit the use of weak entity sets

⟨#⟩

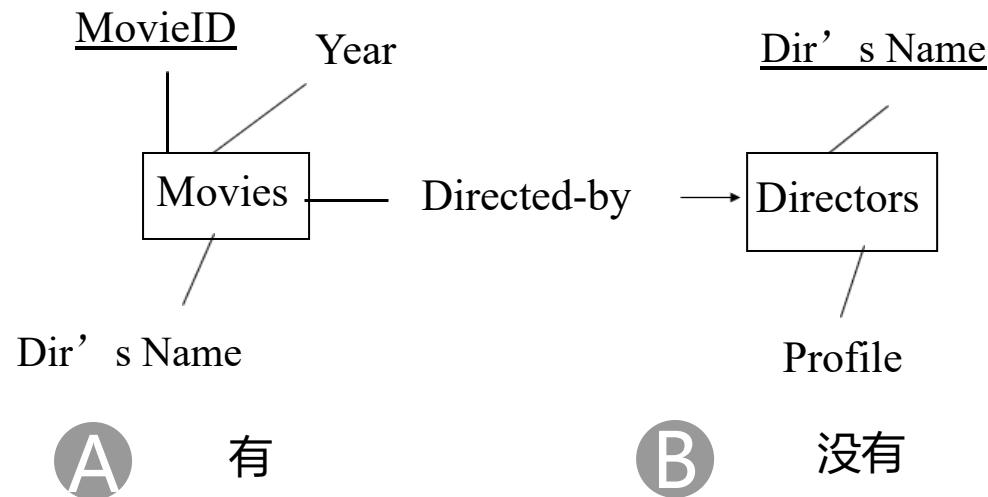
Database System - Nankai

单选题 1分



互动交流一

下面ER图中有冗余吗？



提交

{#}

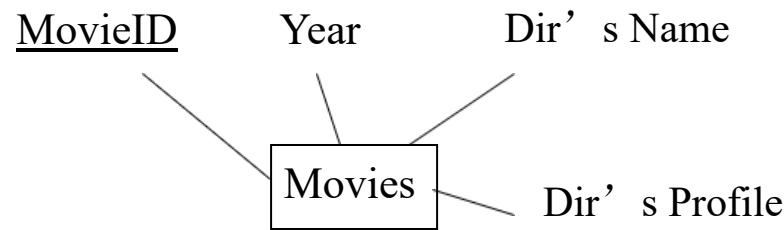
Database System - Nankai

单选题 1分



互动交流二

下面ER图中有冗余吗？



A 有

B 没有

提交

⟨#⟩

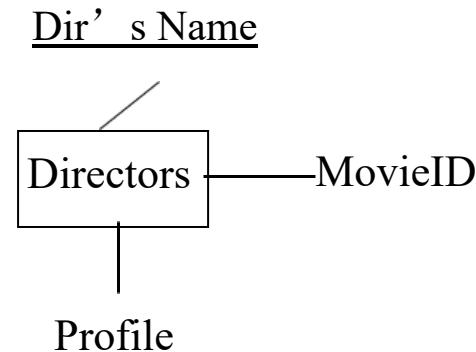
Database System - Nankai

单选题 1分



互动交流三

下面ER图合理吗？



A 合理

B 不合理

提交

⟨#⟩

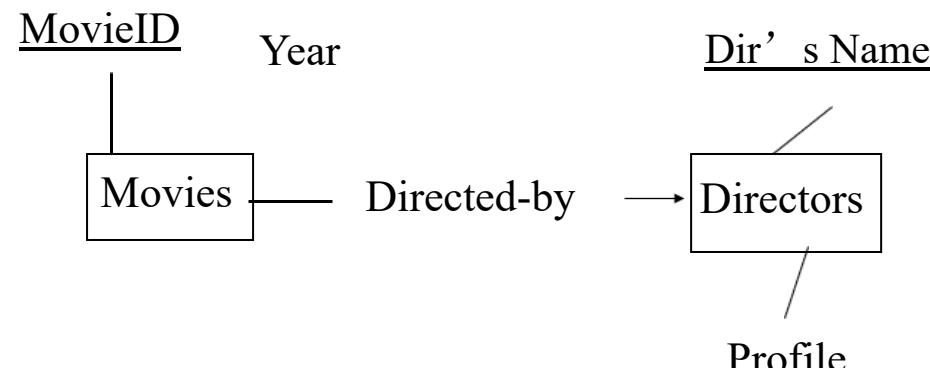
Database System - Nankai

单选题 1分



互动交流四

下面ER图合理吗？



A 合理

B 不合理

提交

⟨#⟩

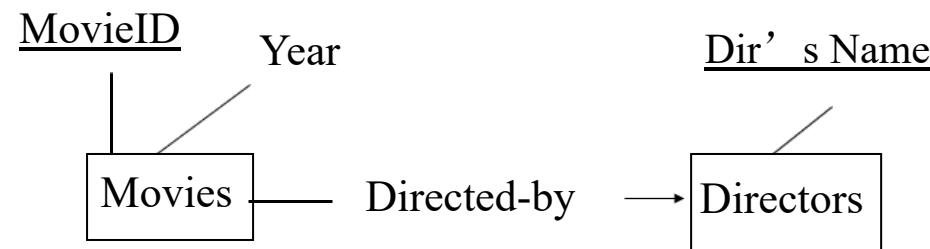
Online Database System - Nankai

单选题 1分



互动交流五

下面ER图合理吗？



A 合理

B 不合理

提交

{#}

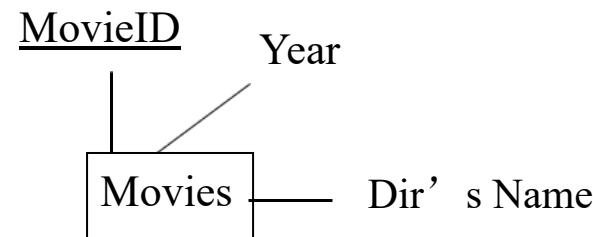
Database System - Nankai

单选题 1分



互动交流六

下面ER图合理吗？



A 合理

B 不合理

提交

⟨#⟩

Database System - Nankai