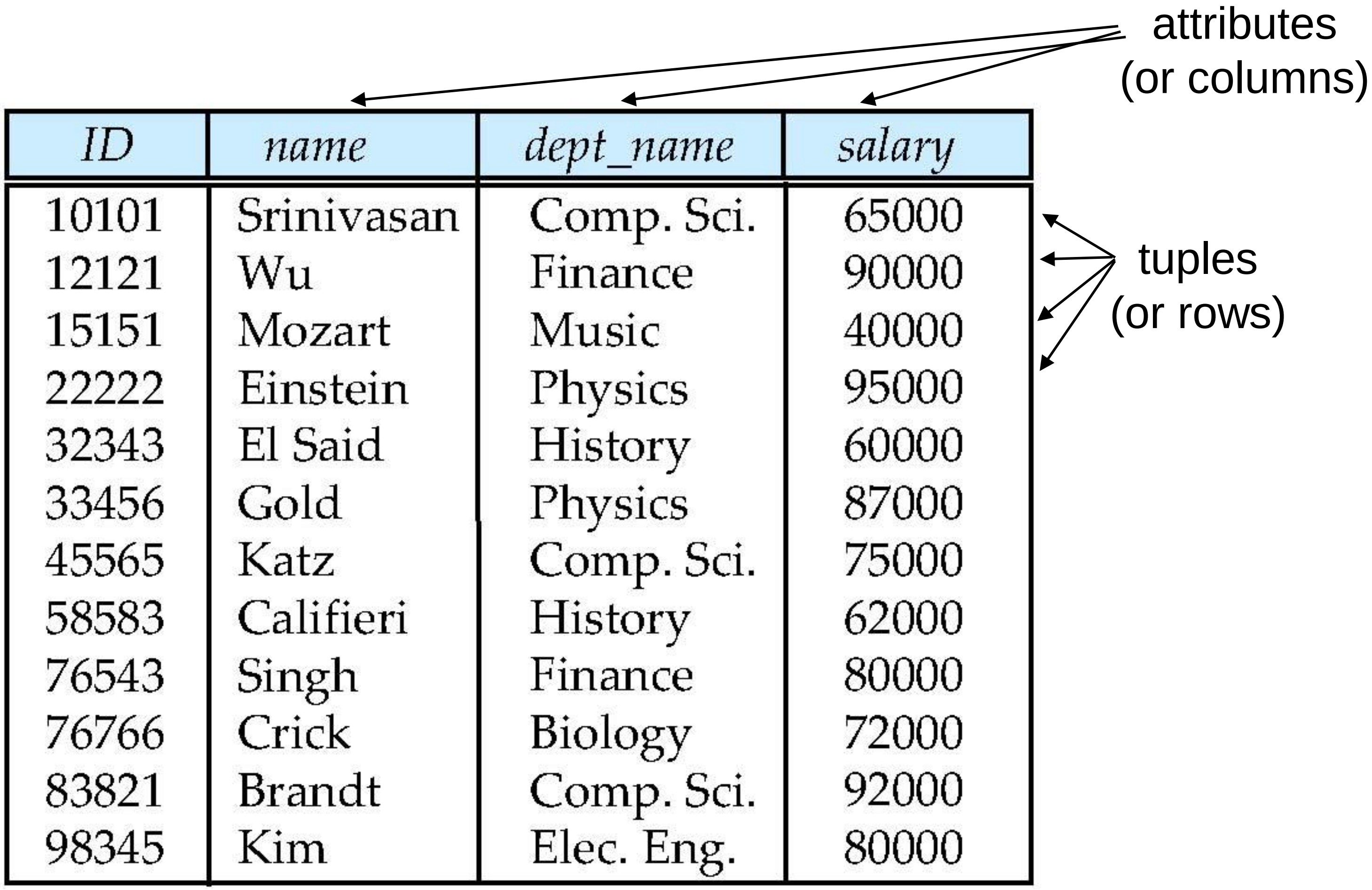
Chapter 2: Intro to Relational Model

- **Data model（数据模型）**: a collection of conceptual tools for describing data, data relationships, data semantics, and consistency constraints.

- **Relational Model（关系模型）:** is the most widely used model today.
  - **Main concept**: relation, basically a table with rows and columns.

Database System Concepts, 6th Ed.

Example of a Relation



attributes
(or columns)

tuples
(or rows)

ID	*name*	*dept_name*	*salary*
10101	Srinivasan	Comp. Sci.	65000
12121	Wu	Finance	90000
15151	Mozart	Music	40000
22222	Einstein	Physics	95000
32343	El Said	History	60000
33456	Gold	Physics	87000
45565	Katz	Comp. Sci.	75000
58583	Califieri	History	62000
76543	Singh	Finance	80000
76766	Crick	Biology	72000
83821	Brandt	Comp. Sci.	92000
98345	Kim	Elec. Eng.	80000

Figure 2.1 The *instructor* relation.

Attribute Domain Types
属性域类型

- The set of allowed values for each attribute is called the **domain** of the attribute。
- 对于属性域，要注意以下两点：
  - Attribute values are (normally) required to be **atomic（原子值 ）**; that is, indivisible（不可分割）
  - The special value ***null*** is a member of every domain
    - 缺点：The null value（空值） causes complications in the definition of many operations

Relation Schema and Instance

理解：变量的类型与值

- *Relation Schema*（*关系模式*）
  - $A_1, A_2, \ldots, A_n$ are *attributes*
  - $R = (A_1, A_2, \ldots, A_n)$ is a *relation schema*

  Example:

  instructor = (*ID*, *name*, *dept_name*, *salary*)

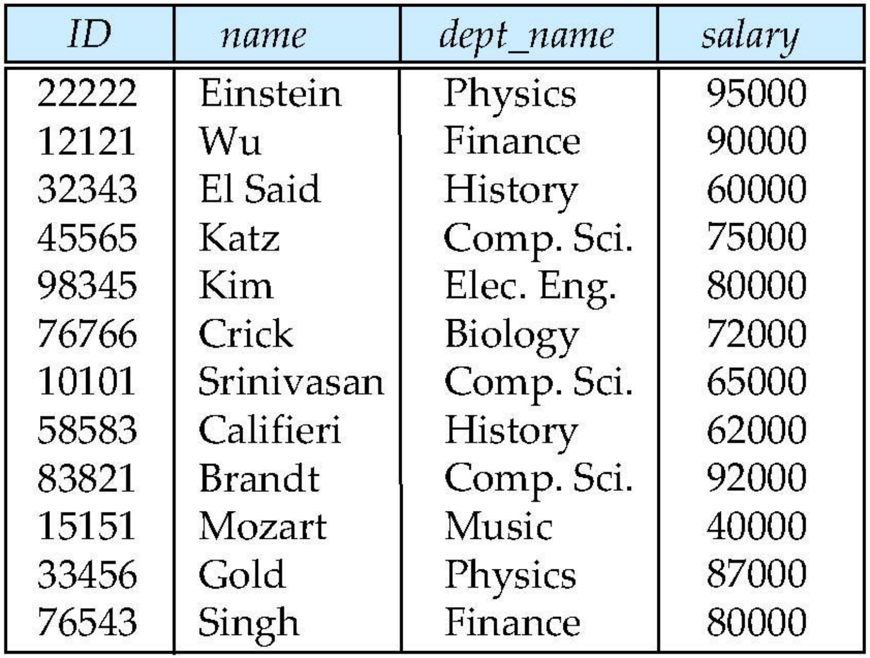
- *Relation Instance*（*关系实例*）
  - Formally, given sets $D_1, D_2, \ldots. D_n$. a **relation instrance** ***r*** is a subset of

    $D_1 \times D_2 \times \ldots \times D_n$

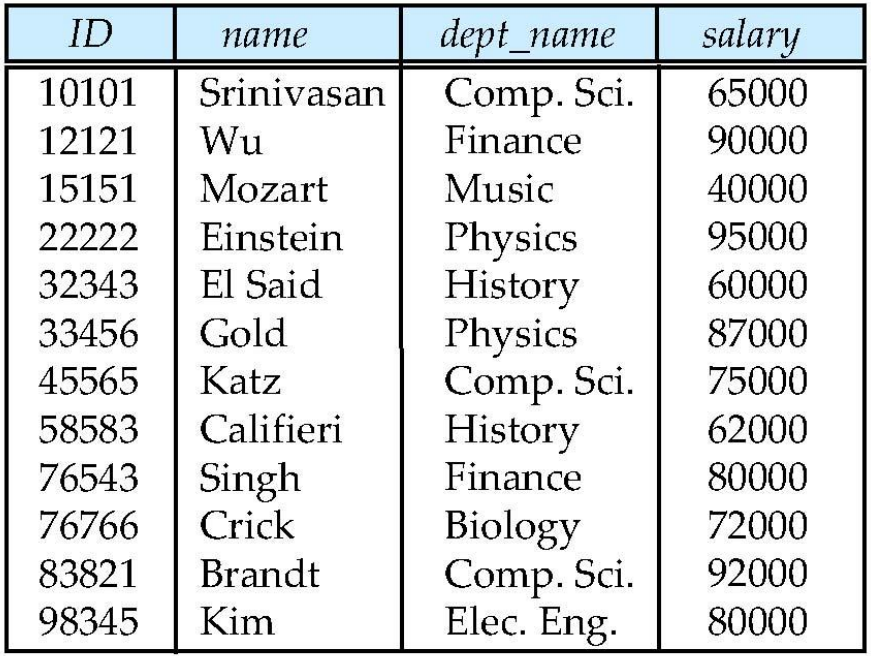
    Thus, a relation instrance r is a set of *n*-tuples $(a_1, a_2, \ldots, a_n)$ where each $a_i \in D_i$
  - **relation** vs. table
    - The current values (**relation instance**) of a relation are specified by a table
    - An element ***t*** of ***r*** is a *tuple*, represented by a *row* in a table

Relations are Unordered
关系是无序的

- Order of tuples is irrelevant (tuples may be stored in an arbitrary order)
- Example: *instructor* relation with unordered tuples



ID	*name*	*dept_name*	*salary*
22222	Einstein	Physics	95000
12121	Wu	Finance	90000
32343	El Said	History	60000
45565	Katz	Comp. Sci.	75000
98345	Kim	Elec. Eng.	80000
76766	Crick	Biology	72000
10101	Srinivasan	Comp. Sci.	65000
58583	Califieri	History	62000
83821	Brandt	Comp. Sci.	92000
15151	Mozart	Music	40000
33456	Gold	Physics	87000
76543	Singh	Finance	80000



ID	*name*	*dept_name*	*salary*
10101	Srinivasan	Comp. Sci.	65000
12121	Wu	Finance	90000
15151	Mozart	Music	40000
22222	Einstein	Physics	95000
32343	El Said	History	60000
33456	Gold	Physics	87000
45565	Katz	Comp. Sci.	75000
58583	Califieri	History	62000
76543	Singh	Finance	80000
76766	Crick	Biology	72000
83821	Brandt	Comp. Sci.	92000
98345	Kim	Elec. Eng.	80000

Database

- A database consists of multiple relations
- 例:Information about an enterprise is broken up into parts

  instructor
  student
  advisor

- Bad design:

  univ (*instructor-ID, name, dept_name, salary, student_Id*, ..)

results in

- repetition of information (e.g., two students have the same instructor)-信息重复
- the need for null values (e.g., represent an student with no advisor)-要使用大量的null值

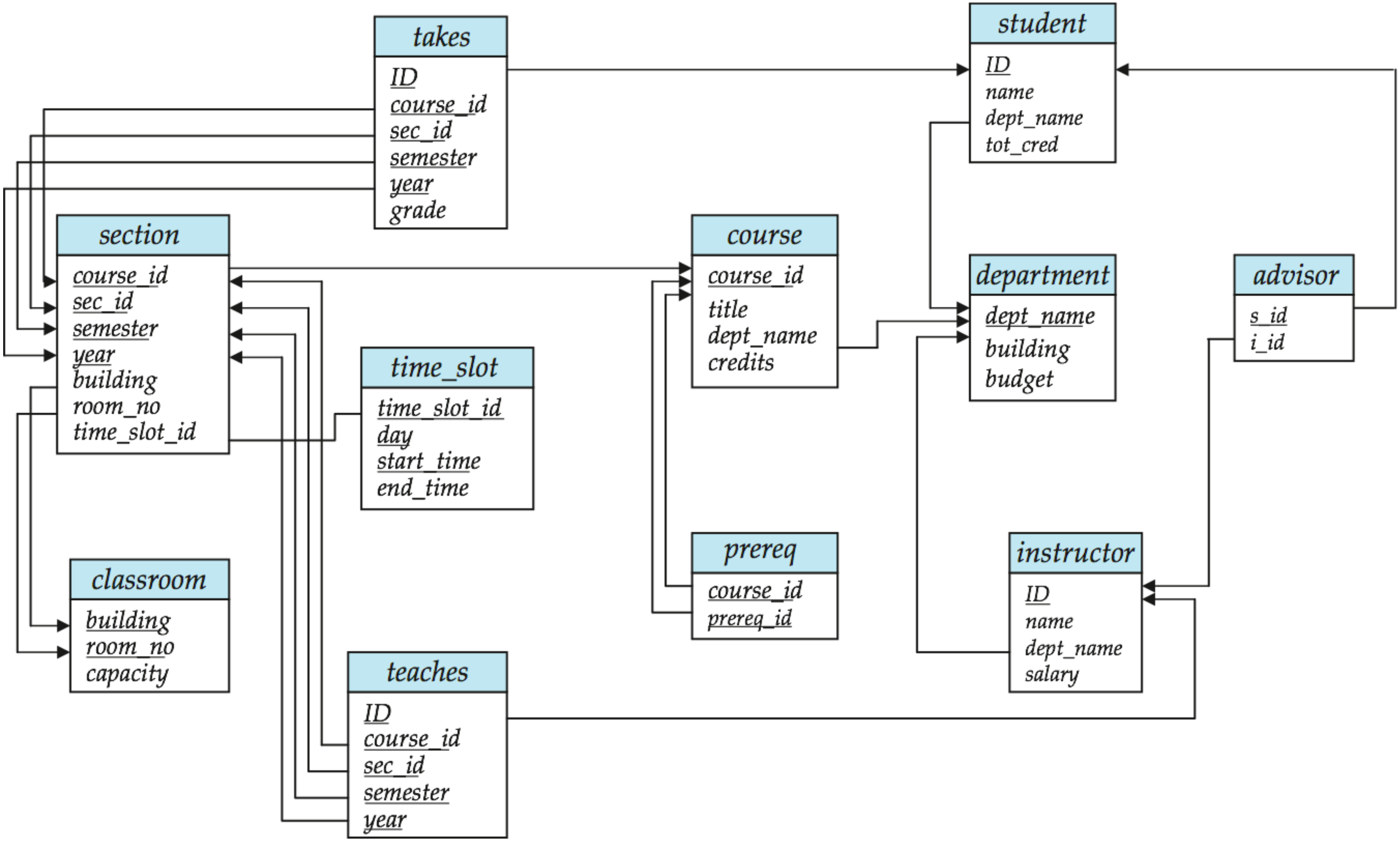
- Normalization theory (Chapter 7) deals with how to design “good” relational schemas

Keys

- Let K $\subseteq$ R， 一组属性
- *K* is a **superkey（超键）** of *R* if values for *K* are sufficient to identify a unique tuple of each possible relation *r(R)*
  - Example: {*ID*} and {ID,name} are both superkeys of *instructor.*
- Superkey *K* is a **candidate key（侯选键）** if *K* is minimal（即没有多余的属性）

  Example: {*ID*} is a candidate key for *Instructor*
- One of the candidate keys is selected to be the **primary key（主键）**.
  - which one? 根据应用语义
- **Foreign key（外键）** constraint: Value in one relation must appear in another 外键值是另一个关系的主键值
  - **Referencing** relation（引用关系，即外键所在的关系）
  - **Referenced** relation(被引有关系，即被引用的主键所在的那个关系)

Schema Diagram（模式图）for University Database



takes
ID
course_id
sec_id
semester
year
grade

student
ID
name
dept_name
tot_cred

section
course_id
sec_id
semester
year
building
room_no
time_slot_id

course
course_id
title
dept_name
credits

department
dept_name
building
budget

advisor
s_id
i_id

time_slot
time_slot_id
day
start_time
end_time

classroom
building
room_no
capacity

prereq
course_id
prereq_id

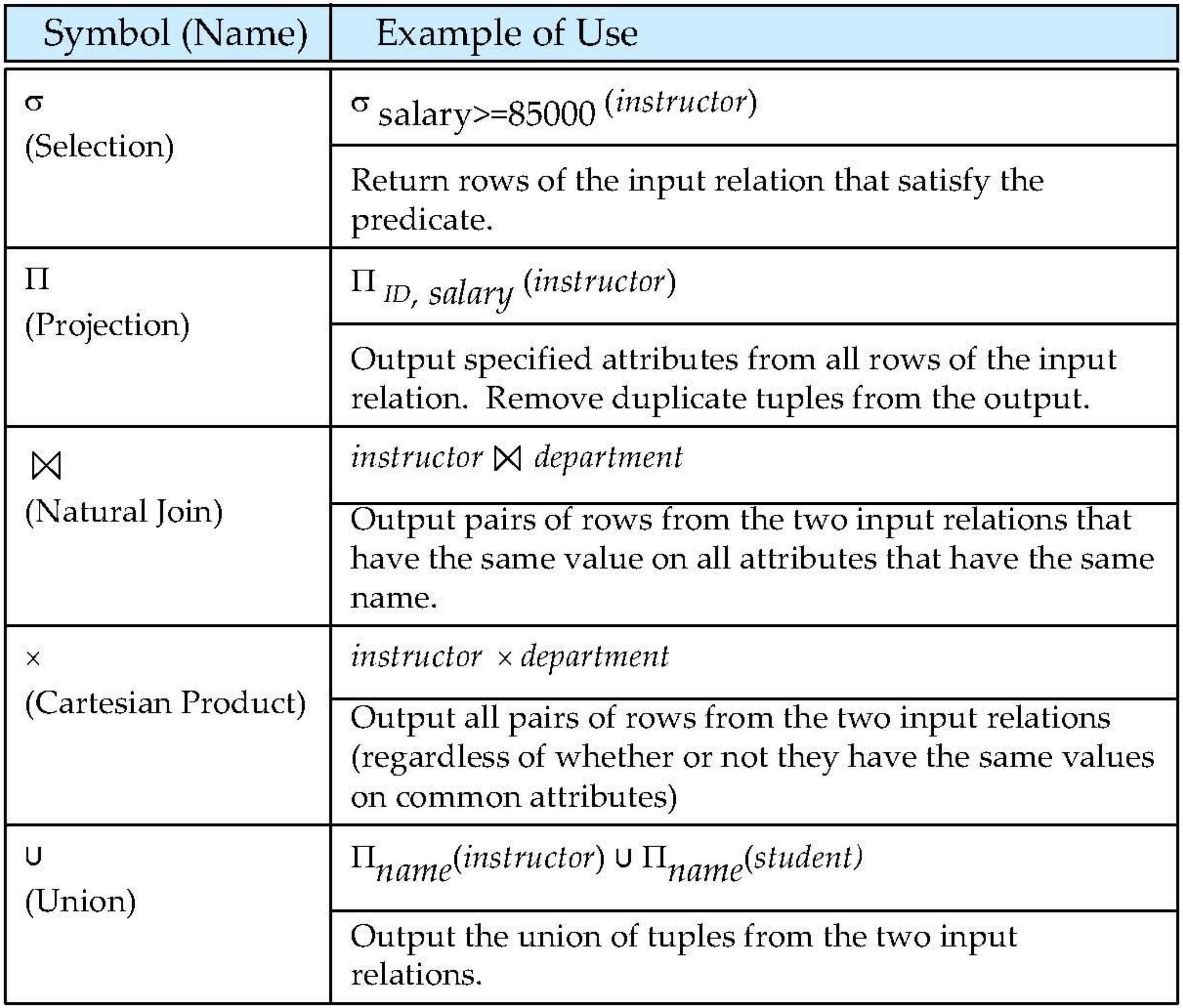
instructor
ID
name
dept_name
salary

teaches
ID
course_id
sec_id
semester
year

Relational Query Languages

- Procedural vs.non-procedural, or declarative
  - Query languages used in practice include elements of both the procedural and the nonprocedural approaches.such as SQL.
- “Pure” languages:
  - Relational algebra 关系代数
  - Tuple relational calculus 元组关系演算
  - Domain relational calculus 域关系演算

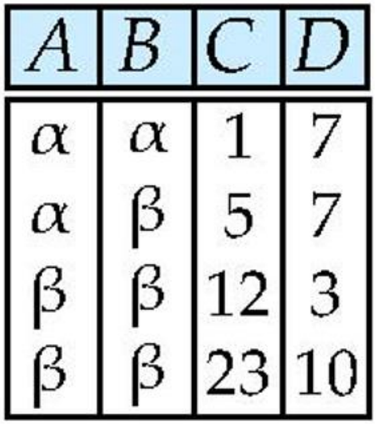
Relational operators



Symbol (Name)	Example of Use
σ (Selection)	$\sigma_{\text{salary>=85000}}(\textit{instructor})$
	Return rows of the input relation that satisfy the predicate.
Π (Projection)	$\Pi_{\textit{ID, salary}}(\textit{instructor})$
	Output specified attributes from all rows of the input relation. Remove duplicate tuples from the output.
⋈ (Natural Join)	$\textit{instructor} \bowtie \textit{department}$
	Output pairs of rows from the two input relations that have the same value on all attributes that have the same name.
× (Cartesian Product)	$\textit{instructor} \times \textit{department}$
	Output all pairs of rows from the two input relations (regardless of whether or not they have the same values on common attributes)
∪ (Union)	$\Pi_{\textit{name}}(\textit{instructor}) \cup \Pi_{\textit{name}}(\textit{student})$
	Output the union of tuples from the two input relations.

Selection of tuples-选择运算

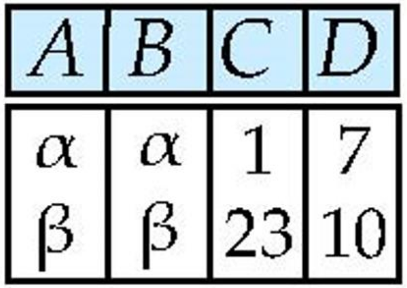
- Relation r



A	B	C	D
α	α	1	7
α	β	5	7
β	β	12	3
β	β	23	10

- Select tuples with A=B and D > 5

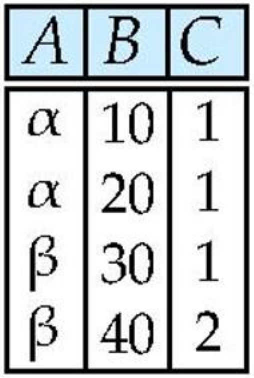
$$\sigma_{A=B \text{ and } D > 5} (r)$$



A	B	C	D
α	α	1	7
β	β	23	10

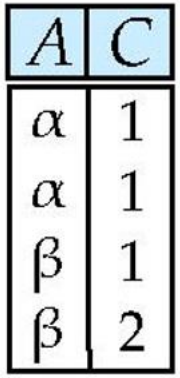
Selection of Columns (Attributes)

- Relation *r*:



A	B	C
α	10	1
α	20	1
β	30	1
β	40	2

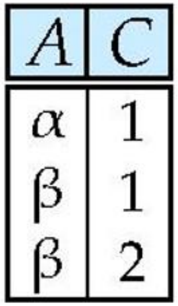
- Select A and C
  - Projection投影运算
  - $\Pi_{A,\ C}\ (r)$



A	C
α	1
α	1
β	1
β	2

消除重复

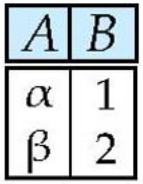
=



A	C
α	1
β	1
β	2

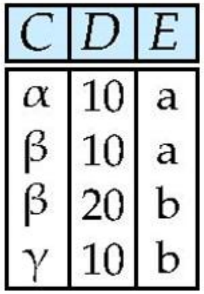
Joining two relations – Cartesian Product 笛卡尔积、叉积

- Relations *r, s*:



A	*B*
α	1
β	2

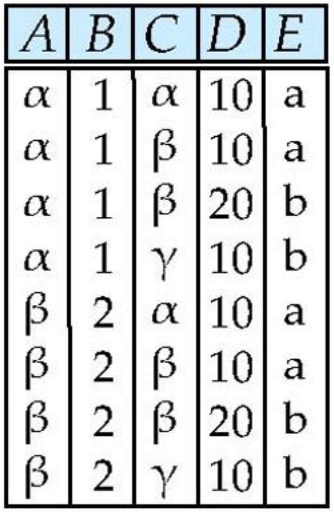
r



C	*D*	*E*
α	10	a
β	10	a
β	20	b
γ	10	b

s

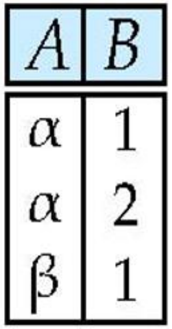
- *r* x *s*:



A	*B*	*C*	*D*	*E*
α	1	α	10	a
α	1	β	10	a
α	1	β	20	b
α	1	γ	10	b
β	2	α	10	a
β	2	β	10	a
β	2	β	20	b
β	2	γ	10	b

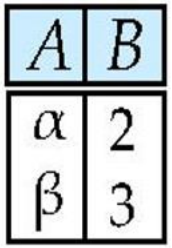
Union of two relations
并运算

- Relations *r, s:*



A	B
α	1
α	2
β	1

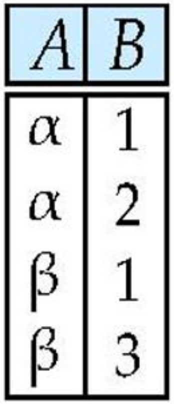
r



A	B
α	2
β	3

s

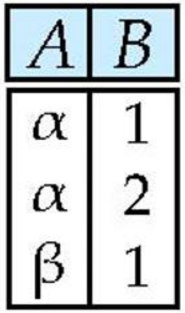
- r ∪ s:



A	B
α	1
α	2
β	1
β	3

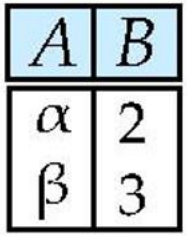
Set difference of two relations
集合差运算

- Relations *r*, *s*:



A	B
α	1
α	2
β	1

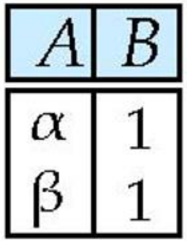
r



A	B
α	2
β	3

s

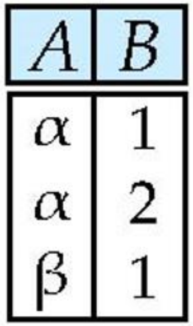
- *r* – *s*:



A	B
α	1
β	1

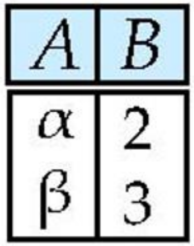
Set Intersection of two relations 交运算

- Relation *r*, *s*:



A	*B*
α	1
α	2
β	1

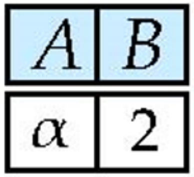
r



A	*B*
α	2
β	3

s

- $r \cap s$



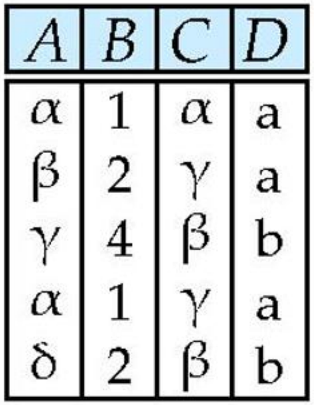
A	*B*
α	2

Joining two relations – Natural Join 自然连接运算

- Let *r* and *s* be relations on schemas *R* and *S* respectively. Then, the "natural join" of relations *r* and *s* is a relation on schema $R \cup S$ obtained as follows:
    - Consider each pair of tuples t_r from *r* and t_s from *s*.
    - If t_r and t_s have the same value on each of the attributes in $R \cap S$, add a tuple *t* to the result, where
        - *t* has the same value as t_r on *r*
        - *t* has the same value as t_s on *s*

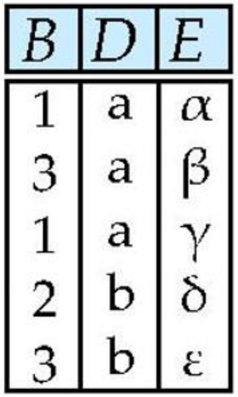
Natural Join Example

- Relations r, s:



A	B	C	D
α	1	α	a
β	2	γ	a
γ	4	β	b
α	1	γ	a
δ	2	β	b

r

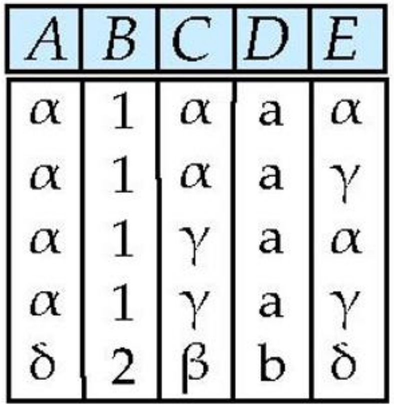


B	D	E
1	a	α
3	a	β
1	a	γ
2	b	δ
3	b	ε

s

- Natural Join 自然连接

  r ⋈ s



A	B	C	D	E
α	1	α	a	α
α	1	α	a	γ
α	1	γ	a	α
α	1	γ	a	γ
δ	2	β	b	δ

1、先计算r x s
2、只保留同名属性值相同的元组
3、同名属性只保留一个副本

总结

本课学习了关系数据模型的基础知识，包括：

关系的模式与关系的实例

关系的超键、侯选键、主键和外键

图形化表示数据库模式的模式图

关系查询语言和关系代数表示查询的基本方法

End of Chapter 2

Database System Concepts, 6th Ed.