DATABASE SYSTEM PRINCIPLE – RELATIONAL MODEL

李旭东 **Li-Xudong**

LEEXUDONG@NANKAI.EDU.CN

NANKAI UNIVERSITY

OBJECTIVES

- Structure of Relational Databases
- Database Schema
- Keys
- Schema Diagrams
- Relational Query Languages
- Relational Operations

STRUCTURE OF RELATIONAL DATABASES

- A relational database consists of a collection of **tables**, each of which is assigned a unique name

attributes
(or columns)

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

tuples
(or rows)

Table: instructor

STRUCTURE OF RELATIONAL DATABASES

- Relational database
  - consists of a collection of tables, each of which is assigned a unique name
  - Table: **Relation**
  - Row: **Tuple**
  - Column: **attribute**
    - Domain(域)*: a set of permitted values for one attribute

STRUCTURE OF RELATIONAL DATABASES – ATTRIBUTE TYPES

- Domain
  - a set of allowed values for one attribute
- Attribute values are (normally) required to be atomic; that is, indivisible(不可再分)
- The special value **null** is a member of every domain. Indicated that the value is “unknown”
- The null value causes complications in the definition of many operations

RELATION, RELATION SCHEMA AND INSTANCE – RELATION

- $A_1, A_2, \ldots, A_n$ are *attributes*
- given sets $D_1, D_2, \ldots. D_n$ a **relation *r*** is a subset of

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

- *Thus, a **relation** is a set of n-tuples (a1, a2, …, an) where each* $a_i \in D_i$
- An element ***t*** of ***r*** is a ***tuple***, represented by a *row* in a table

RELATION, RELATION SCHEMA AND INSTANCE – 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(关系模式)*

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

RELATION SCHEMA: CASES

- student (ID , name, dept_name, tot_cred)
- advisor (s_id, i_id)
- takes (ID , course_id, sec_id, semester, year, grade)
- classroom (building, room_number, capacity)
- time_slot (time_slot_id, day, start_time, end_time)

RELATION, RELATION SCHEMA AND INSTANCE – RELATION INSTANCE

- Relation Instance关系实例
  - A specific instance of a relation, i.e., containing a specific set of rows

RELATION INSTANCE: CASE

- Course table

course_id	*title*	*dept_name*	*credits*
BIO-101	Intro. to Biology	Biology	4
BIO-301	Genetics	Biology	4
BIO-399	Computational Biology	Biology	3
CS-101	Intro. to Computer Science	Comp. Sci.	4
CS-190	Game Design	Comp. Sci.	4
CS-315	Robotics	Comp. Sci.	3
CS-319	Image Processing	Comp. Sci.	3
CS-347	Database System Concepts	Comp. Sci.	3
EE-181	Intro. to Digital Systems	Elec. Eng.	3
FIN-201	Investment Banking	Finance	3
HIS-351	World History	History	3
MU-199	Music Video Production	Music	3
PHY-101	Physical Principles	Physics	4

RELATION INSTANCE: CASE

- Prereq table

course_id	*prereq_id*
BIO-301	BIO-101
BIO-399	BIO-101
CS-190	CS-101
CS-315	CS-101
CS-319	CS-101
CS-347	CS-101
EE-181	PHY-101

RELATIONS ARE UNORDERED

- 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

KEYS码/键

- Let $K \subseteq R$
- *K* is a **superkey 超码** 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 候选码** if *K* is minimal

  Example: {*ID*} is a candidate key for *Instructor*
- One of the candidate keys is selected to be the **primary key主码**.*

WHICH ONE IS PRIMARY KEY?

- If two candidate keys
  - {ID}
  - {name, dept_name}

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

Table: instructor

KEYS
– PRIMARY KEY

- Primary keys
  - It denote a candidate key that is chosen by the database designer as the principal means of identifying tuples within a relation
  - It must be chosen with care
    - Example: the name of a person is obviously not sufficient

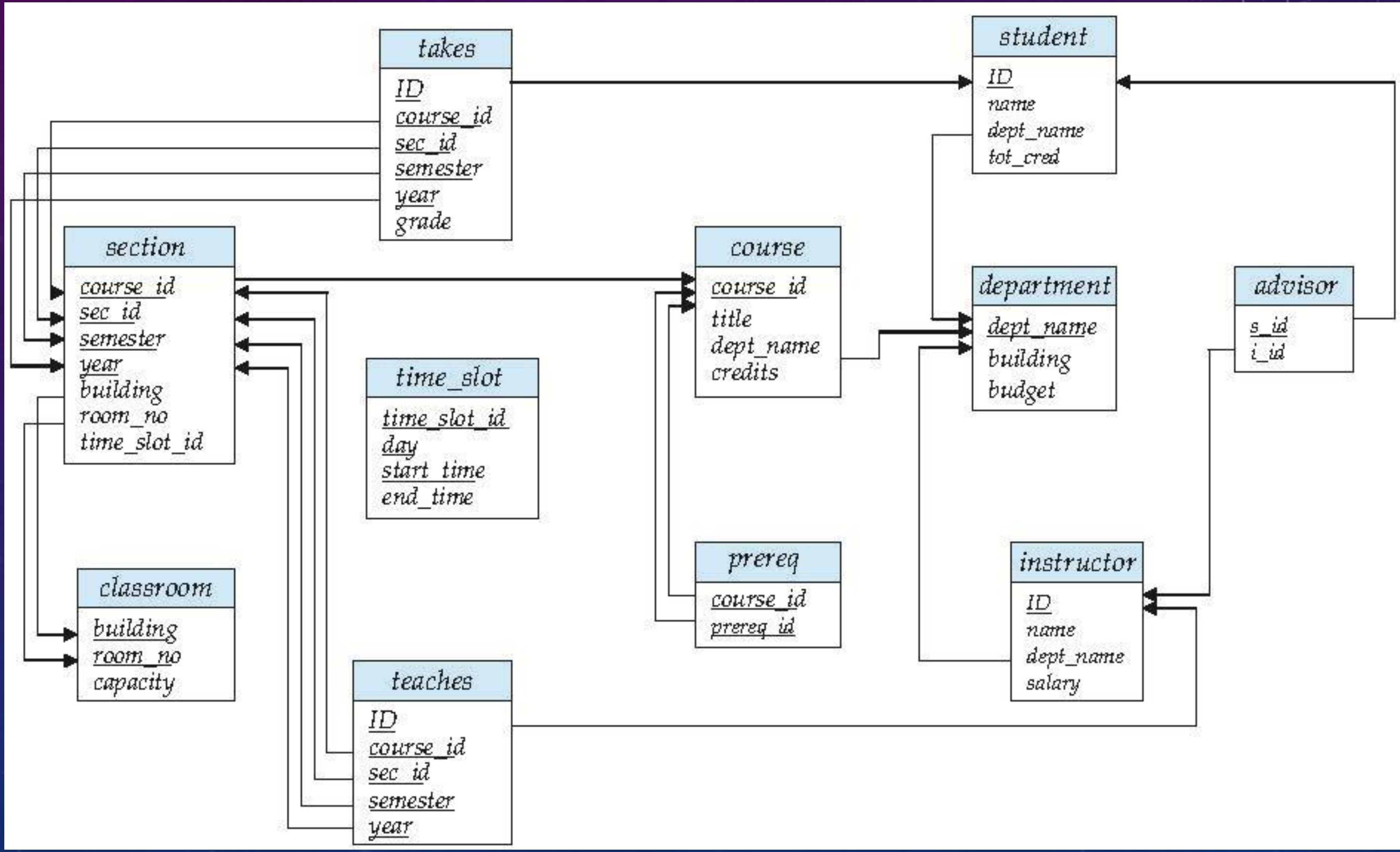
KEYS
– FOREIGN KEY

- **Foreign key外码**
  - the key constraint: Value in one relation must appear in another
  - **Referencing** relation参照关系
  - **Referenced** relation被参照关系
  - Example – *dept_name* in *instructor* is a foreign key from *instructor* referencing *department*
- *Referential integrity constraint参照完整性约束*

SCHEMA DIAGRAM

- Schema diagram模式图
  - A database schema, along with primary key and foreign key dependencies, can be depicted by schema diagrams
  - Each relation appears as a box
    - with the relation name at the top in blue
    - And the attribute listed inside the box
    - Primary key attributed are shown underlined
    - Foreign key dependencies appear as arrows from the foreign key attributes of the referencing relation to the primary key of the referenced relation

SCHEMA DIAGRAM: CASE – UNIVERSITY DATABASE



SCHEMA OF THE UNIVERSITY DATABASE

classroom(building, room_number, capacity)
department(dept_name, building, budget)
course(course_id, title, dept_name, credits)
instructor(ID, name, dept_name, salary)
section(course_id, sec_id, semester, year, building, room_number, time_slot_id)
teaches(ID, course_id, sec_id, semester, year)
student(ID, name, dept_name, tot_cred)
takes(ID, course_id, sec_id, semester, year, grade)
advisor(s_ID, i_ID)
time_slot(time_slot_id, day, start_time, end_time)
prereq(course_id, prereq_id)

RELATIONAL QUERY LANGUAGES

- Query language
  - A language in which a user requests information from the database
- Categories
  - Procedural language 过程式语言
  - Non procedural language: declarative

RELATIONAL QUERY LANGUAGES

- “Pure” languages:
  - Relational algebra关系代数
    - （belong to procedural language）
  - Tuple relational calculus元祖关系演算
  - Domain relational calculus域关系演算
- The above 3 pure languages are equivalent in computing power
  - Relational operations: 6 basic operations*

RELATIONAL OPERATIONS

- Relational Operations
  - Select specific tuples fro a single relation
  - Select certain attributed from a relation
  - Combine two relations by merging pairs of tuples
  - A set union of two similarly structured tables

RELATIONAL OPERATIONS – SELECTION: SELECTION OF ROWS (TUPLES)

- Relation r

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

- $\sigma_{A=B \wedge D > 5}(r)$

A	B	C	D
α	α	1	7
β	β	23	10

RELATIONAL OPERATIONS – SELECTION: SELECTION OF COLUMNS (ATTRIBUTES)

- Relation *r*:

A	B	C
α	10	1
α	20	1
β	30	1
β	40	2

$\Pi_{A,C}(r)$

A	C
α	1
α	1
β	1
β	2

=

A	C
α	1
β	1
β	2

RELATIONAL OPERATIONS
– UNION OF TWO RELATIONS

- 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

RELATIONAL OPERATIONS
– SET DIFFERENCE OF TWO RELATIONS

- 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

RELATIONAL OPERATIONS – SET INTERSECTION OF TWO RELATIONS

- 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

Note: $r \cap s = r - (r - s)$

RELATIONAL OPERATIONS – JOIN: CARTESIAN-PRODUCT笛卡尔积

- 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

Note: Regardless of whether or not they have the same values on common attributed

CARTESIAN 笛卡尔

- a French philosopher, mathematician, and scientist
- 1596~1650
- I think; therefore I am

RELATIONAL OPERATIONS – JOIN: CARTESIAN-PRODUCT->NAMING ISSUE

- 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

RELATIONAL OPERATIONS – RENAMING A TABLE

- Allows us to refer to a relation, (say E) by more than one name.

  $\rho_x(E)$

  returns the expression *E* under the name *X*

- Relations *r*

A	B
α	1
β	2

r

- $r \times \rho_s(r)$

r.A	r.B	s.A	s.B
α	1	α	1
α	1	β	2
β	2	α	1
β	2	β	2

RELATIONAL OPERATIONS – JOIN: NATURAL JOIN自然连接

- 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

RELATIONAL OPERATIONS – JOIN: NATURAL JOIN -> CASES

- Relations r, s:

- Natural Join
  - r ⋈ s

$$\Pi_{A,\ r.B,\ C,\ r.D,\ E}(\sigma_{r.B = s.B \wedge r.D = s.D}\ (r \times s)))$$

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

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

NOTES ABOUT RELATIONAL LANGUAGES

- Each Query input is a table (or set of tables)
- Each query output is a table.
- All data in the output table appears in one of the input tables
- Relational Algebra is not Turning complete
- Can we compute:
  - SUM, AVG, MAX, MIN

RELATIONAL OPERATIONS

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

SUMMARY

- Structure of Relational Databases
- Database Schema
- Keys
- Schema Diagrams
- Relational Query Languages
- Relational Operations

Q&A?

THANKS！

leexudong@nankai.edu.cn