國立陽明交通大學

NATIONAL YANG MING CHIAO TUNG UNIVERSITY

Biological Databases: Theories and Practice

430032

Relational Model

Instructor: Prof. LEE, Tzong-Yi (李宗夷)
Email: leetzongyi@nycu.edu.ctw

Professor
Institute of Bioinformatics and Systems Biology,
National Yang Ming Chiao Tung University

Lecture outline

1. Structure of Relational Databases
2. Database Schema
3. Keys
4. Schema Diagrams
5. Relational Query Languages
6. Relational Operations

Structure of Relational Databases

- A relational database consists of a collection of **tables**, each of which is assigned a unique name.
- Given the *instructor* table, each **row** (tuple) contains four **columns** (attributes): *ID*, *name*, *dept_name*, and *salary*.
- Note that each instructor can be identified by the value of the column *ID*.
  - This indicates that the attribute *ID* can be used to identify an unique instructor.

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)

Another example of Relational Tables

Table/ Relation/ Entity/ Object

Columns/ Attributes/ Characteristics

Rows/ Tuples/ Records/ Instances/ Occurrences

Table name: COURSE
Primary key: CRS_CODE
Foreign key: none

Database name: Ch03_TinyCollege

CRS_CODE	DEPT_CODE	CRS_DESCRIPTION	CRS_CREDIT
ACCT-211	ACCT	Accounting I	3
ACCT-212	ACCT	Accounting II	3
CIS-220	CIS	Intro. to Microcomputing	3
CIS-420	CIS	Database Design and Implementation	4
QM-261	CIS	Intro. to Statistics	3
QM-362	CIS	Statistical Applications	4

Table name: CLASS
Primary key: CLASS_CODE
Foreign key: CRS_CODE

CLASS_CODE	CRS_CODE	CLASS_SECTION	CLASS_TIME	CLASS_ROOM	PROF_NUM
10012	ACCT-211	1	MWF 8:00-8:50 a.m.	BUS311	105
10013	ACCT-211	2	MWF 9:00-9:50 a.m.	BUS200	105
10014	ACCT-211	3	TTh 2:30-3:45 p.m.	BUS252	342
10015	ACCT-212	1	MWF 10:00-10:50 a.m.	BUS311	301
10016	ACCT-212	2	Th 6:00-8:40 p.m.	BUS252	301
10017	CIS-220	1	MWF 9:00-9:50 a.m.	KLR209	228
10018	CIS-220	2	MWF 9:00-9:50 a.m.	KLR211	114
10019	CIS-220	3	MWF 10:00-10:50 a.m.	KLR209	228
10020	CIS-420	1	W 6:00-8:40 p.m.	KLR209	162
10021	QM-261	1	MWF 8:00-8:50 a.m.	KLR200	114
10022	QM-261	2	TTh 1:00-2:15 p.m.	KLR200	114
10023	QM-362	1	MWF 11:00-11:50 a.m.	KLR200	162
10024	QM-362	2	TTh 2:30-3:45 p.m.	KLR200	162

Attribute Types

- The set of allowed values for each attribute is called the **domain** of the attribute.
  - The domain of the ***salary*** attribute of the ***instructor*** relation is the set of all possible salary values.
- Attribute values are (normally) required to be **atomic**; that is, indivisible.
  - Suppose the table ***instructor*** had an attribute ***phone_number***, which contains multiple phone numbers. Then, the domain of ***phone_number*** is not atomic.
- The **null** value is a special value that signifies that the value is **unknown** or does **not exist**.
  - The special value ***null*** is a member of every domain
  - The null value causes complications in the definition of many operations

Relation Schema and Instance

- $R = (A_1, A_2, \ldots, A_n)$ is a *relation schema, where* $A_1, A_2, \ldots, A_n$ are *attributes*
  - Example: *instructor* = (*ID, name, dept_name, salary*)

- Formally, given domain 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 ***r*** is a set of *m*-tuples $(a_1, a_2, \ldots, a_n)$ where each $a_i \in D_i$

- 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.

Data in a Relation 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

Database

- A database consists of multiple relations
- Information about an enterprise (ex. an university) is broken up into parts

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

  student = (*student_ID, name, dept_name, total_credit*)

  advisor = (*student_ID, instructor_ID*)

- Bad design:

  university = (*instructor_ID, name, dept_name, salary, student_ID*, *student_name*, ..)

  results in

  - **repetition of information** (e.g., two students have the same instructor)
  - **the need for null values** (e.g., a student without the information of advisor)
- **Normalization theory** (Chapter 8) deals with how to design "good" relational schemas

A Relation for Describing the Class Information

- Each course in a university may be offered multiple times, across different semesters, or even within a semester.

section = (course_id, sec_id, semester, year, building, room_number, time_slot_id)

course_id	*sec_id*	*semester*	*year*	*building*	*room_number*	*time_slot_id*
BIO-101	1	Summer	2009	Painter	514	B
BIO-301	1	Summer	2010	Painter	514	A
CS-101	1	Fall	2009	Packard	101	H
CS-101	1	Spring	2010	Packard	101	F
CS-190	1	Spring	2009	Taylor	3128	E
CS-190	2	Spring	2009	Taylor	3128	A
CS-315	1	Spring	2010	Watson	120	D
CS-319	1	Spring	2010	Watson	100	B
CS-319	2	Spring	2010	Taylor	3128	C
CS-347	1	Fall	2009	Taylor	3128	A
EE-181	1	Spring	2009	Taylor	3128	C
FIN-201	1	Spring	2010	Packard	101	B
HIS-351	1	Spring	2010	Painter	514	C
MU-199	1	Spring	2010	Packard	101	D
PHY-101	1	Fall	2009	Watson	100	A

A Relation to Describe the Association between Instructors and Classes They Teach

- The relation schema to describe this association is

teaches = (ID, course_id, sec_id, semester, year)

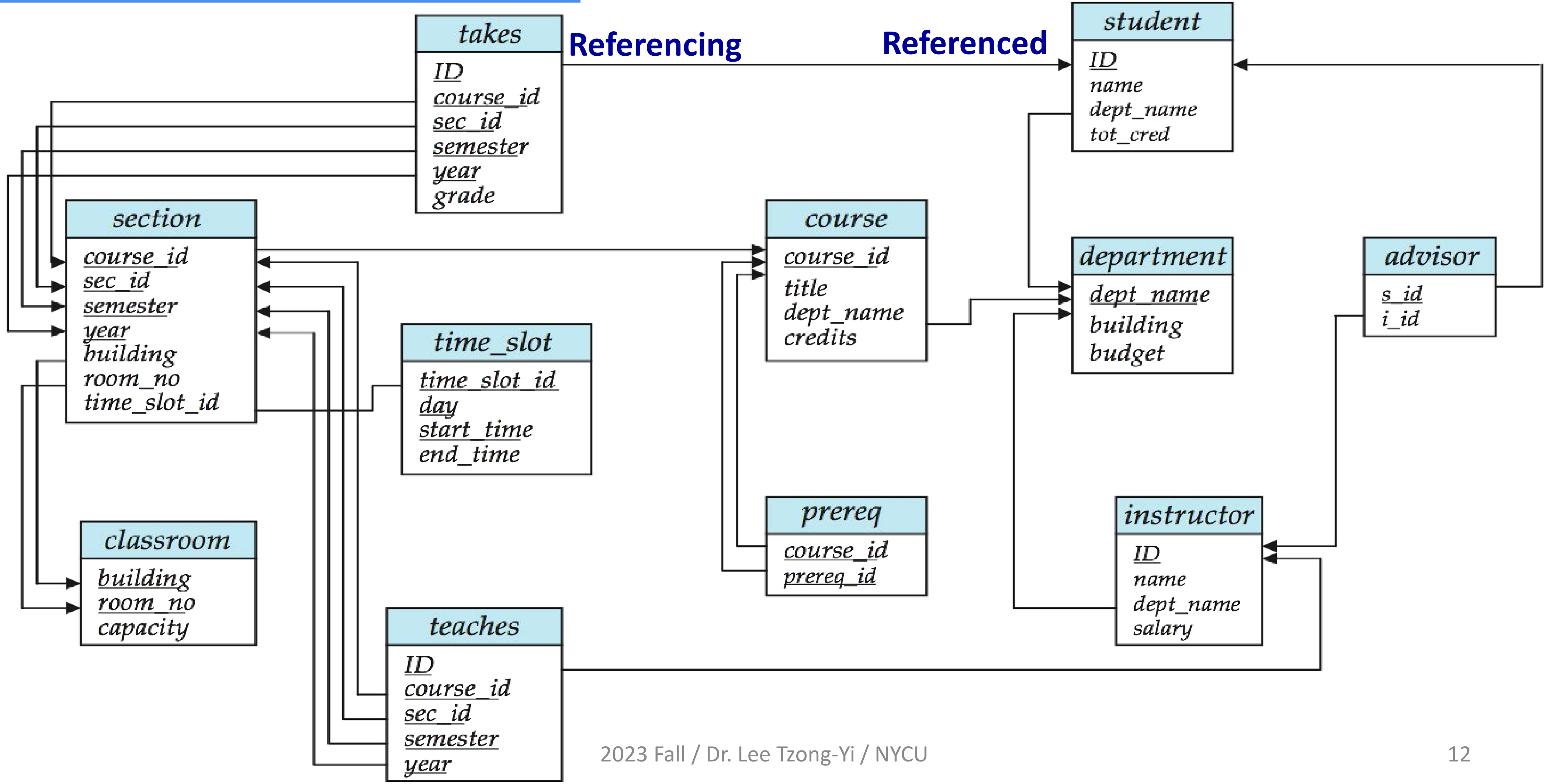
ID	*course_id*	*sec_id*	*semester*	*year*
10101	CS-101	1	Fall	2009
10101	CS-315	1	Spring	2010
10101	CS-347	1	Fall	2009
12121	FIN-201	1	Spring	2010
15151	MU-199	1	Spring	2010
22222	PHY-101	1	Fall	2009
32343	HIS-351	1	Spring	2010
45565	CS-101	1	Spring	2010
45565	CS-319	1	Spring	2010
76766	BIO-101	1	Summer	2009
76766	BIO-301	1	Summer	2010
83821	CS-190	1	Spring	2009
83821	CS-190	2	Spring	2009
83821	CS-319	2	Spring	2010
98345	EE-181	1	Spring	2009

Keys

- Let **R** denotes the set of attributes and **K** $\subseteq$ **R**:
  - *K, which is subset of R,* is a **superkey** if values for *K* are sufficient to identify a unique tuple in the relation.
  - Example: {*ID*} and {ID, name} are both superkeys of *instructor.*
- Superkey ***K*** is a **candidate key** if ***K* is minimal**
  Example: {*ID*} is a candidate key for *Instructor*
- One of the candidate keys is selected to be the **primary key**.
- **Foreign key** constraint: value in one relation must appear in another relation
  - **Referencing** relation
  - **Referenced** relation

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

Schema Diagram for University Database



Relational Query Languages

- A **query language** is a language in which a user requests information from the database.
  - **Procedural language**: the user instructs the system to perform a sequence of operations on the database.
  - **Non-procedural language** (declarative): the user describes the desired information without giving a specific procedure.
- "Pure" languages:
  - Relational algebra
  - Tuple relational calculus
  - Domain relational calculus
- Relational operators

Selection of tuples

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

A	C
α	1
α	1
β	1
β	2

=

A	C
α	1
β	1
β	2

Joining two relations – Cartesian Product

- 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

- Relations *r, s:*

A	B
α	1
α	2
β	1

r

A	B
α	2
β	3

s

- $r \cup s$:

A	B
α	1
α	2
β	1
β	3

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

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

Joining two relations – Natural Join

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

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

t_r

r

t_s

B	D	E
1	a	α
3	a	β
1	a	γ
2	b	δ
3	b	ε

s

R ∪ S = (A, B, C, D, E)

R ∩ S = (B, D)

- Natural Join
  - 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

If t_r and t_s have the same value on each of the attributes in ***R ∩ S***

Operators of Relational Algebra

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)	$\Pi_{ID,\ salary}(\textit{instructor})$ Output specified attributes from all rows of the input relation. Remove duplicate tuples from the output.
$\bowtie$ (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.
$\times$ (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)
$\cup$ (Union)	$\Pi_{name}(\textit{instructor}) \cup \Pi_{name}(\textit{student})$ Output the union of tuples from the two input relations.

Thank you and any questions?