22AIE303: Database Management Systems

Lecture 3

Introduction to Relational Model

Relational Model

A database is a collection of one or more *relations*, where each relation is a table with rows and columns

The relational model uses a collection of tables to represent both entity and relationships among

Relational Model

A database is a collection of one or more *relations*, where each relation is a table with rows and columns

The relational model uses a collection of tables to represent both entity and relationships among

Student

S-ID	S-name
1	aa
2	bb

Course

C-ID	C-name
C1	DBMS
C2	SP

Enroll

E-ID	S-ID	C-ID
E1	1	DBMS
E2	1	SP
E3	2	SP

Relational Model

Attribute / field are used to describe relations (tables)

OR columns of relations are attributes.

Student

S-ID	S-name
1	aa
2	bb

Course

C-ID	C-name
C1	DBMS
C2	SP

Enroll

E-ID	S-ID	C-ID
E1	1	DBMS
E2	1	SP
E3	2	SP

Relational Model

A row in a relation: **tuple** or **record**

Student

S-ID	S-name
1	aa
2	bb

Course

C-ID	C-name
C1	DBMS
C2	SP

Enroll

E-ID	S-ID	C-ID
E1	1	DBMS
E2	1	SP
E3	2	SP

Relational Model

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)

Relational Model

Domain: Set of allowed values for each attribute / A unique set of values permitted for attribute

- **Roll:** Alphanumeric string
- **Name:** string
- **DOB:** Date

Attributes are **atomics** in nature
Atomics: can’t divided into smaller part (indivisible)

DOB: year-date-month (atomic value)

Relational Model

Domain: Set of allowed values for each attribute / / A unique set of values permitted for attribute

- **Roll:** Alphanumeric string
- **First name:** string
- **Last name:** string
- **DOB:** Date

Attributes are **atomics** in nature
Atomics: can’t divided into smaller part (indivisible)

DOB: year-date-month (atomic value)

Relational Model

8.5. Date/Time Types

8.5.1. Date/Time Input
8.5.2. Date/Time Output
8.5.3. Time Zones
8.5.4. Interval Input
8.5.5. Interval Output

PostgreSQL supports the full set of SQL date and time types, shown in Table 8.9. The operations av
according to the Gregorian calendar, even in years before that calendar was introduced (see Sectio

Table 8.9. Date/Time Types

Name	Storage Size	Description
`timestamp [ (p) ] [ without time zone ]`	8 bytes	both date and time (no time zone)
`timestamp [ (p) ] with time zone`	8 bytes	both date and time, with time zone
`date`	4 bytes	date (no time of day)
`time [ (p) ] [ without time zone ]`	8 bytes	time of day (no date)
`time [ (p) ] with time zone`	12 bytes	time of day (no date), with time zone
`interval [ fields ] [ (p) ]`	16 bytes	time interval

Relational Model

Domain Constrain: Specifies an important condition that we want each instant of relation (table) to satisfy.

- Unique constraints (like student roll number should be unique)
- Value constrain (like Balance > 0)
- NULL constrain

Relational Model

Domain Constrain: Specifies an important condition that we want each instant of relation (table) to satisfy.

- **Unique constrain** (like student roll number should be unique)
- Balance > 0
- NULL constrain

Unique constrain on S-ID

Student

S-ID	S-name
1	aa
2	bb
1	cc

X

Relational Model

Domain Constrain: Specifies an important condition that we want each instant of relation (table) to satisfy.

- Unique constrain (like student roll number should be unique)
- Balance > 0
- **NULL constrain**

No – NULL constrain on S-Pho

Student

S-ID	S-name	S-Pho
1	aa	9989
2	bb	NULL
1	cc	7676

Relational Model

Domain Constrain: Specifies an important condition that we want each instant of relation (table) to satisfy.

- Unique constrain (like student roll number should be unique)
- Balance > 0
- **NULL constrain**

NULL constrain on S-Pho

Student

S-ID	S-name	S-Pho
1	aa	9989
2	bb	NULL
1	cc	7676

X

Relational Model

Logical Schema:

Relation_name(attribute1, attribute2, ….. attributeN)

student(stident_id, student_name,student_phoneno)

account(account_number, branch_name,balance)

Relational Model

Logical Schema:

Details

Students(*sid:* `string`, *name:* `string`, *login:* `string`, *age:* `integer`, *gpa:* `real`)

Relational Model

$A_1, A_2, A_3, A_4 \ldots\ldots\ldots, A_n$ **---** Attribute

$R(A_1, A_2, A_3, A_4 \ldots\ldots\ldots, A_n)$ $\leftarrow$ Schema

The **relation** is a set of n-tuples.

tuples $(a_{A1}, a_{2A2}, a_{A3}, a_{A4} \ldots\ldots\ldots, a_{An})$ where $a_i \in D_i$

r (relation) table is a **subset of** $D_1 \times D_2 \times D_3 \times D_4 \ldots\ldots \times D_n$

Relational Model

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

Relations are unordered

Order of tuples is irrelevant (tuple may be stored in an arbitrary order)

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

Relational Model

Degree / Arity: Number of attributes in relation

Cardinality: Number of tuples in a relation

sid	*name*	*login*	*age*	*gpa*
53831	Madayan	madayan@music	11	1.8
53832	Guldu	guldu@music	12	2.0
53688	Smith	smith@ee	18	3.2
53650	Smith	smith@math	19	3.8
53666	Jones	jones@cs	18	3.4
50000	Dave	dave@cs	19	3.3

Degree: 5; Cardinality: 6

Relational Model

Key: An attribute or set of attributes whose value can uniquely identify a tuple in a relation.

Let K is a subset of R

$R(A_1, A_2, A_3, A_4 \ldots\ldots\ldots, A_n)$ ← Schema

K is a **superkey** of R if the values for K are sufficient to identify a unique tuple of each possible relation *r*(R).

Superkey: All possible keys of a Relation.

Example: Student(Rno, name, room_no, address)

superkey:

Rno,

(Rno, name)

(Rno, room_no)

(Rno, room_no, address) etc. etc.

Relational Model

Candidate Key: Minimal superkey OR A key those proper subsets it not a key

A key whose proper subset is not a key.

Example: Student (Roll_no, name, father-name)

Candidate key:

Roll no

(name, father-name)

Relational Model

Primary Key: Chosen candidate key for implementation

Example: Student (Roll_no, name, father-name)

Candidate key:

Roll no

(name, father-name)

If I chose “Roll no” to implement the database then this is the primary key.

Relational Model

Surrogate Key / Synthetic key: unique identifier of each entry in the relation, however not derived from the application data (unlike superkey)

Example: serial number

Relational Model

Roll #	First Name	Last Name	DoB	Passport #	Aadhaar #	Department
15CS10026	Lalit	Dubey	27-Mar-1997	L4032464	1728-6174-9239	Computer
16EE30029	Jatin	Chopra	17-Nov-1996	null	3917-1836-3816	Electrical
15EC10016	Smriti	Mongra	23-Dec-1996	G5432849	2045-9271-0914	Electronics
16CE10038	Dipti	Dutta	02-Feb-1997	null	5719-1948-2918	Civil
15CS30021	Ramdin	Minz	10-Jan-1997	X8811623	4928-4927-5924	Computer

Relational Model

Roll #	First Name	Last Name	DoB	Passport #	Aadhaar #	Department
15CS10026	Lalit	Dubey	27-Mar-1997	L4032464	1728-6174-9239	Computer
16EE30029	Jatin	Chopra	17-Nov-1996	null	3917-1836-3816	Electrical
15EC10016	Smriti	Mongra	23-Dec-1996	G5432849	2045-9271-0914	Electronics
16CE10038	Dipti	Dutta	02-Feb-1997	null	5719-1948-2918	Civil
15CS30021	Ramdin	Minz	10-Jan-1997	X8811623	4928-4927-5924	Computer

Super Key: Roll #, {Roll #, DoB} … etc.

Candidate Keys: Roll #, {First Name, Last Name}, Aadhaar #
Passport # cannot be a key. Why?
Null values are allowed for Passport # (a student may not have a passport)

Primary Key: Roll #

Relational Model

Alternative Key: All candidate keys apart from Primary Key.

-

Relational Model

Roll #	First Name	Last Name	DoB	Passport #	Aadhaar #	Department
15CS10026	Lalit	Dubey	27-Mar-1997	L4032464	1728-6174-9239	Computer
16EE30029	Jatin	Chopra	17-Nov-1996	null	3917-1836-3816	Electrical
15EC10016	Smriti	Mongra	23-Dec-1996	G5432849	2045-9271-0914	Electronics
16CE10038	Dipti	Dutta	02-Feb-1997	null	5719-1948-2918	Civil
15CS30021	Ramdin	Minz	10-Jan-1997	X8811623	4928-4927-5924	Computer

Super Key: Roll #, {Roll #, DoB} … etc.

Candidate Keys: Roll #, {First Name, Last Name}, Aadhaar #
Passport # cannot be a key. Why?
Null values are allowed for Passport # (a student may not have a passport)

Primary Key: Roll # ; **Alternative key**: {First Name, Last Name}

Relational Model

Simple Key: Consists of a single key.

Composite Key: {First name, Last name}

- Consists of more than one attribute to uniquely identify an entity occurrence.
- One or more of the attributes, which make up the key, are not simple keys in their own right.

Relational Model

Referential Integrity:

-

Relational Model

Referential Integrity:

We wish to ensure that a value that appears in one relation for a given set of attributes also appears for a certain set of attributes in another relation. This condition called referential integrity.

Relational Model

Foreign key: Value in one relation must appear in another

- **Referencing** relation

· Enrolment: Foreign Keys – Roll #, Course #

- **Referenced** relation

  Students, Courses

A **compound key** consists of *more than one attribute* to uniquely identify an entity occurrence.

Each attribute, which makes up the key, is a simple key in its own right.

Students

<u>Roll #</u>	First Name	Last Name	DoB	Passport #	Aadhaar #	Department

Courses

<u>Course #</u>	Course Name	Credits	L-T-P	Department

Enrolment

<u>Roll #</u>	<u>Course #</u>	Instructor ID

Schema diagram

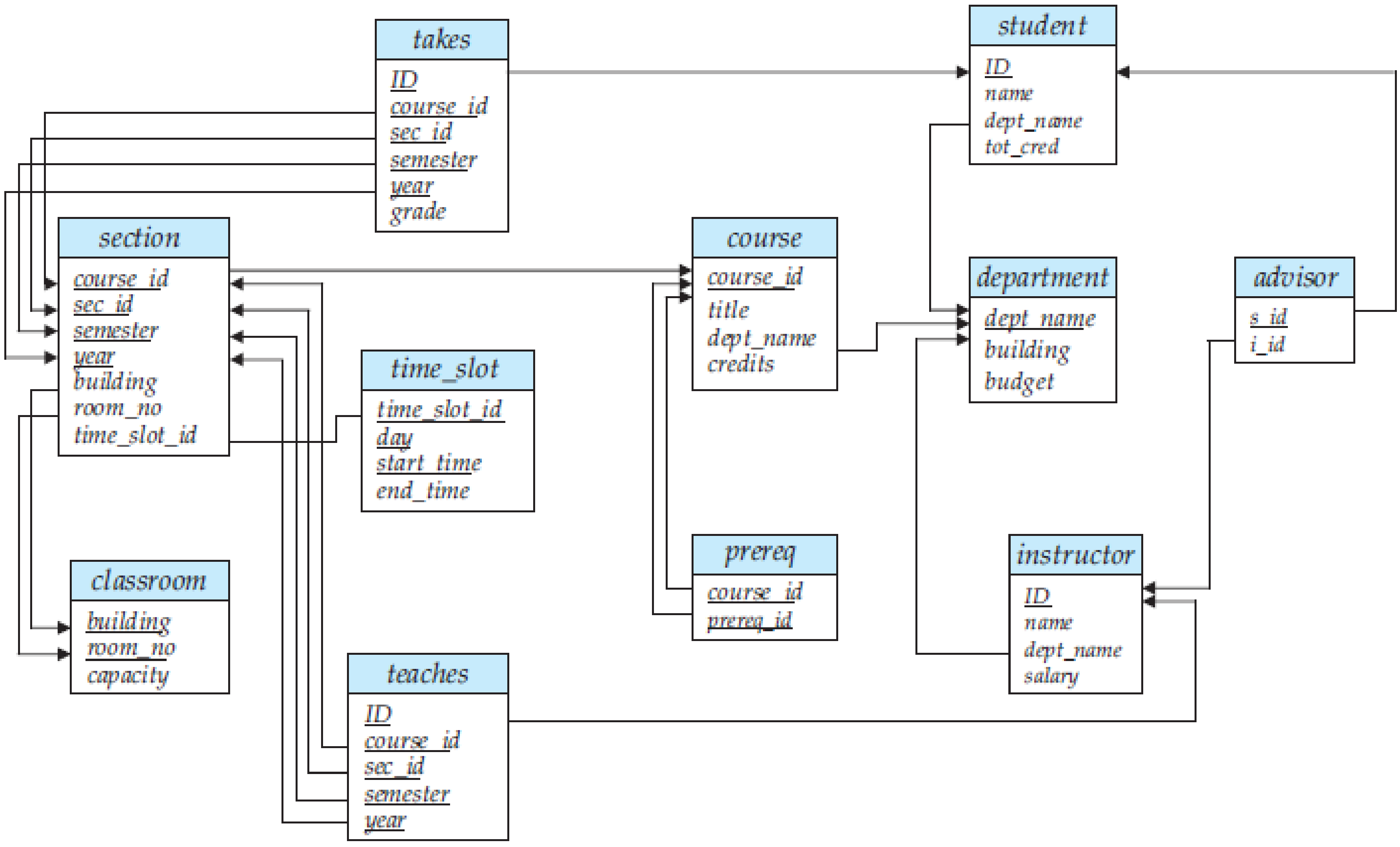

Figure 2.8 Schema diagram for the university database.

Relational Query Language

Procedural vs. Non-procedular/ Declarative Program

Procedural programming requires that the programmer tell the computer what to do

- That is, ***how*** to get the output for the range of required inputs
- The programmer must know an appropriate algorithm

Declarative programming requires a more descriptive style

- The programmer must know ***what*** relationships hold between various entities

- Example: Relational Query Language

Relational Query Language

Procedural vs. Non-procedular/ Declarative Program

Procedural programming requires that the programmer tell the computer what to do

- That is, ***how*** to get the output for the range of required inputs
- The programmer must know an appropriate algorithm

Declarative programming requires a more descriptive style

- The programmer must know ***what*** relationships hold between various entities

- Example: Relational Query Language

Relational Algebra

- Understand relational algebra
- Familiarize with the operators of relational algebra

.

Operation

  - Select
  - Project
  - Union
  - Difference
  - Intersection
  - Cartesian Product
  - Natural Join
  - Rename

Aggregate Operation

Relational Algebra

Relational Algebra is a procedural query language, which takes instances of relations as input and yields instances of relations as output.

.

The main application of relational algebra is to provide a theoretical foundation for relational databases, particularly for query language.

Relational Operation

Relation(r)

.

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

Selection (σ_θ(r)): Is an operation which chooses a subset of rows/tuples from a relation based on a certain condition

$$\sigma_{<\text{selection-condition}>}(\text{Relation name})$$

$\sigma_\theta(r)$ θ is propositional condition

Relational Operation

Relation(r)

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

.

Selection (σ_θ(r)): Is an operation which chooses a subset of rows/tuples from a relation based on a certain condition

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

A	B	C	D
α	α	1	7
β	β	23	10

NULL row can be selected

Relational Operation

Select those tuples of sailors' relation where the rating is a greater than 8

Eid	Ename	Rating	Age
101	Richa	7	24
105	Rohan	9	20
120	Mahesh	8	26
145	Abhishek	10	29

Relational Operation

Select those tuples of sailors relation where the rating is a greater than 8

Eid	Ename	Rating	Age
101	Richa	7	24
105	Rohan	9	20
120	Mahesh	8	26
145	Abhishek	10	29

$$\sigma_{Rating>8}(Sailors)$$

Relational Operation

Select those tuples of sailors relation where the rating is a greater than 8

Eid	Ename	Rating	Age
101	Richa	7	24
105	Rohan	9	20
120	Mahesh	8	26
145	Abhishek	10	29

$\sigma_{Rating>8}(Sailors)$

105	Rohan	9	20
145	Abhishek	10	29

Relational Operation

Select those tuples of sailors relation where the rating is a greater than 8

Eid	Ename	Rating	Age
101	Richa	7	24
105	Rohan	9	20
120	Mahesh	8	26
145	Abhishek	10	29

$$\sigma_{Rating>8}(Sailors)$$

105	Rohan	9	20
145	Abhishek	10	29

SQL:

SELECT * FROM r WHERE condition

Relational Operation

Select operation:

<, <=, =, ≠, >=, >

Relational Operation

Select operation:

<, <=, =, ≠, >=, >

AND: $\wedge$ Conjunction

OR: $\vee$ Disjunctionc

All must be symbol

Relational Operation

Select operation is **commutative.**

$$\sigma_{<cond\ 1>}(\sigma_{<cond\ 2>}(R)) = \sigma_{<cond\ 2>}(\sigma_{<cond\ 1>}(R))$$

Relational Operation

Select operation is **commutative.**

$$\sigma_{<cond\ 1>}(\sigma_{<cond\ 2>}(R)) = \sigma_{<cond\ 2>}(\sigma_{<cond\ 1>}(R))$$

$$= \sigma_{Cond1 \wedge Cond2}(R)$$

Relational Operation

Relation(r)

.

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

Projection (Π_θ(r)): Is an operation which chooses/project a set of column from the original relation.

$$\Pi_{A1,A2,\ldots Am}(r)$$

A1, A2, ...Am is condition / culunm name need to be select

Relational Operation

Relation(r)

.

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

Projection (Π_θ(r)): Is an operation which chooses/project a set of column from the original relation.

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

A	C
α	1
α	1
β	1
β	2

Relational Operation

Relation(r)

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

.

Projection (Π_θ(r)): Is an operation which chooses/project a set of column from the original relation.

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

A	C
α	1
α	1
β	1
β	2

Relational Operation

Relation(r)

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

Projection (Π_θ(r)): Is an operation which chooses/project a set of column from the original relation.

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

A	C
α	1
α	1
β	1
β	2

The relation is a set, in a set every element has to be distinctive.

=

A	C
α	1
β	1
β	2

SQL:
SELECT A,C **FROM** r

Relational Operation

Relation(r)

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

.

Projection (Π_θ(r)): Is an operation which chooses/project a set of column from the original relation.

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

A	C
α	1
α	1
β	1
β	2

The relation is a set, in a set every element has to be distinctive.

=

A	C
α	1
β	1
β	2

Relational Operation

Set: A set is a well-defined collection of distinct objects, Objects in a set are called the elements or members of the set.

.

Key Characteristics of a Set:

- **Distinctness**: Each element in a set is unique. No element appears more than once.
- **Unordered**: The elements of a set do not have a specific order.
- **Well-defined**: A set is defined in such a way that it is clear whether an object belongs to the set or not.

Notation: { }   object separate by ,

Example: A = {1,2,3,4}

Relational Operation

Fetch dept_name and sbuilding of each employee

ID	*name*	*salary*	*dept_name*	*building*	*budget*
10101	Srinivasan	65000	Comp. Sci.	Taylor	100000
12121	Wu	90000	Finance	Painter	120000
15151	Mozart	40000	Music	Packard	80000
22222	Einstein	95000	Physics	Watson	70000
32343	El Said	60000	History	Painter	50000
33456	Gold	87000	Physics	Watson	70000
45565	Katz	75000	Comp. Sci.	Taylor	100000
58583	Califieri	62000	History	Painter	50000
76543	Singh	80000	Finance	Painter	120000
76766	Crick	72000	Biology	Watson	90000
83821	Brandt	92000	Comp. Sci.	Taylor	100000
98345	Kim	80000	Elec. Eng.	Taylor	85000