Introduction to Relational Model

Tags	
# Chapter	2
Date	@September 30, 2023
Person	 Tahsin Islam

Structure of Relational Databases

- It represents data as a **collection of tables**, each with a **unique name**
- A table is also called a **relation**.
- Each row → **Tuple/Row/Record**
- Column headers → **Attributes/Columns/Fields**

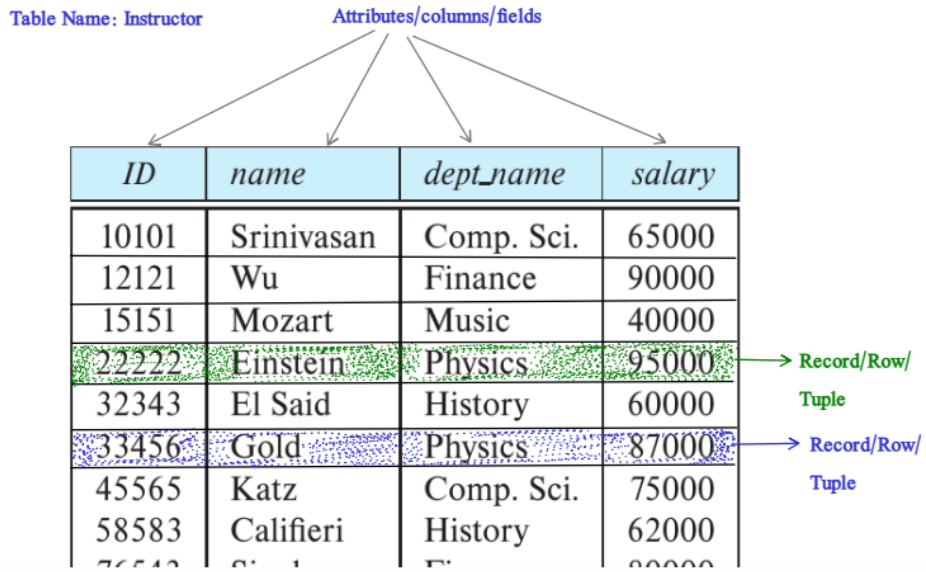

Figure: Relation, attribute and row

- **Domain:** A set of atomic values allowed for an attribute.

> Ex:
>
> a) Name: a string of char that represents the name of a person
>
> b) Employee_age: Possible ages of employees of a company (values between 20 & 70 years old)

- **Atomic:** একটা কলামে একটাই ডাটা থাকবে। যদি name নামক column থাকে সেই কলামে একজনের name শুধু মাত্র Al Jabari ই হবে। তার যদি আরেকটা নাম থাকে যেমন Tauhidur Emon এবং তার name column এ দুইটা নামই যদি দেওয়া থাকে তখন তাকে atomic বলা যাবে না।

> Data gula khudro akare thakbe, eke bhanga jabe na

- **Null:** Indicates that the value is **Unknown**
- **Relation Schema:** Describes a relation

  How to write Relation schema

  > R (A1, A2, A3, A4, … … … , An)
  > R (A1: type, A2 : type, … … … , An : type)

  Example:

  STUDENT(name, roll_no, age, address, phone, grade)

  STUDENT(name : string, roll_no : int, age : int, address : string, phone : string, grade : real)

- **Cardinality:** Total number of tuples present in a relation.

Schema and Instance

1. Schema

- Logical Design of database

> **Structure** of Database

2. Instance

- একটা নির্দিষ্ট সময়ে টেবিলের চেহারা

> Face of table at a specific time

Keys

- Way to **Uniquely identify** each record.
- Evolution of Concepts of Keys:

  Superkey ⇒ Candidate Keys ⇒ Primary Key
- **Notation:**
  - **R** for relation
  - **K** for keys
  - Relation Instance **r(R)**

1. Superkey

- A set of one or more attributes

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

Reg_id	Section	name	age	email_id	phone_no
101	A	Farhan	20		
102	B	Abid	21		
103	B	Shanto	20		
104	C	Abid	19		

In this table, Reg-id, email_id, and phone_no can be unique. So these 3 are candidate keys.

If we group section, name with reg_id we can use that group as superkey.

2. Candidate Key

- Minimal set of attributes that can uniquely identify tuples
- Note: It can be bull, but will not consider same value.

> A Subset of Superkey.

Reg_id	Sec	name	age	email_id	phone_no
101	A	Farhan	20		
102	B	Abid	21		
103	B	Shanto	20		
104	C	Abid	19		

In this table, Reg-id, email_id, and phone_no can be unique. So these 3 are candidate keys.

3. Primary Key

- Primary key can't be
  - Null
  - repeatative

> Primary key has to be Unique

- Format:
  - Informative
  - Non-changeable
  - Efficient to implement

4. Foreign Key

- An attribute or collection of attribute in one table (r1), that refers to the **Primary key** of another table (r2)
- r1 is called **Referencing relation**
- r2 is called **Referenced relation**

Relational Algebra

- Set of operations that take one or two relations as input and produce a new relation
- Six basic operators:
  - select → σ (sigma)
  - project → Π

- union → ∪
- set difference → −
- Cartesian product → ×
- rename → ρ

i. Select Operation

- Notation: $\sigma_p = (r)$

  Example:

  select those tuples of the instructor relation where the instructor is in the "Physics" department.

  In Notation: $\sigma_{deptname=\text{“}physics\text{”}}(instructor)$

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

Figure: instructor relation

ID	name	dept_name	salary
22222	Einstein	Physics	95000
33456	Gold	Physics	87000

Figure: instructor relation **result of Selection**

- Selection Operation: Predicate
  - Comparison: =, ≠, <, >, ⩽, ⩾
  - Combination: ∧ (and), ∨ (or), ¬ (not)

  Example:

  $\sigma_{deptname=\text{“}physics\text{”}\wedge salary>50000}(instructor)$

ii. Projection Operation

- returns its argument relation with certain attributes left out.
- Notation: $\pi_{A1,A2...Ak}(r)$

  Example:

  Select ID, Name and Salary from instructor relation

  Notation: $\pi_{id,name,salary}(instructor)$

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

Figure: instructor relation

ID	name	salary
10101	Srinivasan	65000
12121	Wu	90000
15151	Mozart	40000
22222	Einstein	95000
32343	El Said	60000
33456	Gold	87000
45565	Katz	75000
58583	Califieri	62000
76543	Singh	80000
76766	Crick	72000
83821	Brandt	92000
98345	Kim	80000

Figure: instructor relation **result of Projection**, ordered as per ID

Selection and Projection combined

Notation: $\pi_{name}(\sigma_{deptname="physics"}(instructor))$

iii. Cartesian-Product Operation

- Allows to combine information from two relations

  Example: The Cartesian product of the relations instructor and teaches is written as: `instructor X teachers`

- Both meaningful and meaningless records are found

Dept	Dept Location	Dept Budget
CSE	AB2	2.5
EEE	AB1	2.4

Table: Dept Relation

Name	Prog	DOB	CGPA	Dept
Kim	B.Sc. CSE	1-1-84	3.75	CSE
John	B.Sc.EEE	1-2-85	3.75	EEE
Kim	B.Sc. SWE	3-6-79	3.60	CSE
John	B.Sc. EEE	1-1-84	3.50	EEE

Table: Results Relation

Name	Prog	DOB	CGPA	Dept	Dept	Dept Location	Dept Budget
Kim	B.Sc. CSE	1-1-84	3.75	CSE	CSE	AB2	2.5
Kim	B.Sc. CSE	1-1-84	3.75	CSE	EEE	AB1	2.4
John	B.Sc.EEE	1-2-85	3.75	EEE	CSE	AB2	2.5
John	B.Sc.EEE	1-2-85	3.75	EEE	EEE	AB1	2.4
Kim	B.Sc. SWE	3-6-79	3.60	CSE	CSE	AB2	2.5
Kim	B.Sc. SWE	3-6-79	3.60	CSE	EEE	AB1	2.4
John	B.Sc. EEE	1-1-84	3.50	EEE	CSE	AB2	2.5
John	B.Sc. EEE	1-1-84	3.50	EEE	EEE	AB1	2.4

Table: Resultant Tuples of ***results*** $\times$ ***dept***

- Notation for all tuples: `dept X results`
- Notation for meaningful tuples: $\sigma_{dept.dept=results.dept}$(dept x results)

iv. Union Operation

- Combines two relations
- Selected tuples are concatenated/added back-to-back
- Notation: R ∪ S
- 2 conditions for **compatible** relations
  - Both have the same number of attributes
  - i_{th} attributes of both input relation have to be the same.

> Column Operation

v. Set Difference Operation

- Allows us to find the tuples that are in one relation but not in the other
- First table er attribute gula thakbe result table a.

> Row Operation

vi. Rename Operation

- Notation: $\rho_x(E)$

  It return the result of expression E under the name x.

Example:

1. Find information about courses taught by instructors in the Physics department with salary greater than 70,000

   Notation:

   $\sigma_{deptname=\text{“}Physics\text{”}\wedge salary>70000}(instructor)$

   Or,

   $\sigma_{deptname=\text{“}Physics\text{”}}(\sigma_{\wedge salary>70000}(instructor))$