

# Week - 2.

PAGE NO.

DATE : 21 / 01 / 24

PC  
2-2

Topics:

- Attribute type
- Relation Schema & Instance
- Keys
- Relational Query Language

Ex. of Relat' :-

Attribute (or column)

| ID  | name       | dept-name | Salesy |
|-----|------------|-----------|--------|
| 101 | Ram        | CS        | 90K    |
| 102 | Shyam      | ES        | 50K    |
| 103 | Aam        | History   | 20K    |
| 104 | Chanshym   | Comm.     | 50K    |
| 105 | Radhe shym | Bio       | 2K     |

every entry is a tuple.  
(OR row)

\* ATTRIBUTE :- In relational database, attributes are the characteristics properties that collectively define an entity.  
e.g. ID, name, dept-name

\* DOMAIN :- Set of all allowed value for each attribute.  
Type :- numeric, alpha-numeric, string.

→ Attribute values are normally atomic individual

→ null :- indicate that value is unknown.

## Schema

- Schema is blueprint of database structure.
- organisational rules

Def of attribute & NAME  
 At name is string  
 salary is numeric  
 d) At schema

## Instance

Instance is actual data stored according to rules given by schema

At attr 'name'

Ramesh is instance

Salary - 65K →

- \* **KEYS** :- attribute or set of attribute.  
 → play role in maintaining integrity and relationship b/w tables

### SUPER KEY

- A superkey is set of attribute that can uniquely identify a row in a table.
- A row may contain more than one value.
- e.g.: (Roll, DOB), (ID, Roll, dob, name)

→ every candidate key is superkey ✓

### CANDIDATE KEY

- A superkey that is minimal.
- a table can have multiple candidate key.
- even a table can have one or more candidate key.
- every primary key is candidate key ✓

⑤ SE

⑥ S

⑦ C

③ PRIN

minimal set of attribute that can uniquely identify each and every tuple

⑤ PRIMARY KEY: → A primary key is candidate key that is chosen to be unique identifier for a table.

\* → must be unique, & can't contain null value

→ This key is used to reference rows in other tables.

→ A table can have only one primary key

⑥ SURROGATE KEY: → An artificial key, not based on any real-world attribute  
→ has no inherent meaning.  
→ automatically generated by database

⑦ SECONDARY / ALTERNATE KEY: →

→ other primary key or it's a secondary key.  
→ it can't be null.

→ e.g. first name, last name

⑧ SIMPLE KEY: → contains single attribute  
e.g. table of employee.

⑨ COMPOSITE KEY: → made up of multiple attributes  
→ used when single attribute is not enough to uniquely identify records.  
→ more flexible than simple key but more difficult to understand.

③ Foreign Key :- is an attribute or set of attribute.

- establish a link b/w two table.
- may or may not be null.

④ Compound Key :- made up of multiple attribute.

→ compound key includes both single key & composite key.

- \* PRIMARY key & ~~CANDIDATE~~ <sup>CANDIDATE</sup> key is always Super key.
- \* A foreign key always reference the primary key.
- foreign key can't reference itself.

## RELATIONAL QUERY LANGUAGE

→ Used to interact with relational database and retrieve, manipulate and manage data stored in them.

eg. SQL.

\* Procedural Programming requires the programmer tell the computer what to do.

\* Declarative Programming require more descriptive style.  
→ Programmer must know what relationship hold b/w various entities.

## RELATIONAL OPERATORS.

- A relation is a set, so order of row not matters.
- All rows must be distinct.

| A <sub>1</sub> | B <sub>1</sub> |
|----------------|----------------|
| A <sub>1</sub> | B <sub>1</sub> |
| A <sub>1</sub> | B <sub>2</sub> |
| A <sub>1</sub> | B <sub>2</sub> |
| A <sub>1</sub> | B <sub>3</sub> |

| A              | B              |
|----------------|----------------|
| a <sub>1</sub> | b <sub>1</sub> |
| a <sub>2</sub> | b <sub>2</sub> |
| a <sub>2</sub> | b <sub>2</sub> |

① SELECT OPERATOR ( $\sigma$ ) :-  
 ↴ selecting rows

| A | B | C  | D                    |
|---|---|----|----------------------|
| α | α | 3  | 7 $\rightarrow$ 5    |
| α | β | 5  | 7                    |
| β | β | 12 | 3 $\rightarrow$ 5 X  |
| β | β | 23 | 10 $\rightarrow$ 5 ✓ |

•  $\sigma_{A=B \wedge A > 5} (r)$   
 ↓  
 Note " condition      name of relatin"

| A | B | C  | D  |
|---|---|----|----|
| α | α | 1  | 7  |
| β | β | 23 | 10 |

② PROJECT OPERATOR : ( $\pi$ ) :-  
 ↴ selecting columns

| 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 |

③ Union ( $\cup$ ) & Intersection ( $\cap$ )

| $(r) = \frac{A}{\alpha} \mid \frac{B}{1}$ |   |
|---|---|
| $\alpha$                                  | 1 |
| $\alpha$                                  | 2 |
| B   | 1 |

| $(s) = \frac{A}{\alpha} \mid \frac{B}{2}$ |   |
|---|---|
| $\alpha$                                  | 2 |
| B   | 3 |

no. of column must be same.

UNION of  $(r) \cup (s) = \frac{A}{\alpha} \mid \frac{B}{1}$

domain should be same

Intersection of  $(r) \cap (s) =$

| $= \frac{A}{\alpha} \mid \frac{B}{2}$ |   |
|---------------------------------------|---|
| $\alpha$                              | 2 |

Opposite

difference:  $(r) - (s) = \frac{A}{\alpha} \mid \frac{B}{1}$

$\alpha = 1$  is common  
in both.

~~$(s) - (r) = \frac{A}{\beta} \mid \frac{B}{3}$~~

④ CARTESIAN PRODUCT:  $\frac{A}{\alpha} \mid \frac{B}{1} \quad \frac{C}{\alpha} \mid \frac{D}{2} \mid \frac{E}{a}$

~~$(r) \times (s) = \frac{A}{\alpha} \mid \frac{B}{1} \mid \frac{C}{\alpha} \mid \frac{D}{2} \mid \frac{E}{a}$~~

Delete → removes specific row

Truncate → remove all rows

Drop → drop the table

### \* NATURAL JOIN :- (∞)

| R1 = |   |   |   | S = |   |   |
|------|---|---|---|-----|---|---|
| A    | B | C | D | B   | D | E |
| a    | 1 | a | a | 1   | a | a |
| B    | 2 | r | a | 3   | a | B |
| r    | 4 | B | b | 4   | a | r |

→ r & s have B & D common  
→ it retains only value of same entry & it multiplies

| A | B | C | D | E |
|---|---|---|---|---|
| a | 1 | a | a | a |
| a | 1 | a | a | r |

### DATA DEFINITION LANGUAGE (DDL) :-

↳ deals with structure or Schema of database.

↳ Common DDL commands:

- i) Create :- used to create new table / index.
- ii) Alter : used to modify the structure of existing database.
- iii) Drop : used to delete database object, such as table / index.
- iv) Truncate : used to remove all records from table without removing table structure.
- v) Grant & Revoke : used to manage access control and permission.
- v) Delete : removing the record present in a table

2

PAGE NO.

DATE:

PAGE NO.

20

DATE:

## DATA MANIPULATION LANGUAGE (DML)

↳ focus on manipulating the data stored in database.

↳ common DML commands:-

i) Select : used to retrieve data from one or more table.

ii) Insert : used to add new records to table.

iii) Update : used to modify existing record.

iv) Delete : used to remove records from table.

## \* BASIC QUERY STRUCTURE

i) Select clause:

ex. find the name of all instructors

Select name, address

from instructor

• find dept. name & remove duplicate.

Select distinct dept-name

from instructor

• without removing duplicate

Select all dept-name

from instructor

• \*/asterisk : select all attributes.

Select \*

from instructor

• To find all instructors in CS dept.  
select name  
from instructor  
where dept-name = 'CS'

#### 2-4.4 Additional Basic Operations:

1) '%': (Percent sign)  
select product\_name  
from products  
where product\_name like '%.apple%'

↳ Will return product like "Apple", "apple iphone".

2) '-' (Underscore)

↳ Will return 3-letter words end with 'at'.  
e.g. cat, rat, mat

3) Combination of '%' & '-':

'A-p-e'

↳ Will return word like Apple, Alpin

↳ Concatenation: (add two words)

SELECT CONCAT('Hello', 'world') AS

or

new (ii)

name (ii)

word (ii)

concatenated\_string.

Select 'Hello' || 'world' as ---string.

↳ Substring Extraction:

↳ SUBSTRINg / SUBSTR ('Hello, world', 1, 5) AS  
extracted\_string

↳ Return 1st 5 letters of word.

- 'a%' → start with 'a'
- '%a' → end with 'a'
- '%.001%' → any value that have '001' in any position
- '-.01%' → any value have '.01' at 2nd position
- 'a\_\_%.' → start with 'a' & atleast length 3.
- 'a%.0' → start with 'a' & end with '0'.

→ The percent sign (%) represents zero, one, or multiple characters. and Underscore sign (\_) represents single character.

2.5

Order of Queries:

SELECT > FROM > [JOIN] > WHERE > GROUP BY  
 > HAVING > ORDER BY

Aggregate Functions:

- i) avg:
- ii) min
- iii) max
- iv) sum
- v) count

