Chapter-3

Relational Model (4 hrs) 7-12 marks

Table or Relation : Players_info.

Pid	Pname	SSN	City.
P1	ronaldo	2231	Milan
P2	Messi	2235	Barcelona
P3	hazard	2271	madrid
P4	Pogba	2272	Manchester
P5	bruno	2273	Manchester

3.1 Definition & Terminology

Different terms associated with Relational Model

1. Relation.
   Mathematical name of the table
   example: Players_info.

2. Tuple.
   -Row of a table. example: P1, ronaldo, 2231 Milan is a tuple

3. Cardinality.
   - No of tuples in a relation. eg:- Players-info is of cardinality '5'.

4. Attributes : Column names of a table.
   eg:- Pid Pname SSN City. are attributes.

5. Degree: No. of attribute in a relation.
   eg:- Players_info is of degree 4.

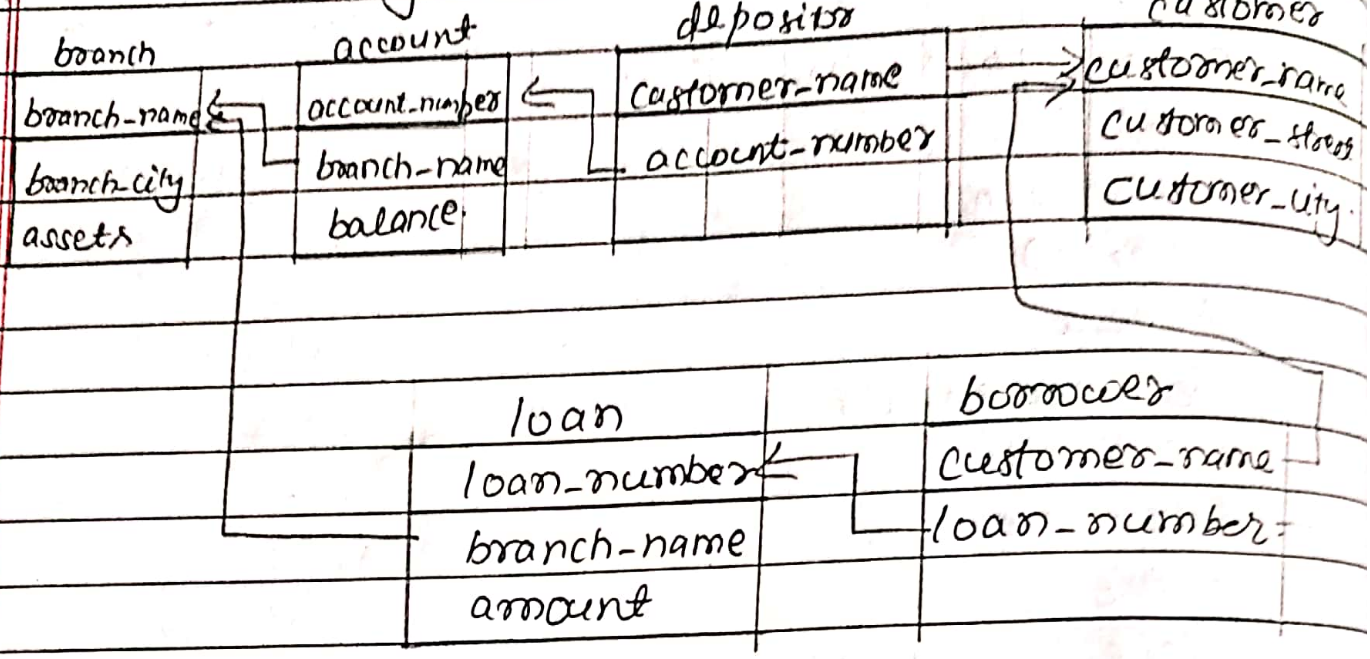
6. Domain: Pools of values in a relation.

Data types in SQL.
int
varchar(n)
char(n)
bigint
decimal (p, d)
smallint
date time
time
date.

Integrity Constraints.
- It ensures that the changes made to the database by authorized user donot result in the loss of data consistency.

Relational model.
- It is the database model in which database is represented as a collection of table (relations). This model is powerful way of representing data and it is now established as primary data representing way for commercial data processing.

Schema Diagram



Q, Draw the schema diagram for college management system.

vImp
Relational Algebra (RA)

- It is a basic set of operation for the relational model. It is procedural query language.
- Relational Algebra is an algebra whose operands are relations & operators are designed to do the most common thing that we need to do with the relation.
- Five basic operations of 'RA' are :-

1) Select (σ)
2) Project (π)
3) Union (∪)
4) Set difference (−)
5) Cartesian Product (×)

1) select operation (σ)
- It is unary operation.
- The select operation is denoted by sigma
- It is used to select a subset of tuples from a relation based on a selection condition.

Syntax :-

$\sigma_{\text{<select condition>}}(R)$, where R is relation

Example :- Relation: Customer

Account_no	Address	Balance
509P	State 4	20,000
510T	State 2	40,000
511W	state3	150000
512P	state4.	250 000

-> select operation selects the tuple that satisfies the given condition.

i) For Address state 4.

$\sigma_{Address = \text{'state4'}}(\text{Customer})$

ii) For balance greater than 1 lakh

$\sigma_{Balance > 100000}(\text{Customer})$

2) Project operation (π)
- It is also an unary operation
- Project operation gives the value of attributes specified
- Duplicate rows are removed from result.

Syntax : $\pi_{attribute-list}(R)$

for eg:- To eliminate address attribute relation Customer.

RA : $\pi_{Account\text{-}no,\ Balance}(customer)$

In SQL : select Account_no, Balance from customer

Account_no	Balance
509 P	20,000
510 T	40,000
511 W	1,50,000
512 P	250 000

Q, Enlist Account_no, Address whose balance is greater than 20,000.

$$\pi_{Account\_no,\ address}(\sigma_{Balance>2000}(customer))$$

$$\Downarrow$$

$$\pi_{attribute}(R)$$

Q, Find address and balance of all customer whose balance is less than 1 lakh.

$$\pi_{address,\ balance}(\sigma_{Balance<100000}(customer))$$

3) Union operation (U).

- It is binary operation
  - Notation: $r \cup s$.
- Duplicate rows are eliminated.

Scanned with CamScanner

Player_name	PId
messi	330
ronaldo	331
pogba	332

Relation:
Football_player.

Player_name	PId
Virat	333
Morgan	334
Rashid	335.

Relation:
Cricket_Player.

→ To find all players name who plays football or cricket.

π_{player_name} (Football_player) $\cup$ π_{player_name} (Cricket_player).

player_name
Messi
ronaldo
pogba
Virat
Morgan
Rashid

→ Set difference operation (−)
It is a binary operation.
- Notation r − s
Example:-
π_{player_name} (Football_player) − π_{player_name} (Cricket player)

player_name
Messi
ronaldo
Pogba.

5, Cartesian Product (x) operation (cross product)

Student-1

Name	Rollno
A	111
B	112

Student-2

Address	Hobby
Satdobato	Football
Kalanki	Singing

Student-1 x student-2

Result:

Name	Rollno	Address	Hobby.
A	111	Satdobato	Football
A	111	Kalanki	Singing
B	112	Satdobato	Football
B	112	Kalanki	Singing.

Imp for SN

Data Dictionary.

- A data dictionary is a collection of names & attributes about data elements that are being used or captured in a database, information system or part of a research project.

It describes the meanings & purpose of data elements within the content of a project and provides guidance on interpretation accepted meanings and representation. - It also provide metadata about data elements.

- The metadata included in a data dictionary can assist in defining the scope and characteristics of data elements as well as the rules for their uses and application.

Advantages of data dictionary.
1. Assist in providing data consistency. across the project.
2. Help define convention that are to be used across a project.
3. Makes data easier to analyze.
4. Enforce the use of data standard
5. Provide consistency in the collection flow of data across multiple member of a tea

Joins in RA:
1. Natural Join (⋈)

Table A.

P	R
b	e
f	h
a	d.

A⋈B.

Table: B.

R	X
d	f
h	k
h	I.

Syntax:

Table name ⋈ Table name.

- no repeated columns in the result set.

2) Equi Join.

- It is basically a natural join in which resultant columns contains repeated columns.

Syntax: $\text{Table-name} \bowtie_{\text{Tablenames}} \text{Tablename}$.

$$\text{Tablename} \bowtie_{\text{Tablename.Attribute} = \text{Tablename.Attribute}} \text{Tablename}.$$

Eg:-

$$A \bowtie_{A.R = B.R} B$$

output:

P	R	R	X
f	h	h	K
f	h	h	I
a	d	d	f

3) Outer Join.

Table: Employee.

Employee-name	Street	City
Cayote	Toon	Hollywood
Rabbit	Tunnel	Carrotville
Smith	Revolver	Death valley
Williams	Seaview	Seatle.

Table: Works.

... can travel in only one route. but
The attribute of passenger are pid(unique), gender and telephone (multivalued)

Table : works.

Employee_name	Branch_name	Salary.
Cayote	Mesa	1500
Rabbit	Mesa	1300
~~Smith~~ Gates.	Redmond	5300
Williams	Redmond	1500 .

Natural join: Employee ⋈ works.

Employee-name	Street	City	Branch_name	Salary.
Cayote	Toon	Hollywood	Mesa	1500
Rabbit	Tunnel	Carrotville	Mesa	1300
Williams	Seaview	Seatle	Redmond	15000

* Left Outer Join.
syntax :- Employee ⟕ works.

Employee_name	Street	City	Branch-name	Salary.
Cayote	Toon	Hollywood	Mesa	1500
Rabbit	Tunnel	Carrotville	Mesa	1300
Smith	Revolver	Death valley	NULL	NULL
Williams	Sea View	Seatle	Redmond	1500

Scanned with CamScanner

* Right Outer Join.

Syntax :- Employee ⟖ works.

Employee_name	Street	City	Branch_name	Salary.
Cayote	Toon	Hollywood	mesa	1500
Rabbit	Tunnel	carrotville	mesa	1300
Gates	NULL	Null	Redmond	5300
Williams	seaview	Seatle.	Redmond	1500

* Full Outer Join.

Syntax: Employee ⟗ Works.

Employee_name	Street	City	Branch_name	salary
Cayote	Toon	Hollywood	Mesa	1500
Rabbit	Tunnel	Carrotville	Mesa	1300
Smith	Revolver	Death valley	NULL	NULL
Williams	seaview	Seatle	Redmond	1500
Gates	NULL	NULL	Redmond	5300.

Q. Consider the relational database. Give the relational algebra expressions for the following.

employee (person_name, street, city)
works (person_name, company_name, salary)
company (company_name, city)
Manages (person_name, manager_name)

1. Find the names of all employees who work for First Bank corporation (FBC).

classmate
Date
Page

2. Find the street and city of the employee whose name is Harish.
3. Find the names and city of residence of all employee who works for FBC.
4. Find the names, street address & city of residence of all employees who work for FBC & earn more than 20000.

1) => $\pi_{person\_name}(\sigma_{company\_name = 'FBC'}(works))$

2) => $\pi_{street, city}(\sigma_{person\_name = 'Harish'}(Employee))$

3) => $\pi_{works.person\_name, employee.city}(\sigma_{works.company\_name = 'FBC'}(employee \bowtie works))$

Or,

(3) $\pi_{works.person\_name, employee.city}(\sigma_{works.company\_name = 'FBC'}(employee \bowtie_{employee.person\_name = works.person\_name} works))$

4) $\pi_{works.person\_name, \cancel{street}, employee.street, employee.city}(\sigma_{company\_name = 'FBC' \wedge salary > 2000}(employee \bowtie works))$.

5) Find the names of all employees who don't work for FBC.

$\Rightarrow$ $\pi_{person\_name}(works) - \pi_{person\_name}(\sigma_{company\_name = 'FBC'}(works))$

Relational Algebra for modification of database:

1. Deletion.
2. Update
3. Insertion.

Table: Depositor.

Customer_name	Account_number.
Abc	A102
Efg	A101
Hik	A115
Lmn	A116
Opq.	A117.

1. Deletion

Syntax: $\boxed{r \leftarrow r - E}$

where, r is the relation.

E is the relational algebra expression.

1) Delete the tuples having account number A115.

$\Rightarrow$ SQL query: Delete from Depositor where Account-number = 'A115';

RA : Depositor $\leftarrow$ Depositor $- \sigma_{Account\_number='A115'}$ (Depositor)

Q.2) Delete the tuple having customer_name Efg & Lmn

RA: Depositor $\leftarrow$ Depositor $- \sigma_{customer\_name='Efg' \wedge customer\_name='LMN'}$ (Depositor).

2. Update:

Syntax: $\boxed{r \leftarrow \pi_{F1, F2, \ldots\ldots Fn}(\sigma_P(R))}$

where, r is the relation.

F1, F2, ..., Fn are the attributes (in order of r).

Fi is an expression (with attribute) that gives new value for attribute.

Q.1) Update the account number A102 to B101.

=> Sql query : Update Depositor set Account_number='B101' where Account_number ='A101';

RA : Depositor $\leftarrow \pi_{customer\_name,\ Account\_number='B101'}(\sigma_{Account\_number='A101'}(\text{Depositor}))$

Q.2. Update the relation so that opq now changes its name to XYZ.

RA : Depositor $\leftarrow \pi_{customer\_name='XYZ',\ Account\_number}(\sigma_{customer\_name='opq'}(\text{Depositor}))$

3) Insertion.

Syntax: $\boxed{r \leftarrow r \cup E}$

where, r is relation.

E is the insertion data.

→ To insert customer named UVW with account number A105.

RA: Depositor ← Depositor ∪ {('UVW', 'A105')}

Q. Consider the relational database. Give the relational algebra expressions for the following:

EMP (name, street, city)

WRK (name, company_name, salary)

COM (company_name, city)

MAN (name, manager_name)

a) Modify the database so that Amrit now lives in Banepa

=> RA: EMP ← π name, street, city='Banepa' (σ name='Amrit' (EMP))

b) Give all employees of 'electronic mart' a 10 percent salary raise.

=> WRK ← π name, company_name, salary = salary * 1.1 (σ company_name = 'electronic mart' (WRK))

c. Delete all tuples in WORK relation for employees of 'Computer Mart'.

=> WORK ← WORK − $\sigma_{company\text{-}name = \text{'Computer Mart'}}$ (WORK)

Aggregate functions in Relational Algebra.

Relation : Works

Employee-name	Branch-name	Salary.
Adams	Perryridge	1500
Gopal	Perryridge	1300
Brown	Perryridge	5300
Johnson	Downtown	1500
Soovena	Downtown	1300
Peterson	Downtown	2500
Rai	Austin	1500
Ray	Austin	1600.

- Aggregate functions take a collection of values and return ~~singal~~ single value as a result.
- sum(), min(), max(), count(), avg() are aggregate functions.
- Syntax is :

$$\boxed{{}_{G_1, G_2 \ldots G_n}\, \mathcal{G}\; F_1(A_1), F_2(A_2) \ldots F_n(A_n)\, (R)}$$

- $G_1, G_2 \ldots G_n$ is the list of attributes on which can be grouped (or can be left empty)
  - Each F_i is an aggregate function.
  - Each A_i is the attribute name.

Scanned with CamScanner

a) Find total sum of salaries of all employee in banks.

=> $\mathcal{G}_{sum(salary)}$ (Works)

b) Find total sum of salaries of each branch name.

=> ${}_{Branch\_name}\mathcal{G}_{sum(salary)}$ (Works)

output:

Branch_name	sum(salary)
Perryridge	8100
Downtown	5300
Austin	3100

Q. Consider the relational database

Doctor(name, age, address)
Works(name, dep_no)
Department (dep_no, floor, room)

Q. Write RA for the following cases:

i) Display floor of doctor named 'Hari'.

=> Display (Project)

=> $\pi_{Department.floor}(\sigma_{Doctor.name = 'Hari'}$ (Doctor ⋈ ~~Works~~ ⋈ Department))

Aggregate function

ii) Count the no. of doctors working on the top floor.

=> ~~$\mathcal{G}_{count(Doctor.name)}$~~ ($\sigma_{Department.floor = 'top'}$ (Doctor ⋈ Works ⋈ Department))

iii) Delete all department of topfloor.

=> $Department \leftarrow Department - \sigma_{floor='top'}(Department)$

iv) Insert the new department as D101, ground, 303.

=> $Department \leftarrow Department \cup \{('D101', 'ground', '303')\}$

update

v) Increase the age of doctor named Abc by 1.

=> $Doctor \leftarrow \pi_{name,\ age=age+1,\ address}(\sigma_{name='Abc'}(Doctor))$.