(1 Q: 7/8 marks)
(4 hours)

Chapter-3

Relational Model

Table: Players_Info

Pid	Pname	SSN	City
P1	ronaldo	2231	Manchestor
P2	messi	2235	Paris
P3	hazard	2271	Madrid
P4	haland	2272	Manchestor
P5	lukaku	2273	Milan

Definitions:-

i) Relation - Mathematical name of the table. E.g. Players_in

ii) Tuple - It means row of a table. ~~E.g. P1 ronaldo~~
E.g. P1 ronaldo 2231 Manchestor is a tuple.

iii) Cardinality - No. of tuple in a relation. E.g. Player_Info table is a cardinality 5.

iv) Attributes - Column name of a table. E.g. Pid, Pname, SSN and City are attributes.

v) Degree - No. of attribute in a relation.
E.g. Player_Info relation is of degree 4.

vi) Domain - Pools of value in a relation.

Notes:-

DATEDIFF.

function that handles larger differences between start date & end date values.

Syntax:-

Date DATEDIFF (datepart, start date, enddate)

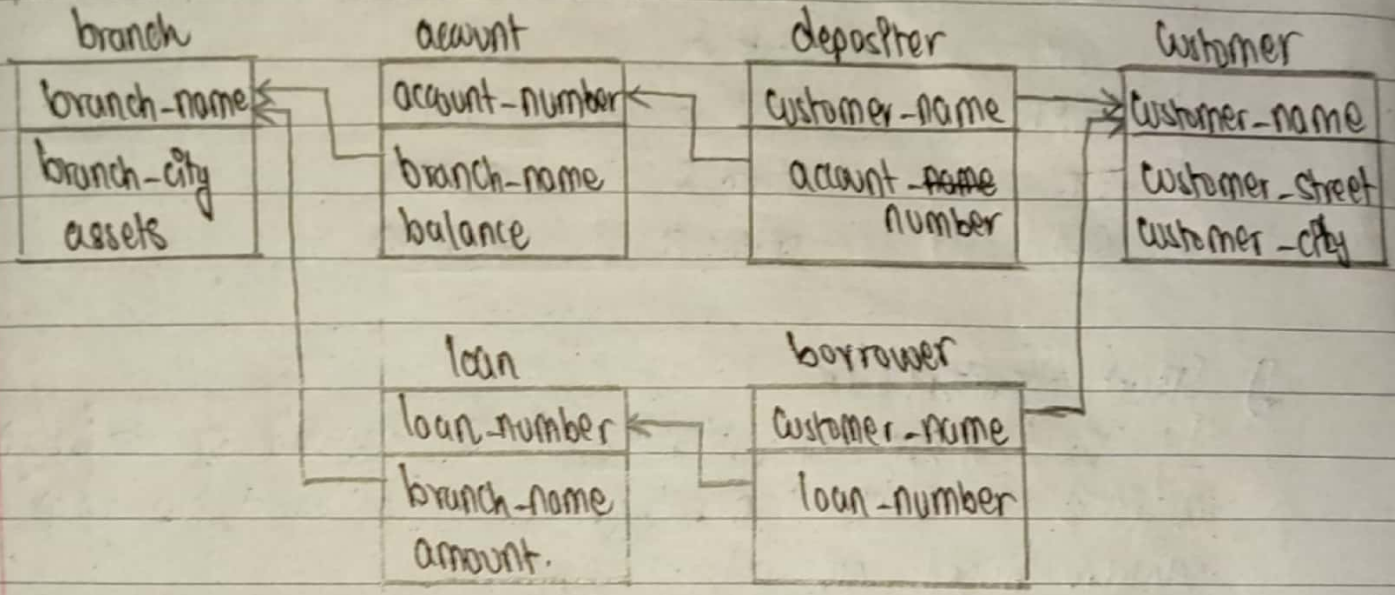
Year → yy, yyyy
Month → mm, m
Day → dd, d
week → wk
Hour → hh
Minute → mi
Second → ss, s

Relational Model:-

It is the database model in which database is represented as a collection of tables. This model is powerful way of representing data and is now established as primary data representing way for commercial data processing.

branch (branch-name, branch-city, assets)
account (account-number, branch-name, balance)
depositor (customer-name, account-number)
customer (customer-name, customer-street, customer-city)
loan (loan-number, branch-name, amount)
borrower (customer-name, loan-number)

Schema Diagram for Relational Model:-



Relational Algebra (RA):-

It is the basic set of operation for the relational model. It is algebra whose operands are relations and operator are designed to do the most common things that we need to do with the relation. Five basic operation of RA are:-

1. Select (σ)
2. Project (π)
3. Union ($\cup$)
4. Set difference ($-$)
5. Cartesian Product ($\times$)
6. Join ($\bowtie$)

Relation R: Customer

Account-no	Address	Balance
5P	State 4	20000
6P	State 2	10000
7PT	State 3	20000
9T	State 2	150000
6W	State 4	250000

1) Select operator (σ):-
It is unary operations. It is denoted by σ (sigma) and is used to select a ~~subject~~ subset of the tuples from a relation based on select condition.

Syntax:-

$$\sigma_{<\text{selection condition}>}(R)$$

Where, R is the name of relation.
The select operation selects the tuples that satisfy the given condition.
For. e.g:
⇒ To find address with state 4.

$$\sigma_{\text{address} = \text{state '4'}}(\text{customer})$$

⇒ For balance greater than 1 lakh (balance > 100000).

$$\sigma_{\text{balance} > 100000}(\text{customer})$$

2) Project (π):-
- It is unary operation.
- Project operation display the values of attribute specified.

The syntax is:-

$\Pi_{<attributes\_list>}(R)$ where, R = name of relation

For e.g.

⇒ To eliminate the address attribute of customer.

$\Pi_{account\_no, Balance}(Customer)$

Output:-

Account no.	Balance
5P	10000
⋮	⋮
⋮	⋮

Q> Enlist account no, address whose balance is greater than 25000.

⇒ $\Pi_{account\_no, \cancel{Balance}\ address}(\sigma_{balance>25000}(Customer))$

Q> Find address of all customer whose balance is less than 80000.

⇒ $\Pi_{address}(\sigma_{balance<80000}(Customer))$

3) Union :- (U)

It is binary operation. Its notation is given by, **R U S**

Where, R and S are relations..

↳ Equal no. of columns & name of columns.

E.g.

Relation: Football_player

Player_name	PId
ronaldo	330
messi	331
haland	332

Relation: Cricket_Player

Player_name	PId
Virat	333
morgan	334
sandip	335

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

Π_{player_name} (Football_player) ∪ Π_{player_name} (Cricket_pla

o/p:-

player_name
messi
ronaldo
haland
Virat
morgan
sandip

4) Set difference operation (−):-

It is also a binary operation. It's notation is given by:

R − S where R and S are relations.

E.g. To display only the player name who plays football only

o/p: Player_name
ronaldo
messi
haland

Π_{player_name} (Football_player) − Π_{player_name} (Cricket_pla

5) Cartesian product (X):- or Cross product or cross join:-

Student-1

Name	Roll no
A	111
B	112

Student-2

Address	Hobby
Satdobato	Music
kalanki	Football

Cartesian product is given by,

Student_1 X Student_2.

O/p:-

Name	Roll no	Address	Hobby
A	111	Satdobato	Music
A	112	kalanki	Football
B	112	Satdobato	Music
B	112	kalanki	Football

6) Joins ($\bowtie$) :- in RA.
It is denoted by $\bowtie$.

Types:-

1. Natural join (Simply denoted by $\bowtie$).
2. Equi join (Denoted by $\bowtie_{table\_1.attribute = table\_2.attribute}$)
3. Outer join.
   - 3.1) left Outer Join ($=\bowtie$)
   - 3.2) Right Outer Join ($\bowtie=$)
   - 3.3) Full outer Join ($=\bowtie=$)

Ans Q7. Explain Joins in RA with examples.

Q>. Consider the relational database. Give the relational algebra expression for the following.
Employee (Person_name, street, city)
works (Person_name, Company_name, Salary)

Company (Company_name, city)
Manages (person_name, manager_name)

Write RA for the following cases:

1) Find the name of all employees who works for First Bank Corporation (FBC).
2) Find the street and city of the employee whose name is 'Han'.
3) Find the names and city of residence of all employee who works for the First Bank Corporation (FBC).
4) Find the names, street, and city of residence of all employee who works for FBC and earn more than 20000.
5) Find the name of all employees who do not work for FBC.

Answers-

1) ~~Π_{person_name} (Employee) $\cup$ Π_{person_name} (Works) $\leftarrow$ work = FBC ($\sigma_{company\_name = FBC}$ (Works))~~

1) $\Pi_{person\_name}(\sigma_{company\_name = 'FBC'}(Works))$

2) $\Pi_{street, city}(\sigma_{person\_name = 'Han'}(Employee))$

4) $\Pi_{employee.person\_name, employee.street, employee.city}(\sigma_{company\_name = 'FBC' \wedge salary > 20000}(employee \bowtie works))$

3) $\Pi_{employee.person\text{-}name,\ employee.city}\left(\sigma_{company\text{-}name='FBC'}(employee \bowtie works)\right)$

OR,

$\Pi_{employee.person\text{-}name,\ employee.city}\left(\sigma_{company\text{-}name='FBC'}(employee \bowtie_{employee.person\text{-}name = works.personname} works)\right)$

5) $\Pi_{person\text{-}name}(Employee) - \Pi_{person\text{-}name}\left(\sigma_{company\text{-}name='FBC'}(works)\right)$

OR,

$\Pi_{person\text{-}name}(works) - \Pi_{person\text{-}name}\left(\sigma_{company\text{-}name='FBC'}(works)\right)$

Imp (S.N)

Data Dictionary :-

- A data dictionary is a collection of names and attributes about data elements that are being used or retrieve in a database or in an information system.
- It describes the meaning and purposes of data elements within the context of a project and provides guidance on interpretation, accepted meanings or values and representation
- It also provides meta data about data elements.

(Imp) - The metadata included in a data dictionary can assist in defining the scope and characteristics of data element as well as the rules for the uses and applications.

Advantages of data dictionary :-

- It helps to define convention that are being used in the

$\wedge \rightarrow$ And
$\vee \rightarrow$ Or

project.
- It helps data easier to analyse.
- It enforce the use of data standard.
- It provides consistency in the collection of data and use of that data among various developer across the project

Relational Algebra for the modification of Database :-

(i) Delete
(ii) Update
(iii) Insertion

(i) Delete :-
Syntax: $r \leftarrow r - E$
where, r is a relation and E is the relational algebra query.

⇒ Example
Relation = Depositor

Customer_name	Account_number
Abc	A102
efg	A103
hjd	A104
klm	A105
nop	A106

⇒ To delete the ~~tuple~~ tuple having account_number A105-
Depositor ← Depositor $- \sigma_{Account\_number = 'A105'}$ (Depositor)

⇒ Delete the tuple having customer-name egg and nop.

$$Depositor \leftarrow Depositor - (\sigma_{customer\_name = 'egg'} (Depositor) \wedge \sigma_{customer\_name = 'nop'} (Depositor))$$

(ii) Update-

Syntax:- $r \leftarrow \Pi_{F_1, F_2, F_3 \ldots F_n} (\sigma_p (r))$

where, 'r' is the relation. $F_1, F_2, \ldots F_n$ are the attributes in order of 'r'. F_i is an expression with attribute that gives new value for the attribute.

Eg.

⇒ Updating the account no A102 to B102.

$$Depositor \leftarrow \Pi_{customer\_name,\ Account\_number = 'B102'} (\sigma_{Account\_number = 'A102'} (Depositor))$$

↳ all columns ↳ F_i = value

⇒ Updating relation so that 'nop' changes its name to 'pon'.

$$Depositor \leftarrow \Pi_{customer\_name = 'pon',\ Account\_number} (\sigma_{customer\_name = 'nop'} (Depositor)).$$

(iii) Insert-

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

where, E is the tuple to be inserted.

⇒ To insert the data 'uvw' and A107.

$$Depositor \leftarrow Depositor \cup \{('uvw', 'A107')\}$$

Q7. Consider the relational database :-
EMP (Name, street, city)
WRK (Name, company name, salary)
COM (Companyname, city)
MAN (Name, mgrname)
Write RA for the following :-

(1) Modify the database so that Abc now lives in Pokhara.
(2) Give all employees of electronics a 10% salary increase
(3) Delete all the tuples in ~~work~~ WRK relation for employees of computer.

⇒ (1) $EMP \leftarrow \Pi_{Name, street, city = 'Pokhara'}(\sigma_{name = 'Abc'}(EMP))$

(2) $WRK \leftarrow \Pi_{Name, companyname, salary = 1.1 * salary}(\sigma_{companyname = 'electronics'}(WRK))$

(3) $WRK \leftarrow WRK - \sigma_{company\_name = 'computer'}(WRK)$

Aggregate functions in RA :-

Syntax is,

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

Aggregate functions takes a collection of values and return a single value as a result. Some of the aggregate functions are ~~some~~ sum(), max(), min(), count(), avg()

from the above syntax, 'E' is any relational algebraic expression and $G_1, G_2 \ldots G_n$ is a list of attribute on which we can group (it can be left empty). Each F_i is an aggregate function and each A_i is an attribute name

Table name Pt_works

Employee_name	Branch_name	Salary
Adams	Perryridge	1500
Gopal	Perryridge	1300
Brown	Perryridge	5300
Johnson	Downtown	1500
Soovera	Downtown	1500
Peterson	Downtown	2500
Rao	Austin	1500
Roy	Austin	1600

Examples-

1) Find the total sum of salary of all employee.

⇒ $\mathcal{G}_{sum(salary)}$ (Pt_works)

2) Find the total sum of salaries of each branch-name.

⇒ $\mathcal{G}_{Branch\text{-}name\ sum(salary)}$ (Pt_works)

on basis of

Output

Branch_name	sum(salary)
Perryridge	8100