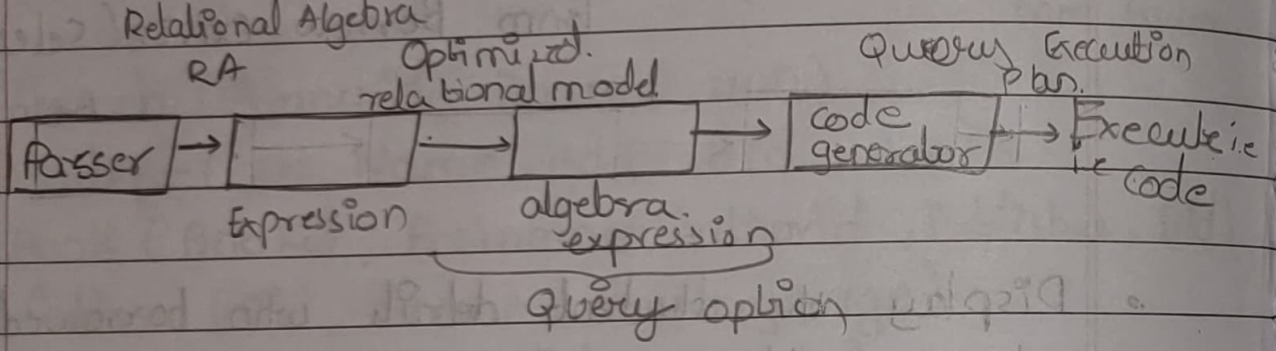
UNIT - II

RELATIONAL DATA MODEL



r → extention

Characteristics of Relations

① Ordering of tuples in a relation r(R)
② Ordering of attributes in a relation schema R
③ Values in a tuple.

t2[Name] = "Reecha"
t2[Name, USN] = "Reecha", "2GI20CS108"

Constraints

① Domain [Datatype]
② Key
③ Entity Integrity
④ Referential Integrity.

Chara

~~Ordering of Tuples of Relation~~
Values.

RELATION OPERATOR

① Unary RO

Horizontal • SELECT [σ] = Ex σ DNO = 4 (EMPLOYEE)

Vertical • PROJECT [π] → π LName, FName, Salary (Employee)

• σ Dno = 4 & AND Salary > 25000 (EMPLOYEE).

• π Name, Salary : DeptNO (Employee)

Method 1 : by nesting the operations.
π Fname, Lname, Salary (σ DNO = 5 (EMPLOYEE))

Method 2 : Renaming & creating intermediate result.
relations one operation at a time.

• DEP5 - EMPS ← σ DNO = 5 (EMPLOYEE)
RESULT ← π Fname, Lname, Salary (DEP5 - EMPS).

• TEMP ← σ DNO = 5 (EMPLOYEE)
R (First-name, Last-name, Salary) ← π Fname, Lname, Salary (TEMP).

② Binary RO

- Union R ∪ S [Duplicate tuple terminated]
- Intersection R ∩ S
- Minus R − S [That are their R but not in S]

Cartesian Product.

BANK DATABASE SCHEMA

- Branch (BName , BCity , Assets)
- Customer (CName , CStreet , CCity)
- Account (ACC-No , BName , Balance)
- Loan (Loan-No , BName , Amount)
- Depositor (CName , ACC-No)
- Borrower (CName , Loan-No)

① Find the names of all the customers who have loan at the bank along with loan no and loan amount

→ Temp ← $\Pi_{CName, Loan\_No, Amount}$ (Borrow ⋈ Loan)

$\Pi_{CName, Loan\_No, Amount}$ (σ (Borrow × Loan))

(2) Find the names of all the branches with the customers who have an account in the bank and who live in Bangalore.

→ Π_{Bname} ($\sigma_{Cby='Bang'}$ (Customer ⋈ Depositor ⋈ Account))

(3) Find the names of all the customers who have both loan and account at the Bank.

→ Π_{Cname} (Borrower ⋈ Depositor)

Π_{Cname} (Borrower) ∩ Π_{Cname} (Depositor)

(4) Find the names of the customers who have loan at the Bangalore branch.

→ Π_{cname} ($\sigma_{BName='Bang'}$ (Borrower ⋈ Loan))

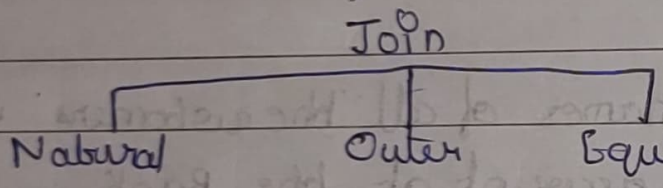
(5) Find the names of all the customers who have the loan at the Mysore but do not have an account at any branch of ~~branch~~ bank.

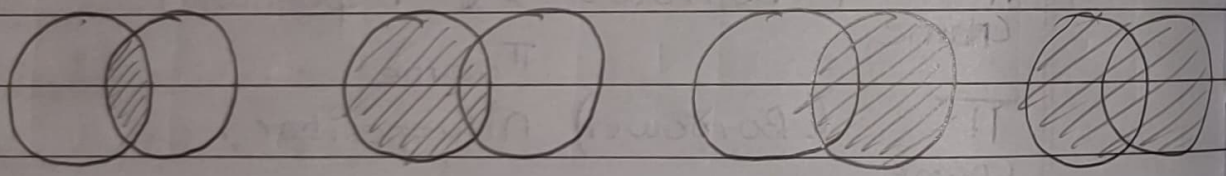
→ Π_{cname} ($\sigma_{BName='Mysore'}$ (Borrower × Loan)) − Π_{Cname} (Depositor)

DERIVED

Join

- Inner [Natural, Equi]
- Outer [Left outer Join, Right Outer Join, Full Outer Join]





COMPANY SCHEMA

Dept (Dnum, Dname, MgrSSN, Mgrdate)

Emp (SSN Name, Bdate, Address, Gender, Salary, Supper-SSN, DNO)

Project (Pnum, Pname, Ploo, Dnum)

① Retrieve the name and address of all Emp who works for admin dept

→ Temp ← $\sigma_{Dname = Admin}$ (Dept)

Result 1 ← Temp ⋈ Emp
Dnum = Dno

Result ← Π name, Address (Result 1)

(2) For every project located in stanford list the Pnum controlling Dnum, Dmgr last Name, Address and DOB

→ Temp ← σ plac = 'stanford' (Project)
Result 1 ← Temp ⋈ Dept
Result 2 ← Emp ⋈ Result 1
MgrSSn = SSN

Result ← Π Pnum, Dnum, Name, Address, Bdate (Result 2)

3 Find the name of all the employee who works on all the project controlled by Dnum = 5

→ Temp ← Π (σ Dnum = 5 (Project))
Temp 1 ← Temp ⋈ Emp
Dnum = Dno

Result ← Π name (Temp 1)

DIVISION OPERATOR (÷, /)

Suited to queries that include the keyword "all" or "every" like "at all", "for all", or "is all", "at every", "for every" or "in every".

Ex

① A

X	Y
a	1
b	2
a	2
d	4

÷ B

Y
1
2

→ A ÷ B

X
a

② A

Sno	Pno
S1	P1
S1	P2
S1	P3
S1	P4
S2	P1
S2	P2
S3	P2
S4	P2
S4	P4

B1

Pno
P2

A ÷ B1

Sno
S1
S2
S3
S4

B2

Pno
P2
P4

A ÷ B2

S1
S4

B3

Pno
P1
P2
P4

A ÷ B3

Pno
S1

classmate

Date

Page

$$A/B = \Pi_x(A) - \Pi_x((\Pi_x(A) \times B) - A)$$

all disqualified tuple

(3)

A	B	C	D	E
α	a	α	0	
α	a	γ		
α	a	γ		
β	a	γ		
β	a	γ		
γ	a	γ		
γ	a	γ		
γ	a	β		

A

D	E

B

(4)

Enrolled E

sid	cid
S1	C1
S2	C1
S1	C2
S3	C2

Cource C

cid
C1
C2

~~Etc~~ $\Pi_{sid}\left(\left(\Pi_{sid}(Enrolled) \times Course\right) - Enrolled\right)$ S2 S3

$\Pi_x(A)$ B A

$\Pi_{sid}(Enrolled) - \Pi_{sid}\left(\left(\Pi_{sid}(Enrolled) \times Course\right) - Enrolled\right)$

Find all customers who have an account at all branches located 'brooklyn'.

→ $r_1 = \Pi$

● Student (name, usn, cid, Percentage, Fname)
Course (Cid, cname)
Enroll (Cid, usn, Grade)
Faculty (FID, Fname, Salary)

① Retrieve the name and % of all students for the course 18CS43

→ Temp ← $\sigma_{\text{cname} = '18CS43'}$ (Course)

Temp1 ← $\sigma\ \Pi_{\text{cid, cname, usn}}$ (Temp ⋈ Enroll) Temp.cid = Enroll.cid

Temp2 ← Temp1 ⋈ Student Temp1.usn = student.usn.

Result ← $\Pi_{\text{name, Percentage}}$ (Temp2)

② List the Dept's having an average salary of the faculties above Rs 30000

→ Temp ← $\sigma_{\text{sum(salary)}}$ (Faculty)

Temp1 ← $\sigma_{\text{avg(salary)}}$ (Faculty ⋈ Temp)

$Temp2 \leftarrow \sigma_{avg(salary) > 30,000}(Temp1)$.

$Result \leftarrow \pi_{Dname}(Temp2 \bowtie_{Did = DPd} Dept)$.

(3) List name of the course having students grade 'A' maximum.

→ $Temp \leftarrow \sigma_{grade = 'A'}(Enroll)$

$Temp1 \leftarrow Temp \bowtie Course$.

$Result \leftarrow \pi_{cname}(Temp2)$.

(1) Retrieve the name of employees who work for 'Research' Dept.

→ $Temp \leftarrow \sigma_{Dname = 'Research'}(Dept)$

$Temp1 \leftarrow Temp \bowtie Emp$

$Result \leftarrow \pi_{Fname}(Temp2)$

(2) List the name of managers who have atleast one dependent.

→ $Temp \leftarrow \sigma_{Dependent >= 1}(Dependent)$

Temp1 ← Dept ⋈ Temp.
Mgrssn = ESSn

Result ← π_{Ename} (Temp1 ⋈ $_{Mgrssn=ssn}$ Emp)

③ List the dependents of particular emp name with John smith

$\pi_{dependentname}$ ($\sigma_{Ename='John\ Smith'}$ (Dependent ⋈ Employee))

④ List the emp with salary greater than 10000

π_{Ename} ($\sigma_{salary>10000}$ [Emp])