

Relational

Algebra

Query Language is a language in which user request information from the database.

Relational Algebra \Rightarrow Procedural Language

→ Unary (Include only 1 table)

→ Binary (Include 2 tables)

Query ke steps \Rightarrow first identify which table is used.

σ write condition to apply

Π what you would like to display

Select Operation :- σ predicate (Table name)

Project Operation :- $\Pi_{\text{col1}, \text{col2}}$ (Table name)

Q:- Find loan no. which amt greater than 60000

$\Pi_{\text{loan-number}} (\sigma_{\text{amount} > 60000} (\text{loan}))$

Q-2 Find name of customers who have either loan or an account or both.

$\Pi_{\text{customer-name}} (\text{depositor})$

$\Pi_{\text{customer-name}} (\text{borrower})$

Q-3 Find name of customer who have an account but not a loan.

$\Pi_{\text{customer-name}} (\text{depositor}) - \Pi_{\text{customer-name}} (\text{borrower})$

Two Tables

Cartesian or Cross product

borrower		depositor	
cu-name	Loan-no°	cu-name	Account-no
Adam	L-16	X	Hoye
Hoye	L-17		Smith

$$\text{No. of columns} = 2 + 2 = 4$$

$$\text{No. of rows} = 2 \times 2 = 4$$

borrower \times depositor

Table formed by cross product

borrower·cuname	Loan-no	depositor·cuname	Account-no
Adam	L-16	Haye	A-102
Adam	L-16	Smith	A-103
Flages	L-17	Haye	A-102
Flages	L-17	Smith	A-103

See Agar koi column ka name same hai
dono table mein then we will display the
column by table name · column name

Ab jo query Hai ki print name of customer
who have both account & loan can
also be written like

$\Pi_{\text{borrower}\cdot\text{c-name}} (\sigma_{\text{borrower}\cdot\text{c-name} = \text{depositor}\cdot\text{c-name}} (\text{borrower} \times \text{depositor}))$

Rename Operation \Rightarrow $\int_{\text{new name}}^{\text{P}} (\text{table name})$

#	Accno	br-name	balance
101	SBI-A	10000	
102	SBI-B	50000	
103	SBI-C	70000	

Ques Maximum balance in Accounts table.

- before karo rename, then karo - cross product condition logao ($acc\text{-balance} < \text{renamename}$)
- Ab ye jo data generate korega it will have acc balance = 10, 50K Kyonki 70K will not be less than 10, 50, 70 = ab issey home milaga 10, 50 K

→ Original table mein se - karo
 $10, 50, 70 - 10, 50 = 70 \rightarrow \text{highest}$

Generate (10,50K)

=> $\prod_{Account} \text{balance} \left(\sigma_{Account \text{balance} < do \text{balance}} \text{Account} \times P_d(\text{Account}) \right)$

~~Ans~~ $\prod_{Account} \text{balance} (Account) - \prod_{Account \text{balance} < do \text{balance}} (\sigma_{Account \text{balance} < do \text{balance}} (Account) \times P_d (Account))$

Natural join opn = $| \times |$

→ Ye jo operation hai isme kya kota
 hai ki jo same column kota hai
 vo ek ho Joato hai Aur vo ho e
 entry aata hai jinke dono column entry
 same ho

Taise Natural join pehli table mein
 borrower $| \times |$ depositor

Customer	Loan no	Acc-no
Hase	9-17	A-102

Theta join opn = $| \times | \circ$

→ Ab jo column same hai usme koi condition
 aa jaye toh
 $| \times |$ balance > 50000

Division Operation

= Find all customers who have an account at all the branches located in Jaipur.

→ Identification

$\sigma_1 = \Pi_{\text{branch-name}} (\square_{\text{branch-city} = \text{jaipur}} (\text{branch}))$

$\sigma_2 = \text{Abisme jo } \sigma_1 \text{ ka data hai vo } \sigma_2$
mein aayega

= $\Pi_{\text{customer-name, branch-name}} (\text{depositor } \times \text{ account})$

$$\gamma = \sigma_2 \div \sigma_1$$

Aggregate functions

$\text{aggfunc (coln)} (\text{table name})$

G (employee)
sum (salary)

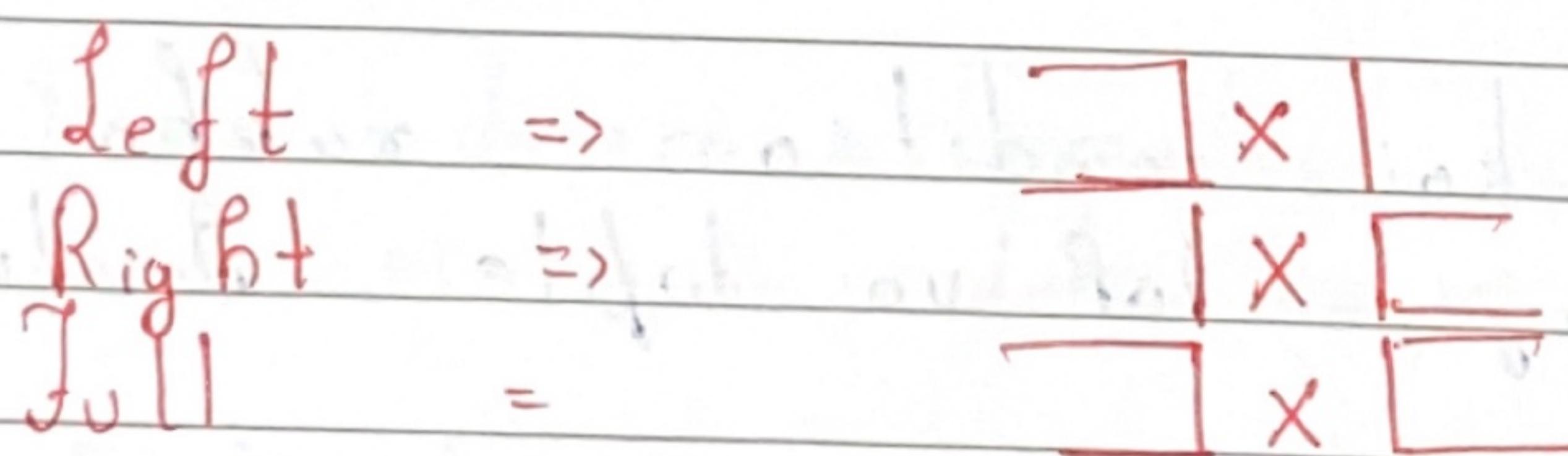
max, min

- Count distinct values or count values

G (employee)
count-distinct (dept)

G (Employee)
count (dept)

Outer join => join ka jo data hai vo
data nahi hai
toh ho saath mein jo



Left => Left ka data (table) hogा jahan
be data Nahi hogा jahan likhenge
NULL.

~~borrower name~~

(Left Join)

uname	loan-no	Account-no
Adam	L-16	NULL
Huye	L-17	A-102

Similarly Right or Full Somaj jao

Null ko Somaj =>

True and Unknown => unknown

False and Unknown => false

Unknown and Unknown => unknown

Null value ke koi condition rakhna agar unknown aye toh vo tuple include nahi hogi

Deletion

$\text{Customer} \leftarrow \text{Customer} - \sigma_{\text{cuname} = \text{Amit}} (\text{customer})$

Assignment

Insertion

$\text{customers} \leftarrow \text{Customer} \cup \{ ("-", "-", "-") \}$

View \Rightarrow Ab jo ooh table query ke
through display karte ho muando ki
wo oob permanent karna chahao.

- Jisme data automatic update ho us
soon as new data aaye
- Abstraction ki layer badi jayegi jo
kisi ko sirf customer name dikhuna ho ya
kuch nahi dikhana ho
- View iss kom auto hai

* View ko create karne ke liye :

Create view all customer as  ye aayega

$\Pi_{br-name, cu-name} (depositor | x | Account)$

U

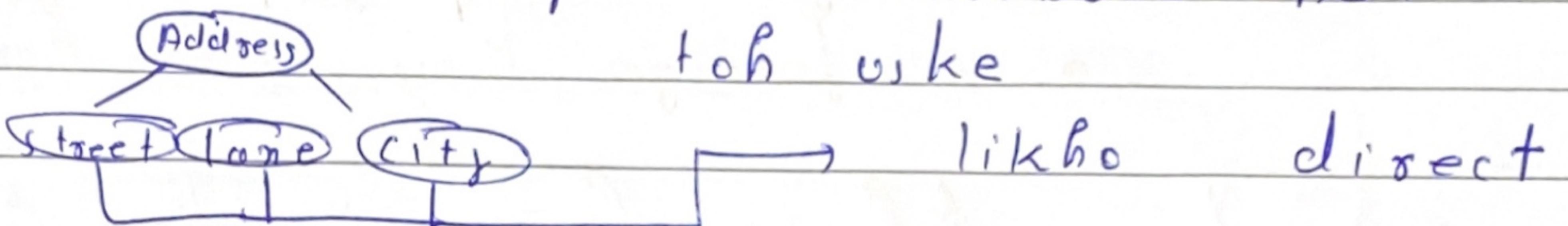
$\Pi_{br-name, cu-name} (borrower | x | loan)$

Conversion of ER To Relational Model

#1

Identify strong Entity Set

- Relation bandar with all attribute
- Ayor koi composite attribute hai



- Leave Multivalue attribute.

#2

Identify Weak Entity Set

- Relation bandar with attribute include Primary key of Strong Entity set.

- Primary os Partial key done karo under line (done as a combine Primary key behave kareng.)

#3

Mapping of 1:1 Relationship types.

abb-1

foreign key

abb-2

Merged Relational

abb-3

Cross Reference

foreign Isme jo do entities ne relationship mein participate kiga hai unme se Kisi ek ko chuno aur dure ki primary key us ke attribute mein daaldo

= (Total participation hone chabiye uska jiski aap primary key as a foreign key use kar sake ho

= Yani jab tak foreign key banegi vahan total participation hona chabiye.

= Agar koi relationship ka attribute hai toh vo bhi vahan daaldo jab tak

Merged Relation (Total participation) is Required

Cross m:n binary relation

#4

Mapping of N:1 Relation type.

Ajo entity N side pe hai
vahan pe foreign key add hogi
use corresponding wale ki primary
key

Agar koi relation ko attribute ho
toh jab foreign key hai vahin
daaldo

#5

Mapping of N:N Relation Type (cross referencing)

Ek maya relation banao jiska naam
hoga relation

Dono Entity ki primary key aayengi

P.K=1	P.K=2
1	101
2	101

relationship ke attributes bhi
aa sakte hain if have

One to One => Mein bhi aip cross referencing laga sakte ho primary key dono mein se koi bhi challega.

N to One => primary key banegi vo jo N side ki entity ki hai

#6 Mapping of Multi Valued Attributes.

Multi valued Attribute ka ek alag relation banega jisme vo alag naam se + primary key of Entity aayegi

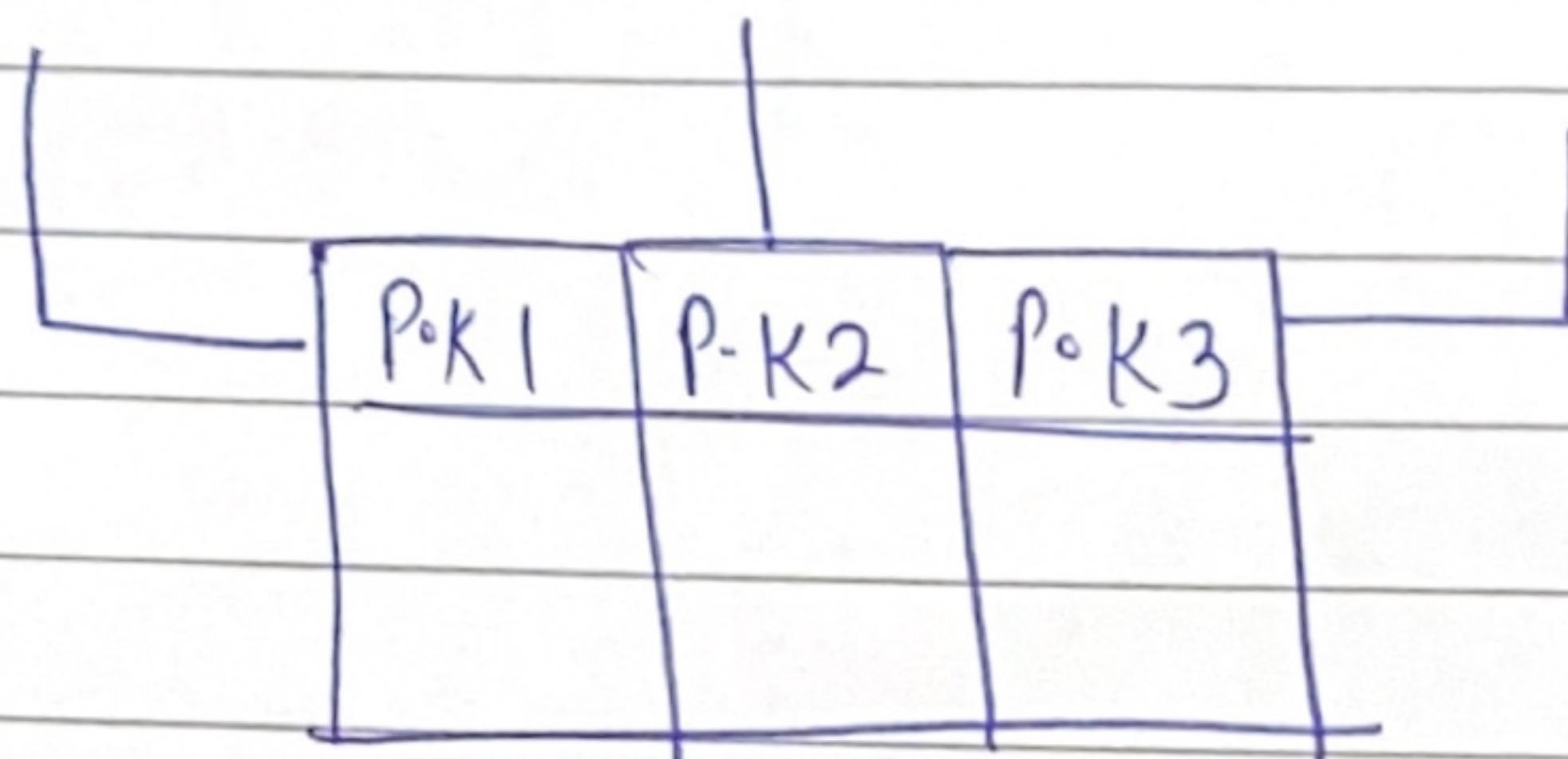
P Key-1	attribute

#7

Mapping of Nary Relationship ($N > 2$)

→ All are strong Entity

→ Sabki primary key augayenge
table mai



Primary Key \Rightarrow Combination of these 3 Keys -