Day

" RELATIONAL ALGEBRA "

→ Relational algebra is a theoretical query language.
→ Every operater in relational algebra take one or two tables as input, and generate a single table as output.
→ R.A donot consider duplicacy.

▷ SELECT OPERATOR :

→ Unary operator , can take one table at a time.

$\sigma_{\text{condition}}$ (table name)

$\sigma_{\text{first\_name} = \text{"Samsoge"}}$ (Employee)

Selection mein saare columns retrieve honge.

- $\sigma_{\text{salary} > 5000}$ (Employees)

$\sigma_{\text{salary} > 6000}$ ($\sigma_{\text{dep\_no} = 10}$ (Employee))

or

$\sigma_{\text{dep\_no} = 10}$ ($\sigma_{\text{salary} > 6000}$ (Employee))

or

$\sigma_{\text{dep\_no} = 10 \wedge \text{salary} > 6000}$ (Employee)

↳ AND

$\vee$ ↳ OR

Date _____ Day _____

PROJECT OPERATOR:

→ Takes one table at a time.

→ Used to select a column.

π column_name (table_name)

π name (Employees)

SELECT * from Employees ⟶ (Employee)

↳ this will be the RA as we not need any condition because the table which is input will be the output.

projection binary operator

select unary operator.

π first_name (σ dep_no = 10 (Employee))

↳ will get first_name of all employees in dep = 10.

OR

TEMP ← σ dep_no = 10 (Employees)

RESULT ← π first_name (TEMP)

RENAME:

→ Denoted by ρ (rho)

ρ_S (B1, B2, B3, ... BN) (table_name)

↳ Changes relation name to S & attribute to B1, B2 ...

ρ_S (table_name) → changes only table name

ρ (B1, ... BN) (table_name) → changes only attribute

If we write

RESULT (F, M, L, S, B, A, SX, SAL, SU, DNO) ←

ρ_{RESULT} (F, M, L, S, ... DNO) (DEPS_EMPS)

Date _____ Day _____

▷ UNION OPERATOR :

→ The result of R ∪ S, is a relation that includes all tuples that are either in R or in S or in both R & S.

→ Duplicates are eliminated (for all set operation)

→ The two operand relations R & S must be type compatible.

→ R & S must have same no. of attributes.

• To retrieve the SSN no. of all employees who either work in dep 5 or directly supervise any employee who works in dep 5.

DEP5_EMP ← $\sigma_{DNO=5}$ (Employees)

π_{SSN} (DEP5_EMP) ∪ $\pi_{SUPERSSN}$ (DEP5_EMP)

▷ INTERSECTION OPERATOR :

→ The result of R ∩ S → include tuples that are in both.

▷ SET DIFFERENCE OPERATOR :

→ The result of R − S → include tuples that are in R but not in S.

↳ The attribute names in result will be same as attribute name in R. R − S ≠ S − R

Date _____ Day _____

CARTISAN PRODUCT:

→ Used to combine tuples from two relations/tables.

$R(A_1, A_2, \ldots A_n) \times R_2(B_1, B_2 \ldots B_m) =$

$Q(A_1, A_2 \ldots A_n, B_1, B_2 \ldots B_m)$

total no. of tuples = $n \times m$

total no. of columns = $n + m$

$A \times B \neq B \times A$

→ Two operands doesn't have to be type compatible.

Account × Depositor

	Acc.no	bno	bal	C no	Depositor. Ano
1	101	X	50	~~50~~ 1	101 ✓
2	101	X	50	2	102
3	101	X	50	3	103
4	102	X	100	1	101
5	102	Y	100	2	102 ✓
6	102	Y	100	3	103
7	103	Z	60	1	101
8	103	Z	60	2	102
9	103	Z	60	3	103 ✓

Account

Ano	bno	bal
101	X	50
102	Y	100
103	Z	60

Depositor

C no	Ano
1	101
2	102
3	103

→ π [Account × Depositor]

Only tuple no 1, 3, & 9 are valid. Other tuples are invalid or surplous.

Date _____ Day _____

To get [illegible]ving bal > 1000 & only valid ones.

$\Pi_{Cno}(\sigma_{Account.Ano = Depositor.Ano \wedge bal > 1000}(Acc \times Deposit))$

↳ for removing surplous tuples.

$\Pi_{cust\_name}(\sigma_{assets > 10,000}(Branch \times Loan \times Borrower) \Pi_{Branch.br\_name = loan.br\_name \wedge loan.loan\_no = branch.loanno})$

▷ JOINS IN RELATIONAL ALGEBRA :

- Retrieve the name of manager of each dept.

DEPT_MGR ← Department $\bowtie_{MGRSSN = SSN}$ Employee

operator join condition

▷ TYPES OF JOIN :

① THETA JOIN :

$R \bowtie_{theta} S$

- The join condition is called theta.
- Theta can be any general boolean expression on the attributes of R & S.

Date

② EQUI JOINS

- Such a join where the only comparasion operator used is "=" is called an Equi join.

③ NATURAL JOINS

- Denoted by "*"
- The join attributes must have the same name in each relation.
- If this is not the case, a renaming operation is applied first.

DEPT_LOCS ← DEPARTMEN * DEPT_Locations

(it will join dep_no of both tables)

joining attribute not included in output.

"DERIVED OPERATORS"

① DIVISION OPERATOR:

jaha se data extract krna hai wo numerator mein. jiski help se nikalna hai wo denominator

A(x,Y) / B(Y) = C

results

E(Sid, Cid) / C(Cid) = H

Sid	Cid
S_1	C_1
S_2	C_1
S_3	C_2
S_3	C_2

Enrolled

Cid
C_1
C_2

Course

- ye division operator, C.cid k corresponding saari we values dega jo Enrolled mein hongi. Yani we students id ~~jinka~~ jo C_1 & C_2 done mein enrolled hai we milj jayenge

- Retrieve Sid of students who enrolled in every course

Page No.

Output → S_1

Date _____ Day _____

→ $R_1 \div R_2$ = Tuples of R_1 associated with all tuples of R_2.

→ $R_1 \div R_2$ is possible iff $R_2 \subset R_1$

S_2 & S_3 doesn't appear in output because they both are not associated with all values of C.cid. S_2 is not enrolled in C_2 & S_3 not in C_1.

Subka subke sath relation banjayega.

ENR_COURSE ← (π_{sid}(Enrolled) x π_{cid}(course))

NOT_ENR_COUR ← (π_{sid}(ENR_COURSE - Enrolled))

↳ getting sid of those student who are not enrolled in all course or for those student enrolled in any one course only.

ENR_STD_ONLY ← π_{sid}(Enrolled) - NOT_ENR_COUR

↳ now subtracting with enroll to get students enrolled in all course

E(Sid, Cid) / C(Cid)

Date

▷ AGGREGATE FUNCTION:

↳ SUM, AVERAGE, MAXIMUM, MINIMUM, COUNT

$\mathcal{F}$ MAX SALARY (Employee) → retrive max salary

$\mathcal{F}$ COUNT SSN, AVERAGE SALARY (Em) → computes count of employees and their avg salary.

Dno	Count ssn	Average Salary
5	4	33250
4	3	31000
1	1	55000

D_{no} $\mathcal{F}$ COUNT SSN, AVERAGE salary (Employee)

▷ Recursive Query:

BORG_SSN ← π_{ssn} ($\sigma_{Pname = 'James' \text{ AND } Pname = 'Borg'}$ (Emp))

SUPERVISION (Ssn1, Ssn2) ← $\pi_{ssn, super\_ssn}$ (Employee)

↳ retrieve all employees with their supervisors.

RESULT_1 ← π_{ssn1} (SUPERVISION $\bowtie_{ssn2 = ssn}$ BORG_SSN)

↳ retrieving supervisee of Borg

RESULT_2 ← π_{ssn1} (SUPERVISION $\bowtie_{ssn2 = ssn}$ RESULT1)

↳ Now getting supervisee of these supervisee that were supervised by Borg.

Date _____ Day _____

RESULT ← RESULT1 ∪ RESULT2

↓

Now combining all the supervisees of, Supervised by Borg Directly or Indirectly.

▷ TYPES OF OUTERJOIN :

(1) Left Join → ⟕
(2) Right Join → ⟖
(3) Full Join → ⟗

Q: Retrieve the name and address of all employees who work for the 'Research' department.

RESEARCH_DEPT ← $\sigma_{DNAME = 'Research'}$ (Department)

RESEARCH_EMPS ← (RESEARCH_DEPT ⋈ Employees)
Dnumber = DNo Employee

RESULT ← $\pi_{FNAME, LNAME, ADDRESS}$ (RESEARCH_EMPS)