Set Intersection ($\cap$)

- Any relation is a set.
- Similar to intersection operation in set theory
- It is a binary operator
- It is a set of all objects that are a member of both A and B
- It is denoted by $\cap$.
- Syntax: $\Pi_{Column}(Relation\_1) \cap \Pi_{column}(Relation\_2)$

Exple: $R=\{A, B, C, D, E, F\}$, $S=\{A, B, \$, E, F, 1, \#\}$

$$R \cap S = \{A, B, E, F\}$$

Example 1: Find the names of all customers who have deposited money and also availed loan

$\Pi_{cu\text{-}name}(Depositor) \cap \Pi_{cu\text{-}name}(Borrower)$

Output: $R \cap S$

Cu_Name
Rose
John

Exple 2: Find the set of all courses taught in the Fall 2022 semester and the Spring 2022 semester

Note: $R \cap S = R-(R-S)$ Exo: Proof this

Assignment

Are the following RA operations commutative $R \cap S$, $R \cup S$ and $R-S$

Assignment Operation (=)

- Similar to assignment operator in PLs
- Denoted by = or ←
- It is a binary operator
- The database might be modified if assignment to a permanent relation is made.
- Useful in the situation where it is required to write relational algebra expressions by using temporal relation variable
- Example: $P = R \cap S$
- It provides a convenient way to express complex queries
- Example

$$Temp1 = Expression1$$
$$Temp2 = Expression2$$
$$Result = Temp1 \div Temp2$$

OR

$$Temp\ 1 \leftarrow R \times S$$
$$Temp\ 2 \leftarrow \pi_{course\_id}(\sigma_{semester = \text{"Spring"}}(Section))$$
$$Result \leftarrow \pi_{cvs}(Temp2)$$

The JOIN Operation (⋈)

- Combines related tuples from 2 relations into a single relation

Syntax: $R \bowtie_{\langle join\ condition \rangle} S$

Example: Consider the 2 tables below

DEPARTMENT

DName	DNo	Mgr_SSN
Research	2	123456
Finance	5	612135

EMPLOYEE

SSN	FName	LName	DNo
132147	Alex	Smith	2
123456	Ashu	Mercy	2
612135	Ngwa	Memy	5
321716	Fred	Mbah	5

1) Retrieve the details of the manager of each department

$Dept\_Mgr \leftarrow DEPARTMENT \bowtie_{Mgr\_SSN = SSN} EMPLOYEE$

OR

$Temp \leftarrow DEPARTMENT \times EMPLOYEE$

$Dept\_Mgr \leftarrow \sigma_{Mgr\_SSN = SSN} (Temp)$

Note: The JOIN Operation is equivalent to a cartesian Product followed by a selection operation

The THETA Join (θ)

< join condition: $A_i \; \theta \; B_j$

Syntax: $R \bowtie_{A_i \theta B_j} S$

$A \Rightarrow$ Attribute of R

$B \Rightarrow$ Attribute of S

$\theta \Rightarrow \{=, <, <=, >, >=, \neq\}$

R

A_1	A_2
20	25
80	40

S

B_1
50
35

Find $R \bowtie_{A_2 = B_1} S$ $\quad R \bowtie_{A_2 > B_1} S$

$R \bowtie_{A_2 < B_1} S$ $\quad R \bowtie_{A_2 >= B_1} S$

$R \bowtie_{A_2 <= B_1} S$ $\quad R \bowtie_{A_2 \neq B_1} S$

$R \bowtie_{A_2 > B_1} S$:

A_1	A_2	B_1
80	40	35

The EQUIJOIN Operation

- The only comparison operator used is "=".

e.g $Dept\_Mgr \leftarrow DEPARTMENT \bowtie_{Mgr\_SSN = SSN} EMPLOYEE$

The NATURAL JOIN Operation (*)

- Can be performed only if there is a common attribute in between the relations.

Exple:

PROJECT

PID	PName	DNum
101	ProjectX	1
102	ProjectY	2
103	ProjectZ	2

DEPARTMENT

DNo	Mgr_SSN
1	123456
2	612345

$Proj\text{-}Dept \leftarrow PROJECT * \rho_{(DNum,\ Mgr\_SSN)}(DEPARTMENT)$

Give the result of this expression

Note: The natural join is an equijoin followed by the removal of unnecessary attributes.

The DIVISION Operation (÷)

Exple: Retrieve the employee ID (EID) of the employees working on all projects

EMPLOYEE ↑ R

EID	PID
1001	1
1002	1
1002	2
1003	2
X	Y

PROJECT ↑ S

PID
1
2

Keyword here is **ALL**

$Res \leftarrow EMPLOYEE \div PROJECT$

Output Res: EID

1002

OR

$T_1 \leftarrow \Pi_X(R)$

$T_2 \leftarrow \Pi_X((T_1 \times S) - R)$

$Res \leftarrow T_1 - T_2$

$\Pi_X(R)$: T_1

EID
1001
1002
1003

$T_1 \times S$

EID	PID
1001	1
1001	2
1002	1
1002	2
1003	1
1003	2

$(T_1 \times S) - R$

EID	PID
1001	2
1003	1

T_2

$\Pi_X((T_1 \times S) - R)$

EID
1001
1003

$Res \leftarrow T_1 - T_2$

EID
1002

Additional Relational Operations

- Aggregate Function ($\mathcal{F}$):

Syntax:

$<grouping\_attributes> \mathcal{F}_{<function\_list>}(R)$

<grouping_attributes> => list of attributes in R

Exple: Retrieve the number of employees and their average salary.

Employee	FName	SSN	Salary	DNo
	Ann	1234	40000	2
	Jeremy	3142	30000	2
	Peter	2134	30000	1
	Elsa	4123	20000	2

$\mathcal{F}_{COUNT_{SSN},\, AVERAGE_{Salary}}(EMPLOYEE)$

Output:

COUNT_SSN	AVERAGE_Salary
4	30000

Exercise: Retrieve the number of employees and their average salary in each department

- OUTER JOIN Operation

i) Left Outer Join ⟕

R	A	B
	1	a
	2	c

S	C	D
	1	b
	3	d

Res	A	B	C	D
	1	a	1	b
	2	c	null	null

$Res \leftarrow R ⟕_{A=C} S$

ii) Right Outer Join: ⟖

$Res \leftarrow R ⟖_{A=C} S$

Res	A	B	C	D
	1	a	1	b
	null	null	3	d

iii) Full Outer Join: ⟗

$Res \leftarrow R ⟗_{A=C} S$

Res	A	B	C	D
	1	a	1	b
	2	c	null	null
	null	null	3	d