

# RELATIONAL CALCULUS

- ⇒ Relational Calculus is a non-procedural query language.
- ⇒ It uses mathematical predicate calculus (or first-order logic) instead of algebra.
- ⇒ The relational calculus tells what to do but never explains how to do.

## Types of Quantifiers used :-

### i) Universal Quantifiers -

The universal quantifier denoted by  $\forall$  is read as for all which means that in a given set of tuples exactly all tuples satisfy a given condition.

### ii) Existential Quantifiers -

The existential quantifier denoted by  $\exists$  is read as there exists which means that in a given set of tuples there is atleast one occurrence

## FREE VARIABLE AND BOUND VARIABLE :-

- ⇒ A tuple variable is said to be free variable unless it is quantified by a  $\exists$  or  $\forall$ .
- ⇒ A tuple variable is said to be bound variable if it is quantified by  $\forall$  or  $\exists$  (quantifiers).



# Types of Relational Calculus :-

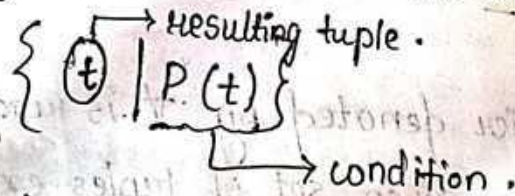
i) Tuple Relational Calculus.

ii) Domain Relational Calculus

## i) Tuple Relational Calculus :-

⇒ It is specified to select the tuples in a relation. In TRC, (tuple relational calculus), filtering variable uses the tuple of a relation.

⇒ The result of the relation can have one or more tuples.



⇒ It is a set of all tuples  $t$  such that, predicate  $P$  is true for  $t$  where  $t$  is the resulting tuple,  $P(t)$  denotes predicate or condition use to fetch the condition tuple.

## Predicate Calculus Formula :-

i) Set of attribute and constraint.

ii) Set of comparison operation.

iii) Set of connectivity. ( $\wedge, \vee, \neg$ )

iv) Implication ( $x \Rightarrow y$ )

v) Quantifiers ( $\forall, \exists$ )



For Example :-

$\{ T.name \mid \text{Author}(T) \text{ AND } T.article = 'database' \}$

Author(T)

T.id	T.Name	T.article
1	Ram	database
2	Shyam	Network
3	Hari	database
4	Geeta	Network

⇒ TRC can be quantified. In TRC, we can use Existential Quantifiers and Universal Quantifiers ( $\forall$ ).

$\{ R \mid \exists T \in \text{Authors} (T.article = 'database' \text{ AND } R.name = T.name) \}$

Q Find the loan no., branch name and amount for all loans of over 1200.

$\{ \text{Branch name} \mid \text{Loan}(T) \mid \{ \text{Branch} \mid \{ t \mid t \in \text{Loan} \wedge t[Amount] > 1200 \} \}$

$\{ R \mid \forall T \in \text{Loan} ($

$\Rightarrow \{ \text{Branch name} \mid \text{Loan}(T), \text{Amount} \mid \text{Loan}(T) \}$

$\Rightarrow \{ \text{Branch name} \}$

$\Rightarrow \{ t \mid t \in \text{Loan} \text{ AND } t[Amount] > 1200 \}$ .

Q Find the loan no for each loan of amount  $> 1200$ .

$\{ R \mid \exists T \in \text{Loan} (t[Amount] > 1200) \wedge (R.name = (Amount) > 1200 \text{ and } R.Loan\_no = T.Loan\_no) \}$



First Name	Last Name	Address	City	State	Zip	Phone	Age	Gender	Marital Status	Occupation	Education	Religion	Political Party	Other
John	Doe	123 Main St	New York	NY	10001	(212) 555-1234	35	M	Married	Software Engineer	Bachelor's	Catholic	Democrat	
Jane	Doe	456 Oak Ave	Los Angeles	CA	90001	(310) 555-5678	28	F	Single	Marketing Executive	Master's	Protestant	Republican	
Robert	Smith	789 Pine Rd	Chicago	IL	60601	(312) 555-9012	42	M	Married	Teacher	Bachelor's	Jewish	Democrat	
Mary	Johnson	101 Elm St	San Francisco	CA	94101	(415) 555-3456	55	F	Married	Retired Nurse	High School	Catholic	Democrat	
David	Wilson	202 Maple Dr	Seattle	WA	98101	(206) 555-7890	30	M	Single	Graphic Designer	Bachelor's	Buddhist	Democrat	
Linda	Green	303 Birch Ln	Portland	OR	97201	(503) 555-2345	40	F	Married	Accountant	Master's	Methodist	Democrat	
Michael	Brown	404 Cedar St	Phoenix	AZ	85001	(602) 555-6789	38	M	Married	Sales Representative	Bachelor's	Anglican	Republican	
Patricia	White	505 Spruce Ave	Denver	CO	80201	(303) 555-0123	45	F	Married	Librarian	Bachelor's	Catholic	Democrat	
Christopher	Black	606 Ash Rd	San Diego	CA	92101	(619) 555-4567	32	M	Single	IT Support	Bachelor's	Evangelical	Republican	
Sarah	Gray	707 Hickory Dr	San Jose	CA	95101	(408) 555-8901	25	F	Single	Product Manager	Master's	Unitarian	Democrat	
James	Miller	808 Walnut St	San Antonio	TX	78201	(214) 555-2345	48	M	Married	Construction Worker	High School	Catholic	Democrat	
Karen	Wright	909 Chestnut Ave	San Jose	CA	95101	(408) 555-6789	33	F	Married	Human Resources	Bachelor's	Protestant	Democrat	
Thomas	King	1010 Sycamore Ln	San Jose	CA	95101	(408) 555-0123	50	M	Married	Retired Engineer	High School	Catholic	Democrat	
Elizabeth	Scott	1111 Magnolia Dr	San Jose	CA	95101	(408) 555-4567	27	F	Single	UX Designer	Master's	Jewish	Democrat	
William	Young	1212 Dogwood St	San Jose	CA	95101	(408) 555-8901	43	M	Married	Software Tester	Bachelor's	Anglican	Democrat	
Amanda	Allen	1313 Redwood Ave	San Jose	CA	95101	(408) 555-2345	31	F	Married	Business Development	Bachelor's	Protestant	Democrat	
Richard	Evans	1414 Cypress Rd	San Jose	CA	95101	(408) 555-6789	47	M	Married	Operations Manager	Bachelor's	Catholic	Democrat	
Michelle	Turner	1515 Juniper Ln	San Jose	CA	95101	(408) 555-0123	29	F	Single	Project Manager	Master's	Unitarian	Democrat	
Steven	Phillips	1616 Fir St	San Jose	CA	95101	(408) 555-4567	36	M	Married	Systems Administrator	Bachelor's	Anglican	Democrat	
Angela	Carter	1717 Laurel Ave	San Jose	CA	95101	(408) 555-8901	26	F	Single	Quality Assurance	Bachelor's	Protestant	Democrat	
Gregory	Morgan	1818 Hawthorn Dr	San Jose	CA	95101	(408) 555-2345	41	M	Married	Network Engineer	Bachelor's	Catholic	Democrat	
Deborah	Baker	1919 Boxwood St	San Jose	CA	95101	(408) 555-6789	34	F	Married	Customer Support	Bachelor's	Anglican	Democrat	
Anthony	Nelson	2020 Yew Ln	San Jose	CA	95101	(408) 555-0123	37	M	Married	IT Consultant	Master's	Protestant	Democrat	
Christina	Harris	2121 Alder Ave	San Jose	CA	95101	(408) 555-4567	24	F	Single	Business Analyst	Bachelor's	Unitarian	Democrat	
Timothy	Clark	2222 Beech Rd	San Jose	CA	95101	(408) 555-8901	44	M	Married	Operations Supervisor	Bachelor's	Catholic	Democrat	
Stephanie	Lewis	2323 Elm St	San Jose	CA	95101	(408) 555-2345	23	F	Single	Marketing Coordinator	Bachelor's	Anglican	Democrat	
Jonathan	Roberts	2424 Oak Ave	San Jose	CA	95101	(408) 555-6789	39	M	Married	Software Developer	Master's	Protestant	Democrat	
Kimberly	Walker	2525 Pine Ln	San Jose	CA	95101	(408) 555-0123	22	F	Single	Product Designer	Bachelor's	Unitarian	Democrat	
Mark	Young	2626 Spruce St	San Jose	CA	95101	(408) 555-4567	46	M	Married	Operations Manager	Bachelor's	Catholic	Democrat</	

$$\Rightarrow \{x \mid x \in \text{Personen} \wedge (\text{Personen.name} = \text{T.customer.name}) \text{ AND } \text{Personen.name} = \text{T.customer.name}\} \text{ AND } \{x \mid x \in \text{Personen} \wedge (\text{Personen.name} = \text{"Günther"}) \text{ AND } \text{Personen.name} = \text{"Günther"}\}$$

Q Find the names of all customers having a loan, an account, or both at the bank.

$$\begin{aligned} & \exists t \mid \exists i \in \text{borrower} \quad (+[\text{customer} - \text{name}] = s[\text{loan}]) \\ & \wedge \exists i \in \text{depositors} \quad (+[\text{customer} - \text{name}] = u[\text{loan}]) \end{aligned}$$



## Domain Relational Calculus :-

- ⇒ The second form of relation is known as Domain relational calculus.
- ⇒ In domain relational calculus, filtering variable uses the domain of attributes.
- ⇒ Domain relational calculus uses the same operators as tuple calculus.
- ⇒ It uses logical connectives  $\wedge$  (and),  $\vee$  (or) and  $\neg$  (not). It uses Existential ( $\exists$ ) and Universal ( $\forall$ ) Quantifiers to bind the variable.

### Notation -

⇒  $\{ \langle a_1, a_2, \dots, a_n \rangle \mid P(a_1, a_2, \dots, a_n) \}$  where  $a_1, a_2$  represents domain variables.

⇒  $P$  represents a predicate similar the predicate calculus.

⇒  $\langle a_1, a_2, \dots, a_n \rangle \in r$ , where  $r$  is the relation on  $n$  attributes and  $a_1, a_2, \dots, a_n$  are domain variable or domain constants.

Q Find Loan-no, Branch-name and Amount for loans over 1200.

⇒  $\{ \langle l, b, a \rangle \mid \langle l, b, a \rangle \in \text{loan} \wedge a > 1200 \}$

Q Find the Loan-no of each loan of an amount greater than 1200.

⇒  $\{ \langle l \rangle \mid \exists b, a (\langle l, b, a \rangle \in \text{loan} \wedge a > 1200) \}$

Q Find the name of all customer's who have a loan of over 1200.

⇒  ~~$\{ \langle c \rangle \mid \exists s, c, t (\langle c, s, t \rangle \in \text{loan}) \}$~~

⇒  ~~$\{ \langle c\text{-name} \rangle \mid \exists (l, b, a) \in \text{borrow} \wedge \langle l \rangle \mid \exists (l, b, a) \in \text{loan} \wedge$~~



$\Rightarrow \{ \langle c \rangle \mid \exists l, b, a (\langle c, l \rangle \in \text{borrow} \wedge \langle l, b, a \rangle \in \text{loan} \wedge a > 1200) \}$

S1			R1		
sid	name	age	sid	bid	date
22	m	45	22	101	today
31	y	55	58	103	tomw
58	z	35			

S1.sid	name	age	R1.sid	bid	date

$\Theta = S1 \bowtie_{S1.sid < R1.sid} R1$

$\text{Eqw}^* = S1 \bowtie_{S1.sid = R1.cid} R1$

Inner Join : —

An inner join includes only those tuples that satisfy the matching criteria, while rest of the tuples are excluded.

General Join : —

A Join operation with such general condition called  $\Theta$  join / Inner join :  $\{ \langle, \rangle, <, >, =, \neq \}$

Eqw Join : —

When the join condition involves with equality comparison then it is called Eqw join.

Dept  $\bowtie_{\text{Mgrcode} = \text{Empcode}}$  Emp

Emp (Empcode, name, address, salary)

Dept (Deptno, Dname, Summary, Empcode)

We need to find all the managers of Dept having salary > 3000.

Dept  $\bowtie$

Emp.salary > 3000



## JOINS :-

⇒ It is a binary operation which takes two relation as an input

⇒ The join operation will check the key attribute is present in both the relation or not. If present then it will retrieve the required information from both the tables by comparing the value of the common key attribute.

⇒ It is denoted by  $\bowtie$

⇒ A join operation combine related tuples from different relation if and only if an even join condition is satisfied

⇒ Types of join :-

- i) Natural Join
- ii) Outer Join
  - Left outer Join
  - Right outer Join
  - Full outer Join
- iii) Equi Join.

### i) Natural Join :-

⇒ A natural join is the set of tuples of all combination in R and S that are equal on their common attribute name.

⇒ It is denoted by  $\bowtie$ .

⇒ It natural join operation first perform cartesian, then it forms a selection forcing equality on those attribute that appear in both relation and finally removes duplicate attribute

Q Select Ename from employee natural join department where employee Empno = department Empno.

Dept  $\bowtie$  Emp



7 Employee -

Empno	Ename	Address
1	A	Delhi
2	B	Mumbai
3	C	Mumbai
4	D	Delhi

7 Department -

Deptno	Dname	Empno
D <sub>1</sub>	HR	1
D <sub>2</sub>	IT	2
D <sub>3</sub>	Marketing	4

Empno	Ename	Address	Deptno	Dname	Empno
✓ 1	A	Delhi	D <sub>1</sub>	HR	1
1	A	Delhi	D <sub>2</sub>	IT	2
1	A	Delhi	D <sub>3</sub>	Marketing	4
2	B	Mumbai	D <sub>1</sub>	HR	1
✓ 2	B	Mumbai	D <sub>2</sub>	IT	2
2	B	Mumbai	D <sub>3</sub>	Marketing	4
3	C	Mumbai	D <sub>1</sub>	HR	1
3	C	Mumbai	D <sub>2</sub>	IT	2
3	C	Mumbai	D <sub>3</sub>	Marketing	4
4	D	Delhi	D <sub>1</sub>	HR	1
4	D	Delhi	D <sub>2</sub>	IT	2
✓ 4	D	Delhi	D <sub>3</sub>	Marketing	4

⇒ E D D



## ii) Outer Join :-

⇒ The outer join operation is an extension of the join operation. It returns all the records from both the tables that satisfy the join condition.

⇒ In other words, these join will not return only the matching records but also return the all unmatched row from one or both tables.

⇒ Types : a) Left Outer Join  
b) Right Outer Join  
c) Full Outer Join.

### a) Left Outer Join -

⇒ The left outer join retrieves all the records from the left table and matching row from right table.

⇒ It will return null when no matching record is found in right side table.

⇒ Employee :-

Ename	Street	City
Ram	Civil Line	Mumbai
Shyam	Park Lane	Kolkata
Ravi	M.G. Street	Delhi
Hari	Nehru Place	Hyderabad

⇒ Workers :-

Ename	Branch	Salary
Ram	Infosys	10000
Shyam	Wipro	20000
Ravi	HCL	30000
Hari	TCS	40000

End



EName	Street	City	EName	Branch	Salary
Ram	CivilLine	Mumbai	Ram	Infosys	10000
Ram	CivilLine	Mumbai	Shyam	Wipro	20000
Ram	CivilLine	Mumbai	Kuber	HCL	30000
Ram	CivilLine	Mumbai	Hari	TCS	40000
Shyam	Parklane	Kolkata	Ram	Infosys	10000
Shyam	Park Lane	Kolkata	Shyam	Wipro	20000
Shyam	Parklane	Kolkata	Kuber	HCL	30000
Shyam	Park Lane	Kolkata	Hari	TCS	40000
Ravi	M.G. Street	Delhi	Ram	Infosys	10000
Ravi	M.G. Street	Delhi	Shyam	Wipro	20000
Ravi	M.G. Street	Delhi	Kuber	HCL	30000
Ravi	M.G. Street	Delhi	Hari	TCS	40000
Hari	Nehru place	Hyderabad	Ram	Infosys	10000
Hari	Nehru place	Hyderabad	Shyam	Wipro	20000
Hari	Nehru place	Hyderabad	Kuber	HCL	30000
Hari	Nehru place	Hyderabad	Hari	TCS	40000

⇒ E MW :-

EName	Street	City	Branch	Salary
Ram	CivilLine	Mumbai	Infosys	10000
Shyam	Parklane	Kolkata	Wipro	20000
Ravi	M.G. Street	Delhi	NULL	NULL
Hari	Nehru Place	Hyderabad	TCS	40000



### b) Right Outer Join -

⇒ The right out retrieve all records from right hand table and matched row from right hand table.

⇒ It will return null when no matching record is found in the left hand table.

⇒ E  $\bowtie$  W :-

Ename	Street	City	Branch	Salary
Ram	CivilLine	Mumbai	Infosys	10000
Shyam	Park Lane	Kolkata	Wipro	20000
Kuber	NULL	NULL	HCL	30000
Hari	Nehru place	Hyderabad	TCS	40000

### c) Full Outer Join -

⇒ The full outer join returns a result that includes all rows from both tables

⇒ The column's of right hand table return NULL when no matching record are found in the left hand table and if no matching records are found in right hand table then left hand table returns NULL.

⇒ E  $\bowtie$  W

Ename	Street	City	Branch	Salary
Ram	CivilLine	Mumbai	Infosys	10000
Shyam	Park Lane	Kolkata	Wipro	20000
Kuber	NULL	NULL	HCL	30000
Hari	Nehru place	Hyderabad	TCS	40000
Ravi	M.G. Street	Delhi	NULL	NULL



### iii) Equi Join :-

= It is also known as Inner Join. It is the most common join. It is based on matching data as per equality condition.

= The equi join uses comparison operator (=) equally.

Employee

Eno	EmpName	Permanent address
1.	A	Delhi
2.	B	Mumbai
3.	C	Mumbai
4.	D	Delhi

Dept

DNo	Work Location	ENo
D <sub>1</sub>	Delhi	1
D <sub>2</sub>	Pune	2
D <sub>3</sub>	Patna	4

Q Find the name of the employee who work in a department having permanent address same as their work location.

=> EmpName - A

Eno	EmpName	Permanent Address	DNo	Work Location	ENo
1	A	Delhi	D <sub>1</sub>	Delhi	1
1	A	Delhi	D <sub>2</sub>	Pune	2
1	A	Delhi	D <sub>3</sub>	Patna	4
2	B	Delhi Mumbai	D <sub>1</sub>	Delhi	1
2	B	Delhi Mumbai	D <sub>2</sub>	Pune	2
2	B	Delhi Mumbai	D <sub>3</sub>	Patna	4
3	C	Mumbai	D <sub>1</sub>	Delhi	1
3	C	Mumbai	D <sub>2</sub>	Pune	2
3	C	Mumbai	D <sub>3</sub>	Patna	4
4	D	Delhi	D <sub>1</sub>	Delhi	1
4	D	Delhi	D <sub>2</sub>	Pune	2
4	D	Delhi	D <sub>3</sub>	Patna	4