RELATIONAL FUNCTION

⇒ Relational database systems are expected be equipped with a query language that assist the users to query the database instance.

Two types of Query Language :-

1) Procedural Query Language -

= In procedural query language the user instruct the system to perform a series of operations to produce desired result.

= Ex :- Relational Algebra (RA) → SQL.

2) Non-Procedural Query Language :-

= User instruct the system to produce desired result, without telling the step by step process.

= Ex :- Relational Calculus (RC) → QBE
(Query By Example)

Relational Algebra :-

⇒ Relational algebra is a procedural query language.

⇒ It gives a step-by-step process to obtain the result of the query.

⇒ It uses operators to perform the query.

⇒ Types of Relational Operation –

i) Select Operation.
ii) Project Operation
iii) Union Operation
iv) Set Intersection
v) Set difference
vi) Cartesian product
vii) Rename Operation

i) Select Operation :-

= The select operation selects tuples that satisfy a given predicate/ condition.

= It is denoted by sigma (σ).

= Notation : $\sigma_{p(r)}$

Where :

- σ is used for selection prediction.
- r is used for relation.
- p is used as a propositional logic formula which may use connectors like : AND, OR, NOT.

⇒ These relational can use as relational operators like $=, \neq, <, \leq, >, \geq$

Q fetch details of student where age should greater than 18:

$\sigma P(r)$.

$= \sigma$ age > 18 (Student)

For example: LOAN Relation: Retrieve all details of loan where branch_nam
perryside.

Branch_name	Loan_No	Amount
Downtown	L-17	1000
Redwood	L-23	2000
Perryride	L-15	1500
Downtown	L-14	1500
Mianus	L-13	500
Roundhill	L-11	900
Perrynide	L-16	130

Input:

σ BRANCH_NAME = "Perryside" (LOAN)

Output:

Branch_Name	Loan_No	Amount
Perryride	L-15	1500
Perryside	L-16	1300

ii) Projection Operation :-

=> This operation shows the list of those attributes that we wish to appear in the result.

=> Rest of the attributes are eliminated from the table.

=> It is denoted by π.(π)

=> Notation : $\pi A_1, A_2, \ldots A_n(r)$

Q Retrieve name of student whose roll_no is 2.

Ans : σ rollno = 2 (Student)

Output -

RollNo	Name	Age.
2	B	20

=> $\pi_{name}(\sigma$ rollno = 2 (Student))

Example - Customer Relation.

Rollno	Name	Age
1	A	20
2	B	21
3	A	19

iii) UNION OPERATION :-

= Suppose there are two relations R and S. The union operation contains all the tuples that are either in R or S or both in R and S.

= It eliminates the duplicate tuples. It is denoted by ∪.

= Notation : R ∪ S

A union operation must hold the following condition:

i) R and S must have the attribute of the same number.

ii) Duplicate tuples are eliminated automatically.

Depositor Relation -

CUSTOMER_NAME	ACCOUNT_NO
Johnson	A-101
Smith	A-121
Mayes	A-321
Turner	A-176
Johnson	A-273
Jones	A-472
Lindsay	A-284

Borrower Relation -

CUSTOMER_NAME	LOAN_NO
Jones	L-17
Smith	L-23
Hayes	L-15
Jackson	L-14
Curry	L-93
Smith	L-11
Williams	L-17

Π Customer_name (Borrow) Π Customer_name (Depositor)

Custome

iv) Intersection Operation :-

= Suppose there are two relations R and S. The set intersection operation contains all tuples that are in both R and S.

= It is denoted by intersection ∩.

= Notation : R ∩ S

= We can use the deposition and borrower table for this operation.

v) Set Difference :-

= Suppose there are two relations R and S. The set intersection operation contains all tuples that are in R but not in S.

= It is denoted by intersection minus (−).

= Notation : R − S

= Example : Using the above depositor table and borrow table.

ii) Cartesian Product :-

⇒ The Cartesian product is used to combine each row in one table with each row in the other table.

⇒ It is also known as a Cross product.

⇒ It is denoted by X.

⇒ Notation : E X D.

Employee:

Emp-id	Emp_name	Emp_dept
1	Smith	A
2	Harry	B
3	John	C

Department:

Dept_no	Dept_name
A	Marketing
B	Sales
C	Legal

⇒ Cartesian Product : Employee X Department.

Emp_id	Emp_name	Emp_dept	Dept_no	Dept_name
1	Smith	A	A	Marketing
1	Smith	A	B	Sales
1	Smith	A	C	legal.
2	Harry	B	A	Marketing
2	Harry	B	B	Sales
2	Harry	B	C	Legal
3	John	C	A	Marketing
3	John	C	B	Sales
3	John	C	C	Legal

vii) RENAME OPERATION :-

⇒ The rename operation is used to rename the output relation. It is denoted by rho(ρ)

⇒ Examples :

We can use the rename operator to rename Student relation to Student1.

- ρ (STUDENT1, STUDENT).

Example:

Branch (branch_name, branch_city, assets).

Customer (customer_name, customer_street, customer_city)

Account (account_no, branch_name, balance).
(f)

Loan (loan_no, branch_name, amount).
(f)

Deposition (customer_name, account_no).
f f

Borrower (customer_name, loan_no).
f f

Loan Table :-

Loan_no	Branch_name	Amount
L-11	Round-Hill	900
L-14	Downtown	1500
L-15	Perryridge	1500
L-16	Perryridge	1300
L-17	Downtown	1000
L-23	Redwood	2000

Questions :-

Q1) Find all loans made at "Perryridge" branch.

⇒ ~~Select from loan.~~

⇒ Select * from loan where branch_name = Perryridge.

$\sigma_{Branch\_name = "Perryridge"}(LOAN)$

↳ Selection Operations.

Q2) Find all loan of over 1200.

$\sigma_{amount > 1200}(loan)$.

Q3) Find all tuple who have taken loans of more than 1200 made by the "Perryridge" branch.

⇒ $\sigma_{Branch\_name = "Perryridge"}(LOAN) \wedge \sigma_{amount > 1200}(loan$

Q4) Find all loan numbers and the amount of the loan.

⇒ ~~$\Pi_{(\sigma_{loan\_no = amount})}$~~ $\Pi_{loan\_number, amount(loan)}$

Q5) Find the loan number of each loan of an amount greater than 1200.

~~$\sigma_{amount > 1200}(Loan)$~~.

$\Pi_{loan\_no}(\sigma_{amount > 1200}(Loan))$.

Q6) Find the names of all customers who have a loan, an account or both from the bank.

⇒ Π Customer_name (Borrower) $\cup$ Π Customer_name (Depositor).

Q7) Find the names of all customers who have a loan and an account of bank.

⇒ Π Customer_name (Borrower) $\cap$ Π Customer_name (Depositor)

Q8) Find the names of all customers who have an account but no loan from the bank.

⇒ Π Customer_name (Borrower) $-$ Π Customer_name (Depositor).

Q9) Find the names of all customers who have a loan at the pennyridge branch.

⇒ ~~Π Customer_name (Borrower)~~

~~Π Borrower~~

⇒ σ Borrower.loan_no = loan.loan_no (borrower $\times$ loan).

⇒ Π Customer name (σ branch_name = "pennyridge" (σ borrower.loan_no = loan.loan_no (borrower $\times$ loan))).

Division Operation (÷ or /) :—

The division operator in relational algebra finds all values in one table that are related to every value in another table.

or

The division operator in relational algebra is a binary operator used to retrieve tuple from one relation (dividend) that are associated with all or every tuples in another relation.

~~R1 ÷ R2 = Tuples of R1 associated with all tuples of R2~~

1) R1 ÷ R2 is possible if and only if R2 ⊂ R1 (R2 is a proper subset of R1)

2) R2 ⊂ R1 → R1 ≠ R2 & every attribute of R2 should be present in R1

Student (R1)

Stdname	CourseName
Tom	DBMS
John	DS
Tom	DS
Tom	CN
John	DBMS
Amy	CN
Amy	DBMS
Amy	DS

Course (R2)

CName
DBMS
DS
CN

3) The relation referenced by division operator will have attributes = (All attributes of A − All attributes of B)

R1 ≠ R2 · yes. Because 2 attributes in Students and 1 attribute in course.

Q. Finding students who are enrolled in all Course.

o/p

Tom
Amy

i.e R1 ÷ R2

Project Assignment

Employee	Project
A	P1
A	P2
A	P3
B	P1
B	P2
C	P1
C	P2
C	P3

Q2

All project

Project
P1
P2
P3

Find employees who are working on all projects listed in all projects.

A
C

Q Find the names of all customer who have a loan at perryridge branch but do not have an account of any branch of the bank.

Fundamental Operations :-

1) Selection Operation (σ) → row / tuple
2) Projection Operation (π) → column / attribute
3) Union ($\cup$)
4) Set Difference ($-$)
5) Cartesian Product ($\times$)

— o —

Branch

Branch-Name	Branch City	assets
Brighton	Brooklyn	7100000
Downtown	B "	900000
Mianus		