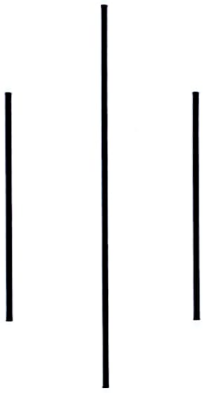
TRIBHUVAN UNIVERSITY

PATAN MULTIPLE CAMPUS

PATAN DHOKA, LALITPUR



SUBJECT: DATABASE MANAGEMENT SYSTEM (BIT 202)

Lab Title : Implementation of Relational Algebra

SUBMITTED BY

NAME: DOLINDRA BAHADUR RAUT
CLASS: BIT – II / I
ROLL NO.: 21
DATE: 2081/12/03

SUBMITTED TO

DEO NARAYAN YADAV

.....
CHECKED BY

2081/12/3

4. Use of the SELECT operation

4.1 Introduction:-

SELECT operation is done by using "selection" operator which is represented by "sigma" (σ). It is used to retrieve tuples (rows) from the relation/table where the given condition is satisfied. It is a unary operator means it requires only one operand.

4.2 Syntax:-

$$\sigma_{<condition>} (Relation\_name).$$

4.3 Examples:-

a) Select the EMPLOYEE tuples whose department is 40, or whose salary is greater than Rs. 50,000.

⇒ Solution:-

$$\sigma_{Dno = 40 \ V \ salary > 50,000} (EMPLOYEE)$$

b) Select the tuples for all employees who either work in department 4 and salary over Rs 25,000, or working in department 5 and salary over Rs. 30,000.

⇒ Solution:-

$$\sigma_{(Dno = 4 \wedge salary > 25,000) V (Dno = 5 \wedge salary > 30,000)} (EMPLOYEE)$$

c) Select all the employees whose salary is between 20,000 to 55,000 (including both).

⇒ solution:

$$\sigma_{salary \geq 20,000 \wedge salary \leq 55,000}(EMPLOYEE)$$

d) Select all employees who either work in department 5 or having salary > 25,000

⇒ solution:

$$\sigma_{Dno = 5 \vee salary > 25,000}(EMPLOYEE)$$

e) Lists all the "dependent" table's tuples whose relationship is equal to Daughter.

⇒ solution:

$$\sigma_{Relationship = \text{"Daughter"}}(DEPENDENT)$$

2. Use of the "PROJECT" operation

2.1 Introduction :-

Projection operator (Π) is used to do project operation. It is denoted by the uppercase Greek letter pi (Π), and used to retrieve certain attributes (columns) from the table/relation.

2.2 Syntax :-

$$\Pi_{<Attribute\ 1,\ Attribute\ 2,\ --->}\ (Relation\_name)$$

2.3 Examples :-

a) List all employee's first name, last name and salary.

⇒ solution:

$$\Pi_{Fname,\ Lname,\ Salary}\ (EMPLOYEE)$$

b) Find the name of departments.

⇒ solution :-

$$\Pi_{Dname}\ (DEPARTMENT)$$

c) List the name of all projects.

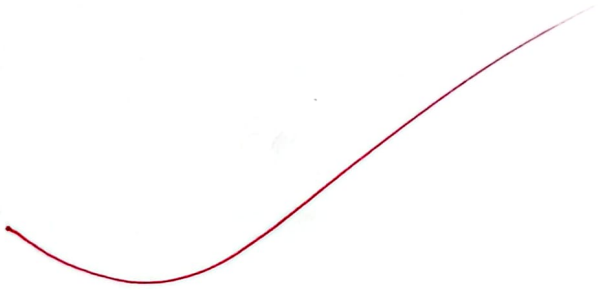
⇒ solution :-

$$\Pi_{pname}\ (PROJECT)$$

d) List all pnumber, plocation from "PROJECT" relation.

⇒ solution :-

$$\Pi_{Pnumber,\ plocation}\ (PROJECT)$$



3. Use of SELECT and PROJECT operation.

3.1 Introduction :-

SELECT and PROJECT operation can be combined when it requires certain attributes from the relation that can be already filtered.

Basically, when we need certain columns which is already filtered, then we use both. In this case, SELECT is used to filter tuples from table and project is used to retrieve certain attributes from such filtered tuples.

3.2 Syntax :-

$$\Pi_{\langle Attribute1, Attribute2 \ldots \rangle} \left(\sigma_{\langle condition \rangle} (Relation\_name) \right)$$

3.3 Examples :-

a) Retrieve the first name, last name, and salary of all employees who work in department number 5.

⇒ solution :-

$$\Pi_{Fname, Lname, salary} \left(\sigma_{Dno = 5} (EMPLOYEE) \right)$$

b) Retrieve all the project's name which comes under department number 5.

⇒ Solution :-

$$\Pi_{pname} \left(\sigma_{Dnum = 5} (PROJECT) \right)$$

4. Use of UNION, INTERSECTION and MINUS operation :-

4.1 UNION :-

4.1.1 Introduction :-

Union operation is done by "Union operator" which is represented by "∪". It is a binary operator meaning that it requires two relations/ tables as operand.
one thing is to be noted that your Union of two relations/ tables, both relations must have the same set of Attributes.

4.1.2 syntax :-

$$\Pi_{column\_name}(Relation1) \cup \Pi_{column\_name}(Relation2)$$

4.1.3 Examples :-

(a) Retrieve the social security numbers of all employee who either work in department 5 or directly supervise an employee who works in department 5.

⇒ solution :-

$$T_1 \leftarrow \Pi_{Ssn}(\sigma_{Dno=5}(EMPLOYEE))$$

$$T_2 \leftarrow \Pi_{Mgr\_ssn}(\sigma_{Dnumber=5}(DEPARTMENT))$$

$$\therefore T_1 \cup T_2$$

This can be also written as,

$$\Pi_{Ssn}(\sigma_{Dno=5}(EMPLOYEE)) \cup \Pi_{Mgr\_ssn}(\sigma_{Dnumber=5}(DEPARTMENT))$$

4.2 INTERSECTION :-

4.2.1 Introduction :-

The "INTERSECTION" operation is done by using '∩' - Intersection symbol. It is a binary operator which means it takes two relations/ tables as input.

The INTERSECTION operation gives the common set of values from two tables.

4.2.2 Syntax :-

$$\prod_{column\_Name}(Relation1) \cap \prod_{column\_Name}(Relation2)$$

4.2.3 Examples :-

a)

4.3 MINUS (set Difference) operation

4.3.1 Introduction :-

MINUS is a binary operation. so It is used to find tuples that are in one relation but are not in another. For instance, R−S means tuples in R but not in S.

It is denoted by minus (−) operater.

4.3.2 syntax :

$$\Pi_{\text{column\_Name}}(\text{Relation 1}) - \Pi_{\text{column\_Name}}(\text{Relation 2})$$

4.3.3 Examples :

q) Retrieve the names of employees who have no dependents.

⇒ solution:

$$T_1 \leftarrow \Pi_{\text{Fname, Lname}}(\text{EMPLOYEE})$$

$$T_2 \leftarrow \Pi_{\text{Fname, Lname}}(\text{EMPLOYEE} \bowtie_{\text{Ssh = ESsh}} \text{DEPENDENT})$$

$$T_1 - T_2$$

5. Use of CARTESIAN PRODUCT operation.

5.1 Introduction :-

"CARTESIAN PRODUCT" operation combines every tuples of one tables with every tuples of another table, producing all possible combination.

5.2 syntax :-

$$T_1 \times T_2$$

where,

T_1 = ~~first~~ relation/table

T_2 = second relation/table

The symbol '×' is used to denote the CARTESIAN PRODUCT.

5.3 Examples :-

a) Retrieve a list of names of each female employee's dependents.

⇒ solution :-

$$T_1 \leftarrow \sigma_{Sex = 'F'}(EMPLOYEE)$$

$$\pi_{\text{~~Fname, Lname~~ Dependent_name}}\left(\sigma_{T_1.SSN = DEPENDENT.Essn}(T_1 \times DEPENDENT)\right)$$

6. Use of the JOIN operation

6.1 Introduction:-

JOIN operation is a binary operation that allows us to combine two or more relations. Joins are of two types: Inner Join and outer Join. Inner Join is further classified into three types: Theta Join, Equi Join, and Natural Join. outer Join also consists of three types: Left outer Join, Right outer Join, and Full outer Join.

6.2 syntax:-

- ~~Let~~ Let suppose A and B be two tables, then,
- Theta Join :- $A \bowtie_{A_i \, \theta \, B_j} B$
- EQUI Join :- $A \bowtie_{A_i = B_j} B$
- Natural Join :- ~~A⋈B~~ $A * B$
- Left outer Join :- $A ⟕ B$
- Right outer Join :- $A ⟖ B$
- Full outer Join :- $A ⟗ B$

6.3 Examples:-

a) Retrieve the name of the manager of each department.

⇒ Solution:-

$$\Pi_{Fname, Lname} \left(EMPLOYEE \underset{EMPLOYEE.Ssh = DEPARTMENT.Mgr\_ssh}{\bowtie} DEPARTMENT \right)$$