CHAPTER 12

RELATIONAL ALGEBRAIC OPERATIONS

The relational algebra is a procedural query language. It consists of a set of operations that takes one or two relations as input and produce a new relation as their result. The operations are :

- Projection(π)
- Selection(σ)
- Union (∪)
- Set intersection(∩)
- Set difference (-)
- Cartesian product (×)
- Rename(ρ)
- Join(⋈)
- Division (÷)
- Assignment (←)

The projection, selection and rename are called unary operations and other are called Binary Operations.

12.1 Unary Operations

12.1.1 Projection(π-PI)

The projection operation is represented by the symbol PI(π). A projection of all its tuples over some set of attributes. Suppose the user wishes to display employee number, name and salary of the relation EMP, then for the same the SQL command is :

SELECT EMPNO, ENAME, SAL FROM EMP;

$\pi_{EMPNO, ENAME, SAL}(EMP)$

Attributes to appear in the result as a subscript to π. The argument relation follows in the parentheses. The projection operation(vertical subset of a relation) is used to either reduce the number of attributes in the resultant relation or to reorder attributes. In the resultant relation the arity(degree or number of attributes) is reduced.

12.1.2 Selection or Restriction Operation (σ - SIGMA)

The selection operation, represented by the symbol σ(sigma). This is an operation that selects only some of the tuples of the relation depends upon conditions(predicate). The predicate appears as a subscript to σ(. The argument relation is in parenthesis after the σ. In general, we allow comparisons using =, ≠, <, ≤, >, ≥ in the selection predicate.

We can combine several predicates into a larger predicate by using the connectives and(∧), or(∨) and not(¬).

Selection operation is the horizontal subset of a relation i.e. the action is defined over the complete set of attribute but only a subset of the tuples are included in the result.

This operation is also referred as Restriction Operation.

Suppose, to find the details of employees in the relation EMP whose salary is less than 2000. The SQL command is

SELECT * FROM EMP WHERE SAL<2000;

Which can be represented as

$\sigma_{SAL<2000}(EMP)$

- Suppose, to find the details of employees in the relation EMP whose salary is less than 2000 and job is CLERK . The SQL command is

SELECT * FROM EMP WHERE SAL<2000 AND JOB='CLERK';

Which can be represented as

$\sigma_{SAL<2000 \wedge JOB="CLERK"}(EMP)$

12.1.3 Composition of Relation Operations

To display the employee number, name and salary whose salary is less than 2000. The command will be

SELECT EMPNO, ENAME, SAL FROM EMP WHERE SAL<2000;

Which can be represented as

$\pi_{EMPNO, ENAME, SAL}(\sigma_{SAL < 2000})(EMP)$

12.2 Set Operation

The Selection and Projection operations extract information from only one relation. There are obviously cases where we would like to combine information from several relations. There are cases where we would like to combine information from several relations. The binary operations of the relational algebra stating, the set operations Union, difference, intersection and product

12.2.1 Union ($\cup$)

The union operation is used to view the tuples which certify by either or both predicates. To display the job of all employees who are working under the department number 10, the required command is

SELECT JOB FROM EMP WHERE DEPTNO=10;

Same, to display the job of all employees who are working under the department number 30 the required command is

SELECT JOB FROM EMP WHERE DEPTNO=30;

But, if the user needs the job of all employees who are working under the department number 10 or 30, then we have to joint the previous two command and make a single command with the help of UNION operation like,

```
SELECT JOB FROM EMP WHERE DEPTNO=10
UNION
SELECT JOB FROM EMP WHERE DEPTNO=30;
```

Which can be represented as

$\pi_{JOB}(\sigma_{DEPTNO=10}(EMP) \cup \sigma_{DEPTNO=30}(EMP))$

12.2.2 Intersection ($\cap$)

The intersection operation is used to view the tuples which certify by both predicates.

Suppose, the user needs the job of all employees who are working under the department number 10 and 30, then the command is,

```
SELECT JOB FROM EMP WHERE DEPTNO=10
INTERSECTION
SELECT JOB FROM EMP WHERE DEPTNO=30;
```

Which can be represented as

$\pi_{JOB}(\sigma_{DEPTNO=10}(EMP) \cap \sigma_{DEPTNO=30}(EMP))$

12.2.3 Set Difference (-)

The set difference operation is used to find tuples that are in one relation(or query) but not in another. Suppose, the user needs the job of all employees who are working under the department 10 and not in department 30, then the command is,

```
SELECT JOB
FROM EMP WHERE DEPTNO=10
MINUS
SELECT JOB
FROM EMP WHERE DEPTNO=30;
```

Which can be represented as

$\pi_{JOB}(\sigma_{DEPTNO=10}(EMP) - \sigma_{DEPTNO=30}(EMP))$

12.2.4 Cartesian-Product (×)

the Cartesian-product operation represented by a cross(×).This operator is used to combine information from two relations. In the result relation contains all combination of two operand relations tuples. The Cartesian Product of relation R and S is written as R×S

Suppose the command

SELECT * FROM EMP, DEPT;

Then it display 56 rows i.e. each combination of emp (14 rows) and dept(4 rows).

12.3 Rename Operation

The rename operation represented by the symbol Rho(ρ). In the database every relations have one name. But the resultant relation of a relational-algebra expressions do not have any name. So, it is impossible to use the resultant relation at the later part of the expression. Using rename operator we can give one name to that expression.

Consider one relational algebra expression = E. the expression

$\rho_x(E)$

return the result of expression E under the name X.

12.4 Join Operations(⋈)

The join operator, as the name suggests, allows the combining of two relations to form a single new relation. this is one binary operation that allows us to combine certain selection and a Cartesian product into one operation

The join operation, represented by the symbol ⋈.

Types of join operator

1. Theta join
2. Equi join
3. Natural join

12.4.1 Theta Join (θ-join)

The Theta join operation defines a relation that contains tuples satisfying the predicate F from the Cartesian product or R and S. The predicate F is of the form R.ai θ S.bi where θ may be one of the comparison operations (<,<=,>,>=,!=).We can rewrite the Theta join in terms of the basic Selection and Cartesian product operations

$R \bowtie_F S = \sigma_F (R \times S)$

12.4.2 Equi Join

The join operation forms a Cartesian product of its two arguments, performs a selection forcing equality (=) on those attributes that appear in both relation schemas, and finally removes duplicate attributes.

Suppose the user want to view to all employee number, name and the name of the department he works. The SQL command is

SELECT EMPNO, ENAME, DNAME FROM EMP, DEPT

WHERE EMP.DEPTNO=DEPT.DEPTNO

which can be represented as

$(\pi_{EMPNO,ENAME}(EMP)) \bowtie_{EMP.DEPTNO=DEPT.DEPTNO} (\pi_{DNAME}(DEPT))$

12.4.3 Natural join

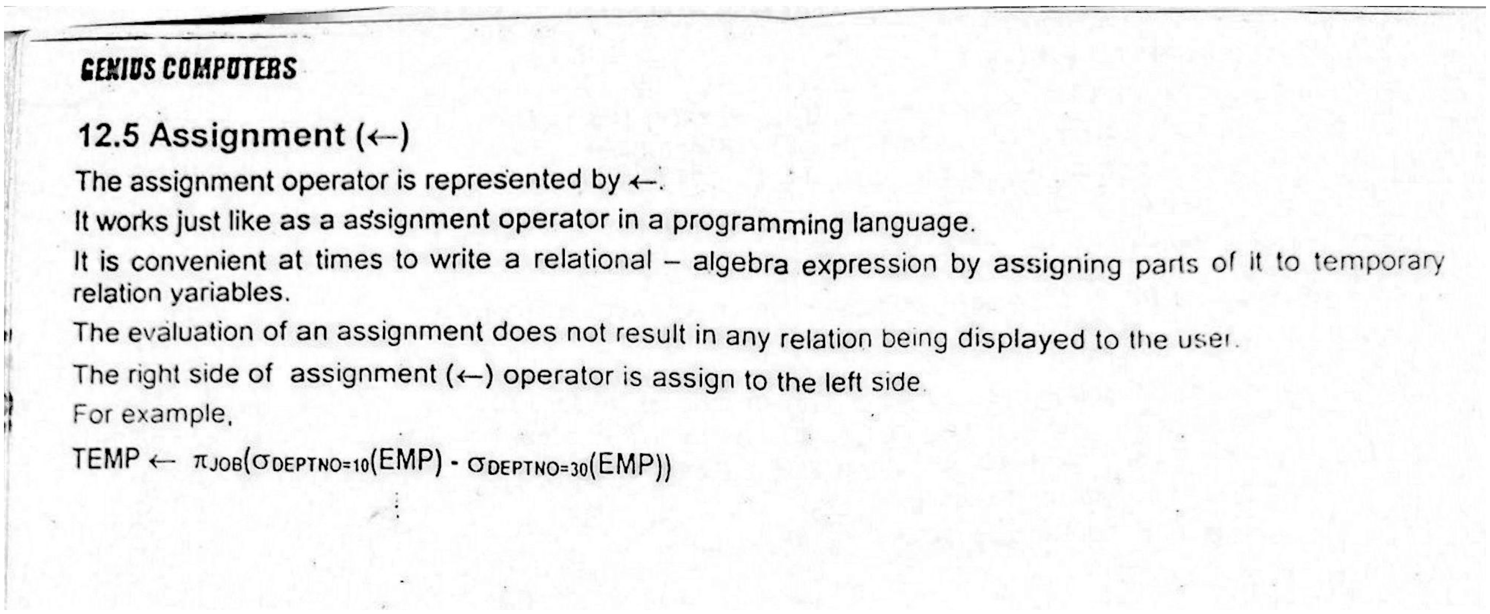
The Natural join is an Equijoin of the two relations R and S over all common attributes x. One occurrence of each common attribute is eliminated from result.

$R \bowtie S$

The Natural join operations an Equijoin over all the attributes in the two relations that have the same name. The degree of a Natural join is the sum of the degrees of the relations R and S less the number of attributes in x.

List the employee number, name, department no and department name of all employees.

$(\pi_{empno,ename,deptno}(EMP)) \bowtie (\pi_{deptno,dname}(DEPT))$



12.5 Assignment (←)

The assignment operator is represented by ←.

It works just like as a assignment operator in a programming language.

It is convenient at times to write a relational – algebra expression by assigning parts of it to temporary relation variables.

The evaluation of an assignment does not result in any relation being displayed to the user.

The right side of assignment (←) operator is assign to the left side.

For example,

$TEMP \leftarrow \pi_{JOB}(\sigma_{DEPTNO=10}(EMP) - \sigma_{DEPTNO=30}(EMP))$

⋮