

Database CS 33 Unit II Topic - Extended Relational Algebra

Topic - Extended Operators in Relational

Extended operators are those operators which can be derived from basic operators. There are mainly three types of extended operators in Relational Algebra:

Join

Intersection

Divide

The relations used to understand extended operators are STUDENT, STUDENT_SPORTS, ALL_SPORTS and EMPLOYEE which are shown in Table 1, Table 2, Table 3 and Table 4 respectively.

STUDENT

ROLL_NO	NAME	ADDRESS	PHONE	AGE
1	RAM	DELHI	9455123451	18
2	RAMESH	GURGAON	9652431543	18
3	SUJIT	ROHTAK	9156253131	20
4	SURESH	DELHI	9156768971	18

Table 1

STUDENT_SPORTS

ROLL_NO	SPORTS
1	Badminton
2	Cricket
2	Badminton
4	Badminton

Table 2

ALL_SPORTS

SPORTS

Badminton

Cricket

Table 3

EMPLOYEE

EMP_NO	NAME	ADDRESS	PHONE	AGE
1	RAM	DELHI	9455123451	18
5	NARESH	HISAR	9782918192	22
6	SWETA	RANCHI	9852617621	21
4	SURESH	DELHI	9156768971	18

Table 4

Intersection (\cap): Intersection on two relations R1 and R2 can only be computed if R1 and R2 are union compatible (These two relation should have same number of attributes and corresponding attributes in two relations have same domain). Intersection operator when applied on two relations as $R1 \cap R2$ will give a relation with tuples which are in R1 as well as R2.

Syntax:

Relation1 \cap Relation2

Example: Find a person who is student as well as employee- $STUDENT \cap EMPLOYEE$

In terms of basic operators (union and minus) :

$STUDENT \cap EMPLOYEE = STUDENT + EMPLOYEE - (STUDENT \cup EMPLOYEE)$

RESULT:

ROLL_NO	NAME	ADDRESS	PHONE	AGE
1	RAM	DELHI	9455123451	18
4	SURESH	DELHI	9156768971	18

Conditional Join(\bowtie): Conditional Join is used when you want to join two or more relation based on some conditions. Example: Select students whose ROLL_NO is greater than EMP_NO of employees

$STUDENT \bowtie_{>} STUDENT.ROLL_NO > EMPLOYEE.EMP_NO EMPLOYEE$

In terms of basic operators (cross product and selection) :

$\sigma (STUDENT.ROLL_NO > EMPLOYEE.EMP_NO)(STUDENT \times EMPLOYEE)$

RESULT:

ROLL_NO	NAME	ADDRESS	PHONE	AGE	EMP_NO	NAME	ADDRESS
2	RAMESH	GURGAON	9652431543	18	1	RAM	DELHI 9455123451
3	SUJIT	ROHTAK	9156253131	20	1	RAM	DELHI 9455123451
4	SURESH	DELHI	9156768971	18	1	RAM	DELHI 9455123451

Equijoin(\bowtie): Equijoin is a special case of conditional join where only equality condition holds between a pair of attributes. As values of two attributes will be equal in result of equijoin, only one attribute will be appeared in result.

Example: Select students whose ROLL_NO is equal to EMP_NO of employees

$STUDENT \bowtie_{=} STUDENT.ROLL_NO = EMPLOYEE.EMP_NO EMPLOYEE$

In terms of basic operators (cross product, selection and projection) :

$\Pi (STUDENT.ROLL_NO, STUDENT.NAME, STUDENT.ADDRESS, STUDENT.PHONE, STUDENT.AGE \text{ EMPLOYEE.NAME, EMPLOYEE.ADDRESS, EMPLOYEE.PHONE, EMPLOYEE.AGE})(\sigma (STUDENT.ROLL_NO = EMPLOYEE.EMP_NO) (STUDENT \times EMPLOYEE))$

RESULT:

ROLL_NO	NAME	ADDRESS	PHONE	AGE	NAME	ADDRESS	PHONE
1	RAM	DELHI 9455123451	18	RAM	DELHI 9455123451	18	
4	SURESH	DELHI 9156768971	18	SURESH	DELHI 9156768971	18	

Natural Join(\bowtie): It is a special case of equijoin in which equality condition hold on all attributes which have same name in relations R and S (relations on which join operation is applied). While applying natural join on two relations, there is no need to write equality condition explicitly.

Database CS 33 Unit II Topic - Extended Relational Algebra

Natural Join will also return the similar attributes only once as their value will be same in resulting relation.

Example: Select students whose ROLL_NO is equal to ROLL_NO of STUDENT_SPORTS as:

$STUDENT \bowtie STUDENT_SPORTS$

In terms of basic operators (cross product, selection and projection) :

$\Pi(STUDENT.ROLL_NO, STUDENT.NAME, STUDENT.ADDRESS, STUDENT.PHONE, STUDENT.AGE \text{ STUDENT_SPORTS.SPORTS})(\sigma (STUDENT.ROLL_NO=STUDENT_SPORTS.ROLL_NO) (STUDENT \times STUDENT_SPORTS))$

RESULT:

ROLL_NO	NAME	ADDRESS	PHONE	AGE	SPORTS
1	RAM	DELHI	9455123451	18	Badminton
2	RAMESH	GURGAON	9652431543	18	Cricket
2	RAMESH	GURGAON	9652431543	18	Badminton
4	SURESH	DELHI	9156768971	18	Badminton

Natural Join is by default inner join because the tuples which does not satisfy the conditions of join does not appear in result set. e.g.; The tuple having ROLL_NO 3 in STUDENT does not match with any tuple in STUDENT_SPORTS, so it has not been a part of result set.

Left Outer Join(\bowtie): When applying join on two relations R and S, some tuples of R or S does not appear in result set which does not satisfy the join conditions. But Left Outer Joins gives all tuples of R in the result set. The tuples of R which do not satisfy join condition will have values as NULL for attributes of S.

Example: Select students whose ROLL_NO is greater than EMP_NO of employees and details of other students as well

$STUDENT \bowtie_{STUDENT.ROLL_NO > EMPLOYEE.EMP_NO} EMPLOYEE$

RESULT

ROLL_NO	NAME	ADDRESS	PHONE	AGE	EMP_NO	NAME	ADDRESS
2	RAMESH	GURGAON	9652431543	18	1	RAM	DELHI 9455123451
18							
3	SUJIT ROHTAK	9156253131	20	1	RAM	DELHI 9455123451	18

Sanjeev Kumar Sinha 9931917742 MCA course Department of Statistics, P. U.

Database CS 33 Unit II Topic - Extended Relational Algebra

```

4    SURESH    DELHI 9156768971 18    1    RAM    DELHI 9455123451 18
1    RAM    DELHI 9455123451 18    NULL NULL NULL NULL NULL
    
```

Right Outer Join(\bowtie): When applying join on two relations R and S, some tuples of R or S does not appear in result set which does not satisfy the join conditions. But Right Outer Joins gives all tuples of S in the result set. The tuples of S which do not satisfy join condition will have values as NULL for attributes of R.

Example: Select students whose ROLL_NO is greater than EMP_NO of employees and details of other Employees as well

STUDENT \bowtie STUDENT.ROLL_NO>EMPLOYEE.EMP_NOEMPLOYEE

RESULT:

ROLL_NO	NAME	ADDRESS	PHONE	AGE	EMP_NO	NAME	ADDRESS
2	RAMESH	GURGAON	9652431543	18	1	RAM	DELHI 9455123451
3	SUJIT	ROHTAK	9156253131	20	1	RAM	DELHI 9455123451 18
4	SURESH	DELHI	9156768971	18	1	RAM	DELHI 9455123451 18
NULL	NULL	NULL	NULL	5		NARESH	HISAR 9782918192 22
NULL	NULL	NULL	NULL	6		SWETA	RANCHI 9852617621 21
NULL	NULL	NULL	NULL	4		SURESH	DELHI 9156768971 18

Full Outer Join(\bowtie): When applying join on two relations R and S, some tuples of R or S does not appear in result set which does not satisfy the join conditions. But Full Outer Joins gives all tuples of S and all tuples of R in the result set. The tuples of S which do not satisfy join condition will have values as NULL for attributes of R and vice versa.

Example:Select students whose ROLL_NO is greater than EMP_NO of employees and details of other Employees as well and other Students as well

STUDENT \bowtie STUDENT.ROLL_NO>EMPLOYEE.EMP_NOEMPLOYEE

RESULT:

ROLL_NO	NAME	ADDRESS	PHONE	AGE	EMP_NO	NAME	ADDRESS
2	RAMESH	GURGAON	9652431543	18	1	RAM	DELHI 9455123451

18

3	SUJIT	ROHTAK	9156253131	20	1	RAM	DELHI	9455123451	18
4	SURESH	DELHI	9156768971	18	1	RAM	DELHI	9455123451	18
NULL	NULL	NULL	NULL	NULL	5	NARESH	HISAR	9782918192	22
NULL	NULL	NULL	NULL	NULL	6	SWETA	RANCHI	9852617621	21
NULL	NULL	NULL	NULL	NULL	4	SURESH	DELHI	9156768971	18
1	RAM	DELHI	9455123451	18	NULL	NULL	NULL	NULL	NULL

Division Operator (\div): Division operator $A \div B$ can be applied if and only if:

Attributes of B is proper subset of Attributes of A.

The relation returned by division operator will have attributes = (All attributes of A – All Attributes of B)

The relation returned by division operator will return those tuples from relation A which are associated to every B's tuple.

Consider the relation STUDENT_SPORTS and ALL_SPORTS given in Table 2 and Table 3 above.

To apply division operator as

$STUDENT_SPORTS \div ALL_SPORTS$

The operation is valid as attributes in ALL_SPORTS is a proper subset of attributes in STUDENT_SPORTS.

The attributes in resulting relation will have attributes $\{ROLL_NO, SPORTS\} - \{SPORTS\} = ROLL_NO$

The tuples in resulting relation will have those ROLL_NO which are associated with all B's tuple {Badminton, Cricket}. ROLL_NO 1 and 4 are associated to Badminton only. ROLL_NO 2 is associated to all tuples of B. So the resulting relation will be:

ROLL_NO

2