IE301
Analysis and Design of Data Systems

Lecture 17

Relational Algebra

Aram Keryan

November 11, 2015

Introduction

Relational Algebra is a family of algebra used for modelling the data stored in *relational database*, and defining queries on it.

- SQL is based on concepts of Relational Algebra.

Analogy between Relational Algebra and Arithmetic:

Arithmetic is the elementary branch of mathematics that deals with study of *numbers* and properties of *operations* on them, like addition, subtraction, multiplication, and division.

In **Relational Algebra** the analogy of numbers are *relations* and its own set of operations to manipulate with relations, like *select*, *project*, *join* and other.

Unary Relational Operations SELECT & PROJECT

SELECT and PROJECT operations are unary because they operate on single relations.

The SELECT operation is used to choose a *subset of the tuples* from a relation that satisfies a **selection condition**.

Example: Select the EMPLOYEE tuples whose department is 4

$$\sigma_{Dno=4}(EMPLOYEE)$$

Example: Select the EMPLOYEE tuples whose salary is greater than $30,000

$$\sigma_{Salary>30,000}(EMPLOYEE)$$

✔ Result of a SELECT operation is also a relation that has the same attributes as the initial relation.

The SELECT Operation

General denotation of SELECT:

$$\sigma_{<selection\ condition>}(R), \quad \text{where}$$

σ – sigma, denotes SELECT operation
R – name of a relation or a relational algebra expression
$< selection\ condition >$ – is a Boolean expression that is made up of ***clauses*** of the form:

or $< attribute\ name >< comparison\ op >< constant\ value >$
$< attribute\ name >< comparison\ op >< attribute\ name >$

where

$< comparison\ op >$ normally one of the operators$\{=, <, \leq, >, \geq, \neq\}$

✔ the selection operation is applied to *each tuple individually*

The SELECT Operation

Clauses can be connected by the standard Boolean operators and, or, and not to form a general selection condition.

Example: Select the tuples for all employees who either work in department 4 and make over $25,000 per year, or work in department 5 and make over $30,000:

$$\sigma_{(Dno=4\ AND\ Salary>25000)\ OR\ (Dno=5\ AND\ Salary>30000)}(EMPLOYEE)$$

Result:

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5

The SELECT Operation

The SELECT operation is different from the SELECT clause of SQL.

In SQL, the SELECT condition is typically specified in the WHERE clause of a query

Example: Select the tuples for all employees who work in department 4 and make over $25,000 per year

$$\sigma_{Dno=4\ AND\ Salary>25{,}000}(EMPLOYEE)$$

corresponds to:

```
SELECT *
FROM EMPLOYEE
WHERE Dno = 4 AND Salary > 25,000;
```

The PROJECT Operation

The PROJECT operation is used to select certain *attributes* from a relation R.

General denotation of PROJECT:

$$\boldsymbol{\pi}_{<\boldsymbol{attribute\ list}>}(\boldsymbol{R}), \quad \text{where}$$

$\boldsymbol{\pi}$ – pi, denotes PROJECT operation
$\boldsymbol{R}$ – name of a relation or a relational algebra expression
$< \boldsymbol{attribute\ list} >$ – is a sub list of desired attributes from R:

The PROJECT Operation

Example: List each employee's SSN, first and last name and salary

$$\pi_{Ssn,\ Fname,\ Lname,\ Salary}(EMPLOYEE)$$

Result:

Ssn	Fname	Lname	Salary
123456789	John	Smith	30000
333445555	Franklin	Wong	40000
999887777	Alicia	Zelaya	25000
987654321	Jennifer	Wallace	43000
666884444	Ramesh	Narayan	38000
453453453	Joyce	English	25000
987987987	Ahmad	Jabbar	25000
888665555	James	Borg	55000

The PROJECT Operation

If the attribute list includes only non-key attributes of R, duplicate tuples are likely to occur. The PROJECT operation removes any duplicate tuples resulting in valid relation.

Example: List Sex and Salary of employees

$$\pi_{Sex,\,Salary}(EMPLOYEE)$$

Result:

Sex	Salary
M	30000
M	40000
F	25000
F	43000
M	38000
M	25000
M	55000

Since the result of a PROJECT operation is also a relation (that is a set of tuples– all the tuples are distinct) that is why PROJECT operation removes all the duplicates if such exist.

The number of tuples in a relation resulting from a PROJECT operation is always less than or equal to the number of tuples in *R*

The PROJECT Operation

Example: List Sex and Salary of employees

$$\pi_{Sex,\,Salary}(EMPLOYEE)$$

corresponds to:

```
SELECT DISTINCT Sex, Salary
FROM EMPLOYEE;
```

Sequence of operations

Example: List first name, last name and salary of employees who work in department 5.

$$\pi_{Fname,Lname,Salary}(\sigma_{Dno=5}(EMPLOYEE))$$

corresponds to:

```
SELECT Fname, Lname, Salary
FROM EMPLOYEE
WHERE Dno = 5;
```

Rename operation

$$\pi_{Fname,Lname,Salary}(\sigma_{Dno=5}(EMPLOYEE))$$

Result of an above Relational Algebra Expression is an *EMPLOYEE* relation with *Fname, Lname,* and *Salary* attribures :

$$EMPLOYEE(Fname, Lname, Salary);$$

It is sometimes simpler to break down a complex sequence of operations by specifying intermediate result relations than to write a single relational algebra expression. But in this case you would have to give name to each intermediate relation:

Step 1: $\rho_{DEP5\_EMPS}(\sigma_{Dno=5}(EMPLOYEE))$

Step 2: $\pi_{Fname,Lname,Salary}(DEP5\_EMPS)$

Rename operation

(1) $\boldsymbol{\rho_{S(B_1, B_2, \ldots, B_n)}(R)}$, *where*

S – new name of a relation
$B_1, B_2, \ldots, B_n$ - new names of attributes

✔ Renames both relation and attributes

(2) $\boldsymbol{\rho_S(R)}$

✔ Renames only the relation

(3) $\boldsymbol{\rho_{(B_1, B_2, \ldots, B_n)}(R)}$

✔ Renames only the attributes