DATABASE MANAGEMENT SYSTEMS (IT 2040)

LECTURE 05 – RELATIONAL ALGEBRA

LECTURE CONTENT

- Introduction to relational algebra
- Unary operators
- Set operations
- Binary operators
- Aggregate functions and grouping

LEARNING OUTCOMES

- At the end of this lecture students should be able to
  - Write relational algebra statements to cater a given user query

INTRODUCTION TO FORMAL QUERY LANGUAGES

- Relational algebra is the set of operations defined in relational model for manipulating relational databases
- Relational algebra is used as the basis for implementing and optimizing relational database management systems (RDBMS)
- The concepts of relational algebra are incorporated in Structured Query Language (SQL) for RDBMS .

INTRODUCTION TO FORMAL QUERY LANGUAGES (CONTD.)

- Relational algebra consists of a number of operators such as,
  - Unary operators that operate on a single relation
  - Set operators that are based on set theory
  - Binary operators that operate on two tables and
  - Aggregate functions that can summarize data
- These operators can operate on one or more relations and produce a new relation which can be further manipulated using relational algebra.

UNARY OPERATORS – THE PROJECT OPERATOR (Π)

- The project operation, selects certain columns from the table and discards the other columns.
- Duplicate rows are eliminated

EMP

ID	Name	Salary	Dno
123456789	John Smith	30000	5
333445555	Franklin Wong	40000	5
999887777	Alicia Zelaya	25000	4
987654321	Jennifer Wallace	43000	1
666884444	Ramesh Narayan	38000	2
453453453	Joyce English	25000	4

⇨ $\pi_{\text{Name, Salary}}(\text{EMP})$ ⇨

Name	Salary
John Smith	30000
Franklin Wong	40000
Alicia Zelaya	25000
Jennifer Wallace	43000
Ramesh Narayan	38000
Joyce English	25000

UNARY OPERATORS – THE SELECT OPERATOR (Σ)

- The select operation is used to select a subset of tuples form a relation that satisfies a selection condition.
- The *selection condition* = boolean expression with a set of clauses as below connected by AND, OR, NOT boolean operators
- The condition may include =, <, >, ≥, ≤, ≠

EMP

ID	Name	Salary	Dno
123456789	John Smith	30000	5
333445555	Franklin Wong	40000	5
999887777	Alicia Zelaya	25000	4
987654321	Jennifer Wallace	43000	1
666884444	Ramesh Narayan	38000	2
453453453	Joyce English	25000	4

$$\sigma_{Dno=5}(EMP)$$

ID	Name	Salary	Dno
123456789	John Smith	30000	5
333445555	Franklin Wong	40000	5

UNARY OPERATORS – THE SELECT OPERATOR (Σ)

EMP

ID	Name	Salary	Dno
123456789	John Smith	30000	5
333445555	Franklin Wong	40000	5
999887777	Alicia Zelaya	25000	4
987654321	Jennifer Wallace	43000	1
666884444	Ramesh Narayan	38000	2
453453453	Joyce English	25000	4

$\sigma_{\text{Salary>= 40000}}$ (EMP)

ID	Name	Salary	Dno
333445555	Franklin Wong	40000	5
987654321	Jennifer Wallace	43000	1

EMP

ID	Name	Salary	Dno
123456789	John Smith	30000	5
333445555	Franklin Wong	40000	5
999887777	Alicia Zelaya	25000	4
987654321	Jennifer Wallace	43000	1
666884444	Ramesh Narayan	38000	2
453453453	Joyce English	25000	4

$\sigma_{\text{Dno=5 and Salary>= 30000}}$ (EMP)

ID	Name	Salary	Dno
333445555	Franklin Wong	40000	5

UNARY OPERATORS – THE RENAME OPERATOR (**P**)

- In general, several relational algebra operators are applied one after another..
- This could be done by nesting relational operators or it could be applied one operation at a time and creating intermediate relations.
- For the latter case, a name should be given to the relation resulted by applying the first operation before the second operator is applied.
- For example, $\boldsymbol{\pi}_{\text{Name, Salary}}$ $(\boldsymbol{\sigma}_{\text{Dno=5}}$ (EMP)) could be performed with,
  - Dep5_emps $\leftarrow \boldsymbol{\sigma}_{\text{Dno=5}}$ (EMP) OR $\boldsymbol{\rho}_{\text{Dep5\_emp}}(\boldsymbol{\sigma}_{\text{Dno=5}}$ (EMP))
  - RES $\leftarrow \boldsymbol{\pi}_{\text{Name, Salary}}$ (Dep5Dep5_emp_emps)
- where intermediate and result relations are renamed

UNARY OPERATORS – THE RENAME OPERATOR (P)

- Renaming method used in previous slide could be used for renaming attributes in the intermediate and result relations.
- For example,
  - Temp ← $\boldsymbol{\sigma}_{\text{Dno=5}}$ (EMP)
  - R(full_name, sal) ← $\boldsymbol{\pi}_{\text{Name, Salary}}$ (Temp)

OR $\rho_{\text{R(Full\_name,sal)}}$ (σ Dno=5 (EMP))

SET OPERATORS

- **Union compatibility**:Two relations R & S are said to be union-compatible if the following holds:
- degree of R & S are equal, and
- if corresponding fields of R & S, taken from left to right, have the same domains
- Three operations namely, union, intersection and set difference could be performed on two union-compatible relations R & S.

SET OPERATORS – UNION OPERATOR

- The result of this operation, denoted by R ∪ S, is a relation that includes tuples that are either in R or S (or both).

R

ID	Name	Salary	Dno
123456789	John Smith	30000	5
333445555	Franklin Wong	40000	5

S

ID	Name	Salary	Dno
999887777	Alicia Zelaya	25000	4
453453453	Joyce English	25000	4

R U S

123456789	John Smith	30000	5
333445555	Franklin Wong	40000	5
999887777	Alicia Zelaya	25000	4
453453453	Joyce English	25000	4

SET OPERATORS - INTERSECTION OPERATOR

- The result of this operation, denoted by R ∩ S, is a relation that included tuples common to R and S.

R

ID	Name	Salary	Dno
123456789	John Smith	30000	5
333445555	Franklin Wong	40000	5
666884444	Ramesh Narayan	38000	2

S

ID	Name	Salary	Dno
333445555	Franklin Wong	40000	5
453453453	Joyce English	25000	4

R ∩ S

333445555	Franklin Wong	40000	5

SET OPERATORS – SET DIFFERENCE OPERATOR

- The result of this operation, denoted by R – S, is a relation that includes all tuples in R that is not in S.

R

ID	Name	Salary	Dno
123456789	John Smith	30000	5
333445555	Franklin Wong	40000	5
666884444	Ramesh Narayan	38000	2

S

ID	Name	Salary	Dno
333445555	Franklin Wong	40000	5
453453453	Joyce English	25000	4

R - S

123456789	John Smith	30000	5
666884444	Ramesh Narayan	38000	2

BINARY OPERATORS – CARTESIAN PRODUCT (X)

- Cartesian product which is also known as cross product combine every tuple from one relation with every tuple from the other relation.

EMP

ID	Name	Dno
123456789	John Smith	5
666884444	Ramesh Narayan	4
987654321	Jennifer Wallace	4

DEPT

Dnumber	Dname
1	HQ
4	Research
5	Finance

EMP X DEPT

ID	Name	Dno	Dnumber	Dname
123456789	John Smith	5	1	HQ
123456789	John Smith	5	4	Research
123456789	John Smith	5	5	Finance
666884444	Ramesh Narayan	4	1	HQ
666884444	Ramesh Narayan	4	4	Research
666884444	Ramesh Narayan	4	5	Finance
987654321	Jennifer Wallace	4	1	HQ
987654321	Jennifer Wallace	4	4	Research
987654321	Jennifer Wallace	4	5	Finance

BINARY OPERATORS – EQUI JOIN (⋈)

- The join operator ⋈ combine related two tuples from two relations into single tuple. With equi join, the comparison operator used for joining is '='.
- Results of an equi join would have one or more pairs of attributes that have identical values in every tuple.

EMP

ID	Name	Dno
123456789	John Smith	5
666884444	Ramesh Narayan	4
987654321	Jennifer Wallace	4

DEPT

Dnumber	Dname
1	HQ
4	Research
5	Finance

EMP ⋈ Dno=Dnumber DEPT

ID	Name	Dno	Dnumber	Dname
123456789	John Smith	5	5	Finance
666884444	Ramesh Narayan	4	4	Research
987654321	Jennifer Wallace	4	4	Research

BINARY OPERATORS – NATURAL JOIN

- Natural join is a join similar to the equi join but removes the second redundant attribute from the results of an equi join.
- Natural join requires that the two join attributes have the same name in both relations.

EMP

ID	Name	Dno
123456789	John Smith	5
666884444	Ramesh Narayan	4
987654321	Jennifer Wallace	4

DEPT

Dno	Dname
1	HQ
4	Research
5	Finance

EMP * DEPT

ID	Name	Dno	Dname
123456789	John Smith	5	Finance
666884444	Ramesh Narayan	4	Research
987654321	Jennifer Wallace	4	Research

OUTER JOINS

- Join operation ignores the non-matching tuples
- Join attributes with null values are also eliminated
- Some queries require these tuples
- Three new join operators are introduced
    - Left outer join
    - Right outer join
    - Full outer join

LEFT OUTER JOIN (⟕)

- Outer join between relation R and S (R⟕S)
  - Provides all combinations of tuples in R and S that are equal on their join attribute
  - Also returns tuples in R that have no matching tuples in S

Employee

Name	EmpID	Dept
Harry	3415	Finance
Sally	2241	Sales
George	3401	Finance
Harriet	2202	Sales

Dept

DeptName	Mgr
Sales	Harriet

Employee $⟕_{\text{DeptName=DeptName}}$ Dept

Name	EmpID	Dept	DeptName	Mgr
Harry	3415	Finance	Null	Null
Sally	2241	Sales	Sales	Harriet
George	3401	Finance	Null	Null
Harriet	2202	Sales	Sales	Harriet

RIGHT OUTER JOIN (⟖)

- Outer join between relation R and S (R⟖S)
  - Provides all combinations of tuples in R and S that are equal on their join attribute
  - Also returns tuples in S that have no matching tuples in R

Employee

Name	EmpID	Dept
Harry	3415	Finance
Sally	2241	Sales
George	3401	Finance
Harriet	2202	Sales

Dept

DeptName	Mgr
Sales	Harriet
Production	Charles

Employee ⟖$_{\text{DeptName=DeptName}}$ Dept

Name	EmpID	Dept	DeptName	Mgr
Sally	2241	Sales	Sales	Harriet
Harriet	2202	Sales	Sales	Harriet
Null	Null	Null	Production	Charles

FULL OUTER JOIN (⟗)

- Full outer join in effect combines the results of the left and right outer joins

Employee

Name	EmpID	DeptName
Harry	3415	Finance
Sally	2241	Sales
George	3401	Finance
Harriet	2202	Sales

Dept

DeptName	Mgr
Sales	Harriet
Production	Charles

Employee ⟗$_{DeptName=DeptName}$ Dept

Name	EmpID	Dept	DeptName	Mgr
Harry	3415	Finance	Null	Null
Sally	2241	Sales	Sales	Harriet
George	3401	Finance	Null	Null
Harriet	2202	Sales	Sales	Harriet
Null	Null	Null	Production	Charles

AGGREGATE FUNCTIONS

- Aggregate functions are used in simple statistical queries that summarize information from tuples in a relation.
- Common functions applied include SUM, AVERAGE, MAXIMUM, MINIMUM and COUNT.
- Syntax : $\Im_{\text{<function list>}}$ (R)
- Example :

EMP

ID	Name	Salary
123456789	John Smith	30000
333445555	Franklin Wong	40000
999887777	Alicia Zelaya	25000

$\Im_{\text{Count ID ’ Average Salary}}$ (EMP)

3	47500

GROUPING

- A commonly usage of aggregate functions is grouping tuples in a relation by the value of some of their attributes and then applying aggregate functions independently for each group.
- Syntax : <grouping attribute> $\Im$ <function list> (EMP)

EMP

ID	Name	Salary	Dno
123456789	John Smith	30000	5
333445555	Franklin Wong	40000	5
999887777	Alicia Zelaya	25000	4
987654321	Jennifer Wallace	43000	1
666884444	Ramesh Narayan	38000	2
453453453	Joyce English	25000	4

Dno $\Im_{\text{Count ID ,Average Salary}}$ (EMP)

Dno$\Im_{\text{Count ID ,Average Salary}}$ (EMP)

1	1	43000
2	1	38000
4	2	25000
5	2	35000

WHAT YOU HAVE TO DO BY NEXT WEEK

- Try out the self-test questions on the course web.
- Complete the tutorial.