UNIT III
Relational Algebra

Dr. Geetha Mary A
Associate Professor,
SCSE, VITU

Sources:
Pearson Education, Inc. 2011, Elmasri/Navathe, Fundamentals of Database Systems, Sixth Edition
McGraw Hill Education , 2010, Silberschatz, Korth and Sudarshan, Database System Concepts, Sixth edition

Relational Algebra

- The basic set of operations for the relational model is known as the relational algebra.
- These operations enable a user to specify basic retrieval requests.
- The result of a retrieval is a new relation, which may have been formed from one or more relations. The **algebra operations** thus produce new relations, which can be further manipulated using operations of the same algebra.
- A sequence of relational algebra operations forms a **relational algebra expression**, whose result will also be a relation that represents the result of a database query (or retrieval request).

Dr. Geetha Mary A, Asso Prof, SCSE, VIT University

SELECT OPERATION

SELECT operation is used to select a *subset* of the tuples from a relation that satisfy a **selection condition**

Example: To select the EMPLOYEE tuples whose department number is four

$$\boldsymbol{\sigma}_{\textbf{DNO = 4}}\ \textbf{(EMPLOYEE)}$$

To select the EMPLOYEE tuples whose salary is greater than \$30,000

$$\boldsymbol{\sigma}_{\textbf{SALARY > 30,000}}\ \textbf{(EMPLOYEE)}$$

In general: $\boldsymbol{\sigma}_{\text{<selection condition>}}(R)$

where the symbol $\boldsymbol{\sigma}$ (sigma) denotes the select operator

Dr. Geetha Mary A, Asso Prof, SCSE, VIT University

SELECT OPERATION PROPERTIES

- The SELECT operation $\boldsymbol{\sigma}_{<\text{selection condition}>}(R)$ produces a relation S that has the same schema as R
- The SELECT operation $\boldsymbol{\sigma}$ is **commutative;** i.e.,

$$\boldsymbol{\sigma}_{<\text{condition1}>}(\boldsymbol{\sigma}_{<\text{condition2}>}(R)) = \boldsymbol{\sigma}_{<\text{condition2}>}(\boldsymbol{\sigma}_{<\text{condition1}>}(R))$$

- A cascaded SELECT operation **may be applied in any order;** i.e.,

$$\boldsymbol{\sigma}_{<\text{condition1}>}(\boldsymbol{\sigma}_{<\text{condition2}>}(\boldsymbol{\sigma}_{<\text{condition3}>}(R))$$

$$= \boldsymbol{\sigma}_{<\text{condition2}>}(\boldsymbol{\sigma}_{<\text{condition3}>}(\boldsymbol{\sigma}_{<\text{condition1}>}(R)))$$

- A cascaded SELECT operation may be replaced by a single selection with a conjunction of all the conditions**;** i.e.,

$$\boldsymbol{\sigma}_{<\text{condition1}>}(\boldsymbol{\sigma}_{<\text{condition2}>}(\boldsymbol{\sigma}_{<\text{condition3}>}(R))$$

$$= \boldsymbol{\sigma}_{<\text{condition1}>\ \text{AND}\ <\text{condition2}>\ \text{AND}\ <\text{condition3}>}(R)))$$

Dr. Geetha Mary A, Asso Prof, SCSE, VIT University

PROJECT OPERATION

This operation selects certain *columns* from the table and discards the other columns

Example: To list each employee's first and last name and salary, the following is used:

$$\pi_{\text{LNAME, FNAME,SALARY}}(\textbf{EMPLOYEE})$$

The general : π<attribute list>(R)

where π (pi) represents the project operation

RENAME OPERATION

a single **relational algebra expression** by nesting the operations

or

apply one operation at a time and create **intermediate result relations**.

↓

must give names to the relations that hold the intermediate results.

Example: To retrieve the first name, last name, and salary of all employees who work in department number 5,

$$\boldsymbol{\pi_{\text{FNAME, LNAME, SALARY}}(\sigma_{\text{DNO=5}}(\text{EMPLOYEE}))}$$

OR We can explicitly show the sequence of operations, giving a name to each intermediate relation:

$$\textbf{DEP5\_EMPS} \leftarrow \boldsymbol{\sigma_{\text{DNO=5}}(\text{EMPLOYEE})}$$

$$\textbf{RESULT} \leftarrow \boldsymbol{\pi_{\text{FNAME, LNAME, SALARY}}(\text{DEP5\_EMPS})}$$

Dr. Geetha Mary A, Asso Prof, SCSE, VIT University

Unary Relational Operations (cont.)

- **Rename Operation (cont.)**

  The rename operator is ρ

  The general Rename operation can be expressed by any of the following forms:

– $\rho_{S (B1, B2, \ldots, Bn)} (R)$ is a renamed relation S based on R with column names B_1, B_1, …..B_n.

– $\rho_S (R)$ is a renamed relation S based on R (which does not specify column names).

Dr. Geetha Mary A, Asso Prof, SCSE, VIT University

UNION OPERATION

The result of this operation, denoted by R ∪ S, is a relation that includes all tuples that are either in R or in S or in both R and S. Duplicate tuples are eliminated.

Example: To retrieve the social security numbers of all employees who either work in department 5 or directly supervise an employee who works in department 5, we can use the union operation as follows:

$$\textbf{DEP5\_EMPS} \leftarrow \sigma_{\textbf{DNO=5}} \textbf{(EMPLOYEE)}$$

$$\textbf{RESULT1} \leftarrow \pi_{\textbf{SSN}}\textbf{(DEP5\_EMPS)}$$

$$\textbf{RESULT2(SSN)} \leftarrow \pi_{\textbf{SUPERSSN}}\textbf{(DEP5\_EMPS)}$$

$$\textbf{RESULT} \leftarrow \textbf{RESULT1} \cup \textbf{RESULT2}$$

The union operation produces the tuples that are in either RESULT1 or RESULT2 or both. The two operands must be "type compatible".

STUDENT	FN	LN
	Susan	Yao
	Ramesh	Shah
	Johnny	Kohler
	Barbara	Jones
	Amy	Ford
	Jimmy	Wang
	Ernest	Gilbert

INSTRUCTOR	FNAME	LNAME
	John	Smith
	Ricardo	Browne
	Susan	Yao
	Francis	Johnson
	Ramesh	Shah

Student $\cup$ Instructor

FN	LN
Susan	Yao
Ramesh	Shah
Johnny	Kohler
Barbara	Jones
Amy	Ford
Jimmy	Wang
Ernest	Gilbert
John	Smith
Ricardo	Browne
Francis	Johnson

Relational Algebra Operations From Set Theory (cont.)

- **INTERSECTION OPERATION**

The result of this operation, denoted by R ∩ S, is a relation that includes all tuples that are in both R and S. The two operands must be "type compatible"

Example: The result of the intersection operation (figure below) includes only those who are both students and instructors.

FN	LN
Susan	Yao
Ramesh	Shah

SET DIFFERENCE (OR MINUS) OPERATION

The result of this operation, denoted by R - S, is a relation that includes all tuples that are in R but not in S. The two operands must be "type compatible”.

Example: The figure shows the names of students who are not instructors, and the names of instructors who are not students.

STUDENT – INSTRUCTOR

FN	LN
Johnny	Kohler
Barbara	Jones
Amy	Ford
Jimmy	Wang
Ernest	Gilbert

INSTRUCTOR – STUDENT

FNAME	LNAME
John	Smith
Ricardo	Browne
Francis	Johnson

Dr. Geetha Mary A, Asso Prof, SCSE, VIT University

- Both union and intersection are *commutative operations;* that is

  $$\mathbf{R \cup S = S \cup R, \text{ and } R \cap S = S \cap R}$$

- Both union and intersection can be treated as n-ary operations applicable to any number of relations as both are *associative operations;* that is

  $$\mathbf{R \cup (S \cup T) = (R \cup S) \cup T, \text{ and } (R \cap S) \cap T = R \cap (S \cap T)}$$

- The minus operation is *not commutative;* that is, in general

  $$\mathbf{R - S \neq S - R}$$

Dr. Geetha Mary A, Asso Prof, SCSE, VIT University

“After climbing a great hill, one only finds that there are many more hills to climb”

Nelson Mandela

CARTESIAN (OR CROSS PRODUCT) OPERATION

- This operation is used to combine tuples from two relations in a combinatorial fashion.
- The result of $R(A_1, A_2, \ldots, A_n)$ x $S(B_1, B_2, \ldots, B_m)$ is a relation Q with degree n + m attributes $Q(A_1, A_2, \ldots, A_n, B_1, B_2, \ldots, B_m)$, in that order. The resulting relation Q has one tuple for each combination of tuples—one from R and one from S.
- The two operands do NOT have to be "type compatible"

Example:

FEMALE_EMPS ← σ $_{\text{SEX='F'}}$**(EMPLOYEE)**
EMPNAMES ← π $_{\text{FNAME, LNAME, SSN}}$ **(FEMALE_EMPS)**

EMP_DEPENDENTS ← EMPNAMES x DEPENDENT

Dr. Geetha Mary A, Asso Prof, SCSE, VIT University

FEMALE_EMPS	FNAME	MINIT	LNAME	SSN	BDATE	ADDRESS	SEX	SALARY	SUPERSSN	DNO
	Alicia	J	Zelaya	999887777	1968-07-19	3321 Castle,Spring,TX	F	25000	987654321	4
	Jennifer	S	Wallace	987654321	1941-06-20	291 Berry,Bellaire,TX	F	43000	888665555	4
	Joyce	A	English	453453453	1972-07-31	5631 Rice,Houston,TX	F	25000	333445555	5

EMPNAMES	FNAME	LNAME	SSN
	Alicia	Zelaya	999887777
	Jennifer	Wallace	987654321
	Joyce	English	453453453

DEPENDENT	ESSN	DEPENDENT_NAME	SEX	BDATE	RELATIONSHIP
	333445555	Alice	F	1986-04-05	DAUGHTER
	333445555	Theodore	M	1983-10-25	SON
	333445555	Joy	F	1958-05-03	SPOUSE
	987654321	Abner	M	1942-02-28	SPOUSE
	123456789	Michael	M	1988-01-04	SON
	123456789	Alice	F	1988-12-30	DAUGHTER
	123456789	Elizabeth	F	1967-05-05	SPOUSE

Dr. Geetha Mary A, Asso Prof, SCSE, VIT University

EMP_DEPENDENTS	FNAME	LNAME	SSN	ESSN	DEPENDENT_NAME	SEX	BDATE	• • •
	Alicia	Zelaya	999887777	333445555	Alice	F	1986-04-05	• • •
	Alicia	Zelaya	999887777	333445555	Theodore	M	1983-10-25	• • •
	Alicia	Zelaya	999887777	333445555	Joy	F	1958-05-03	• • •
	Alicia	Zelaya	999887777	987654321	Abner	M	1942-02-28	• • •
	Alicia	Zelaya	999887777	123456789	Michael	M	1988-01-04	• • •
	Alicia	Zelaya	999887777	123456789	Alice	F	1988-12-30	• • •
	Alicia	Zelaya	999887777	123456789	Elizabeth	F	1967-05-05	• • •
	Jennifer	Wallace	987654321	333445555	Alice	F	1986-04-05	• • •
	Jennifer	Wallace	987654321	333445555	Theodore	M	1983-10-25	• • •
	Jennifer	Wallace	987654321	333445555	Joy	F	1958-05-03	• • •
	Jennifer	Wallace	987654321	987654321	Abner	M	1942-02-28	• • •
	Jennifer	Wallace	987654321	123456789	Michael	M	1988-01-04	• • •
	Jennifer	Wallace	987654321	123456789	Alice	F	1988-12-30	• • •
	Jennifer	Wallace	987654321	123456789	Elizabeth	F	1967-05-05	• • •
	Joyce	English	453453453	333445555	Alice	F	1986-04-05	• • •
	Joyce	English	453453453	333445555	Theodore	M	1983-10-25	• • •
	Joyce	English	453453453	333445555	Joy	F	1958-05-03	• • •
	Joyce	English	453453453	987654321	Abner	M	1942-02-28	• • •
	Joyce	English	453453453	123456789	Michael	M	1988-01-04	• • •
	Joyce	English	453453453	123456789	Alice	F	1988-12-30	• • •
	Joyce	English	453453453	123456789	Elizabeth	F	1967-05-05	• • •

JOIN OPERATION

- The sequence of cartesian product followed by select is used quite commonly to identify and select related tuples from two relations, a special operation, called **JOIN**. It is denoted by a $\bowtie$
- Allows us to process relationships among relations.
- The general form of a join operation on two relations $R(A_1, A_2, \ldots, A_n)$ and $S(B_1, B_2, \ldots, B_m)$ is:

$$R \bowtie_{\text{<join condition>}} S$$

where R and S can be any relations that result from general *relational algebra expressions.*

JOIN OPERATION

Example: Suppose that we want to retrieve the name of the manager of each department.

$$\textbf{DEPT\_MGR} \leftarrow \textbf{DEPARTMENT} \bowtie_{\textbf{MGRSSN=SSN}} \textbf{EMPLOYEE}$$

DEPT_MGR	DNAME	DNUMBER	MGRSSN	• • •	FNAME	MINIT	LNAME	SSN	• • •
	Research	5	333445555	• • •	Franklin	T	Wong	333445555	• • •
	Administration	4	987654321	• • •	Jennifer	S	Wallace	987654321	• • •
	Headquarters	1	888665555	• • •	James	E	Borg	888665555	• • •

EQUIJOIN OPERATION

- A join with comparison operator used is =, is called an EQUIJOIN.
- As a result of an EQUIJOIN one or more pairs of attributes (whose names need not be identical) that have *identical values* in every tuple.

The JOIN seen in the previous example was EQUIJOIN.

NATURAL JOIN OPERATION

- Because one of each pair of attributes with identical values is superfluous, a new operation called natural join—denoted by *—was created to get rid of the second (superfluous) attribute in an EQUIJOIN condition.
- The standard definition of natural join requires that the two join attributes, or each pair of corresponding join attributes, have the **same name** in both relations. If this is not the case, a renaming operation is applied first.

Example: To apply a natural join on the DNUMBER attributes of DEPARTMENT and DEPT_LOCATIONS, it is sufficient to write:

DEPT_LOCS ← DEPARTMENT * DEPT_LOCATIONS

DEPARTMENT	DNAME	DNUMBER	MGRSSN	MGRSTARTDATE
	Research	5	333445555	1988-05-22
	Administration	4	987654321	1995-01-01
	Headquarters	1	888665555	1981-06-19

DEPT_LOCATIONS	DNUMBER	DLOCATION
	1	Houston
	4	Stafford
	5	Bellaire
	5	Sugarland
	5	Houston

DEPT_LOCS	DNAME	DNUMBER	MGRSSN	MGRSTARTDATE	LOCATION
	Headquarters	1	888665555	1981-06-19	Houston
	Administration	4	987654321	1995-01-01	Stafford
	Research	5	333445555	1988-05-22	Bellaire
	Research	5	333445555	1988-05-22	Sugarland
	Research	5	333445555	1988-05-22	Houston

Dr. Geetha Mary A, Asso Prof, SCSE,
VIT University

Complete Set of Relational Operations

- The set of operations including **select σ, project π , union ∪, set difference - , and cartesian product X** is called a complete set because any other relational algebra expression can be expressed by a combination of these five operations.

- For example:

  $$\mathbf{R \cap S = (R \cup S) - ((R - S) \cup (S - R))}$$

  $$R \bowtie_{<\text{join condition}>} S = \boldsymbol{\sigma}_{<\text{join condition}>} \mathbf{(R\ X\ S)}$$

Dr. Geetha Mary A, Asso Prof, SCSE, VIT University

DIVISION OPERATION

- The division operation is applied to two relations $R(Z) \div S(X)$, where X subset Z. Let $Y = Z - X$ (and hence $Z = X \cup Y$); that is, let Y be the set of attributes of R that are not attributes of S.

- For a tuple t to appear in the result T of the DIVISION, the values in t must appear in R in combination with *every* tuple in S.

The DIVISION operation. (a) Dividing SSN_PNOS by SMITH_PNOS. (b) $T \leftarrow R \div S$.

(a)

SSN_PNOS

ESSN	PNO
123456789	1
123456789	2
666884444	3
453453453	1
453453453	2
333445555	2
333445555	3
333445555	10
333445555	20
999887777	30
999887777	10
987987987	10
987987987	30
987654321	30
987654321	20
888665555	20

SMITH_PNOS

PNO
1
2

SSNS

SSN
123456789
453453453

(b)

R

A	B
a1	b1
a2	b1
a3	b1
a4	b1
a1	b2
a3	b2
a2	b3
a3	b3
a4	b3
a1	b4
a2	b4
a3	b4

S

A
a1
a2
a3

T

B
b1
b4

Dr. Geetha Mary A, Asso Prof, SCSE, VIT University

Aggregate Functions and Grouping

- A type of request that cannot be expressed in the basic relational algebra is to specify mathematical **aggregate functions** on collections of values from the database.

- Examples of such functions include retrieving the average or total salary of all employees or the total number of employee tuples. These functions are used in simple statistical queries that summarize information from the database tuples.

- Common functions applied to collections of numeric values include SUM, AVERAGE, MAXIMUM, and MINIMUM. The COUNT function is used for counting tuples or values.

(a)

R	DNO	NO_OF_EMPLOYEES	AVERAGE_SAL
	5	4	33250
	4	3	31000
	1	1	55000

(b)

DNO	COUNT_SSN	AVERAGE_SALARY
5	4	33250
4	3	31000
1	1	55000

(c)

COUNT_SSN	AVERAGE_SALARY
8	35125

a) $\rho_{R\ (DNO,\ NO\_OF\_EMPLOYEES, AVERAGE\_SAL)}$ $(_{DNO}\ \mathcal{F}_{COUNT\ SSN,\ AVERAGE\ Salary}$ **(Employee))**

b) $_{DNO}\ \mathcal{F}_{COUNT\ SSN,\ AVERAGE\ Salary}$ **(Employee)**

c) $\mathcal{F}_{COUNT\ SSN,\ AVERAGE\ Salary}$ **(Employee)**

Dr. Geetha Mary A, Asso Prof, SCSE, VIT University

Recursive Closure Operations

- An example of a recursive operation is to retrieve all SUPERVISEES of an EMPLOYEE e at all levels—that is, all EMPLOYEE e' directly supervised by e; all employees e" directly supervised by each employee e'; all employees e'" directly supervised by each employee e"; and so on .
- Although it is possible to retrieve employees at each level and then take their union, we cannot, in general, specify a query such as "retrieve the supervisees of 'James Borg' at all levels" without utilizing a looping mechanism.

To specify the SSN's of all employees e' directly supervised – at level one – by the employee e whose name is 'James Borg'

$$\textbf{BORG\_SSN} \leftarrow \pi_{SSN}(\sigma_{\text{FNAME='James' } \textbf{AND} \text{ LNAME='Borg'}}(\textbf{EMPLOYEE}))$$

$$\textbf{SUPERVISION(SSN1,SSN2)} \leftarrow \pi_{\text{SSN, SUPERSSN}}(\textbf{EMPLOYEE})$$

$$\textbf{RESULT1(SSN)} \leftarrow \pi_{\text{SSN1}}(\textbf{SUPERVISION} \bowtie_{\text{SSN2=SSN}} \textbf{BORG\_SSN})$$

To retrive all employees supervised by Borg at level 2 – that is all employees e'' supervised by some employee e' who is directly supervised by Borg

$$\textbf{RESULT2(SSN)} \leftarrow \pi_{\text{SSN1}}(\textbf{SUPERVISION} \bowtie_{\text{SSN2=SSN}} \textbf{RESULT1})$$

Dr. Geetha Mary A, Asso Prof,
SCSE, VIT University

(Borg's SSN is 888665555)

SUPERVISION	SSN1 (SSN)	SSN2 (SUPERSSN)
	123456789	333445555
	333445555	888665555
	999887777	987654321
	987654321	888665555
	666884444	333445555
	453453453	333445555
	987987987	987654321

RESULT 1	SSN
	333445555
	987654321

(Supervised by Borg)

RESULT 2	SSN
	123456789
	999887777
	666884444
	453453453
	987987987

(Supervised by Borg's subordinates)

RESULT	SSN
	123456789
	999887777
	666884444
	453453453
	987987987
	333445555
	987654321

(RESULT1 ∪ RESULT2)

Both set of employees supervised at levels 1 and 2 by 'James Borg'

RESULT ← RESULT1 ∪ RESULT2

Dr. Geetha Mary A, Asso Prof, SCSE, VIT University

The OUTER JOIN Operation

- In NATURAL JOIN tuples without a *matching* (or *related*) tuple are eliminated from the join result. Tuples with null in the join attributes are also eliminated. This amounts to loss of information.
- A set of operations, called outer joins, can be used when we want to keep all the tuples in R, or all those in S, or all those in both relations in the result of the join, regardless of whether or not they have matching tuples in the other relation.
- The left outer join operation keeps every tuple in the *first* or *left* relation R in R ⟕ S; if no matching tuple is found in S, then the attributes of S in the join result are filled or "padded" with null values.
- A similar operation, right outer join, keeps every tuple in the *second* or right relation S in the result of R ⟖ S.
- A third operation, full outer join, denoted by ⟗ keeps all tuples in both the left and the right relations when no matching tuples are found, padding them with null values as needed.

LEFT OUTER JOIN

RESULT	FNAME	MINIT	LNAME	DNAME
	John	B	Smith	null
	Franklin	T	Wong	Research
	Alicia	J	Zelaya	null
	Jennifer	S	Wallace	Administration
	Ramesh	K	Narayan	null
	Joyce	A	English	null
	Ahmad	V	Jabbar	null
	James	E	Borg	Headquarters

TEMP $\leftarrow$ **EMPLOYEE** ⟕ **Dno**=Dnumber **DEPARTMENT**

RESULT $\leftarrow$ π FNAME, MINIT,LNAME,DNAME **(TEMP)**

Dr. Geetha Mary A, Asso Prof, SCSE, VIT University

Examples of Queries in Relational Algebra

- **Q1: Retrieve the name and address of all employees who work for the 'Research' department.**

  RESEARCH_DEPT ← σ DNAME='Research' (DEPARTMENT)

  RESEARCH_EMPS ← (RESEARCH_DEPT $\bowtie$ DNUMBER= DNOEMPLOYEE EMPLOYEE)

  RESULT ← π FNAME, LNAME, ADDRESS (RESEARCH_EMPS)

- **Q6: Retrieve the names of employees who have no dependents.**

  ALL_EMPS ← π SSN(EMPLOYEE)

  EMPS_WITH_DEPS(SSN) ← π ESSN(DEPENDENT)

  EMPS_WITHOUT_DEPS ← (ALL_EMPS - EMPS_WITH_DEPS)

  RESULT ← π LNAME, FNAME (EMPS_WITHOUT_DEPS * EMPLOYEE)

Dr. Geetha Mary A, Asso Prof, SCSE, VIT University