Relational Algebra

Dr. P. Mirunalini
SSNCE

January 02, 2020

Session Objectives

- To learn about the need for Relational Algebra
- To learn about Unary Relational Operations
- To learn about Binary Relational Operations
- To learn about Additional Relational Operations

Relational Algebra

- Relational algebra is the basic **set of operations** for the relational model
- These operations enable a user to specify **basic retrieval requests** (or queries)
- The result of an operation is a **new relation**, which may have been formed from one or more input relations
- **The algebra operations thus produced new relations** can be further manipulated using operations of the same algebra
- A sequence of relational algebra operations forms a **relational algebra expression**

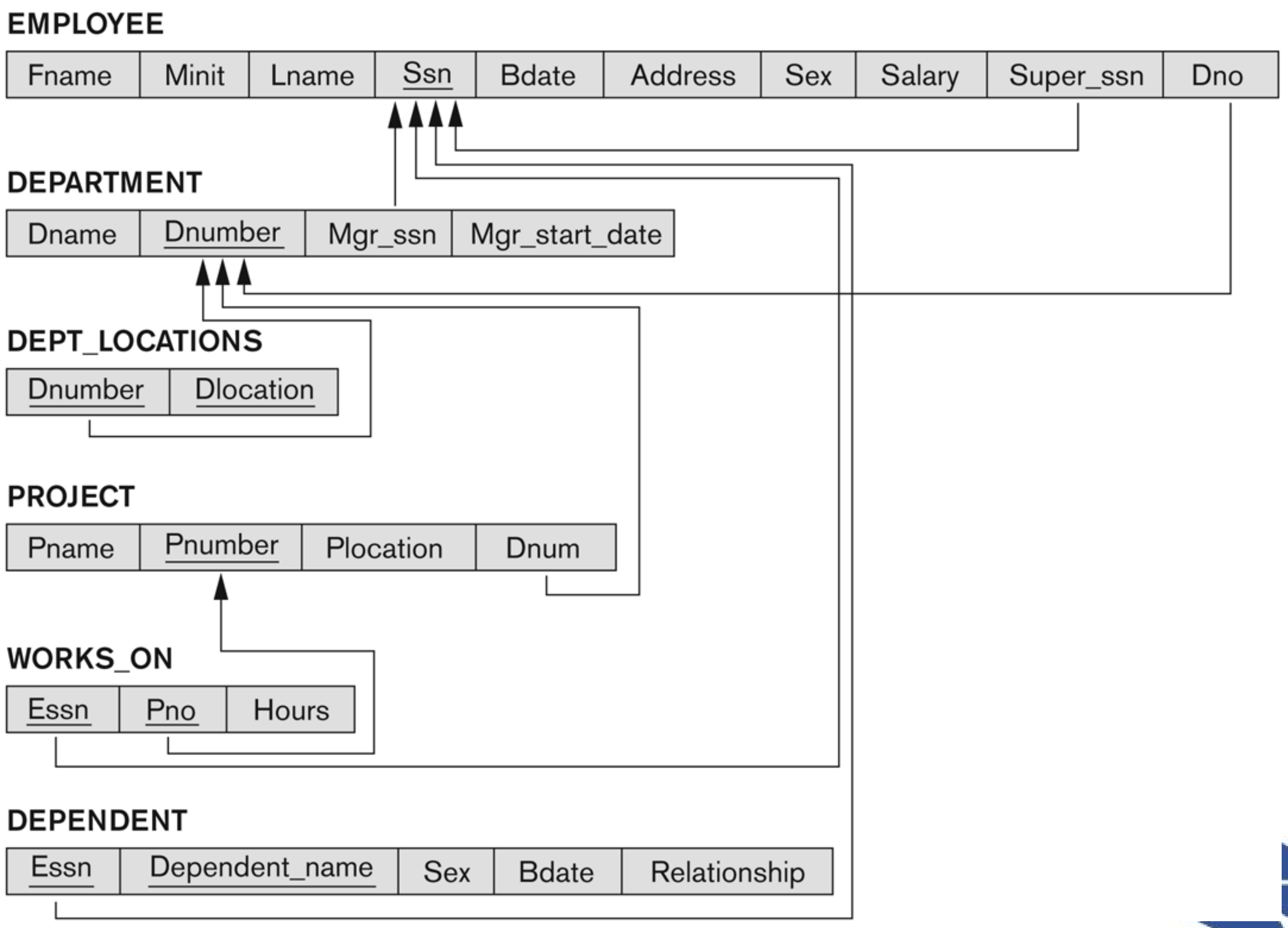
Relational Algebra - Overview

Relational Algebra consists of several groups of operations

- Unary Relational Operations
  - SELECT (symbol: **σ** (sigma))
  - PROJECT (symbol: **π** (pi))
  - RENAME (symbol: **ρ** (rho))
- Relational Algebra Operations From Set Theory
  - UNION (∪), INTERSECTION (∩), DIFFERENCE (or MINUS, **–**)
  - CARTESIAN PRODUCT (**x**)
- Binary Relational Operations
  - JOIN (several variations of JOIN exist)
  - DIVISION
- Additional Relational Operations
  - OUTER JOINS, AGGREGATE FUNCTIONS (These compute summary of information: for example, SUM, COUNT, AVG, MIN, MAX)

COMPANY Database Schema

Figure 5.7
Referential integrity constraints displayed on the COMPANY relational database schema.



Relational Algebra

- Let relation r have attributes X,Y,..,Z,
- Let p be a truth-valued function whose parameters are, some subset of X,Y,..,Z.
- Then the restriction of **r according to p**–
- It is a relation with the same heading as r and with body consisting of all tuples of r such that p evaluates to TRUE for the tuple in question.
- **The expression p is a predicate**

Selection Operation- Example

Relation r

A	B	C	D
α	α	1	7
α	β	5	7
β	β	12	3
β	β	23	10

$\sigma_{A=B \wedge D > 5}(r)$

A	B	C	D
α	α	1	7
β	β	23	10

Relational Algebra- Select Operation

In general, the select operation is denoted by $\sigma_{\text{<selection condition>}}(R)$ where

- The symbol **σ** (sigma) is used to denote the **select operator**
- The selection condition is a **Boolean (conditional) expression** specified on the attributes of relation R
- The selection condition acts as a filter
  - Keeps only those tuples that satisfy the qualifying condition whereas the other tuples are discarded (filtered out)
- Tuples that make the condition **true** appear in the result of the operation
- More than one condition : Clauses can be connected by Boolean operators AND, OR, and NOT.

Examples

1. **Select the EMPLOYEE tuples whose department number is 4:**

$$\sigma_{\text{DNO = 4}}\text{(EMPLOYEE)}$$

2. **Select the employee tuples whose salary is greater than $30,000:**

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

3. **Select the employee tuples whose salary greater then $30,000:**

And whose department no is 4

$$\sigma_{\text{DNO = 4 AND SALARY>3000}}\text{(EMPLOYEE)}$$

4. To **select the EMPLOYEE tuples whose department number is four or those whose salary is greater than $30,000**

$$\sigma_{\text{DNO = 4 OR SALARY > 30,000}}\text{(EMPLOYEE)}$$

SELECT Operation Properties

- The SELECT operation **σ** $_{\textbf{<selection condition>}}$ **(R)** produces a relation **S** that has the same schema (**same attributes**) as R
  - **SELECT σ is commutative: σ $_{A \wedge B}$ (R) = σ $_{B \wedge A}$(R)**
- **A cascade (sequence) of SELECT operations may be applied in any order**:
  - σ<cond1> (σ <cond2> (σ <cond3> (R)) = σ <cond2> (σ <cond3> (σ <cond1> (R)))
  - **A cascade of SELECT operations may be replaced by a single selection with a conjunction of all the conditions:**
    - σ<cond1> (σ < cond2> (σ <cond3> (R)) = σ <cond1> AND < cond2> AND < cond3> (R)))
    - The number of tuples in the result of a SELECT is less than (or equal to) the number of tuples in the input relation R

Unary Relational Operations: PROJECT

- **Relation *r*:**

A	B	C
α	10	1
α	20	1
β	30	1
β	40	2

$\Pi_{A,C}(r)$

A	C
α	1
α	1
β	1
β	2

=

A	C
α	1
β	1
β	2

Unary Relational Operations: PROJECT

- The general form of the *project* operation is:
    - $\pi_{\text{<attribute list>}}(R)$
    - **π (pi)** is the symbol used to represent the *project* operation
    - **<attribute list>** is the **desired list of attribute**s from relation R.

This operation keeps (listed) certain *columns* (attributes) from a relation and discards the other columns.

- The project operation removes any duplicate tuples
- The result of the project operation must be set of tuples.
- **Example: To list each employee's first and last name and salary, the following is used:**

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

Unary Relational Operations: PROJECT

- The number of tuples in the result of projection π <list> (R) is always less or equal to the number of tuples in R (list non-key)

- If the list of attributes **includes a key of R,** then the number of tuples in the result of PROJECT is equal to the number of tuples in R

- **PROJECT is not commutative:**

➔π <list2> (π <list1> (R)) ≠ π <list1> (π <list2> (R))

Figure 5.6
One possible database state for the COMPANY relational database schema.

EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
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
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	NULL	1

DEPARTMENT

Dname	Dnumber	Mgr_ssn	Mgr_start_date
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

WORKS_ON

Essn	Pno	Hours
123456789	1	32.5
123456789	2	7.5
666884444	3	40.0
453453453	1	20.0
453453453	2	20.0
333445555	2	10.0
333445555	3	10.0
333445555	10	10.0
333445555	20	10.0
999887777	30	30.0
999887777	10	10.0
987987987	10	35.0
987987987	30	5.0
987654321	30	20.0
987654321	20	15.0
888665555	20	NULL

PROJECT

Pname	Pnumber	Plocation	Dnum
ProductX	1	Bellaire	5
ProductY	2	Sugarland	5
ProductZ	3	Houston	5
Computerization	10	Stafford	4
Reorganization	20	Houston	1
Newbenefits	30	Stafford	4

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

Unary Relational Operations: PROJECT

- (a) $\sigma_{(DNO=4 \text{ AND } SALARY>25000) \text{ OR } (DNO=5 \text{ AND } SALARY>30000)}$(EMPLOYEE).
- (b) $\pi_{LNAME,FNAME,SALARY}$(EMPLOYEE).
- (c) $\pi_{SEX,SALARY}$(EMPLOYEE).

(a)

FNAME	MINIT	LNAME	SSN	BDATE	ADDRESS	SEX	SALARY	SUPERSSN	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 FireOak,Humble,TX	M	38000	333445555	5

(b)

LNAME	FNAME	SALARY
Smith	John	30000
Wong	Franklin	40000
Zelaya	Alicia	25000
Wallace	Jennifer	43000
Narayan	Ramesh	38000
English	Joyce	25000
Jabbar	Ahmad	25000
Borg	James	55000

(c)

SEX	SALARY
M	30000
M	40000
F	25000
F	43000
M	38000
M	25000
M	55000

Relational Algebra Expressions

- To avoid applying several relational algebraic operations one after the other
  - Write the operations as a **single relational algebra expression** by nesting the operations, or
  - Apply one operation at a time and create **intermediate result relations.**

- In the latter case, we must give name to the relations that hold the intermediate results.
- If no renaming is applied, the name of the attributes in the resulting relation of a **SELECT** operation are the same as those in the original relation and in the same order.
- For a **PROJECT** with no renaming, the resulting relation has the same attribute names as those in the projection list.

Relational Algebra Expressions

- **Example:** To retrieve the first name, last name, and salary of all employees who work in department number 5,
- A single relational algebra expression :

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

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

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

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

Unary Relational Operations: RENAME

- **The rename operator is ρ (*rho*)**
- The general Rename operation can be expressed by any of the following forms:
- $\rho_{S (B_1, B_2, \ldots, B_n)} (R)$ (changes both)

  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).
  - $\rho_{(B_1, B_2, \ldots, B_n)} (R)$ is a renamed relation with column names B_1, B_1, …..B_n

Unary Relational Operations: RENAME

- $\textbf{TEMP} \leftarrow \sigma_{\text{DNO=5}}(\textbf{EMPLOYEE})$
  $\textbf{R(FIRSTNAME, LASTNAME, SALARY)} \leftarrow \pi_{\text{FNAME, LNAME, SALARY}}(\textbf{TEMP})$

(a)

FNAME	LNAME	SALARY
John	Smith	30000
Franklin	Wong	40000
Ramesh	Narayan	38000
Joyce	English	25000

(b)

TEMP	FNAME	MINIT	LNAME	SSN	BDATE	ADDRESS	SEX	SALARY	SUPERSSN	DNO
	John	B	Smith	123456789	1965-01-09	731 Fondren,Houston,TX	M	30000	333445555	5
	Franklin	T	Wong	333445555	1955-12-08	638 Voss,Houston,TX	M	40000	888665555	5
	Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak,Humble,TX	M	38000	333445555	5
	Joyce	A	English	453453453	1972-07-31	5631 Rice,Houston,TX	F	25000	333445555	5

R	FIRSTNAME	LASTNAME	SALARY
	John	Smith	30000
	Franklin	Wong	40000
	Ramesh	Narayan	38000
	Joyce	English	25000

Union Operations - Example

- **Relations *r, s:***

r

A	B
α	1
α	2
β	1

s

A	B
α	2
β	3

- **r $\cup$ s:**

A	B
α	1
α	2
β	1
β	3

Set Difference - Example

- **Relations *r*, *s*:**

A	B
α	1
α	2
β	1

r

A	B
α	2
β	3

s

- ***r* – *s*:**

A	B
α	1
β	1

Relational Algebra from Set Theory

- **Type Compatibility :** The operand relations $R_1(A_1, A_2, ..., A_n)$ and $R_2(B_1, B_2, ..., B_n)$ must have the same number of attributes, and the domains of corresponding attributes must be compatible; that is,

$$\mathbf{dom(A_i) = dom(B_i) \text{ for } i=1, 2, ..., n.}$$

- The resulting relation for $R_1 \cup R_2$, $R_1 \cap R_2$, or $R_1 - R_2$ has the same attribute names as the *first* operand relation R1 (by convention).

Set Theory – UNION

- UNION operation is, denoted by $R \cup S$,
  - It is a relation that includes all tuples that are either in R or in S or in both R and S.
  - The two relation must be "*type compatible*".
  - 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:

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

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

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

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

Set Theory – UNION Example

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:

DEP5_EMPS $\leftarrow \sigma_{DNO=5}$ **(EMPLOYEE)**

RESULT1 $\leftarrow \pi_{SSN}$**(DEP5_EMPS)**

RESULT2(SSN) $\leftarrow \pi_{SUPERSSN}$**(DEP5_EMPS)**

RESULT $\leftarrow$ **RESULT1** $\cup$ **RESULT2**

RESULT1	SSN
	123456789
	333445555
	666884444
	453453453

RESULT2	SSN
	333445555
	888665555

RESULT	SSN
	123456789
	333445555
	666884444
	453453453
	888665555

Relational Algebra from Set Theory

- Intersection operation : denoted by R ∩ S, is a relation that includes all tuples that are in both R and S.
- Set Difference (or MINUS) operation : denoted by R - S, is a relation that includes all tuples that are in R but not in S.

Relational Algebra from Set Theory

(a) Two union-compatible relations.

(b) STUDENT ∪ INSTRUCTOR.

(c) STUDENT ∩ INSTRUCTOR.

(d) STUDENT – INSTRUCTOR. (e) INSTRUCTOR – STUDENT

(a)

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

(b)

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

(c)

FN	LN
Susan	Yao
Ramesh	Shah

(d)

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

(e)

FNAME	LNAME
John	Smith
Ricardo	Browne
Francis	Johnson

Properties of UNION, INTERSECT, and DIFFERENCE

- Notice that both union and intersection are commutative operations; that is

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

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

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

$$R \cup (S \cup T) = (R \cup S) \cup T$$

$$(R \cap S) \cap T = R \cap (S \cap T)$$

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

$$R - S \neq S - R$$

Cartesian Product- Example

- **Relations *r, s*:**

A	B
α	1
β	2

r

C	D	E
α	10	a
β	10	a
β	20	b
γ	10	b

s

- ***r* x *s*:**

A	B	C	D	E
α	1	α	10	a
α	1	β	10	a
α	1	β	20	b
α	1	γ	10	b
β	2	α	10	a
β	2	β	10	a
β	2	β	20	b
β	2	γ	10	b

Set Operations

– **CARTESIAN (or cross product) Operation**
  - Used to combine tuples from two relations in a combinatorial fashion.

    $\mathbf{R(A_1, A_2, ..., A_n) \ x \ S(B_1, B_2, ..., B_m)}$

    $\mathbf{Q(A_1, A_2, ..., A_n, B_1, B_2, ..., B_m)}$ with degree n+m attributes.
  - The resulting relation Q has one tuple for each combination of tuples—one from R and one from S.
  - Hence, if R has n_R tuples (denoted as $|R| = n_R$), and S has n_S tuples, then

    | R x S | will have $n_R * n_S$ tuples.
  - The two operands do NOT have to be "type compatible"

Set Operations

FEMALE_EMPS ← σ $_{SEX='F'}$(EMPLOYEE)

EMPNAMES ← π $_{FNAME,\ LNAME,\ SSN}$ **(FEMALE_EMPS)**

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

Set Operations

EMP_DEPENDENTS ← EMPNAMES x DEPENDENT

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	• • •

EMP_DEPENDENTS will contain every combination of EMPNAMES and DEPENDENT.
whether or not they are actually related

Binary Relational Operations

- **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 denoted by ⋈ <joincondition>
  - This operation is very important for any relational database with more than a single relation, because it allows us to process relationships among relations.
  - The general form of a join operation on two relations R(A1, A2, . . ., An) and S(B1, B2, . . ., Bm) is:

    R⋈<join condition>S

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

Binary Relational Operations

- **EQUIJOIN Operation:** The most common use of join involves join conditions with equality comparisons only.
- Such a join, where the only comparison operator used is =, is called an EQUIJOIN.
- In the result of an EQUIJOIN we always have 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.

Binary Relational Operations

- **Example:** Retrieve the name of the manager of each department

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

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	• • •

Binary Relational Operations

- **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.

Binary Relational Operations

(a) PROJ_DEPT ← PROJECT * DEPT

(b) DEPT_LOCS ← DEPARTMENT * DEPT_LOCATIONS

(a)

PROJ_DEPT	PNAME	<u>PNUMBER</u>	PLOCATION	DNUM	DNAME	MGRSSN	MGRSTARTDATE
	ProductX	1	Bellaire	5	Research	333445555	1988-05-22
	ProductY	2	Sugarland	5	Research	333445555	1988-05-22
	ProductZ	3	Houston	5	Research	333445555	1988-05-22
	Computerization	10	Stafford	4	Administration	987654321	1995-01-01
	Reorganization	20	Houston	1	Headquarters	888665555	1981-06-19
	Newbenefits	30	Stafford	4	Administration	987654321	1995-01-01

(b)

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

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.

Additional Relational Operations

- **The OUTER JOIN Operation**
  - 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.

Additional Relational Operations

- **The OUTER JOIN Operation**

  - Example: List all employee names and also the name of the departments they manage *if they happen to manage a department*

Figure 7.5 Schema diagram for the COMPANY relational database schema; the primary keys are underlined.

EMPLOYEE

FNAME	MINIT	LNAME	SSN	BDATE	ADDRESS	SEX	SALARY	SUPERSSN	DNO

DEPARTMENT

DNAME	DNUMBER	MGRSSN	MGRSTARTDATE

DEPT_LOCATIONS

DNUMBER	DLOCATION

PROJECT

PNAME	PNUMBER	PLOCATION	DNUM

WORKS_ON

ESSN	PNO	HOURS

DEPENDENT

ESSN	DEPENDENT_NAME	SEX	BDATE	RELATIONSHIP

Figure 5.6
One possible database state for the COMPANY relational database schema.

EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
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
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	NULL	1

DEPARTMENT

Dname	Dnumber	Mgr_ssn	Mgr_start_date
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

WORKS_ON

Essn	Pno	Hours
123456789	1	32.5
123456789	2	7.5
666884444	3	40.0
453453453	1	20.0
453453453	2	20.0
333445555	2	10.0
333445555	3	10.0
333445555	10	10.0
333445555	20	10.0
999887777	30	30.0
999887777	10	10.0
987987987	10	35.0
987987987	30	5.0
987654321	30	20.0
987654321	20	15.0
888665555	20	NULL

PROJECT

Pname	Pnumber	Plocation	Dnum
ProductX	1	Bellaire	5
ProductY	2	Sugarland	5
ProductZ	3	Houston	5
Computerization	10	Stafford	4
Reorganization	20	Houston	1
Newbenefits	30	Stafford	4

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

Additional Relational Operations

- **The OUTER JOIN Operation**
  - Example: List all employee names and also the name of the departments they manage *if they happen to manage a department*
  - Solution:

TEMP <-- (EMPLOYEE $⟕_{SSN=MGRSSN}$ DEPARTMENT)

RESULT <-- $\pi_{FNAME,MINIT,LNAME,DNAME}$ (TEMP)

Additional Relational Operations

- TEMP <-- (EMPLOYEE $\bowtie_{SSN=MGRSSN}$ DEPARTMENT)
  RESULT <-- $\pi_{FNAME,MINIT,LNAME,DNAME}$ (TEMP)

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

Examples of Queries in RA

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

Examples of Queries in RA

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

$$\text{RESEARCH\_DEPT} \leftarrow \sigma_{\text{DNAME='Research'}}(\text{DEPARTMENT})$$

$$\text{RESEARCH\_EMPS} \leftarrow (\text{RESEARCH\_DEPT} \bowtie_{\text{DNUMBER=DNO}} \text{EMPLOYEE})$$

$$\text{RESULT} \leftarrow \pi_{\text{FNAME, LNAME, ADDRESS}}(\text{RESEARCH\_EMPS})$$

Additional Relational Operations

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

Additional Relational Operations

- **Aggregate Functions and Grouping**
  - Aggregate functions are defined by using the symbol “*script F*”

    $$\langle\textit{grouping attributes}\rangle \, \mathfrak{J}_{\langle\textit{function list}\rangle} \, (\boldsymbol{R})$$

  - <grouping attributes> is the list of attributes in R
  - <function list> is a list of (<function><attribute>) pairs
  - Example: To retrieve each department number, the number of employees in the department, and their average salary:

Additional Relational Operations

- $\rho_{R(\textbf{Dno, No\_of\_Employees, Average\_Sal})} \leftarrow\!\!-\!\!-\!\!-$

$$_{DNO}\,\mathfrak{I}_{COUNT\ SSN,\ AVERAGE\ SALARY}\ (\textit{EMPLOYEE})$$

(a)

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

$$_{DNO}\,\mathfrak{I}_{COUNT\ SSN,\ AVERAGE\ SALARY}\ (\textit{EMPLOYEE})$$

(b)

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

$$\mathfrak{I}_{COUNT\ SSN,\ AVERAGE\ SALARY}\ (\textit{EMPLOYEE})$$

(c)

COUNT_SSN	AVERAGE_SALARY
8	35125

Summary

Unary Relational Operations

Binary Relational Operations

Additional Relational Operations

References

- Fundamentals of Database Systems Ramez Elamsri, Navathe, 7th Edition