HOA SEN UNIVERSITY

Lecture 10
The Relational Algebra and Relational Calculus – 1

Objectives

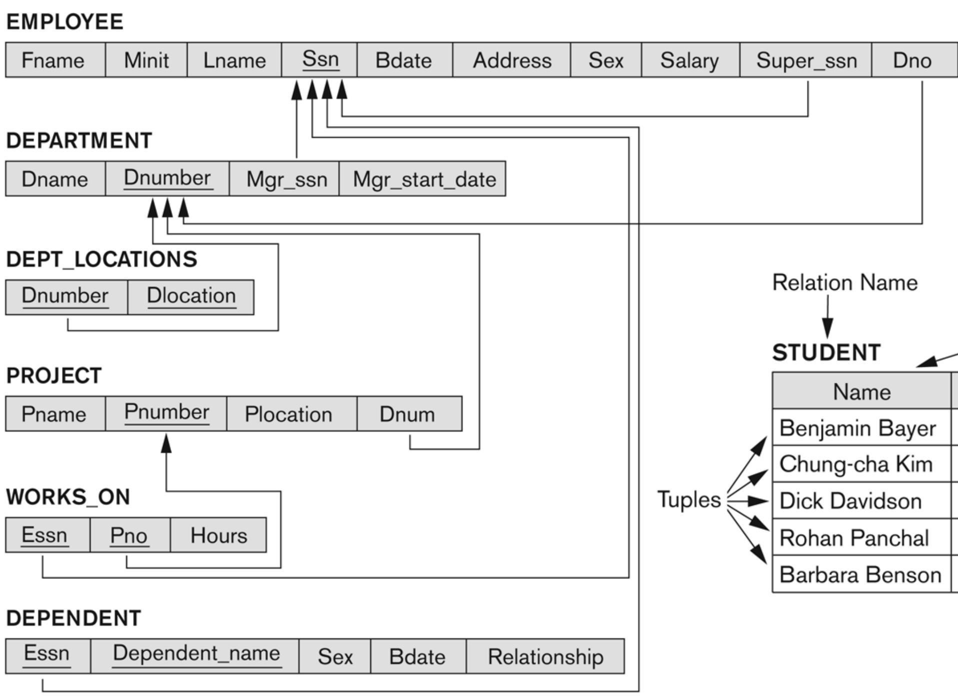
- Relational Algebra
  - Unary Relational Operations
  - Relational Algebra Operations From Set Theory
- Relational Algebra
  - Binary Relational Operations
  - Additional Relational Operations
  - Examples of Queries in Relational Algebra

- Ref.: Chapter 8

ĐẠI HỌC HOA SEN

Review Relational Database Model

- Database: relations (tables)
- Relation: attributes (columns) – tuples (rows)
- Attribute: domain (data type)



Relation Name

Attributes

STUDENT

Tuples

Name	Ssn	Home_phone	Address	Office_phone	Age	Gpa
Benjamin Bayer	305-61-2435	373-1616	2918 Bluebonnet Lane	NULL	19	3.21
Chung-cha Kim	381-62-1245	375-4409	125 Kirby Road	NULL	18	2.89
Dick Davidson	422-11-2320	NULL	3452 Elgin Road	749-1253	25	3.53
Rohan Panchal	489-22-1100	376-9821	265 Lark Lane	749-6492	28	3.93
Barbara Benson	533-69-1238	839-8461	7384 Fontana Lane	NULL	19	3.25

Relational Algebra Overview

- 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
  - This property makes the algebra “closed” (all objects in relational algebra are relations)
- A sequence of relational algebra operations forms a **relational algebra expression**
  - The result of a relational algebra expression is also a relation that represents the result of a database query (or retrieval request)

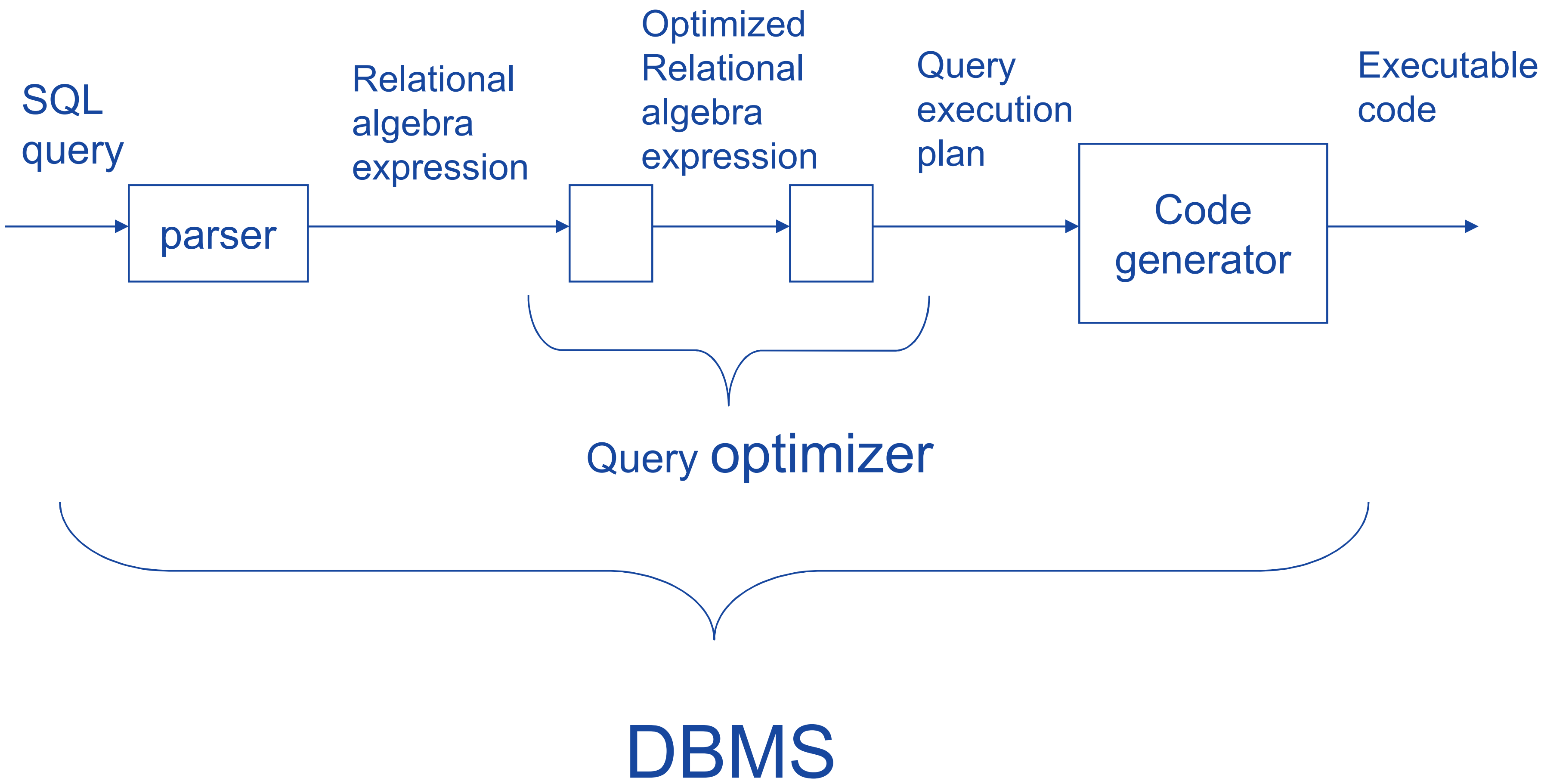
Relational Algebra Overview

- Relational Algebra consists of several groups of operations
  - Unary Relational Operations
    - SELECT (σ (sigma))
    - PROJECT (π (pi))
    - RENAME (ρ (rho))
  - Relational Algebra Operations From Set Theory
    - UNION ($\cup$)
    - INTERSECTION ($\cap$)
    - DIFFERENCE (or MINUS, –)
    - CARTESIAN PRODUCT (**x**)

Relational Algebra Overview

- Relational Algebra consists of several groups of operations
  - Binary Relational Operations
    - JOIN (several variations of JOIN exist)
    - DIVISION
  - Additional Relational Operations
    - OUTER JOINS, OUTER UNION
    - AGGREGATE FUNCTIONS (These compute summary of information: for example, SUM, COUNT, AVG, MIN, MAX)

Relational Algebra in a DBMS



Unary Relational Operations: SELECT

- The SELECT operation (σ) is used to select a *subset* of the tuples from a relation based on a **selection condition (P)**.
  - $r(P) = \{t : t \in r \ \& \ t(P)\}$
  - The selection condition acts as a **filter**
  - Keeps only those tuples that satisfy the qualifying condition
  - Tuples satisfying the condition are *selected* whereas the other tuples are discarded (*filtered out*)

Unary Relational Operations: SELECT

- In general, the *select* operation is denoted by

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

  - 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
    - tuples that make the condition **true** are selected
      - appear in the result of the operation
    - tuples that make the condition **false** are filtered out
      - discarded from the result of the operation

ĐẠI HỌC HOA SEN

- Examples:
  - Select the EMPLOYEE tuples whose department number is 4:

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

EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
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
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4

- Examples:
  - Select the employee tuples whose salary is greater than $30,000:

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

EMPLOYEE

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
James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	NULL	1

Example of Select

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

EMPLOYEE

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

Unary Relational Operations: PROJECT

- PROJECT Operation is denoted by π (pi)
  - $r.X = \{t.X : t \in r\}$
  - This operation keeps certain *columns* (attributes) from a relation and discards the other columns.
  - PROJECT creates a vertical partitioning
    - The list of specified columns (attributes) is kept in each tuple
    - The other attributes in each tuple are discarded

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 attributes from relation R.
- The project operation *removes any duplicate tuples*
  - This is because the result of the *project* operation must be a *set of tuples*
    - Mathematical sets *do not allow* duplicate elements.

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

$$\pi_{\text{LName, FName, Salary}}(\text{EMPLOYEE})$$

EMPLOYEE

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

Example of Project

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

- $\pi_{\text{LName, FName,Salary}}(\text{EMPLOYEE})$

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

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

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

Relational Algebra Expressions

- We may want to apply several relational algebra operations one after the other
  - Either we can write the operations as a single **relational algebra expression** by nesting the operations, or
  - We can apply one operation at a time and create **intermediate result relations**.

- In the latter case, we must give names to the relations that hold the intermediate results.

Single expression vs sequence of relational operations (Example)

- To retrieve the first name, last name, and salary of all employees who work in department number 5, we must apply a select and a project operation
- We can write a *single relational algebra expression* as follows:
  - $\pi_{\text{FName, LName, Salary}}(\sigma_{\text{Dno=5}}(\text{EMPLOYEE}))$

Fname	Lname	Salary
John	Smith	30000
Franklin	Wong	40000
Ramesh	Narayan	38000
Joyce	English	25000

Single expression vs sequence of relational operations (Example)

- OR We can explicitly show the *sequence of operations*, giving a name to each intermediate relation:
  - DEP5_EMPS $\leftarrow \sigma_{DNo=5}$(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
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

  - RESULT $\leftarrow \pi_{\text{Fname, Lname, Salary}}$ (DEP5_EMPS)

Fname	Lname	Salary
John	Smith	30000
Franklin	Wong	40000
Ramesh	Narayan	38000
Joyce	English	25000

Unary Relational Operations: RENAME

- The RENAME operator is denoted by

  ρ (rho)

- In some cases, we may want to *rename* the attributes of a relation or the relation name or both
  - Useful when a query requires multiple operations
  - Necessary in some cases (see JOIN operation later)

Unary Relational Operations: RENAME (2)

- The general RENAME operation ρ can be expressed by any of the following forms:
  - $\rho_{S\ (B1,\ B2,\ \ldots,\ Bn\ )}(R)$ changes both:
    - the relation name to S, *and*
    - the column (attribute) names to $B_1, B_2, \ldots..B_n$
  - $\rho_S(R)$ changes:
    - the *relation name* only to S
  - $\rho_{(B1,\ B2,\ \ldots,\ Bn\ )}(R)$ changes:
    - the *column (attribute) names* only to $B_1, B_2, \ldots..B_n$

Example of Rename

- DEP5_EMPS $\leftarrow \sigma_{\text{DNo=5}}$ (EMPLOYEE)
- RESULT $\leftarrow \pi_{\text{FName, LName, Salary}}$ (DEP5_EMPS)

Or

- List the new attribute name:

RESULT(FName, LName, Salary) $\leftarrow \pi_{\text{FName, LName, Salary}}$ (DEP5_EMPS)

- Rename operation:

$\rho_{\text{RESULT (First\_Name, Last\_Name, Salary)}}$(DEPT5_EMPS)

Relational Algebra Operations from Set Theory: UNION

- UNION Operation:
  - $R \cup S = \{t \mid t \in R \vee t \in S\}$
  - Binary operation, denoted by $\cup$
  - The result of $R \cup S$, 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
  - The two operand relations R and S must be "type compatible" (or UNION compatible)
    - R and S must have same number of attributes
    - Each pair of corresponding attributes must be type compatible (have same or compatible domains)

Example of Union

- To retrieve the social security numbers of all employees who either *work in department 5* (RESULT1 below) **or** *directly supervise an employee who works in department 5*
  - We can use the UNION operation as follows:

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

$$RESULT1 \leftarrow \pi_{\ Ssn}(DEP5\_EMPS)$$

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

$$RESULT \leftarrow RESULT1 \cup RESULT2$$

- The union operation produces the tuples that are in either RESULT1 or RESULT2 or both

	SSN
1	123456789
2	333445555
3	453453453
4	666884444

$\cup$

SuperSSN
333445555
888665555

$\rightarrow$

SSN
123456789
333445555
453453453
666884444
888665555

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

Relational Algebra Operations from Set Theory: INTERSECTION

- INTERSECTION Operation:
  - $R \cap S = \{t \mid t \in R \wedge t \in S\}$
  - INTERSECTION is denoted by $\cap$
  - The result of the operation $R \cap S$, is a relation that includes all tuples that are in both R and S
  - The attribute names in the result will be the same as the attribute names in R
  - The two operand relations R and S must be "type compatible"

Example of Intersection

- To retrieve the social security numbers of all employees who either *work in department 5* (RESULT1 below) **and** *directly supervise an employee who works in department 5*
  - We can use the INTERSECTION operation as follows:

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

    $RESULT1 \leftarrow \pi_{Ssn}(DEP5\_EMPS)$

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

    $RESULT \leftarrow RESULT1 \cap RESULT2$
- The intersection operation produces the tuples that are in RESULT1 and RESULT2

	SSN
1	123456789
2	333445555
3	453453453
4	666884444

∩

SuperSSN
333445555
888665555

→

SSN
333445555

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

Relational Algebra Operations from Set Theory: SET DIFFERENCE

- SET DIFFERENCE (also called MINUS or EXCEPT)
  - $R - S = \{t \mid t \in R \wedge t \notin S\}$
  - is denoted by –
  - The result of R – S, is a relation that includes all tuples that are in R but not in S
  - The attribute names in the result will be the same as the attribute names in R
  - The two operand relations R and S must be "type compatible"

Example of Minus

- To retrieve the social security numbers of all employees who *work in department 5* (RESULT1 below) **<u>and</u>** **<u>not</u>** *directly supervise an employee who works in department 5*
  - We can use the UNION operation as follows:

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

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

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

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

- The intersection operation produces the tuples that are in RESULT1 but not in RESULT2

	SSN
1	123456789
2	333445555
3	453453453
4	666884444

−

SuperSSN
333445555
888665555

→

SSN
123456789
453453453
666884444

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

The set operations UNION, INTERSECTION, and MINUS.
(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

Relational Algebra Operations from Set Theory: CARTESIAN PRODUCT

- CARTESIAN (or CROSS) PRODUCT Operation
  - This operation is used to combine tuples from two relations in a combinatorial fashion.
  - Denoted by $R(A_1, A_2, \ldots, A_n) \times S(B_1, B_2, \ldots, B_m)$
  - Result 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 state 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"

Relational Algebra Operations from Set Theory: CARTESIAN PRODUCT

- Generally, CROSS PRODUCT is not a meaningful operation
  - Can become meaningful when followed by other operations

- Example (not meaningful):

  FEMALE_EMPS $\leftarrow \sigma_{\text{Sex='F'}}$(EMPLOYEE)

  EMPNAMES $\leftarrow \pi_{\text{Fname, Lname, Ssn}}$ (FEMALE_EMPS)

  EMP_DEPENDENTS $\leftarrow$ EMPNAMES x DEPENDENT

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

ĐẠI HỌC HOA SEN

FEMALE_EMPS

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
Alicia	J	Zelaya	999887777	1968-07-19	3321Castle, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291Berry, 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

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	...

Relational Algebra Operations from Set Theory: CARTESIAN PRODUCT

- To keep only combinations where the DEPENDENT is related to the EMPLOYEE, we add a SELECT operation as follows
- Example (meaningful):

  FEMALE_EMPS $\leftarrow \sigma_{Sex='F'}$(EMPLOYEE)

  EMPNAMES $\leftarrow \pi_{Fname, Lname, Ssn}$ (FEMALE_EMPS)

  EMP_DEPENDENTS $\leftarrow$ EMPNAMES x DEPENDENT

  ACTUAL_DEPS $\leftarrow \sigma_{Ssn=Essn}$(EMP_DEPENDENTS)

  RESULT $\leftarrow \pi_{Fname, Lname, Dependent\_name}$ (ACTUAL_DEPS)
- RESULT will now contain the name of female employees and their dependents

ACTUAL_DEPENDENTS

Fname	Lname	Ssn	Essn	Dependent_name	Sex	Bdate	...
Jennifer	Wallace	987654321	987654321	Abner	M	1942-02-28	...

RESULT

Fname	Lname	Dependent_name
Jennifer	Wallace	Abner

Binary Relational Operations: JOIN

- JOIN Operation (denoted by $\bowtie$)
  - 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 combines this sequence into a single operation
  - This operation is very important for any relational database with more than a single relation, because it allows us *combine related tuples* from various 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*.

Example Binary Relational Operations: JOIN

- Example: Suppose that we want to retrieve the name of the manager of each department.
  - To get the manager's name, we need to combine each DEPARTMENT tuple with the EMPLOYEE tuple whose SSN value matches the MGRSSN value in the department tuple.
  - We do this by using the join ⋈ operation.

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

  - MgrSSN = SSN is the join condition
    - Combines each department record with the employee who manages the department
    - The join condition

DEPARTMENT.MgrSSN= EMPLOYEE.SSN

DEPT_MGR

Dname	Dnumber	Mgr_ssn	...	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: EQUIJOIN [=]

- 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 an EQUIJOIN.

Binary Relational Operations: NATURAL JOIN [*]

- Another variation of JOIN called ***NATURAL JOIN*** (denoted by *) was created to get rid of the second (superfluous) attribute in an EQUIJOIN condition.
  - because one of each pair of attributes with identical values is superfluous
- 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 Binary Relational Operations
NATURAL JOIN

- 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
    - Only attribute with the same name is Dnumber
    - An implicit join condition is created based on this attribute:

  DEPARTMENT.Dnumber = DEPT_LOCATIONS.Dnumber

- Another example: Q ← R(A,B,C,D) * S(C,D,E)
  - The implicit join condition includes *each pair* of attributes with the same name, “AND”ed together:
    - R.C=S.C AND R.D=S.D
  - Result keeps only one attribute of each such pair:
    - Q(A,B,C,D,E)

Complete Set of Relational Operations

- The set of operations (*complete set*):

  SELECT σ PROJECT π RENAME ρ

  UNION $\cup$ INTERSECTION $\cap$

  DIFFERENCE (MINUS or EXCEPT) $-$

  CARTESIAN PRODUCT X

- For example:
  - $R \cap S = (R \cup S) - ((R - S) \cup (S - R))$
  - $R \bowtie_{\text{<join condition>}} S = \sigma_{\text{<join condition>}} (R \times S)$

Q & A