Chapter 3

The Relational Model

Department: Computer
Course: DBMS
Faculty: Sana Shaikh

Relational Algebra

✔ Is **basic set of operations** for the relational model

✔ enable a user to specify **basic retrieval requests**

✔ The **result of an operation is a *new relation***, which may have been formed from one or more *input* relations

✔ The **algebra operations** thus produce new relations, can be further manipulated using operations of the same algebra

✔A sequence of relational algebra operations forms a **relational algebra expression**

Topics to be covered:

- Relational Algebra operations

Learning Outcomes:

Students should be able to:
- understand concept of relational algebra and how it can be used to communicate with the database

Company Database

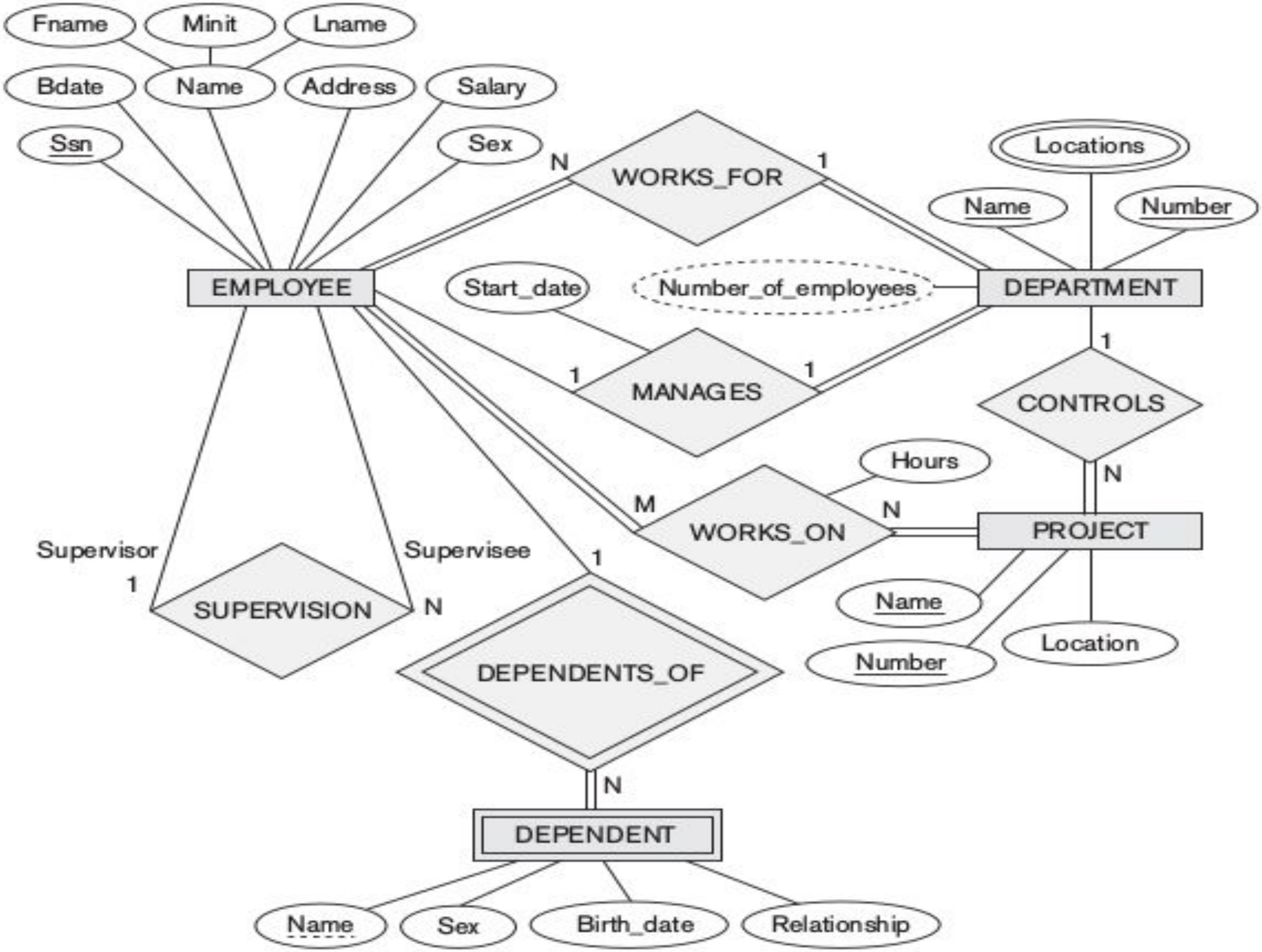
The company is organized into departments. Each department has a unique name, a unique number, and a particular employee who manages the department. We keep track of the start date when that employee began managing the department. A department may have several locations.

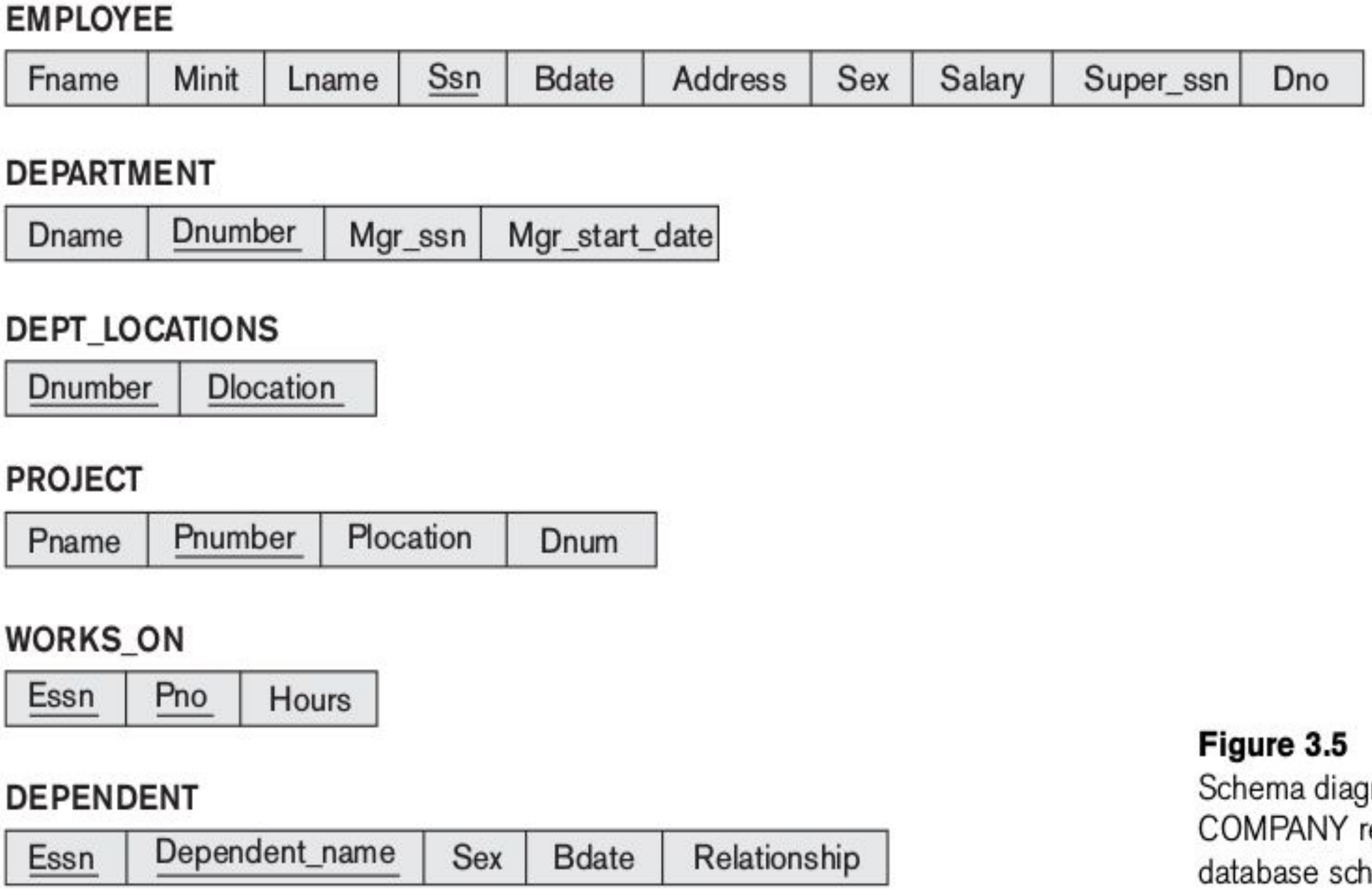
A department controls a number of projects, each of which has a unique name, a unique number, and a single location.

We store each employee's name, Social Security number, 2 address, salary, sex (gender), and birth date. An employee is assigned to one department, but may work on several projects, which are not necessarily controlled by the same department. We keep track of the current number of hours per week that an employee works on each project. We also keep track of the direct supervisor of each employee (who is another employee).

We want to keep track of the dependents of each employee for insurance purposes. We keep each dependent's first name, sex, birth date, and relationship to the employee.

Figure 9.1
The ER conceptual schema diagram for the COMPANY database.





EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno

DEPARTMENT

Dname	Dnumber	Mgr_ssn	Mgr_start_date

DEPT_LOCATIONS

Dnumber	Dlocation

PROJECT

Pname	Pnumber	Plocation	Dnum

WORKS_ON

Essn	Pno	Hours

DEPENDENT

Essn	Dependent_name	Sex	Bdate	Relationship

Figure 3.5
Schema diagram for the COMPANY relational database schema.

Figure 3.6
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

Relational Algebra consists of several groups of operations

- ❑ Unary Relational Operations
- ❑ Relational Algebra Operations From Set Theory
- ❑ Binary Relational Operations
- ❑ Additional Relational Operations

Relational Algebra consists of several groups of operations

❑ Unary Relational Operations

- ✔SELECT (symbol: **σ** (sigma))
- ✔PROJECT (symbol: **Π** (pi))
- ✔RENAME (symbol: **٩** (rho))

Relational Algebra consists of several groups of operations

❑ Unary Relational Operations

✔SELECT (symbol: **σ** (sigma))

✔PROJECT (symbol: **Π** (pi))

✔RENAME (symbol: ٩ (rho))

❑ Relational Algebra Operations From Set Theory

✔ UNION (**U**)

✔ INTERSECTION (∩)

✔ DIFFERENCE (or MINUS, –)

✔ CARTESIAN PRODUCT (**x**)

Relational Algebra consists of several groups of operations

- ❑ Binary Relational Operations
  - ✔JOIN (several variations of JOIN exist)
  - ✔DIVISION

Relational Algebra consists of several groups of operations

❑ Binary Relational Operations

- ✔JOIN (several variations of JOIN exist)
- ✔DIVISION

❑ Additional Relational Operations

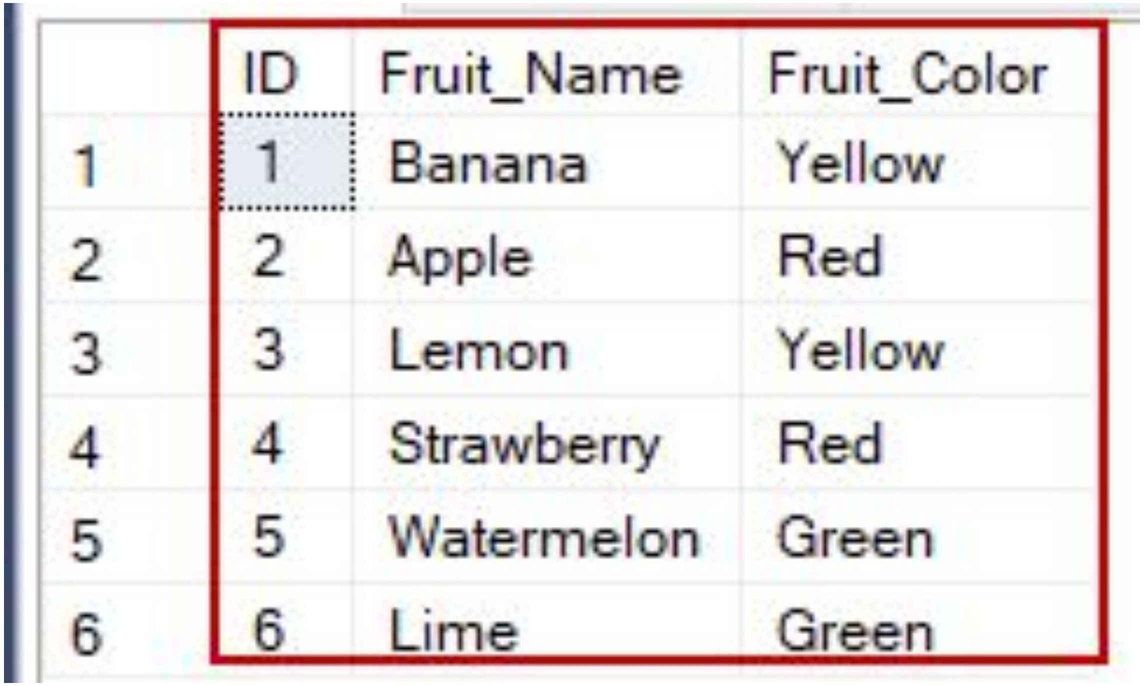
- ✔OUTER JOINS, OUTER UNION
- ✔AGGREGATE FUNCTIONS
  - SUM
  - COUNT
  - AVG
  - MIN
  - MAX

Unary Relational Operations: SELECT

- ❑ is used to select a *subset* of the tuples from a relation based on a **selection condition**.
- ❑ The selection condition acts as a **filter.**
- ❑ Also known as horizontal partition.
- ❑ In general, the *select* operation is denoted by:

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

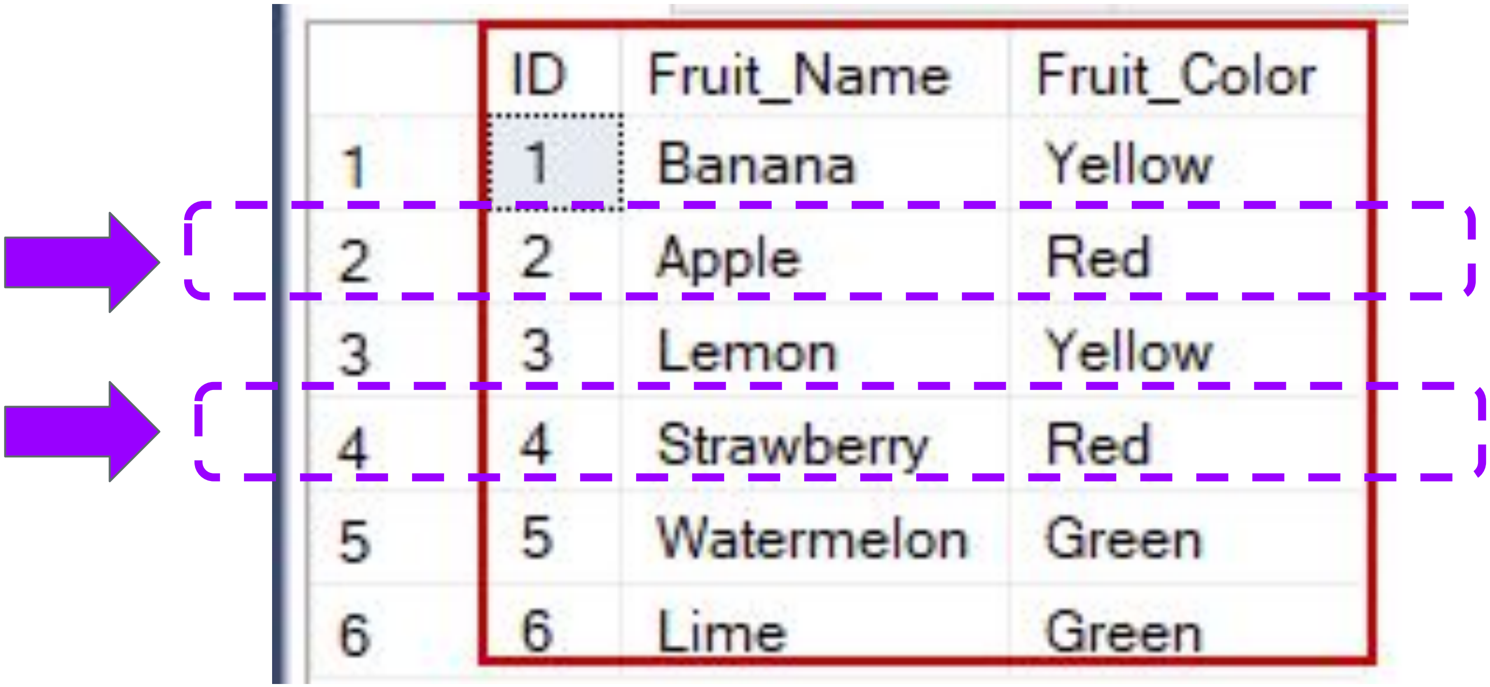
Table - Fruits



	ID	Fruit_Name	Fruit_Color
1	1	Banana	Yellow
2	2	Apple	Red
3	3	Lemon	Yellow
4	4	Strawberry	Red
5	5	Watermelon	Green
6	6	Lime	Green

Query - Extract details of fruits whose colour is “Red”

Table - Fruits



	ID	Fruit_Name	Fruit_Color
1	1	Banana	Yellow
2	2	Apple	Red
3	3	Lemon	Yellow
4	4	Strawberry	Red
5	5	Watermelon	Green
6	6	Lime	Green

Unary Relational Operations: SELECT

Examples:

1. Select the EMPLOYEE tuples whose department number is 4
2. Select the employee tuples whose salary is greater than $30,000:

Unary Relational Operations: SELECT

Examples:

1. Select the EMPLOYEE tuples whose department number is 4
2. Select the employee tuples whose salary is greater than $30,000:

Answer_1 □ $\sigma_{\text{DNO} = 4}$ (EMPLOYEE)

Answer_2 □ $\sigma_{\text{SALARY} > 30{,}000}$ (EMPLOYEE)

Unary Relational Operations: SELECT

Examples:

1. Select the EMPLOYEE tuples whose department number is 4
2. Select the employee tuples whose salary is greater than $30,000:

Answer_1 □ $\sigma_{\text{DNO} = 4}$ (EMPLOYEE)

Answer_2 □ $\sigma_{\text{SALARY} > 30{,}000}$ (EMPLOYEE)

Question :

To select the tuples of all employees who either work in department 4 and make over $25,000 or work in department 5 and make over $30,000

SELECT Operation Properties

1. The SELECT operation $\sigma_{\text{<selection condition>}}(R)$ produces a relation S that has the **same schema (same attributes)** as R.

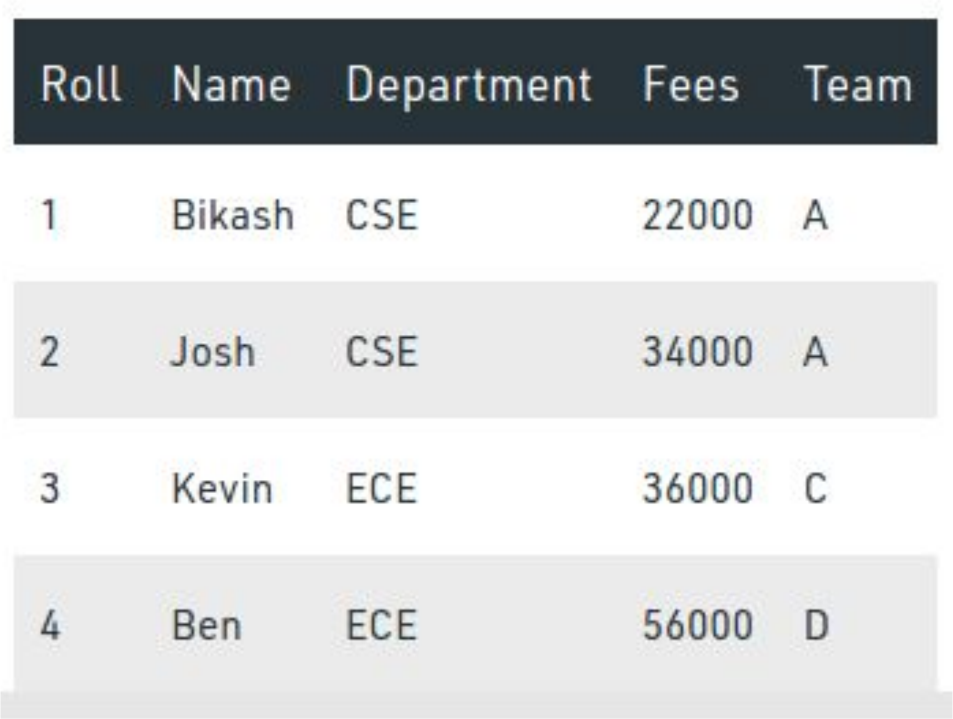
2. SELECT is **commutative**.

3. Because of commutative property, a cascade (sequence) of SELECT operations may be applied in **any order**.

4. A cascade of SELECT operations may be **replaced by a single selection** with a conjunction of all the conditions.

5. 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

Example-2: Given a relation **Student(Roll, Name, Class, Fees, Team)** with the following tuples:



Roll	Name	Department	Fees	Team
1	Bikash	CSE	22000	A
2	Josh	CSE	34000	A
3	Kevin	ECE	36000	C
4	Ben	ECE	56000	D

1) Select all the student of Team A
2) Select all the students whose fees is greater than or equal to 30000 and belongs to Team D
3) Select all the students of department ECE whose fees is greater than equal to 10000 and belongs to Team other than A

Unary Relational Operations: PROJECT

❑ PROJECT Operation is denoted by **Π** (pi)

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

❑ PROJECT creates a vertical partitioning

❑ General Form :

$$\mathbf{\Pi}_{\text{<List of Attributes>}}\,\mathbf{(R)}$$

Unary Relational Operations: PROJECT

❑ PROJECT Operation is denoted by **Π** (pi)

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

❑ PROJECT creates a vertical partitioning

❑ General Form :

$$\Pi_{\text{<List of Attributes>}}(R)$$

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

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

PROJECT Operation Properties

- ❑ The number of tuples in the result of projection operation is always less or equal to the number of tuples in R

- ❑ 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*

- ❑ $\Pi_{<list1>} (\Pi_{<list2>} (R)) = \Pi_{<list2>} (\Pi_{<list1>} (R))$ not always true

- ❑ $\Pi_{<list1>} (\Pi_{<list2>} (R)) = \Pi_{<list1>} (R)$ as long as <list2> contains the attributes in <list1>

Examples of applying SELECT and PROJECT operations

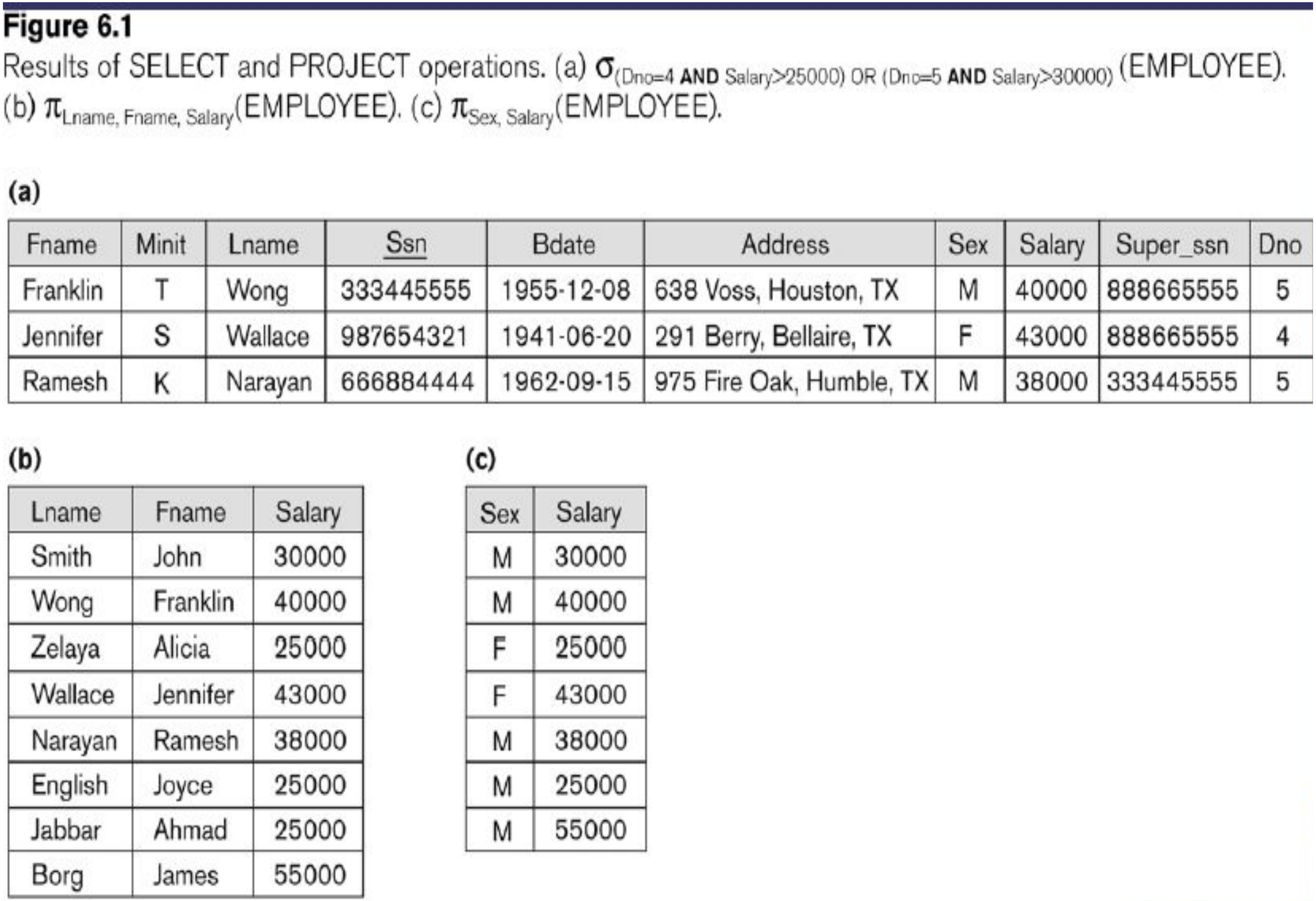


Figure 6.1

Results of SELECT and PROJECT operations. (a) $\sigma_{(\text{Dno}=4 \textbf{ AND } \text{Salary}>25000) \text{ OR } (\text{Dno}=5 \textbf{ AND } \text{Salary}>30000)}$(EMPLOYEE). (b) $\pi_{\text{Lname, Fname, Salary}}$(EMPLOYEE). (c) $\pi_{\text{Sex, Salary}}$(EMPLOYEE).

(a)

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

(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

➲ 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.**

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

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

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

- ➲ We can write a *single relational algebra expression as* follows:

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

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

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

➲ We can write a *single relational algebra expression as* follows:

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

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

$$\textbf{DEP5\_EMPS} \leftarrow \sigma_{\text{DNO=5}}(\textbf{EMPLOYEE})$$
$$\textbf{RESULT} \leftarrow \Pi_{\text{FNAME, LNAME, SALARY}}(\textbf{DEP5\_EMPS})$$

➲ We can write a *single relational algebra expression as* follows:

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

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

$$\textbf{DEP5\_EMPS} \leftarrow \sigma_{\text{DNO=5}}(\textbf{EMPLOYEE})$$
$$\textbf{RESULT} \leftarrow \Pi_{\text{FNAME, LNAME, SALARY}}(\textbf{DEP5\_EMPS})$$

➲ We can write a *single relational algebra expression as* follows:

$$\textbf{RESULT} \leftarrow \Pi_{\text{FNAME, LNAME, SALARY}}(\sigma_{\text{DNO=5}}(\textbf{EMPLOYEE}))$$

*Unary Relational Operations: **RENAME***

➲ The RENAME operator is denoted by ٩ (rho)

➲ In some cases, we may want to *rename the* attributes of a relation or the relation name or both

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

□ ٩ $_{\text{S (B1, B2, …, Bn)}}$(R) changes both:

-- the relation name to S, *and*

-- the column (attribute) names to B1, B1, …..Bn)

□

٩ $_{\text{S}}$(R) changes:

-- the *relation name only to S*

□ ٩ $_{\text{(B1, B2, …, Bn)}}$(R) changes:

-- the *column (attribute) names only to B1, B1, …..Bn)*

Example of applying multiple operations and RENAME

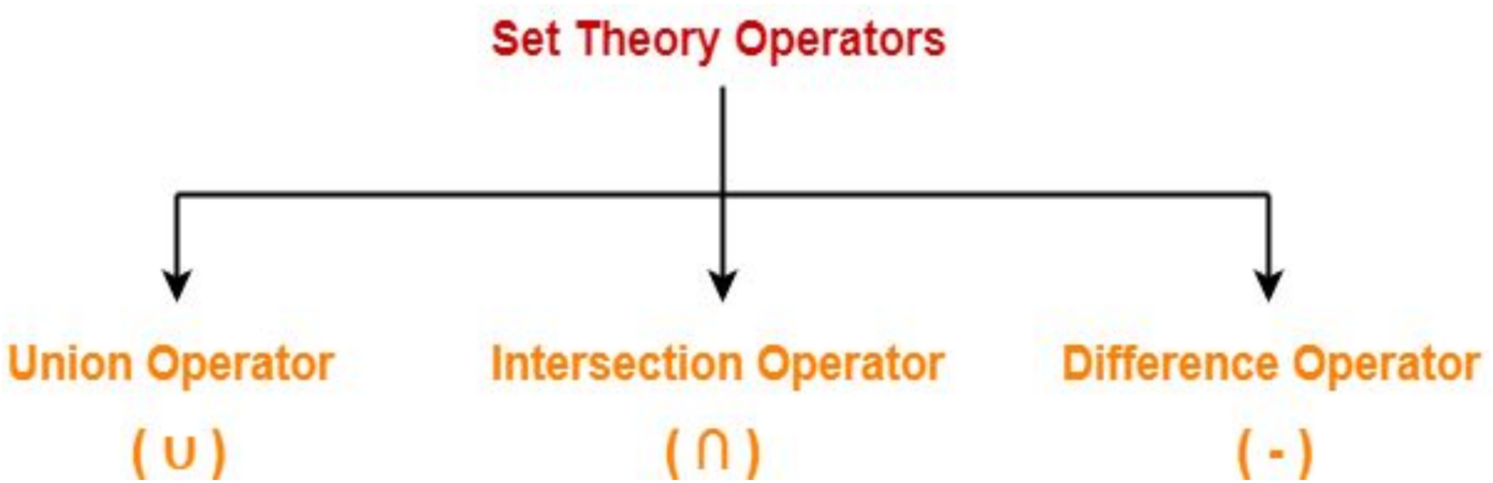
Example-1: Query to rename the relation **Student as SEStudent** and the attributes of Student – **RollNo, SName as (Sno, Name)**.

Example of applying multiple operations and RENAME

Example-1: Query to rename the relation **Student as SEStudent** and the attributes of Student – **RollNo, SName as (Sno, Name)**.

$$\rho_{\text{SEStudent(Sno, Name)}}\ \pi_{\text{RollNo, SName}}(\text{Student})$$

Relational Algebra Operations from Set Theory



Set Theory Operators

Union Operator

(∪)

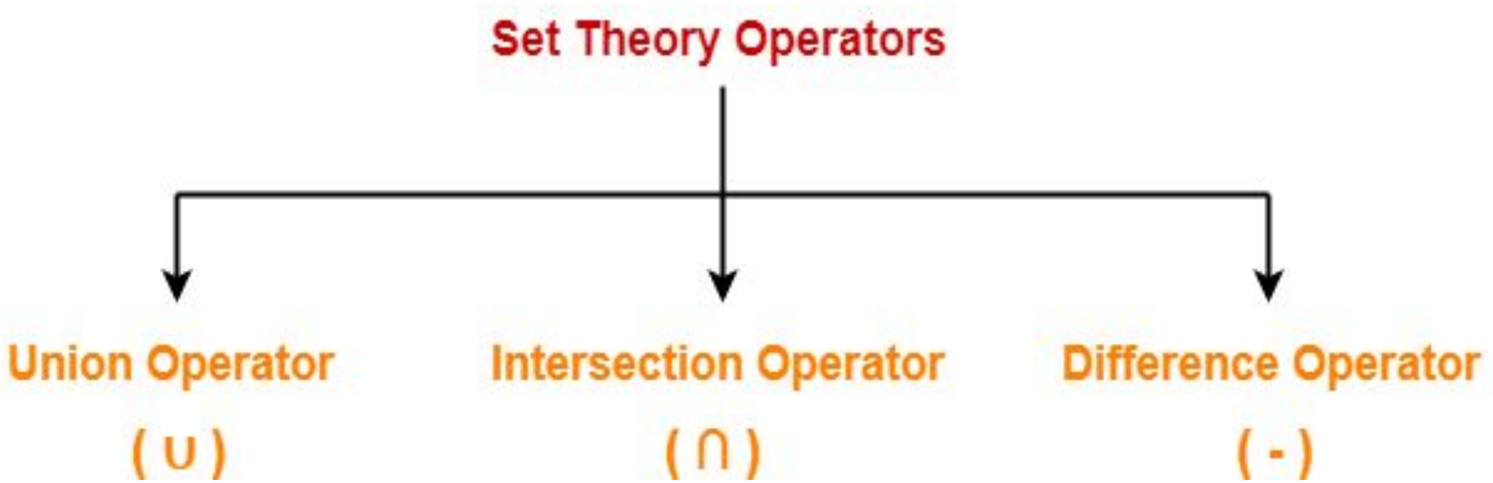
Intersection Operator

(∩)

Difference Operator

(-)

Relational Algebra Operations from Set Theory



Set Theory Operators

Union Operator

(∪)

Intersection Operator

(∩)

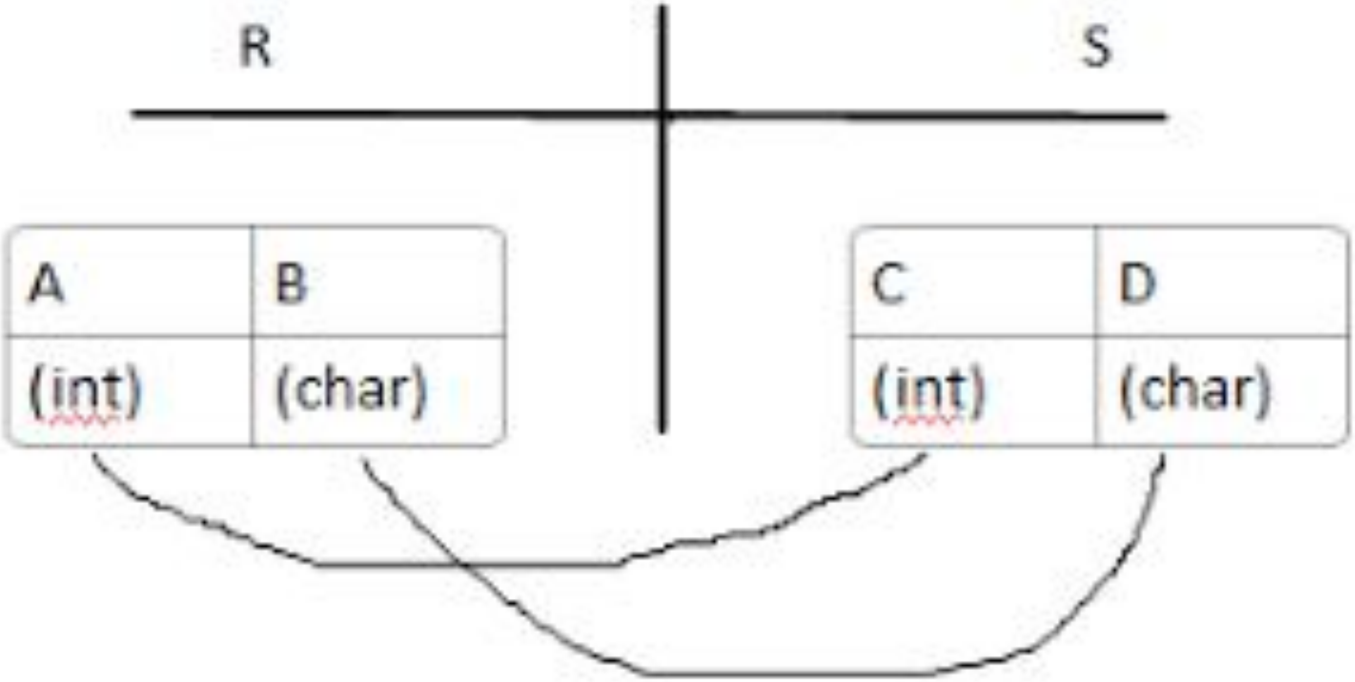
Difference Operator

(-)

- Type Compatibility of operands is required for the binary set operation UNION ‘U’, (also for INTERSECTION ‘n’, and SET DIFFERENCE ‘–’)

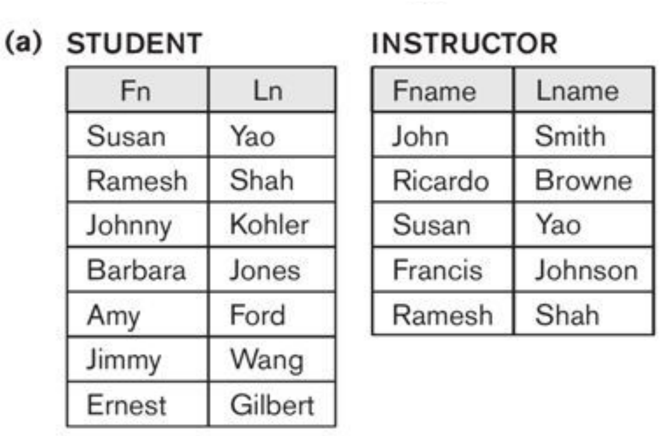
➲ R1(A1, A2, ..., An) and R2(B1, B2, ..., Bn) are type compatible if:

- they have the same number of attributes, and
- the domains of corresponding attributes are type compatible (i.e. dom(Ai)=dom(Bi) for i=1, 2, ..., n).



Are Student and Instructor (both tables) are type compatible ??

Example



(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

Relational Algebra Operations from Set Theory: UNION

➲ UNION Operation

- It is a Binary operation, denoted by ‘U’

- The result of R U 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)

UNION Example

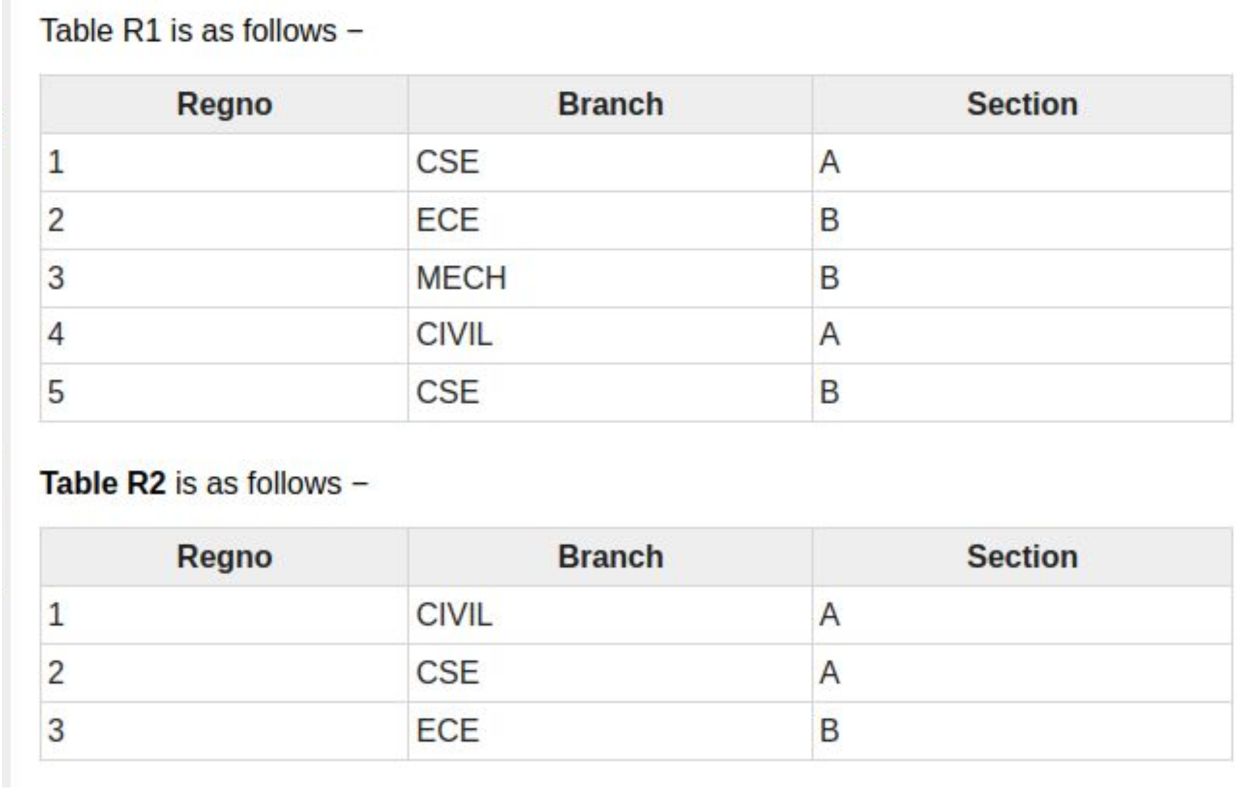


Table R1 is as follows –

Regno	Branch	Section
1	CSE	A
2	ECE	B
3	MECH	B
4	CIVIL	A
5	CSE	B

Table R2 is as follows –

Regno	Branch	Section
1	CIVIL	A
2	CSE	A
3	ECE	B

1) **Union of R1 and R2 …….**

2) **To display all the regno of R1 and R2 …….**

3) **To retrieve branch and section of all the students from R1 and R2 …**

UNION Example

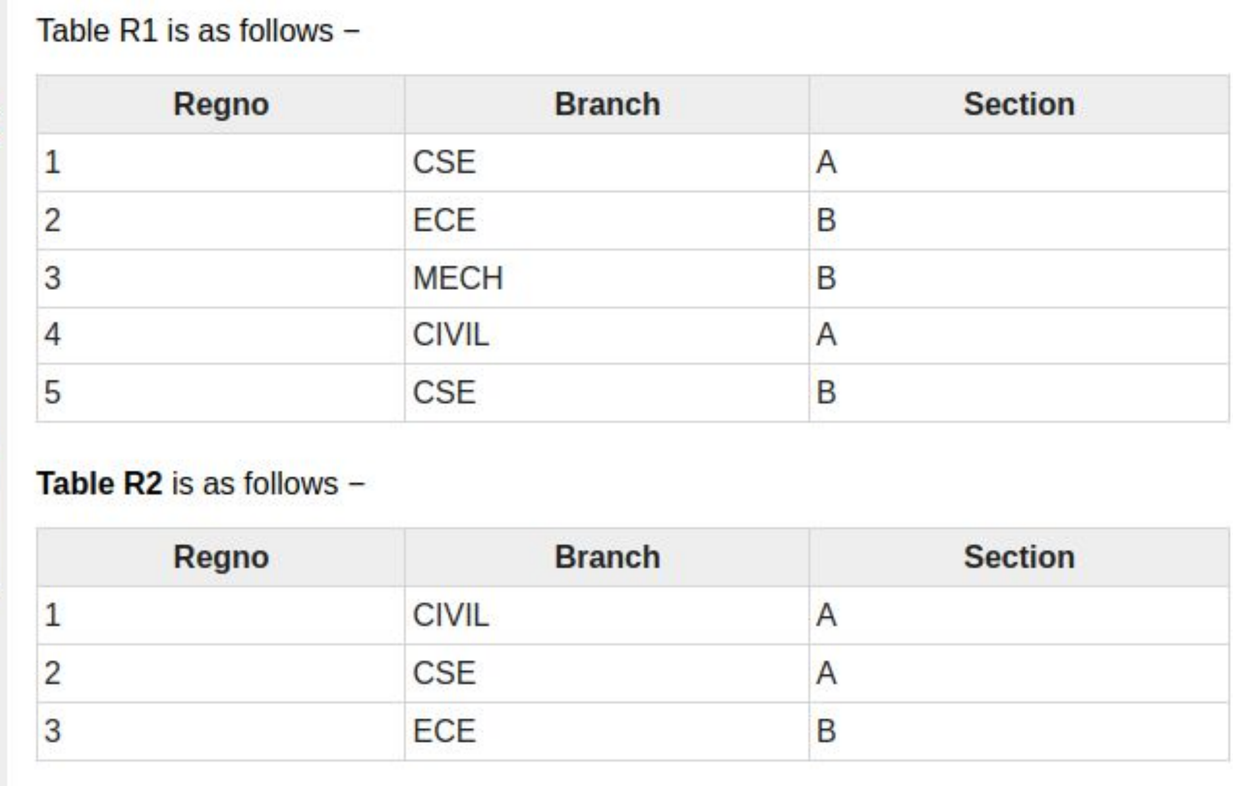


Table R1 is as follows –

Regno	Branch	Section
1	CSE	A
2	ECE	B
3	MECH	B
4	CIVIL	A
5	CSE	B

Table R2 is as follows –

Regno	Branch	Section
1	CIVIL	A
2	CSE	A
3	ECE	B

1) Union of R1 and R2 …….
   Result1 ← R1 u R2

2) To display all the regno of R1 and R2 …….
   Result2 ← ∏regno(R1) U ∏regno(R2)

3) To retrieve branch and section of all the students from R1 and R2 …
   Result3 ← ∏ branch, section (R1) U ∏ branch, section (R2)

Relational Algebra Operations from Set Theory: UNION

➲ UNION Operation

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*

Relational Algebra Operations from Set Theory: UNION

➲ UNION Operation

Example:

To retrieve the social security numbers of all employees who either *work in department 5 (RESULT1) or directly supervise an employee who works in department 5 (RESULT2*)

Relational Algebra Operations from Set Theory: UNION

➲ UNION Operation

Example:

To retrieve the social security numbers of all employees who either *work in department 5 (RESULT1) or directly supervise an employee who works in department 5 (RESULT2*)

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

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

Relational Algebra Operations from Set Theory: UNION

➲ UNION Operation

Example:

To retrieve the social security numbers of all employees who either *work in department 5 (RESULT1) or directly supervise an employee who works in department 5 (RESULT2*)

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

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

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

Relational Algebra Operations from Set Theory: UNION

➲ UNION Operation

Example:
To retrieve the social security numbers of all employees who either *work in department 5 (RESULT1) or directly supervise an employee who works in department 5 (RESULT2)*

DEP5_EMPS ← $\sigma_{DNO=5}$ (EMPLOYEE)
RESULT1 ← Π_{SSN}(DEP5_EMPS)

RESULT2(SSN) ← $\Pi_{SUPERSSN}$(DEP5_EMPS)

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

RESULT ← RESULT1 U RESULT2

Relational Algebra Operations from Set Theory: UNION

➲ UNION Operation

Example:

To retrieve the social security numbers of all employees who either *work in department 5 (RESULT1) or directly supervise an employee who works in department 5 (RESULT2)*

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

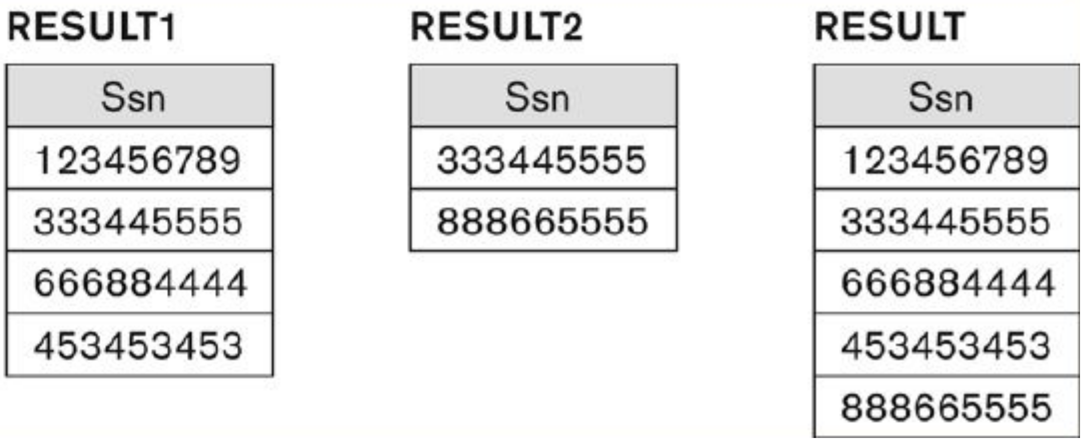
RESULT1 ← Π_{SSN}(DEP5_EMPS)

RESULT2(SSN) ← $\Pi_{SUPERSSN}$(DEP5_EMPS)

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

RESULT ← RESULT1 U RESULT2

Figure 6.3
Result of the UNION operation RESULT ← RESULT1 ∪ RESULT2.



RESULT1

Ssn
123456789
333445555
666884444
453453453

RESULT2

Ssn
333445555
888665555

RESULT

Ssn
123456789
333445555
666884444
453453453
888665555

Relational Algebra Operations from Set Theory: INTERSECTION

➲ INTERSECTION is denoted by ‘n’

➲ The result of the operation R n 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”

Relational Algebra Operations from Set Theory: SET DIFFERENCE

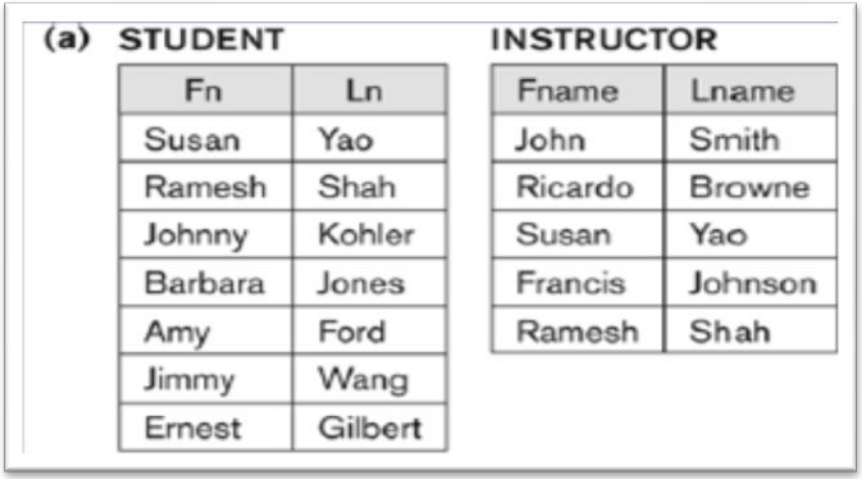
➲ SET DIFFERENCE (also called MINUS or EXCEPT) is denoted by '–'

➲ 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 to illustrate the result of UNION, INTERSECT, and DIFFERENCE



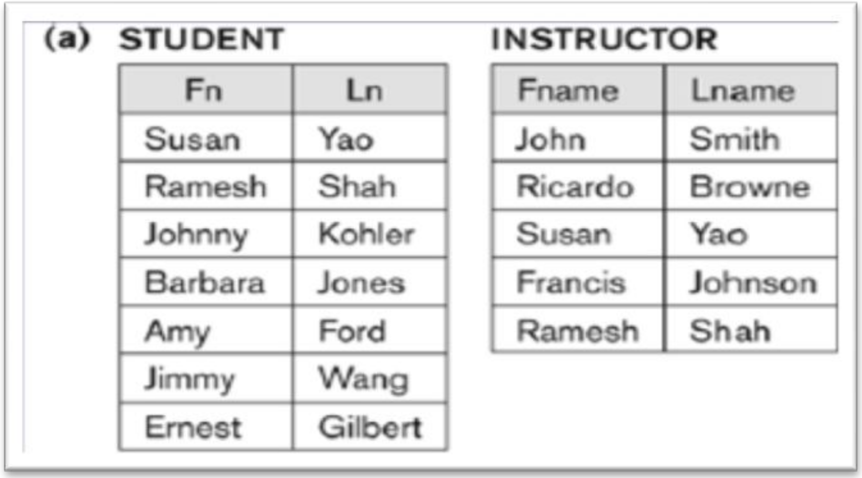
(a) STUDENT

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

Example to illustrate the result of UNION, INTERSECT, and DIFFERENCE



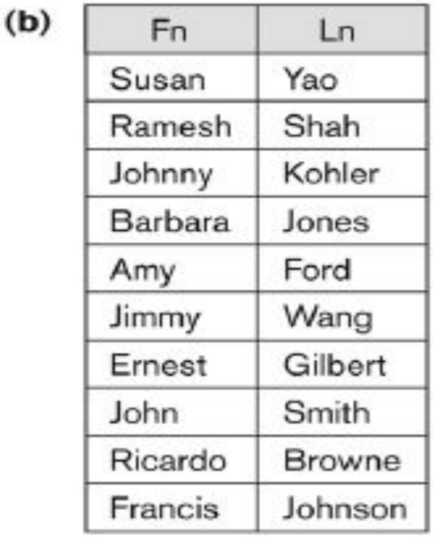
(a) STUDENT

Fn	Ln
Susan	Yao
Ramesh	Shah
Johnny	Kohler
Barbara	Jones
Amy	Ford
Jimmy	Wang
Ernest	Gilbert

INSTRUCTOR

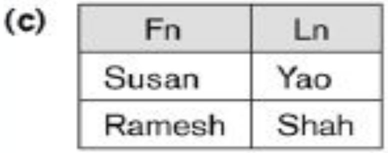
Fname	Lname
John	Smith
Ricardo	Browne
Susan	Yao
Francis	Johnson
Ramesh	Shah

The set operations UNION, INTERSECTION, and MINUS. (a) Two union-compatible relations.



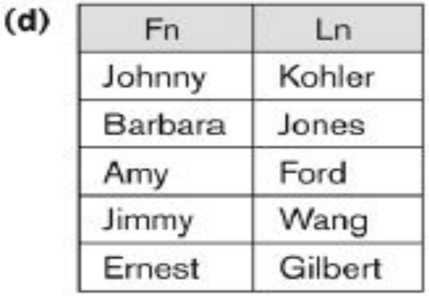
(b)

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



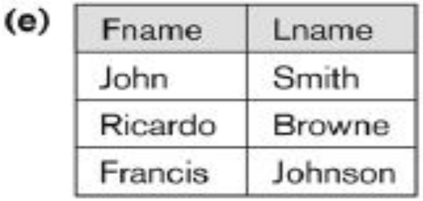
(c)

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

(b) STUDENT ∪ INSTRUCTOR. (c) STUDENT ∩ INSTRUCTOR. (d) STUDENT − INSTRUCTOR.
(e) INSTRUCTOR − STUDENT.

Some properties of UNION, INTERSECT, and DIFFERENCE

➲ Notice that both union and intersection are *commutative* operations; that is

- R u S = S u R, and R n S = S n 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 u (S u T) = (R u S) u T
- (R n S) n T = R n (S n T)

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

- R – S ≠ S – R

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 ‘x’

□ Example : R(A1, A2, . . ., An) x S(B1, B2, . . ., Bm)

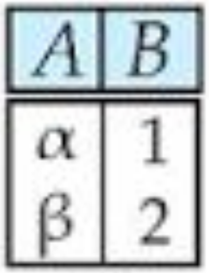
Result is a relation Q with degree **n + m attributes**:
Q(A1, A2, . . ., An, B1, B2, . . ., Bm), in that order.

- 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.**

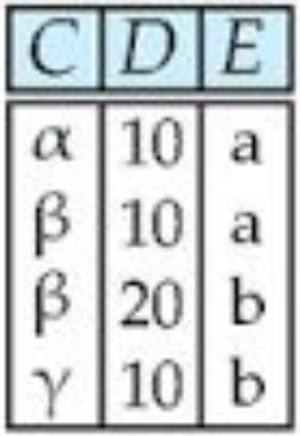
□ **The two operands do NOT have to be "type compatible”**

CARTESIAN PRODUCT
Example



A	*B*
α	1
β	2

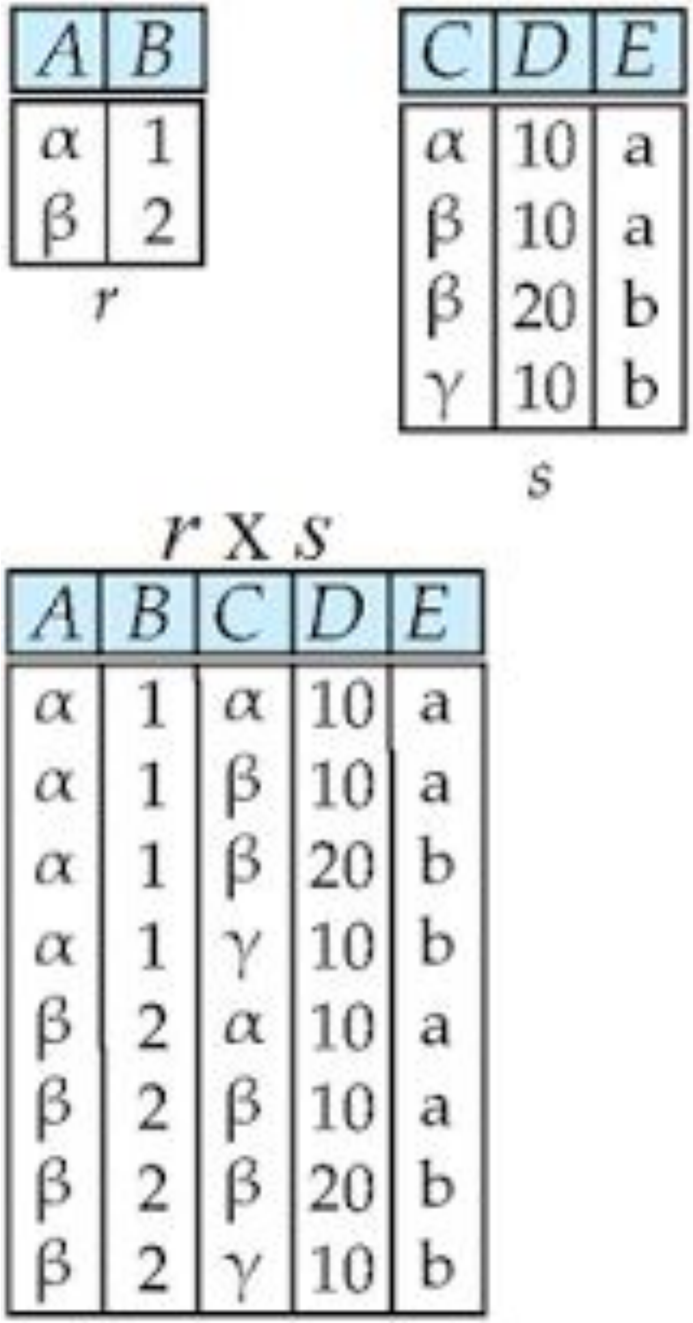
r



C	*D*	*E*
α	10	a
β	10	a
β	20	b
γ	10	b

s

CARTESIAN PRODUCT
Example



A	B
α	1
β	2

r

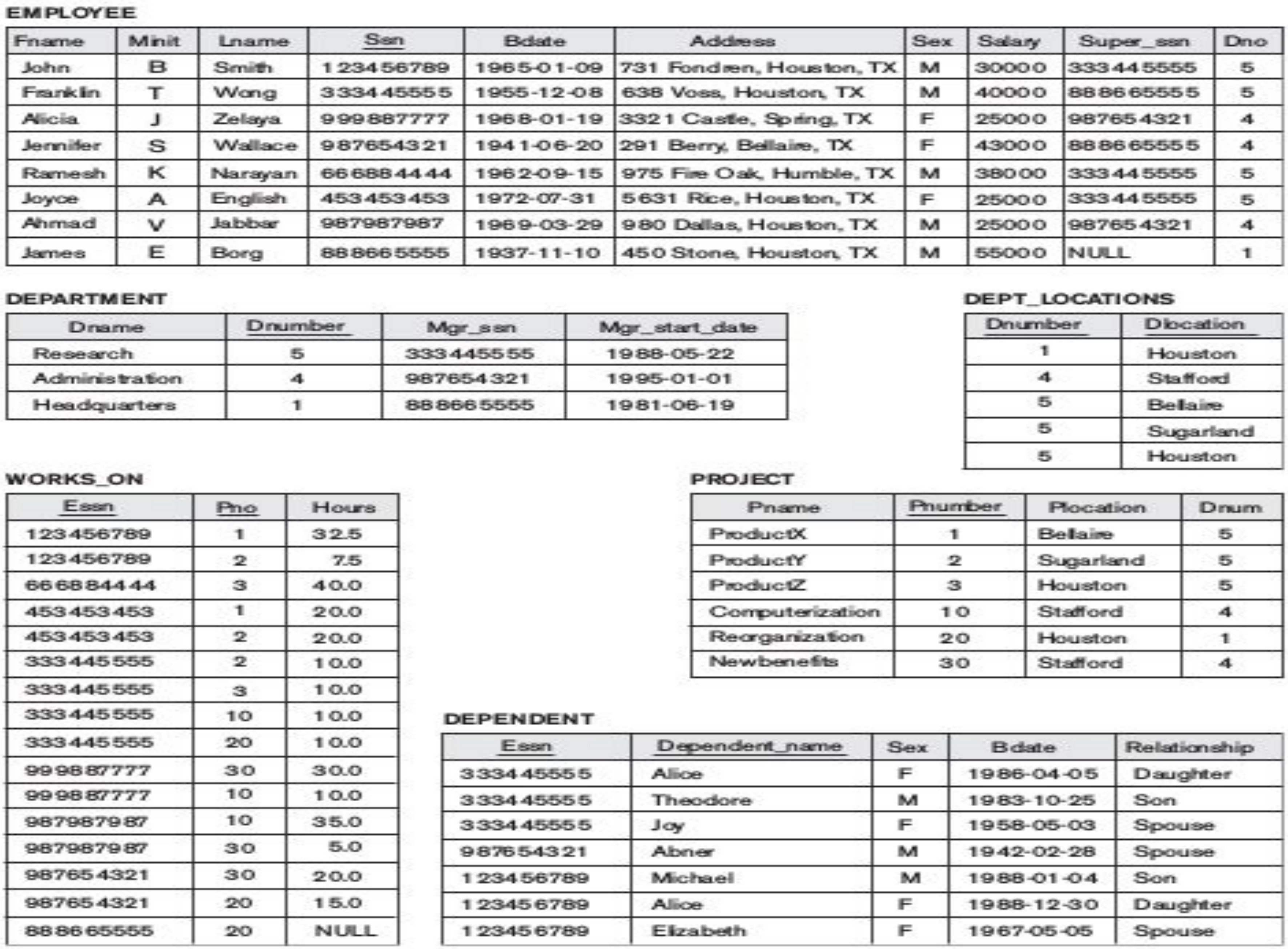
C	D	E
α	10	a
β	10	a
β	20	b
γ	10	b

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

Figure 3.6
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

Relational Algebra Operations from Set Theory: CARTESIAN PRODUCT (cont.)

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

➲ **Example (not meaningful):**

Suppose we want to retrieve a list of names of each female employee's dependents.

Solution:

FEMALE_EMPS ← $\sigma_{\text{SEX='F'}}$(EMPLOYEE)

EMPNAMES ← $\Pi_{\text{FNAME, LNAME, SSN}}$ (FEMALE_EMPS)

□ EMP_DEPENDENTS ← EMPNAMES x DEPENDENT

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

Figure 6.5
The Cartesian Product (Cross Product) operation.

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 (cont.)

➲ To keep only combinations where the DEPENDENT is related to the EMPLOYEE, we add **a SELECT operation** as follows:

➲ **Example (meaningful)**:

Suppose we want to retrieve a list of names of each female employee's dependents.

Solution:

□ FEMALE_EMPS ← $\sigma_{\text{SEX='F'}}$(EMPLOYEE)

□ EMPNAMES ← $\Pi_{\text{FNAME, LNAME, SSN}}$ (FEMALE_EMPS)

□ EMP_DEPENDENTS ← EMPNAMES x DEPENDENT

□ **ACTUAL_DEPS ← $\sigma_{\text{SSN=ESSN}}$(EMP_DEPENDENTS)**

□ **RESULT ← $\Pi_{\text{FNAME, LNAME, DEPENDENT\_NAME}}$ (ACTUAL_DEPS)**

➲ **RESULT will now contain the name of female employees and their dependents**

Figure 6.5
The Cartesian Product (Cross Product) operation.

FEMALE_EMPS

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	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

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	. . .

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** ⋈)

□ The sequence of **CARTESIAN PRODUCT followed by SELECT** is used quite commonly to identify and select related tuples from two relations, **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(A1,A2, . . ., An) and S(B1, B2, . . ., Bm) is:

$$\mathbf{R} \bowtie_{\textbf{<join condition>}} \mathbf{S}$$

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

Binary Relational Operations: JOIN (cont.)

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

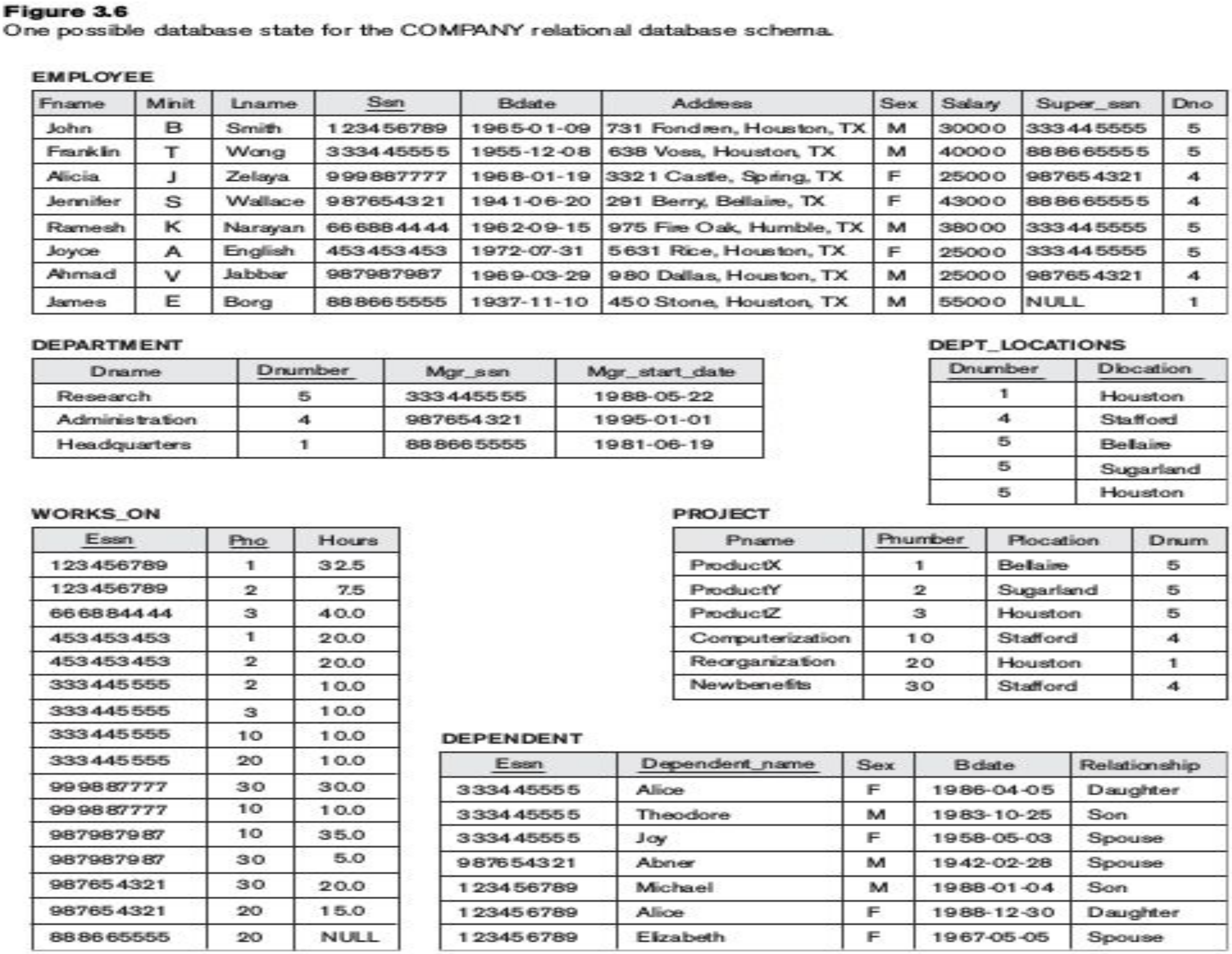


Figure 3.6
One possible database state for the COMPANY relational database schema.

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

Binary Relational Operations: JOIN (cont.)

➲ 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.

Binary Relational Operations: JOIN (cont.)

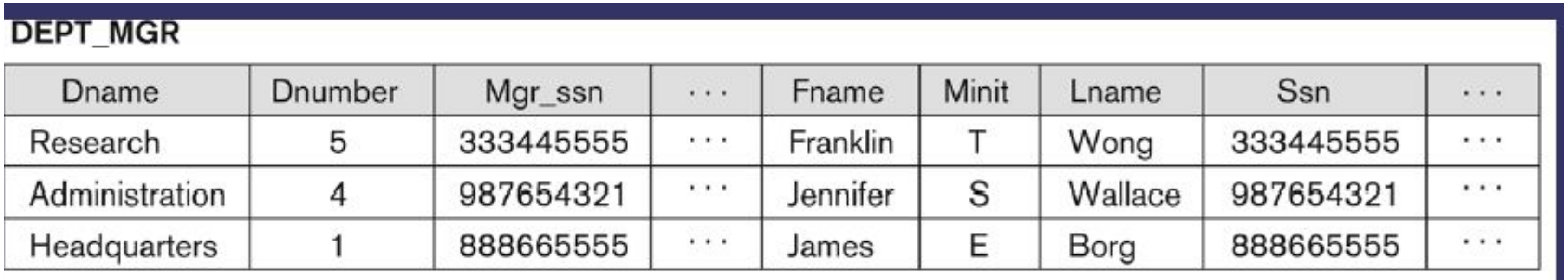
➲ Example: Suppose that we want to retrieve the name of the manager of each department.

□ 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.

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

➲ where MGRSSN=SSN is the join condition
□ Combines each department record with the employee who manages the department

□ The join condition can also be specified as :
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	. . .

Figure 6.6
Result of the JOIN operation

Some properties of JOIN

➲ Consider the following JOIN operation:

□ R(A1, A2, . . ., An) $\underset{R.Ai= S.Bj}{\bowtie}$ S(B1, B2, . . ., Bm)

□ Result is a relation Q with degree n + m attributes:
Q(A1, A2, . . ., An, B1, B2, . . ., Bm), in that order.

□ The resulting relation state has one tuple for each combination of tuples—r from R and s from S, but *only if they satisfy the join condition r[Ai]=s[Bj]*

□ Hence, if R has n_R tuples, and S has n_S tuples, then the join result will generally have ***less than* $\mathbf{n_R * n_S}$ *tuples.***

□ **Only related tuples (based on the join condition) will appear in the result**

□ The general case of JOIN operation is called a Theta-join: R $\bowtie_{theta}$ S

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.

Binary Relational Operations: NATURAL JOIN Operation

□ Another variation of JOIN called **NATURAL JOIN** — **denoted by '*'**

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

Binary Relational Operations: NATURAL JOIN Operation

□ Another variation of JOIN called **NATURAL JOIN — denoted by '*'**

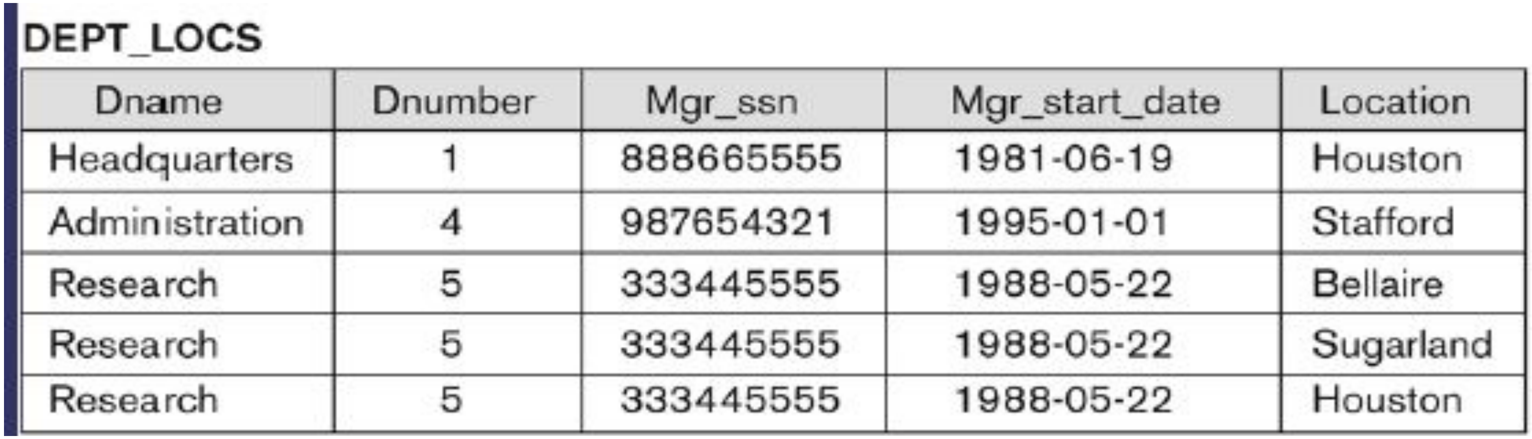
□ The standard definition of natural join requires that the two join attributes, or each pair of corresponding join attributes, ***have the same name in both relations***

□ If this is not the case, a renaming operation is applied first.

➲ **Example 1: 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



DEPT_LOCS

Dname	Dnumber	Mgr_ssn	Mgr_start_date	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

Binary Relational Operations NATURALnJOIN (contd.)

- **Example 2: To apply a natural join on DEPARTMENT and Project:**

Binary Relational Operations NATURALnJOIN (contd.)

➲ **Example 2: To apply a natural join on DEPARTMENT and Project:**

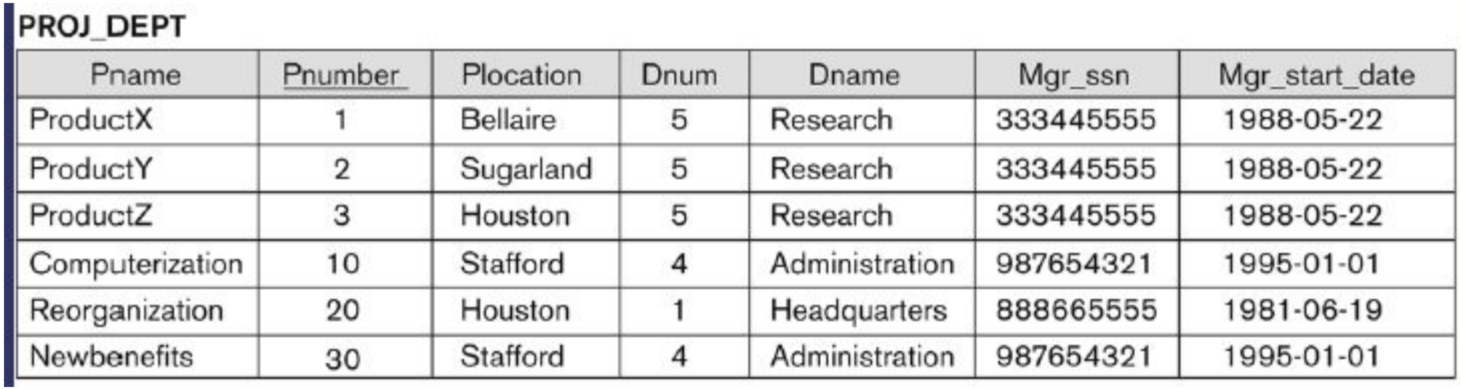
✔Then we have to first rename Dnumber attribute of DEPARTMENT to DNUM
✔And then apply Natural JOIN

PROJ_DEPT PROJECT * ٩ (Dname, Dnum ,mgr_ssn , mgr_start_Date) DEPARTMENT

OR

DEPT ٩ (Dname, Dnum ,mgr_ssn , mgr_start_Date) DEPARTMENT

PROJ_DEPT PROJECT * DEPT



PROJ_DEPT

Pname	Pnumber	Plocation	Dnum	Dname	Mgr_ssn	Mgr_start_date
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

Complete Set of Relational Operations

➲ The set of operations including:

SELECT (symbol: **σ** (sigma))

PROJECT (symbol: **Π** (pi))

UNION (**U**)

DIFFERENCE (**-**)

RENAME (symbol: **٩** (rho))

CARTESIAN PRODUCT (**X**)

- Is called a *complete set because any other* relational algebra expression can be expressed by a combination of these operations.

➲ For example:

□ R n S = (R u S) – ((R - S) u (S - R))

□ $R \bowtie_{\text{<join condition>}} S = \sigma_{\text{<join condition>}} (R \; X \; S)$

Binary Relational Operations: DIVISION

□ The division operation is applied to two relations R(Z) , S(X), where X subset Z.

□ 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.

Example: T □ R ÷ S

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

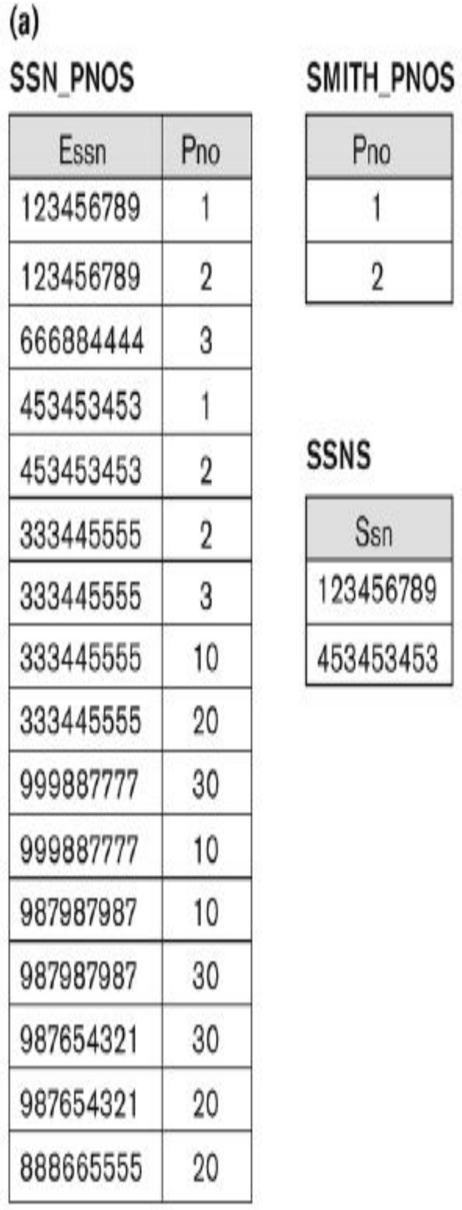
=

T

B
b1
b4

Example of DIVISION

Retrieve the names of all employees who work on all the projects that 'John Smith' works on.



(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

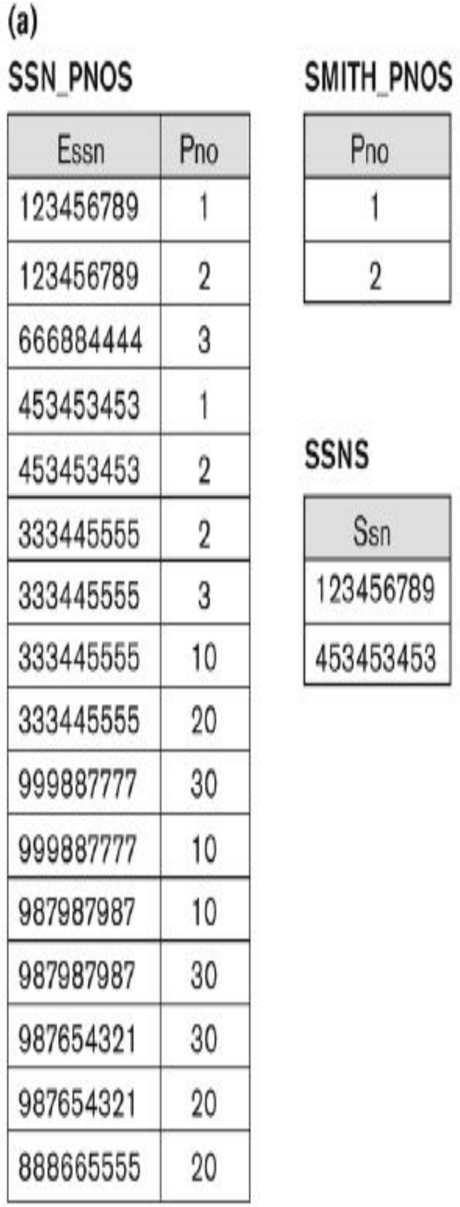
Example of DIVISION

Retrieve the names of all employees who work on all the projects that 'John Smith' works on.

1. ***Retrieve the list of project numbers that 'John Smith' works on:***

2. ***Retrieve all employee's ESSN and Pno who works on the project:***

3. ***Finally , apply division operation which gives desired employee's SSN.***



(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

Example of DIVISION

Retrieve the names of all employees who work on all the projects that ‘John Smith’ works on.

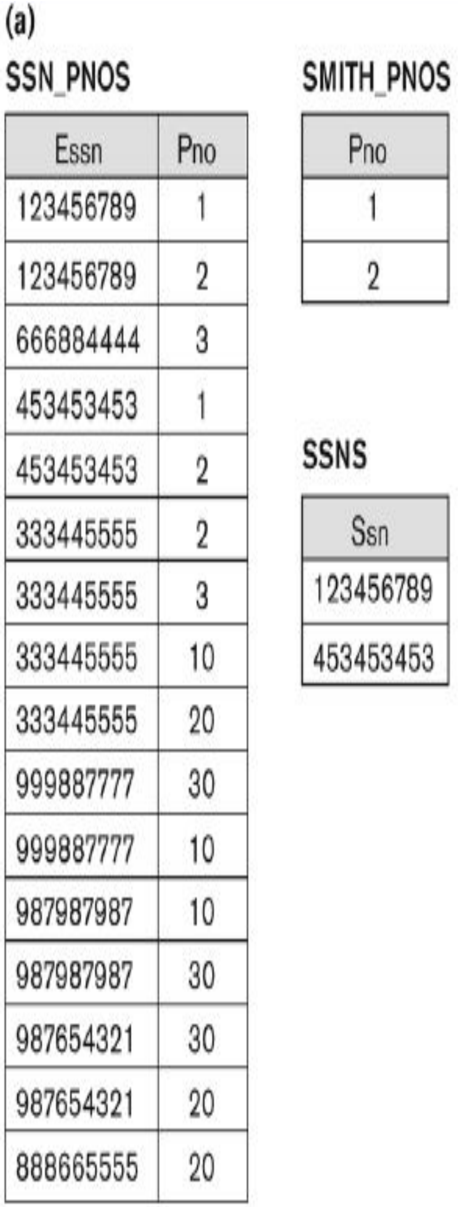
1. ***Retrieve the list of project numbers that ‘John Smith’ works on:***

$SMITH \; \square \; \sigma_{fname=‘John’ \; AND \; Lname=‘Smith’}(EMPLOYEE)$

$SMITH\_PNOS \; \square \; \Pi_{Pno} \; (WORKS\_ON \bowtie_{ESSN=SSN} SMITH)$

2. ***Retrieve all employee’s ESSN and Pno who works on the project:***

3 . ***Finally , apply division operation which gives desired employee’s SSN.***



(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

Example of DIVISION

Retrieve the names of all employees who work on all the projects that 'John Smith' works on.

1. ***Retrieve the list of project numbers that 'John Smith' works on:***

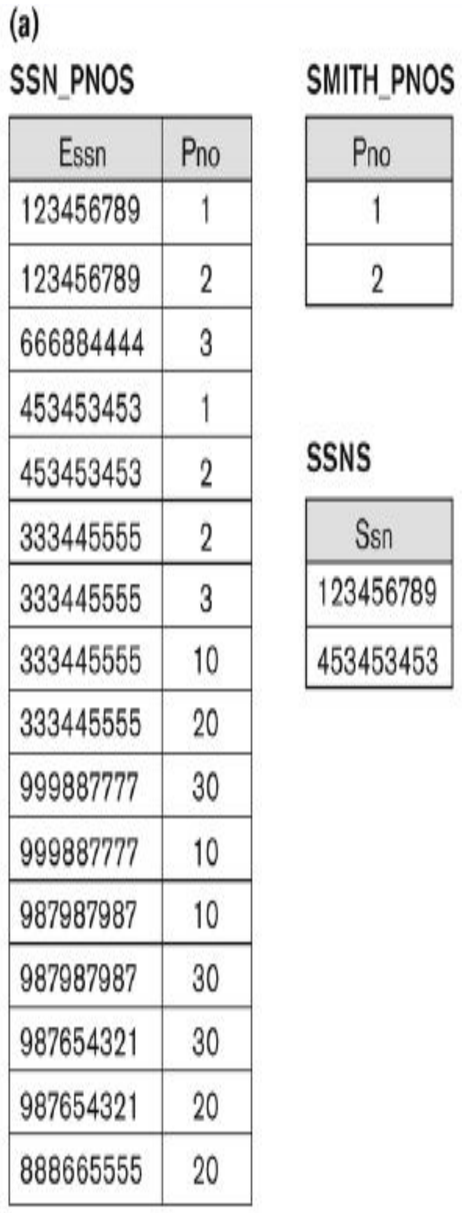
SMITH □ $\sigma_{\text{fname='John' AND Lname='Smith'}}$ (EMPLOYEE)

SMITH_PNOS □ Π_{Pno} (WORKS_ON $\bowtie_{\text{ESSN=SSN}}$ SMITH)

2. ***Retrieve all employee's ESSN and Pno who works on the project:***

SSN_PNOS □ $\Pi_{\text{ESSN , Pno}}$ (WORKS_ON)

3 . ***Finally , apply division operation which gives desired employee's SSN.***



(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

Example of DIVISION

Retrieve the names of all employees who work on all the projects that 'John Smith' works on.

1. ***Retrieve the list of project numbers that 'John Smith' works on:***

$$SMITH \leftarrow \sigma_{\text{fname='John' AND Lname='Smith'}}(EMPLOYEE)$$

$$SMITH\_PNOS \leftarrow \Pi_{Pno}\ (WORKS\_ON \bowtie_{ESSN=SSN} SMITH)$$

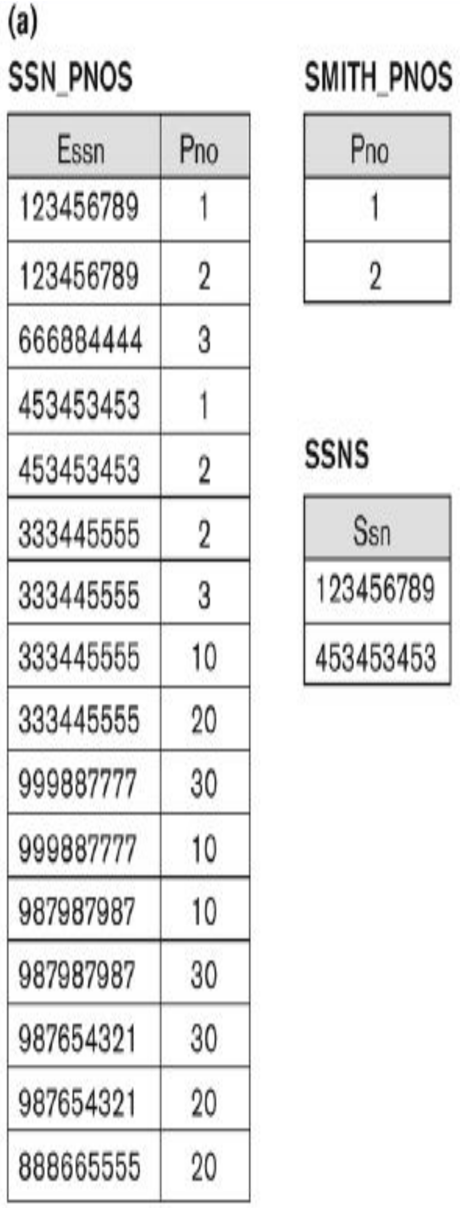
2. ***Retrieve all employee's ESSN and Pno who works on the project:***

$$SSN\_PNOS \leftarrow \Pi_{ESSN\ ,\ Pno}\ (WORKS\_ON)$$

3. ***Finally , apply division operation which gives desired employee's SSN.***

$$SSNS(SSN) \leftarrow SSN\_PNOS \div SMITH\_PNOS$$

$$RESULT \leftarrow \Pi_{Fname,\ Lname}\ (SSNS * EMPLOYEE)$$



(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

Recap of Relational Algebra Operations

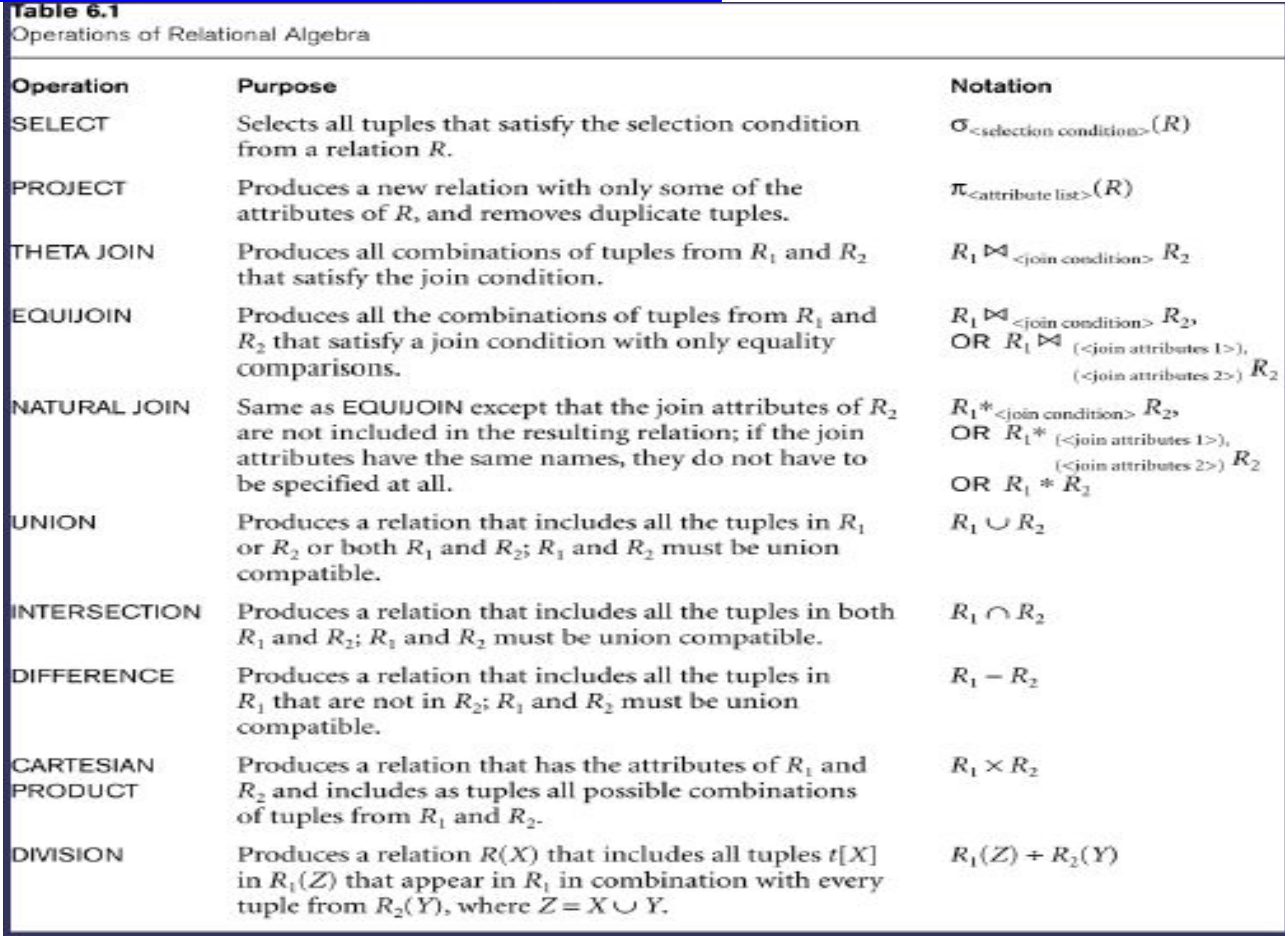


Table 6.1

Operations of Relational Algebra

Operation	Purpose	Notation
SELECT	Selects all tuples that satisfy the selection condition from a relation R.	$\sigma_{\text{<selection condition>}}(R)$
PROJECT	Produces a new relation with only some of the attributes of R, and removes duplicate tuples.	$\pi_{\text{<attribute list>}}(R)$
THETA JOIN	Produces all combinations of tuples from R_1 and R_2 that satisfy the join condition.	$R_1 \bowtie_{\text{<join condition>}} R_2$
EQUIJOIN	Produces all the combinations of tuples from R_1 and R_2 that satisfy a join condition with only equality comparisons.	$R_1 \bowtie_{\text{<join condition>}} R_2$, OR $R_1 \bowtie_{(\text{<join attributes 1>}),\,(\text{<join attributes 2>})} R_2$
NATURAL JOIN	Same as EQUIJOIN except that the join attributes of R_2 are not included in the resulting relation; if the join attributes have the same names, they do not have to be specified at all.	$R_1 *_{\text{<join condition>}} R_2$, OR $R_1 *_{(\text{<join attributes 1>}),\,(\text{<join attributes 2>})} R_2$ OR $R_1 * R_2$
UNION	Produces a relation that includes all the tuples in R_1 or R_2 or both R_1 and R_2; R_1 and R_2 must be union compatible.	$R_1 \cup R_2$
INTERSECTION	Produces a relation that includes all the tuples in both R_1 and R_2; R_1 and R_2 must be union compatible.	$R_1 \cap R_2$
DIFFERENCE	Produces a relation that includes all the tuples in R_1 that are not in R_2; R_1 and R_2 must be union compatible.	$R_1 - R_2$
CARTESIAN PRODUCT	Produces a relation that has the attributes of R_1 and R_2 and includes as tuples all possible combinations of tuples from R_1 and R_2.	$R_1 \times R_2$
DIVISION	Produces a relation $R(X)$ that includes all tuples $t[X]$ in $R_1(Z)$ that appear in R_1 in combination with every tuple from $R_2(Y)$, where $Z = X \cup Y$.	$R_1(Z) \div R_2(Y)$