



Institut Informatika & Bisnis  
**DARMAJAYA**  
Yayasan Alfian Husin



**Kampus  
Merdeka**  
INDONESIA JAYA

**MERDEKA  
BELAJAR**

DATA SCIENCE DARMAJAYA  
“YOUR BEST FUTURE IN DATA”

MEETING: [5 & 6]

# **THE RELATIONAL LANGUAGE**

BY: HENDRA KURNIAWAN



# The Relational Language

1. Relational Algebra.
2. Relational Calculus.



# Learning Objectives

1. The meaning of the term “relational completeness”.
2. How to form queries in the relational algebra.
3. How to form queries in the tuple relational calculus.
4. How to form queries in the domain relational calculus.



# Relational Algebra

The relational algebra is a theoretical language with operations that work on one or more relations to define another relation without changing the original relation(s).

The relational algebra is a relation-at-a-time (or set) language in which all tuples, possibly from several relations, are manipulated in one statement without looping. There are several variations of syntax for relational algebra commands and we use a common symbolic notation for the commands and present it informally.



# 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
  - This property makes the algebra “closed” (all objects in relational algebra are relations)



# Relational Algebra

- The **algebra operations** thus produce new relations
  - These can be further manipulated using operations of the same algebra
- 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)

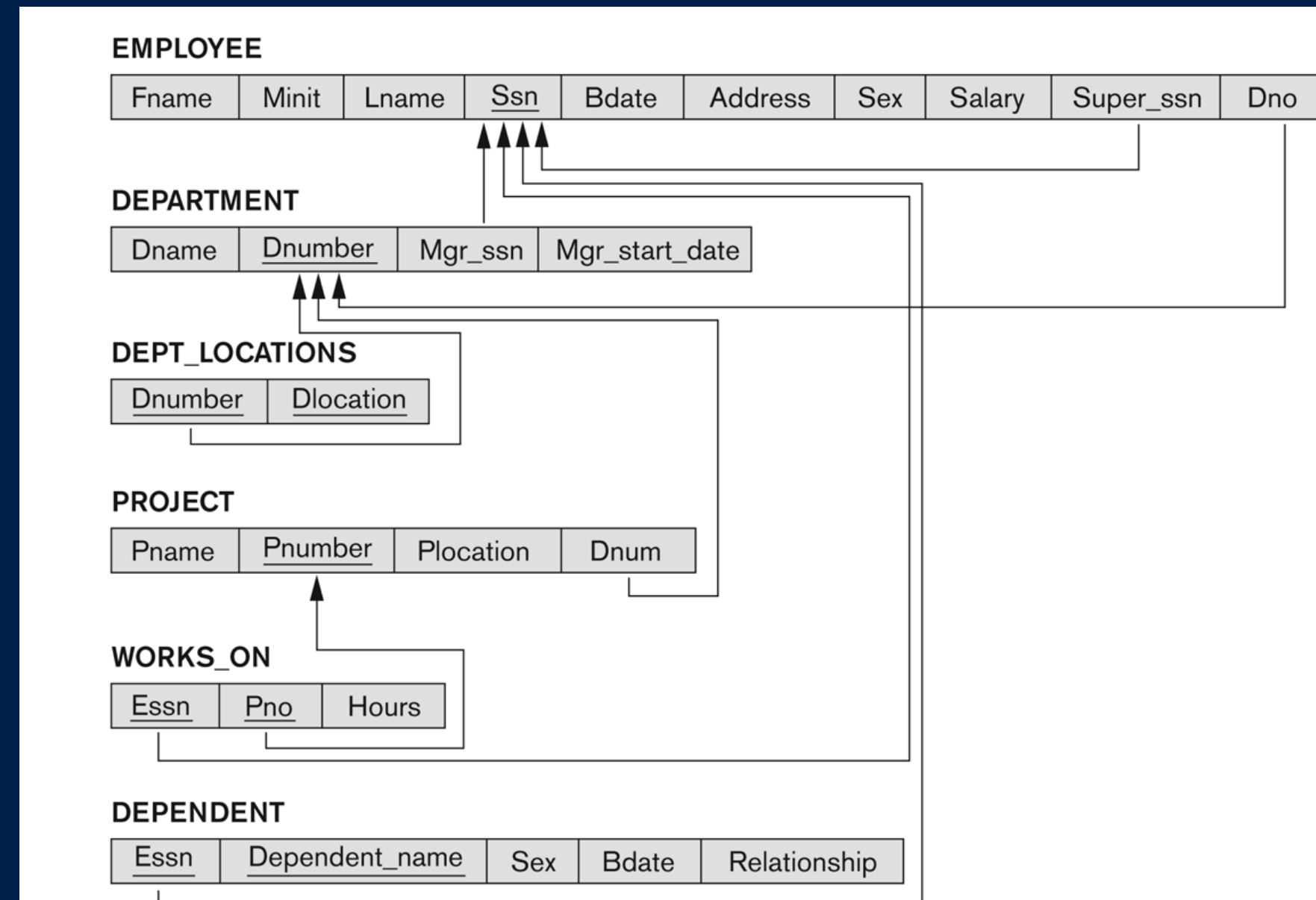


# Relational Algebra

- Relational Algebra consists of several groups of operations
  - Unary Relational Operations
    - SELECT (symbol:  $\sigma$  (sigma))
    - PROJECT (symbol:  $\pi$  (pi))
    - RENAME (symbol:  $\rho$  (rho))
  - Relational Algebra Operations From Set Theory
    - UNION ( $\cup$ ), INTERSECTION ( $\cap$ ), DIFFERENCE (or MINUS,  $-$ )
    - CARTESIAN PRODUCT ( $\times$ )
  - 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

All examples discussed below refer to the COMPANY database shown here.



# Unary Relational Operations: SELECT

- The SELECT operation (denoted by  $\sigma$  (sigma)) is used to select a *subset* of the tuples from a relation based on a **selection condition**.
  - 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*)
- Examples:
  - Select the EMPLOYEE tuples whose department number is 4:

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

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

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



## Unary Relational Operations: SELECT

–In general, the *select* operation is denoted by  $\sigma_{\langle \text{selection condition} \rangle}(R)$   
where

- the symbol  $\sigma$  (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

# Unary Relational Operations: SELECT

- SELECT Operation Properties

- The SELECT operation  $\sigma_{\langle \text{selection condition} \rangle}(R)$  produces a relation S that has the same schema (same attributes) as R

- SELECT  $\sigma$  is commutative:

- $\sigma_{\langle \text{condition1} \rangle}(\sigma_{\langle \text{condition2} \rangle}(R)) = \sigma_{\langle \text{condition2} \rangle}(\sigma_{\langle \text{condition1} \rangle}(R))$

- Because of commutativity property, a cascade (sequence) of SELECT operations may be applied in any order:

- $\sigma_{\langle \text{cond1} \rangle}(\sigma_{\langle \text{cond2} \rangle}(\sigma_{\langle \text{cond3} \rangle}(R))) = \sigma_{\langle \text{cond2} \rangle}(\sigma_{\langle \text{cond3} \rangle}(\sigma_{\langle \text{cond1} \rangle}(R)))$

- A cascade of SELECT operations may be replaced by a single selection with a conjunction of all the conditions:

- $\sigma_{\langle \text{cond1} \rangle}(\sigma_{\langle \text{cond2} \rangle}(\sigma_{\langle \text{cond3} \rangle}(R))) = \sigma_{\langle \text{cond1} \rangle \text{ AND } \langle \text{cond2} \rangle \text{ AND } \langle \text{cond3} \rangle}(R)$

- 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

# The following query results refer to this database state

**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				DEPT_LOCATIONS	
Dname	Dnumber	Mgr_ssn	Mgr_start_date	Dnumber	Dlocation
Research	5	333445555	1988-05-22	1	Houston
Administration	4	987654321	1995-01-01	4	Stafford
Headquarters	1	888665555	1981-06-19	5	Bellaire
				5	Sugarland
				5	Houston

WORKS_ON			PROJECT			
Essn	Pno	Hours	Pname	Pnumber	Plocation	Dnum
123456789	1	32.5	ProductX	1	Bellaire	5
123456789	2	7.5	ProductY	2	Sugarland	5
666884444	3	40.0	ProductZ	3	Houston	5
453453453	1	20.0	Computerization	10	Stafford	4
453453453	2	20.0	Reorganization	20	Houston	1
333445555	2	10.0	Newbenefits	30	Stafford	4
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				

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

- PROJECT Operation is denoted by  $\pi$  (pi)
- 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
- 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 general form of the *project* operation is:

$$\pi_{\langle \text{attribute list} \rangle}(R)$$

- $\pi$  (pi) is the symbol used to represent the *project* operation
- $\langle \text{attribute list} \rangle$  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.



# Unary Relational Operations: PROJECT

- PROJECT Operation Properties
  - The number of tuples in the result of projection  $\pi_{\langle \text{list} \rangle}(R)$  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
    - $\pi_{\langle \text{list1} \rangle}(\pi_{\langle \text{list2} \rangle}(R)) = \pi_{\langle \text{list1} \rangle}(R)$  as long as  $\langle \text{list2} \rangle$  contains the attributes in  $\langle \text{list1} \rangle$

# Examples of applying SELECT and PROJECT operations

**Figure 6.1**

Results of SELECT and PROJECT operations. (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	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**.
- In the later case, we must give names to the relations that hold the intermediate results.

## Single expression versus 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:

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

- 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 denoted by  $\rho$  (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)

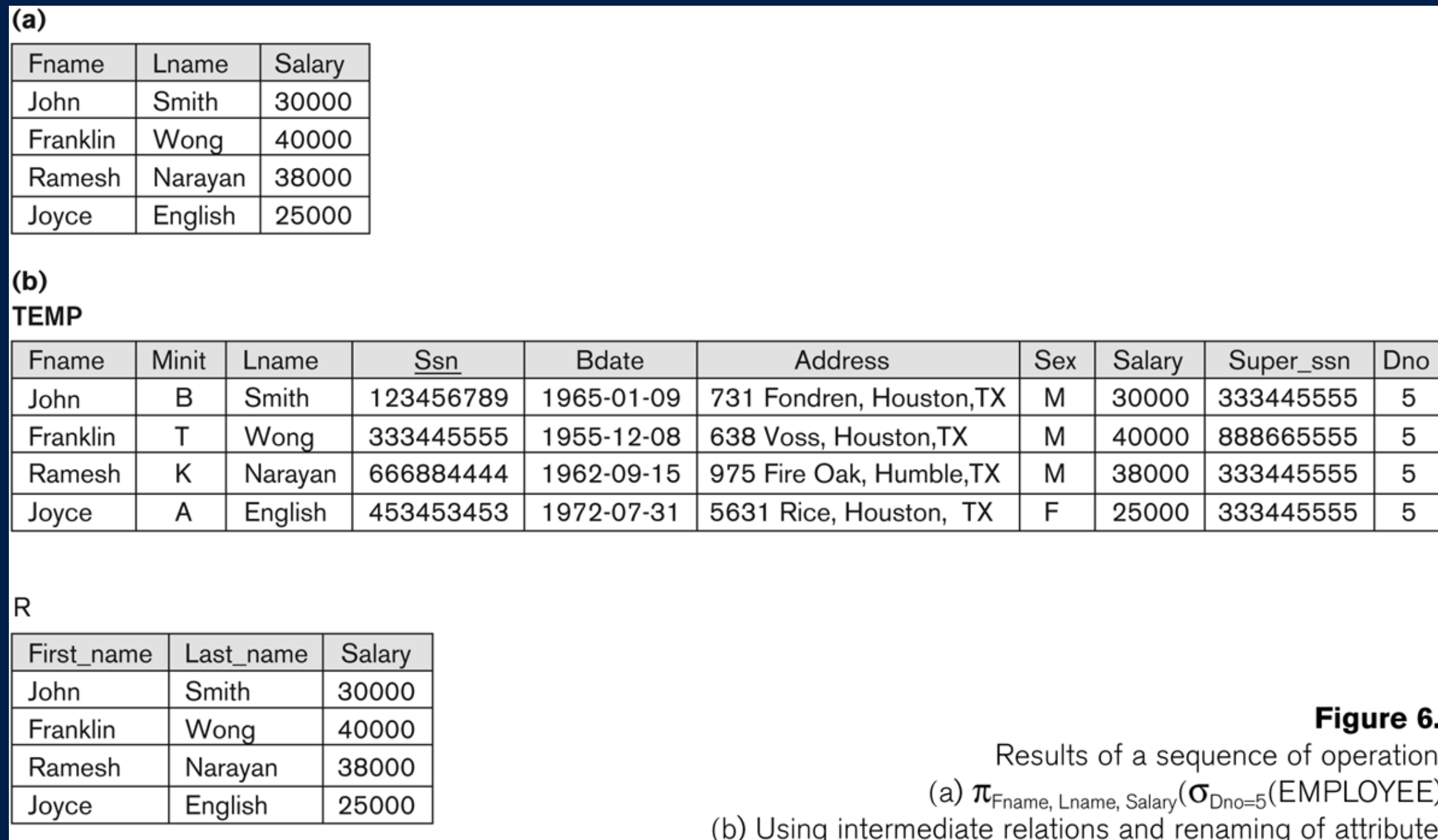
# Unary Relational Operations: RENAME

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

## Unary Relational Operations: RENAME

- For convenience, we also use a *shorthand* for renaming attributes in an intermediate relation:
  - If we write:
    - $\text{RESULT} \leftarrow \pi_{\text{FNAME, LNAME, SALARY}} (\text{DEP5\_EMPS})$
    - RESULT will have the *same attribute names* as DEP5\_EMPS (same attributes as EMPLOYEE)
  - If we write:
    - $\text{RESULT} (\text{F, M, L, S, B, A, SX, SAL, SU, DNO}) \leftarrow \pi_{\text{FNAME, LNAME, SALARY}} (\text{DEP5\_EMPS})$
    - The 10 attributes of DEP5\_EMPS are *renamed* to F, M, L, S, B, A, SX, SAL, SU, DNO, respectively

# Example of applying multiple operations and RENAME





# Relational Algebra Operations from Set Theory: UNION

- UNION Operation
  - 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)

# Relational Algebra Operations from Set Theory: UNION

- Example:
  - 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* (RESULT2 below)
  - We can use the UNION operation as follows:

$$\text{DEP5\_EMPS} \leftarrow \sigma_{\text{DNO}=5} (\text{EMPLOYEE})$$
$$\text{RESULT1} \leftarrow \pi_{\text{SSN}} (\text{DEP5\_EMPS})$$
$$\text{RESULT2}(\text{SSN}) \leftarrow \pi_{\text{SUPERSSN}} (\text{DEP5\_EMPS})$$
$$\text{RESULT} \leftarrow \text{RESULT1} \cup \text{RESULT2}$$

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

# Example of the result of a UNION operation

**Figure 6.3**

Result of the  
UNION operation  
 $\text{RESULT} \leftarrow \text{RESULT1} \cup \text{RESULT2}$ .

**RESULT1**

Ssn
123456789
333445555
666884444
453453453

**RESULT2**

Ssn
333445555
888665555

**RESULT**

Ssn
123456789
333445555
666884444
453453453
888665555



# Relational Algebra Operations from Set Theory: INTERSECTION

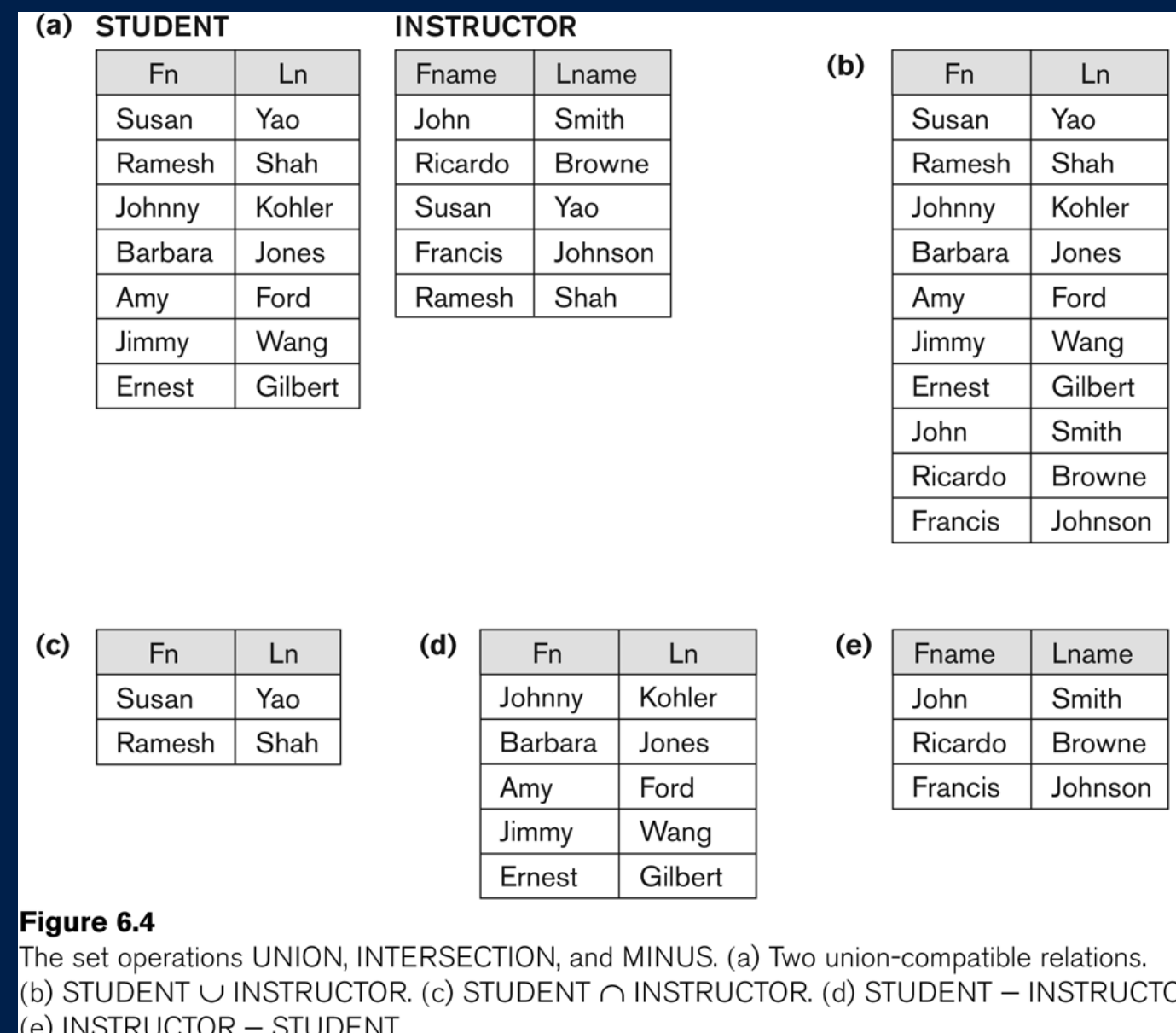
- INTERSECTION is denoted by  $\cap$
- The result of the operation  $R \cap S$ , is a relation that includes all tuples that are in both R and S
  - The attribute names in the result will be the same as the attribute names in R
- The two operand relations R and S must be “type compatible”



# Relational Algebra Operations from Set Theory: INTERSECTION

- 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



## Some properties of UNION, INTERSECT, and DIFFERENCE

- Notice that both union and intersection are *commutative* operations; that is
  - $R \cup S = S \cup R$ , 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
  - $R \cup (S \cup T) = (R \cup S) \cup T$
  - $(R \cap S) \cap T = R \cap (S \cap T)$
- The minus operation is not commutative; that is, in general
  - $R - S \neq S - R$

# 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  $R(A_1, A_2, \dots, A_n) \times S(B_1, B_2, \dots, B_m)$
  - Result is a relation  $Q$  with degree  $n + m$  attributes:
    - $Q(A_1, A_2, \dots, A_n, B_1, B_2, \dots, B_m)$ , 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  $R \times S$  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 (cont.)

- Generally, CROSS PRODUCT is not a meaningful operation
  - Can become meaningful when followed by other operations
- Example (not meaningful):
  - $FEMALE\_EMPS \leftarrow \sigma_{SEX='F'}(EMPLOYEE)$
  - $EMP\_NAMES \leftarrow \pi_{FNAME, LNAME, SSN}(FEMALE\_EMPS)$
  - $EMP\_DEPENDENTS \leftarrow EMP\_NAMES \times DEPENDENT$
- $EMP\_DEPENDENTS$  will contain every combination of  $EMP\_NAMES$  and  $DEPENDENT$ 
  - whether or not they are actually related

## 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):
  - $FEMALE\_EMPS \leftarrow \sigma_{SEX='F'}(EMPLOYEE)$
  - $EMP\_NAMES \leftarrow \pi_{FNAME, LNAME, SSN}(FEMALE\_EMPS)$
  - $EMP\_DEPENDENTS \leftarrow EMP\_NAMES \times DEPENDENT$
  - $ACTUAL\_DEPS \leftarrow \sigma_{SSN=ESSN}(EMP\_DEPENDENTS)$
  - $RESULT \leftarrow \pi_{FNAME, LNAME, DEPENDENT\_NAME}(ACTUAL\_DEPS)$
- RESULT will now contain the name of female employees and their dependents



# Example of applying CARTESIAN PRODUCT

**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



# CONCLUSION

In summary, relational algebra is a fundamental concept in the field of database management that provides a set of operations to manipulate and retrieve data from relational databases. These operations include Selection, which allows for the retrieval of specific rows based on certain conditions; Projection, which focuses on selecting specific columns from a table; Union, which combines data from two or more tables while eliminating duplicates; Intersection, which returns only the common rows between two tables; Difference, which returns the rows unique to one table compared to another; and Rename, which allows for the renaming of tables and columns for clarity and consistency. These operations are crucial for querying and manipulating data in relational databases, forming the basis for many database management systems and SQL queries.



# REFERENCES

1. Connolly. T., Begg. Carolyn. 2015. Database System: A Pratical Approach to Design, Implementation, and Management. Sixth Edition. Global Edition. Pearson.
2. Elmasri, Navathe. ...Fundamentals of Database Systems 5<sup>th</sup> Edition



Institut Informatika & Bisnis  
**DARMAJAYA**  
Yayasan Alfian Husin



**Kampus  
Merdeka**  
INDONESIA JAYA

**MERDEKA  
BELAJAR**

# THANK YOU!!

DATA SCIENCE DARMAJAYA "YOUR BEST FUTURE IN DATA"