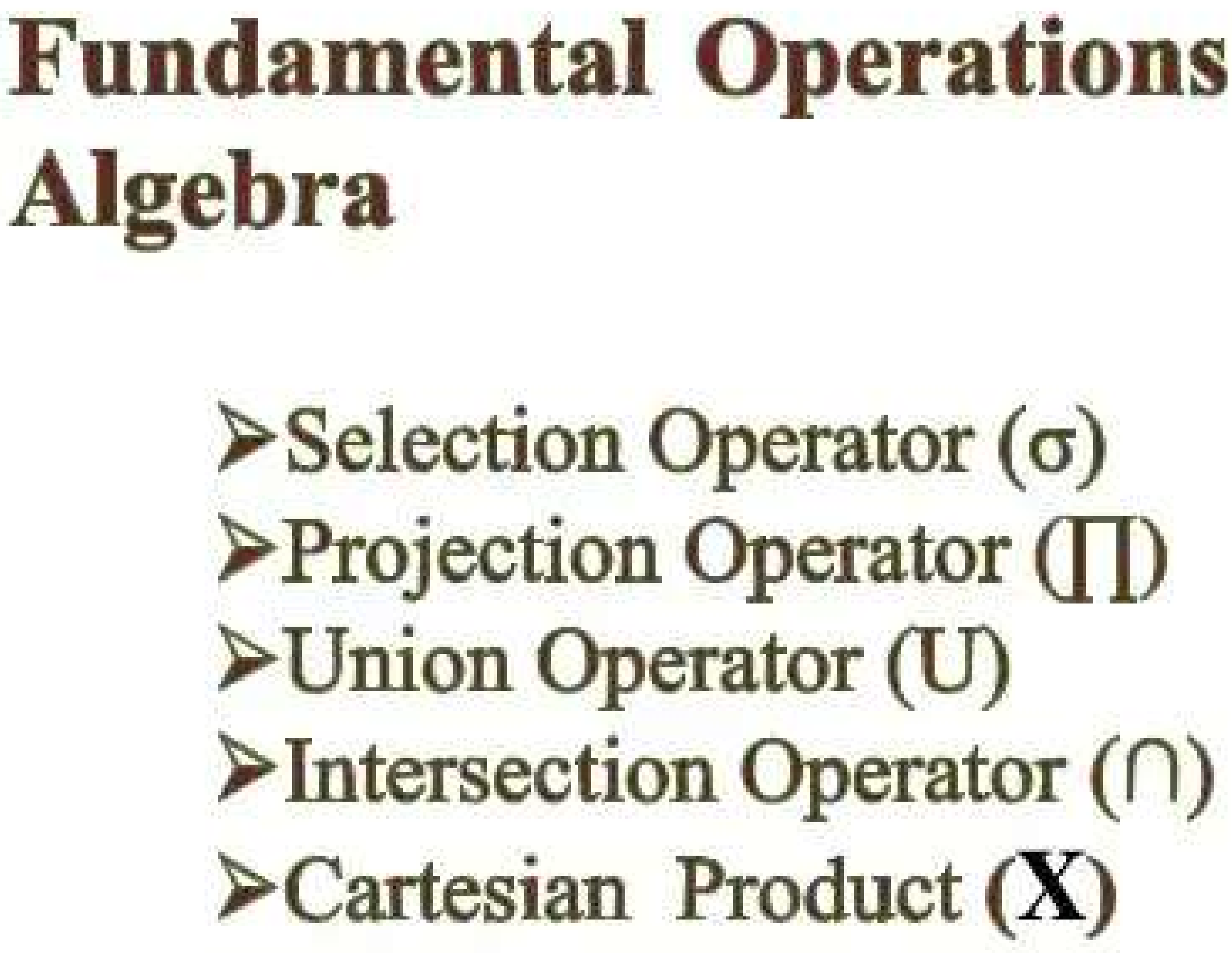
UNIT-2
Relational Operator

Relational Algebra

- A collection of operators for manipulating relations
- A relation is a set of tuples
- The result of each relational algebra expression is a relation
- For this discussion we will strictly following the definition of a set so the will be no duplicate tuples in a relation
- We will relax this constraint when we talk about relational algebra in the context of query processing



Fundamental Operations in Relational Algebra

- Selection Operator (σ)
- Projection Operator (∏)
- Union Operator (U)
- Intersection Operator (∩)
- Cartesian Product (X)
- Difference (-)

Unary Relational Operations
SELECT and PROJECT

The SELECT Operation:

- The SELECT operation is used to choose a *subset* of the tuples from a relation that satisfies a **selection condition.**

- The SELECT operation is denoted by

  σ <selection condition> (*R*)

  where the symbol σ (sigma) is used to denote the SELECT operator.

- The selection condition is a Boolean expression (condition) specified on the attributes of relation *R*.

  - σ Dno=4(EMPLOYEE)

  - σ Salary>30000(EMPLOYEE)

  Where Dno is Department No.

- <comparison op> is normally one of the operators {=, <, ≤, >, ≥, ≠}.
- Clauses can be connected by the standard Boolean operators *and*, *or*, and *not* to form a general selection condition.

Selection (σ)

eno	ename	sal	desig
IT1	ALI	500	TUTOR
BUS2	AHMED	1000	HEAD
IT2	SABA	400	CLERK
IT3	SALEH	500	TUTOR
BUS1	BADER	650	TUTOR

$\sigma_{sal > 500}$ (employee) - it will select rows having salary > 500

eno	ename	sal	desig
BUS2	AHMED	1000	HEAD
BUS1	BADER	650	TUTOR

PROJECTION

- In relational algebra, a projection is a unary operation written as $\pi_{a_1, \ldots, a_n}(R)$ where $a_1,\ldots,a_n$ is a set of attribute names.
- The result of such projection is defined as the set that is obtained when all tuples in *R* are restricted to the set $\{a_1,\ldots,a_n\}$.
- Example:

Person

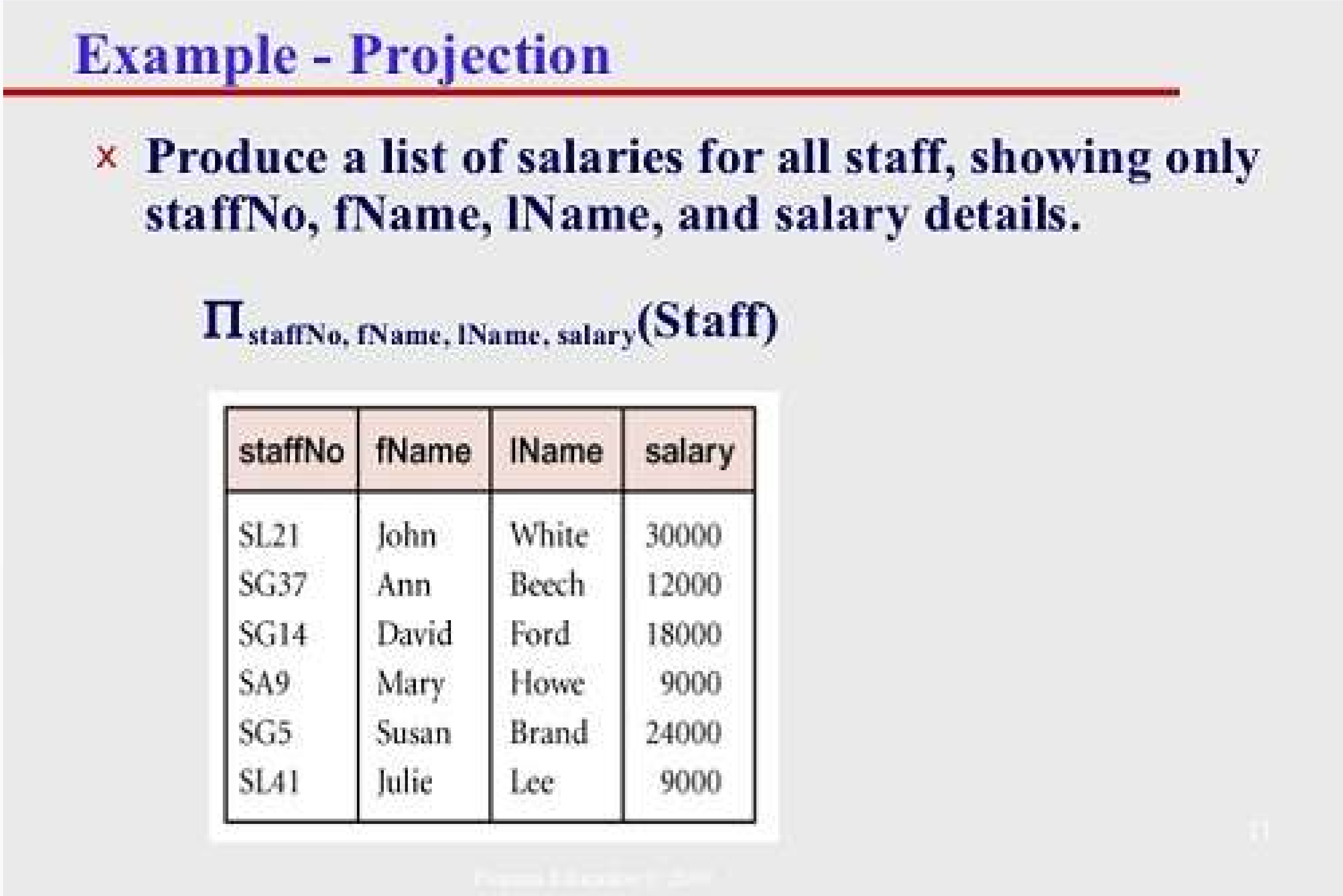
Name	Age	Weight
Harry	34	80
Sally	28	64
George	29	70
Helena	54	54
Peter	34	80

$\pi_{\text{Age, Weight}}(\text{Person})$

Age	Weight
34	80
28	64
29	70
54	54

DISTINCT

(http://en.wikipedia.org/wiki/Projection_%28relational_algebra%29)



Example - Projection

- Produce a list of salaries for all staff, showing only staffNo, fName, lName, and salary details.

$\Pi_{\text{staffNo, fName, lName, salary}}(\text{Staff})$

staffNo	fName	lName	salary
SL21	John	White	30000
SG37	Ann	Beech	12000
SG14	David	Ford	18000
SA9	Mary	Howe	9000
SG5	Susan	Brand	24000
SL41	Julie	Lee	9000

Set Based: UNION, INTERSECTION, DIFFERENCE

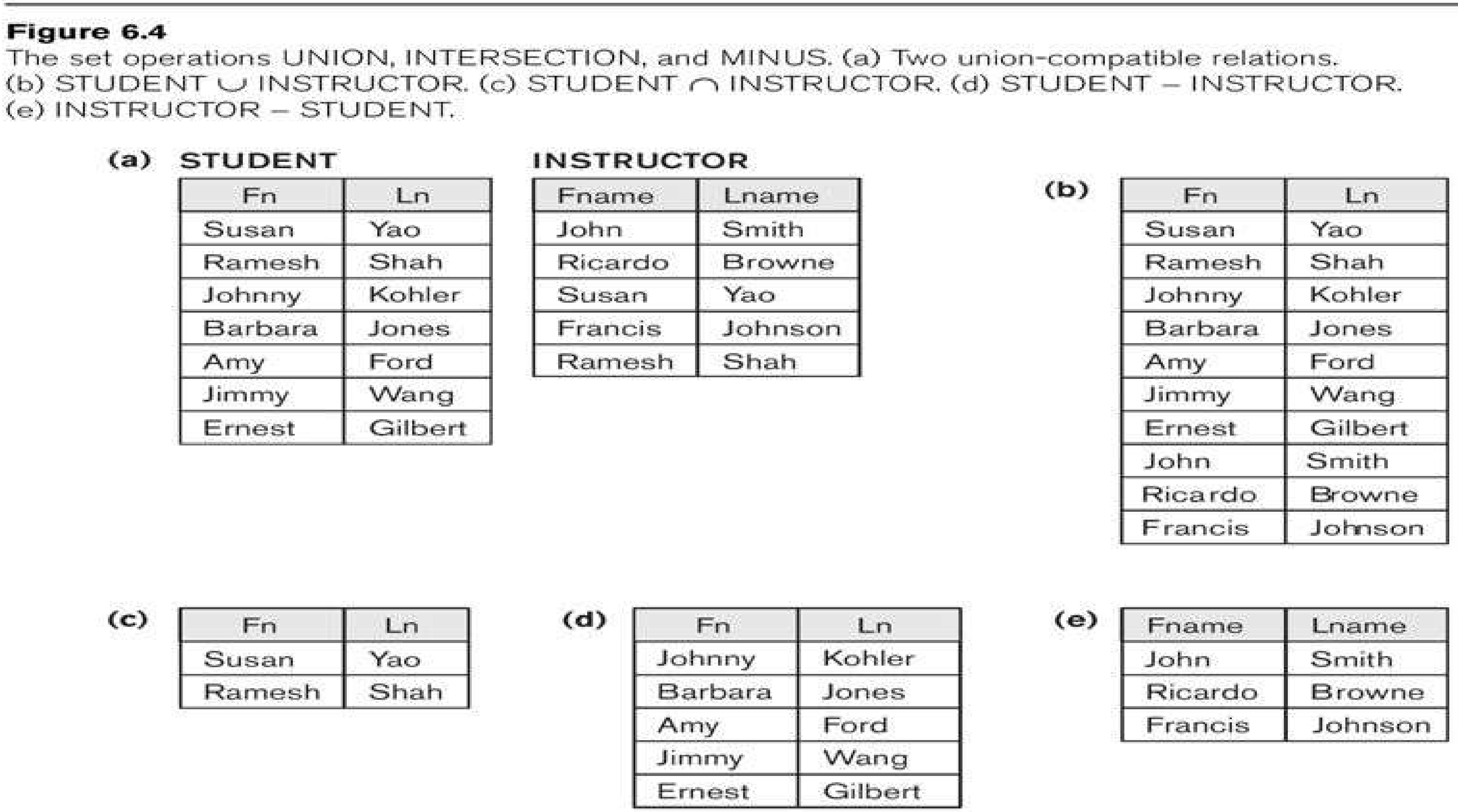


Figure 6.4
The set operations UNION, INTERSECTION, and MINUS. (a) Two union-compatible relations.
(b) STUDENT ∪ INSTRUCTOR. (c) STUDENT ∩ INSTRUCTOR. (d) STUDENT – INSTRUCTOR.
(e) INSTRUCTOR – STUDENT.

(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

SET DIFFERENCE operation

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

(d)

Fn	Ln
Johnny	Kohler
Barbara	Jones
Amy	Ford
Jimmy	Wang
Ernest	Gilbert

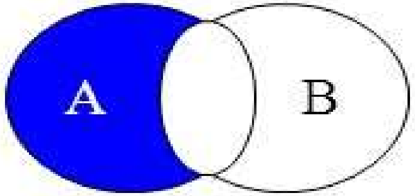
INSTRUCTOR - STUDENT

(e)

Fname	Lname
John	Smith
Ricardo	Browne
Francis	Johnson

STUDENT - INSTRUCTOR

Suppose names of people are distinct

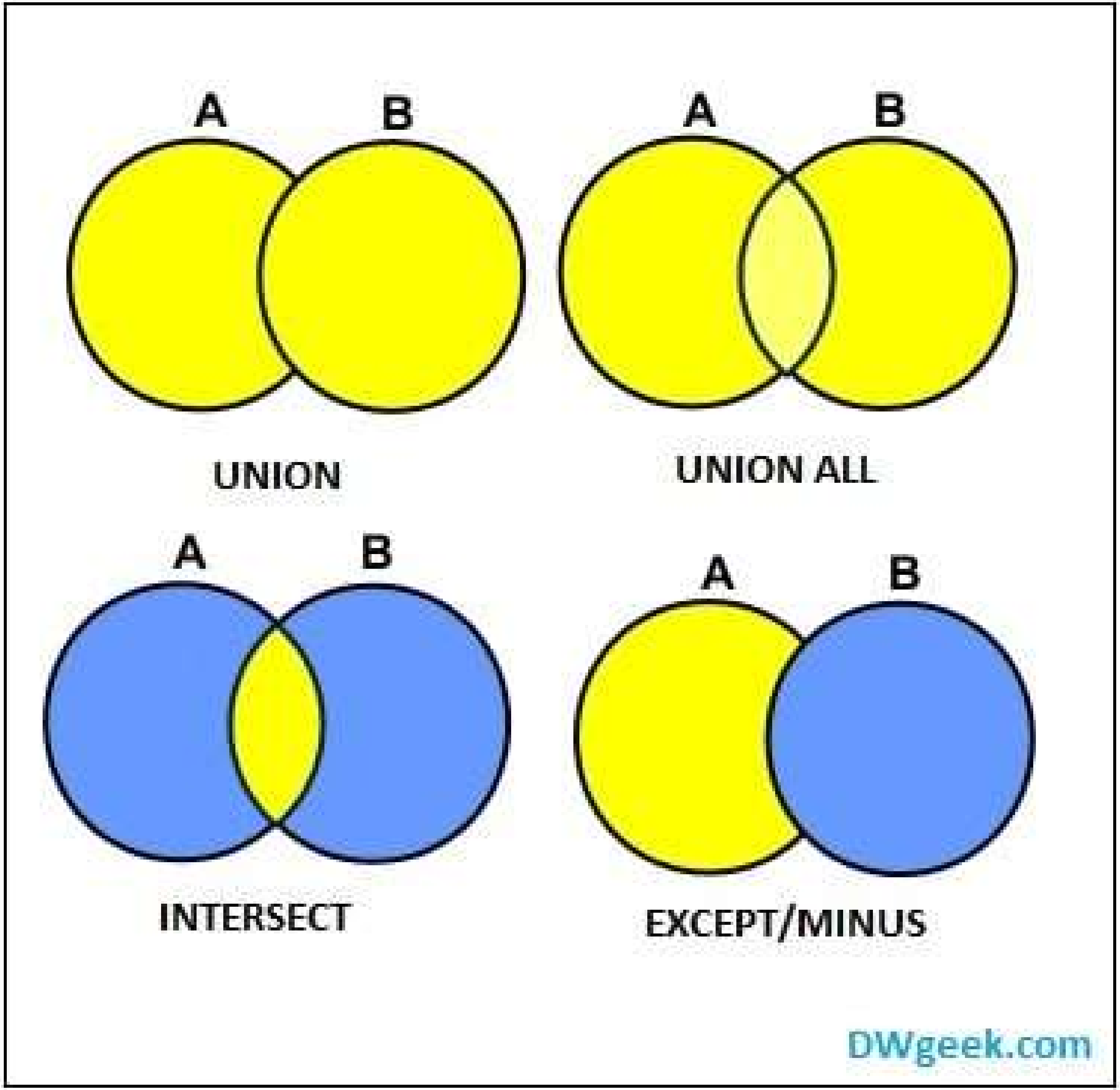


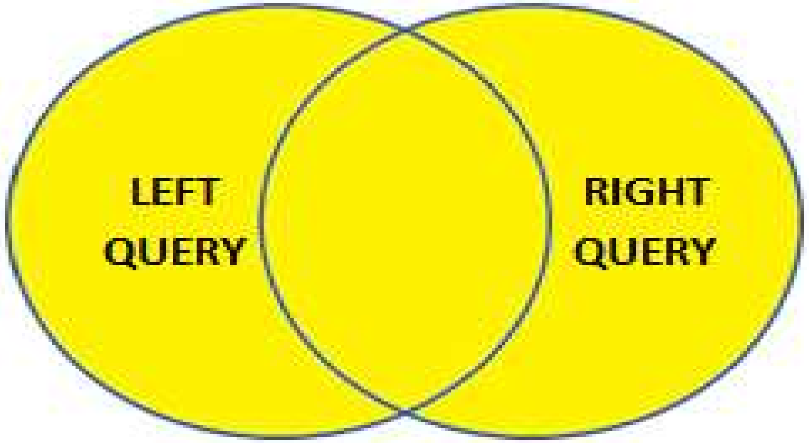
(d) RESULT=INSTRUCTOR - STUDENT

(e) RESULT=STUDENT - INSTRUCTOR

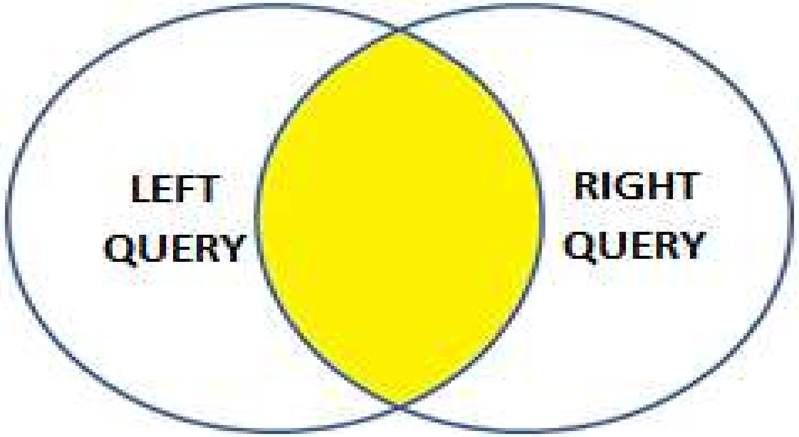
SQL for previous example Fig 6.4:

```
(SELECT Fn, Ln FROM  STUDENT)
MINUS
(SELECT Fname, Lname  FROM INSTRUCTOR);
```

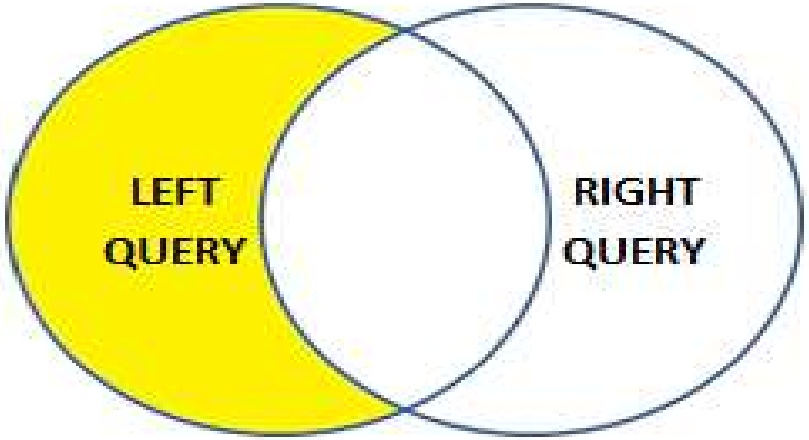

A
B
UNION
A
B
UNION ALL
A
B
INTERSECT
A
B
EXCEPT/MINUS
DWgeek.com



UNION operator returns all the unique rows from both the left and the right query. UNION ALL includes the duplicates as well



INTERSECT operator retrieves the common unique rows from both the left and the right query



EXCEPT operator returns unique rows from the left query that aren't in the right query's results

CARTESIAN PRODUCT example

R

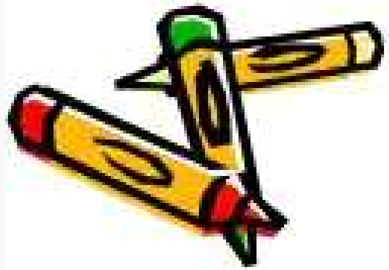
A	1
B	2
D	3
F	4
E	5

S

A	1
C	2
D	3
E	4

R CROSS S

A	1	A	1
A	1	C	2
A	1	D	3
A	1	E	4
B	2	A	1
B	2	C	2
B	2	D	3
B	2	E	4
D	3	A	1
D	3	C	2
D	3	D	3
D	3	E	4
F	4	A	1
F	4	C	2
F	4	D	3
F	4	E	4
E	5	A	1
E	5	C	2
E	5	D	3
E	5	E	4



Student

S_id	Name	Class	Age
1	Andrew	5	25
2	Angel	10	30
3	Anamika	8	35

Course

C_id	C_name
11	Foundation C
21	C++

Student X Course

S_id	Name	Class	Age	C_id	C_name
1	Andrew	5	25	11	Foundation C
1	Andrew	5	25	21	C++
2	Angel	10	30	11	Foundation C
2	Angel	10	30	21	C++
3	Anamika	8	35	11	Foundation C
3	Anamika	8	35	21	C++