



SQL language: basics

Set operators

Set operators

- The UNION operator
- The INTERSECT operator
- The EXCEPT operator



Set operators

The UNION operator

The UNION operator

➤ Set union operator

A UNION B

➤ It performs the union of the two relational expressions A and B

- relational expressions A and B may be generated by SELECT statements
- it requires schema compatibility between A and B
- removal of duplicates
 - UNION removes duplicates
 - UNION ALL does not remove duplicates

UNION: example

➤ Find the codes of products that are either red or supplied by supplier S2 (or both)

P

<u>PId</u>	PName	Color	Size	Store
P1	Jumper	Red	40	London
P2	Jeans	Green	48	Paris
P3	Blouse	Blue	48	Rome
P4	Blouse	Red	44	London
P5	Skirt	Blue	40	Paris
P6	Shorts	Red	42	London

SP

<u>SId</u>	<u>PId</u>	Qty
S1	P1	300
S1	P2	200
S1	P3	400
S1	P4	200
S1	P5	100
S1	P6	100
S2	P1	300
S2	P2	400
S3	P2	200
S4	P3	200
S4	P4	300
S4	P5	400

UNION: example

➤ Find *the codes of products that are* either *red* or supplied by supplier S2 (or both)

```
SELECT PId  
FROM P  
WHERE Color='Red'
```

P

<u>PId</u>	PName	Color	Size	Store
P1	Jumper	Red	40	London
P2	Jeans	Green	48	Paris
P3	Blouse	Blue	48	Rome
P4	Blouse	Red	44	London
P5	Skirt	Blue	40	Paris
P6	Shorts	Red	42	London



PId
P1
P6

UNION: example

- Find the *codes of the products that are* either red or *supplied by supplier S2* (or both)

SP

<u>SId</u>	<u>PId</u>	Qty
S1	P1	300
S1	P2	200
S1	P3	400
S1	P4	200
S1	P5	100
S1	P6	100
S2	P1	300
S2	P2	400
S3	P2	200
S4	P3	200
S4	P4	300
S4	P5	400

```
SELECT PId  
FROM SP  
WHERE SId='S2'
```



PId
P1
P2

UNION: example

- Find the codes of products that are either red or supplied by supplier S2 (or both)

```
SELECT PId
FROM P
WHERE Color='Red'
UNION
SELECT PId
FROM SP
WHERE SId='S2';
```

PId
P1
P6

PId
P1
P2



R

PId
P1
P2
P6

UNION: example

- Find the codes of products that are either red or supplied by supplier S2 (or both)

```
SELECT PId
FROM P
WHERE Color='Red'
UNION
SELECT PId
FROM SP
WHERE SId='S2';
```

R

PId
P1
P2
P6

Duplicate removal

UNION: example

➤ Find the codes of products that are either red or supplied by supplier S2 (or both)

*Schema
compatibility*

```
SELECT PId
FROM P
WHERE Color='Red'
UNION
SELECT PId
FROM SP
WHERE SId='S2';
```

UNION ALL: example

- Find the codes of products that are either red or supplied by supplier S2 (or both)

```
SELECT PId
FROM P
WHERE Color='Red'
UNION ALL
SELECT PId
FROM SP
WHERE SId='S2';
```

PId
P1
P6

PId
P1
P2



R

PId
P1
P1
P2
P6



Set operators

The INTERSECT operator

The INTERSECT operator

➤ Set intersection operator

$A \text{ INTERSECT } B$

- It performs the intersection of the two relational expressions A and B
- relational expressions A and B may be generated by SELECT statements
 - it requires schema compatibility between A and B

INTERSECT: example

➤ Find the cities where both one or more suppliers and one or more stores are based

P

<u>PId</u>	PName	Color	Size	Store
P1	Jumper	Red	40	London
P2	Jeans	Green	48	Paris
P3	Blouse	Blue	48	Rome
P4	Blouse	Red	44	London
P5	Skirt	Blue	40	Paris
P6	Shorts	Red	42	London

S

<u>SId</u>	SName	#Employees	City
S1	Smith	20	London
S2	Jones	10	Paris
S3	Blake	30	Paris
S4	Clark	20	London
S5	Adams	30	Athens

INTERSECT: example

- Find *the cities where* both *one or more suppliers* and one or more stores *are based*

```
SELECT City  
FROM S
```

S

<u>SId</u>	SName	#Employees	City
S1	Smith	20	London
S2	Jones	10	Paris
S3	Blake	30	Paris
S4	Clark	20	London
S5	Adams	30	Athens



City
London
Paris
Paris
London
Athens

INTERSECT: example

- Find *the cities where* both one or more suppliers and *one or more stores are based*

```
SELECT Store  
FROM P
```

P

<u>PId</u>	PName	Color	Size	Store
P1	Jumper	Red	40	London
P2	Jeans	Green	48	Paris
P3	Blouse	Blue	48	Rome
P4	Blouse	Red	44	London
P5	Skirt	Blue	40	Paris
P6	Shorts	Red	42	London



Store
London
Paris
Rome
London
Paris
London

INTERSECT: example

- Find the cities where both one or more suppliers and one or more stores are based

```
SELECT City
FROM S
INTERSECT
SELECT Store
FROM P;
```

City
London
Paris
Paris
London
Athens

Store
London
Paris
Rome
London
Paris
London



R

London
Paris

Equivalence with other operators

- The intersection operation may also be performed by means of
- a join
 - the IN operator

Equivalence with join

- The **FROM** clause contains the relations involved in the intersection
- The **WHERE** clause contains join conditions between the attributes listed in the **SELECT** clauses of relational expressions A and B

Equivalence with join: example

- Find the cities where both one or more suppliers and one or more stores are based

```
SELECT City  
FROM S, P  
WHERE S.City=P.Store;
```


Equivalence with the IN operator

- One of the two relational expressions is turned into a nested query using operator **IN**
- The attributes in the outer **SELECT** clause, together by a tuple constructor, make up the left-hand side of the **IN** operator

Equivalence with IN: example

- Find the cities where both one or more suppliers and one or more stores are based

```
SELECT Store
FROM P
WHERE Store IN (SELECT City
                FROM S);
```



Set operators

The EXCEPT operator

The EXCEPT operator

➤ Set difference operator

$A \text{ EXCEPT } B$

- It subtracts relational expression B from relational expression A
- it requires schema compatibility between A and B

EXCEPT: example

➤ Find the cities where one or more suppliers, but no stores are based

P

<u>PId</u>	PName	Color	Size	Store
P1	Jumper	Red	40	London
P2	Jeans	Green	48	Paris
P3	Blouse	Blue	48	Rome
P4	Blouse	Red	44	London
P5	Skirt	Blue	40	Paris
P6	Shorts	Red	42	London

S

<u>SId</u>	SName	#Employees	City
S1	Smith	20	London
S2	Jones	10	Paris
S3	Blake	30	Paris
S4	Clark	20	London
S5	Adams	30	Athens

EXCEPT: example

➤ Find *the cities where one or more suppliers*, but no stores *are based*

```
SELECT City  
FROM S
```

S

<u>SId</u>	SName	#Employees	City
S1	Smith	20	London
S2	Jones	10	Paris
S3	Blake	30	Paris
S4	Clark	20	London
S5	Adams	30	Athens



City
London
Paris
Paris
London
Athens

EXCEPT: example

➤ Find *the cities where* one or more suppliers, but no *stores are based*

```
SELECT Store  
FROM P
```

P

<u>PId</u>	PName	Color	Size	Store
P1	Jumper	Red	40	London
P2	Jeans	Green	48	Paris
P3	Blouse	Blue	48	Rome
P4	Blouse	Red	44	London
P5	Skirt	Blue	40	Paris
P6	Shorts	Red	42	London



Store
London
Paris
Rome
London
Paris
London

EXCEPT: example

- Find the cities where one or more suppliers, but no stores are based

```
SELECT City
FROM S
EXCEPT
SELECT Store
FROM P;
```

City
London
Paris
Paris
London
Athens

Store
London
Paris
Rome
London
Paris
London



R

Athens

Equivalence with the NOT IN operator

- The EXCEPT operation may also be performed by means of the NOT IN operator
- relational expression B is nested within the NOT IN operator
 - the attributes in the SELECT clause of relational expression A, together by a tuple constructor, make up the left-hand side of the NOT IN operator

Equivalence with the NOT IN operator: example

- Find the cities where one or more suppliers, but no stores are based

```
SELECT City
FROM S
WHERE City NOT IN (SELECT Store
                   FROM P);
```