

» Гл. ас. д-р Георги Чолаков

» Базы от данни

**Релационна алгебра** >

# Въведение

Третата и последна част на релационния модел (манипулативната част) се състои от множество от оператори, които образуват така наречената релационна алгебра. Тя дефинира теоретичен модел за манипулиране на данните чрез релационни оператори с цел извличане на полезна информация.

Codd дефинира 8 операции:

- > SELECT (RESTRICT)
- > PROJECT
- > UNION
- > INTERSECT
- > DIFFERENCE
- > PRODUCT
- > JOIN
- > DIVIDE



# Затвореност

Релационната алгебра притежава свойството **затвореност** - резултатът от всяка релационна операция е отново релация (**релационна затвореност**).

- > Така всеки изход от една операция може да бъде вход на друга операция;
- > Ще е възможно да създаваме вградени изрази - т.е. операндите могат да бъдат представени посредством изрази;



# SELECT (RESTRICT)


Приложен върху една таблица (унарн) връща отново таблица, която съдържа всички колони от оригиналната и само записите, отговарящи на специфицираните условия (предикати).

Т.е. операторът SELECT връща хоризонтално подмножество от таблицата, т.е. редовете, които имат стойности на атрибутите си, отговарящи на зададените условия.



# SELECT – пример с таблицата PRODUCTS

PRODUCT_ID	NAME	PRICE	DESCR
1733	Външен хард диск ADATA Durable HD330	109.00	1TB, Shock Sensor, 2.5", USB 3.1, Черен
1734	Таблет Xiaomi mi Pad 4	206.00	Таблет Xiaomi mi Pad 4 ,3GB RAM 32 GB ROM ,black ...
1737	Таблет Samsung Tab3 T113	198.00	Таблет Samsung Tab3 T113 Lite Value Edition с проце...
1739	USB памет Kingston DataTraveler 100 G3	9.00	6GB, USB 3.0
1742	Безжични слушалки I7s PLUS	15.00	Безжични слушалки I7s PLUS с кутия за зареждане ...
1743	Лаптоп 2 in 1 Lenovo YOGA Glass C930-13IKB	4800.00	Лаптоп 2 in 1 Lenovo YOGA Glass C930-13IKB with pro...
1745	Таблет Lenovo Tab Yoga 3 YT3-X50M	349.00	Таблет Lenovo Tab Yoga 3 YT3-X50M, 10.1', Quad-Cor...
1750	Монитор LED TN AOC 24.5	299.00	Монитор LED TN AOC 24.5", Full HD, FreeSync 144Hz...
1763	Процесор AMD FX-8350	147.00	Процесор AMD FX-8350, 4.0GHz, 16MB, 125W, AM3+.
1768	Лаптоп 2 in 1 HP ENVY x360 13-ag0001nn	1845.00	Лаптоп 2 in 1 HP ENVY x360 13-ag0001nn с процесор...

**SELECT \* FROM PRODUCTS WHERE PRODUCT\_ID = 1745**

PRODUCT_ID	NAME	PRICE	DESCR
1745	Таблет Lenovo Tab Yoga 3 YT3-X50M	349.00	Таблет Lenovo Tab Yoga 3 YT3-X50M...

**SELECT \* FROM PRODUCTS WHERE PRICE > 300**

PRODUCT_ID	NAME	PRICE	DESCR
1743	Лаптоп 2 in 1 Lenovo YOGA Glass C930-13IKB	4800.00	Лаптоп 2 in 1 Lenovo YOGA Glas...
1745	Таблет Lenovo Tab Yoga 3 YT3-X50M	349.00	Таблет Lenovo Tab Yoga 3 YT3-...
1768	Лаптоп 2 in 1 HP ENVY x360 13-ag0001nn	1845.00	Лаптоп 2 in 1 HP ENVY x360 13-a...



# PROJECT


Приложен върху една таблица (унарен) връща отново таблица, която съдържа всички стойности от редовете, но само за избраните колони от оригиналната.

Т.е. операторът PROJECT връща вертикално подмножество от таблицата, т.е. всички редове, но само с избраните колони.



# PROJECT – пример с таблицата PRODUCTS

PRODUCT_ID	NAME	PRICE	DESCR
1733	Външен хард диск ADATA Durable HD330	109.00	1TB, Shock Sensor, 2.5", USB 3.1, Черен
1734	Таблет Xiaomi mi Pad 4	206.00	Таблет Xiaomi mi Pad 4 ,3GB RAM 32 GB ROM ,black ...
1737	Таблет Samsung Tab3 T113	198.00	Таблет Samsung Tab3 T113 Lite Value Edition с проце...
1739	USB памет Kingston DataTraveler 100 G3	9.00	6GB, USB 3.0
1742	Безжични слушалки I7s PLUS	15.00	Безжични слушалки I7s PLUS с кутия за зареждане ...
1743	Лаптоп 2 in 1 Lenovo YOGA Glass C930-13IKB	4800.00	Лаптоп 2 in 1 Lenovo YOGA Glass C930-13IKB with pro...
1745	Таблет Lenovo Tab Yoga 3 YT3-X50M	349.00	Таблет Lenovo Tab Yoga 3 YT3-X50M, 10.1', Quad-Cor...
1750	Монитор LED TN AOC 24.5	299.00	Монитор LED TN AOC 24.5", Full HD, FreeSync 144Hz...
1763	Процесор AMD FX-8350	147.00	Процесор AMD FX-8350, 4.0GHz, 16MB, 125W, AM3+.
1768	Лаптоп 2 in 1 HP ENVY x360 13-ag0001nn	1845.00	Лаптоп 2 in 1 HP ENVY x360 13-ag0001nn с процесор...

**SELECT** name **FROM** PRODUCTS

NAME
USB памет Kingston DataTraveler 100 G3
Безжични слушалки I7s PLUS
Външен хард диск ADATA Durable HD330
Лаптоп 2 in 1 HP ENVY x360 13-ag0001nn
Лаптоп 2 in 1 Lenovo YOGA Glass C930-13I...
Монитор LED TN AOC 24.5
Процесор AMD FX-8350
Таблет Lenovo Tab Yoga 3 YT3-X50M
Таблет Samsung Tab3 T113
Таблет Xiaomi mi Pad 4

**SELECT** price, product\_id **FROM** PRODUCTS

PRICE	PRODUCT_ID
9.00	1739
15.00	1742
109.00	1733
147.00	1763
198.00	1737
206.00	1734
299.00	1750
349.00	1745
1845.00	1768
4800.00	1743



# PROJECT – примери

SQL SELECT lname, fname, gender, country\_id FROM CUSTOMERS

SQL Output Statistics

```
SELECT lname, fname, gender, country_id
FROM CUSTOMERS
```

	LNNAME	FNAME	GENDER	COUNTRY_ID
1	Генев	Калоян	M	BG
2	Петров	Христо	M	BG
3	Тодорова	Марина	M	BG
4	Георгиев	Петър	M	BG
5	Георгиев	Марин	M	BG
6	Христов	Мартин	M	BG
7	Костов	Момчил	M	BG
8	Господинов	Младен	M	BG
9	Кирилов	Калоян	M	BG
10	Костов	Кирил	M	BG
11	Петрова	Ивайла	F	BG

mydb@ORCL-UNI-OUT [19:06]

Solution1

SQLQuery2.sql - 19...ADE\_GUIDE (sa (53))\*

```
SELECT FNAME, LNAME, SALARY, HIRE_DATE
FROM EMPLOYEES
```

177 %

Results Messages

	FNAME	LNAME	SALARY	HIRE_DATE
1	Иван	Иванов	27000....	2011-06-17
2	Петя	Василева	18000....	2015-09-21
3	Димитър	Господин...	19000....	2012-01-13
4	Александър	Петров	12000....	2010-01-03
5	Боряна	Иванова	12000....	2010-05-21
6	Димо	Тодоров	7800.00	2005-06-25
7	Васил	Петров	8800.00	2008-02-05
8	Диана	Лазарова	7200.00	2009-02-07
9	Нина	Георгиева	12000....	2015-08-17

Query executed successfully. 194.141.98.98 (12.0 RTM) sa (53) PCTRADE\_GUIDE 00:00:00 101 rows



# Съвместимост на типове

Нека разгледаме обединението:

- » В математиката обединението на две множества е множеството на всички елементи, принадлежащи към едно от двете оригинални множества;
- » Понеже релацията е множество от записи – възможно е да конструираме обединението на две релации;
- » Но възможно ли е да възникнат проблеми?



# Обединение

LECTURERS

FName	LName	Title	Faculty
Стоян	Колев	Проф.	ФМИ
Петър	Иванов	Доц.	Право
Венета	Георгиева	Ст.н.с.	Икономика
Спас	Петров	Доц.	ФМИ

SUBJECTS

Subject	Horarium	Hall
Бази от данни	30	422
Изкуствен интелект	20	424

Резултат:

?	?	?	?
Стоян	Колев	Проф.	ФМИ
Петър	Иванов	Доц.	Право
Изкуствен интелект	20	424	
Венета	Георгиева	Ст.н.с.	Икономика
Спас	Петров	Доц.	ФМИ
Изкуствен интелект	20	424	



- » Въпреки, че резултатът е множество от (разнотипни) редове, той не е релация!
- » Релацията не може да съдържа смесени типове записи!
- » Искаме резултатът да е релация, за да запазим свойството затвореност;
- » Следователно, обединението в релационната алгебра не е идентично с математическото обединение;
- » По-скоро то е специален случай, при който изискваме двете входни релации да бъдат от **един и същ тип**.



# Съвместимост на типове

- » Двете релации да имат идентични заглавни части - т.е.:
  - > Да имат еднакви множества от имена на атрибутите;
  - > Кореспондиращите атрибути да са дефинирани върху еднакви области.
- » Съвместимост по типове се изисква за операторите:
  - > Обединение (UNION)
  - > Сечение (INTERSECT)
  - > Разлика (DIFFERENCE)



# UNION


Създава релация, която се състои от всички записи, които се появяват във всяка една или и в двете релации.

- » Обединението на две релации A и B със съвместими типове поражда трета релация със:
  - > заглавна част като на A и B (трябва да бъдат съвместими);
  - > тяло – множеството от всички записи, принадлежащи на A, B или на двете, като дубликатите се елиминират.





# UNION – пример

PRODUCT_ID	NAME	PRICE
1734	Таблет Xiaomi mi Pad 4	206.00
1749	Монитор Acer Gaming LED IPS 23.8	237.00
3087	Монитор Gaming LED IPS LG 25	224.00

UNION

PRODUCT_ID	NAME	PRICE
1734	Таблет Xiaomi mi Pad 4	206.00
1749	Монитор Acer Gaming LED IPS 23.8	237.00
2373	Таблет HUAWEI MediaPad T3	256.00
2879	Дънна платка GIGABYTE Z370P D3	256.00
3087	Монитор Gaming LED IPS LG 25	224.00

UNION ALL

PRODUCT_ID	NAME	PRICE
1749	Монитор Acer Gaming LED IPS 23.8	237.00
2373	Таблет HUAWEI MediaPad T3	256.00
2879	Дънна платка GIGABYTE Z370P D3	256.00
3087	Монитор Gaming LED IPS LG 25	224.00

PRODUCT_ID	NAME	PRICE
1734	Таблет Xiaomi mi Pad 4	206.00
1749	Монитор Acer Gaming LED IPS 23.8	237.00
1749	Монитор Acer Gaming LED IPS 23.8	237.00
2373	Таблет HUAWEI MediaPad T3	256.00
2879	Дънна платка GIGABYTE Z370P D3	256.00
3087	Монитор Gaming LED IPS LG 25	224.00
3087	Монитор Gaming LED IPS LG 25	224.00



# UNION – пример 2

## UNION

Solution1 - SQLQuery1.sql - 194.1...

SQLQuery1.sql - 19...CTRADE\_1 (sa (52))\*

```
SELECT fname, lname
FROM employees
UNION
SELECT fname, lname
FROM customers
ORDER BY fname, lname
```

110 %

Results Messages

	fname	lname
21	Боряна	Иванова
22	Васил	Петров
23	Васил	Танев
24	Виолета	Донева
25	Георги	Гелов
26	Георги	Георгиев
27	Георги	Димитров
28	Георги	Стефанов
29	Георги	Христов
30	Гергана	Вълкова
31	Гергана	Костова
32	Гергана	Петрова

3 (12.0 RTM) | sa (52) | PCTRADE\_1 | 00:00:00 | 211 rows

## UNION ALL

Solution1 - SQLQuery1.sql - 194.1...

SQLQuery1.sql - 19...CTRADE\_1 (sa (52))\*

```
SELECT fname, lname
FROM employees
UNION ALL
SELECT fname, lname
FROM customers
ORDER BY fname, lname
```

110 %

Results Messages

	fname	lname
21	Боряна	Димитрова
22	Боряна	Иванова
23	Васил	Петров
24	Васил	Танев
25	Виолета	Донева
26	Георги	Гелов
27	Георги	Георгиев
28	Георги	Георгиев
29	Георги	Димитров
30	Георги	Стефанов
31	Георги	Стефанов
32	Георги	Христов

3 (12.0 RTM) | sa (52) | PCTRADE\_1 | 00:00:00 | 218 rows



# INTERSECT


Създава нова релация, състояща се от всички  $n$ -торки, които се появяват в двете релации едновременно.

- » Сечението на две релации  $A$  и  $B$  със съвместими типове поражда трета релация със:
  - > заглавна част като на  $A$  и  $B$  (трябва да бъдат съвместими);
  - > тяло – множеството от всички записи, принадлежащи едновременно на  $A$  и  $B$ , т.е. общите за двете релации  $n$ -торки (записи).



# INTERSECT - пример

PRODUCT_ID	NAME	PRICE
1734	Таблет Xiaomi mi Pad 4	206.00
1749	Монитор Acer Gaming LED IPS 23.8	237.00
3087	Монитор Gaming LED IPS LG 25	224.00

INTERSECT



PRODUCT_ID	NAME	PRICE
1749	Монитор Acer Gaming LED IPS 23.8	237.00
3087	Монитор Gaming LED IPS LG 25	224.00

PRODUCT_ID	NAME	PRICE
1749	Монитор Acer Gaming LED IPS 23.8	237.00
2373	Таблет HUAWEI MediaPad T3	256.00
2879	Дънна платка GIGABYTE Z370P D3	256.00
3087	Монитор Gaming LED IPS LG 25	224.00



# INTERSECT – пример

Solution1 - SQLQuery1.sql - 194.141.98.9...

SQLQuery1.sql - 19...CTRADE\_1 (sa (52))\*

```
SELECT fname, lname  
FROM employees  
ORDER BY fname, lname
```

Results

	fname	lname
1	Александра	Бонева
2	Александра	Видева
3	Александър	Емилов
4	Александър	Кръстев
5	Александър	Михов
6	Александър	Петров
7	Алина	Христова
8	Антон	Канев
9	Антон	Петров
10	Антон	Филипов
11	Борис	Григоров

11.98.98 (12.0 RTM) | sa (52) | PCTRADE\_1

INTERSECT



Solution1 - SQLQuery1.sql - 194.141.98.9...

SQLQuery1.sql - 19...CTRADE\_1 (sa (52))\*

```
SELECT fname, lname  
FROM customers  
ORDER BY fname, lname
```

Results

	fname	lname
1	Александър	Стафанов
2	Ирина	Костадинова
3	Рик	Ромеро
4	Elizabeth	Brown
5	Александър	Миланов
6	Ангел	Ангел Николов
7	Ангел	Илиев
8	Антон	Петров
9	Блейк	Зайнър
10	Борис	Григоров
11	Ванеса	Тейлор

11.98.98 (12.0 RTM) | sa (52) | PCTRADE\_1 | 00:00:00 | 118 rows

Solution1 - SQLQuery1.sql - 194.141.98.9...

SQLQuery1.sql - 19...CTRADE\_1 (sa (52))\*

```
SELECT fname, lname  
FROM employees  
INTERSECT  
SELECT fname, lname  
FROM customers  
ORDER BY fname, lname
```

Results

	fname	lname
1	Антон	Петров
2	Георги	Георгиев
3	Димитър	Петров
4	Иван	Иванов

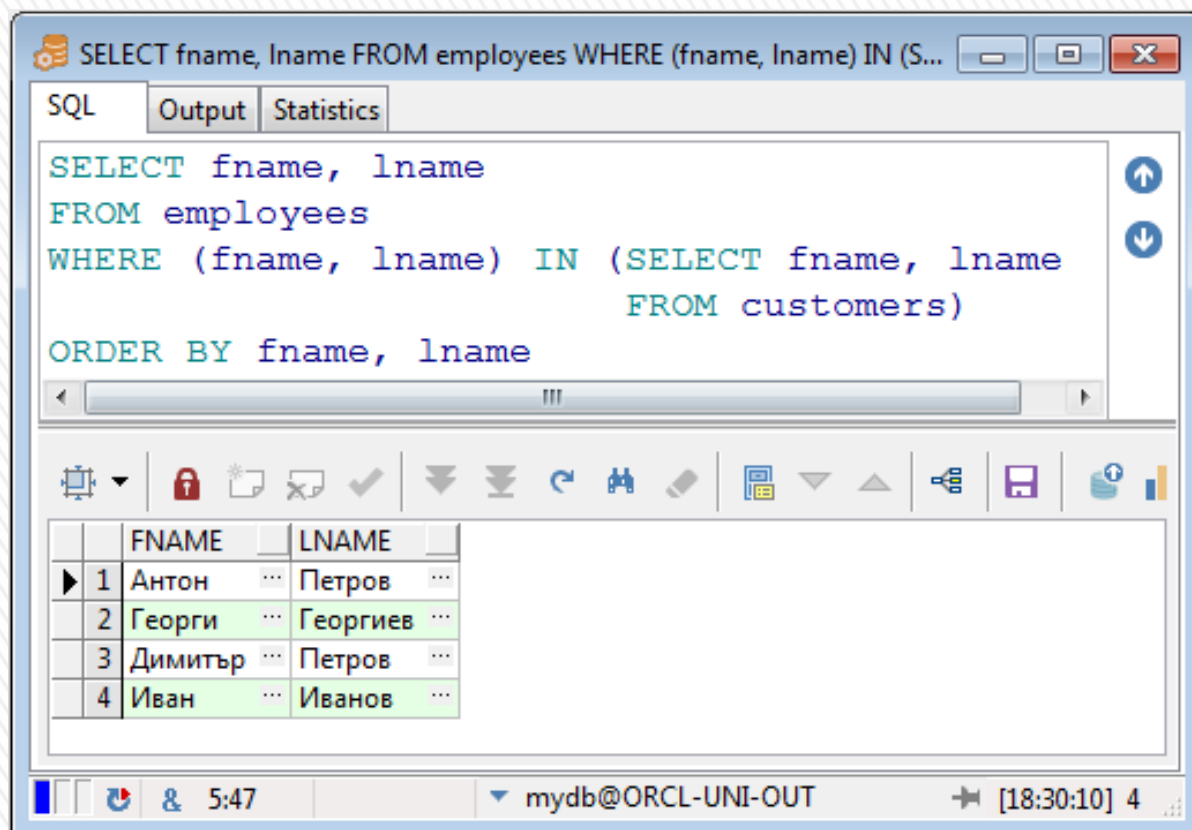
11.98.98 (12.0 RTM) | sa (52) | PCTRADE\_1 | 00:00:00 | 4 rows





# INTERSECT – пример с IN

## PL/SQL



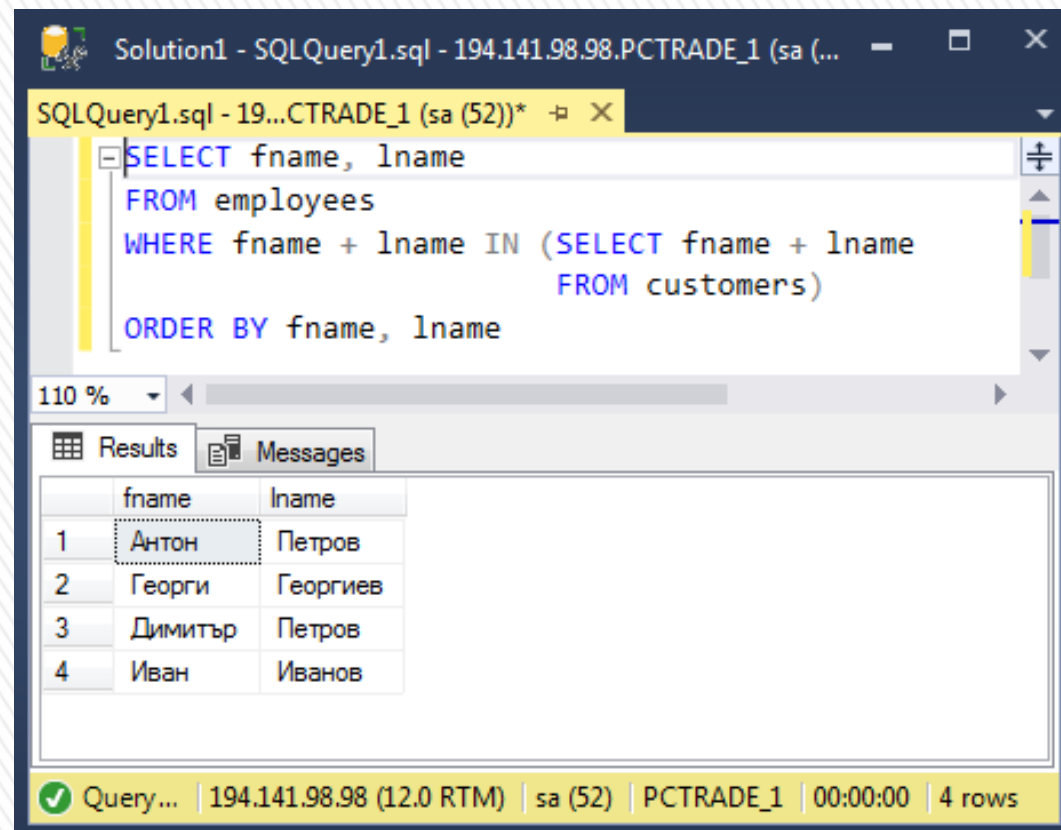
SQL

```
SELECT fname, lname  
FROM employees  
WHERE (fname, lname) IN (SELECT fname, lname  
                        FROM customers)  
  
ORDER BY fname, lname
```

	FNAME	LNAME
1	Антон	Петров
2	Георги	Георгиев
3	Димитър	Петров
4	Иван	Иванов

mydb@ORCL-UNI-OUT [18:30:10] 4

## T-SQL



Solution1 - SQLQuery1.sql - 194.141.98.98.PCTRADE\_1 (sa (...)

```
SELECT fname, lname  
FROM employees  
WHERE fname + lname IN (SELECT fname + lname  
                        FROM customers)  
  
ORDER BY fname, lname
```

110 %

Results Messages

	fname	lname
1	Антон	Петров
2	Георги	Георгиев
3	Димитър	Петров
4	Иван	Иванов

Query... 194.141.98.98 (12.0 RTM) | sa (52) | PCTRADE\_1 | 00:00:00 | 4 rows

# DIFFERENCE


Създава нова релация, състояща се от всички записи, които се появяват в първата, но не и във втората релация.

- » Разликата на две релации А и В със съвместими типове поражда трета релация със:
  - > заглавна част като на А и В (трябва да бъдат съвместими);
  - > тяло – множеството от всички записи, принадлежащи на А и не принадлежащи на В.



# DIFFERENCE – пример

A

PRODUCT_ID	NAME	PRICE
1734	Таблет Xiaomi mi Pad 4	206.00
1749	Монитор Acer Gaming LED IPS 23.8	237.00
3087	Монитор Gaming LED IPS LG 25	224.00

**DIFFERENCE**

B

PRODUCT_ID	NAME	PRICE
1749	Монитор Acer Gaming LED IPS 23.8	237.00
2373	Таблет HUAWEI MediaPad T3	256.00
2879	Дънна платка GIGABYTE Z370P D3	256.00
3087	Монитор Gaming LED IPS LG 25	224.00

PRODUCT_ID	NAME	PRICE
1734	Таблет Xiaomi mi Pad 4	206.00

A - B

B - A

PRODUCT_ID	NAME	PRICE
2373	Таблет HUAWEI MediaPad T3	256.00
2879	Дънна платка GIGABYTE Z370P D3	256.00



# DIFFERENCE – пример

A

PRODUCT_ID	NAME	PRICE
1734	Таблет Xiaomi mi Pad 4	206.00
1749	Монитор Acer Gaming LED IPS 23.8	237.00
3087	Монитор Gaming LED IPS LG 25	224.00

DIFFERENCE

B

PRODUCT_ID	NAME	PRICE
1749	Монитор Acer Gaming LED IPS 23.8	237.00
2373	Таблет HUAWEI MediaPad T3	256.00
2879	Дънна платка GIGABYTE Z370P D3	256.00
3087	Монитор Gaming LED IPS LG 25	224.00

A - B

B - A

```
SELECT PRODUCT_ID, NAME, PRICE FROM PRODUCTS WHERE PRICE BETWEEN 200 AND 250
MINUS
SELECT PRODUCT_ID, NAME, PRICE FROM PRODUCTS WHERE PRICE BETWEEN 220 AND 260
```

```
SELECT PRODUCT_ID, NAME, PRICE FROM PRODUCTS WHERE PRICE BETWEEN 220 AND 260
EXCEPT
SELECT PRODUCT_ID, NAME, PRICE FROM PRODUCTS WHERE PRICE BETWEEN 200 AND 250
```

	PRODUCT_ID	NAME	PRICE
1	2373	Таблет HUAWEI MediaPad T3	256.00
2	2879	Дънна платка GIGABYTE Z370P D3	256.00

# DIFFERENCE – пример 2

A

PRODUCT_ID	NAME	PRICE
1734	Таблет Xiaomi mi Pad 4	206.00
1749	Монитор Acer Gaming LED IPS 23.8	237.00
3087	Монитор Gaming LED IPS LG 25	224.00

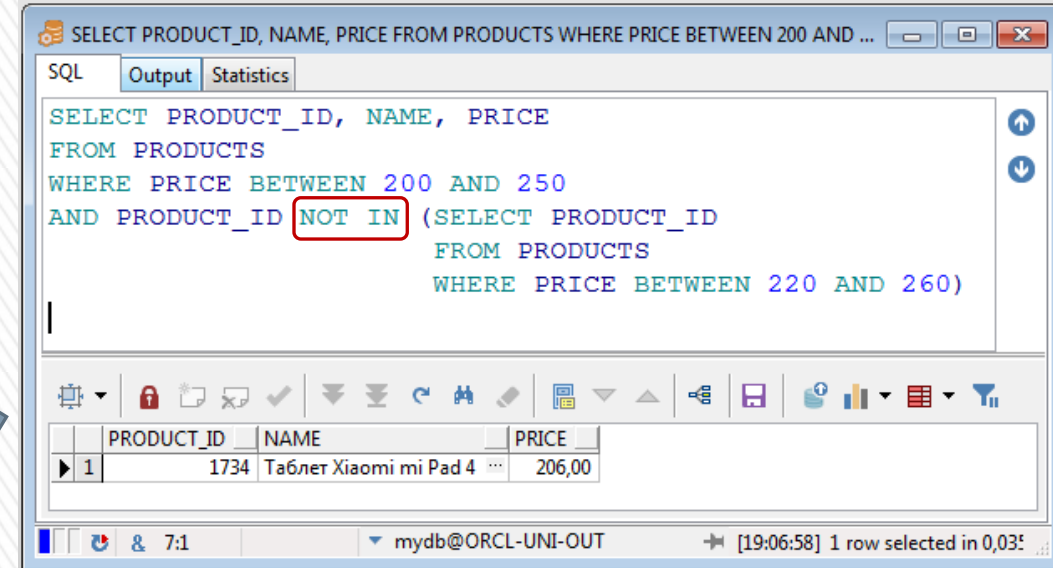
DIFFERENCE

B

PRODUCT_ID	NAME	PRICE
1749	Монитор Acer Gaming LED IPS 23.8	237.00
2373	Таблет HUAWEI MediaPad T3	256.00
2879	Дънна платка GIGABYTE Z370P D3	256.00
3087	Монитор Gaming LED IPS LG 25	224.00

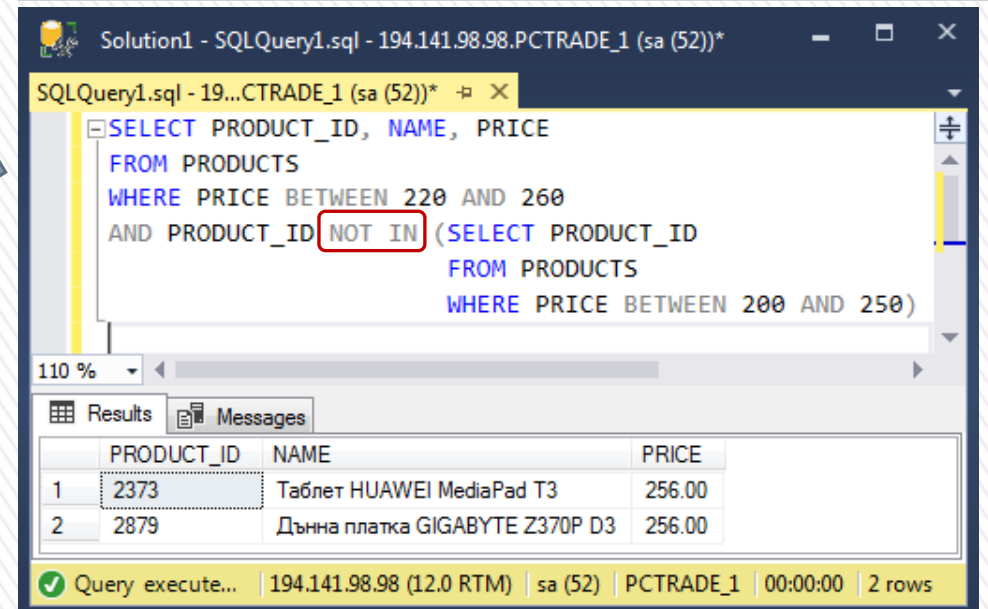
A - B

B - A



```
SELECT PRODUCT_ID, NAME, PRICE
FROM PRODUCTS
WHERE PRICE BETWEEN 200 AND 250
AND PRODUCT_ID NOT IN (SELECT PRODUCT_ID
                        FROM PRODUCTS
                        WHERE PRICE BETWEEN 220 AND 260)
```

The screenshot shows the SQL Developer interface. The SQL window contains the query. The Output window shows one row selected: PRODUCT\_ID 1734, NAME Таблет Xiaomi mi Pad 4, PRICE 206.00.



```
SELECT PRODUCT_ID, NAME, PRICE
FROM PRODUCTS
WHERE PRICE BETWEEN 220 AND 260
AND PRODUCT_ID NOT IN (SELECT PRODUCT_ID
                        FROM PRODUCTS
                        WHERE PRICE BETWEEN 200 AND 250)
```

The screenshot shows the SQL Developer interface. The SQL window contains the query. The Results window shows two rows: PRODUCT\_ID 2373, NAME Таблет HUAWEI MediaPad T3, PRICE 256.00; and PRODUCT\_ID 2879, NAME Дънна платка GIGABYTE Z370P D3, PRICE 256.00.



# PRODUCT (CARTESIAN)

A	×	1	=	A	1
B		2		A	2
C				B	1
				B	2
				C	1
				C	2

Създава нова релация, състояща се от всички възможни комбинации от записи от двете релации.

- » Произведението (Декартово) на две релации A и B поражда трета релация със:
  - > заглавна част обединението на заглавните части на A и B;
  - > тяло – множеството от всички записи t, където t е конкатенацията на всеки запис от A с всеки от B;
  - > степен на резултата – сумата от степените на A и B;
  - > кардиналност на резултата – произведението от кардиналностите на A и B.



# PRODUCT – пример

**A**

REGION_ID	NAME
1	Източна Европа
3	Азия
5	Западна Европа

**PRODUCT**

**A x B**

**B**

COUNTRY_ID	NAME	REGION_ID
BG	България	1
CH	Швейцария	5
CN	Китай	3
DE	Германия	5

	REGION_ID	NAME	COUNTRY_ID	NAME	REGION_ID
1	1	Източна Европа	BG	България	1
2	1	Източна Европа	CH	Швейцария	5
3	1	Източна Европа	CN	Китай	3
4	1	Източна Европа	DE	Германия	5
5	3	Азия	BG	България	1
6	3	Азия	CH	Швейцария	5
7	3	Азия	CN	Китай	3
8	3	Азия	DE	Германия	5
9	5	Западна Европа	BG	България	1
10	5	Западна Европа	CH	Швейцария	5
11	5	Западна Европа	CN	Китай	3
12	5	Западна Европа	DE	Германия	5

# JOIN

- » Това е сред най-полезните оператори в релационната алгебра и е най-често използвания способ за комбиниране на данни от две или повече релации;

Ще разгледаме следните видове JOIN операции:

- » Cross join
- » Natural join
- » Theta ( $\Theta$ ) join
- » Equi-join
- » Semi-join
- » Anti-join
- » Outer join



# CROSS JOIN

A

REGION_ID	NAME
1	Източна Европа
3	Азия
5	Западна Европа

CROSS JOIN

B

COUNTRY_ID	NAME	REGION_ID
BG	България	1
CH	Швейцария	5
CN	Китай	3
DE	Германия	5

	REGION_ID	NAME	COUNTRY_ID	NAME	REGION_ID
1	1	Източна Европа	BG	България	1
2	1	Източна Европа	CH	Швейцария	5
3	1	Източна Европа	CN	Китай	3
4	1	Източна Европа	DE	Германия	5
5	3	Азия	BG	България	1
6	3	Азия	CH	Швейцария	5
7	3	Азия	CN	Китай	3
8	3	Азия	DE	Германия	5
9	5	Западна Европа	BG	България	1
10	5	Западна Европа	CH	Швейцария	5
11	5	Западна Европа	CN	Китай	3
12	5	Западна Европа	DE	Германия	5

Създава нова релация от двете дадени релации, като:

- ✓ Заглавната част е обединението на двете заглавни части на операндите;
- ✓ Тялото съдържа всички възможни комбинации от записите в двете релации (еквивалентен на PRODUCT).



# NATURAL JOIN

A

REGION_ID	REGION_NAME
1	Europe
2	Americas
3	Asia

REGION_ID	REGION_NAME	COUNTRY_ID	COUNTRY_NAME
1	Europe	CH	Switzerland
1	Europe	DE	Germany
2	Americas	AR	Argentina
3	Asia	CN	China

## NATURAL JOIN

B

COUNTRY_ID	COUNTRY_NAME	REGION_ID
AR	Argentina	2
CH	Switzerland	1
CN	China	3
DE	Germany	1

Създава нова релация от двете дадени релации, като:

- ✓ Заглавната част е обединението на двете заглавни части на операндите;
- ✓ Тялото съдържа всички възможни свързани двойки от записи, така че всяка двойка да има равенство на стойностите във всички атрибути с еднакви имена (общите) от двете релации.





# NATURAL JOIN – 1.PRODUCT

» Може да бъде представен като: 1. Декартово произведение (PRODUCT), последвано от операциите 2. SELECT и 3. PROJECT.

A

REGION_ID	REGION_NAME
1	Europe
2	Americas
3	Asia

A x B

PRODUCT



B

COUNTRY_ID	COUNTRY_NAME	REGION_ID
AR	Argentina	2
CH	Switzerland	1
CN	China	3
DE	Germany	1

REGION_ID	REGION_NAME	COUNTRY_ID	COUNTRY_NAME	REGION_ID
1	Europe	AR	Argentina	2
1	Europe	CH	Switzerland	1
1	Europe	CN	China	3
1	Europe	DE	Germany	1
2	Americas	AR	Argentina	2
2	Americas	CH	Switzerland	1
2	Americas	CN	China	3
2	Americas	DE	Germany	1
3	Asia	AR	Argentina	2
3	Asia	CH	Switzerland	1
3	Asia	CN	China	3
3	Asia	DE	Germany	1



# NATURAL JOIN – 2.SELECT

REGION_ID	REGION_NAME	COUNTRY_ID	COUNTRY_NAME	REGION_ID
1	Europe	AR	Argentina	2
1	Europe	CH	Switzerland	1
1	Europe	CN	China	3
1	Europe	DE	Germany	1
2	Americas	AR	Argentina	2
2	Americas	CH	Switzerland	1
2	Americas	CN	China	3
2	Americas	DE	Germany	1
3	Asia	AR	Argentina	2
3	Asia	CH	Switzerland	1
3	Asia	CN	China	3
3	Asia	DE	Germany	1

SELECT

A.REGION\_ID = B.REGION\_ID



REGION_ID	REGION_NAME	COUNTRY_ID	COUNTRY_NAME	REGION_ID
1	Europe	CH	Switzerland	1
1	Europe	DE	Germany	1
2	Americas	AR	Argentina	2
3	Asia	CN	China	3



# NATURAL JOIN – 3.PROJECT

REGION_ID	REGION_NAME	COUNTRY_ID	COUNTRY_NAME	REGION_ID
1	Europe	CH	Switzerland	1
1	Europe	DE	Germany	1
2	Americas	AR	Argentina	2
3	Asia	CN	China	3

Без дубликати на колони

**PROJECT**



REGION_ID	REGION_NAME	COUNTRY_ID	COUNTRY_NAME
1	Europe	CH	Switzerland
1	Europe	DE	Germany
2	Americas	AR	Argentina
3	Asia	CN	China



# NATURAL JOIN – Пример

```
SELECT * FROM regions NATURAL JOIN countries  
ORDER BY region_id
```

Output x Query Result x

SQL | All Rows Fetched: 25 in 0,019 seconds

REGION_ID	REGION_NAME	COUNTRY_ID	COUNTRY_NAME
1	Europe	NL	Netherlands
1	Europe	FR	France
1	Europe	UK	United Kingdom
1	Europe	DK	Denmark
1	Europe	BE	Belgium
1	Europe	CH	Switzerland
1	Europe	IT	Italy
1	Europe	DE	Germany
2	Americas	US	United States of America
2	Americas	CA	Canada
2	Americas	MX	Mexico
2	Americas	BR	Brazil
2	Americas	AR	Argentina
3	Asia	ML	Malaysia
3	Asia	JP	Japan
3	Asia	IN	India
3	Asia	CN	China
3	Asia	AU	Australia
3	Asia	SG	Singapore
4	Middle East and Africa	IL	Israel
4	Middle East and Africa	ZM	Zambia
4	Middle East and Africa	EG	Egypt
4	Middle East and Africa	ZW	Zimbabwe
4	Middle East and Africa	NG	Nigeria
4	Middle East and Africa	KW	Kuwait

Забележки:

- ✓ Не всички SQL езици поддържат синтаксиса за NATURAL JOIN;
- ✓ От фигурата се вижда, че атрибутът REGION\_ID участва само веднъж в резултата, за разлика от примера с PRODUCT.



# $\Theta$ -JOIN

- » Нека релациите A и B нямат общи атрибути и нека  $\Theta$  е валиден оператор за сравнение ( $=$ ,  $>$ ,  $<$ ,  $\geq$ ,  $\leq$ ,  $<>$ ). Тогава  $\Theta$ -join на релацията A върху атрибута X с релацията B върху атрибута Y е резултатът от изпълнението на израза:

**$(A \times B) \text{ WHERE } X \Theta Y$**

т.е. резултатната релация е със:

- » заглавна част – като на Декартовото произведение на A и B, т.е. обединението на заглавните части на A и B;
- » тяло – множеството на всички записи, принадлежащи на Декартовото произведение, за които  $X \Theta Y$  е вярно.





# ⊕-JOIN – Пример 1

- » Нека имаме следните релации A и B, съдържащи съответно преподаватели и университети, в които те преподават и нека в случая ⊕ е операторът за сравнение '='. Тогава резултатната релация ще съдържа само записите от Декартовото произведение, оцветени в жълто:

A	TEACH_IN	FName	LName	Title
	101	Стоян	Колев	Проф.
	102	Петър	Иванов	Доц.
	101	Венета	Георгиева	Ст.н.с.

B	UNI_ID	Name	Short
	101	Пловдивски университет	ПУ
	102	Софийски университет	СУ

**(A × B) WHERE A.TEACH\_IN ⊕ B.UNI\_ID**

TEACH_IN	FName	LName	Title	UNI_ID	Name	Short
101	Стоян	Колев	Проф.	101	Пловдивски университет	ПУ
101	Стоян	Колев	Проф.	102	Софийски университет	СУ
102	Петър	Иванов	Доц.	101	Пловдивски университет	ПУ
102	Петър	Иванов	Доц.	102	Софийски университет	СУ
101	Венета	Георгиева	Ст.н.с.	101	Пловдивски университет	ПУ
101	Венета	Георгиева	Ст.н.с.	102	Софийски университет	СУ



# Ⓜ-JOIN – Пример 2

Solution1 - SQLQuery1.sql - 194.141.98.98.TradeCompany (sa (52))\*

SQLQuery1.sql - 19...eCompany (sa (52))\*

```
SELECT *  
FROM REGIONS, COUNTRIES  
WHERE REGIONS.REGION_ID = COUNTRIES.REGION_ID
```

236 %

Results Messages

	REGION_ID	NAME	COUNTRY_ID	NAME	REGION_ID
1	1	Източна Европа	BG	България	1
2	5	Западна Европа	CH	Швейцария	5
3	3	Азия	CN	Китай	3
4	5	Западна Европа	DE	Германия	5



# EQUI-JOIN

- » Това е частен случай на  $\Theta$ -join, в който операторът  $\Theta$  е само операторът за сравнение  $=$ .



# SEMI-JOIN

- » Връща редовете от първата релация, за които има поне един съвпадащ от втората релация;
- » Разликата между него и досега описаните е, че редовете от първата релация ще участват в резултата най-много по веднъж;
- » Дори втората релация да има два съвпадащи за ред от първата, само едно копие на реда ще бъде върнато в резултата;
- » Реализира се с предикатите EXISTS или IN.



# SEMI-JOIN – Пример

» Ще използваме таблиците REGIONS и COUNTRIES за демонстрация

SQLQuery1.sql - 19...eCompany (sa (52))\*

```
SELECT *  
FROM REGIONS
```

236 %

	REGION_ID	NAME
1	1	Източна Европа
2	2	Америка
3	3	Азия
4	4	Среден изток и Африка
5	5	Западна Европа
6	6	Северен полюс

SQLQuery1.sql - 19...eCompany (sa (52))\*

```
SELECT *  
FROM COUNTRIES
```

236 %

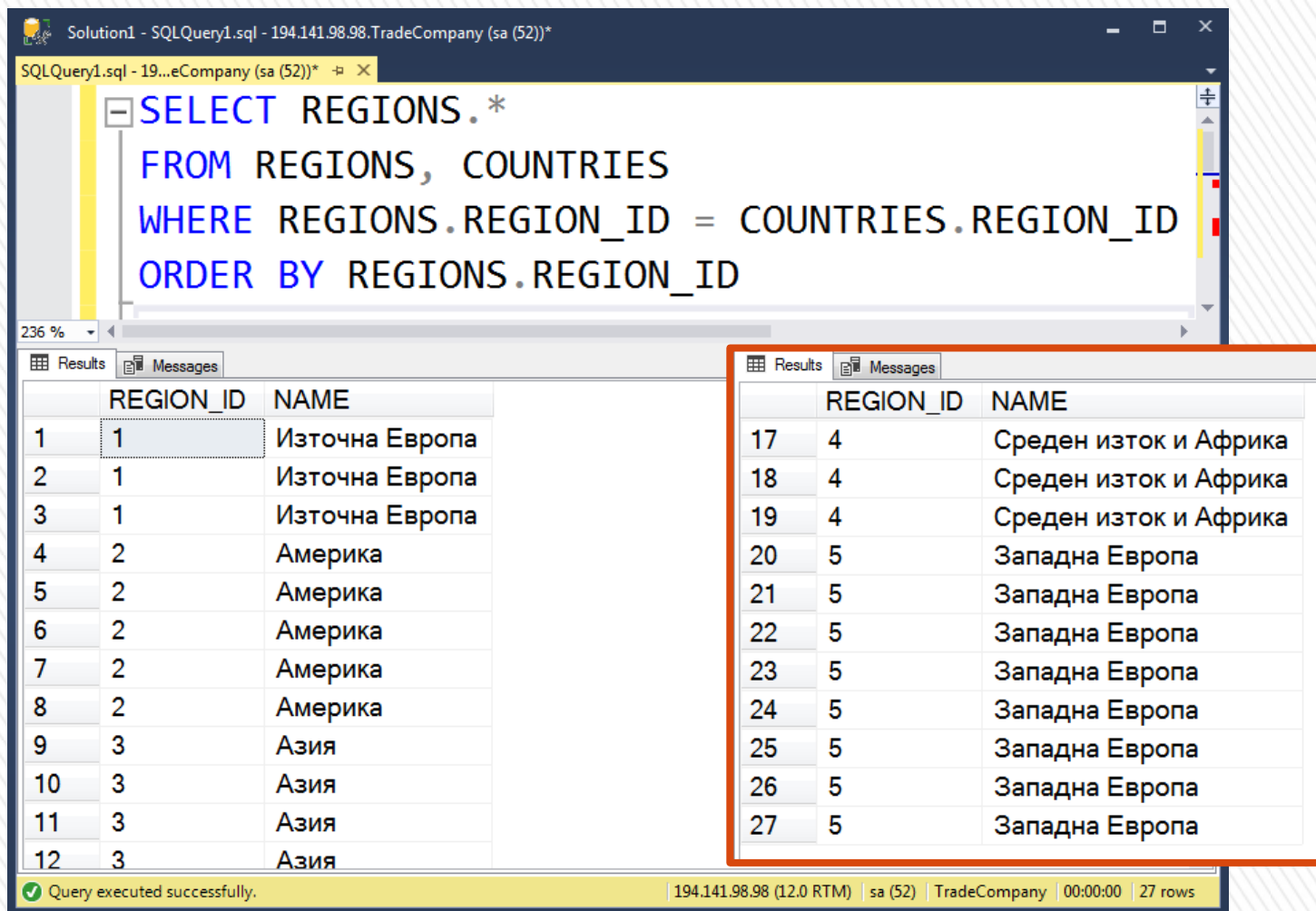
	COUNTRY_ID	NAME	REGION_ID
1	AR	Аржентина	2
2	AU	Австралия	NULL
3	BE	Белгия	5
4	BG	България	1
5	BR	Бразилия	2
6	CA	Канада	2
7	CH	Швейцария	5
8	CN	Китай	3
9	DE	Германия	5





# SEMI-JOIN – Пример

» Ще покажем всички региони, за които има поне една въведена държава:



```
SELECT REGIONS.*
FROM REGIONS, COUNTRIES
WHERE REGIONS.REGION_ID = COUNTRIES.REGION_ID
ORDER BY REGIONS.REGION_ID
```

	REGION_ID	NAME
1	1	Източна Европа
2	1	Източна Европа
3	1	Източна Европа
4	2	Америка
5	2	Америка
6	2	Америка
7	2	Америка
8	2	Америка
9	3	Азия
10	3	Азия
11	3	Азия
12	3	Азия

	REGION_ID	NAME
17	4	Среден изток и Африка
18	4	Среден изток и Африка
19	4	Среден изток и Африка
20	5	Западна Европа
21	5	Западна Европа
22	5	Западна Европа
23	5	Западна Европа
24	5	Западна Европа
25	5	Западна Европа
26	5	Западна Европа
27	5	Западна Европа

Query executed successfully. | 194.141.98.98 (12.0 RTM) | sa (52) | TradeCompany | 00:00:00 | 27 rows

Ясно се вижда, че редовете за регионите се повтарят толкова пъти, колкото държави има в региона.

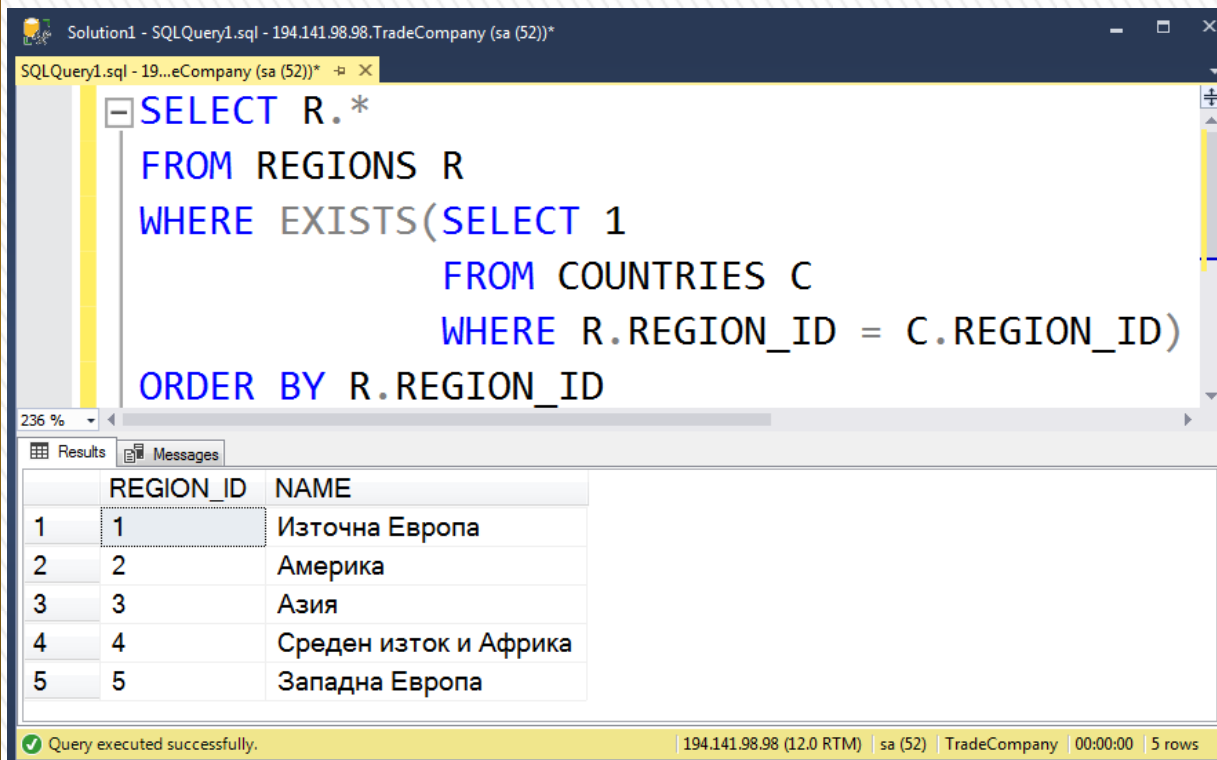
Също се вижда, че регион с REGION\_ID = 6 не участва в резултата, защото няма държави в този регион.

Дубликатите могат да бъдат елиминирани с ключовата дума DISTINCT, но това ще повлияе само на визуализацията им, не и на ефективността на извличане на данните.



# SEMI-JOIN – Пример

» Следните заявки ще извлекат същите данни, но без дубликати и по-ефективно:



The screenshot shows a SQL query window with the following text:

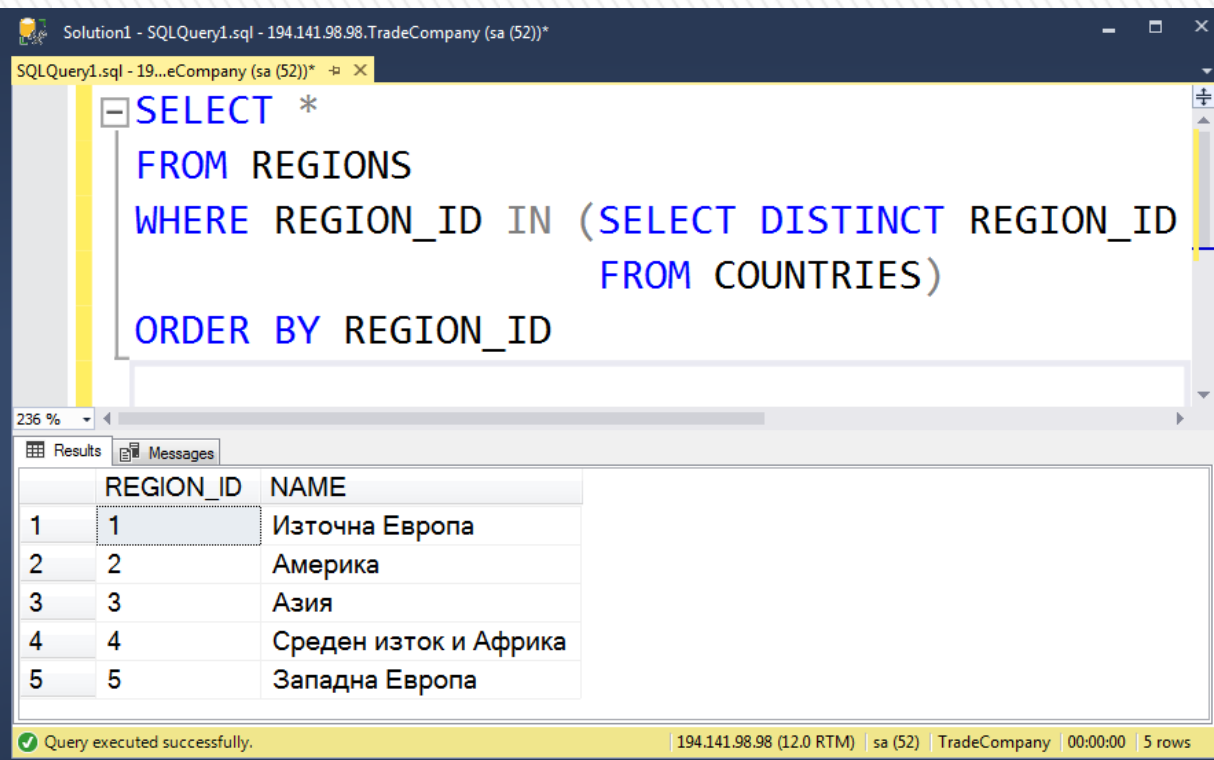
```
SELECT R.*
FROM REGIONS R
WHERE EXISTS(SELECT 1
             FROM COUNTRIES C
             WHERE R.REGION_ID = C.REGION_ID)
ORDER BY R.REGION_ID
```

Below the query, the Results tab displays a table with 5 rows:

	REGION_ID	NAME
1	1	Източна Европа
2	2	Америка
3	3	Азия
4	4	Среден изток и Африка
5	5	Западна Европа

The status bar at the bottom indicates: Query executed successfully. | 194.141.98.98 (12.0 RTM) | sa (52) | TradeCompany | 00:00:00 | 5 rows

Тук броят върнати редове във вложената заявка се проверява дали е 0 или повече, също така няма значение какви колони връща тя.



The screenshot shows a SQL query window with the following text:

```
SELECT *
FROM REGIONS
WHERE REGION_ID IN (SELECT DISTINCT REGION_ID
                   FROM COUNTRIES)
ORDER BY REGION_ID
```

Below the query, the Results tab displays a table with 5 rows:

	REGION_ID	NAME
1	1	Източна Европа
2	2	Америка
3	3	Азия
4	4	Среден изток и Африка
5	5	Западна Европа

The status bar at the bottom indicates: Query executed successfully. | 194.141.98.98 (12.0 RTM) | sa (52) | TradeCompany | 00:00:00 | 5 rows

Тук за всеки ред от REGIONS се проверява стойността на REGION\_ID дали е измежду върнатите от вложената заявка.



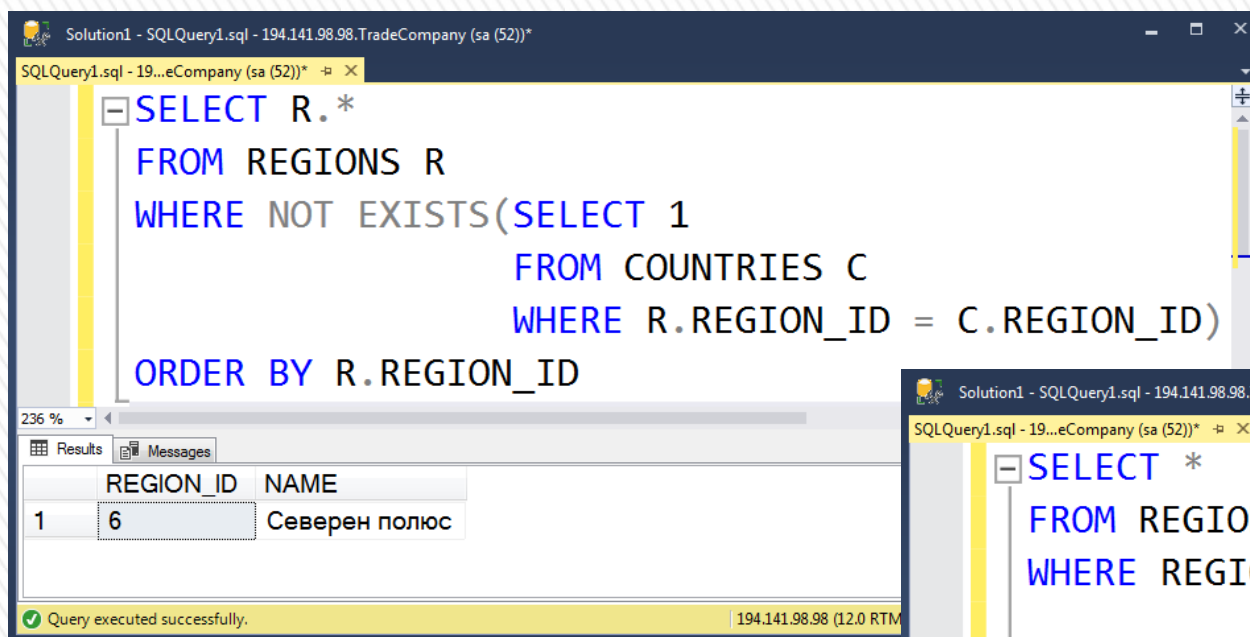
# ANTI-JOIN

- » Операторът ANTI-JOIN между две релации върши обратното на SEMI-JOIN: връща редовете от първата релация, които нямат съвпадения във втората.
- » По своята природа това е операторът за разлика (MINUS), но може да бъде реализиран и с предикатите NOT EXISTS или NOT IN.



# ANTI-JOIN – Пример

» Следните заявки извличат регионите, които нямат държави:



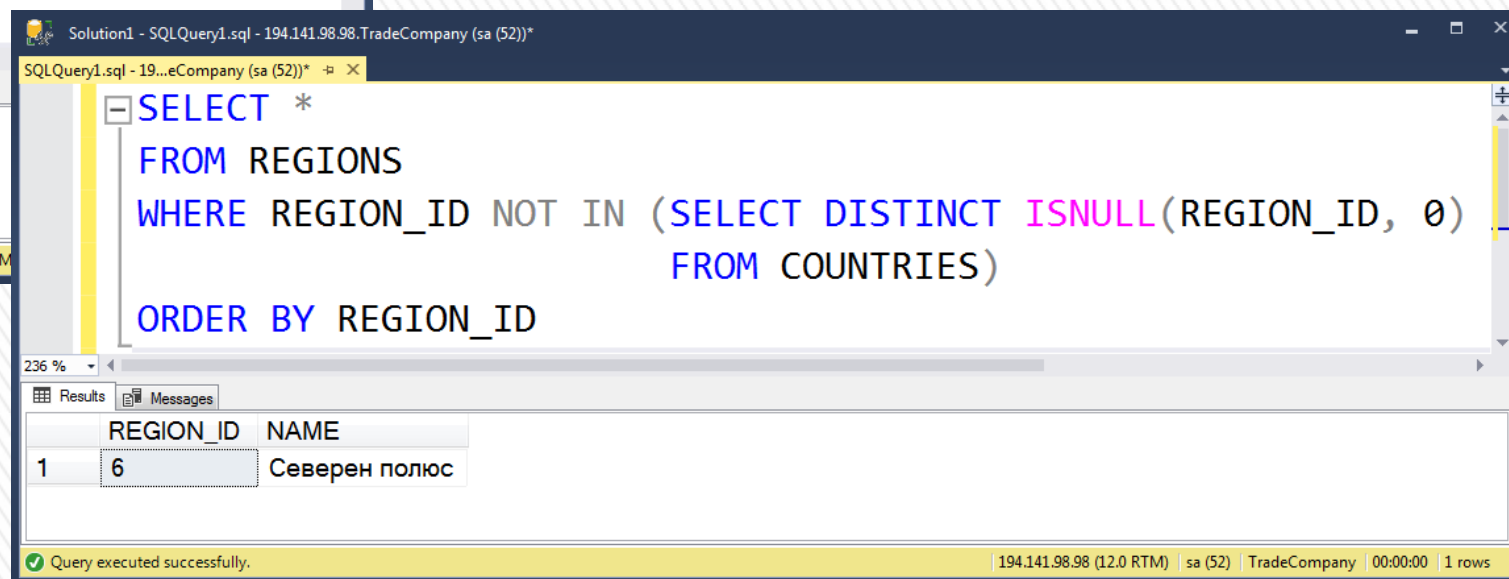
The screenshot shows a SQL query window with the following text:

```
SELECT R.*  
FROM REGIONS R  
WHERE NOT EXISTS(SELECT 1  
                  FROM COUNTRIES C  
                  WHERE R.REGION_ID = C.REGION_ID)  
ORDER BY R.REGION_ID
```

Below the query, the results are displayed in a table:

REGION_ID	NAME
6	Северен полюс

The status bar at the bottom indicates "Query executed successfully." and "194.141.98.98 (12.0 RTM)".



The screenshot shows a SQL query window with the following text:

```
SELECT *  
FROM REGIONS  
WHERE REGION_ID NOT IN (SELECT DISTINCT ISNULL(REGION_ID, 0)  
                        FROM COUNTRIES)  
ORDER BY REGION_ID
```

Below the query, the results are displayed in a table:

REGION_ID	NAME
6	Северен полюс

The status bar at the bottom indicates "Query executed successfully." and "194.141.98.98 (12.0 RTM) | sa (52) | TradeCompany | 00:00:00 | 1 rows".

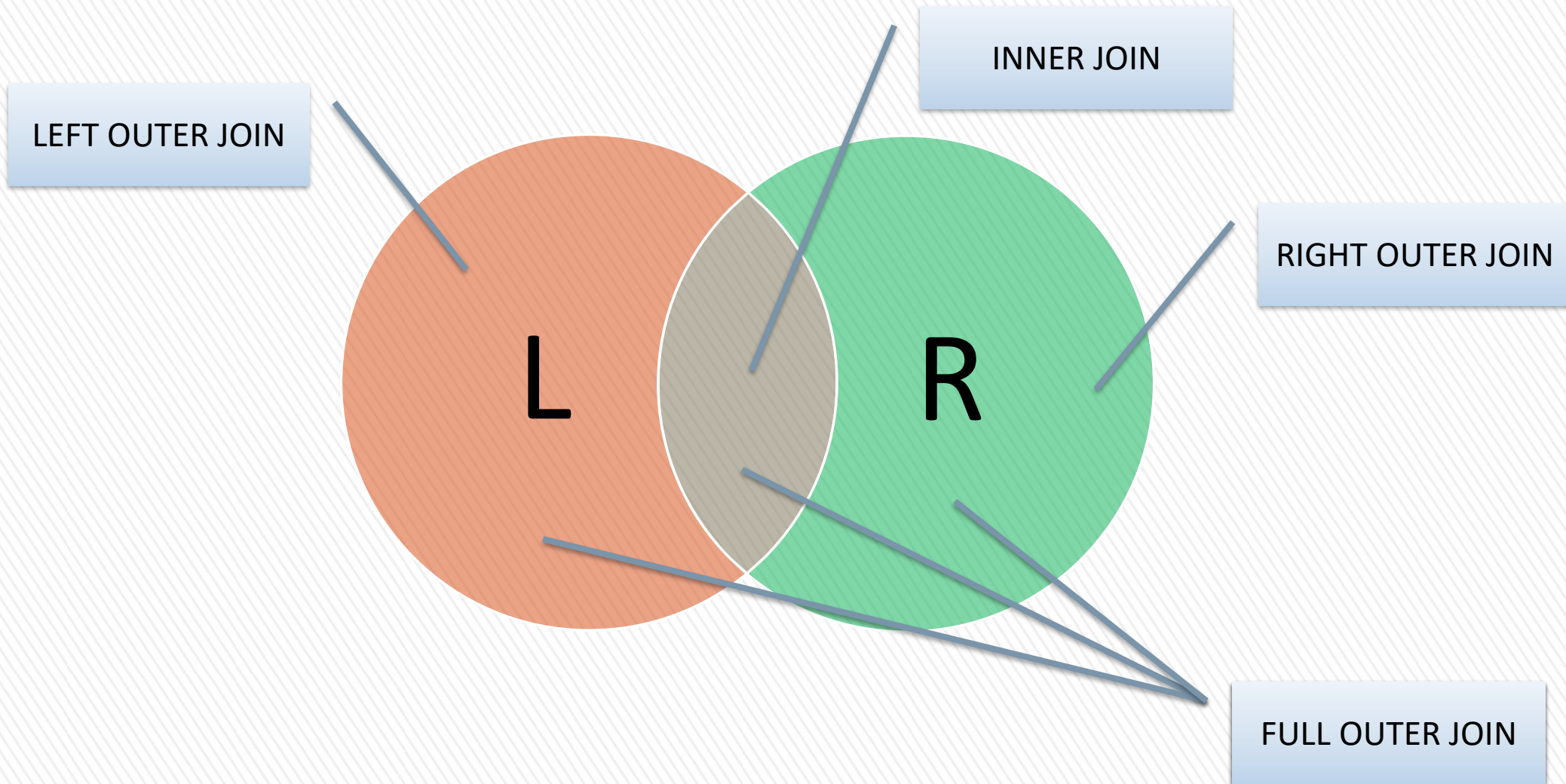
# INNER/OUTER JOIN

- » Изброените дотук видове JOIN операции (NATURAL JOIN,  $\Theta$ -JOIN) реализират т.нар. вътрешни съединения, характерни с това, че в резултата участват само редовете от двете релации, които имат съвпадения.
- » За случаите, в които ще се налага от една от двете или и от двете релации да бъдат запазени всички редове в резултатната релация, се използват външни съединения.





# INNER/OUTER JOIN

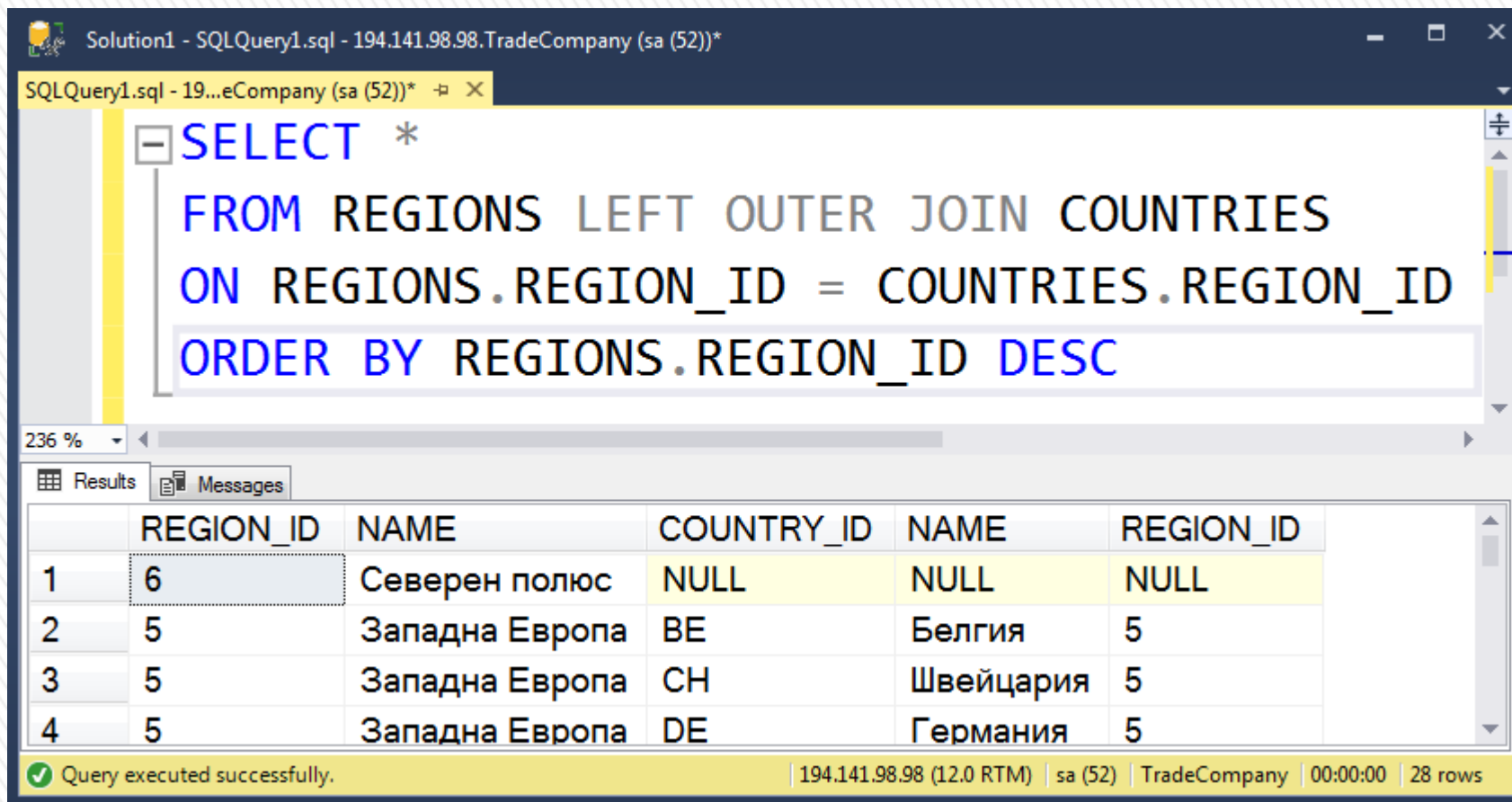


# OUTER JOIN

- » Външното съединение генерира релация, в която записите, които нямат съвпадения в двете релации, могат също да участват в резултата;
- » Видове:
  1. *Left Outer Join*: съединение, в което записите от L, които нямат съответни в R (сравнение в общите атрибути), също ще участват в резултатната релация.
  2. *Right Outer Join*: съединение, в което записите от R, които нямат съответни в L, също ще участват в резултатната релация.
  3. *Full Outer Join*: съединение, в което записите от L, които нямат съответни в R, ще участват в резултатната релация, както и тези от R, които нямат съответни в L, също ще участват в резултатната релация.



# LEFT OUTER JOIN – Пример



The screenshot displays the SQL Server Enterprise Manager interface. The top pane shows the following SQL query:

```
SELECT *  
FROM REGIONS LEFT OUTER JOIN COUNTRIES  
ON REGIONS.REGION_ID = COUNTRIES.REGION_ID  
ORDER BY REGIONS.REGION_ID DESC
```

The bottom pane shows the results of the query in a table with 6 columns: REGION\_ID, NAME, COUNTRY\_ID, NAME, and REGION\_ID. The results are as follows:

	REGION_ID	NAME	COUNTRY_ID	NAME	REGION_ID
1	6	Северен полюс	NULL	NULL	NULL
2	5	Западна Европа	BE	Белгия	5
3	5	Западна Европа	CH	Швейцария	5
4	5	Западна Европа	DE	Германия	5

The status bar at the bottom indicates: Query executed successfully. | 194.141.98.98 (12.0 RTM) | sa (52) | TradeCompany | 00:00:00 | 28 rows



# RIGHT OUTER JOIN – Пример

Solution1 - SQLQuery1.sql - 194.141.98.98.TradeCompany (sa (52))\*

SQLQuery1.sql - 19...eCompany (sa (52))\*

```
SELECT *  
FROM REGIONS RIGHT OUTER JOIN COUNTRIES  
ON REGIONS.REGION_ID = COUNTRIES.REGION_ID  
ORDER BY COUNTRIES.NAME
```

236 %

Results Messages

	REGION_ID	NAME	COUNTRY_ID	NAME	REGION_ID
1	NULL	NULL	AU	Австралия	NULL
2	2	Америка	AR	Аржентина	2
3	5	Западна Европа	BE	Белгия	5
4	2	Америка	BR	Бразилия	2
5	1	Източна Европа	BG	България	1
6	5	Западна Европа	UK	Великобритания	5
7	5	Западна Европа	DE	Германия	5
8	NULL	NULL	GR	Гърция	NULL
9	5	Западна Европа	DK	Дания	5
10	4	Среден изток ...	EG	Египет	4

Query executed successfully. 194.141.98.98 (12.0 RTM) sa (52) TradeCompany 00:00:00 29 rows



# FULL OUTER JOIN – Пример

Solution1 - SQLQuery1.sql - 194.141.98.98.TradeCompany (sa (52))\*

SQLQuery1.sql - 19...eCompany (sa (52))\*

```
SELECT *  
FROM REGIONS FULL OUTER JOIN COUNTRIES  
ON REGIONS.REGION_ID = COUNTRIES.REGION_ID  
ORDER BY COUNTRIES.NAME
```

236 %

Results Messages

	REGION_ID	NAME	COUNTRY_ID	NAME	REGION_ID
1	6	Северен полюс	NULL	NULL	NULL
2	NULL	NULL	AU	Австралия	NULL
3	2	Америка	AR	Аржентина	2
4	5	Западна Европа	BE	Белгия	5
5	2	Америка	BR	Бразилия	2
6	1	Източна Европа	BG	България	1
7	5	Западна Европа	UK	Великобритания	5
8	5	Западна Европа	DE	Германия	5
9	NULL	NULL	GR	Гърция	NULL
10	5	Западна Европа	DK	Дания	5

Query executed successfully. 194.141.98.98 (12.0 RTM) | sa (52) | TradeCompany | 00:00:00 | 30 rows






# DIVIDE

A	B
A1	X
A1	Y
A1	Z
A2	X
A2	Y

B
X
Z

A
A1



DIVIDE – от две релации генерира нова релация, която съдържа всички стойности на атрибута A от първата релация, които съответстват (равни в другия атрибут B) на всички стойности на атрибута B от втората релация.



# DIVIDE

- » Нека релациите  $A$  и  $B$  имат заглавни части  $\{X_1, \dots, X_m, Y_1, \dots, Y_n\}$  и  $\{Y_1, \dots, Y_n\}$ , т.е. атрибутите  $\{Y_1, \dots, Y_n\}$  са общи за двете релации, а  $B$  няма други атрибути. Допускаме, че общите атрибути са дефинирани върху общи домейни. Нека разглеждаме  $\{X_1, \dots, X_m\}$  – като  $X$ ,  $\{Y_1, \dots, Y_n\}$  – като  $Y$ .

$A \text{ DIVIDE } B$  е релация със:

- > заглавна част  $\{X\}$ ;
- > тяло – множеството на всички записи  $\{X:x\}$  така, че един запис  $\{X:x, Y:y\}$  се появява в  $A$  за всички записи  $\{Y:y\}$ , появяващи се в  $B$ .

Накратко:

- > операторът за деление дели една релация  $A$  от степен  $m+n$  на друга релация  $B$  от степен  $n$  и създава нова релация от степен  $m$ ;
- >  $(m+i)$ -тият атрибут на  $A$  и  $i$ -тият атрибут на  $B$  трябва да са дефинирани върху един и същ домейн.



# DIVIDE – Пример 1

Искаме да извлечем всички студенти, които са завършили поставените им задачи по Бази от данни и ще бъдат допуснати до изпит.

**Completed**

Student	Task
Стоян Колев	БД - самостоятелна работа
Стоян Колев	БД – упражнения
Венета Георгиева	БД – упражнения
Венета Георгиева	Курсов проект СИ
Венета Георгиева	БД - самостоятелна работа
Иван Пенев	БД – упражнения
Иван Пенев	Компютърна графика

÷

**Projects**

Task
БД - самостоятелна работа
БД – упражнения

=

Student
Стоян Колев
Венета Георгиева



# DIVIDE – Пример 2

Да извлечем всички служители, отговарящи на изброените изисквания.

## EMPS

name	skill
Иван Георгиев	SQL
Иван Георгиев	XML
Иван Георгиев	C#
Мария Петрова	SQL
Мария Петрова	XML
Мария Петрова	Java
Стефан Георгиев	SQL
Стефан Георгиев	XML
Стефан Георгиев	Java
Соня Колева	SQL
Соня Колева	.Net
Соня Колева	Java

÷

## REQUIREMENTS

skill
Java
SQL
XML

=

SQLQuery1.sql - 19...eCompany (sa (52))\*

```
SELECT E.NAME, COUNT(E.NAME)
FROM EMPS E, REQUIREMENTS R
WHERE E.SKILL = R.SKILL
GROUP BY E.NAME
HAVING COUNT(E.NAME) = (SELECT COUNT(*) FROM REQUIREMENTS)
```

236 %

Results Messages

	NAME	(No column name)
1	Мария Петрова	3
2	Стефан Георгиев	3

# Алгебрични свойства на операторите

## Асоциативност:

- > UNION:  $(A \cup B) \cup C \Leftrightarrow A \cup (B \cup C)$
- > INTERSECTION:  $(A \cap B) \cap C \Leftrightarrow A \cap (B \cap C)$
- > PRODUCT:  $(A \times B) \times C \Leftrightarrow A \times (B \times C)$
- > JOIN:  $(A \Join B) \Join C \Leftrightarrow A \Join (B \Join C) \Leftrightarrow A \Join B \Join C$

## Комутативност:

- > UNION:  $A \cup B = B \cup A$
- > INTERSECTION:  $A \cap B = B \cap A$
- > PRODUCT:  $A \times B \Leftrightarrow B \times A$
- > JOIN:  $A \Join B \Leftrightarrow B \Join A$





# За какво е релационната алгебра?

Демонстрираните примери бяха предимно за извличане на данни, но това не означава, че релационната алгебра е приложима само при извличане. Нейната основна цел е да позволи писането на изрази, които да послужат за:

- > Дефиниране на обхват за извлечени данни – задаване на условия, на които да отговарят резултатите;
- > Дефиниране на обхват за промяна на данни – при въвеждане, промяна и изтриване;
- > Дефиниране на (именувани) виртуални релации – изгледи, напр.;
- > Дефиниране на правила за сигурност;
- > Дефиниране на правила за цялостност;
- > И др.

