



»Гл. ас. д-р Георги Чолаков

»Бази 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 (унарен) 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
1753	Ръчен видео дикт. ADATA Dualink HD330	159.00	1TB, Shock Sensor, 2.5", USB 3.1, чепен
1754	Tablet Xplore on Pad 4	299.00	Tablet Xplore on Pad 4, 32GB RAM, 32 GB ROM, 8-яч.
1757	Tablet Samsung Tab3 T113	199.00	Tablet Samsung Tab3 T113 Lite Value Edition с процес...
1759	USB накрай Kingston DataTraveler 100 G2	9.00	8GB, USB 3.0
1762	Безжичен слушалки TP-PLUS	15.00	Безжичен слушалки TP-PLUS с кутин за зареждане
1763	Латтон 2 in 1 Lenovo YOGA Glass C930-13MB	4800.00	Латтон 2 in 1 Lenovo YOGA Glass C930-13MB with pro...
1765	Tablet Lenovo Tab Yoga 3 YTT3-X30M	349.00	Tablet Lenovo Tab Yoga 3 YTT3-X30M, 10.1", QuadCam...
1769	Monitor LG 27N AOC 24.5"	299.00	Monitor LG 27N AOC 24.5", Full HD, FreeSync 144Hz
1783	Процесор AMD FX-8350	147.00	Процесор AMD FX-8350, 4-core, 10MB, 120MHz, AM3+
1788	Латтон 2 in 1 HP ENVY x360 13-яг0010m	1845.00	Латтон 2 in 1 HP ENVY x360 13-яг0010m с процесор...

SELECT * FROM PRODUCTS WHERE PRODUCT_ID = 1745

PRODUCT_ID	NAME	PRICE	DESCR
1745	Tablet Lenovo Tab Yoga 3 YTT3-X30M	349.00	Tablet Lenovo Tab Yoga 3 YTT3-X30M...

SELECT * FROM PRODUCTS WHERE PRICE > 300

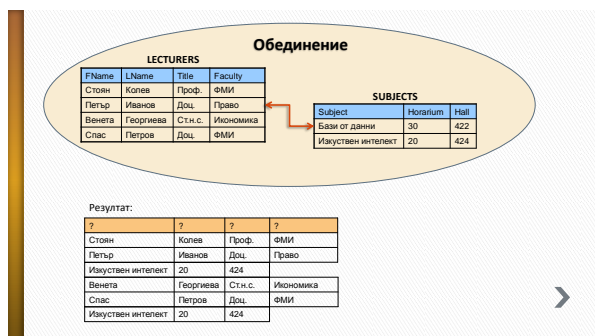
PRODUCT_ID	NAME	PRICE	DESCR
1763	Латтон 2 in 1 Lenovo YOGA Glass C930-13MB	4800.00	Латтон 2 in 1 Lenovo YOGA Glas...
1765	Tablet Lenovo Tab Yoga 3 YTT3-X30M	349.00	Tablet Lenovo Tab Yoga 3 YTT3...
1788	Латтон 2 in 1 HP ENVY x360 13-яг0010m	1845.00	Латтон 2 in 1 HP ENVY x360 13-я...

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.





» 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**.

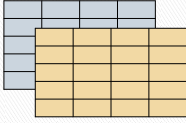
» Следователно, обединението в релационната алгебра не е идентично с математическото обединение;

» 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 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, В или на двете, като дубликатите се елиминират.



UNION – пример

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

UNION

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

UNION ALL

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

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



UNION – пример 2

UNION	
SELECT frame, lname FROM employees UNION SELECT frame, lname FROM customers ORDER BY frame, lname	
frame	lname
21	Borisov
22	Borisov
23	Borisov
24	Borisov
25	Georgiev
26	Georgiev
27	Georgiev
28	Georgiev
29	Georgiev
30	Georgiev
31	Georgiev
32	Georgiev

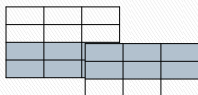
SELECT frame, lname FROM employees UNION SELECT frame, lname FROM customers ORDER BY frame, lname



UNION ALL	
SELECT frame, lname FROM employees UNION ALL SELECT frame, lname FROM customers ORDER BY frame, lname	
frame	lname
21	Borisov
22	Borisov
23	Borisov
24	Borisov
25	Georgiev
26	Georgiev
27	Georgiev
28	Georgiev
29	Georgiev
30	Georgiev
31	Georgiev
32	Georgiev

SELECT frame, lname FROM employees UNION ALL SELECT frame, lname FROM customers ORDER BY frame, lname

INTERSECT



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

- » Сечението 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 едновременно на А и В, т.е. общите за двете релации п-торки (записи).



INTERSECT - пример

PRODUCT_ID	NAME	PRICE
1734	Tablet 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	Tablet HUAWEI MediaPad T3	256.00
2879	Дъшка memory GIGABYTE Z370P D3	256.00
3087	Монитор Gaming LED IPS LG 25	224.00



INTERSECT – пример

SQL Query (SQL Server Enterprise Manager) - 104141.MB.A

```

SELECT ProductName, Price
FROM Products
ORDER BY ProductName

```

SQL Query (SQL Server Enterprise Manager) - 104141.MB.A

```

SELECT ProductName, Price
FROM Customers
ORDER BY ProductName

```

SQL Query (SQL Server Enterprise Manager) - 104141.MB.A

```

SELECT ProductName, Price
FROM Products
INTERSECT
SELECT ProductName, Price
FROM Customers
ORDER BY ProductName

```

Results (SQL Server Enterprise Manager) - 104141.MB.A

ProductID	ProductName	Price
1749	Монитор Acer Gaming LED IPS 23.8	237.00
3087	Монитор Gaming LED IPS LG 25	224.00

INTERSECT – пример с IN

PL/SQL

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

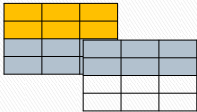
FNAME	LNAME
1. Адам	Патсон
2. Герми	Гриббл
3. Демитри	Патсон
4. Иван	Иванов

T-SQL

```
SQLQuery1 - 19.11.2019 12:41:00 PCTRADE_1 (sa [L...])
SELECT fname, lname
FROM employees
WHERE fname + lname IN (SELECT fname + lname
FROM customers)
ORDER BY fname, lname
```

fname	lname
1. Адам	Патсон
2. Герми	Гриббл
3. Демитри	Патсон
4. Иван	Иванов

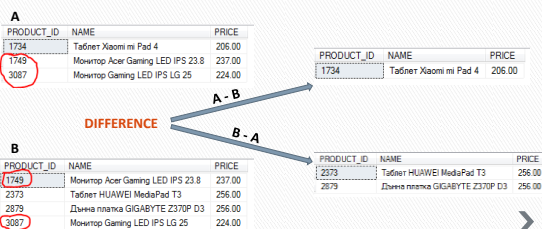
DIFFERENCE



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

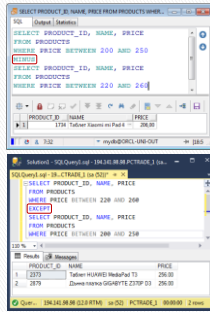


DIFFERENCE – пример

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

DIFFERENCE

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

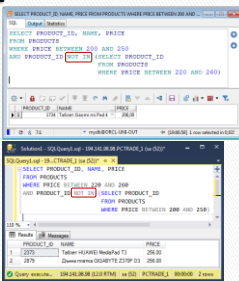


DIFFERENCE – пример 2

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

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



PRODUCT (CARTESIAN)

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

JOIN

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

Ще разгледаме следните видове JOIN операции:

- » Natural join
- » Theta (θ) join
- » Equi-join
- » Semi-join
- » Anti-join
- » Outer join



NATURAL JOIN

A

REGION_ID	NAME
1	Източна Европа
3	Азия
5	Западна Европа

NATURAL 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
1	Източна Европа	BG	България
2	Западна Европа	CH	Швейцария
3	Азия	CN	Китай
4	Западна Европа	DE	Германия

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 всяки атрибути с еднакви имена (общите) от двете релации.



NATURAL JOIN – 1.PRODUCT

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

A

REGION_ID	NAME
1	Източна Европа
3	Азия
5	Западна Европа

B

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

PRODUCT $A \times B$

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

NATURAL JOIN – 2.SELECT

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

SELECT $A.REGION_ID = B.REGION_ID$

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

NATURAL JOIN – 3.PROJECT

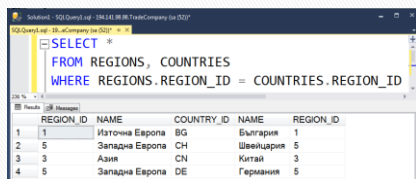
REGION_ID	NAME	COUNTRY_ID	NAME	REGION_ID
1	Източна Европа	BG	България	1
5	Западна Европа	CN	Швейцария	5
3	Азия	CN	Китай	3
5	Западна Европа	DE	Германия	5

PROJECT Без дубликати на колонии

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

[illegible]

Θ-JOIN – Пример 2



```

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

	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 Θ -join, в който операторът Θ 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

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

COUNTRY_ID	NAME	REGION_ID
1	Аргентина	2
2	Австралия	NULL
3	Белгия	5
4	България	1
5	Бразилия	2
6	Канада	2
7	Швейцария	5
8	Китай	3
9	Германия	5

SEMI-JOIN – Пример

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

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

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:

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

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

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

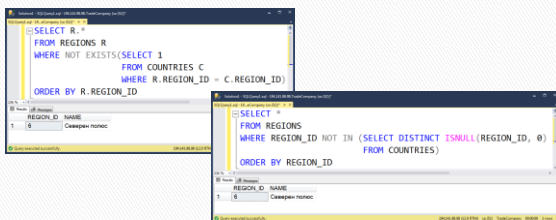
ANTI-JOIN

- » Операторът ANTI-JOIN между две релации върши обратното на SEMI-JOIN: връща редовете от първата релация, които нямат съпадения 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:

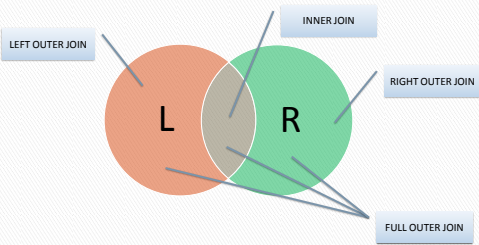


INNER/OUTER JOIN

- » Изброените дотук видове JOIN операции (NATURAL JOIN, Θ-JOIN) реализират т.нар. **вътрешни съединения**, характерни с това, че в резултата участват **само** редовете от двете релации, които имат съпадения.
- » 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 съпадения 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.

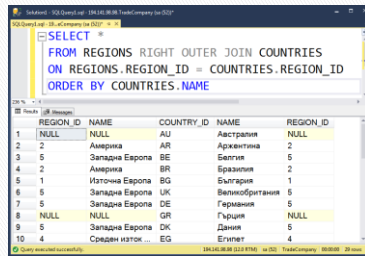
LEFT OUTER JOIN – Пример

```

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

	REGION_ID	NAME	COUNTRY_ID	NAME	REGION_ID
1	5	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

RIGHT OUTER JOIN – Пример



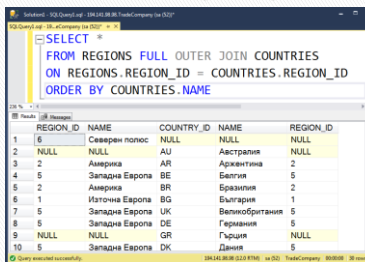
```

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

```

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

FULL OUTER JOIN – Пример



```

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

```

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

DIVIDE

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

DIVIDE – от две релации генерира нова релация, която съдържа всички стойности на атрибута А от първата релация, които съответстват (равни в другия атрибут В) на всички стойности на атрибута В 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 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+n)-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 Базис от данни и ще бъдат допуснати до изпит.

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



DIVIDE – Пример 2

Да извлечем всички служители, отговарящи на изброените изисквания.

EMPMS		÷	REQUIREMENTS		=		
name	skill		skill			NAME	
Иван Георгиев	SQL		Java			1	Мария Петрова
Иван Георгиев	XML		SQL			2	Стефан Георгиев
Иван Георгиев	CF		XML				
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						

```

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

```

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.