

10. Релационна алгебра

Лекционен курс “Бази от данни”

Въведение

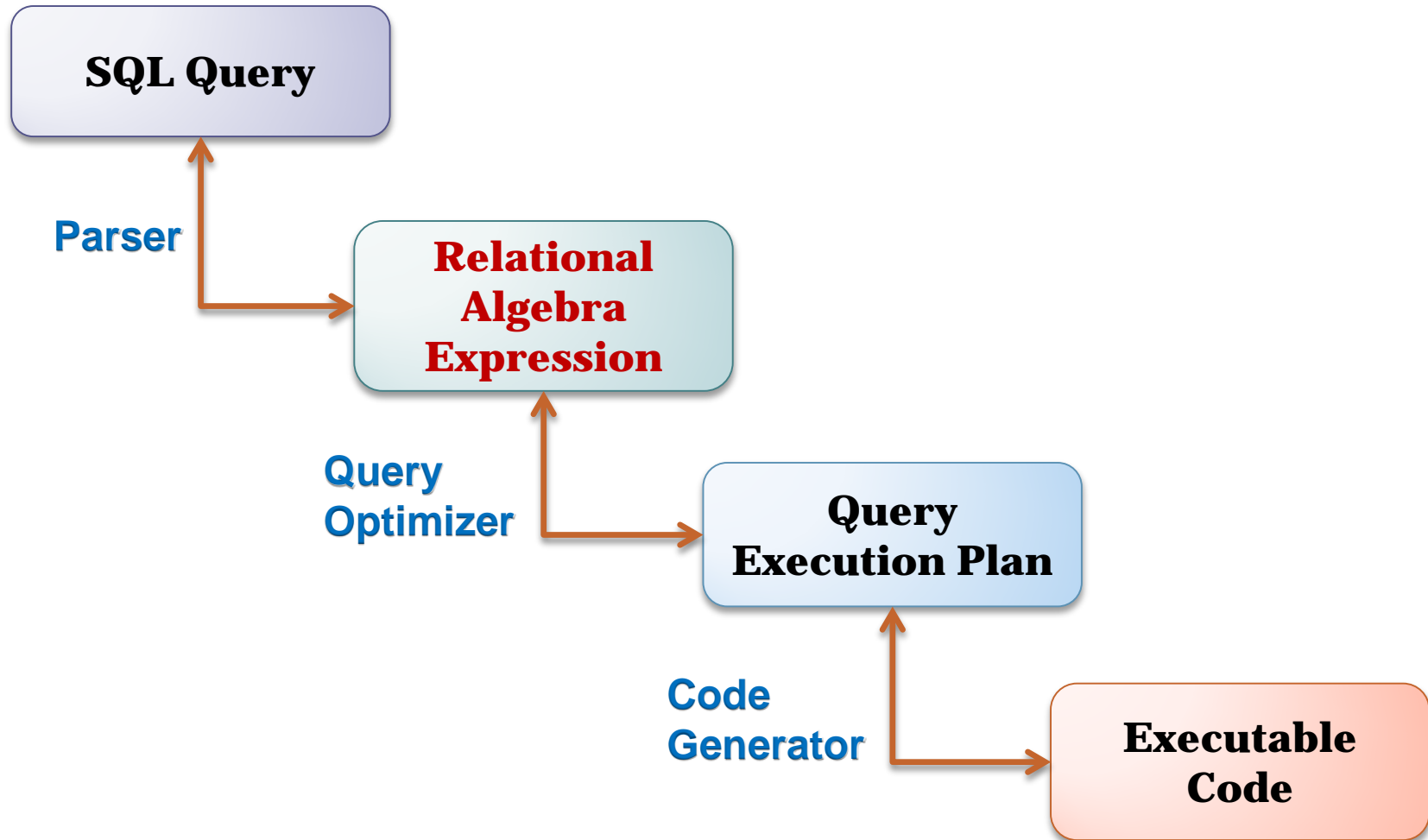
Третата и последна част на релационния модел (манипулативната част) се състои от множество от оператори, които образуват така наречената релационна алгебра.

Всеки оператор от релационната алгебра има една (унарнен) или две (бинарен) релации като входни данни и връща като резултат една нова релация.

Codd дефинира 8 операции, разделени в две групи:

- ✓ множество на традиционните (класическите) операции;
- ✓ специални релационни операции.

Къде точно в архитектурата?



Затвореност

Резултатът на всяка релационна операция е друга релация - това свойство се нарича 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;
- Затвореността има два аспекта:
 - Затвореност на заглавните части на релациите;
 - Затвореност на телата на релациите.

Релационната алгебра гарантира, че всички релации имат собствени заглавни части - т.е. заглавни части, в които всеки атрибут има собствено име, което е уникално в съдържащата го релация.

Заглавната част е добре дефинирана и позната на системата за базовите релации - но за релации, които се извеждат?

- Напр. `LECTURER join FACULTY` описва сливането на двете релации върху `F_ID` – атрибута;
- Каква е заглавната част на резултата?

LECTURER

L_NUM	SALARY	F_ID

JOIN

FACULTY

F_ID	F_NAME



L_NUM	SALARY	F_ID	F_NAME

- Затвореността изисква тя отново да е заглавна част и системата трябва да знае каква е тя;
- Така системата трябва да има информация за такова множество от подходящи имена на атрибути, че да ни разреши да цитираме тези имена в последващи операции;
- Напр., не можем да запишем израза

**(LECTURER join FACULTY)
where L_NUM > 30**

ако не знаем, че в резултата от израза

(LECTURER join FACULTY)

има атрибут с име L_NUM

Т.е. това, от което се нуждаем, е едно множество от правила за наследяване на имената на атрибути, създадено в алгебрата – такова, че ако знаем имената на атрибутите на входа на който да е релационен оператор, ние можем да предскажем имената на атрибутите на изхода на тази операция.

Като подготвяща стъпка да постигнем тази цел въвеждаме един нов оператор RENAME:

- Преименува атрибути в специфицираната релация;
- Връща едно ново копие на релацията, като някои атрибути са зададени с други имена;
- Напр., FACULTY rename L# as L_NUM.

Класически релационни оператори

Множеството на традиционните оператори
включва:

- Обединение (UNION);
- Сечение (INTERSECTION);
- Разлика (DIFFERENCE);
- Произведение (CARTESIAN PRODUCT, CROSS-PRODUCT).

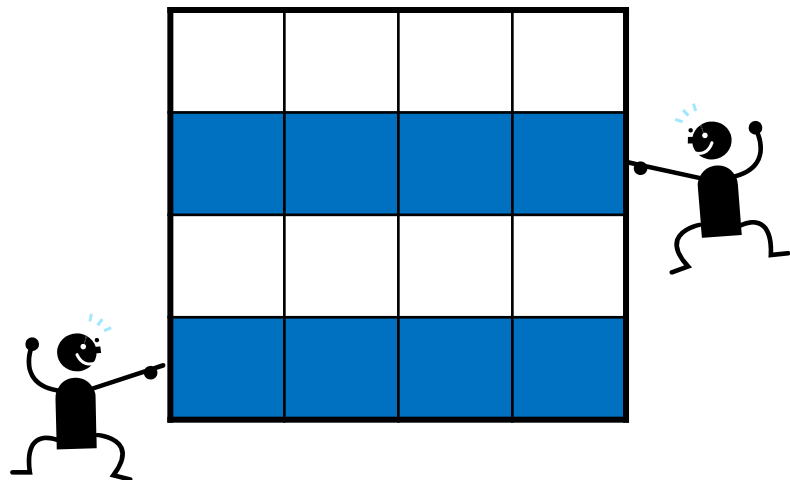
Всяка операция има два операнда, т.е. тези оператори са бинарни.

Специални релационни оператори

Това са операторите:

- Ограничение (избор) – **SELECTION (RESTRICTION)** – унарен;
- Проекция (**PROJECTION**) – унарен;
- Естествено сливане (**JOIN**) – бинарен;
- Частно (**DIVISION**) – бинарен.

SELECTION



SELECT (RESTRICT) – връща релация, която съдържа всички записи, отговарящи на специфицираните условия (предикати).

Операторът RESTRICT връща хоризонтално подмножество от релацията, т.е. редовете, които имат стойности на атрибутите си, отговарящи на зададените условия.

SELECTION

Нека Θ (тита) представлява някой валиден оператор за сравнение ($=$, \neq , $<$, $>$...). Тита-избор на релацията R върху атрибутите X и Y

$R \text{ WHERE } R.X \Theta R.Y$

е релация със същата заглавна част като на R и тяло множеството на всички записи t на R така, че сравнението **$t.X \Theta t.Y$** е вярно за всички t (атрибутите X и Y са дефинирани върху един и същ домейн).

На мястото на всеки един атрибут може да се специфицира константна стойност.

$R \text{ WHERE } R.X \Theta \text{constant}$

По този начин тита-избор операторът произвежда “хоризонтално” подмножество на дадена релация, състоящо се от записите, отговарящи на специфицираното сравнение.

SELECTION - пример

Нека имаме следната релация R, съдържаща преподаватели от различни факултети:

R

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

R WHERE FACULTY = 'ФМИ'

FName	LName	Title	Faculty
Стоян	Колев	Проф.	ФМИ
Спас	Петров	Доц.	ФМИ

R WHERE FACULTY = 'ФМИ' AND TITLE = 'Проф.'

FName	LName	Title	Faculty
Стоян	Колев	Проф.	ФМИ

SELECTION - пример

⊞: >

SQL Window - select employee_id, first_name...

SQL Output Statistics

```
select employee_id, first_name, salary
from employees
where salary > 11000
```

	EMPLOYEE_ID	FIRST_NAME	SALARY
1	100	Steven	24000,00
2	101	Neena	17000,00
3	102	Lex	17000,00
4	108	Nancy	12008,00
5	145	John	14000,00
6	146	Karen	13500,00
7	147	Alberto	12000,00
8	168	Lisa	11500,00
9	201	Michael	13000,00
10	205	Shelley	12008,00

4:21 10 rows selected in 0,031 seconds

⊞: =

SQLQuery2.sql - G...rthwind (sa (53))*

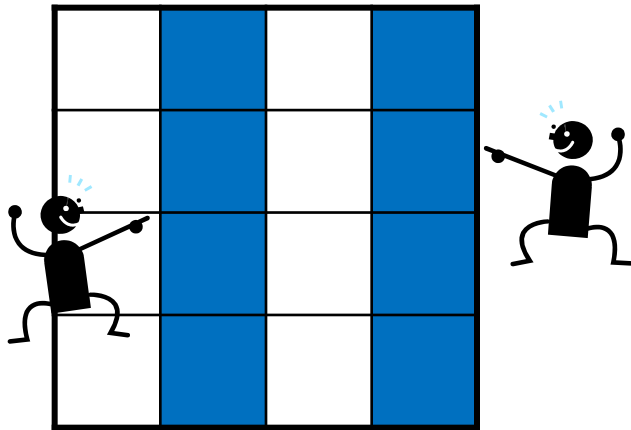
```
SELECT CustomerID, CompanyName, Address, Country
FROM Customers
WHERE Country = 'Mexico'
```

Results Messages

	CustomerID	CompanyName	Address	Country
1	ANATR	Ana Trujillo Emparedados y helados	Avda. de la Constitución 2222	Mexico
2	ANTON	Antonio Moreno Taquería	Mataderos 2312	Mexico
3	CENTC	Centro comercial Moctezuma	Sierras de Granada 9993	Mexico
4	PERIC	Pericles Comidas clásicas	Calle Dr. Jorge Cash 321	Mexico
5	TORTU	Tortuga Restaurante	Avda. Azteca 123	Mexico

Query... GCHOLAKOV-UNIPC\SQLEXPRESS ... sa (53) Northwind 00:00:00 5 rows

PROJECTION



PROJECT - връща релация,
която съдържа всички записи (без
дубликати) със специфицирани
атрибути от дадена релация.

Проекцията на релацията R върху атрибутите X, Y, \dots, Z

$R [X, Y, \dots, Z]$

е релация със заглавна част $\{X, Y, \dots, Z\}$ и тяло множеството от всички записи $\{X:x, Y:y, \dots, Z:z\}$ така, че един запис t се появява в R с X -стойност x , Y -стойност y , ..., Z -стойност z .

По този начин проекцията създава “вертикално” подмножество на дадена релация – т.е. това подмножество се получава чрез избор на специфични атрибути в специфициран ред.

Проекцията предоставя начин за пренареждане на атрибутите в една релация.

PROJECTION - пример

Нека имаме следната релация R, съдържаща преподаватели от различни факултети:

R

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

R [FName, LName]

FName	LName
Стоян	Колев
Петър	Иванов
Венета	Георгиева
Спас	Петров

R [Faculty, Title, FName]

Faculty	Title	FName
ФМИ	Проф.	Стоян
Право	Доц.	Петър
Икономика	Ст.н.с.	Венета
ФМИ	Доц.	Спас

PROJECTION - пример

SQL Window - select last_name, first_na...

SQL Output Statistics

```
select last_name, first_name
from employees
```

	LAST_NAME	FIRST_NAME
1	Abel	Ellen
2	Ande	Sundar
3	Atkinson	Mozhe
4	Austin	David
5	Baer	Hermann
6	Baida	Shelli
7	Banda	Amit
8	Bates	Elizabeth
9	Bell	Sarah
10	Bernstein	David
11	Bissot	Laura
12	Bloom	Harrison
13	Bull	Alexis
14	Cabrio	Anthony
15	Cambrault	Gerald
16	Cambrault	Nanette
17	Chen	John

1:20 107 rows selected in 0,125 second

SQLQuery2.sql - G...rthwind (sa (53))*

```
SELECT Country, CompanyName, Phone, ContactName
FROM Customers
```

Results Messages

	Country	CompanyName	Phone	ContactName
1	Germany	Alfreds Futterkiste	030-0074321	Maria Anders
2	Mexico	Ana Trujillo Emparedados y helados	(5) 555-4729	Ana Trujillo
3	Mexico	Antonio Moreno Taquería	(5) 555-3932	Antonio Moreno
4	UK	Around the Horn	(171) 555-7788	Thomas Hardy
5	Sweden	Berglunds snabbköp	0921-12 34 65	Christina Berglund
6	Germany	Blauer See Delikatessen	0621-08460	Hanna Moos
7	France	Blondesddsl père et fils	88.60.15.31	Frédérique Citeaux
8	Spain	Bólido Comidas preparadas	(91) 555 22 82	Martín Sommer
9	France	Bon app'	91.24.45.40	Laurence Leblan
10	Canada	Bottom-Dollar Markets	(604) 555-4729	Elizabeth Lincoln
11	UK	B's Beverages	(171) 555-1212	Victoria Ashworth
12	Argentina	Cactus Comidas para llevar	(1) 135-5555	Patricio Simpson
13	Mexico	Centro comercial Moctezuma	(5) 555-3392	Francisco Chang
14	Switzerland	Chop-suey Chinese	0452-076545	Yang Wang
15	Brazil	Comércio Mineiro	(11) 555-7647	Pedro Afonso

Query exe... GCHOLAKOV-UNIPC\SQL EXPRESS ... sa (53) Northwind 00:00:00 91 rows

PROJECTION

Забележки:

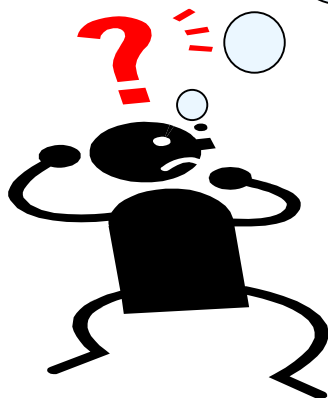
- Не можем да специфицираме един атрибут повече от веднъж;
- Когато всички атрибути на релацията бъдат указани, проекцията се нарича *идентична*;
- Проекция във формат $R []$ се нарича *нулева проекция*.

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.

Могат ли да се появят
някои проблеми при
обединението на две
релации?



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

Обединение

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

Резултат:

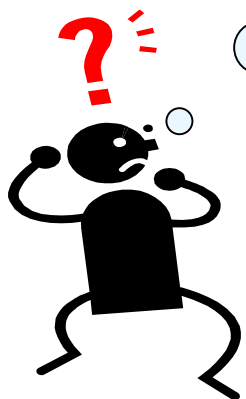
?	?	?	?
Стоян	Колев	Проф.	ФМИ
Петър	Иванов	Доц.	Право
Изкуствен интелект	20	424	
Венета	Георгиева	Ст.н.с.	Икономика
Спас	Петров	Доц.	ФМИ
Изкуствен интелект	20	424	

- Въпреки, че резултатът е множество от 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.

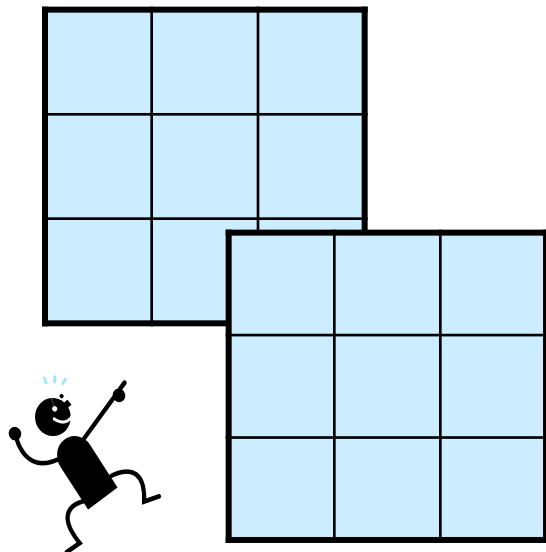
Кои оператори
изискват съвместимост
на типове?



Съвместимост по типове се изисква за операторите:

- ▣ Обединение
- ▣ Сечение
- ▣ Разлика

UNION



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 1

Нека имаме следните релации А и В, съдържащи преподавателите във факултети на ПУ и на СУ:

A

Name	Faculty
Стоян Колев	ФМИ
Петър Иванов	Филология
Венета Георгиева	Право

B

Name	Faculty
Десислава Янева	ФМИ
Стоян Колев	ФМИ

A UNION B

Name	Faculty
Стоян Колев	ФМИ
Петър Иванов	Филология
Десислава Янева	ФМИ
Венета Георгиева	Право

UNION - пример 2

Нека имаме следните релации EMPLOYEES и CUSTOMERS, съдържащи служители и клиенти на дадена компания.

SQL Window - Query data of table OE.CUSTOMERS@ORCL

SQL Output Statistics

```
select * from oe.customers t
```

	CUSTOMER_ID	CUST_FIRST_NAME	CUST_LAST_NAME
1	147	Ishwarya	Roberts
2	148	Gustav	Steenburgen
3	149	Markus	Rampling
4	150	Goldie	Slater
5	151	Divine	Aykroyd
6	152	Dieter	Matthau
7	153	Divine	Sheen
8	154	Frederic	Grodin
9	155	Frederico	Romero
10	156	Goldie	Mustard

1:29 319 rows selected in 2,668 seconds

SQL Window - Query data of table HR.EMPLOYEES@ORCL

SQL Output Statistics

```
select * from hr.employees t
```

	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL
1	198	Donald	OConnell	DOCONNEL
2	199	Douglas	Grant	DGRANT
3	200	Jennifer	Whalen	JWHALEN
4	201	Michael	Hartstein	MHARTSTE
5	202	Pat	Fay	PFAY
6	203	Susan	Mavris	SMAVRIS
7	204	Hermann	Baer	HBAER
8	205	Shelley	Higgins	SHIGGINS
9	206	William	Gietz	WGIFTZ

1:29 107 rows selected in 0,171 seconds

UNION - пример 2

CUSTOMERS UNION EMPLOYEES

SQL Window - select c.customer_id as id, c.cust_first_name as fname, c.cust_last_name from oe.customers c uni ...

SQL Output Statistics

```
select c.customer_id as id, c.cust_first_name as fname, c.cust_last_name
from oe.customers c
union
select e.employee_id, e.first_name, e.last_name
from hr.employees e
```

	ID	FNAME	CUST_LAST_NAME
1	100	Steven	King
2	101	Constantin	Welles
3	101	Neena	Kochhar
4	102	Harrison	Pacino
5	102	Lex	De Haan
6	103	Alexander	Hunold
7	103	Manisha	Taylor
8	104	Bruce	Ernst
9	104	Harrison	Sutherland
10	105	David	Austin
11	105	Matthias	MacGraw
12	106	Matthias	Hannah
13	106	Valli	Pataballa
14	107	Diana	Lorentz

2:20 426 rows selected in 0,249 seconds

UNION ALL - пример 3

CUSTOMERS UNION EMPLOYEES

SQL Window - select c.cust_first_name as fname fr...

SQL Output Statistics

```
select c.cust_first_name as fname
from oe.customers c
union
select e.first_name
from hr.employees e
order by 1
```

	FNAME
1	Adam
2	Ajay
3	Alain
4	Alan
5	Alana
6	Albert
7	Alberto
8	Alec
9	Alexander
10	Alexis

6:11 248 rows selected in 0,14 seconds

CUSTOMERS UNION ALL EMPLOYEES

SQL Window - select c.cust_first_name as fname fr...

SQL Output Statistics

```
select c.cust_first_name as fname
from oe.customers c
union all
select e.first_name
from hr.employees e
order by 1
```

	FNAME
1	Adam
2	Ajay
3	Ajay
4	Alain
5	Alain
6	Alain
7	Alan
8	Alan
9	Alana
10	Albert

6:11 426 rows selected in 0,203 seconds

UNION ALL - пример с T-SQL

The screenshot shows a SQL Server Enterprise Manager window titled "SQLQuery2.sql - G...rthwind (sa (53))*". The query editor contains the following T-SQL code:

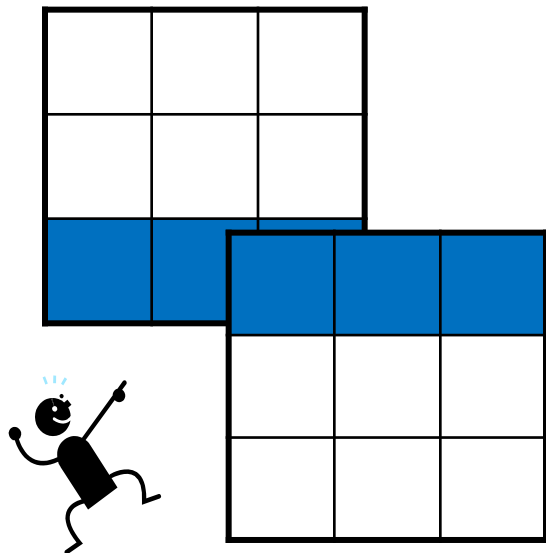
```
SELECT Country FROM Customers  
UNION ALL  
SELECT Country FROM Employees
```

Below the query editor, the "Results" tab is active, displaying a table with 15 rows of data. The table has a single column named "Country". The data is as follows:

	Country
1	Germany
2	Mexico
3	Mexico
4	UK
5	Sweden
6	Germany
7	France
8	Spain
9	France
10	Canada
11	UK
12	Argentina
13	Mexico
14	Switzerland
15	Brazil

The status bar at the bottom of the window indicates: "Query ex... | GCHOLAKOV-UNIPC\SQLEXPRESS ... | sa (53) | Northwind | 00:00:00 | 100 rows".

INTERSECTION



INTERSECTION – създава нова релация, състояща се от всички записи, 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).

INTERSECTION – пример 1

Нека имаме следните релации A и B, съдържащи преподавателите във факултетите на ПУ и на СУ:

A

Name	Faculty
Стоян Колев	ФМИ
Петър Иванов	Филология
Венета Георгиева	Право

B

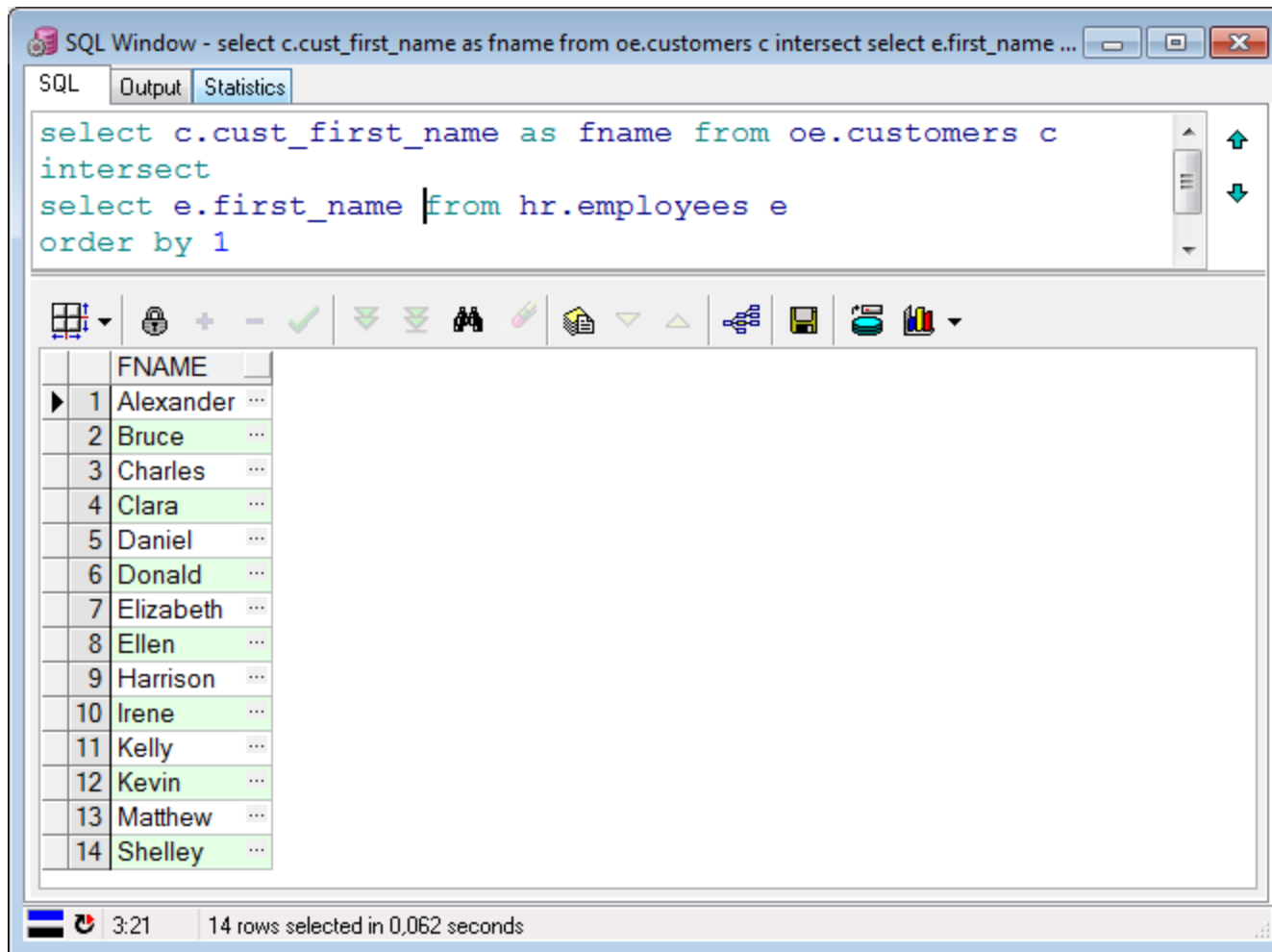
Name	Faculty
Десислава Янева	ФМИ
Стоян Колев	ФМИ

A INTERSECTION B

Name	Faculty
Стоян Колев	ФМИ

INTERSECTION - пример 2

CUSTOMERS INTERSECT EMPLOYEES



The screenshot shows an SQL Window titled "SQL Window - select c.cust_first_name as fname from oe.customers c intersect select e.first_name ...". The window has three tabs: "SQL", "Output", and "Statistics". The "SQL" tab is active, displaying the following query:

```
select c.cust_first_name as fname from oe.customers c
intersect
select e.first_name from hr.employees e
order by 1
```

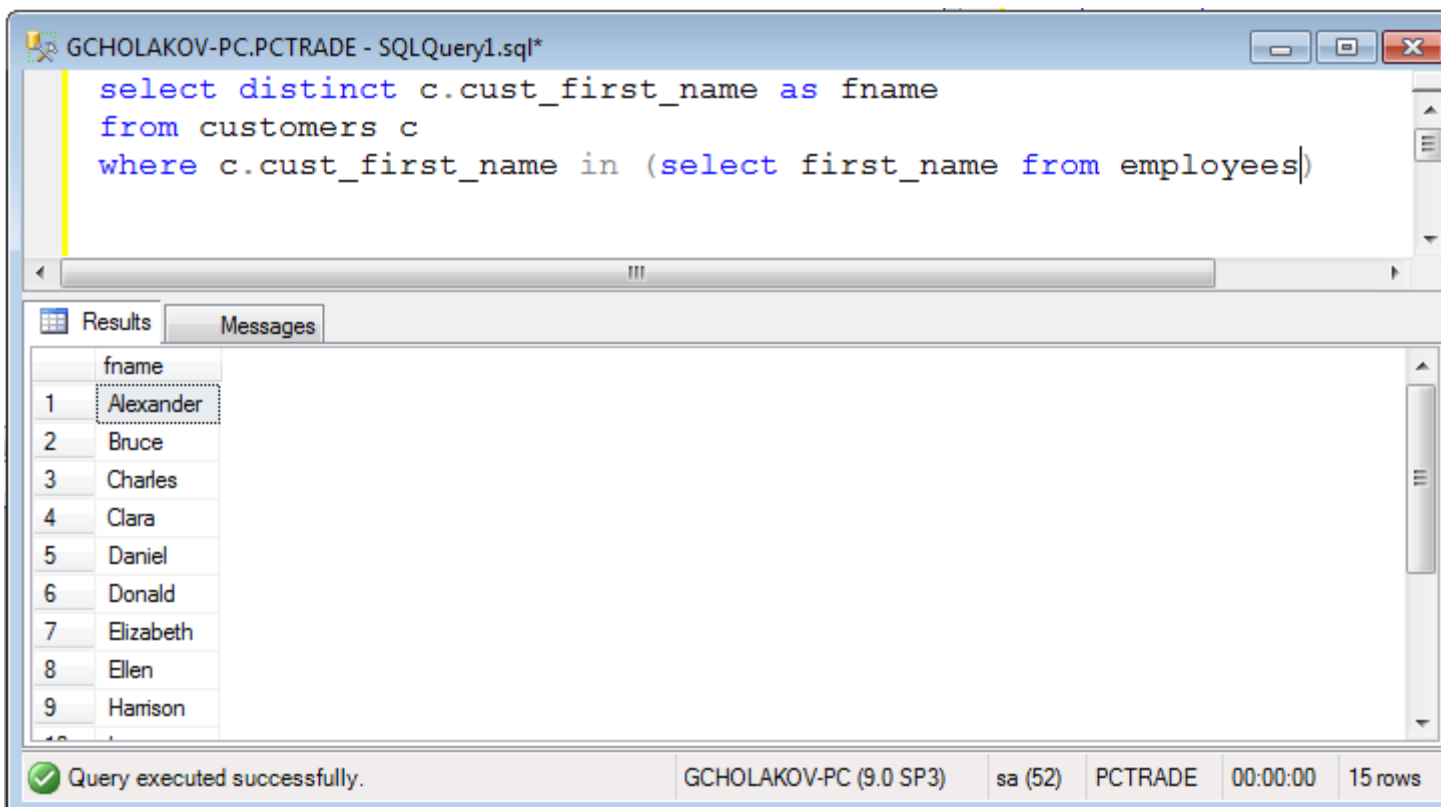
Below the query editor is a toolbar with various icons for database operations. The results are displayed in a table with the following data:

	FNAME
1	Alexander
2	Bruce
3	Charles
4	Clara
5	Daniel
6	Donald
7	Elizabeth
8	Ellen
9	Harrison
10	Irene
11	Kelly
12	Kevin
13	Matthew
14	Shelley

The status bar at the bottom indicates "3:21" and "14 rows selected in 0,062 seconds".

INTERSECTION – пример 3

Някои СУБД не поддържат оператор INTERSECT. При тях може да се използва алтернативен код.



The screenshot shows a SQL Server Enterprise Manager window titled "GCHOLAKOV-PC.PCTRADE - SQLQuery1.sql*". The query window contains the following SQL code:

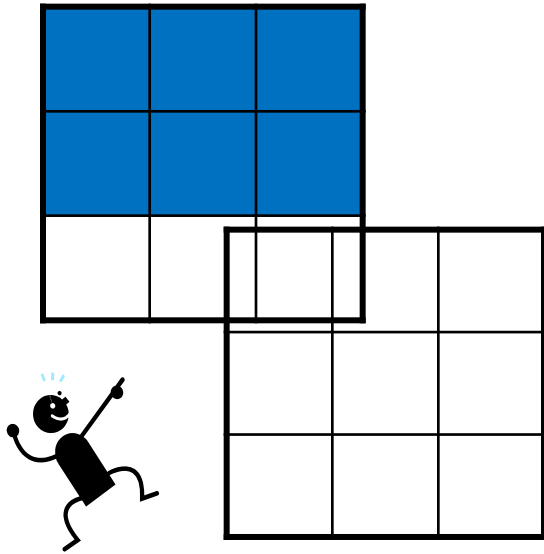
```
select distinct c.cust_first_name as fname
from customers c
where c.cust_first_name in (select first_name from employees)
```

Below the query window, the "Results" tab is active, displaying a table with the following data:

	fname
1	Alexander
2	Bruce
3	Charles
4	Clara
5	Daniel
6	Donald
7	Elizabeth
8	Ellen
9	Harrison

The status bar at the bottom indicates: "Query executed successfully. GCHOLAKOV-PC (9.0 SP3) sa (52) PCTRADE 00:00:00 15 rows".

DIFFERENCE (MINUS)



DIFFERENCE – създава релация, състояща се от всички записи, които се появяват в първата, но не и във втората релация.

Разликата 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 и не принадлежащи на B.

DIFFERENCE – пример 1

Нека имаме следните релации A и B, съдържащи преподавателите във факултетите на ПУ и на СУ:

A

Name	Faculty
Стоян Колев	ФМИ
Петър Иванов	Филология
Венета Георгиева	Право

B

Name	Faculty
Десислава Янева	ФМИ
Стоян Колев	ФМИ

A MINUS B

Name	Faculty
Венета Георгиева	Право
Петър Иванов	Филология

B MINUS A

Name	Faculty
Десислава Янева	ФМИ

DIFFERENCE - пример 2

CUSTOMERS MINUS EMPLOYEES

SQL Window - select c.cust_first_name as fname from...

SQL Output Statistics

```
select c.cust_first_name as fname
from oe.customers c
minus
select e.first_name
from hr.employees e
```

	FNAME
1	Ajay
2	Alain
3	Alan
4	Albert
5	Alec
6	Alfred
7	Ali
8	Alice
9	Ally
10	Alonso
11	Amanda
12	Amrish
13	Ben
14	Billy

5:20 157 rows selected in 0,109 seconds

EMPLOYEES MINUS CUSTOMERS

SQL Window - select e.first_name from hr.employees...

SQL Output Statistics

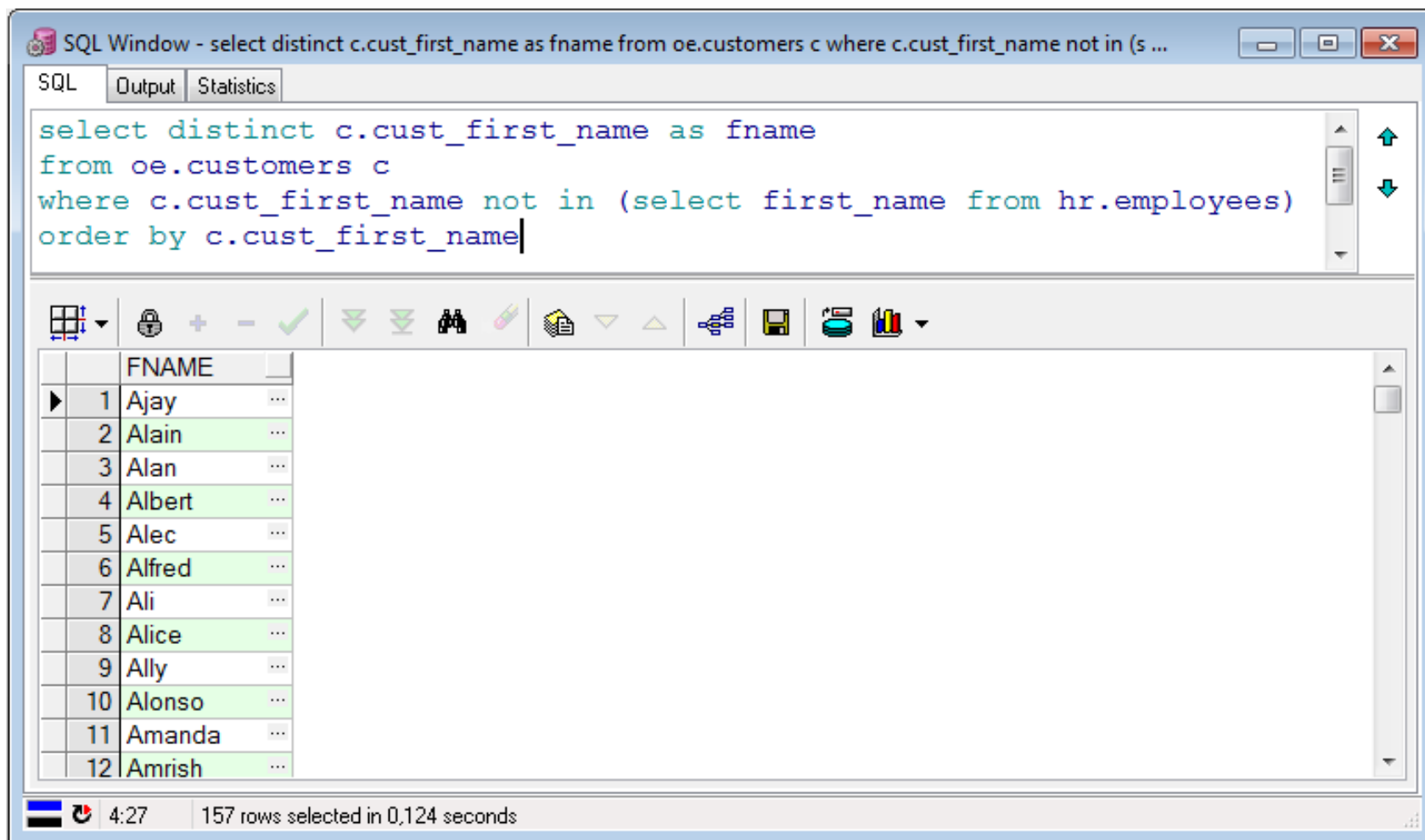
```
select e.first_name
from hr.employees e
minus
select c.cust_first_name as fname
from oe.customers c
```

	FIRST_NAME
1	Adam
2	Alana
3	Alberto
4	Alexis
5	Allan
6	Alyssa
7	Amit
8	Anthony
9	Britney
10	Christopher
11	Curtis
12	Danielle
13	David
14	Don

5:20 77 rows selected in 0,109 seconds

DIFFERENCE - пример 3

Пример за алтернативен код:



The screenshot shows an SQL Window with the following title bar: "SQL Window - select distinct c.cust_first_name as fname from oe.customers c where c.cust_first_name not in (s ...". The window has three tabs: "SQL", "Output", and "Statistics". The "SQL" tab is active, displaying the following query:

```
select distinct c.cust_first_name as fname
from oe.customers c
where c.cust_first_name not in (select first_name from hr.employees)
order by c.cust_first_name
```

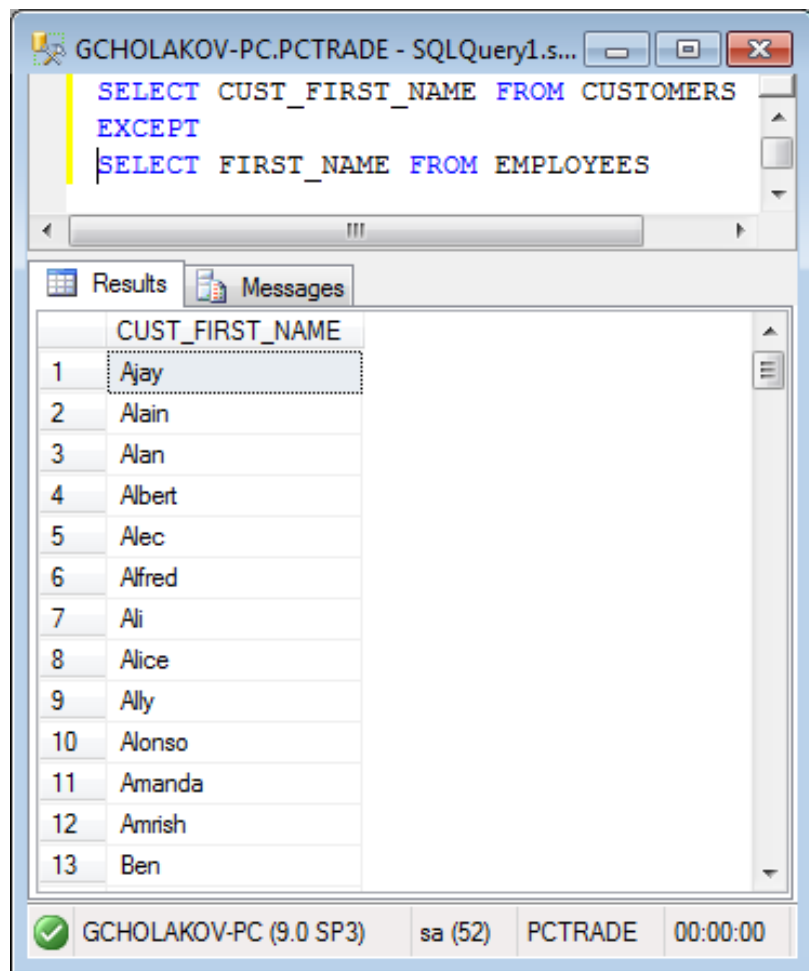
Below the query editor is a toolbar with various icons for SQL operations. The "Output" tab is active, displaying a table with the following data:

	FNAME
1	Ajay
2	Alain
3	Alan
4	Albert
5	Alec
6	Alfred
7	Ali
8	Alice
9	Ally
10	Alonso
11	Amanda
12	Amrish

The status bar at the bottom of the window shows the time "4:27" and the message "157 rows selected in 0,124 seconds".

DIFFERENCE – пример T-SQL

Някои СУБД не поддържат оператор MINUS или използват друга ключова дума – EXCEPT в T-SQL:



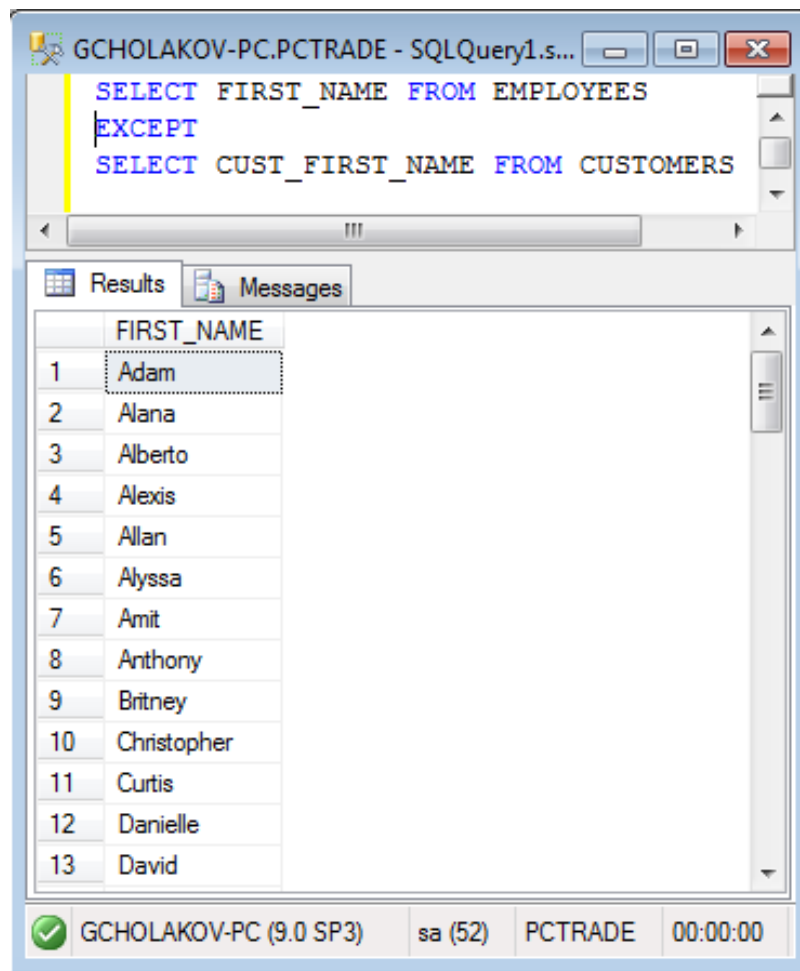
The screenshot shows a SQL Server Enterprise Manager window titled "GCHOLAKOV-PC.PCTRADE - SQLQuery1.s...". The query editor contains the following T-SQL code:

```
SELECT CUST_FIRST_NAME FROM CUSTOMERS  
EXCEPT  
SELECT FIRST_NAME FROM EMPLOYEES
```

The Results tab is active, displaying a table with the following data:

	CUST_FIRST_NAME
1	Ajay
2	Alain
3	Alan
4	Albert
5	Alec
6	Alfred
7	Ali
8	Alice
9	Ally
10	Alonso
11	Amanda
12	Amrish
13	Ben

The status bar at the bottom indicates: GCHOLAKOV-PC (9.0 SP3) sa (52) PCTRADE 00:00:00.



The screenshot shows a SQL Server Enterprise Manager window titled "GCHOLAKOV-PC.PCTRADE - SQLQuery1.s...". The query editor contains the following T-SQL code:

```
SELECT FIRST_NAME FROM EMPLOYEES  
EXCEPT  
SELECT CUST_FIRST_NAME FROM CUSTOMERS
```

The Results tab is active, displaying a table with the following data:

	FIRST_NAME
1	Adam
2	Alana
3	Alberto
4	Alexis
5	Allan
6	Alyssa
7	Amit
8	Anthony
9	Britney
10	Christopher
11	Curtis
12	Danielle
13	David

The status bar at the bottom indicates: GCHOLAKOV-PC (9.0 SP3) sa (52) PCTRADE 00:00:00.

SELECTION - ревизирана

Дефинираната по-рано операция тита-избор (избор) разрешава само прости сравнение в WHERE клаузата, но е възможно дефиницията да се разшири с описаните след това оператори, както е показано чрез следните равенства:

- **R WHERE c1 AND c2**

е дефинирана като еквивалентна на

(R WHERE c1) INTERSECTION (R WHERE c2)

- **R WHERE c1 OR c2**

е еквивалентна на

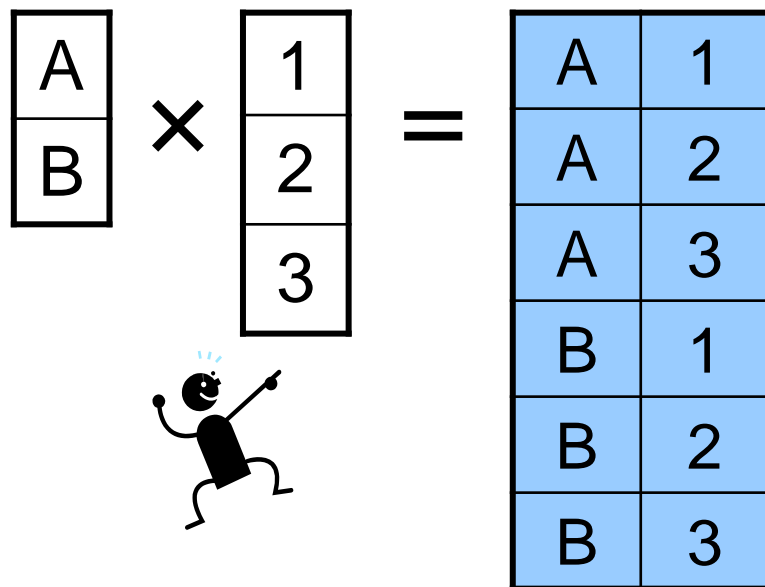
(R WHERE c1) UNION (R WHERE c2)

- **R WHERE NOT c**

е еквивалентна на

R MINUS (R WHERE c)

PRODUCT



PRODUCT - създава нова релация от две дадени релации, която се състои от всички възможни конкатенирани двойки от записи от двете релации.

Произведението 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 обединението на всеки запис от A с всеки от B.

PRODUCT

Произведението на две релации често се нарича Декартово, защото резултатът е релация с тяло множеството на всички подредени двойки записи от двете релации.

Но за да се запази свойството затвореност, т.е. резултатът от операцията отново да е релация, трябва резултатната релация да съдържа записи, а не подредени двойки от записи.

В релационната алгебра Декартовото произведение е една разширена форма на операцията, където всяка подредена двойка от записи се заменя от отделен запис, който представлява обединение (конкатенация) на двата записа.

Т.е., ако са дадени два записа $\{A_1:a_1, \dots, A_m:a_m\}$ и $\{B_1:b_1, \dots, B_n:b_n\}$ обединението има вида $\{A_1:a_1, \dots, A_m:a_m, B_1:b_1, \dots, B_n:b_n\}$

Степен на резултата – сумата от степените на A и B.

Кардиналност на резултата – произведението от кардиналностите на A и B.

PRODUCT – пример 1

Нека имаме следните релации A и B, съдържащи съответно преподаватели и университети, в които те преподават:

A

UNI_ID	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

UNI_ID	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

PRODUCT - пример 2

SQL Window - Query data of tab...

SQL Output Statistics

```
select * from regions t
```

	REGION_ID	REGION_NAME
1	1	Europe
2	2	Americas
3	3	Asia
4	4	Middle East and Africa

1:1 region_id, number, manda

×

SQL Window - select * from countries t

SQL Output Statistics

```
select * from countries t
```

	COUNTRY_ID	COUNTRY_NAME	REGION_ID
1	AR	Argentina	2
2	AU	Australia	3
3	BE	Belgium	1
4	BR	Brazil	2
5	CA	Canada	2
6	CH	Switzerland	1
7	CN	China	3
8	DE	Germany	1
9	DK	Denmark	1
10	EG	Egypt	4
11	FR	France	1
12	IL	Israel	4
13	IN	India	3
14	IT	Italy	1
15	JP	Japan	3
16	KW	Kuwait	4

1:15 25 rows selected in 0,016 seconds

PRODUCT - пример 2

=

SQL Window - select * from countries, regions order by country_id

SQL Output Statistics

```
select *  
from countries, regions  
order by country_id
```

	COUNTRY_ID	COUNTRY_NAME	REGION_ID	REGION_ID	REGION_NAME
1	AR	Argentina	2	1	Europe
2	AR	Argentina	2	2	Americas
3	AR	Argentina	2	3	Asia
4	AR	Argentina	2	4	Middle East and Africa
5	AU	Australia	3	1	Europe
6	AU	Australia	3	4	Middle East and Africa
7	AU	Australia	3	3	Asia
8	AU	Australia	3	2	Americas
9	BE	Belgium	1	4	Middle East and Africa
10	BE	Belgium	1	2	Americas
11	BE	Belgium	1	3	Asia
12	BE	Belgium	1	1	Europe
13	BR	Brazil	2	3	Asia
14	BR	Brazil	2	4	Middle East and Africa
15	BR	Brazil	2	1	Europe
16	BR	Brazil	2	2	Americas
17	CA	Canada	2	4	Middle East and Africa
18	CA	Canada	2	1	Europe
19	CA	Canada	2	2	Americas
20	CA	Canada	2	3	Asia

3:20 100 rows selected in 0,063 seconds

PRODUCT

Забележки:

- Възможно е да възникне проблем – ако заглавните части на двете релации имат атрибути с еднакви имена – тогава трябва да използваме оператора **RENAME** за преименуване на тези атрибути;
- От примера се вижда, че резултатната релация съдържа доста неверни данни и ще се наложи да се изпълнят допълнителни операции, за да се извлекат само значещите данни.

Алгебрични свойства на релационните оператори

Асоциативност:

- UNION (обединение)

$$(A \text{ UNION } B) \text{ UNION } C \Leftrightarrow A \text{ UNION } (B \text{ UNION } C)$$

- INTERSECTION (сечение)

$$(A \text{ INTERSECT } B) \text{ INTERSECT } C \Leftrightarrow A \text{ INTERSECT } (B \text{ INTERSECT } C)$$

- PRODUCT (произведение)

$$(A \text{ TIMES } B) \text{ TIMES } C \Leftrightarrow A \text{ TIMES } (B \text{ TIMES } C)$$

Комутативност:

- UNION (обединение) : $A \text{ UNION } B = B \text{ UNION } A$
- INTERSECTION (сечение) : $A \text{ INTERSECT } B = B \text{ INTERSECT } A$
- PRODUCT (произведение) : $A \text{ TIMES } B \Leftrightarrow B \text{ TIMES } A$

JOIN

Операцията join е една от най-полезните в релационната алгебра и е най-често използвания способ за комбиниране на данни от две или повече релации.

Въпреки че join операцията може да бъде представена като Декартово произведение, последвано от операциите selection и projection, нейното използване в практиката се среща много често.

Още повече, резултатът от Декартовото произведение обикновено е много по-голям от резултата, получен от join. Но все пак относно въпроси, касаещи ефективността на извличане на данни, е трудно да се отговори еднозначно кой от двата подхода би бил по-ефективен – това зависи много от конкретната приложна област, т.е. от данните, техните типове, наличието на индекси и т.н.

JOIN

Ще разгледаме следните видове JOIN операции:

- Natural join
- Theta (Θ) join
- Equi-join
- Semi-join
- Anti-join
- Outer join

NATURAL JOIN



C	A
C1	A1
C2	A2
C3	A3

A	B
A1	B1
A2	B2
A3	B3

=

A	B	C
A1	B1	C1
A2	B2	C2
A3	B3	C3



NATURAL JOIN (естествено сливане) - създава нова релация от двете дадени релации, която съдържа всички възможни свързани двойки от записи, (по един от всяка релация), така че всяка двойка да има равенство на стойностите във всички атрибути с еднакви имена от двете релации.

NATURAL JOIN

Нека са дадени релациите A и B със заглавни части $\{X_1, \dots, X_m, Y_1, \dots, Y_n\}$ и $\{Y_1, \dots, Y_n, Z_1, \dots, Z_p\}$, т.е. атрибутите $\{Y_1, \dots, Y_n\}$ са общи за двете релации. Допускаме, че общите атрибути са дефинирани върху общи домейни.

По-нататък ще разглеждаме:

- $\{X_1, \dots, X_m\}$ – като X ;
- $\{Y_1, \dots, Y_n\}$ – като Y ;
- $\{Z_1, \dots, Z_p\}$ – като Z .

NATURAL JOIN

Естественото сливане ($A \text{ NATURAL JOIN } B$) е релация със:

- ✓ заглавна част $\{X, Y, Z\}$;
- ✓ тяло – множеството на всички записи $\{X:x, Y:y, Z:z\}$ с равенство в общите атрибути така, че един запис се появява в A с X -стойностите и с Y -стойностите, а в B – с Y -стойностите и Z -стойностите.

JOIN операторът е:

- ✓ асоциативен:

$$(A \text{ JOIN } B) \text{ JOIN } C \Leftrightarrow A \text{ JOIN } (B \text{ JOIN } C) \Leftrightarrow A \text{ JOIN } B \text{ JOIN } C;$$

- ✓ комутативен: $A \text{ JOIN } B \Leftrightarrow B \text{ JOIN } A$

NATURAL JOIN - пример

SQL Window - select * from countries natural join regions order by country_id

SQL Output Statistics

```
select *  
from countries natural join regions  
order by country_id
```

	REGION_ID	COUNTRY_ID	COUNTRY_NAME	REGION_NAME
1	2	AR	Argentina	Americas
2	3	AU	Australia	Asia
3	1	BE	Belgium	Europe
4	2	BR	Brazil	Americas
5	2	CA	Canada	Americas
6	1	CH	Switzerland	Europe
7	3	CN	China	Asia
8	1	DE	Germany	Europe
9	1	DK	Denmark	Europe
10	4	EG	Egypt	Middle East and Africa
11	1	FR	France	Europe
12	4	IL	Israel	Middle East and Africa
13	3	IN	India	Asia

2:28 region_id, number, optional, Region ID for the country. Foreign key to region_id column in tr

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.

⊞-JOIN

Нека релациите A и B нямат общи атрибути и нека Θ е валиден оператор за сравнение ($>$, $<$, \geq , \leq , $<>$). Тогава Θ -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

SQL Window - select * from countries c join regions r on c.region_id = r.region_id order by 2

SQL Output Statistics

```
select *  
from countries c join regions r on c.region_id = r.region_id  
order by 2
```

	COUNTRY_ID	COUNTRY_NAME	REGION_ID	REGION_ID	REGION_NAME
1	AR	Argentina	...	2	Americas
2	AU	Australia	...	2	Americas
3	BE	Belgium	...	2	Americas
4	BR	Brazil	...	2	Americas
5	CA	Canada	...	2	Americas
6	CN	China	...	2	Americas
7	DK	Denmark	...	2	Americas
8	EG	Egypt	...	2	Americas
9	FR	France	...	2	Americas

3:11 25 rows

SQL Window - select * from countries c, regions r where c.region_id = r.region_id order by 2

SQL Output Statistics

```
select *  
from countries c, regions r  
where c.region_id = r.region_id  
order by 2
```

	COUNTRY_ID	COUNTRY_NAME	REGION_ID	REGION_ID	REGION_NAME
1	AR	Argentina	...	2	Americas
2	AU	Australia	...	3	Asia
3	BE	Belgium	...	1	Europe
4	BR	Brazil	...	2	Americas
5	CA	Canada	...	2	Americas
6	CN	China	...	3	Asia
7	DK	Denmark	...	1	Europe
8	EG	Egypt	...	4	Middle East and Africa
9	FR	France	...	1	Europe

4:11 25 rows selected in 0,047 seconds

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

Да предположим, че имаме таблиците DEPT и EMP, които съдържат отдели и служители и нека във всеки отдел има поне един служител.

SQL Window - Query data of tabl...

SQL Output Statistics

```
select * from dept
```

	DEPTNO	DNAME	LOC
1	10	ACCOUNTING	NEW YORK
2	20	RESEARCH	DALLAS
3	30	SALES	CHICAGO
4	40	OPERATIONS	BOSTON

1:15 loc, varchar2(13), optional

SQL Window - Query data of table SCOTT.EMP@ORCL

SQL Output Statistics

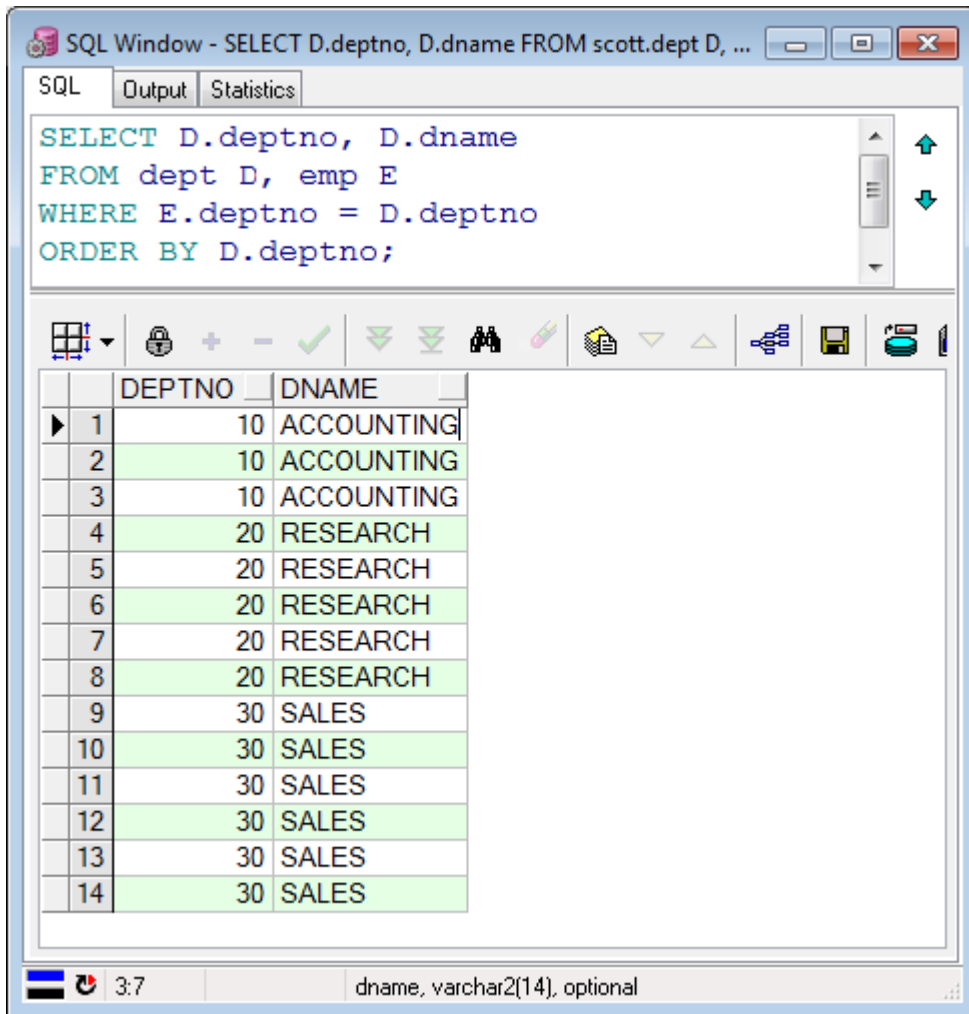
```
select * from emp
```

	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
1	7369	SMITH	CLERK	7902	17.12.1980 г.	800,00		20
2	7499	ALLEN	SALESMAN	7698	20.2.1981 г.	1600,00	300,00	30
3	7521	WARD	SALESMAN	7698	22.2.1981 г.	1250,00	500,00	30
4	7566	JONES	MANAGER	7839	2.4.1981 г.	2975,00		20
5	7654	MARTIN	SALESMAN	7698	28.9.1981 г.	1250,00	1400,00	30
6	7698	BLAKE	MANAGER	7839	1.5.1981 г.	2850,00		30
7	7782	CLARK	MANAGER	7839	9.6.1981 г.	2450,00		10
8	7788	SCOTT	ANALYST	7566	19.4.1987 г.	3000,00		20
9	7839	KING	PRESIDENT		17.11.1981 г.	5000,00		10
10	7844	TURNER	SALESMAN	7698	8.9.1981 г.	1500,00	0,00	30
11	7876	ADAMS	CLERK	7788	23.5.1987 г.	1100,00		20
12	7900	JAMES	CLERK	7698	3.12.1981 г.	950,00		30
13	7902	FORD	ANALYST	7566	3.12.1981 г.	3000,00		20
14	7934	MILLER	CLERK	7782	23.1.1982 г.	1300,00		10

1:18 comm, number(7,2), optional

SEMI-JOIN - пример

Да покажем всички отдели, които имат поне един служител.



SQL Window - SELECT D.deptno, D.dname FROM scott.dept D, ...

SQL Output Statistics

```
SELECT D.deptno, D.dname
FROM dept D, emp E
WHERE E.deptno = D.deptno
ORDER BY D.deptno;
```

	DEPTNO	DNAME
1	10	ACCOUNTING
2	10	ACCOUNTING
3	10	ACCOUNTING
4	20	RESEARCH
5	20	RESEARCH
6	20	RESEARCH
7	20	RESEARCH
8	20	RESEARCH
9	30	SALES
10	30	SALES
11	30	SALES
12	30	SALES
13	30	SALES
14	30	SALES

3:7 dname, varchar2(14), optional

Ясно се вижда, че редовете за отделите се повтарят толкова пъти, колкото служители има в отдел.

Също се вижда, че отдел с DEPTNO = 40 не участва в резултата, защото няма служители в този отдел.

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

Следната заявка ще извлече същите данни, но без дубликати и по-ефективно.

SQL Window - SELECT D.deptno, D.dname FROM scott.dept D ...

SQL Output Statistics

```
SELECT D.deptno, D.dname
FROM dept D
WHERE EXISTS (SELECT 1
              FROM emp E
              WHERE E.deptno = D.deptno)
```

	DEPTNO	DNAME
1	10	ACCOUNTING
2	20	RESEARCH
3	30	SALES

5:20 deptno, number(2), mandatory

Тук дали отделът има 1 или 1000 служителя няма значение, защото системата ще спира да търси служители за отдела още щом намери първия, вместо да търси всички служители.

SQL Window - SELECT D.deptno, D.dname FROM sc...

SQL Output Statistics

```
SELECT D.deptno, D.dname
FROM dept D
WHERE D.deptno IN
      (SELECT distinct deptno
       FROM emp)
```

	DEPTNO	DNAME
1	10	ACCOUNTING
2	20	RESEARCH
3	30	SALES

5:12 deptno, number(2), mandatory

В тази заявка се проверява стойността на всеки ред в deptno дали е измежду тези, върнати от вложената заявка.

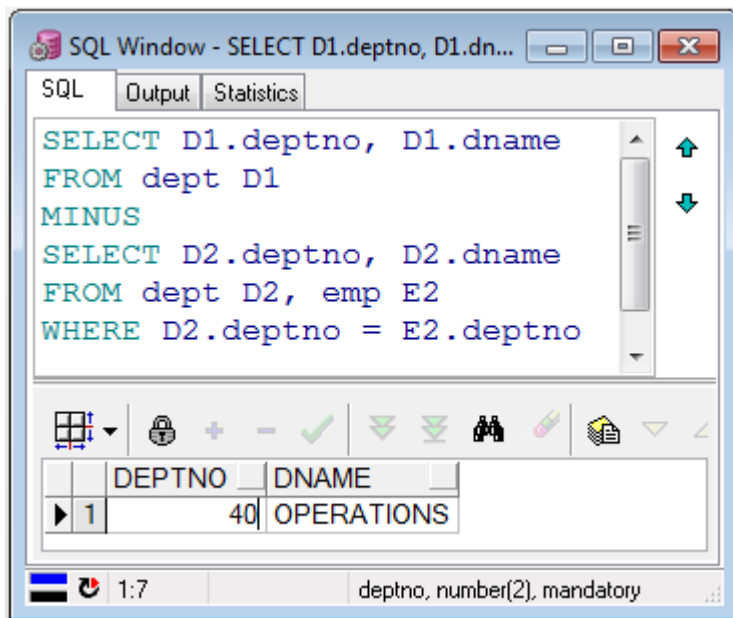
ANTI-JOIN

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

Следните две заявки извличат по двата начина отделите, които нямат служители.

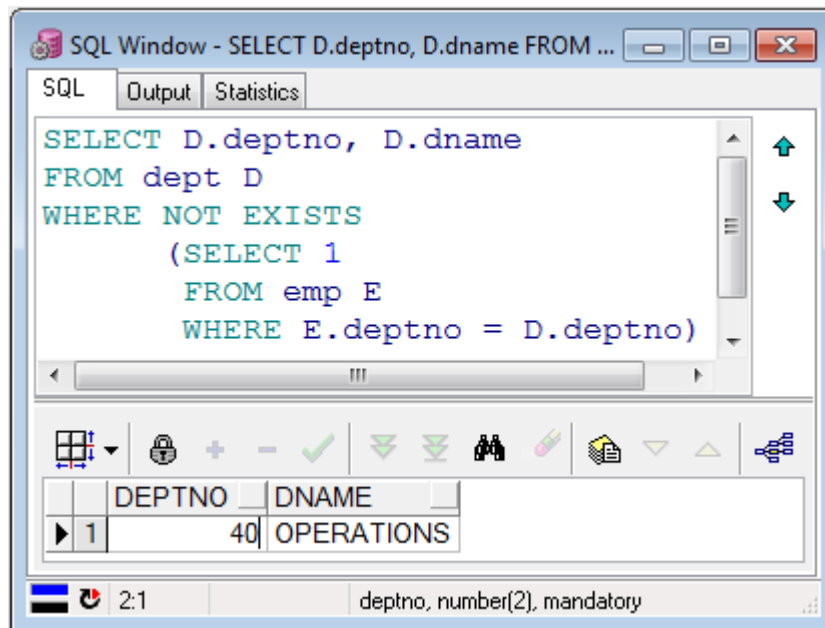


SQL Window - SELECT D1.deptno, D1.dname

```
SQL      Output  Statistics
SELECT D1.deptno, D1.dname
FROM dept D1
MINUS
SELECT D2.deptno, D2.dname
FROM dept D2, emp E2
WHERE D2.deptno = E2.deptno
```

	DEPTNO	DNAME
1	40	OPERATIONS

1:7 deptno, number(2), mandatory



SQL Window - SELECT D.deptno, D.dname FROM ...

```
SQL      Output  Statistics
SELECT D.deptno, D.dname
FROM dept D
WHERE NOT EXISTS
      (SELECT 1
       FROM emp E
       WHERE E.deptno = D.deptno)
```

	DEPTNO	DNAME
1	40	OPERATIONS

2:1 deptno, number(2), mandatory

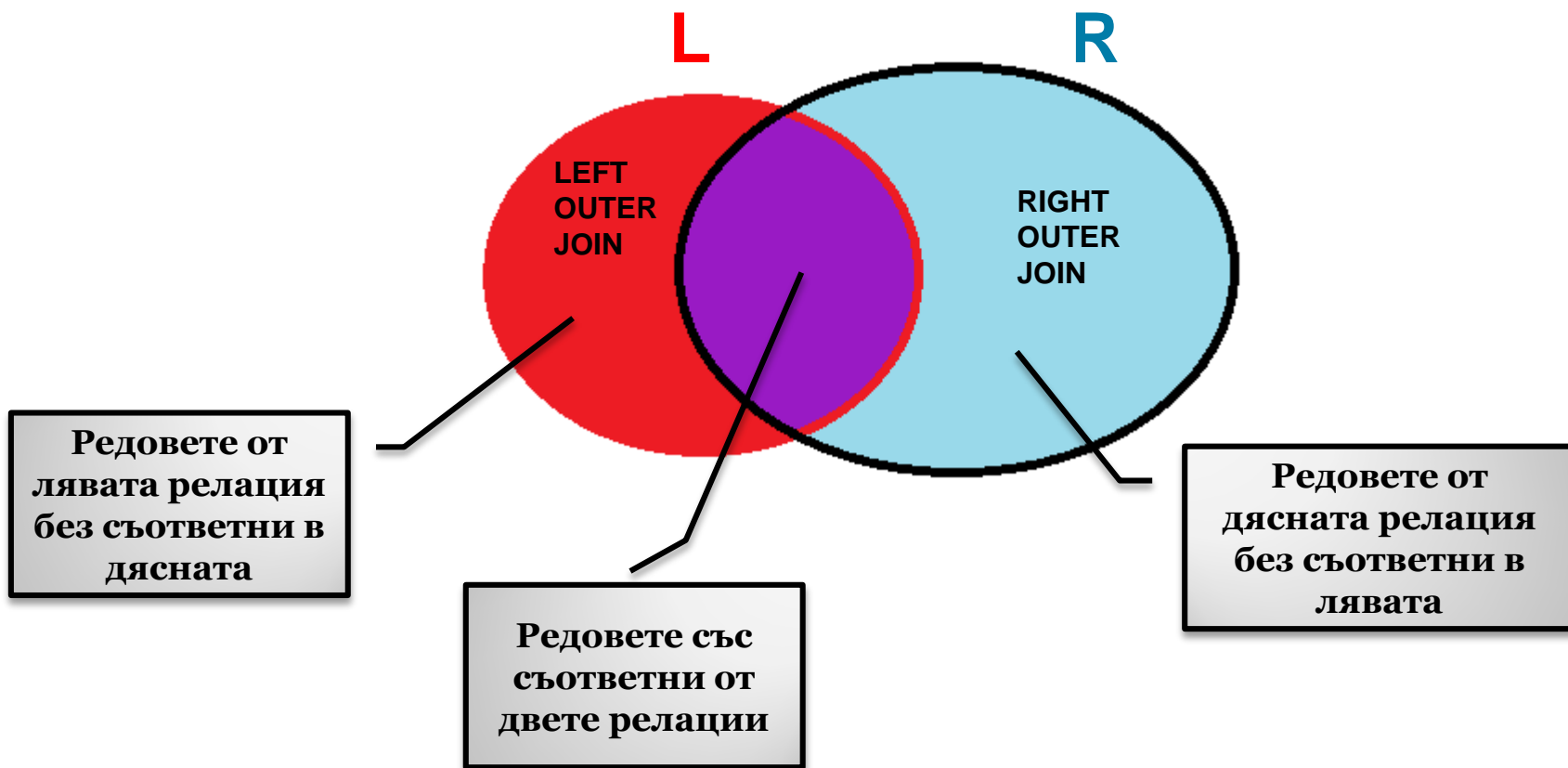
INNER JOIN

Изброените дотук видове join операции (natural join, Θ -join, equi join) реализират т.нар. вътрешни съединения, характерни с това, че в резултатът 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.

OUTER JOIN

Външното съединение генерира релация, в която записите, които нямат съвпадения в двете релации, могат също да бъдат запазени в резултата. Нека за примерите използваме релацията L като лява, а R като дясна.



OUTER JOIN

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

SQL Window - select * from countries c left join regions r on c.region_id = r.region_id order by 2

SQL Output Statistics

```
select *  
from countries c left outer join regions r  
on c.region_id = r.region_id  
order by 2
```

	COUNTRY_ID	COUNTRY_NAME	REGION_ID	REGION_ID	REGION_NAME
1	AR	Argentina	...	2	Americas
2	AU	Australia	...	3	Asia
3	BE	Belgium	...	1	Europe
4	BR	Brazil	...	2	Americas
5	BG	Bulgaria	...		
6	CA	Canada	...	2	Americas
7	CN	China	...	3	Asia
8	DK	Denmark	...	1	Europe
9	EG	Egypt	...	4	Middle East and Africa
10	FR	France	...	1	Europe
11	DE	Germany	...	1	Europe
12	IN	India	...	3	Asia
13	IL	Israel	...	4	Middle East and Africa
14	IT	Italy	...	1	Europe

3:1 26 rows selected in 0,032 seconds

RIGHT OUTER JOIN - пример

SQL Window - select * from countries c right outer join regions r on c.region_id = r.region_id order by 2 desc

SQL Output Statistics

```
select *  
from countries c right outer join regions r  
on c.region_id = r.region_id  
order by 2 desc
```

	COUNTRY_ID	COUNTRY_NAME	REGION_ID	REGION_ID	REGION_NAME
1				5	North Pole
2	ZW	Zimbabwe	4	4	Middle East and Africa
3	ZM	Zambia	4	4	Middle East and Africa
4	US	United States of America	2	2	Americas
5	UK	United Kingdom	1	1	Europe
6	CH	Switzerland	1	1	Europe
7	SG	Singapore	3	3	Asia
8	NG	Nigeria	4	4	Middle East and Africa
9	NL	Netherlands	1	1	Europe
10	MX	Mexico	2	2	Americas
11	ML	Malaysia	3	3	Asia
12	KW	Kuwait	4	4	Middle East and Africa
13	JP	Japan	3	3	Asia
14	IT	Italy	1	1	Europe

4:16 country_id, char(2), mandatory, Primary key of countries table.

FULL OUTER JOIN - пример

SQL Window - select * from countries c full outer join regions r on c.region_id = r.region_id order by c.regi ...

SQL Output Statistics

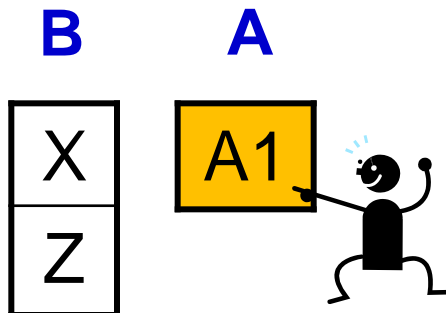
```
select *  
from countries c full outer join regions r  
on c.region_id = r.region_id  
order by c.region_id desc, r.region_id desc |
```

	COUNTRY_ID	COUNTRY_NAME	REGION_ID	REGION_ID	REGION_NAME
1	BG	Bulgaria	...		
2				5	North Pole
3	EG	Egypt	...	4	Middle East and Africa
4	ZM	Zambia	...	4	Middle East and Africa
5	NG	Nigeria	...	4	Middle East and Africa
6	KW	Kuwait	...	4	Middle East and Africa
7	IL	Israel	...	4	Middle East and Africa
8	ZW	Zimbabwe	...	4	Middle East and Africa
9	CN	China	...	3	Asia
10	AU	Australia	...	3	Asia
11	SG	Singapore	...	3	Asia
12	ML	Malaysia	...	3	Asia
13	JP	Japan	...	3	Asia
14	IN	India	...	3	Asia

4:45 27 rows selected in 0,032 seconds

Частно

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



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.

Частно

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.

Частно – 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	Компилятор
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

Частно – пример 2

Да извлечем всички преподаватели, които изнасят лекции едновременно в ПУ и СУ.

Lecturers

LEC	UNI
Стоян Колев	ПУ
Петър Иванов	СУ
Венета Георгиева	ТУ
Стоян Колев	СУ
Петър Иванов	ПУ
Петър Иванов	ТУ
Венета Георгиева	СУ

÷

Universities

UNI
ПУ
СУ

=

LEC
Стоян Колев
Петър Иванов

Частно - пример 2-1

	lec	uni
▶	Стоян Колев	ПУ
	Петър Иванов	СУ
	Венета Георгиева	ТУ
	Стоян Колев	СУ
	Петър Иванов	ПУ
	Петър Иванов	ТУ
	Венета Георгиева	СУ

	uni
▶	СУ
	ПУ

GCHOLAKOV-PC.Division - SQLQuery3.sql*

```
SELECT DISTINCT lec
FROM lecturers AS L1
WHERE NOT EXISTS
    (SELECT *
     FROM universities
     WHERE NOT EXISTS
        (SELECT *
         FROM lecturers AS L2
         WHERE L1.lec = L2.lec
              AND L2.uni = universities.uni));
```

Results Messages

	lec
1	Петър Иванов
2	Стоян Колев

Query execu... GCHOLAKOV-PC (9.0 SP3) sa (53) Division 00:00:00 2 rows

Частно - пример 2-2

	lec	uni
▶	Стоян Колев	ПУ
	Петър Иванов	СУ
	Венета Георгиева	ТУ
	Стоян Колев	СУ
	Петър Иванов	ПУ
	Петър Иванов	ТУ
	Венета Георгиева	СУ

	uni
▶	СУ
	ПУ

GCHOLAKOV-PC.Division - SQLQuery4.sql*

```
SELECT L1.lec
FROM lecturers AS L1, universities AS U
WHERE L1.uni = U.uni
GROUP BY L1.lec
HAVING COUNT(L1.uni) = (SELECT COUNT(uni) FROM universities);
```

Results Messages

	lec
1	Петър Иванов
2	Стоян Колев

Query executed successfully. GCHOLAKOV-PC (9.0 SP3) sa (55) Division 00:00:00 2 rows

Частно – пример 3

Искаме да извлечем всички доставчици, които доставят частите P1 и P3 едновременно.

Suppliers_Parts

Supp	Part
S1	P1
S2	P1
S4	P1
S1	P3
S3	P4
S4	P4
S4	P3

÷

Parts

Part
P1
P3

=

Supp
S1
S4

Частно - пример 4

A

Supp	Part
S1	P1
S1	P2
S1	P3
S1	P4
S2	P1
S2	P2
S3	P2
S4	P2
S4	P4

B1

Part
P2

B2

Part
P2
P4

B3

Part
P1
P2
P4

A / B1

Supp
S1
S2
S3
S4

A / B2

Supp
S1
S4

A / B3

Supp
S1

Частно – пример 5-1

Table - dbo.A...

	name	skill
▶	Георгиев	T-SQL
	Иванов	T-SQL
	Иванов	XML
	Колева	Java
	Колева	XML
	Петров	Java
	Петров	T-SQL
	Петров	XML
	Стоянов	Java
	Стоянов	T-SQL
	Стоянов	XML

1 of 11

T.

	skill
	XML
▶	Java
	T-SQL

2

GCHOLAKOV-PC.Division - D:\Университетски\Лекци...

```
SELECT A.name, COUNT(A.skill) AS skills
FROM ApplicantSkills AS A, Requirements AS R
WHERE A.skill = R.skill
GROUP BY A.name
```

Results Messages

	name	skills
1	Георгиев	1
2	Иванов	2
3	Колева	2
4	Петров	3
5	Стоянов	3

Query... GCHOLAKOV-PC (9.0 SP3) sa (55) Division 00:00:00 5 rows

GCHOLAKOV-PC.Division - D:\Университетски\Лекции\Бази данни\Samples\div...

```
SELECT A.name
FROM ApplicantSkills AS A, Requirements AS R
WHERE A.skill = R.skill
GROUP BY A.name
HAVING COUNT(A.skill) = (SELECT COUNT(skill) FROM Requirements);
```

Results Messages

	name
1	Петров
2	Стоянов

Query executed successfully. GCHOLAKOV-PC (9.0 SP3) sa (55) Division 00:00:00 2 rows

Частно – пример 5-2

	name	skill
▶	Георгиев	T-SQL
	Иванов	T-SQL
	Иванов	XML
	Колева	Java
	Колева	XML
	Петров	Java
	Петров	T-SQL
	Петров	XML
	Стоянов	Java
	Стоянов	T-SQL
	Стоянов	XML

	skill
▶	XML
	Java
	T-SQL

GCHOLAKOV-PC.Division - D:\Университетски\Лекции\Бази данни\S...

```
SELECT DISTINCT name
FROM ApplicantSkills AS L1
WHERE NOT EXISTS
    (SELECT *
     FROM Requirements
     WHERE NOT EXISTS
        (SELECT *
         FROM ApplicantSkills AS L2
         WHERE L1.name = L2.name
              AND L2.skill = Requirements.skill));
```

Results

	name
1	Петров
2	Стоянов

Query executed succe... GCHOLAKOV-PC (9.0 SP3) sa (53) Division 00:00:00 2 rows

За какво ни 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.

Разширение и сумиране

Много изследователи предлагат нови алгебрични оператори, като допълнение на тези на д-р Codd.

Ще разгледаме два такива:

- ▣ **EXTEND**
- ▣ **SUMMARIZE**

EXTEND

В много случаи искаме да включим в резултатната релация нови атрибути, стойностите на които са резултат от някакви изчисления.

Например: да включим атрибут, който съдържа данъка –

```
EXTEND LECTURER ADD (SALARY * 0.1) AS TAX
```

Резултатът е релация със:

- заглавна част - заглавната част на оригиналната релация, разширена с новия атрибут;
- тяло - всички записи от оригиналната релация, разширени с изчислената стойност на новия атрибут.

EXTEND - пример

LECTURER

ID	FName	LName	Title	Salary
101	Стоян	Колев	Проф.	550.00
102	Петър	Иванов	Доц.	432.76
103	Венета	Георгиева	Ст.н.с.	389.23

EXTEND LECTURER ADD (SALARY * 0.1) AS TAX

ID	FName	LName	Title	Salary	Tax
101	Стоян	Колев	Проф.	550.00	55.00
102	Петър	Иванов	Доц.	432.76	43.28
103	Венета	Георгиева	Ст.н.с.	389.23	38.92

SUMMARIZE

Този оператор извършва вертикално изчисление.

Резултатът е релация със:

- заглавна част - $\{A_1, \dots, A_n, Z\}$;
- тяло - всички записи t така, че t е един запис на проекция на A върху A_1, \dots, A_n , разширен с една стойност за новия атрибут Z ;
- новата Z -стойност е получена чрез изчисляване на израз за всички записи от A , които имат едни и същи стойности за A_1, \dots, A_n .

Степен на резултатната релация - степен на проекцията на оригиналната релация + 1.

Кардиналност на резултатната релация - кардиналност на проекцията на оригиналната релация.

SUMMARIZE

LECTURER

L_ID	SALARY	F_ID	F_NAME
L1	340		
L1	380		
L2	320		
L2	290		

**SUMMARIZE LECTURER
BY (L_ID)
ADD SUM(SALARY) AS S_TOTAL**

L_ID	S_TOTAL
L1	720
L2	610

LECTURER

ID	Name	Title	Faculty	Salary
101	Стоян Колев	Проф.	ФМИ	550.00
102	Петър Иванов	Проф.	ФМИ	522.76
103	Венета Георгиева	Проф.	Икономика	730.23
104	Мария Михайлова	Проф.	ФМИ	609.70
105	Стоил Караджов	Проф.	ФМИ	590.00
106	Милен Петров	Проф.	Икономика	670.00
107	Васил Ганчев	Доц.	ФМИ	415.20
108	Надя Тодорова	Доц.	ФМИ	453.00

**SUMMARIZE LECTURER
BY (TITLE, FACULTY)
ADD AVG(SALARY) AS AVGSAL**

Title	Faculty	AvgSal
Проф.	ФМИ	568.12
Проф.	Икономика	700.16
Доц.	ФМИ	434.10

- AvgSal е атрибутът Z;
- Атрибутите с еднакви стойности A_1, \dots, A_n са Title, Faculty, т.е. $n = 2$.

Релационни сравнения

expression @ *expression*

където *expression* са изрази, които се изчисляват до типово съвместими релации, а @ е някой от следните оператори за сравнение:

= равно

≠ различно

≤ подмножество на

< истинско подмножество на

≥ супермножество на

> истинско супермножество

Примери:

1. $S [CITY] = P [CITY]$, означаващ проекцията на доставчиците върху атрибута CITY същата ли е като тази на частите върху CITY?
2. $S [Supp] > SP [Supp]$, означаващ има ли доставчици, които не доставят никакви части?

В практиката често се налага да се провери дали дадена релация е празна, т.е. не съдържа записи. Нека тогава дефинираме логическа функция:

IS_EMPTY(*expression*)

която връща true, ако резултатът от изчислението на израза е празен, и false в противен случай.

Друг чест случай е да се налага да проверим дали даден запис t се среща в дадена релация R .

$$\{ t \} \leq R$$

Друго изразяване на това условие, познато от SQL, е

$$t \text{ IN } R$$

където IN е оператор за проверка на принадлежност към множество.