

14. Лекция

📁 Category	Empty
📎 Files	Empty
🕒 Created	June 6, 2023 1:25 PM
📅 Reminder	Empty
📌 Status	Open
🔗 URL	Empty
🕒 Updated	June 6, 2023 3:58 PM

Въпроси

Кои са основните класове операции?

Кои не можем да получим от други, кои можем да получим от други? (основни и допълнителни)

Мултимножества

SQL работи с мултимножества, елимира дубликатите при явно посочване (с distinct)

Операции върху мултимножества

Обединение, сечение и разлика - особености при мултимножествата

Върху множества нямаше повтарящи се кортежи, а при мултимножества може да има повтарящи се кортежи.

Прилага се върху релации, които имат съвместими/еквивалентни релационни схеми, т.е. трябва да имат еднакъв брой атрибути, атрибутите трябва да си съответсват.

Броят на кортежите в новополучената релация е равен на броя на кортежите в едната релация + броя на кортежите в другата релация

Един кортеж се среща толкова пъти в обединението на 2 мултимножества, колкото е съмата от срещанията му във всяко от мултимножествата

Трябва да приготвим по-значещи примери:

(всички студенти, които са се явили на изпита по УТК) \cup (всички студенти, които са се явили на изпита по Блокчейн)

Индивидуално двете релации нямат повтарящи се кортежи, но е възможно да има студенти, явили се и на двата изпита и тогава ще има повтарящи се кортежи.

Сечение

Един кортеж се среща в сечението на 2 мултимножества, толкова пъти, колкото е минимумът от срещанията му във всяко от мултимножествата

Ако имаме ситуация, в която и в двете релации всички кортежи са уникални, то и в сечението ще има само уникални елементи

Ако в едната релация има повторение на кортежи и в другата има и те се засичат по някакъв начин

Следователно в тази операция не се прави никаква друга промяна.

Разлика

Един кортеж се среща толкова пъти в разликата на мултимножествата А-Б, колкото е броят на срещанията му в А - броят на срещанията му в Б.

Пример 1: $\{1,2,1,1\} - \{1,2,3\} = \{1,1\}$

Пример 2: $\{1,2,3\} - \{1,2,1,1\} = \{2,3\}$

Нямаме нужда от допълнителна операция след това.

Тази операция има едно и също приложение и при множества и при мултимножества

Не всички алгебрични закони, които важат за множества не важат за мултимножества

- комутативният закон важи и при двата начина

Обединение при множество: $(S \cup S = S)$

Обединението като мултимножество е различно от обединението като множество.

$S \cup S \neq S$

Селекция

Резултатът от операцията представлява мултимножество

Ако изходната релация е множество, то и резултатната ще е множество.

Прилага се за всеки кортеж , еднакъв кортеж при множества и мултимножества

Проекция

Резултатът от операцията представлява мултимножество.

Възможно е без да има повтарящи се кортежи в изходната релация, резултатната да е мултимножество.

При мултимножества, за разлика от множества, дубликатите не се елиминират

Декартово произведение

Ако сме с множества, то резултатът е множество

Ако сме в реляционна алгебра с мултимножества, то резултатът ще е мултимножество

Не се налага нищо друго да правим.

Всеки кортеж от едната релация се свързва с всеки кортеж от другата, независимо от това дали се повтарят кортежите или не.

Съединение

Разлика между еквисъединение и натурално съединение

- естественото е частен случай на еквисъединението, при който се премахват дублираните стойности
- те се правят по абсолютно всички съвпадащи атрибути.

Това води до опасност от грешки

Трябва да внимаваме дали атрибутите с еднакви имена в съединението носят същия семантичен смисъл.

theta join

R(A,	B)	S(B,	C)
	1	2			3	4	
	5	6			7	8	
	1	2					

$$R \bowtie_{R.B < S.B} S =$$

A	R.B	S.B	C
1	2	3	4
1	2	7	8
5	6	7	8
1	2	3	4
1	2	7	8

Има ли разлика между проявата на тези съединения при множества и мултимножества. - не би трябвало да има, защото theta join ще ни даде повтарящи се кортежи, ако в изходните е имало.

Допълнителни оператори

1. **DELTA** (δ) = отстраняване на дубликати от мултимножества

$$R1 : \delta(R2)$$

В R1 съдържа само по едно копие на всеки кортеж, който се среща в R2 повече от един път.

2. **TAU** (τ) = сортиране на кортежи
3. **Разширена проекция** (π_L): аритметични операции, преименуване на колони
4. **GAMMA** (γ) = групиране и агрегиране
5. **Външно свързване**: предотвратява "висящи кортежи" = кортежи, които не участват в свързването

Сортиране

$$R1 := T_L(R2)$$

L - списък от атрибути на R2

R_1 - списък от кортежите на R_2 , сортирани първо по първия атрибут на списъка L , после по втория и т.н.

По подразбиране е възходящ ред на сортиране.

Може по един атрибут да сортираме възходящо, а по друг низходящо.

$$R =$$

A	B
1	2
3	4
5	2

$$TAU_B(R) = [(5,2), (1,2), (3,4)]$$

Ако сортираме само по 1 атрибут, кортежите ще появят по произволен ред спрямо останалите атрибути. Това, че в изходната релация имаме първо (1,2) и после (5,2), не значи че и в резултатната ще е така.

Разширена проекция

Използвайки същия π_L оператор, позволяваме списъкът L да съдържа произволни изрази от атрибутите:

1. Единичен атрибут на R
2. Израз $x \rightarrow y$ (преименуване на атрибута x в y)
3. Аритметика върху атрибутите: $A+B$

Пример: Разширена проекция

$$R =$$

A	B
1	2
3	4

$$\pi_{A+B, A, B}(R) =$$

A+B	A	B
3	1	2
7	3	4

$$\pi_{A+B \rightarrow C, A, B}(R) =$$

C	A	B
3	1	2
7	3	4

Агрегиращи оператори

Агрегиращите оператори се прилагат върху целите колони и дават единичен резултат

Кога са ни необходими тези оператори? За да получим обобщени резултати.

При случаите, когато прилагаме агрегиращите оператори върху цялата релация.

Пример: агрегиране

Като обобщение на цифров атрибут може да намерим

- SUM
- AVG
- MIN and MAX
- COUNT

R =

A	B
1	3
3	4
3	2

SUM(A) = 7
 COUNT(A) = 3
 MAX(B) = 4
 MIN (B) = 2
 AVG(B) = 3

(Примерно заплата във фирмата: сума на заплатите, средна, ...)

Оператор за групиране

- $R_1 := \gamma_L(R_2)$.
- L списък от елементи, които са:
 1. Индивидуални атрибути (*grouping attributes*) .
 2. $AGG(A)$, където AGG е агрегиращ оператор и A е атрибут.

Приложение на $\text{GAMMA}_L(R)$

- Групираме R спрямо всички групиращи атрибути от списъка L .
- Във всяка група изчисляваме $AGG(A)$ за всяко агрегиране върху списъка L .
- Резултатът с-жа един кортеж за всяка група:
 1. Групиращи атрибути и
 2. Техните групови агрегации.

Пример: групиране/агрегиране

$$R =$$

A	B	C
1	2	3
4	5	6
1	2	5

$$\gamma_{A,B,AVG(C)}(R) = ??$$

I – групиране в R :

A	B	C
1	2	3
1	2	5
4	5	6

II – средна стойност на C в групата

A	B	AVG(C)
1	2	4
4	5	6

Разликата е, че тук за всяка група получаваме отделен резултат от тази агрегираща функция, а върху цялата релация получавахме само 1 число. Изходът от операциите е различен

Групирането по-горе е по атрибута A и по атрибута B.

Външно свързване (Outerjoin)

$$R \bowtie_C S$$

Кортежите на R , които не могат да образуват двойки (да се свържат) с кортеж от S се наричат **висящи**. Аналогично за S .

Outerjoin (\bowtie^o) запазва висящите кортежи, като ги включва в резултата, допълвайки ги с NULL values.

Пример: външно свързване

$$R =$$

A	B
1	2
4	5

$$S =$$

B	C
2	3
6	7

(1,2) се свързва с (2,3),
но остават 2 висящи кортежа.

$$R \bowtie^o S =$$

A	B	C
1	2	3
4	5	NULL
NULL	6	7

Естествено свързване би ни дало само кортежа (1,2,3).

Ако искаме да имаме и другите два кортежа, то трябва да напишем null в атрибутите на R или S , които не знаем.

Попълваме само атрибутите

Name	Sign	Name	Sign
SELECT	σ	INTERSECT	\cap
Project	π	MINUS	$-$
Multiply (*)	\times	TAU (sort)	τ
JOIN	\bowtie	DELTA (remove duplicates)	δ
RENAME	ρ	GAMMA (group and aggregate)	γ
UNION	\cup	OUTERJOIN	\bowtie°

Задача

Employee

ID	Name	Address	Gender	Salary	deptNo
1002	J. Blake	123, Maple Str., NY	M	5000	10
1005	A. Smith	456 Oak Str, NY	F	4500	20
1024	K. Brown	784 Oak Str. Boston	M	4200	20
1053	M. Grey	243 Wash, LA	F	4300	30

Department

DeptNo	DName	Location
10	HR	NY
20	Finance	Boston
30	Programming	LA

Какво ще получим при пълно външно свързване?

Employee \bowtie *Department*

Result:

ID	Name	Address	Gender	Salary	deptNo	DName	Location
1002	J. Blake	123, Maple Str., NY	M	5000	10	HR	NY
1005	A. Smith	456 Oak Str, NY	F	4500	20	Finance	Boston
1024	K. Brown	784 Oak Str. Boston	M	4200	20	Finance	Boston
1053	M. Grey	243 Wash, LA	F	4300	30	Programming	LA

Един от нашите колеги отива на стаж в тази фирма:

<u>ID</u>	Name	Address	Gender	Salary	deptNo
1002	J. Blake	123, Maple Str., NY	M	5000	10
1005	A. Smith	456 Oak Str, NY	F	4500	20
1024	K. Brown	784 Oak Str. Boston	M	4200	20
1053	M. Grey	243 Wash, LA	F	4300	30
12000	П. Иванов	56 ..., LA	M	1200	NULL

ID - ключ

DeptNo - ключ

(СУБД е case insensitive)

Ако имахме само естествено съединение, то сумата от заплатите щеше да е 17000, щеше да липсва заплатата на колегата ни.

За да добавим и висящият кортеж на нашият колега, трябва да направим left join

Employee ⋈_L *Department*

ID	Name	Address	Gender	Salary	deptNo	DName	Location
1002	J. Blake	123, Maple Str., NY	M	5000	10	HR	NY
1005	A. Smith	456 Oak Str, NY	F	4500	20	Finance	Boston
1024	K. Brown	784 Oak Str. Boston	M	4200	20	Finance	Boston
1053	M. Grey	243 Wash, LA	F	4300	30	Programming	LA
12000	П. Иванов	56 ..., LA	M	1200	NULL	NULL	NULL

Ако фирмата се разширява и прави нов департамент:

DeptNo	DName	Location
10	HR	NY
20	Finance	Boston
30	Programming	LA
40	Marketing	LA

Ако ръководството иска справка, то той ще получи резултатът от естествения join по-горе, защото маркетингът още няма никакъв служител. За да включим и кортеж за маркетинга:

Employee \bowtie_R *Department*

ID	Name	Address	Gender	Salary	deptNo	DName	Location
1002	J. Blake	123, Maple Str., NY	M	5000	10	HR	NY
1005	A. Smith	456 Oak Str, NY	F	4500	20	Finance	Boston
1024	K. Brown	784 Oak Str. Boston	M	4200	20	Finance	Boston
1053	M. Grey	243 Wash, LA	F	4300	30	Programming	LA
NULL	NULL	NULL	NULL	NULL	40	Marketing	LA

Ако искаме всичката информация:

Employee ⋈ *Department*

ID	Name	Address	Gender	Salary	deptNo	DName	Location
1002	J. Blake	123, Maple Str., NY	M	5000	10	HR	NY
1005	A. Smith	456 Oak Str, NY	F	4500	20	Finance	Boston
1024	K. Brown	784 Oak Str. Boston	M	4200	20	Finance	Boston
1053	M. Grey	243 Wash, LA	F	4300	30	Programming	LA
12000	П. Иванов	56 ..., LA	M	1200	NULL	NULL	NULL
NULL	NULL	NULL	NULL	NULL	40	Marketing	LA

Връзката на DeptNo е **връзка между външен ключ и ключ**.

Може ли външен ключ да сочи към първичен ключ в същата релация - да.

Пример:

<u>ID</u>	Name	Address	Gender	Salary	deptNo	Manager
1002	J. Blake	123, Maple Str., NY	M	5000	10	-
1005	A. Smith	456 Oak Str, NY	F	4500	20	1002
1024	K. Brown	784 Oak Str. Boston	M	4200	20	1005
1053	M. Grey	243 Wash, LA	F	4300	30	
12000	П. Иванов	56 ..., LA	M	1200	NULL	

Като оператори в релационната алгебра как ще изразим връзката между външния ключ и първичния.

Ако има кортеж със стойност във външния ключ, то тази стойност трябва да съществува в първичния ключ, който е сочен.

Изразяване в термините на релационната алгебра