Релационна алгебра

Мултимножества

(Bags)

РА върху мултимножества

- Мултимножество *(bag)*
  - за разлика от множествата един елемент може да се среща повече от 1 път
  - *Multiset* , *bag*
- Пример: {1,2,1,3}
- Редът в мултимножеството не е от значение
  - {1,2,1} = {1,1,2} като мултимножества,
  - но [1,2,1] <> [1,1,2] като списъци

Защо се използват мултимножества?

- SQL на практика работи върху мултимножества.
  - SQL елиминира дубликатите при явно посочване
- Някои оператори (проекция) са много по-ефективни върху мултимножества, отколкото върху множества.

Операции върху мултимножества

- Обединение, сечение и разлика - особености при мултимножества.

Обединение на мултимножества

- Обединение - един кортеж се среща толкова пъти в обединението на 2 мултимножества, колкото е сумата от срещанията му във всяко от мултимножествата.
- Пример: {1,2,1} $\cup$ {1,1,2,3,1} = {1,1,1,1,1,2,2,3}

Сечение на мултимножества

- Сечение – един кортеж се среща в сечението на на 2 мултимножества, толкова пъти, колкото е минимумът от срещанията му във всяко от мултимножествата.
- Пример: $\{1,2,1\} \cap \{1,2,3\} = \{1,2\}$.

Разлика на мултимножества

- Разлика - един кортеж се среща толкова пъти в разликата на мултимножествата $A - B$, колкото е броят на срещанията му в А минус броят на срещанията му в В.

- Пример 1: {1,2,1,1} – {1,2,3} = {1,1}.
- Пример 2: {1,2,3} - {1,2,1,1} = ?

Bag Laws != Set Laws

- Някои, но *не всички* алгебрични закон, които важат за множества не важат за мултимножества.
- Пример: комутативен закон при обединение ($R \cup S = S \cup R$) важи и при множества.

Разлика в операциите върху мултимножества и множества

- Множества:
  обединение -> ($S \cup S = S$).

Разлика в операциите върху мултимножества и множества

- Множества: обединение -> ($S \cup S = S$).
- Мултимножества: ако x се среща n пъти в S, ще се среща $2n$ пъти в $S \cup S$.
- $S \cup S <> S$

Операции върху мултимножества

- Селекция
- Проекция
- Декартово п-ние (и съединение)

Пример: селекция

R(

A,	B
1	2
5	6
1	2

)

$\sigma_{A+B<5}$ (R) =

A	B
1	2
1	2

Пример: проекция

R(

A,	B
1	2
5	6
1	2

)

$\pi_A (R) =$

A
1
5
1

Пример: декартово п-ние

R(

A,	B
1	2
5	6
1	2

)

S(

B,	C
3	4
7	8

)

$R \times S =$

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

Операции върху мултимножества

- Селекция - прилага се за всеки кортеж, еднакъв ефект при множества и мултимножества.
- Проекция - при мултимножества дубликатите не се елиминират.
- Декартово п-ние (и съединение) – всеки кортеж от едната релация се свързва с всеки кортеж от другата, независимо от това дали се повтарят кортежите или не.

Пример: Theta-Join върху мултимножества

R(

A,	B
1	2
5	6
1	2

)

S(

B,	C
3	4
7	8

)

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

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

Релационна алгебра

Допълнителни оператори

Допълнителни оператори

1. DELTA (δ) = отстраняване на дубликати от мултимножества.
2. TAU (τ) = сортиране на кортежи.
3. *Разширена проекция* : артитметични операции,преименуване на колони.
4. GAMMA (γ) = групиране и агрегиране.
5. *Външно свързване* : предотвратява "висящи кортежи" = кортежи, които не участват в свързването.

Отстраняване на дубликати

- R1 := δ(R2)
- R1 съдържа само по едно копие на всеки кортеж, който се среща в R2 повече от един път.

Пример: Отстраняване на дубликати

R =

A	B
1	2
3	4
1	2

$\delta(R)$ =

A	B
1	2
3	4

Сортиране

- R1 := τ_L (R2).
  - *L* списък от атрибути на R2.
- R1 – списък от кортежите на R2, сортирани първо по първия атрибут на списъка *L*, после по втория атрибут на *L и т.н.*
- TAU е единственият оператор, чийто резултат е списък от кортежи, а не множество от кортежи

Пример: Сортиране

R =

A	B
1	2
3	4
5	2

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

Разширена проекция

- Използвайки същия π_L оператор, позволяваме списъкът L да съдържа произволни изрази от атрибутите:
  1. Единичен атрибут на R
  2. Израз: **x -> 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

$\pi_{A+B\text{->}C,A,B}$ (R) =

C	A	B
3	1	2
7	3	4

Агрегиращи оператори

- Агрегиращите оператори се прилагат върху целите колони и дават единичен резултат
- Примери:
  - SUM
  - AVG
  - MIN and MAX.
  - COUNT

Пример: агрегиране

R =

A	B
1	3
3	4
3	2

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

Оператор за групиране

- R1 := γ_L (R2).
- *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 $GAMMA_L(R)$

- 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

$\gamma_{A,B,\mathrm{AVG}(C)}$ (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

Външно свързване (Outerjoin)

- $R \bowtie_C S$.
- Кортежите от R , които не могат да образуват двойка (да се свържат) с кортеж от S се наричат “висящи” *(dangling).*
  – Аналогично за S.
- Outerjoin ($\overset{\circ}{\bowtie}$) 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.

$R \overset{\circ}{\bowtie} S$ =

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

Оператори на релационната алгебра

SELECT	σ	INTERSECT	$\cap$
PROJ	π	MINUS	$-$
*	$\times$	TAU	τ
JOIN	$\bowtie$	DELTA	δ
RENAME	ρ	GAMMA	γ
UNION	$\cup$	OUTERJOIN	$\overset{\circ}{\bowtie}$

Ограничения върху релации

Две форми за представяне на ограниченията

- Нека R, S са изрази от РА
  - R = $\emptyset$ “релацията R не трябва да съдържа кортежи”
  - R $\subseteq$ S “всеки кортеж от релацията R трябва да присъства като кортеж в S”

Пример 1:външен ключ

Movie(<u>Title,Year</u>,length,filmType, studioName, producerC#)

MovieExec(name,address,<u>Cert#</u>,netWorth)

- producerC# на всеки кортеж за филм трябва да да присъства като cert# компонент на някои кортежи в MovieExec (референциален интегритет)

Пример 1: външен ключ

Movie(<u>Title,Year</u>,length,filmType, studioName, producerC#)

MovieExec(name,address,<u>Cert#</u>,netWorth)

- Чрез релационната алгебра

$$\pi_{producerC\#}(Movie) \subseteq \pi_{cert\#}(MovieExec)$$

$$\pi_{producerC\#}(Movie) - \pi_{cert\#}(MovieExec) = \emptyset$$

Пример 2

- Всеки филм от релацията

`StarsIn(movieTitle,movieYear,starName)`

- се съдържа и в релацията

`Movie(title,year,length,inColor,studioName,producerC#)`

- Чрез релационната алгебра:

$$\pi_{\text{movieTitle,movieYear}}(\text{StarsIn}) \subseteq \pi_{\text{title,year}}(\text{Movie})$$

Пример 3: функционални зависимости

- Изразяване на FD:
  - name $\rightarrow$ address
  - MovieStar(name,address,gender,birthday)
- Чрез релационната алгебра

$$\sigma_{MS1.name=MS2.name \text{ AND } MS1.address \neq MS2.address} (\rho_{MS1}(MovieStar) \times \rho_{MS2}(MovieStar)) = \emptyset$$

Пример 4: ограничения на домейна

- Допустими значения за атрибута *gender* (пол) са символните константи ‘F’ (female – жена) и ‘M’ (male - мъж).
- В релационната алгебра:

$$\sigma_{\text{gender} \neq \text{'F' AND gender} \neq \text{'M'}}(\text{MovieStar}) = \emptyset$$