

13. Лекция. Релационна алгебра

Category	Empty
Files	Empty
Created	May 30, 2023 1:21 PM
Reminder	Empty
Status	Open
URL	Empty
Updated	June 6, 2023 1:27 PM

Обединение - Union (binary, commutative, associative)

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

Базираме се на базата данни:

Movie(Title, Year, length, inColor, studioName, producerC#)

StarsIn(MovieTitle, MovieYear, StarName)

Релационна алгебра

Операциите се извършват върху релациите

Същност:

Операнди - променливи или стойности, от които се конструират нови стойности

Оператори -

Релационна схема - подредени атрибути (има значение конкретното подреждане)

Какво означава да **съберем** две релации:

- обединение на кортежи, когато имат съвпадащи или еквивалентни/съпоставими релационни схеми

Какво означава **изваждане**

- изваждаме кортежите на едната релация от кортежите на другата, като релациите имат еквивалентни релационни схеми

Код въвежда

- обединение - събирането от математиката
- разлика - изваждането
- декартово произведение
- селекция
- проекция

Съединения

Реализация на операторите на РА в езиците на заявките

- множества и мултимножества - в някои случаи може да се случат дублирания на кортежи. Примерно при проекцията може да се получат в междинни резултати да има дублиращи се кортежи.

Съвременните СУБД-та работят с временни мултимножества, защото чистенето на повтарящите се кортежи на всяка стъпка е твърде времеемко.

Сега ще разгледаме релационна алгебра върху множества

Защо ни трябва релационна алгебра?

Всички DBMSs използват релационната алгебра като междинен език за изчисление на заявките

- синтактичен анализ на SQL и транслиране в изрази от релационната алгебра
- Изразите могат да бъдат неефективни

Множество от правила за обработка на алгебрични изрази

- не съществуват в SQL
- изразите се конвертират в други, по-оптимални
- извършва се от query optimizer

Обработка на заявките

1. SQL query
Parser
2. Relational Algebra Expression
Query Optimizer
3. Query execution plan
Code Generator
4. Executable code

Създаване на нови (по-комплексни) ралции на базата на съществуващи

- множество от оператори

Основни класове операции

1. **Операции върху множества**
 - a. обединение
 - b. сечение
 - c. разлика
2. **Селекция и проекция** - специфични за рел. алгебра
3. **Комбиниране на кортежите на 2 релации**
 - a. декартово произведение

Трябва да имаме операции, които ни позволяват съвместяване/съединяване на таблици

4. Преименуване

Няма как да се присъедини към нито един от другите класове

Основи

1. Ядро на РА
 - a. релациите са множества от кортежи
2. Разширена РА
 - a. релациите са мултимножества от кортежи (съдържат повторения на кортежи)
3. 4те основни класа операции
 - a. операции върху множества
 - b. отстраняват части от релациите
 - c. комбинират кортежи от 2 релации
 - d. преименуват

Теоретико-множествени операции

Обединение - Union (binary, commutative, associative)

$$R \cup S$$

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

Като резултат получаваме нова релация със същата релационна схема, където кортежите са обединение от кортежите на 2те релации

Сечение - Intersection (binary, commutative, associative)

$$R \cap S$$

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

Като резултат получаваме нова релация със същата релационна схема, където кортежите са само общите кортежите на 2те релации

Разлика - Set Difference (binary)

$$R - S$$

$$!! \quad R - S \neq S - R$$

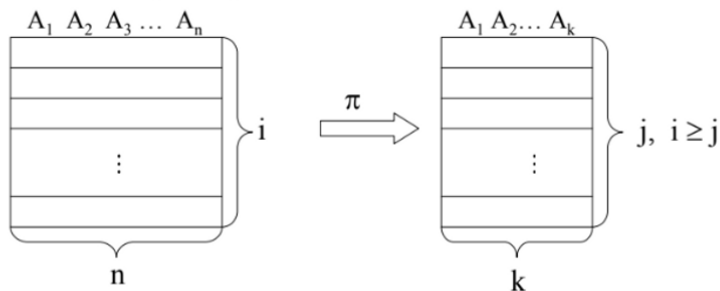
Прилага се върху релации, които имат съвместими/еквивалентни релационни схеми, т.е. трябва да имат еднакъв брой атрибути, атрибутите трябва да си съответсват

► Пример

Проекция- unary

Project (unary)

- $\pi_{\langle \text{attr list} \rangle} (R)$
- $\langle \text{attr list} \rangle$ списък атрибути (колони)
- Пример: $\pi_{\text{title, year, length}} (\text{Movie})$ "horizontal restriction"



$$\pi_{\langle \text{attr_list} \rangle} (R)$$

$\langle \text{attr_list} \rangle$ - списък с атрибутите, които искаме да получим в новата релация

Тази операция се нарича хоризонтална рестрикция, защото всъщност скъсяваме по хоризонтала тази релация, защото не взимаме всички атрибути, а само част от тях.

(Ако има повтарящи се кортежи в новата таблица, те се махат)

В общия случай бр. на кортежите в новата релация е \leq на кортежите в изходната релация.

Резултатът отново е релация, но тя съдържа само атрибутите посочени в $\langle \text{attr_list} \rangle$.

В стратегията за декомпозиция правим проекция.

► Пример

distinct стои веднага след select, защото само така показваме, че се отнася за всички, а не е свързано с някоя от отделните клаузи като from, group by

Последна клауза, която може да ползваме в select е order by, за да не се мъчи СУБД всеки път да сортира.

Select or Restrict (unary, commutative)

$$\sigma_{\langle predicate \rangle}(R)$$

$\langle predicate \rangle$ - условен израз

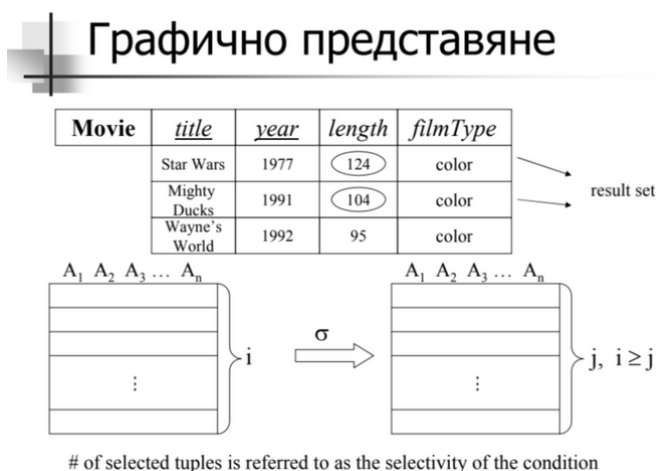
- $\langle attribute \rangle \langle op \rangle \langle attribute \rangle$
- $\langle attribute \rangle \langle op \rangle \langle constant \rangle$
- $op \in \{=, \neq, <, >, \leq, \dots, \text{AND}, \text{OR}\}$

Резултатът е нова релация със същата реляционна схема, но отрязваме част от кортежите. Оставяме само кортежите, които отговарят на условията израз.

Целта на тази операция е да отсее тези кортежи, които не са ни необходими за даден резултат.

Казваме, че това е вертикална рестрикция, защото новата релация е със същите атрибути, но броят на кортежите ще е ограничен, т.е. вертикално ще бъде по-къса в общия случай. Очаква се, че не всички кортежи ще отговарят на условието

► Пример



Декартово произведение

Cartesian product (binary, commutative, associative)

$$R \times S$$

Множеството от всички двойки, при които първият елемент е произволен елемент от R, а вторият - от S

Резултатът е нова релация с релационна схема, която е обединение на схемите на R и S

В общият случай можем да имаме еднакви имена на атрибути в двете релации, за да означим че става дума за едната или другата релация използваме първото име:

<Име на релация>.<име на атрибут>

► Пример

При SQL декартово неявно се получава, когато във from изброим няколко неща.

Декартовото произведение е много тежко като операция

Join operations

Theta Join (binary)

$$R \bowtie_C S$$

Където C е условието на свързване

Изпълнение:

- Декартово произведение на R и S
- Избор на кортежите, които удовлетворяват условието C

Theta Join (binary)

- $R \bowtie_C S$, където C е условието на свързване
- Изпълнение
 - 1: Декартово произведение на R и S
 - 2: Избор на кортежите, които удовлетворяват условието C
- Като резултат от декартовото произведение, схемата на резултата е обединение на схемите на R и S

Като резултат от декартовото произведение, схемата на резултата е обединение на схемите на R и S

► Пример

Налага ни се да правим join заради

Обикновено търсим съединение по някакви атрибути

тета съединение - theta join

Гръцката буква θ се използва за означение на условието

Theta join, при което join condition включва само съвпадение по атрибутите := **еквисъединение (equijoin)**

- формира базата за "joining" relations

Естествено съединение (natural join) - разширение на equijoin

- означение $R \bowtie S$
- свързване по всички атрибути с еднакви имена
- автоматично отстраняване на повтарящата се колона

► Пример

Преименуване (Rename)

Операторът ρ дава нова схема на релацията

- Операторът ρ дава нова схема на релацията
- $R1 := \rho_{R1(A1, \dots, An)}(R2)$ превръща R1 в релация с атрибути $A1, \dots, An$ и същите кортежи като на R2.
- $R1(A1, \dots, An) := R2$.
- Пр: $\rho_{MyMovie} (Movie)$
- Пр: $\rho_{MyMovie(title, year, length, type)} (Movie)$

► Пример

Класификация - Основни и допълнителни

Основни

Select	set difference
Project	cartesian product
union	rename

- всички други могат с тях

Допълнителни

Сечение $R \cap S = R - (R - S)$

theta join $R \bowtie_C S = \sigma_C(R \times S)$

natural join $R \bowtie S = \pi_L(\sigma_C(R \times S))$

- Където L - списък с атрибутите в R, атрибутите в S, които не са в R
- C - условие от вида $R.A_1 = S.A_1 \text{ AND } R.A_2 = S.A_2 \text{ AND } \dots \text{ AND } R.A_n = S.A_n$, където A_1, A_2, \dots, A_n са всички атрибути в схемите на R и S

Създаване на сложни изрази

Алгебрата представя по естествен начин последователността от операции

- аритметика - - - $(x+4)*(y-3)$

Релационна алгебра - 3 нотации

1. Последователности от ператори за присвояване - линейна нотация за алгебрични изрази
2. изрази с няколко оператори
3. дървета от изрази

Последователност от присвоявания

Създаване на временни имена на релации

преименуване

Пример: $R_3 := R_1 \bowtie_C R_2$ може да се представи като:

$$R_4 := R_1 \times R_2$$

$$R_3 := \sigma_C(R_4)$$

Приоритет на операторите

1. unary operators - **select, project, rename**
2. декартово произведение и съединение
3. Сечение
4. обединение и разлика
 - използване на скоби

Трябва да запомним приоритета

Пример:

Expression Trees

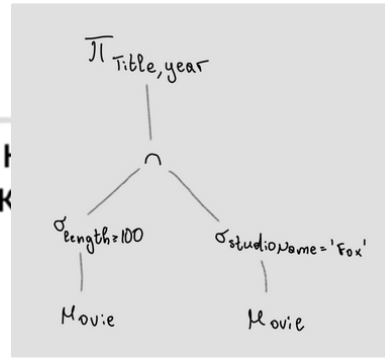
Листата са операнди - релации

Вътрешните възли - оператори, приложени към техните наследници

Пример

- “Кои са имената и годините на филмите на студио Fox, които са поне 100 мин ?”

1. $LMovies \leftarrow \sigma_{length \geq 100} (Movie)$
2. $FoxMovies \leftarrow \sigma_{studioName='Fox'} (Movie)$
3. $Temp \leftarrow LMovies \cap FoxMovies$
4. $Result \leftarrow \pi_{Title, Year} (Temp)$



$$Result \leftarrow \pi_{Title, Year}(\sigma_{studioName='Fox' \text{ AND } length \geq 100}(Movie))$$

Или

$$Result \leftarrow \pi_{Title, Year}(\sigma_{length \geq 100}(Movie) \cap \sigma_{studioName='Fox'}(Movie))$$

Пример

Актьори, които участват във филми с продължителност поне 100 минути

Movie1 (title, year, length, filmType, studioName)

Movie2(title, year, starName)