Relacijska algebra

Iztok Savnik, FAMNIT

Viri

- *Prosojnice: „Cow Book“, R.Ramakrishnan, http://pages.cs.wisc.edu/~dbbook/*
- *Raghu Ramakrishnan, Johannes Gehrke, Database Management Systems, McGraw-Hill, 3rd ed., 2007.*

Relacijski povpraševalni jeziki

- *Povpravševalni jezik:* Omogoča urejanje podatkov in poizvedovanje po podatkih v podatkovni bazi.
- Relacijski model podpira enostavne PJ z veliko izrazno močjo:
  - Formalne osnove v logiki.
  - Omogoča optimizacijo poizvedb.
- Povpraševalni jezik **!=** Programski jezik !
  - PJ ni “Računsko kompleten”.
  - PJ ni namenjen za kompleksne izračune.
  - PJ podpira enostaven in učinkovit dostop do velikih zbirk podatkov.

Formalni relacijski povpraševalni jeziki

- Dva formalna (matematična) jezika tvorita osnovo za “realne” povpraševalne jezke (npr. SQL) in njihovo implementacijo:
  - *Relacijska algebra*: Bolj proceduralen jezik, uporaben za predstavitev plana izvajanja poizvedb.
  - *Relacijski račun*: Omogoča uporabnikom opisati kaj želijo in ne toliko kako to izračunati – deklarativni jezik.

Osnove

- Poizvedba je izvršena nad instancami relacij in rezultat poizvedbe je instanca neke relacije.
  - *Sheme vhodnih* relacij so fiksne.
  - *Shema rezultata* je tudi fiksna -- določi se s pravili gradnikov povpraševalnega jezika.
- Notacija osnovana na poziciji oz. imenih atributov:
  - Notacija osnovana na poziciji primernejša za programe; notacija osnovana na imenih je bolj berljiva.
  - Obe se uporabljata v SQL.

Primer podatkovne baze

Mornarji(mid:integer, mime:string, ocena:integer, star:integer)
Ladje(lid:integer, lime:string, barva:string)
Rezervacije(mid:integer, lid:integer, datum:date)

Primeri relacij

- Relaciji “Mornarji” in “Rezervacije” za naše primere.
- Uporabljali bomo notaciji osnovani na poziciji in imenih.
- Imena atributov v vmesnih in končnem rezutatu poizvedb se podedujejo od vhodnih relacij.

R1

mid	lid	dan
22	101	10/10/96
58	103	11/12/96

S1

mid	mime	ocena	star
22	novak	7	45.0
31	kranjc	8	55.5
58	petelin	10	35.0

S2

mid	mime	ocena	star
28	volk	9	35.0
31	kranjc	8	55.5
44	jauk	5	35.0
58	petelin	10	35.0

Relacijska algebra

- Osnovne operacije
  - *Selekcija* (δ) Izbere podmnožico n-teric iz relacije.
  - *Projekcija* (Π) Izbere določene stolpce relacije.
  - *Produkt* (×) Omogoča kombiniranje dveh relacij.
  - *Razlika* (-) N-terice iz prve in ne iz druge relacije.
  - *Unija* (U) N-terice iz obeh relacij.
- Dodatne operacije
  - Presek, *Stik*, Deljenje, Preimenovanje.
  - Niso nujne, so pa ZELO (!) koristne.
- Vsaka operacija vrne relacijo kot rezultat.
  - Operacije se lahko sestavljajo – funkcijski jezik.

Projekcija

- Izbere atribute, ki so v *listi projekcije* iz relacije.
- *Shema* rezultata vsebuje samo atribute, ki so v listi projekcije z istimi imeni kot v vhodni relaciji.
- Duplikati?
  - Realni sistemi tipično ne odstranijo duplikate, če uporabnik tega ne zahteva. (Zakaj ne?)

mime	ocena
volk	9
kranjc	8
jauk	5
petelin	10

$$\pi_{mime,ocena}(S2)$$

star
35.0
55.5

$$\pi_{star}(S2)$$

Selekcija

- Izbere vrstice, ki zadoščajo *pogoju selekcije*.
- Ni duplikatov v rezulatu. (Zakaj?)
- *Shema* rezulata je identična shemi vhodnih relacij.
- *Relacija*, ki je rezultat je lahko vhodna relacija drugi relacijski operaciji! (*Kompozicija operacij*.)

mid	mime	ocena	star
28	volk	9	35.0
58	petelin	10	35.0

$$\sigma_{ocena>8}(S2)$$

mime	ocena
volk	9
petelin	10

$$\pi_{mime,ocena}(\sigma_{ocena>8}(S2))$$

Unija, Presek, Razlika

- Vse operacije so binarne in vhodni relaciji morata biti *unija-kompatibilni*:
  - Enako število atributov.
  - `Pripadajoča' polja imajo enake tipe.
- Kaj je *shema* rezultata?

mid	mime	ocena	star
22	novak	7	45.0
31	kranjc	8	55.5
58	petelin	10	35.0
44	jauk	5	35.0
28	volk	9	35.0

$S1 \cup S2$

mid	mime	ocena	star
22	novak	7	45.0

$S1 - S2$

mid	mime	ocena	star
31	kranjc	8	55.5
58	petelin	10	35.0

$S1 \cap S2$

Produkt

- *Kartezijski produkt*: vsaka vrstica S1 se poveže z vsako vrstico R1.
- *Shema rezultata* ima po en atribut za vsak atribut relacij S1 in R1; imena od operandov.
  - *Konflikt*: S1 in R1 imata atribut *mid*.

(mid)	mime	ocena	star	(mid)	lid	day
22	novak	7	45.0	22	101	10/ 10/ 9
22	novak	7	45.0	58	103	11/ 12/ 9
31	kranjc	8	55.5	22	101	10/ 10/ 9
31	kranjc	8	55.5	58	103	11/ 12/ 9
58	petelin	10	35.0	22	101	10/ 10/ 9
58	petelin	10	35.0	58	103	11/ 12/ 9

- *Preimenovanje*: $\rho(C(1\rightarrow mid1,5\rightarrow mid2),S1\times R1)$

Stik (Join)

- *Stik s pogojem*: $R \bowtie_c S = \sigma_c (R \times S)$

(mid)	mime	ocena	star	(mid)	lid	dan
22	novak	7	45.0	58	103	11/ 12/ 96
31	kranjc	8	55.5	58	103	11/ 12/ 96

$$S1 \bowtie_{S1.mid < R1.mid} R1$$

- *Shema rezultata:* enako kot kartezijski produkt.
- Manj n-teric kot produkt; da se izračunati hitreje.
- Včasih ga imenujejo *theta-stik.*

Stiki

- *Equi-Stik*: Poseben primer stika, kjer je pogoj stika uporablja samo pogoj enačaj.

mid	mime	ocena	star	lid	dan
22	novak	7	45.0	101	10/ 10/ 96
58	petelin	10	35.0	103	11/ 12/ 96

$$S1 \quad {}_{mid} \quad R1$$

- *Shema rezultata:* podobno kot kartezijski produkt; samo ena vrednost enačenih atributov je v rezultatu.
- *Naravni Stik*: Equi-Stik po vseh skupnih atributih.

Deljenje

- Ni osnovna operacija; uporabna za izražanje vprašanj kot na primer:
  - *Poišči mornarje, ki so rezervirali vse ladje.*
- Naj ima *A* dva atributa *x* in *y*; *B* pa samo en atribut *y*:
  - $A/B = \left\{ \langle x \rangle \mid \exists \langle x, y \rangle \in A \;\; \forall \langle y \rangle \in B \right\}$
  - ***A/B* vsebuje vse n-terice *x* (mornarji) tako da za vsako n-terico *y* (ladja) v *B*, obstaja n-terica *xy* v *A*.**
  - *Ali*: Če množica vrednosti *y* (ladje) povezana z vrednostjo *x* (mornarji) v *A* vsebuje vse vrednosti *y* v *B*, potem je vrednost *x* v *A/B*.
- V splošnem sta *x* in *y* lahko poljubna seznama atributov; *y* je seznam atrributov v *B*, in $x \cup y$ je seznam atributov v *A*.

Primer deljenja A/B

sno	pno
s1	p1
s1	p2
s1	p3
s1	p4
s2	p1
s2	p2
s3	p2
s4	p2
s4	p4

A

pno
p2

B1

sno
s1
s2
s3
s4

A/B1

pno
p2
p4

B2

sno
s1
s4

A/B2

pno
p1
p2
p4

B3

sno
s1

A/B3

Izražanje A/B z osnovnimi operacijami

- Deljenje ni nujno potrebna operacija; uporabna bljižnica.
  - To je načeloma res tudi za stike, čeprav omogočajo stiki učinkovito implementacijo poizvedb.
- *Ideja*: *A/B* = izračunaj vse vrednosti *x*, ki niso izločeni z vrednostjo *y* v *B*.
  - *x* je izločena z *y* v primeru, da z dodajanjem vrednosti *y* iz *B* dobimo n-terico *xy*, ki ni v *A*.

Izločene vrednosti *x*: $\pi_x((\pi_x(A) \times B) - A)$

A/B: $\pi_x(A)$ - Izločene vrednosti

Ekvivalence operacij RA

- *Selekcija:* $\delta_{c1 \wedge .. \wedge cn}(R) = \delta_{c1}(\ldots(\delta_{cn}(R)\ldots)$ (*Razcep*)

  $\delta_{c1}(\delta_{c2}(R)) = \delta_{c2}(\delta_{c1}(R))$ (*Komutativnost*)

❖ *Projekcija:* $\Pi_{a1,..,an}(R) = \Pi_{a1}(\ldots(\Pi_{an}(R)\ldots)$ *(Razcep)*

❖ *Stik:* $R \bowtie (S \bowtie T) = (R \bowtie S) \bowtie T$ *(Asociativnost)*

$(R \bowtie S) = (S \bowtie R)$ (*Komutativnost*)

☞ Dokaži: $R \bowtie (S \bowtie T) = (T \bowtie R) \bowtie S$

Ekvivalence operacij RA

- Spuščanje selekcije/projekcije proti listom.

$\delta(R \bowtie S) = \delta(R) \bowtie S$, če selekcija izbira samo atribute R.

$\Pi(R \bowtie S) = \Pi(R) \bowtie S$, če projekcija uporabi samo atr. R.

- Projekcija:
  - Projekcija je komutativna s selekcijo, ki uporablja atribute projekcije.
- Selekcija, ki vsebuje primerjavo atributov obeh argumentov kartezijskega produkta se prevede v stik.

Poišči imena mornarjev, ki so rezervirali ladjo #103

- Rešitev 1: $\pi_{mime}((\sigma_{lid=103}\text{Rezervacije}) \bowtie Mornarji)$

❖ Rešitev 2: $\rho(Temp1, \sigma_{lid=103}\text{Rezervacije})$

$\rho\ (Temp2,\ Temp1 \bowtie Sailors)$

$\pi_{mime}(Temp2)$

❖ Rešitev 3: $\pi_{mime}(\sigma_{lid=103}(\text{Rezervacije} \bowtie Mornarji))$

Poišči vse mornarje, ki so rezervirali rdečo ladjo.

- Podatki o ladjah se nahajajo v relaciji Ladje; potrebujemo še en stik.

$$\rho(Temp,(\sigma_{barva=rdeca} Ladje))$$

$$\pi_{mime}(Temp \bowtie Rezervacije \bowtie Mornarji)$$

- Kako bi pohitrili vprašanje?

Poišči vse mornarje, ki so rezervirali rdečo ali zeleno ladjo.

- Identificiramo vse rdeče in zelene ladje in potem poiščemo mornarje, ki so rezervirali eno izmed izbranih ladij:

$$\rho(Temp,(\sigma_{barva=rdeca \vee barva=zelena} Ladje))$$

$$\pi_{mime}(Temp \bowtie Rezervacije \bowtie Mornarji)$$

- ❖ Temp se da definirati z unijo! (Kako?)
- ❖ Kaj se zgodi, če je ∨ zamenjan z ∧ ?

Poišči mornarje, ki so rezervirali rdečo in zeleno ladjo.

- Prejšnji način ne deluje.
- Poiščemo mornarje, ki so rezervirali rdeče ladje, mornarje, ki so rezervirali zelene ladje in potem naredimo presek.

$$\rho(Temp1,\pi_{mid}((\sigma_{barva=rdeca}Ladje)\bowtie \mathrm{Re}zervacije))$$

$$\rho(Temp2,\pi_{sid}((\sigma_{barva=zelena}Ladje)\bowtie \mathrm{Re}zervacije))$$

$$\pi_{mime}((Temp1\cap Temp2)\bowtie Mornarji)$$

Poišči imena mornarjev, ki so rezervirali vse ladje

- Uporaba deljenja; sheme vhodnih relacij morajo biti pazljivo izbrane:

$$\rho(Temp,(\pi_{mid,lid}\text{Re}zervacije)/(\pi_{lid}Ladje))$$

$$\pi_{mime}(Temp \quad Mornarji)$$

❖ … mornarje, ki so rezervirali vse ladje ‘Delfin’:

$$\ldots\ldots \quad /\pi_{lid}(\sigma_{\text{lime}=delfin}Ladje)$$

Ponovitev

- Relacijski model ima formalne povpraševalne jezike, ki so enostavni in imajo veliko izrazno moč.
- Relacijska algebra je proceduralen jezik; uporabna je za interno predstavitev vprašanj.
- Veliko načinov za izražanje enega samega stavka; optimizator izbere tistega, ki poišče rezultat najhitreje.