



INFORMACIJE

Literatura

- Ramakrishnan, R., Gehke, J.: Database management Systems, 3rd Edition, McGraw-Hill, 2004
- Prosojnice
- Korth, F.H., Silberschatz, A., Sudarshan, S.: Database Systems Concepts, 6th Edition, McGraw-Hill, 2010

Ocena:

- Pisni izpit = 40%
- Domaće naloge = 40%
- Ustni izpit = 20%

UVOD

- SVPB - sistem za upravljanje podatkovnih baz
- Info. sistem temelji na relacijah - relacijski sistem
- Podatkovni modeli:
 - Hierarhični - starejši, nadomeščen z relacijskim,
 - Relacijski - uporaben za različne sisteme,
 - Objektni,...
- SVPB - podatki v tabelah,

Datoteke vs. SVPB

- Aplikacije prenašajo velike količine podatkov med dinamičnim spominom in diskom
- Posebna koda za posamezne poizvedbe
- Zaščita podatkov pred nekonsistenco, ki je lahko posledica večih hkratnih uporabnikov
- Zaščita pred izpadom sistema
- Varnost in kontrola dostopa

Zakaj SVPB?

- Aplikacija mora brati/pisati velike količine podatkov iz diska
 - Vmesni pomnilnik, bloki, učinkovit dostop do podatkov
 - Podatkovna neodvisnost
- Zmanjšan čas razvoja aplikacije
- Podatkovna integriteta in varnost
- Uniformno administriranje podatkov
- Hkraten dostop, transakcije zaščita pred sist. napakami

Podatkovni modeli

- Podatkovni model je zbirka konceptualnih gradnikov (jezik) za opis podatkov
- Sheme
- Relacijski podatkovni model

Abstrakcija

- Več pogledov, ena konceptualna shema in fizična shema
- Primer:

Konceptualna shema:

- Študenti (sid:string, ime:string, login:string...)
- Predmeti (pid:string, pime:string, točke:int)
- Upis (sid:string, pid:string, ocena:string)

Fizična shema

- Relacije shranjene v urejenih datotekah
- Index je def. na 1. stolpcu relacije Študenti

Zunanja shema (Pogled)

- Predmet-Info (pid:string, upisani:int)

Podatkovna neodvisnost

- Aplikacije se nekvirajo s tem kako so podatki shranjeni
- Logična PN: Zaščita pred spremembami v logični strukturi
- Fizična PN: Zaščita pred spremembami fizične strukture

Kontrola sočasnega dostopa

- Sočasno izvajanje uporabniških programov je bistveno za dobre performanse SUPB

Transakcija

- Osnovni koncept, ki je atomarna sekvence akcij SUPB
- Vsaka transakcija, ki je izvrši v celoti, mora pustiti PB v konsistentnem stanju, če je PB konsistentna

Razvrščanje sočasnih transakcij

- SUPB zagotavlja izvajanje v zaporedju
- smrti objem (Dead lock!)

Zagotavljanje atomičnosti

- SUPB zagotovi atomičnost, četudi se zgodi sistemska napaka
- WAL sistem - write ahead logging

Dnevnik (log)

- Shrani vse spremembe v PB

Struktura SUPB

- Tipična nivojska arhitektura
- Nivoji morajo omogočati kontrolo hkratnega dostopa in reševanje

RELACIJSKI PODATKOVNI MODEL

Zakaj študij relacijskega modela?

- Naj bolj široko uporabljen PM
 - Relacije imajo močne matematične osnove
 - System R, IBM, 1974 (first implementation of SQL)
 - 80% vseh SVPB je relacijskih
- 1980-1985: vzpon objektno-usmerjenih sistemov
 - ObjectStore, ObjectDataBase++, GemStone/S,...
 - Samo peščica OO SVPB na trgu leta 2000
- 1995-2000: Objektno-relacijski model
 - Relacije objektov so preslikane v ravninski relacijski model
 - Implementacije: Oracle, DB2, Sybase ...
- 2000-2010: NoSQL gibanje
 - Kaj se je zgodilo?
 - ↳ Masovni podatki, internetni inform. sistemi, nestrukturirani podatki, semi-strukturirani podatki, raziskovalni podatki, multimedija ...
 - ↳ Obstoječi SVPB nimajo primerne skalabilnosti in dostopnosti za delo z masovnimi podatki
 - Iztočasno:
 - ↳ Nove tehnologije: obsežen RAM, SSD diski, masovni HDDs, vzporedne in porazdeljene arhitekture, več procesorski sistemi, več jedrni procesorji, "samostojni" sistemi, ...
 - Rezultat:
 - ↳ Vzpon NoSQL sistemov

- Obstojeci NoSQL sistemi:
 - MongoDB, CouchDB, Dynamo, ...
- 2010-2020: NewSQL sistemi
 - Razvita tehnologija se uporabi v novih relacijskih SVPB!
 - Na novo razviti SVPB
 - Google: Megastore, Spanner, F1
 - Amazon: RDS

Relacijska PB

- Relacijska PB: množica relacij
- Relacija: 2 dela
 - Instanca: tabela, ki ima vrstice & stolpce
 $\# \text{vrstic} = \text{kardinalnost}$, $\# \text{stolpcev} = \text{stopnja}$
 - Shema: določa ime relacije ter imena in tipe vseh stolpcev

Relacijski povpraševalni jeziki

- Največja moč relacijskega modela
 - SQL enostaven povpraševalni jezik
- Vprašanja se lahko pišejo intuitivno *
- Razlog: natančna semantika
- Optimizator pogosto prevede operacije (logično ekvivalentna izvedba, ki vrne isti rezultat)

Povpraševalni jezik SQL

- Razvit pri IBM leta 1970
- Potreba po standardu

Kreiranje relacij v SQL

- Z besedo CREATE, specificiramo tipe za vse stolpce

Brisanje in spreminjanje relacij

- Brišemo z besedo DROP
- Spreminjamo z ALTER

Dodajanje in brisanje zapisov

- Ustavljanje zapisa z INSERT INTO
- Brišemo zapise z DELETE

Integritetne omejitve (10)

- 10: pogoji, ki mora biti izpolnjen za vsako n-terico relacije

*

Primarni ključ

- Množica atributov je ključ če:
 1. Ne obstajata dva enaka zapisa
 2. To nevelja za nobeno podmnožico
- Če 2. ne velja, imamo super ključ
- Lahko obstaja več ključev na za relacijo, primarnega izberemo

Primarni in kandidatski ključ v SQL

- Običajno je navoljo več kandidatskih ključev, imed njih izberemo enega primarnega
- Ob nepravilni uporabi 10 lahko onemogočimo vnos dejanskih podatkov iz realnega sveta

Tuji ključ in referenčna integriteta

- Tuji ključ: Mn. atrib. neke relacije, ki referencira zapise druge relacije

Zagotavljanje referenčne integritete

- Če brišemo iz ene tabele, nato izbrišemo iz druge tabele
- Zavrtnemo brisanje iz tabel

Od kod so prišle IO omejitve?

- IO so osnovane na pomem okolja, ki ga modeliramo
-

RELACIJSKA ALGEBRA

Relacijski povpraševalni jezik

- Povpraševalni jezik: Omogoča urejanje podatkov in proizvodovanje po podatkih v PB
- Relacijski model podpira enostavne PS z veliko izrazno močjo
- Povpraševalni jezik != programski jezik

Formalni relacijski PS

- Dva formalna (matematična) jezika tvorita osnovo za "realne" PS (npr. SQL)
 - Relacijska algebra
 - Relacijski račun

Osnove

- Poizvedba je izvršena nad instancami relacij in rezultat poizvedbe je instanca neke relacije
 - Shema vhodnih relacij - fiksna
 - Shema rezultata - fiksna \rightarrow določena s pravili gradniki T₃
- Notacija osnovana na poziciji oz. imenih atributov

Relacijska algebra

- Osnovne operacije
 - Selekcija (σ)
 - Projekcija (π)
 - Produkt (\times)
 - Razlika ($-$)
 - Unija (\cup)
- 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 attribute v listi projekcije iz relacije
- Shema rezultata vsebuje samo attribute, ki so v listi projekcije z istim imeni kot v vhodni relaciji
- Duplikati? Realni sistemi tipično ne odstranijo duplikatov, če uporabnik tega ne zahteva

Selekcija

- Izbere vrstice, ki ustrezajo pogoju
- Ni duplikatov v rezultatu
- Shema rezultata identična shemi vhodnih relacij
- Rezultat je lahko vhodna relacija drugi operaciji

Unija, presek, razlika

- Vse operacije so binarne in vhodni relaciji morata biti unija-kompatibilni
 - Enako št. atributov
 - "Pripadajoča" polja imajo enake tipe

Produkt

- Karteziski produkt: vsaka vrstica S1 se poveže z vsako vrstico R1
- Shema rezultata ima po en atribut za vsak atribut relacij
- Preimenuvanje

Stik (Join)

- S pogojem:

$$R \bowtie_c R = \sigma_c(R \times S)$$

- Shema rezultata: enaka kart. prod.
- Manj nteric kot produkt (izračuna hitreje)
- Theta-stik

Stiki

- Equi-stik: Poseben primer, izvede stik po skupnih atributih
- Shema rezultata: podobno karteziskemu prod., samo ena vrednost enačenih atributov je v rezultatu
- Naravni stik: Equi-stik po vseh skupnih atributih

Deljenje

- Ni osnovna operacija, uporabna za izražanje vprašanj
 - Poišči mornarje, ki so rezervirali vse ladje
- Naj ima A dva atributa x in y , B pa samo atribut y
 - $A/B = \{ \langle x \rangle \mid \exists \langle x, y \rangle \in A \ \forall \langle y \rangle \in B \}$
 - A/B vsebuje vse n -terice x (mornarji) tako, da za vsako n -terico y (ladja) v B , obstaja n -terica xy v A
- V splošnem sta x in y lahko poljubna seznama atributov; y je seznam atributov v B , in $x \cup y$ je seznam atributov v A .

Izražanje A/B z osnovnimi operacijami

- Deljenje ni nujno potrebna operacija; uporabna bližnjica
- Ideja: A/B = izračunaj vse vrednosti x , ki niso izločene z vrednostjo y v B

Ekvivalence operacij RA

- Selekcija: $\sigma_{a_1 \dots a_n}(R) = \sigma_{a_1}(\dots(\sigma_{a_n}(R))\dots)$ - Razcep
- $\sigma_a(\sigma_a(R)) = \sigma_a(R)$ - Komutativnost
- Projekcija: $\pi_{a_1}(R) = \pi_{a_1}(\dots(\pi_{a_n}(R))\dots)$ - Razcep
- Stik: $R \bowtie (S \bowtie T) = (R \bowtie S) \bowtie T$ - Asociativnost
- $R \bowtie S = S \bowtie R$ - Komutativnost

⇒ Dokazi $R \bowtie (S \bowtie T) = (T \bowtie R) \bowtie S$

- Spuščanje selekcije/projekcije proti listom
- $\sigma(R \bowtie S) = \sigma(R) \bowtie S$, če selekcija izbira samo attribute R
- $\pi(R \bowtie S) = \pi(R) \bowtie S$, če projekcija uporabi samo attribute R

RELACIJSKI RAČUN

Relacijski račun

- Dva jezika: n -terični RR (TRR) in domenski RR (DRR)
- Iznajir vsebujejo spr., konst., primerjalne operacije, logične operacije in kvantifikatorje
 - TRR: spremenljivke omejene na n -terice
 - DRR: $-11-$ na domene atributov
 - TRR in DRR so podmnožice predikatnega računa
- Iznajir imenljivo formule

N -terični RR

- Poizvedba ima obliko $\{T \mid p(T)\}$, kjer T edina prosta spr.
- Rezultat vsebuje n -terice T , za katere $p(T)$ vrne *true*
- TRR je rekurzivno def.

TRR Formule

- Atomarni izrazi:

$R \in \text{Rname}$ ali $R.a$ op $S.b$ ali $R.a$ op constant
op = $\{<, >, \leq, \geq, =, \neq\}$

Formule:

Atomarni izraz, ali

$\neg p, p \wedge q, p \vee q$ - p, q izrazi, ali

$\exists R(p(R))$, kjer n -terica R prosta v $p(R)$, ali

$\forall R(p(R))$, kjer n -terica R prosta v $p(R)$

- Uporaba kvantifikatorjev povezuje spremenljivko R

Domenški RR

- Vprašanje ima obliko: $\{\langle x_1, \dots, x_n \rangle \mid p(x_1, \dots, x_n)\}$
- Odgovor vsebuje n -terice $\langle x_1, \dots, x_n \rangle$ za katere vrne izraz $p(x_1, \dots, x_n)$ vrednost true.
- Formula rekurzivno def.

DRR Formule

- Atomarna formula
 $\langle x_1, \dots, x_n \rangle \in R_{\text{name}}$, ali $x \text{ op } y$ ali $x \text{ op constant}$
 $\text{op} = \{<, >, \leq, \geq, =, \neq\}$
- Formula
 - atomična formula, ali
 - $p \wedge q, p \vee q, \neg p$, kjer so p in q formule, ali
 - $\exists x(p(x))$, kjer spr. x prosta v $p(x)$, ali
 - $\forall x(p(x))$, kjer spr. x prosta v $p(x)$
- Uporaba kvantifikatorjev poveže spr. x