

O SNOVE

PO DATKOVNIH
BAZ

Predavanja

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%
- Domáče nálož = 40%
- Ustni izpit = 20%

UVOD

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

Datoteke vs. SUPB

- Aplikacije prenosajo velike količine podatkov med dinamičnim spominom in diskom
- Poselena koda za posamezne prijedelce
- Zaščita podatkov pred nekonsistencijo, ki je lahko posledica večih hkratnih uporabnikov
- Zaščita pred izpadom sistema
- Varnost in kontrola dostopa

Zakaj SUPB?

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

Podatkovni modeli

- Podatkovni model je zbirka konceptualnih građevina (jezik) za opis podataka
- Seme
- 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 neurejenih datotekah
- Index je def. na 1. stolpcu relacije Študenti

Zunanja shema (Pogled)

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

Podatkovna neodvisnost

- Aplikacije se nekvarjajo 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 performance SUPB

Transakcija

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

Razvrščanje sočasnih transakcij

- SUPB zagotavlja izvajanje v zaporedju
- Smrtni objem (Dead Lock!)

Zagotavljanje atomičnosti

- SUPB zagotovi atomičnost, četudi se izgodi sistemski napaka
- WAL sistem - Write ahead logging

Dnevnik (log.)

- Shrani vse spremembe v PB

Struktura SUPB

- Tipična nivojska arhitektura
- Nivoji morajo omogočati kontrolo hravnega dostopa in rezerviranje

RELACIJSKI PODATKOVNI KODEL

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
- 1980-1995: Vzpon objektno-usmerjenih sistemov
 - ObjectStore, ObjectDatabase, GemStone/S, ...
 - Samo petica 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
 - Izkosno:
 - ↳ Nove tehnologije: obvezen RAM, SSD disk, masovni HDDs, vzporedne in porazdeljene arhitekture, več procesorski sistemi, več jedrnih 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 & stolpcce
 #vrstic = kardinalnost, #stolpcov = stolpc
 - Shema: določa ime relacije ter imena in tipe vseh stolpcov

Relacijski poupravovalni jeziki

- Največja moč relacijskega modela
- SQL enostaven poupravovalni jezik
- Upravljanje se lahko pišejo intuitivno
 - Razlog: način na semantika *
 - Optimizator pogosto preuredi operacije (logično ekvivalentna poizvedba, ki vrne isti rezultat)

Poupravovalni jezik SQL

- Razvit pri IBM leta 1970
- Potreba po standardu

Kreiranje relacij v SQL

- Z besedo CREATE, specificirano type za vse stolpce

Brisanje in spremicanje relacij

- Brisemo z besedo DROP
- Spremujemo z ALTER

Dodajanje in brisanje zapisov

- Ustavljanje zapisa z INSERT INTO
- Brisemo zapise z DELETE

Integritetne omeitve (IO)

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

*

Primarni ključ

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

Primarni in kandidatni ključ v SQL

- Običajno je navoljo več kandidatnih ključev, ilmed njih izberemo enega primarnega
- Ob nepravilni uporabi IO lahko onemogočimo vnos dejanskih podatkov iz realnega sveta

Tuji ključi in referenčna integriteta

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

Zagotavljanje referenčne integritete

- Če brišemo iz ene tabele, nato izbrisemo iz druge tabele
- Zavrnemo brisanje iz tabel

Od kod so prišle 10 omejitve?

- 10 so osnovane na pomen okolja, ki ga modeliramo
-

RELACIJSKA ALGEBRA

Relacijski poupravevalni jezik

- Poupravevalni jezik: Omogoča urejanje podatkov in poizvedovanje po podatkih v PB
- Relacijski model podpira enostavne PJ z veliko izrazno močjo
- Poupravevalni jezik != programski jezik

Formalni relacijski PJ

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

Osnove

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

Relacijska algebra

- Osnovne operacije
 - Selekcija (δ)
 - Projekcija (π)
 - Produkt (\times)
 - Razlika (-)
 - Unija (U)
- Dodatne operacije
 - Presek, stik, deljenje, preimenovanje
 - Niso ujne, so pa zelo koristne
- Vsaka operacija vrne relacijo kot rezultat
 - Operacije se lahko sestavljajo - funkcionalni jezik

Projekcija

- Izbere attribute v listi projekcije iz relacije
- Shema rezultata vsebuje samo attribute, ki so v listi projekcije z istimi imeni kot v vhodni relaciji
- Duplicati? Realni sistemi tipično ne odstranijo duplicates, če uporabnik tega ne zahteva

Selekcija

- Izbere vrstice, ki ustrezajo pogoju
- Ni duplicates 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
 - "Prispadajoča" polja imajo enake fipe

Produkt

- Karteristični produkt: vsaka vrstica S_1 se poveže z vsako vrstico R_1
- Shema rezultata ima po en atribut za vsak atribut relacij
- Preimenovanje

Stik (Join)

- S pogojem:

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

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

Stiki

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

Definicije

- Ni osnovna operacija, uporabna za izražanje vprašanj
 - Poisci mornarje, ki so rezervirali vse ladje
- Naj ima A dva atributa x in y , B pa samo atribut y
 - $A/B = \{ \langle x \rangle \mid \exists \langle x, y \rangle \in A \wedge (y) \in B \}$
 - 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 počitoma sestavni atributorji, q je sestavni atributou v B, in $x \cup y$ je sestavni atributou v A.

Izražanje A/B z osnovnimi operacijami

- Definicije ni nujno potrebna operacija; uporabna bljivosten
- Ideja: $A/B =$ izračknaj vse vrednosti x , ki niso izločene z vrednostjo y v B

Ekvivalenze operacij RA

- Selekcija: $\delta_{c_1, \dots, c_n}(R) = \delta_{c_1}(\dots(\delta_{c_n}(R)\dots))$ - Razcep
 $\delta_{c_1}(\delta_{c_2}(R)) = \delta_{c_2}(\delta_{c_1}(R))$ - 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 profi 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 atribute R

RELAČIJSKI RACUN

Relacijski račun

- Dva jazika: n-terični RR (TRR) in domenski RR (DRR)
- Kraji vsebujejo spr., konst., primerjalne operacije, logične operacije in kvantifikatorje
 - TRR: spremenljivke omajene na n-terice
 - DRR: -/- na domene atributov
 - TRR in DRR so podmnogice predikatnega računa
- Kraje imenujemo 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:

RERname ali R.a op S.b ali Ra op constant
op = { $<$, $>$, \leq , \geq , $=$, \neq }

Formule:

Atomarni izraz, ali

$\neg p$, $p \wedge q$, $p \vee q$ — p,q izraza, 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

DomenSKI RR

- Upravičanje 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

$\langle x_1, \dots, x_n \rangle \in R_{name}$, ali x op Y ali x op constant
Op = { $<$, $>$, \leq , \geq , $=$, \neq }

- Formula

- atomična formula, ali

- $p_1 q, p \vee q, \neg p$, 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

QUERY-BY-EXAMPLE (QBE)

28.10.24

Vvod

- "GUI" za poizvedovanje
- Osnovan na DRR
- Narejen pred GUI
- Primeren za enostavna upražnjenja

*



DISKI IN DATOTEKE

Diski in datoteke

- SPUTB shranjuje podatke na diskih
- To ima veliko posledic na zasnovu SPUTB
 - READ: prenos pod. med diskom in RAM
 - WRITE: prenos med RAM in diskom
 - Obe operaciji časovno potratni zato se mora njihova poteka planirati

Zakaj ne vsega shramti v dinamični spomin

- Prevelika cena
- Dinamični spomin ne shranjuje podatkov

*

4.11.24

Pomnilnik SPUTB

- Podatki v datotekah
- Znotraj datoteke zapisi, ki so urejeni po straneh
- Dostop preko naslovov
- Umesni pomnilnik - okvir, kjer so shranjene strani

Urediten blokovi na disku

- Pomemben koncept: „Naslednja“ stran
 - Bloki na isti sledi, ki sledijo
 - Bloki na istem cilindru, ki sledijo
 - Bloki na sosednjem cilindru
- Bloki naj bi bili organizirani sekvenčno na disku, da \rightarrow minimizira čas iskanja in rotacijska zakasitev
- Sekvenčno skeniranje: branje v naprej, t.j. več sekvenčnih strani se prebere v naprej, pridobitev na času

Delo z diskovnim prostorom

- Nižji nivo SUPB dela z diskovnim pomnilniškim prostorom
- Višji nivoji zahtevajo od nižjega alokacijo/dealočacijo strani in branje in pisanje strani
- Zahteve po sekvenci strani izvršena, da sistem alocira stran v enem sekvenčnem prostoru.

Vmesni pomnilnik v SUPB

- Podatki morajo biti v RAM-u, da SUPB lahko dela z njimi
- Uporablja se tabela parov: $\langle \text{okvir\#}, \text{stranID} \rangle$
- Problemi:
 - lahko se hitro napolni

Zahteva po strani

- Če ni v bazu:
 - izberi okvir za zamenjavo
 - če je okvir „umazan“ potem se mora zapisati na disk
 - Preberi izbrano stran v izbrani skvir
- Tripi stran in vrni njen naslov

Strategije za zamenjavo strani

- Okvir izbran s strategijo zamenjave
 - LRU, Ura(Clock), MRU, itd.
- Strategija ima lahko velik učinek na # I/O operacij
- Sekvenčno prelivanje
 - Grota situacija povzročena z LRU + ponavljajoč sekvenčni pregled tabele
 - # okvirjev < # str. datoteke vsaka zahteva povzroči I/O. MRU je v tem primeru boljša

SUPB vs OS file system

- OS ureja diskovni prostor & vmesni pomnilnik
Zakaj ne bi OS za SUPB urejal ta pravila?
- Zarlike v OS podpori: prenosljivost
- Nekatere omejitve, npr. datoteke se ne morajo raztezati preko velikosti diska
- Delo z vmesnim pomnilnikom v SUPB zahteva zmožnost:
 - Prijeti stran v vmesni pomnilnik, prisiliti shranjevanje na disk
 - Prilagoditve strategije zamenjave strani in branje strani v naprej na osnovi obnašanja tipične operacije

Format zapisa: Fiksna dolžina

- Informacija o tipih polja je enaka za vse zapise
v datoteki
- Poisci i-to polje ne zahteva pregled vseh zapisov

Format zapisa: Variabilna dolžina

- Dva alternativna formata
- Polja razmejena s poslednjim simbolom

*

*

Neurejena datoteka z direktorijem strani

- Zapis podstrani na glavni strani vsebuje tudi
st. prostih zvezgov na podstrani
- Direktorij je zbirka strani, implementacija vernamou
je ena alternativa

Sistemski katalogi

- Za vsak indeks:
 - Struktura in iskalni ključi
- Za vsako relacijo:
 - ime, ime datoteka, datotečna struktura
 - imena in tipi atributov
 - imena indeksov
 - Integrirane omogitve
- Za vsak pogled
 - ime pogleda in definicija
- Statistika, autorizacija, velikost barema strani

PREGLED INDEKSOV

Podatki na pomnilniških medijih

- Diski imajo direkten dostop do blokov
- Trakovi imajo sekvenčni dostop do strani
- Dajte ena organizacija: Organizacija podatkov na pomnilniških medijih
- Arhitektura: Umesni pomnilnik

Alternativne datotečne organizacije

- Več alternativ
 - Neurejene datoteke
 - Sortirane datoteke
 - Indeksi

Indeksi

- Indeks poliftri iskanje zapisov na osnovi polj iskalnih ključev
- Indeks vsebuje množico podatkovnih vpisov
 $k^* = k + \text{rid}$
- Indeks podpira učinkovito iskanje podatkovnih vpisov
 k^* z dano vrednostjo ključa k .

Podatkovni opis k^*

- Lahko je:
 - Podatkovni zapis z vrednostjo ključa k
 - $\langle k, \text{rid} \rangle$
 - $\langle k, \text{seznam rid} \rangle$
- Izberi alt. za pod. vpise je ortogonalna izberi indeksne tehnike za iskanje vpisov

Podatkovni vpis - možnosti

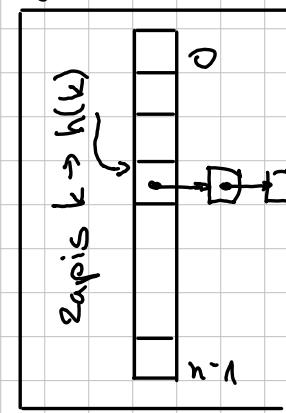
1. Datotečna struktura; samo en indeks lahko uporabi možnost 1; v primeru da podatkovni zapisi zasedajo veliko prostora, je št. strani, ki vsebujejo podatke, veliko
2. + 3. Podatkovni vpisi so tipično precej manjši kot pod. zapisi, možnost 3 bolj optimalna od možnosti 2 glede prostora

B+ drevo

- Listi vsebujejo podatkovne vpise, ki so urejeni
- Notranje strani imajo indeksne vpise

Razpršilni indeksi

- Dobra rešitev za selekcijo z enakostjo
- Indeks je zbirka skupkov
 - Skupek = primarna str. + niti ali več prelivnih strani
 - Skupek vsebuje podatkovne vpise
- Razpršilna fja $h: h(r) = \text{skupek } k$, kateremu pripada podatkovni zapis r .
 h uporabi polja iskalnega ključa relacije r



Klasifikacija indeksov

- Primarni vs sekundarni
 - Če iskalni ključ vsebuje primarni ključ potem indeks imenujemo primarni
 - Unique indeks: iskalni ključ je kandidatni ključ
- Poverzan vs Nepoverzan

*

DREVESNI INDEKSI

Prenesni indeksi

- Drevesno strukturirani indeksi podpirajo iskanje po enakosti in po področju
- ISAM:
 - Statična struktura
 - S časom drevo postane zelo neuravnoteženo
- B+tree
 - Dinamična struktura
 - Po polnemu uravnoteženo

Iskanje področja

- Če so podatki v urejeni datoteki naredimo binarno iskanje in nato preglej
- cena bin. iskanja je visoka

ISAM

- Indeksna datoteka je vseeno lahko velika
- Listi vsebujejo podatkovne v pise
- Ostajajo prazne prelivne strani

B+tree

- Kompleksnost op. Ustavi/2brisi = $\log_F n$ ($F = \text{fanout}, n = \text{št. listov}$)
- Drevo zelo uravnotežje
- Indeksne strani vsaj 50% polne

11.11.24

- Vsako vozlišče vsebuje $d \leq m \leq 2d$ vpisov (d imenujemo stopnja drevesa)
- Učinkovito podpira iskanje po enakosti in iskanje področja

Ustavljanje pod. vpisa v B+ drevo

- Poisci list L
- Ustavi pod. opis v L
 - Če L ni poln končano
 - Če L poln razcepi v L₁ in L₂, prepisi srednji ključ gor in porazdeli ključe
- To se lahko zgorodi rekurzivno
- Razcepi širijo drevo, razcep končna ravnica dreva

Brisanje pod. vpisa iz B+ drevesa

- Začni pri korenem in poišči list z vpisom
- Izbrisí vpis
 - če L usaj pač poln zaključi
 - če vsebuje L d-1 vpisov
 - Poskusimo porazdeliti vse zapise tako, da si jih sposodimo od sosedov
 - Drugače zlji h in soseda
- V primeru zlitja moramo izbrisati zapis, ki kaže na na L ali soseda iz starša

Stiskanje s predpohani ključev

- Pomembno je povzeti fan-out
- Vrednosti ključev v indeksnih vpisov edino usmerjajo
- Pogosto jih lahko stisнемo
- Ustavljanje/Brisanje mora biti prilagojeno

Masovno polnjenje

- Hitrejše od ustavljanja posameznih el., če je podatkov veliko
- Inicializacija: uredimo podatkovne opise, dodaj bazalec na prvo stran v novem korenem

RAZPRŠLJNI INDEKSI

Uvod

- Razpršlji indeksi so najboljši za iskanje po enakosti
- Ne podpirajo iskanja po področju
- Obstajajo statični in dinamični razpršlji indeksi
- Komprohisi podobni ISAM vs. B+ drevesom

Statični razpršlji indeks

- Primarne stvari so fiksne, zasežene sekventno ter niso nikoli sproščene
- $h(k) \bmod N = \text{skupek}$, k kateremu pripada pod-upis s ključem k ($N = \# \text{ skupkov}$)
- Skupki vsebujejo pod-upise
- Razpršljna fja se uporabi na iskalnem ključu.
Vrednosti se morajo razpršiti na $[0, N-1]$
 - $h(k) = a \cdot k + b \Rightarrow$ običajno deluje dobro
 - a in b konstanti
- Razvije se lahko dolga veriga prelivnih strani, kar poslabša učinkovitost pod-strukture

Razširjeni razpršlji indeksi

- Dinamični razpršlji indeks
 - Branje in pišanje skupkov potратno
- Ideja: Direktorij kazalcev na skupke
 - Po dvojemu $\# \text{ skupkov}$ (strani) s podvojitoji dir in razcepimo samo skupek, ki je napolnjen
 - Dir je veliko manjši kot celotna datoteka zato se podvojitev izvrši hitro (N je prelivnih strani!)
 - Trik je v predmetni razpršljive fič

Linearni razpršilni indeks

- Shema z dinamičnim razpr. indeksem
- Lin. razpr. indeks reši problem z dolgimi prelivnimi stranmi
- Ideja: uporabimo družino razpr. fij: h_0, h_1, \dots
 - $h_i(\text{key}) = h(\text{key}) \bmod (2^i N)$; $N = \text{čačno št. skupkov}$
 - h je neka razpr. fja (zaloga vrednosti $n_i \in [0, N-1]$)
 - Če $N=2^d$, za nek d , h_i aplicira h in gleda zadnjih d_i bitov, kjer $d_i = d_{\text{str}}(i)$ ($i = \text{level}$)
 - h_{i+1} podvoji zalogo vrednosti h_i (podobno podvaj. dir-a)
- Dir-u se izognemo z uporabo prelivnih str. in s cikličnim razcepljanjem skupkov
 - Razcepljanje poteka v ciklakh (runda so eakfjuci, ko so vsi skupki razcepljeni)
 - Trenutna št. runde je level
 - Iskanje: Da bi poiskali skupek r , poišči $h_{\text{level}}(r)$
 - Če $h_{\text{level}}(r)$ je v področju $[N_{\text{left}}, N_{\text{right}}]$, smo našli
 - Oicer, r (a hko priпадa skupku $h_{\text{level}}(r)$ ali skupku $h_{\text{level}}(r) + N_{\text{right}}$). Aplicirati je potrebno $h_{\text{level}+1}(r)$, da bi izvedeli

Linearno razprševanje

- Vstavi: Poisci skupek z uporabo h_{level} / $h_{\text{level+1}}$
 - Če skupek polni:
 - dodaj prelivne strane in vstavi
 - (Mogoče) Razcepi Next in ga povečaj ($\text{Next} \leftarrow 1$)
 - Izberemo lahko bateri kolikor kriterij za razcepljanje
 - Ker skupki razcepljeni ciklično se ne morejo razviti dolge prelivne verige
 - Podvojevanje dir-a je razširjivem razprševanjem podobno
 - Preblop med razpršilnimi fijami je impliciten s posiceranjem

LH je varianta ETI

- Shemski podobni
 - Začni z EH indeksom, ki ima dir z N elementi
 - Uporabi prelivne strani in razcepi skupke cílicno
 - Najprej razcepi skupko O
- Dir se podvojuje postopoma, skupki se breirajo fizíčno zaporedno na disku

EVALUACIJA RELACIJSKIH OPERACIJ

Pogosto uporabljene tehnike

- Algoritmni za evaluacijo:
 - Indeksiranje - za izbor majhnega št. n-teric
 - Iteracija - včasih hitreje pregledamo vse n-terice
 - Particije
 - Sortiranje

Statistika in katalogi

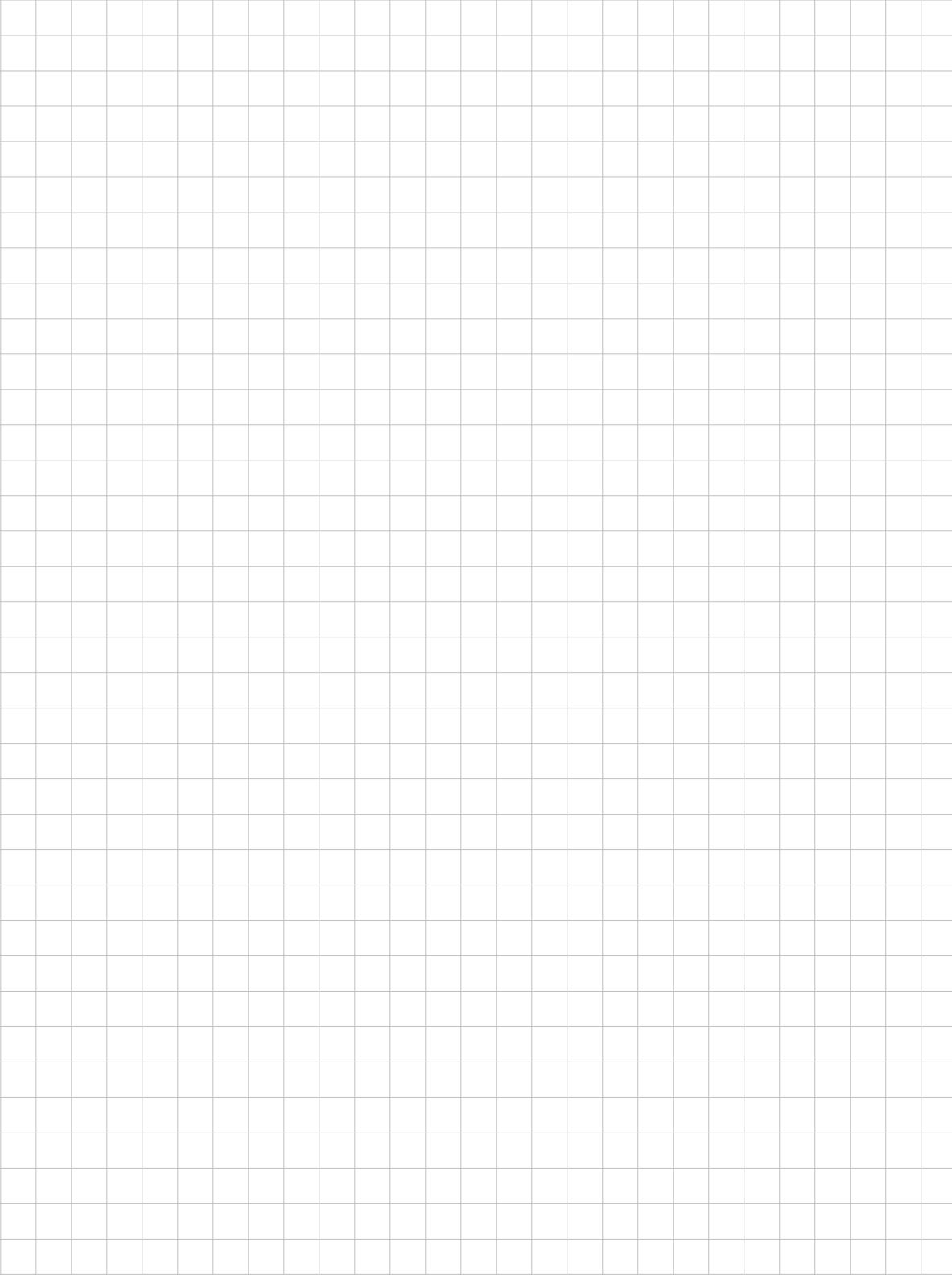
- Katalogi vsebujejo vsaj:
 - # zapisov, # strani,

*

Dostopna pot

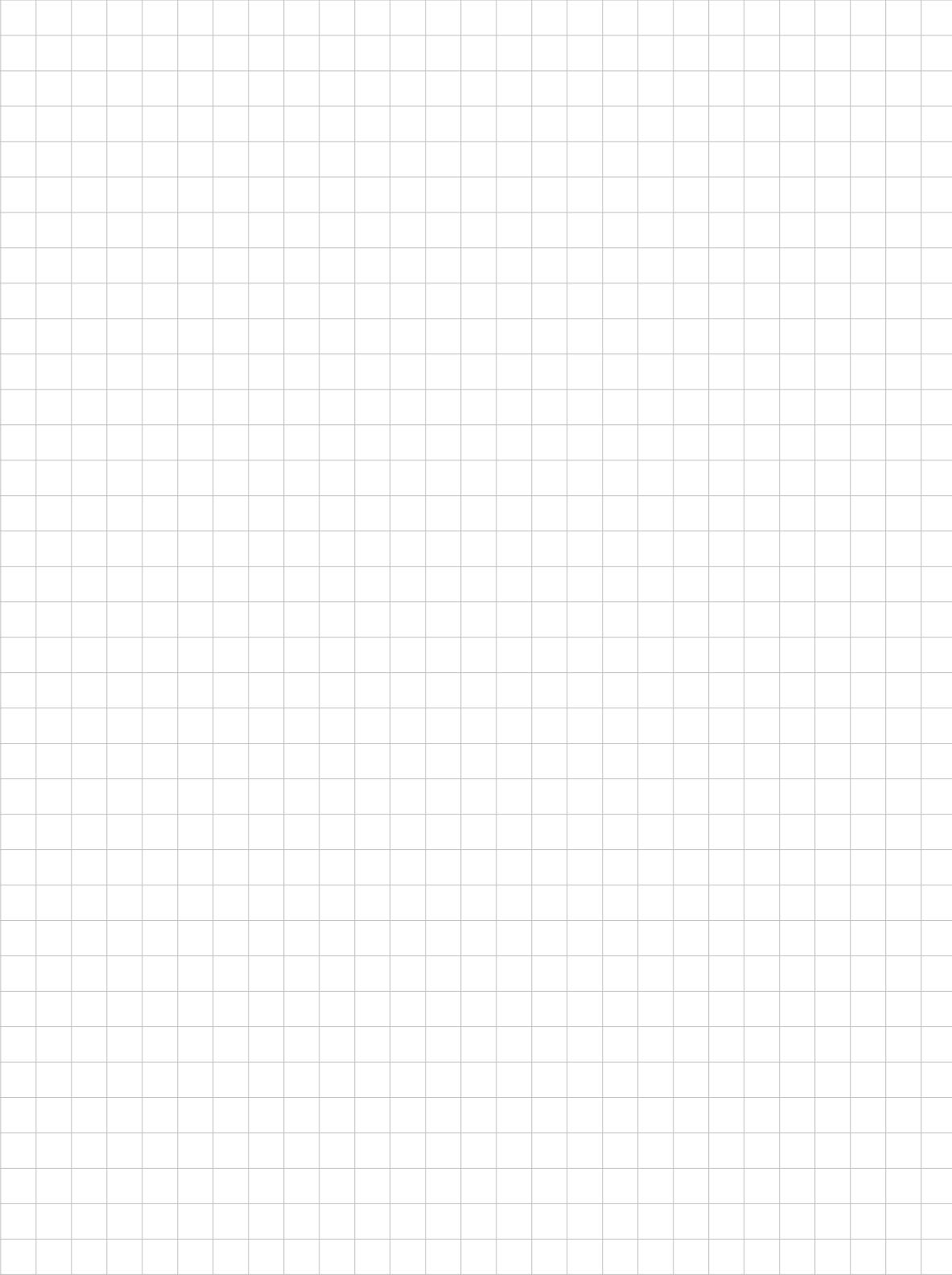
- Metoda za branje zapisov
 - Pregled datoteke, ali indeks, ki se ujema s selekcijo

Sortir au jeu
*



Dva pristopa k splošni selekciji:

1. Poisci najbolj selektivno dostopno pot, preberi zapise z izbranim DP in uporabi preostale pogoje, ki se ne ujemajo z indeksom
2. Če imamo na razpolago dva ali več indeksov z alternativama (2) ali (3) za pod. vpise





Spolzni pogoji stika

- Enakost večih atributov:

x

Pogoji neenakosti

x

Operacije nad množicami

- Presek in kartesijski produkt posebno
- Unija in razlika se implementirata podobno
- Unija s sortiranjem
- Unija z razprš. fjo

Agregacijske operacije

- Brez grupiranja
 - zahteva pregled relacije
 - pregled možen samo z indeksom
- Z grupiranjem
 - Sortiramo po group-by atributih, potem pregled
 - Podobno na porazdelitvi po group-by
 - Z okrevesnim indeksom

Upisi izrazenalnika

- Če več operacij v izvajaju je ocenjevanje prostih str. le predvidovanje

OPTIMIZACIJA POZVEDB

Osnove

- Potek:
 - Plan izvajanja
 - Prevajanje SQL v RA
 - Ekvivalenca operacij RA
 - Ocena plana
 - Def. problema
 - prostor rešitev
 - Tipični relacijski optimizator
 - Načrtovanje alter. planov
 - System R

Program v SUTB

s1 *

- s2 • Dobimo drevesno strukturirani cevovod po katerem se pretekajo n-terice

Optimizacija drevesa operacij

- Logična optim.:
 - Isčemo izraz RA, ki izvede najhitreje
 - Zato potrebno določiti fizično implementacijo drevesa RA
- Fizična optim.:
 - Uporabimo fizično alg. relacij
 - Isčemo optimizirane alg.
 - Fizična opt. se prepela z logično

Prevajanje SQL v RA

- Dekompozicija poizvedbe v bloke
 - Celoten SQL stavek se razdeli v bloke
 - *

Ekvivalenca operacij RA

- Zapis izraza RA določa drevno operacijo RA
- *

- Projekcija je komutativna s selekcijo, ki uporablja atribute projekcije
- Selekcija, ki vsebuje primerjavo afr. oben. arg. karterijskega prod. se prevede v stik
- Komutativnost Π in σ s \bowtie ; potiskanje selekcije/projekcije proti listom

Cena planov

- Ocena vedno približek
- Statistika shranjena v sist. katalogih in uporabljena za oceno
- Definirana vsaka operacija posebej
- Vrame v upoštev CPU in I/O

Statistike in katalogi:

- Podatki o relacijah in indeksih v podatkovni bazi
- Katalog: običajno vsebujejo: # n-teric, # strani relacije, # različnih vrednosti ključev, # strani za indekse, višino indeksa ...
*

Ocene selektivnosti pogojev operacij RA

- Predpostavke:
 - Vrednosti atr.. so enakomerno porazdeljene
 - Pogoji: so med seboj neodvisni
- Selekcija:
 - Pogoj $A = \text{vrednost}$ ima $SP = 1/n\text{Keys}(A)$
 - Pogoj $A = A_2$ ima $SP = 1/\text{Max}(n\text{Keys}(A_1), n\text{Keys}(A_2))$
 - Pogoj $A > \text{value}$ ima $SP = (\text{high}(1) - \text{value}) / (\text{high}(1) - \text{low}(1))$
 - $SP(p(A_i) \wedge p(A_j)) = SP(p(A_i)) \cdot SF(p(A_j))$
 - $SP(p(A_i) \vee p(A_j)) = SP(p(A_i)) + SP(p(A_j)) - (SP(p(A_i)) \cdot SP(p(A_j)))$
 - $SP(A \in \{\text{value}\}) = SP(A = \text{value}) \cdot \text{Card}(\{\text{values}\})$
- Projekcija:
 - Selektivnost proj. je $SP = 1$
- Stik

*

- Selekcija: $\text{card}(\sigma_F(R)) = SP(F) \times \text{card}(P)$
- Projekcija: $\text{card}(\pi_A(R)) = \text{card}(R)$
- Kart. prod: $\text{card}(R \times S) = \text{card}(R) \cdot \text{card}(S)$
- Stik:
 - Unija:
 - Log. meja: $\text{card}(R \cup S) = \text{card}(R) + \text{card}(S)$
 - Sp. meja: $\text{card}(R \cup S) = \max\{\text{card}(R), \text{card}(S)\}$

- Parlika:

- Zg. meja: $\text{card}(R-S) = \text{card}(R)$
- Sp. meja: \emptyset

Cena dostopne poti

- Odvisna od:

- Selek. atomarnih pogojev uporabljenih za dostopne poti
- Obstojecih indeksov

- Cena branja dela tabele iz diska z uporabo izbrane poti

Ocena za plane z eno relacijo

*

Ocena planov nad več relacijami

*



2.11.24

Tipični relacijski optimizator

Nastevanje alt. planov

1. Poizvedbe nad 1 relacijo

- Kombinacije selekcij, projekcij, ...
- Pregled vseh možnih DP
- Izberemo metodo z najnižjo ceno

2. Poizvedbe nad večjimi operacijami

- Prostor je prevelik in se ga pogosto omeji
- Pregled dela prostora
 - Odvisen od algor.
 - System R - levi usmerjeni plani
 - Zig-zag drevesa

Nastevanje levi usmerjenih planov

- Razlikujejo se po vrstnem redu relacijskih metodi dostopa za vsako relacijo, izvedbo stika
- Izberemo najboljše plane na mn. relacij
- Ko algor. išče najboljše plane za K relacij:
 - Vzamemo najb. plane za vse stike K-1 relacij
 - Dodamo eno izmed preostalih relacij
- Za vsako podmn. relacij iz K relacij je potrebno shraniti samo najcenejši plan za vsako zanimiv uterenost n-teric

- Order by, Group by ... uporabimo v zadnji fazi
- Plan za N-1 relacij se ne kombinira z dodatno relacijo, če ni pogoja stika med izreki WHERE stanka
- Načinjib omenjenem prostoru rešitev je predstavljena pristop če vedno eksponenten v ēt. tabel

Plan 1 (brez indeksov)

- Osnovna razlike: spusti selekcije
- Cesta veliko bolj ngodna
- Nepremično vseh atributov uferic

Plan 2 (z indeksi)

- Pohitritev selekuje nad rezervacijami
- Pohitritev stika
- Pregled rezervacij
- Stik z indeksom

Vgneredene poizvedbe

- Vgnereden blok se optimizira neodvisno
- Zunanji blok je optimiziran s ceno "felicanje" vgnedenega bloka
- Implicitna urejenost vgnedenih blokov omogočuje pregled nekaterih dobitnih poizvedb
- Poizvedbe brez gnedjenja boljše

PREGLED UPRAVLJANJA TRANSAKCIJ

Transakcije

- Sočasno izvajanje upor. programov je ključno za dobre performanse SVPB
- Upr. program lahko izvrši več op. nad prebranimi pod.
- Transakcija je abstr. pogled SVPB na upr. prog.

Sočasnost v SUPB

- Uporabniki si pri delu lahko predstavljajo, da se vsaka transakcija izvaja posebej
 - SUPB sočasnost doseže z izmenjevanjem akcij več transakcij *

Atomičnost transakcij

- Transakcije je lahko zaključena ali prekinjena
- čelo pomembna lastnost, ki jo zagotavlja SUPB je Atomičnost transakcije. SUPB vse zapise v dnevnik tako, da jih je mogoče izvršiti.

Urejanje transakcij

- Zaporeden razpored: dve transakcije se ne prekrivata
- Ekvivalenten razpored: rezultat izvajanja 1. rasporeda 2. razporedu
- Zaporedna uredljivost: razpored akcij je ekvivalenten nekemu zap. rasporedu

Anomalije

- Branje nepotrjenih podatkov
- Neponovljivo branie
- Prepis nepotrjenih podatkov

Kontrola sočasnosti z zaklepanjem

- Protokol doslednega 2 farnega zaklepanja (2F2)
- striktno 2F2 dovoli le zap. uredljive razporede

Dnevnik

- Transakcije v dnevnik zapiše staro in novo vrednost ter počrti/prekine
- Zapisi dnevnika so povezani z ID transakcij
- Dnevnik se pogosto podvoji in archivira na stalnem medij
- Vse aktivnosti nad dnevnikom izvaja SUTB transparentno

Obnovitev po razkritvi

- Alg. Aries ima 3 faze:
 1. analiza
 2. ponovi(redo)
 3. razveljavi(undo)

KONTROLA VZPOREDNOSTI

Konfliktno uređljivi rasporedi

- 2 razp. sta konfliktno ekvivalentna, če:
 - vsebujejo enake akcije na istih rasporedih
 - vsak par konf. akcij je urejen na isti način
- Raspored je konf. uređljiv, če je s konf. ekvivalentem

Zaporedna uređljivost po okvirih

- Razponedla S1 in S2 sta ekvivalentna po okvirih:
 - Če Ti preberes zač. vrednost A v S1, potem Ti preberes zač. vrednost A tudi v S2
 - Če Ti preberes vrednost A, ki jo napiše Ti; v S1, potem Ti preberes vrednost A, ki jo Ti; napiše v S2
 - Če Ti napiše končno vrednost A v S1, potem Ti napiše končno vrednost A tudi v S2

Dosledno 2F2

*

Prekinutes transakcije

- Če je transa. prekinjena se vse akcije izničene

*



Smetni objem

- Cikelj transakcij, ki čakajo druga na drugo na odklep virov
- 2 načina obravnavanja:
 - Pretečevanje SO
 - Detekcija SO

Pošumnne tehnike zaklepanja

- Dinamične podatkovne baze
 - Fantomski problem
 - Če zprostimo, da je TB fiksna kolekcija objektov potem tudi struktura ZFZ ne zagotovi serializabilnosti
 - Probleme lahko rešimo z:
 - Zahle pa njen indeksa
 - Zaklepanjem predikatov
- UEE-nivojsko zaklepanje
 - Težko se je oddeliti, kateri nivo zakleniti
 - Ne bi se smeli oddeliti o tem
 - Podatkovni „kontainerji“ so ugnedeni
- Zaklepanje BT dreve
 - Enostaven alg. za zaklepanje dreves
 - Toči: začne pri korenih in na poti do lista vedno zaklepi o troka z S zaklepom in odkleni starejšo
 - Ustavi/Brisi: začni pri korenih in potuj navzdol z X zaklpi po poteku. Ko je otrok zaklenjen, preveri, če je varen
 - Varno vozlišče: vozlišče, ki se spremeni, spremembce ne bo širilo na drug
 - Boljši alg.
 - Iskanje: kot prej
 - Ustavi/Brisi: kot prej, razen če list ni varen sprosi vse zaklape in uporabi prejšnje Ustavljanje/Brisanje
 - Spekulira, da bo spremenil samo list

Optimistična kontrola vzporednosti

Kung-Robinson model

- Xact ima tri faze:
 - Read: Xact prebere iz PB vendar naredi spremembe na lokalni kopiji objektov
 - Validate: Preveri konflikte
 - Write: Naredi lokalne kopije sprememb javne

Problemi optimistične kontrole vzporednosti

- Mora vežiti vse aktivnosti branja/pisanja
- Mora preveriti konflikte pred validacijo in mora izvajati validiranje pisanih globalno
- OK V ponovno počne Xact

Optimističen 2F2

- *

KV s časovnimi īgi

- Privedi vsakemu objektu vrahui časovni īig in pisalni časovni īig
- Privedi vsaki Xact čas. īigi ko se začne