

[P273] 9

# Пројектовање база података

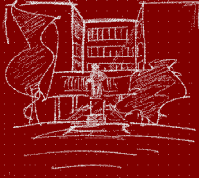


Саша Малков  
Универзитет у Београду  
Математички факултет  
2023/2024

[P273]

# Пројектовање база података

Саша Малков




Тема 9

## Физички модел базе података

[P273] Пројектовање база података – Саша Малков – 2023/24 – час 9 1

Физичко моделирање

## Физички модел базе података




- Физички модел је најнижи модел БП
- Описује конкретну имплементацију
- Узима у обзир многе техничке аспекте имплементације
  - У другом плану су аспекти важни за концептуални и логички модел
  - Потребна су другачија знања и вештине – више техничка него пословна
- Често се назива и „моделирање података“ (енгл. *data modeling*)
- Данас ћемо да обрадимо већи број појмова и проблема у вези са физичким моделирањем

Универзитет у Београду – Математички факултет

[P273] Пројектовање база података – Саша Малков – 2023/24 – час 9 2

Физичко моделирање

## Елементи физичког модела



- Структуре података
  - Уобичајено је да се на физичком нивоу говори о табелама и колонама, а не о релацијама и атрибутима
- Интерна (физичка) организација података
  - простори за табеле, контејнери, странице, бафери,...
- Помоћне компоненте
  - индекси

Универзитет у Београду – Математички факултет

[P273] Пројектовање база података – Саша Малков – 2023/24 – час 9 3



## Елементи физичког моделирања

- Основни послови при физичком моделирању су:
  - Процена оптерећења
  - Одређивање физичке организације података
  - Управљање радном меморијом



## Процена оптерећења

- Да бисмо могли да проценимо оптерећење и перформансе, морамо да имамо:
  - модел обраде података (енгл. *process model*)
  - неструктурне захтеве
  - матрицу ентитета и процеса (ПЧМБ)
  - захтеване перформансе
  - препознате конкретне циљне имплементације СУБП
  - препозната евентуална ограничења простора
  - препознате евентуалне развојне проблеме



## Модел обраде података

- Околности додавања нових редова
- Околности ажурирања постојећих редова
- Околности брисања редова
- Околности читања редова



## Модел обраде података (2)

- **Околности додавања нових редова**
  - колико редова у просеку (нпр. дневно)
  - колико редова при највећем оптерећењу (нпр. у секунди)
  - да ли су примарни кључеви слични и да ли зависе од времена
- ...



## Модел обраде података (3)

- ...
- **Околности ажурирања постојећих редова**
  - колико редова у просеку (нпр. дневно)
  - колико редова при највећем оптерећењу (нпр. у секунди)
  - колика је вероватноћа да се редови са сличним ПК истовремено користе (због закључавања)
- ...



## Модел обраде података (4)

- ...
- **Околности брисања редова**
  - колико редова у просеку (нпр. дневно)
  - колико редова при највећем оптерећењу (нпр. у секунди)
  - да ли се бришу појединачно или у групама
- ...



## Модел обраде података (5)

- ...
- **Околности читања редова**
  - учесталост читања
  - колико редова се чита једним упитом
  - које колоне се користе за одабир редова
  - које друге табеле се често користе заједно са посматраном



## Неструктурни захтеви

- **Трајање података**
  - колико дуго се подаци задржавају у табели пре брисања или архивирања?
- **Обим података**
  - колико ће редова бити у табели при пуштању у рад и како ће се број редова мењати током времена?
- **Расположивост података**
  - да ли су подаци потребни стално или повремено, колико често и дуго подаци могу/смеју да буду недоступни корисницима?
- **Ажураност података**
  - колико ажурни морају да буду подаци који се користе?
  - (да ли могу да се раздвоје копије за мењање и читање,...)
- **Безбедносни захтеви**



## Апликативни захтеви

- Матрица ентитета и процеса
  - описује у којим се процесима користе које табеле и на који начин
- Евентуални развојни проблеми
  - да ли су потребни неки посебни развојни стручњаци
  - нпр. за сложене алгоритме, извештаје, анализе и сл.



## Остале информације

- Захтеване перформансе
  - обично у облику времена одзива након задавања акције
  - неки послови морају да се одвијају интерактивно
    - морају да буду врло ефикасни (или бар одзивни)
  - неки послови не морају да се одвијају интерактивно
    - они могу да се обављају и пакетно или одложено
- Познате конкретне циљне имплементације СУБП
  - не само производ, него и конкретна верзија
  - зато што се разликују на физичком нивоу
- Евентуална ограничења простора
  - да ли постоје потенцијални проблеми са складишним простором



## Основни методи оптимизације БП

- Оптимизација на нивоу структуре података
  - физички модел података *се мења* у односу на логички модел
  - декомпозиција и сличне технике
  - *више о томе на наредним часовима*
- Оптимизација на нивоу интерне организације података
  - остварује се кроз управљање интерном организацијом података, помоћним компонентама и ресурсима
  - не мења се физички модел, већ се уређују аспекти његове имплементације
- Оптимизација на нивоу упита
  - упити се пишу тако да се омогући њихово ефикасније извршавање
  - физички модел се не разликује у односу на логички
  - данас то углавном служи као увод у претходна два вида оптимизације
  - *више о томе на наредним часовима*



## Опт. на нивоу интерне организације података

- Физичка организација података
  - простори за табеле, странице, бафери страница
- Помоћне компоненте
  - индекси
- Управљање ресурсима
  - пре свега радном меморијом



## Физичка организација података

- Логичка организација као основно место чувања података види релацију, тј. табелу
- Физичка организација иде и даље од тога и описује
  - начин записивања табела
  - начин приступања подацима
  - могућности управљања физичком организацијом
  - физичку организацију обраде – архитектура хардвера и софтвера
- СУБП обично омогућавају веома прецизно управљање елементима физичке организације података



## Елементи физичке организације

- Сваки СУБП има донекле специфичан начин физичке организације података и могућности управљања физичком организацијом података
- Елементе физичке организације релационих база података ћемо размотрити на примеру РСУБП *IBM DB2*



## Трајни и привремени елементи

- Основу физичке организације података чини
  - организација трајно записаних података
  - трајни записи података се чувају на дисковима
- Основу употребе и ефикасне имплементације програмских елемената СУБП чини
  - организација радне меморије
  - током употребе и обраде података, они се премештају у радну меморију
- За поуздан и ефикасан рад је кључна добра усклађеност ова два сегмента организације



## Организација РСУБП

- Два основна елемента сваког СУБП су
  - инстанце
    - једна инстанца је један *процес* (или скуп процеса) који управља радом базе података
      - једна инстанца може да се односи на више база података
      - у дистрибуираним или партиционисаним системима више инстанци може да дели послове одржавања једне базе података
    - обухвата програмске елементе СУБП
  - базе података
    - једна база података је једна целовита колекција података која се одржава (углавном) независно од других база података
    - обухвата све трајне податке и метаподатке једне целовите колекције података



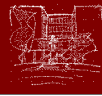
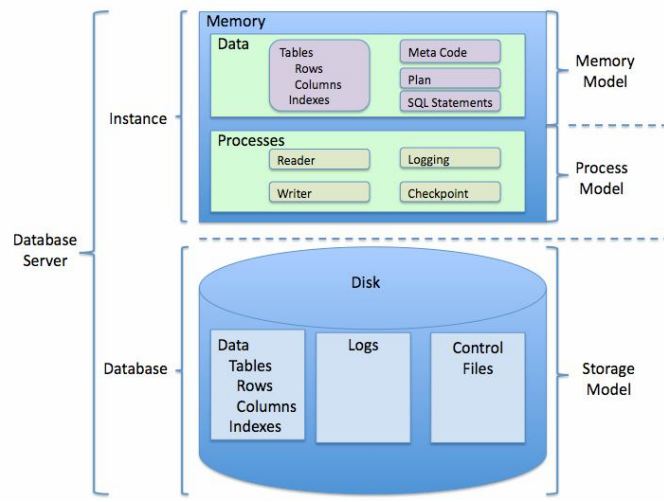
## Инстанце

- Обично се каже да је главни елемент инстанце процес
  - једна инстанца је један покренут СУБП
    - може да се покрене више инстанци једног инсталираног СУБП
  - инстанцу обично чини скуп процеса који имају различита задужења
    - координатор трансакција
    - процеси за читање – често посебан процес или нит за сваки физички уређај
    - процеси за асинхроно писање – самостално записују измењене (потврђене) странице и тако растеређују бафер страница
    - агенти – често по процес или нит за сваку сесију (конекцију)
    - и друго
- Део управљања меморијом се одвија на нивоу инстанце
  - други ниво управљања је на нивоу сесије (конекције)



## Однос инстанци и трајних података

- Све операције се одвијају и оквиру инстанце
  - и читање и мењање података
  - сваки захтев клијента иде преко неке инстанце
    - клијен не може да приступа непосредно трајним записима
- Трајним записима приступа искључиво одговарајућа инстанца
  - то је основни предуслов за поузданост и безбедност података
- Сваки трајни податак мора да буде преписан у радну меморију пре било које операције
  - то је предуслов да инстанца може да га користи
  - ова претпоставка је кључна за дефинисање прецизнијих елемената организације



## База података, као део физичког модела

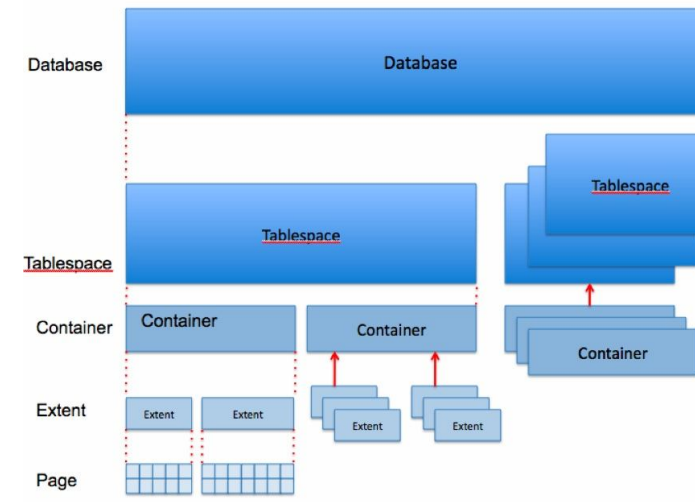
- У контексту физичког модела, под базом *логашака* се често подразумева скуп физичких трајних записа (датотека) који се односе на једну базу података (у ширем смислу)
  - **Просјор за табеле**
    - највиши ниво организације трајних података
    - трајни подаци се записују у форми записа табела (редова, вредности атрибута) и индекса, а у неком простору за табеле
  - **Дневник трансакција**
    - свака потврђена измена мора да буде записана у дневнику при потврђивању трансакције
    - често се записује и пре тога
  - **Контролне датотеке**



## Простор за табеле

- Основни складишни простор се назива *простор за табеле* (енгл. *table space*)
  - Свака база података мора да има један или више простора за табеле
  - Један простор за табеле припада тачно једној бази података
  - Један простор за табеле може да садржи више табела
  - У партиционисаним базама података једна табела може да буде и у више простора за табеле (хор. фрагментација)
- Простор за табеле представља *логички ниво* физичке организације
  - одређују се особине и начин употребе простора за табеле
  - не одређује се непосредно где се подаци налазе
  - али му се додељује скуп физичких контејнера за чување података

## Простор за табеле (2)



(wikimedia.org)



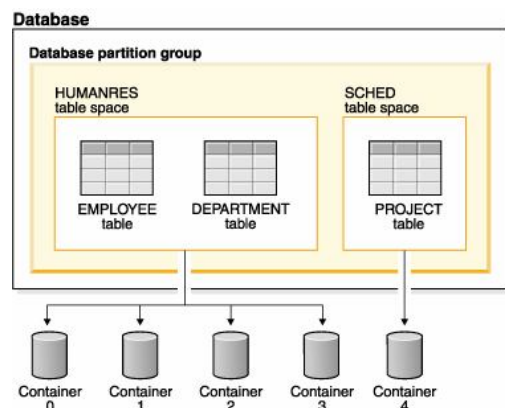
## Простор за табеле (3)

- На нивоу простора за табеле дефинишу се:
  - величина физичке странице (4, 8, 16 или 32 KB)
    - страница је јединица писања и организовања редова
  - величина јединице читања (енгл. *extent size*), обично у распону 2-256 страница
    - за ОЛТП се користе мање странице и јединице читања, а за складишта података веће странице и јединице читања
  - да ли се подаци чувају компримовано или не
  - да ли и како може да се аутоматски повећава
  - начин и услови баферисања страница
    - сваком простору за табеле се додељује тзв. бафер страница (енгл. *buffer pool*)
- Простору за табеле се додељују физичке локација складиштења
  - скуп *контејнера*



## Контејнери

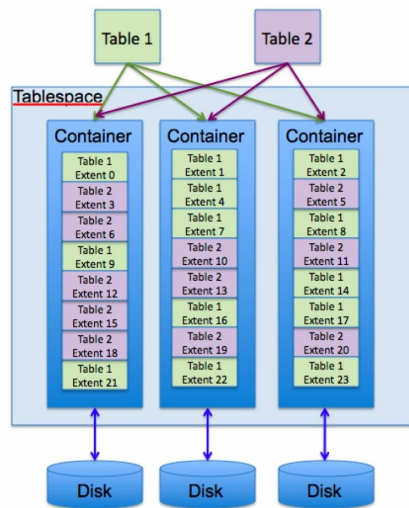
- Физичка локација складиштења је *контејнер*
  - Контејнер може да буде директоријум, партиција диска, диск, преалоциран фајл, али и тзв. *складишна група* (енгл. *storage group*)
    - складишна група предефинише скуп контејнера који затим може да се употребљава као целина
  - Простор за табеле садржи један или више контејнера
  - Један контејнер припада само једном простору за табеле
  - СУБП самостално уређује која страница које табеле ће бити у ком контејнеру изабраног простора за табеле
    - али на нивоу јединица читања а не појединачних страница



(www.ibm.com)

## Страница

- *Страница* је основни елемент физичког записа табеле
  - све операције се физички одвијају над страницама
  - све операције писања трајних података се одвијају над целим страницама
    - може да се записује појединачна страница или скуп страница
  - све операције читања трајних података се одвијају над целим јединицама за читање
    - чита се једна или више *јединице за читање* (енгл. *extent*)
- Свака табела (и сваки индекс) се састоји од страница
  - тачније, свака табела се записује у скупу страница једног простора за табеле
- Када се повећава простор за записивање табеле, увек се додаје (резервише) цео број јединица за читање



(wikimedia.org)

## Страница (2)

- Величина странице је једна од основних карактеристика простора за табеле
  - све странице једног простора за табеле су исте величине
- Ако су странице велике
  - редови табеле могу да буду већи
    - обично један ред мора да стане у страницу
    - ако и може да се "прелије" у више редова, то смањује ефикасност
    - може да се направи више колона
  - странице могу да садрже више редова
  - мање се тражи по диску
  - мања је дубина индекса
  - уобичајено за складишта података и аналитичке базе података
- Ако су странице мале
  - већа је искоришћеност простора на диску
  - мање је непотребног читања и писања
  - уобичајено за трансакционе базе података





## Страница (3)

- Страница се састоји од заглавља (енгл. *header*) и записа (енгл. *record*)
  - заглавље садржи основне податке о записима
  - записи задрже податке (табела или индекса)
- Не мора бити попуњена до краја
  - ако табела има много непопуњених страница, онда је повећано заузеће простора и успорен рад
    - оптимизација попуњености страница може да се изводи аутоматски или мануелно, али има релативно високу цену, зато што се индекси реферишу на странице
  - код индекса је слично, али неке врсте индекса увек имају "оптималну" организацију
    - нпр. алгоритми ажурирања индекса заснованих на Б-стаблима увек одржавају попуњеност страница у опсегу 50%-100%

## DB2 REORGCHK

...  
Table statistics:  
F1: 100 \* OVERFLOW / CARD < 5  
F2: 100 \* (Effective Space Utilization of Data Pages) > 70  
F3: 100 \* (Required Pages / Total Pages) > 80  
...

SCHEMA.NAME	CARD	OV	NP	FP	ACTBLK	TSIZE	F1	F2	F3	REORG
Table: IP.DISORDER_NUMERIC	4088695	0	7198	7198	-	2.33e+08	0	99	100	---
Table: IP.MAIN	704	0	18	18	-	548416	0	100	100	---
Table: IP.CDS	4357	1169	16	21	-	535911	26	86	80	*--
Table: IP.RIPITI_IC	3.6e+07	0	32190	32195	-	9.08e+08	0	86	99	---
Table: IP.RPT_IN	0	0	0	1	-	0	0	-	0	---
Table: IP.SPIDER2_ALL_2	527752	0	841	1698	-	27443104	0	49	49	**
Table: IP.SPIDER2_ALL	527758	0	829	2509	-	26915658	0	32	33	**

...  
...

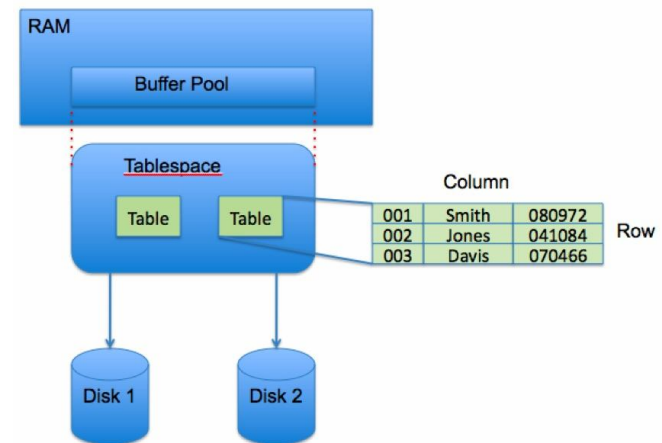
Annotations:  
 - Број редова (points to NP)  
 - Број редова који су премештени у другу страницу због величине (points to FP)  
 - Број непразних страница (points to ACTBLK)  
 - Укупан број страница (points to CARD)



## Бафер за странице

- Бафер за странице (енгл. *buffer pool*) је меморијски простор предвиђен за чување копија једног броја страница ради њихове употребе
  - Свако читање из базе података захтева да се прво одговарајуће странице прочитају из контејнера и упишу у бафер за странице
  - Свако писање почиње мењањем садржаја одговарајућих страница у баферу страница, које се касније (асинхронно) преписују у контејнере
- Што је бафер за странице већи, то је број приступа диску мањи
  - више страница је у баферу
  - приступ подацима је бржи

## Бафер за странице





## Бафер за странице (2)

- Бафер за странице се односи на један простор за табеле
  - Добро конфигурисање простора за табеле и бафера за странице може да буде од пресудног утицаја на перформансе
  - Чак и када цела база података физички стаје на један диск, табеле и индексе има смисла распоредити у више простора за табеле, да би се лакше и боље управљало баферима страница
- У идеалном случају цела база података је у меморији
  - Ако то није могуће, тежи се да у меморији буду бар најважнији индекси (или бар првих неколико нивоа индекса) и мање табеле (шифарници)
  - Зато се индекси и често реферисане табеле стављају у посебне просторе за табеле, да би могло да се обезбеди да буду стално у меморији
    - т.ј. да би имали посебне бафере за странице



## Други важни концепти

- Физичка организација података почива на још много важних концепата
  - партиционисане табеле
  - колонске табеле
  - компресија података
  - материјализовани погледи
  - катанци или алтернативне технике за изоловање података
  - различити други концепти, често специфични за имплементације



## Партиционисане табеле

- Садржај табеле може да се подели на више физичких целина (партиција) (обично на основу вредности кључа)
  - На пример, сви подаци о студијама би могли да се партиционису по години уписа студента
- Табела се и даље логички употребљава као неподељена целина
- Свака партиција се физички понаша као посебна табела
  - партиције могу да иду у различите просторе за табеле
  - могу одвојено да се бришу, архивирају, рестаурирају...
- Тиме се омогућава
  - лакше архивирање старих података,
  - различито управљање баферисањем нових и старих података
  - боља паралелизација читања и писања и друго



## Колонске табеле

- Колонске табеле (тј. табеле организоване по колонама) су физичка репрезентација у којој се подаци не чувају “ред-по-ред” него “колона-по-колона”
- Намењене превасходно аналитичким базама података
  - и у другим случајевима, ако се на табели *изузетно ретко* врши рестрикција и *скоро никада* ажурирање



## Компресија података

- Савремени СУБП обично омогућавају да се подаци компримују, како би заузимали мање места
- То обично не утиче на начин употребе података
- Савремен харвер омогућава да се компримовање и декомпримовање таквих података врши „у ходу“, без видљивог утицаја на перформансе
- Слично је и са енкриптовањем



## Материјализовани погледи

- Материјализовани погледи (енгл. *materialized query table*) су табеле чија структура и садржај су дефинисани датим упитом
- Врсте:
  - системски одржаване
  - кориснички одржаване
  - табеле у сенци (*shadow table*)
    - одржавају се репликацијом
  - федеративне
    - одржавају се алатима за федеративне базе података



## Материјализовани погледи (2)

- Основна намена им је убрзавање сложених (обично аналитичких) упита
  - над подацима који се релативно ретко мењају
  - који се релативно лако ажурирају у случају промене података:
    - агрегирани упити
    - спајања и агрегације над више табела
    - упити над често коришћеним скупом података



## Стратегије изоловања трансакција

- Циљ је међусобно изоловање трансакција
  - рад једне трансакције не сме да утиче на рад друге
  - посебно, не сме да допринесе да друга трансакција ради неисправно
  - свака трансакција „би требало“ да ради као да је сама на систему
  - т.ј. „изоловање“ је привидно али ефективно
- Две основне стратегије су
  - песимистичка стратегија
  - оптимистичка стратегија



## Песимистичко изоловање

- Уобичајено је за базе података
- Претпоставља се да ће више трансакција радити у исто време
  - и да ће различите трансакције често користити (исте) податке
- Остварује се помоћу система катанаца
  - сваки податак се пре приступања закључава одговарајућим катанцем
  - катанац се чува до краја трансакције и тако се обезбеђује да нико не може да му приступи на нерегуларан начин



## Катанци

- Механизам катанаца је уобичајено средство остваривања изолованости трансакција
- Сваки катанац има:
  - објекат који се закључава
  - трајање
  - врсту катанца



## Катанци (2)

- Објекат може бити
  - вредност (атрибут)
  - ред табеле
  - страница табеле
  - група страница
  - цела табела
  - простор за табеле
  - индекс
- Величина објекта одређује *трануларности* катанца



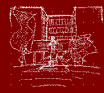
## Ескалација катанца

- Величина катанца
  - Катанац може да закључава један ред или више редова
  - Или једну страницу или више страница
- Велики број катанаца може да значајно успори рад
  - због много више провера при раду...
  - зато је број катанаца ограничен (било бројем катанаца било количином меморије која је резервисана за катанце)
- Када се превазиђе допуштен број катанаца долази до *ескалације катанца*
  - Више мањих катанаца се замењује једним већим
  - На пример, више катанаца на редовима се замењује катанцем на страници
  - Или, више катанаца на страницама се замењује катанцем на табели



## Оптимистичко изоловање

- Уобичајено је за дуготрајне трансакције (нпр. отвори се формулар и попуњава се или ажурира врло дуго, можда и цео сат или дуже)
- Претпоставља се да друге трансакције неће користити исте податке
  - или да ће то радити веома ретко
- Обично се имплементира уз памћење стања коришћених података
  - при читању се запамти и када је податак прочитан
  - када дође време за писање
    - сва писања чине једну *песимистичку* трансакцију
    - за сваки податак на коме почива трансакција се проверава да ли је у међувремену био промењен
    - ако јесте, трансакција се прекида



## Оптимистичко изоловање (2)

- База података може да подржи оптимистичко закључавање прилагођавањем физичке структуре
  - свакој табели се додаје колона са временом последње промене реда
  - или се додаје *шокен измена*
  - или се имплицитно скривају колоне
  - или се додају функције за читање идентификатора реда и проверавање стања реда (на пример последње измене)



## Индекси

- Индекси су помоћне структуре података које омогућавају брже приступање подацима, тј. брже претраживање по унапред изабраном кључу
- Свака табела може да има више индекса са различитим кључевима
- Индекс је физичка структура података која се записује у изабраном простору за табеле



## Индекси (2)

- Дефиниција индекса обухвата
  - врсту (структуру) индекса
  - колоне које чине услов уређивања, тј. кључ приступања
  - остала својства индекса:
    - да ли је индекс (тј. одговарајући кључ) јединствен или није
    - да ли је индекс једносмеран или двосмеран
    - да ли је уређујући (групишући) или не
    - да ли је партиционисан или није



## Јединствени индекси

- Јединствени индекси не дозвољавају понављање више редова са истим кључем
- Користе се и као средство за имплементирање интегритета кључа (јединствености)
- Три врсте:
  - забрањене недефинисане вредности
  - дозвољене нед. вредности, сматрамо да су једнаке
  - дозвољене нед. вредности, сматрамо да су различите



## Групишући индекси

- Обезбеђују да су редови у табели поређани у одговарајућем поретку
  - Највише један такав индекс по табели
- Убрзавају издвајање „подсеквенце“ редова у датом поретку
  - на пример:
    - ...where volume between 200 and 220
- Успоравају одржавање
  - не мења се само индекс него и физички запис редова



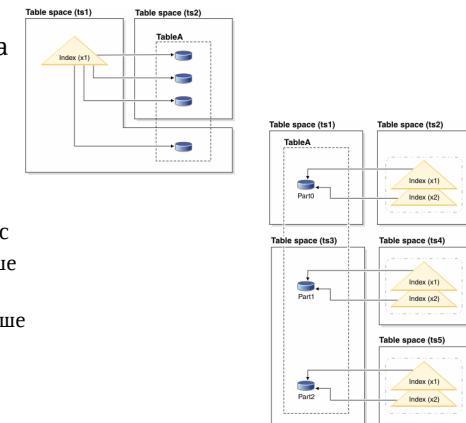
## Групишући индекси (2)

- Зову се и у „уређујући“ (енгл. *sorting* или *clustering*)
- Кандидати за групишуће индексе су колоне
  - које се често траже у опсезима
  - по којима се често уређује резултат
  - које припадају страном кључу по коме се најчешће врши спајање
  - колоне примарног кључа
- Предности
  - вишеструко убрзано читање низова редова по услову
- Слабости
  - додатно успорено одржавање
  - не убрзава приступање појединачним редовима



## Партиционисани индекси

- Партиционисана табела може да има
  - непартиционисани индекс
    - цео индекс је у једној партицији
  - партиционисани индекс
    - индекс се налази у више простора за табеле
    - обично се партициониса на исти начин као и табела





## Двосмерни индекси

- Двосмерни индекс омогућава претраживање у оба смера
- За неке врсте индекса то не представља велику промену у имплементацији
- За неке друге може да представља



## Индекси са структуром Б-стабла

- Подаци се чувају у балансираном дрвету
  - сваки ред табеле је реферисан из листа једнаке дубине
  - чворови и листови садрже по више података
    - јединица организације индекса је *страница*
    - величина чворова одговара страници простора за табеле
- Ако се не наведе другачије, данас се обично подразумева да се у бази података праве овакви индекси

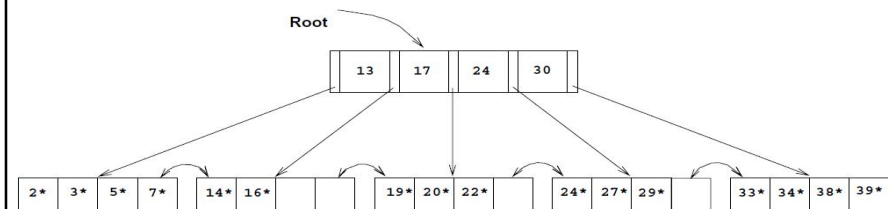


## Индекси са структуром Б-стабла (2)

- Предности
  - једноставни и ефикасни алгоритми за одржавање
- Слабости
  - није посебно ефикасан ако је много редова а мало различитих вредности кључа, тј. ако има много понављања кључева



## Индекси са структуром Б-стабла, илустрација



Погледати у литератури...



## Бит-мапирани индекси

- Индекс је организован као низ вредности кључа
- Уз сваку вредност кључа следи низ битова, за сваки ред табеле по један
  - вредност бита је један ако одговарајући ред има баш ту вредност кључа



## Бит-мапирани индекси (2)

- Предности
  - ако има релативно мало различитих вредности кључа, онда је оваква структура ефикаснија од Б-стабла
  - ефикасно се комбинује употреба више индекса
- Слабости
  - у случају много различитих вредности кључева, индекс постаје веома велики, а тиме и слабо ефикасан
  - одржавање индекса је често значајно скупље него у случају Б-стабла



## Хеш индекси

- Индекс се имплементира као хеш табела са датим кључем
  - рачунају се хеш вредности на основу кључних атрибута, а онда се помоћу добијене вредности непосредно приступа подацима
  - алтернатива је да се прави б-стабло а да се као кључ користе хеш вредности



## Хеш индекси (2)

- Алтернатива класичним индексима б-стабла
  - Ефикаснији су за приступање појединачним редовима са тачно задатим кључем
- Нису добри за секвенцијално приступање већем броју редова или ако оператор поређења није једнакост
  - нпр. ако је услов ( $X < 20$ ), онда хеш табела није корисна
  - такође, ако је кључ од више атрибута, а у услову не наводимо неки од њих, хеш табела није корисна
- Обично се не користе као трајне структуре, али се праве за приступање привременим табелама (међуреферентима) при извршавању упита, посебно када се ради о већем броју спајања





## Табеле без индекса

- У случају малих табела, може да буде брже да се увек претраже сви редове него да се користе индекси
  - нпр. неке врсте шифарника
- Прихватљива граница величине зависи од имплементације и начина употребе
  - обично од неколико редова до највише пар стотина редова
    - тј. ако цела табела стаје у свега неколико страница



## Индекси са додатним колонама

- Савремени РСУБП омогућавају да се индексу осим колона кључа додају и још неке колоне, које не чине услов уређења
- Ако упит захтева само колоне кључа и те додатне колоне, онда се читање завршава на индексу
  - не морају да се читају странице табеле
- Употреба је нешто скупља
  - повећава се физичка величина индекса
  - повећава се и дубина Б-стабла (зато што се смањује број елемената по једној страници)



## Индекси, резиме

- Предности индекса
  - омогућавају бржи приступ конкретним редовима табеле
  - ако читање захтева само колоне садржане у индексу, онда табели не мора ни да се приступања (тзв. индекси са додатним колонама)
- Слабости индекса
  - индекси морају да се одржавају
    - свака операција додавања и брисања, као и операције мењања колона које чине кључ индекса, захтевају ажурирање индекса
  - заузимају простор
  - могу да повећају број закључавања или грануларност катанаца
  - подижу цену реорганизовања или преношења табела



## Колико индекса је идеално?

- Да ли неки индекс доноси више користи или штете зависи од тога
  - да ли се индекс уопште користи у упитима
  - колико употреба индекса доприноси перформансама упита у којима се користи
  - колико су такви упити чести
  - колико је често и скупо ажурирање индекса
- Идеалан број и врста индекса зависе од врсте, намене и структуре табеле и базе података, као и од начина употребе
  - обично се препоручује
    - за трансакционе табеле 3-5 индекса
    - за аналитичке табеле без ограничења (све док је ажурирање прихватљиво)
  - да ли је индекс користан или не потребно је да се утврди експериментално



## Управљање меморијом

- Веома је важно да се при администрирању СУБП добро управља меморијом
- Ако се меморија не користи довољно, или се користи погрешно, сувише ће се приступати диску
- ВАЖНО: СУБП не користи виртуалну меморију за приступање страницама базе података
  - бафери страница морају да буду у физичкој меморији
  - у супротном се двоструко приступа диску...



## Употреба меморије

- СУБП користи меморију на много различитих начина:
  - главни и помоћни бафери страница
  - листе катанаца
  - кеш каталога базе података (података о структури и организацији података)
  - кеш пакета (података о програмима)
  - интерни хип базе података (користи се при израчунавању упита)
  - хип помоћних алата (за уређивања, претраживања, резервне копије,...)
  - меморија агената (свака успостављена сесија (енгл. *connection*) са базом има свој простор)
  - и друго...



## Употреба меморије, пример DB2 (1)

- Меморија се дели на четири целине:
  - Дељена меморија инстанце (чвора)
  - Дељена меморија базе података
  - Дељена меморија групе апликација
  - Приватна меморија агента (процеса или нити)



## Употреба меморије, пример DB2 (2)

- Дељена меморија инстанце (чвора):
  - меморија инстанце
  - хип за праћење рада и стања (монитори)
  - аудит бафери
  - бафери за брзу комуникацију међу чворовима



## Употреба меморије, пример DB2 (3)

- Дељена меморија базе података обухвата већину раније наведених ставки:
  - главни бафери страница
  - скривени (помоћни) бафери страница
  - хип за листе катанаца
  - кеш каталога базе података (података о структури и организацији података)
  - кеш пакета (података о програмима)
  - хип помоћних алата (за уређивања, претраживања, резервне копије,...)
  - бафери за дневник трансакција

## Литература за тему



- Teorey, et.al, **Database Design, know it all**, 2008.
- Ramakrishnan, Gehrke, **Database Management Systems, 2.ed**, 2000.
- IBM DB2 Version 11.5 Knowledge Center
  - [https://www.ibm.com/support/knowledgecenter/SSEPGG\\_11.5.0](https://www.ibm.com/support/knowledgecenter/SSEPGG_11.5.0)