

Увод у релационе базе података

13а



Саша Малков
Универзитет у Београду
Математички факултет
2023/2024

[PM13]
Увод у РБП
Саша Малков



Тема 7

Физички модел базе података - Индекси -

[PM13] Увод у релационе базе података - Саша Малков - 2023/24 - час 13а

1

Физички модел базе података / Индекси



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

Универзитет у Београду - Математички факултет

[PM13] Увод у релационе базе података - Саша Малков - 2023/24 - час 13а

2

Физички модел базе података / Индекси



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

Универзитет у Београду - Математички факултет

[PM13] Увод у релационе базе података - Саша Малков - 2023/24 - час 13а

3



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 читање „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



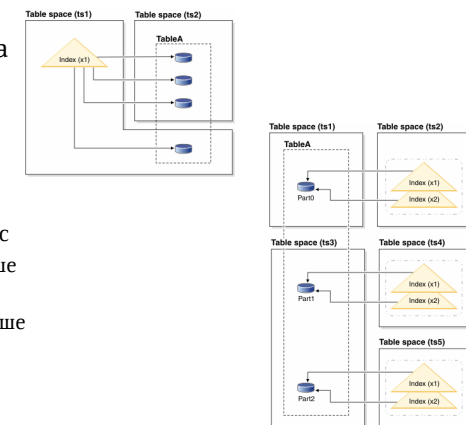
Групишући индекси (2)

- 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

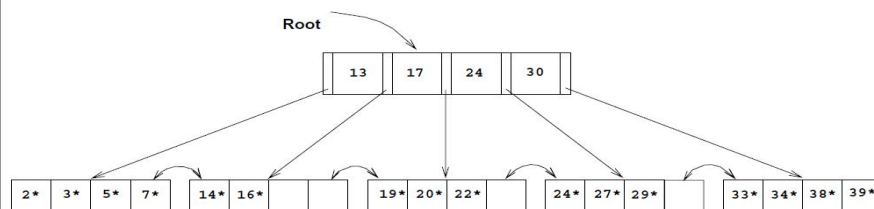
- Двосмерни индекс омогућава претраживање у оба смера
- За неке врсте индекса то не представља велику промену у имплементацији
- За неке друге може да представља



Индекси са структуром Б-стабла

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

Индекси са структуром Б-стабла, илустрација



Погледати детаљније у литератури...



Индекси са структуром Б-стабла (2)

- Предности
 - једноставни и веома ефикасни алгоритми за одржавање
 - чворови се додају у ширину и навише
 - чворови су увек попуњени између 50% и 100%
- Слабости
 - није посебно ефикасан ако је много редова а мало различитих вредности кључа, тј. ако има много понављања кључева



Бит-мапирани индекси

- Индекс је организован као низ парова (вредност кључа, битови индекса)
- 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
 - зависно од критеријума, комбиновање се остварује конјункцијом или дисјункцијом низова битова
- Слабости
 - у случају много различитих вредности кључева, индекс постаје веома велики, а тиме и слабо ефикасан
 - одржавање индекса је често значајно скупле него у случају Б-стабла



Бит-мапирани индекси (3)

- Битмапирани индекси су једина врста индекса која омогућава да се при одабиру редова једне табеле комбинује више индекса
- Изузетак (али непотпуни) су СУБП (на пример DB2) који у току извршавања упита праве битмапе сличне индексима да би филтрирали податке по више индекса



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 (уместо непосредних података)



Хеш индекси (2)

- Алтернатива класичним индексима б-стабла
 - Ефикаснији су за приступање појединачним редовима са тачно задатим кључем
 - Релативно лако и ефикасно се праве
 - али је одржавање мање ефикасно
- 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



Хеш индекси (3)

- Веома ретко се користе као трајне структуре
 - многи СУБП и немају наредбе за њихово прављење
- Често се користе за приступање привременим табелама (међурезултатима при извршавању упита), посебно када се упиту изводи више спајања



Табеле без индекса

- У случају малих табела може да буде брже да се увек претраже сви редови 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



Индекси са додатним колонама (2)

- Употреба може да буде нешто скупља
 - повећава се физичка величина индекса
 - повећава се и дубина Б-стабла (зато што се смањује број елемената по једној страници)
- Да ли је у конкретној ситуацији боље додавати индексу колоне или не, обично мора да се испроба и измери
 - ефикасније, зато што се избегава приступање табели
 - спорије, зато што се индекс повећава



Прављење индекса (DB2)

```
CREATE [UNIQUE] INDEX <ime>
ON <ime table> ( {<kolona> [ ASC | DESC | RANDOM ]}* )
[ IN <ime prostora za table> ]
[ INCLUDE (<lista kolona>) ]
[ CLUSTER ]
[ ALLOW REVERSE SCANS ]
[ DISALLOW REVERSE SCANS ]
[ INCLUDE NULL KEYS ]
[ EXCLUDE NULL KEYS ]
[ ... ]
```



Прављење индекса (DB2) (2)

- Основна наредба прави индекс са структуром б-стабла
 - са датим именом (и у датом схеми)
 - над датом табелом
 - где се редови уређују према наведеном списку колона
 - и у наведеном редоследу
 - ASC – растући
 - DESC – опадајући
 - RANDOM – случајан редослед
 - заправо начин да се наведе да се колона додаје индексу без уређивања
 - уместо колона могу да се наведу и рачунски изрази са колонама



Прављење индекса (DB2) (3)

- Кључна реч *UNIQUE* наглашава да је индекс без понављања
 - одговара дефиницији јединственог кључа
 - недефинисана вредност се понаша као вредност – не сме да се понови
- Опција *IN...* одређује простор за табеле у коме се прави индекс
- Опција *INCLUDE...* одређује које додатне колоне се записују у индексу
- Опција *CLUSTER* одређује да се прави групишући индекс



Прављење индекса (DB2) (4)

- Опција *ALLOW REVERSE SCANS* одређује да индекс може да се користи једнако ефикасно за уређивање и претраживање у оба смера
- Опција *DISALLOW REVERSE SCANS* одређује да индекс не може да се користи за уређивање и претраживање у оба смера (подразумевано понашање)
- Опција *INCLUDE NULL KEYS* одређује да се ред индексира и када су сви делови кључа недефинисани (подразумевано понашање)
- Опција *INCLUDE NULL KEYS* одређује да се ред не индексира када су сви делови кључа недефинисани
- ...и многе друге опције...



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 колоне 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
- СУБП обично не користи виртуалну меморију за приступање страницама базе података
 - бафери страница морају да буду у физичкој меморији
 - у супротном се ефекат баферисања доводи у питање



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...



На примеру DB2

- Меморија се дели на четири целине:
 - Дељена меморија инстанце (чвора)
 - Дељена меморија базе података
 - Дељена меморија групе апликација
 - Приватна меморија агента (процеса или нити)
- Све врсте дељене меморије деле сви агенти (процеси или нити које приступају подацима)



На примеру DB2

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



На примеру DB2

- 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



- Гордана Павловић-Лажетић, **Увод у релационе базе података**, 2. изд. *Математички факултет*, 1999.
 - доступно онлајн: <http://poincare.matf.bg.ac.rs/~gordana/urbp-2016.htm>
- Документација за *DB2 11.5*:
 - онлајн:
 - https://www.ibm.com/support/knowledgecenter/SSEPGG_11.5.0
 - ПДФ:
 - <https://www.ibm.com/support/pages/node/627743>
- Визуализација алгоритама:
 - <https://www.cs.usfca.edu/~galles/visualization/Algorithms.html>