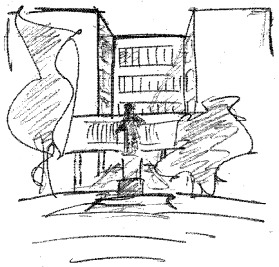
[P273]

Пројектовање база података

13



Саша Малков

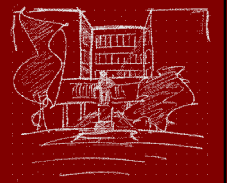
Универзитет у Београду

Математички факултет

2023/2024

[P273]

Пројектовање база података



Саша Малков

Тема 15

Пројектовање база података за подршку одлучивању

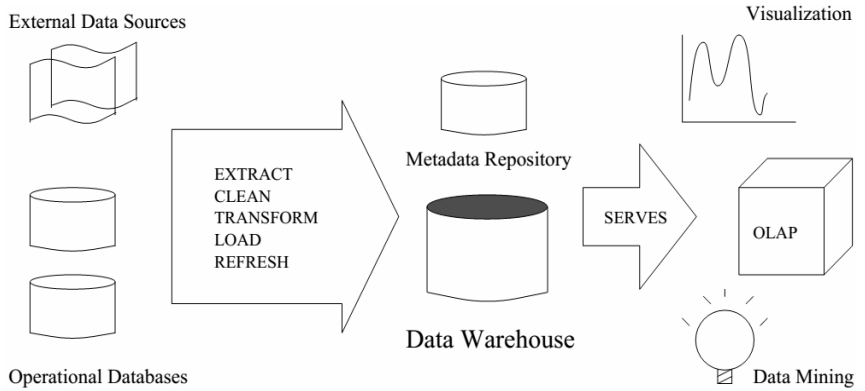
Подршка одлучивању

- Пружање информација од стратешког значаја
  - на основу анализа прукупљених информација
- Различити називи:
  - подршка одлучивању (енгл. „*decision support*“)
  - подршка планирању (енгл. „*planning support*“)
  - пословно обавештавање (енгл. „*business intelligence*“)
  - пословна интелигенција

Подршка одлучивању (2)

- Основне теме у оквиру подршке одлучивању су
  - **Прикупљање и одржавање** великих колекција података
  - **Анализирање** тих колекција података
- Основни појмови
  - **Складишта података** - базе података које су пројектоване специфично за подршку одлучивању и не користе се за трансакциону употребу
  - **Онлајн аналитичка обрада** (енгл. *OLAP*)
    - аналитичка обрада података постављањем сложених аналитичких упита
    - у ужем смислу – аналитичка обрада која се одвија “уживо” над базом података која може да се користи и за друге намене
  - **Истраживање података** (енгл. *Data Mining*) – примена различитих математичких и рачунарских метода ради извођења сложенијих закључака и образаца из постојећих података
- У оквиру ове теме ћемо размотрити складишта података и ОЛАП
  - истраживањем података се бави посебан предмет

Подршка одлучивању (3)



Складишта података

- Специфичне базе података
  - основна намена је аналитичка обрада података
  - над подацима који нису “сасвим” ажурни
    - садрже “историјске” податке
    - периодично се допуњавају и ажурирају свежим подацима
    - тзв „офлајн аналитичка обрада“
  - прилагођене су читању и анализи података
  - нису погодне за редовно ажурирање
- Подаци се обично прикупљају из различитих извора
  - често се ради о огромним колекцијама података
- Пројектују се у складу са специфичностима

Онлајн аналитичка обрада

- У ширем смислу – свака аналитичка обрада података
  - обично се почиње од општијих упита и читања
  - па се наставља постепеним фокусирањем на одређене групе података и профињивањем упита
  - подразумева одређен ниво интерактивности
    - не обавезно у писању и постављању упита, већ најчешће у употреби алата који то раде уместо корисника
- У ужем смислу - аналитичка обрада која се изводи над „живим“ подацима
  - над свим довршеним трансакцијама
- У елементе ОЛАП-а спадају и одговарајући елементи упитних језика и технике њихове употребе
  - у основи могу да се примењују на све врсте база података

Онлајн аналитичка обрада (2)

- Иако је основна природа упита веома слична, њихова имплементација може да се веома разликује у зависности од врсте базе података
  - структура складишта података је значајно другачија од уобичајених ОЛТП база података
  - начин употребе складишта података се разликује од уобичајених ОЛТП база података

Онлајн аналитичка обрада (3)

- Код ОЛТП/ОЛАП база података аналитичка обрада се одвија истовремено са трансакционом обрадом
  - пројекат базе мора да подржи обе намене
  - једна примена не сме да (значајно) омета другу
  - старији подаци се често периодично бришу ради растерећивања система
- Код складишта података се обрада одвија уз претпоставку непроменљивости података
  - обично се не ради никакво изоловање упита
  - структура података је прилагођена упитима из висок степен редундантности
  - обим података је обично далеко већи
  - обично се чувају сви историјски подаци

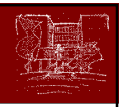
Истраживање података

- Извођење квалитативно нових информација из великих скупова података
- Теорија и примена техника
  - развој и примена алгоритама
- Скоро увек на складиштима података
  - обрада је најчешће сложена и дуго траје
  - подаци не би требало да се мењају током обраде

Подршка одлучивању – специфичности

- Прављење сложених извештаја
- Истраживање података
- Сложена обрада захтева да подаци буду на једном месту
  - релативно ретко се користе дистрибуиране базе
  - обично се праве складишта података
- При доношењу стратешких одлука обично нису важне најсвежије информације
  - не смета ако се изоставе из обраде

Складиште података



- ***Складиште података*** (енгл. „*data warehouse*“) је простор за прикупљање (одлагање) историјских података, који се интегрисано употребљавају ради пружања подршке одлучивању
  - најважније су историјске информације
  - текуће информације (резултати најсвежијих трансакција) имају далеко мањи значај
- Продукциони системи садрже и користе податке који су неопходни за свакодневни рад
  - важне су све свеже информације
  - и у облику који је за то погодан

ОЛТП базе података	Складишта података
транскационо оријентисане	оријентисане према теми
хиљаде корисника	неколицина корисника
релативно мале (до неколико *GB*)	релативно велике (од неколико GB до неколико PB)
фокус је на текућим подацима	фокус је на историјским подацима
нормализовани подаци (нема редундантности, обично више табела са по мало колона)	денормализовани подаци (има редундантности, обично мало табела са по више колона)
непрекидно појединачно ажурирање	периодично пакетно ажурирање
једноставни и сложени упити, најчешће мањег обима	веома сложени упити, често веома великог обима

Величина складишта података

- Величина складишта података се ретко објављује, али ево пар примера:
  - 17.02.2014. Гинисова књига рекорда је евидентирала да је база података *SAP Colocation Lab, Santa Clara, California, USA* достигла 12.1 PB (12,100 TB)
    - 221 x 10^{12} редова
    - пуњење 2 x 10^{12} редова за сат (око 90 TB)
    - 100 x 10^{9} докумената
    - пуњење 1 x 10^{9} редова за сат (око 25 TB)
    - 30 x 10^{9} извора (корисници, сензори, мобилни уређаји,...)
    - редова за сат (око 25 TB)
    - 25 чворова са по 4 процесора са 10 језгара (*Intel Xeon E7-4870*)
      - 1 TB RAM
  - Неки чланци указују и на већа складишта:
    - *Adweek*, април 2014: "*Facebook* је достигао 300 PB са дневним приливом од 600 TB"
    - *itNews*, 10.05.2014: "...*eBay* складиште обухвата око 90 PB података..."

Основни захтеви СП

- Подаци се организују тематски
- Могућности интегрисања података
- Подаци су непроменљиви (током обраде)
- Ажурирање се одвија периодично, у масовним пакетима
- Подаци су често потребни на различитим нивоима грануларности
- Флексибилност у односу на захтеве и циљеве
- Могућност „мењања историје" и „будуће историје" ради анализа типа „шта ако"
- Специфичне корисничке апликације са одговарајућим корисничким интерфејсом

Тематска организованост

- Теме одговарају групама и врстама података
  - обично према функцијама ИС
  - нпр. продаја, вођење пројеката, кадрови и сл.
- Теме су обично међусобно независне у односу на трансакције
  - тј. ретко се једном трансакцијом мењају подаци у више области
- Свака тематска област се посебно пројектује и имплементира у СП
  - мада могу да имају пресеке (димензионе табеле...)
  - обично свака тематска област има своју посебну концептуалну схему у продукционој бази података

Тематска организованост (2)

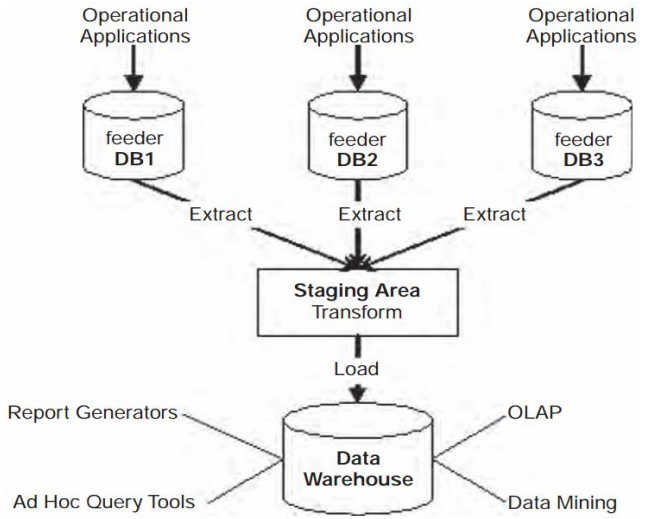
- За сваку тему се дефинише
  - циљ прикупљања информација
    - што прецизнија врста анализа које ће се обављати
  - спецификација врста и грануларности података који се прикупљају
  - начин и период прикупљања и ажурирања података
  - структура дела базе података који се односи на тему
- Централни подаци теме се не деле међу темама
- Више тема може да дели неке информације о димензијама

Могућности интегрисања

- Иако се свака тематска целина пројектује и имплементира независно, корисно је да постоји могућност интегрисања
  - интегрисање тема – ради извођења сложенијих упоредних анализа
  - интегрисање извора – подаци се сакупљају из више извора и интегришу у целину
- Или су све целине у једној БП која садржи СП, или се за интегрисање примењују технике федеративних база података

Подаци су непроменљиви

- Подаци се ажурирају периодично и масовно
  - у великим пакетима
  - између два пакета ажурирања су непроменљиви
- Ажурирање се обично своди на додавање нових *припремљених* података
  - подаци се **пречишћавају** да би се избацили непоуздани и нерелевантни подаци
  - **трансформишу** се да би се чували у одговарајућем облику, лаком за ефикасно претраживање и анализирање
  - СП се **пуни** тако припремљеним подацима
  - након пуњења се ажурирају помоћне табеле
    - на пример, материјализовани погледи
  - механизми за **освежавање** прате измене (у ОЛТП базама података или пристиглим пакетима измена) и по потреби (на основу предефинисаних критеријума, нпр. време или обим података...) иницирају нови циклус ажурирања



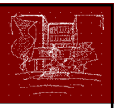
Вишеструка грануларност

- Анализе података се врше на различитим нивоима грануларности
- Анализе су ефикасније ако подаци већ постоје припремљени на свим тим различитим нивоима
  - нпр. одвојено се воде збирни нумерички подаци који описују тачно време, сат у дану, део дана, дан у недељи, месец и сл.
  - или подаци о локацији, тако да се редундантно за сваки догађај осим тачне адресе воде и ознаке месне заједнице, општине, града, региона и сл.
- Наравно, сви виши нивои се могу извести из нижих, али то захтева додатну обраду (макар и обуично сумирање), па је обично ефикасније да се подаци воде вишеструко

Флексибилна структура

- Пословно окружење је веома динамично
- СП мора да буде довољно флексибилно да омогући што једноставније проширивање скупа посматраних података
  - додавање нових тема
  - додавање нових колона у постојеће табеле чињеница
  - додавање нових димензионих табела
  - додавање нових типова података и операција за рад са њима

Платформа



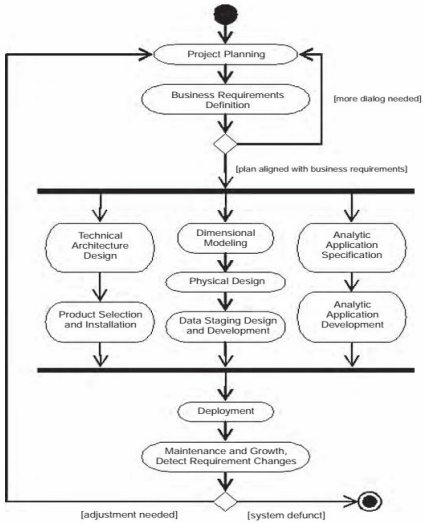
- Због начина пројектовања и уобичајеног постављања сложених упита, често се користе релационе базе података
- Поред њих користе се и БП прилагођене за дистрибуирано чување података
  - *Hadoop*
  - *Hive*

Преписивање историје

- Веома често су корисне анализе типа „шта ако?“
- Такве анализе захтевају „мењање историје“ и „будуће историје“
- За те намене може да се користи хоризонтално партиционисање
  - на пример, ако подаци за сваки месец иду у посебну партицију, онда „дописивање“, „одбацивање“ или „замењивање“ неког месеца може да се оствари релативно једноставно, искључивањем, додавањем или замењивањем активних партиција

Животни циклус СП

- Слично као и за друге базе података:
  - Прикупљање и анализа захтева
  - Логичко пројектовање
  - Физичко пројектовање
  - Дистрибуирање података
  - Имплементација, праћење и мењање



Прикупљање и анализа захтева СП

- Анализа крајњих захтева и циљева
- Прављење спецификација захтева
- Дефинисање архитектуре и оквирно планирање капацитета, уређаја и алата
- У основи све то одговара концептуалном пројектовању
  - али је пословни задатак практично увек познат и исти
  - мењају се само подаци који се анализирају
  - зато се често не користи термин *концептуално пројектовање*

Логичко пројектовање СП

- Пројектовање табела и погледа складишта података
- Као основа се користи „димензионо моделирање“:
  - основни облик је тзв. *звездаста схема* (енгл. *star schema*)
  - користи се и схема пахиљице (енгл. *snowflake schema*)

Звездаста схема

- У центру пажње је једна велика **табела чињеница**
  - садржи све податке или бар референце на све податке који су од значаја за конкретну тему
- Све додатне информације се записују у **димензионим табелама**
  - на све податке у димензионим табелама се реферише непосредно из табеле чињеница

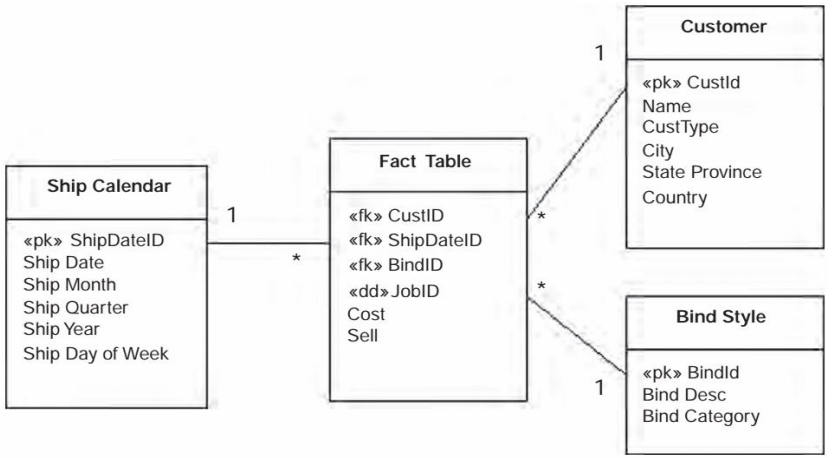
Табела чињеница

- Једној теми одговара једна велика табела чињеница
  - веома ретко има потребе и смисла да се за једну тему прави више табела чињеница
  - користе се и термини “табела података”, “табела ставки” и други
- Табела чињеница је у центру пажње
  - садржи све податке или бар референце на све податке који су од значаја за конкретну тему
  - практично сви аналитички упити се постављају над табелом чињеница
- Основни подаци у табели чињеница су денормализовани:
  - вредносни подаци који могу да се упоређују
  - кључеви у односу на димензионе табеле

Димензионе табеле

- Димензионе табеле описују појединости димензија које се наводе у табели чињеница
  - једној теми уобичајено одговара већи број димензионих табела
  - оне су обично мање табеле са свега пар колона
    - најчешће само нумерички примарни кључ и један или два додатна описна атрибута
    - појединачни редови могу да описују и опсеге вредности
- Димензионе табеле се повезују са табелом чињеница ради прављења „читљивих“ извештаја
  - нису нормализоване
  - обично нису предмет анализе

Звездаста схема, пример 1



Звездаста схема, пример 2

locid	*city*	*state*	*country*
1	Ames	Iowa	USA
2	Chennai	TN	India
5	Tempe	Arizona	USA

Locations

pid	*pname*	*category*	*price*
11	Lee Jeans	Apparel	25
12	Zord	Toys	18
13	Biro Pen	Stationery	2

Products

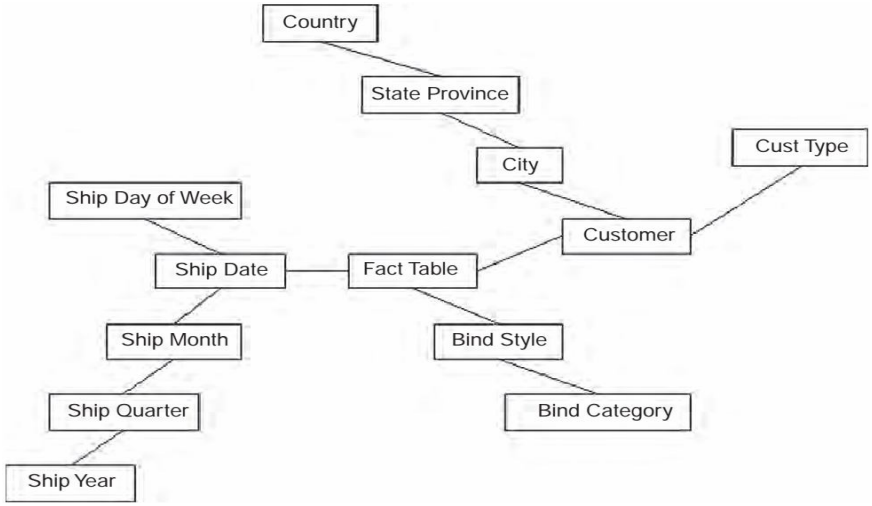
pid	*timeid*	*locid*	*sales*
11	1	1	25
11	2	1	8
11	3	1	15
12	1	1	30
12	2	1	20
12	3	1	50
13	1	1	8
13	2	1	10
13	3	1	10
11	1	2	35
11	2	2	22
11	3	2	10
12	1	2	26
12	2	2	45
12	3	2	20
13	1	2	20
13	2	2	40
13	3	2	5

Sales

Схема пахуљице

- Слично као звездаста схема
- Разлика је у томе што су димензионе табеле нормализоване
  - не морају све да буду нормализоване

Схема пахиљице, пример



Вишеструке табеле чињеница

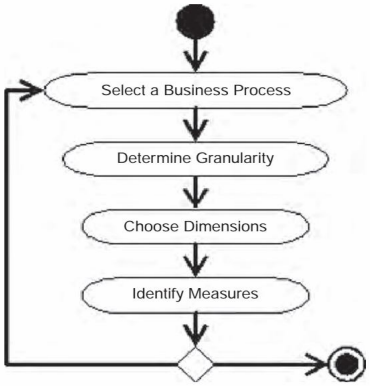
- Једно складиште података може да садржи више табела чињеница
  - за различите теме / скупове података или
  - за исте теме али различите нивое грануларности
- Веома ретко се истим упитом обрађује више табела чињеница
  - зато што свака садржи све податке потребне за ту тему
  - чак и када се користе заједно, обично се не спајају табеле чињеница већ резултати добијени над њима
- Више табела чињеница може да се повезује на исте димензионе табеле
- Могу да се повезују извештаји добијени над њима

Димензионо моделирање

- Димензионо моделирање је основа логичког пројектовања складишта података
- Циљ је прецизирање табеле чињеница и димензионих табела
  - Који подаци се воде непосредно?
    - Обично вредносни подаци који су предмет поређења или израчунавања
  - Који подаци се представљају димензије?
    - Обично подаци по којима се чињенице групишу и класификују
  - Које грануларности димензија су потребне?

Димензионо моделирање (2)

- Обично се ради редом:
  - Одабирање теме (пословног процеса)
  - Одређивање грануларности
  - Одабирање димензија
  - Дефинисање мера
  - Повезивање са другим димензионим моделима
    - опциони корак



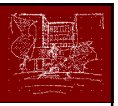
Пример 1.1

- Ако је тема анализирање испита на факултетима УБ онда бисмо у првом кораку могли да препознамо следеће атрибуте чињеница:
  - факултет
  - предмет
  - студент
  - наставник
  - датум и време
  - оцена

Пример 1.2

- Ако очекујемо да се подаци анализирају по месецима и недељама у години, али и по испитним роковима, онда уместо једног атрибута
  - датум и време
- Можемо да уведемо више атрибута
  - датум и време
  - месец
  - редни број недеље
  - испитни рок
  - време
- Добили смо вишеструку грануларност података у табели чињеница
  - наравно, ако знамо датум и време онда бисмо одатле могли увек да израчунамо све остало (осим можда испитни рок, ако има преклапања) али би то захтевало додатне ресурсе
  - вишеструка грануларност уводи редундантност и повећава обим података али омогућава далеко ефикаснији рад
    - могуће је индексирање по сваком од нивоа грануларности

Пример 1.3

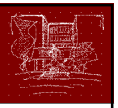


- Ако је потребно да анализирамо податке по групацијама факултета, онда уместо једног атрибута
  - факултет
- можемо да уведемо више атрибута
  - факултет
  - групација

Пример 1.4

- Слично, ако је потребно да анализирамо податке по врстама предмета, онда уместо једног атрибута
  - предмет
- можемо да уведемо више атрибута
  - предмет
  - област предмета
  - врста предмета
  - начин испитивања
  - број часова предмета

Пример 1.5

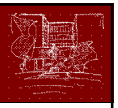


- Наравно, за детаљне анализе желимо много више података о студентима, па уместо једног атрибута
  - студент
- можемо да уведемо више атрибута
  - студент
  - пол студента
  - студијски програм
  - ниво студијског програма
  - редни број уписане године
  - ...

Пример 1.6

- Није другачије ни са наставницима, па уместо једног атрибута
  - наставник
- можемо да уведемо више атрибута
  - наставник
  - звање наставника (у време полагања испита)
  - катедра наставника (у време полагања испита)
  - старост наставника (у време полагања испита)
  - пол наставника (у време полагања испита)
  - врста наставника (стално запослен, спољни сарадник,...)

Пример 1.7



- За одређене врсте анализа можемо да изменимо грануларност чак и наизглед атомичног податка
  - оцена
- који можемо да допунимо са више атрибута
  - оцена
  - позитивна оцена (логичка вредност)
  - висока оцена (логичка вредност)
- или да допунимо описним атрибутом који би представљао димензију
  - врста оцене (нпр. негативна, ниска, висока)

- Од скромног почетног скупа атрибута добијамо, нпр. следећу табелу чињеница:
  - факултет
  - групација
  - предмет
  - област предмета
  - врста предмета
  - начин испитивања
  - број часова предмета
  - студент
  - пол студента
  - студијски програм
  - ниво студијског програма
  - редни број уписане године
  - наставник
  - звање наставника
  - катедра наставника
  - старост наставника
  - пол наставника
  - врста наставника
  - датум и време
  - месец
  - редни број недеље
  - испитни рок
  - школска година
  - време
  - оцена

Пример 1.9

- Након обликовања табеле чињеница, пројектовање димензионих табела је знатно једноставније
- За сваку димензију (атрибут табеле чињеница који није јединствено вредносно одређен) обично правимо по једну димензиону табелу
  - факултет
  - групација
  - студент
  - студијски програм
  - наставник
  - катедра
  - ...
- Обично нема много избора, али их ипак може бити:
  - да ли ћемо да користимо звездасту схему или схему пахуљице
  - да ли ћемо за више грануларности да користимо исту табелу димензија
    - на пример, можемо да направимо димензију "период" и да различити редови у њој представљају дане, недеље, месеце и слично

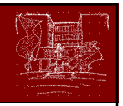
Физичко пројектовање СП (1)

- Пројектовање индекса
  - индекси спајања
    - намењени су за ефикасно повезивање табеле чињеница и димензионих табела
    - праве се на димензионим табелама
    - обично хеш или б-стабла
  - индекси претраживања
    - за ефикасно претраживање табеле чињеница
    - праве се на табели чињеница
    - често бит-мапирани индекси
  - често се прави много индекса
    - убрзавају се упити
    - не постоје трансакције којима би то сметало
    - пакетно ажурирање се обично изводи са одложеним/накнадним освежавањем индекса
- ...

Физичко пројектовање СП (2)

- ...
- Пројектовање материјализованих погледа
  - логички одговарају погледима, а физички одговарају табелама
  - након првог израчунавања резултати се чувају све док се подаци не промене
    - што се у случају складишта дешава ретко
  - ефикаснија алтернатива прављењу више табела чињеница за разчичите нивое грануларности или навођењу редундантних нивоа грануларности у једној табели чињеница
- ...

Физичко пројектовање СП (3)



- ...
- Пројектовање партиција
  - хоризонтално, по групама редова
    - обично по локацијама или временским периодима
  - вертикално, по групама колона
    - обично по различитим карактеристикама
    - слично као да се прави више табела чињеница, које имају однос 1-1

Дистрибуирање података СП

- Обично се тежи да СП буду централизована,
  - ако је неопходно онда се дистрибуирају на више рачунара у једном центру, само због перформанси
- Пројектују се партиционисање, репликација и распоређивање података

Имплементација, праћење и мењање СП

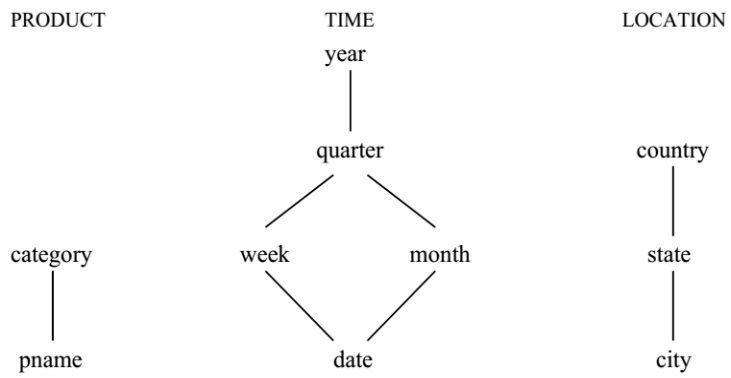
- Прављење табела, погледа, скриптова и других метаподатака
- Процедуре и скриптови за ажурирање података:
  - издвајање података
  - пречишћавање података
  - трансформисање података
  - пуњење података
  - освежавање индекса
  - развој корисничких апликација
- Праћење и унапређивање рада
  - перформансе
  - садржај и квалитет извештаја

Аутоматске збирне табеле

- Једна од препознатљивих карактеристика база података за подршку одлучивању
- Имплементирају се као врста материјализованих погледа
  - у терминологији складишта података и ОЛАП-а се АЗТ често називају и само ***погледи***
- Основна намена је да представљају алтернативне "табеле" чињеница за грубље нивое грануларности
  - најчешће се израчунавају груписањем редова табела чињеница и сумирањем вредносних података
  - на пример, ако табела чињеница садржи податке о вредности продаје по данима, онда АЗТ може да обухвати исте податке, али сумарно по месецима, а друга АЗТ сумарно по годинама
- Користе се за убрзавање израчунавања често употребљаваних статистика
- Сваки пут када наступају промене података, АЗТ се означава као неажурна
  - при првој наредној употреби се прво ажурира па затим и користи
  - ажурирају се само подаци који се односе на мењане податке

Експлозија броја погледа (1)

- Агрегирани материјализовани погледи граде хијерархије у односу на димензије и нивое грануларности
- На пример, ако је T временска димензија
  - табела чињеница у односу на T има ниво агрегације 0
  - први поглед, који агрегира по сатима, има ниво агрегације 1
  - следећи, који агрегира по данима, има ниво агрегације 2
  - итд.
- Ако нека димензија има само један ниво агрегирања, онда
  - основни ниво података представља ниво 0
  - агрегација по свим вредностима димензије је ниво 1

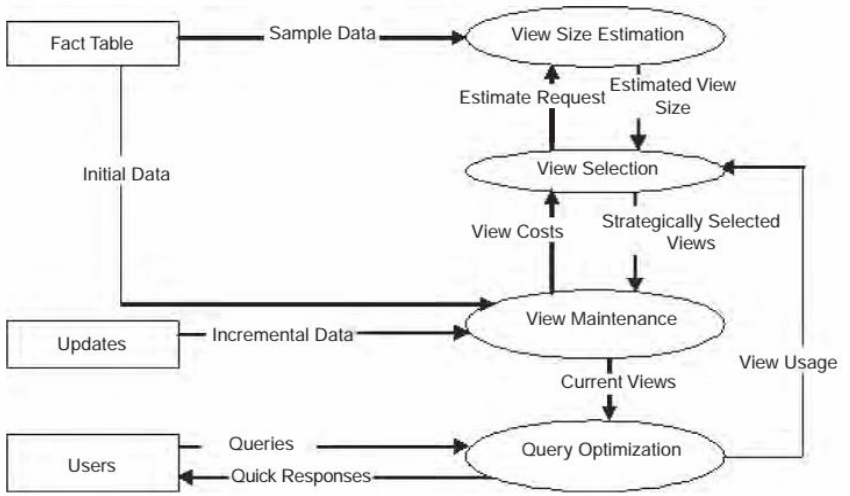


Експлозија броја погледа (2)

- Ако две димензије имају m и n нивоа, онда постоји m x n погледа који покривају све комбинације за ове димензије
- Ако имамо већи број димензија:
  - d је број димензија,
  - h_i су бројеви нивоа по димензијама
  - онда је укупан број могућих погледа: $N = \prod_{i=1}^{d} h_i$
- Ако је g геометријска средина броја нивоа по димензијама, онда је укупан број могућих погледа реда: $N = g^d$
- Често је немогуће направити и одржавати све ове погледе над великим табелама чињеница
- Зато је један од најважнијих корака при пројектовању СП одабирање оних погледа који ће бити направљени и одржавани

План оптимизације погледа

- У суштини представља оптимизациони посао
  - потребно је да се одаберу погледи за материјализацију
  - тако да буде што већа ефикасност аналитичких послова
  - тако да буде што мање поскупљење трансакција
    - тј, у контексту СП, да редовно ажурирање података буде остварив у предвиђеним временским роковима
- Посао се дели на четири основна потпосла:
  - процена величине погледа
  - одабир материјализованих погледа
  - управљање одржавањем материјализованих погледа
  - оптимизација упита над материјализованим погледима



Процена величине погледа (1)

- Што су погледи већи, то је њихово одржавање скупље
- Зато је веома важно је да се добро процени њихова величина
- Карденова (*Cardenas*, 1975) формула, уз претпоставку униформне расподеле:

$$N = v\left(1 - \left(1 - 1/v\right)^n\right)$$

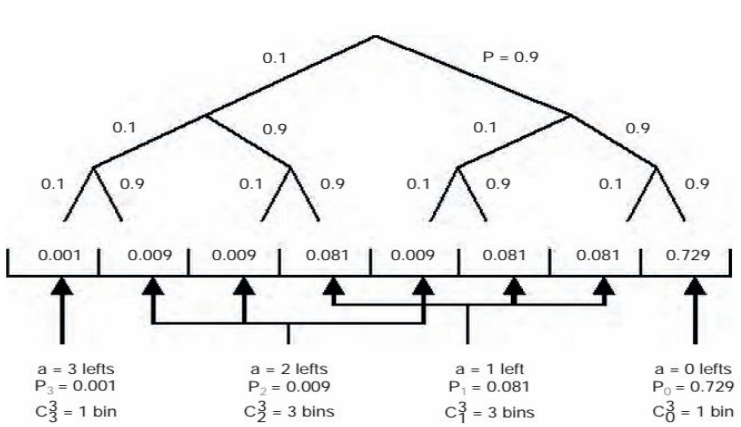
  - N је процењен број редова погледа
  - n је број редова у табели чињеница
  - v је број могућих различитих кључева у простору погледа
    - тј. теоријски највећи могући број редова у погледу
  - Ово је једноставна и ефикасна формула
    - али је претпоставка униформне расподеле сувише рестриктивна
    - зато што су подаци обично груписани на неки начин

Процена величине погледа (2)

- Алтернатива је биномна мултифрактална расподела:

$$N = \sum_{a=0}^{k} C_a^k \left(1 - \left(1 - P_a\right)^n\right)$$

  - k је дубина дрвета одлучивања (груписања)
  - C_a^k је број група података који могу да се достигну избором a левих и k-a десних грана
  - P је вероватноћа избора десне гране на неком месту у дрвету
    - приметимо да се претпоставља да је униформна
  - P_a је вероватноћа достизања изабране групе пролажењем кроз a левих грана $P_a = P^{k-a}(1-P)^a$



Одабир материјализованих погледа

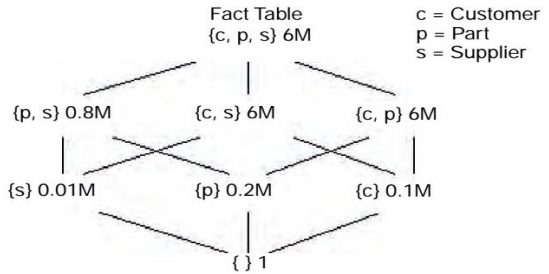
- Алгоритам *HRU* (*Harinarayan*, 1996)
- Итеративни алгоритам праћења локалних минимума
  - не гарантује се глобални оптимум
  - у пракси даје добре резултате за мало димензија
  - велики пораст сложености проблема у случају много димензија
    - реда $O(k\ 2^{2d})$
    - где је k број погледа који се бирају, а d број димензија

Одабир материјализованих погледа – *HRU*

- Прави се структура решетке хиперкоцке
  - екстремни супротни чворови су
    - табела чињеница (све димензије)
    - пуна агрегација (један ред, без димензија)
  - унутрашњи чворови су сви различити погледи по димензијама
    - повезују се чворови који се разликују за по једну агрегацију
  - представљамо упрошћени облик без хијерархије (свака димензија има само нивое 0 и 1), али је слично и са њима
- Сваки чвор се означава неагрегираним димензијама и процењеним бројем редова

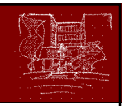
HRU (2)

- Сваки чвор се означава неагрегираним димензијама и процењеним бројем редова
- Ради се у више итерација
  - У сваком кораку се полази се од табеле чињеница
  - Тражи се чвор чије би постојање највише смањило број редова који се обрађују, тако што се за сваки чвор:
    - претпоставља се да користи уштеда Q за тај чвор (рачуна се као разлика броја редова у том чвору у односу на број редова који се користи за рачунање тог чвора)
    - укупна уштеда обухвата уштеду за тај чвор, и све чворове “испод” њега



	Iteration 1 Benefit	Iteration 2 Benefit
{p, s}	**5.2M × 4 = 20.8M**	
{c, s}	0 × 4 = 0	0 × 2 = 0
{c, p}	0 × 4 = 0	0 × 2 = 0
{s}	5.99M × 2 = 11.98M	0.79M × 2 = 1.58M
{p}	5.8M × 2 = 11.6M	0.6M × 2 = 1.2M
{c}	5.9M × 2 = 11.8M	**5.9M × 2 = 11.8M**
{}	6M − 1	0.8M − 1

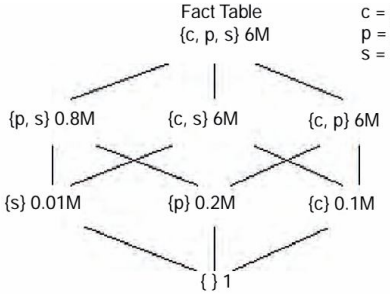
Одабир материјализованих погледа - *HRU*



- У општијем облику
  - ако димензија има више нивоа агрегације, додаје се више хијерархијских чворова
  - према процени учесталости употребе чворовима се могу дати различите тежине
- Слабости
  - У сваком кораку се поново рачуна цена односно добитак за све преостале чворове

Алгоритам ПГА

- ПГА (*Polynomial Greedy Algorithm*) (*Nadeau*, *Teorey*, 2002)
- Такође бира по један поглед у кораку, али уз мање рачунања
- Дели сваки корак на
  - фазу номинације и
  - фазу избора
- Фаза номинације тражи „добре кандидате“
  - номинује се најмањи поглед (или један од њих) који је један ниво испод табеле чињеница,
  - затим најмањи који је један ниво испод номинованог
  - и тако до пуне агрегације
- Фаза избора израчунава добит само за добре кандидате
  - одабире се онај који даје највећу добит
- У наредном циклусу се понавља поступак, али се номинују само кандидати који нису претходно били номиновани



Candidates	Iteration 1 Benefit
{p,s}	**5.2M x 4 = 20.8M**
{s}	5.99M x 2 = 11.98M
{}	6M - 1

Candidates	Iteration 2 Benefit
{c,s}	0 x 2 = 0
{c}	**5.9M x 2 = 11.8M**
{}	6M - 1

Одржавање погледа

- Одржавање погледа обухвата
  - иницијално материјализовање погледа и
  - касније ажурирање материјализованог погледа
- Тежи се да ажурирање што мање касни за стварним временом измене
  - тзв. „*realtime analytics*” или „*active warehousing*”, с тим да се други израз понегде користи и за ОЛАП системе

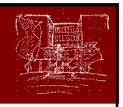
Оптимизација упита

- У највећој мери слично оптимизацијама обичних упита
- Додатно, опитимизација обухвата тражење најбољег пута за израчунавање погледа који нису материјализовани

Карактеристике упита ОЛАП

- Упити над великом количином података
- Агрегатне функције (и сложени облици груписања)
  - различите статистичке функције
- Подаци се често групишу по времену или локацији
- Сложени услови издвајања података
  - велик број конјункција и дисјункција
- Чести су сложени подупити
  - повезивање више извештаја

Честе ОЛАП операције



- Међу најчешће ОЛАП операције спадају:
  - умотавање (*roll-up*)
  - бушење (раскопавање, *drill-down*)
  - унакрсно табелирање

Умотавање

- Умотавање је операција која постепено подиже ниво хијерархије по димензији, све док се не дође до пуне агрегације
- На пример, агрегира се редом
  - по улици
  - по граду
  - по региону
  - по држави
  - све скупа

Умотавање (2)

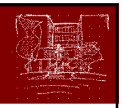
- SQL подржава умотавање клаузулом “постепеног груписања”:

```
SELECT
   a1,
   a2,
   sum(c3)
FROM tabela_cinjenica
GROUP BY
   ROLLUP (a1, a2)
```


Умотавање (3)

- Претходни упит ће у једном извештају обухватити податке који представљају унију три нивоа (колико има димензија плус 1) груписања:
  - SELECT a1, a2, sum(c3)
    FROM tabela_cinjenica
    GROUP BY a1, a2
  - SELECT a1, sum(c3)
    FROM tabela_cinjenica
    GROUP BY a1
  - SELECT sum(c3)
    FROM tabela_cinjenica

Израчунавање хиперкоцке



- Умотавање представља кретање низ једну изабрану путању кроз раније представљени граф хиперкоцке од табеле чињеница до пуне агрегације
- Општији случај анализе је израчунавање свих чворова хиперкоцке у једном пролазу
- За то се користи груписање са клаузулом *CUBE*, где се наводе димензије које нас занимају:
  - SELECT a1, a2, sum(c3)
    FROM tabela_cinjenica
    GROUP BY CUBE(a1, a2)

Израчунавање хиперкоцке (2)

- Претходни упит ће у једном извештају обухватити податке који представљају унију свих нивоа груписања по наведеним атрибутима, тј. унију свих чворова хиперкоцке чији су чворови одређени датим скупом атрибута:
  - SELECT a1, a2, sum(c3)
FROM tabela_cinjenica
GROUP BY a1, a2
  - SELECT a1, sum(c3)
FROM tabela_cinjenica
GROUP BY a1
  - SELECT a2, sum(c3)
FROM tabela_cinjenica
GROUP BY a2
  - SELECT sum(c3)
FROM tabela_cinjenica

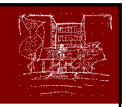
Бушење

- Бушење је операција инверзна умотавању
  - (или *раскопавање*, енгл. *drill-down*)
- Обично не обухвата читав простор димензије већ само сужавање у изабраном смеру
  - потенцијално и по сасвим другој димензији
- На пример,
  - прво посматрамо агрегацију по државама
  - па онда детаљније по градовима у једној држави
  - па онда по години за изабрани град
  - ...

Бушење (2)

- Бушење се имплементира као итеративно
  - прецизирање (проширивање) услова груписања
    - тј. додавање атрибута у клаузуле *SELECT* и *GROUP BY*
  - и фиксирање услова претраживања
    - тј. додавање услова рестрикције у клаузули *WHERE*
- На пример:
  - прво посматрамо агрегацију по државама
  - па фиксирамо једну државу рестрикцијом у клаузули *WHERE*
  - па проширимо груписање на градове додавањем атрибута у клаузуле *SELECT* и *GROUP BY*
  - па фиксирамо један град рестрикцијом у клаузули *WHERE*
  - ...

Ункрсно табелирање



- Изаберу се две димензије и једна мера
- Направи се таблица
  - чије врсте одговарају вредностима једне од димензија
  - чије колоне одговарају вредностима друге димензије
  - садржај ћелија представља агрегацију мере по пресеку димензија
  - сумирају се подаци по врстама и колонама и укупно

Ункрсно табелирање, пример

	WI	CA	Total
1995	63	81	144
1996	38	107	145
1997	75	35	110
Total	176	223	399

Литература за ову тему

- *Teorey, Lightstone, Nadeau, Jagadish,* ***Database Modeling and Design****, Morgan Kaufmann Pubs. 5.ed, 2011.*
- *Ramakrishnan, Gehrke,* ***Database Management Systems****, McGraw Hill, 2.ed, 2000.*
- *Chris Todman,* ***Designing a Data Warehouse****, Prentice Hall PTR, 2001.*
  - превод: *Пројектовање складишта података, ЦЕТ*