



Fizička organizacija BP Data Warehouse sistema

*Modeli fizičke strukture BP
DW sistema*

Sadržaj

- Projektovanje DW sistema
- Tehnike indeksiranja
- Jednostavni bitmap indeks
- Bit-slice indeks
- Kodirani bitmap indeks
- Projekcioni indeks
- Komparacija i upotreba bitmap indeksa
- Spojni indeks
- Fizička organizacija i dimenzionisanje BP
- Fizičko particioniranje BP

Projektovanje DW sistema

- Tipične aktivnosti
 - analiza i specifikacija korisničkih zahteva
 - specifikacija poslovnog modela
 - projektovanje šeme DW BP
 - projektovanje konceptualne šeme DW BP
 - specifikacija logičkog modela
 - projektovanje implementacione šeme DW BP
 - specifikacija dimenzionog modela
 - **projektovanje fizičke organizacije šeme DW BP**
 - **specifikacija fizičkog modela**
 - projektovanje arhitekture DW sistema
 - projektovanje ECTL softverske podrške
 - projektovanje softverske podrške za izveštavanje i analizu podataka

Sadržaj

- Projektovanje DW sistema
- Tehnike indeksiranja
- Jednostavni bitmap indeks
- Bit-slice indeks
- Kodirani bitmap indeks
- Projekcioni indeks
- Komparacija i upotreba bitmap indeksa
- Spojni indeks
- Fizička organizacija i dimenzionisanje BP
- Fizičko particioniranje BP

Tehnike indeksiranja

- OLTP sistemi
 - dominantna upotreba klasičnih B⁺ indeksa
 - ne preveliki broj indeksa po tabeli (maksimalno do 3-4)
 - visoka selektivnost indeksa
- DW sistemi
 - velika količina podataka
 - prevashodno važni upiti
 - zahtev za dobrim performansama upita
 - iniciraju razvoj posebnih tehnika indeksiranja
 - bitmap indeksi
 - projekcioni indeksi
 - spojni indeksi

Tehnike indeksiranja

- Klasični B⁺ indeksi
 - direktni ili reverzni
 - unique ili nonunique
 - kompresovani ili nekompresovani
 - SUBP Oracle: indeksno organizovane tabele (IOT)
- Bitmap indeksi
 - jednostavni bitmap indeksi
 - bit-slice indeksi
 - kodirani bitmap indeksi
- Projekcioni indeksi
- Spojni indeksi

Sadržaj

- Projektovanje DW sistema
- Tehnike indeksiranja
- Jednostavni bitmap indeks
- Bit-slice indeks
- Kodirani bitmap indeks
- Projekcioni indeks
- Komparacija i upotreba bitmap indeksa
- Spojni indeks
- Fizička organizacija i dimenzionisanje BP
- Fizičko particioniranje BP

Jednostavni bitmap indeks

- **Jednostavni bitmap indeks**
 - indeksira se atribut (kolona) A tabele T sa n torke
 - postoji d različitih vrednosti za A u T
 - za svaku vrednost A uvodi se bit vektor dužine n
 - sadrži n binarnih vrednosti, za svaku torku tabele po jedan bit
 - primena tehnike relativnog adresiranja torke tabele
 - posmatra se vrednost $A = a$ i odgovarajući vektor
 - i -ta pozicija u vektoru ima vrednost
 - 1, ako za i -tu torku t_i u tabeli važi $t_i(A) = a$
 - 0, ako za i -tu torku t_i u tabeli važi $t_i(A) \neq a$
 - skup svih d bit vektora indeksiranog atributa predstavlja bitmap indeks

Jednostavni bitmap indeks

- **Jednostavni bitmap indeks**
 - kolone bitmap indeksa predstavljaju bit vektore, svaki dužine d

Bitmap indeks za A

a_1	a_2	...	a_d
0	1	...	0
0	0	...	1
...
1	0	...	0
0	0	...	1
0	1	...	0

Tabela T

...	A	...
	a_2	
	a_d	
	...	
	a_1	
	a_d	
	a_2	

Jednostavni bitmap indeks

- Primer
 - tabela *Student*({*StID*, *StPrg*, *StTip*}, {*StID*})
 - *StID* – identifikaciona oznaka studenta
 - *StPrg* – profil studijskog programa, moguće vrednosti
 - *C* – računarske nauke
 - *M* – matematika
 - *S* – statistika
 - *StTip* - tip studijskog programa, moguće vrednosti
 - *O* – osnovne studije
 - *M* – akademske master studije
 - *D* – doktorske studije
 - indeksiraju se atributi *StPrg* i *StTip*
 - **CREATE BITMAP INDEX** *IDX_Std1* **ON** *Student* (*StPrg*)
 - **CREATE BITMAP INDEX** *IDX_Std2* **ON** *Student* (*StTip*)

Jednostavni bitmap indeks

- Primer

Indeks za *StPrg*

<i>C</i>	<i>M</i>	<i>S</i>
1	0	0
0	1	0
0	0	1
1	0	0
1	0	0
0	0	1
0	1	0
1	0	0
1	0	0
1	0	0

Indeks za *StTip*

<i>D</i>	<i>M</i>	<i>O</i>
0	0	1
1	0	0
1	0	0
0	1	0
0	0	1
0	1	0
0	0	1
0	0	1
0	1	0
0	0	1

Student

<i>StId</i>	<i>StPrg</i>	<i>StTip</i>
07	<i>C</i>	<i>O</i>
13	<i>M</i>	<i>D</i>
55	<i>S</i>	<i>D</i>
10	<i>C</i>	<i>M</i>
97	<i>C</i>	<i>O</i>
11	<i>S</i>	<i>M</i>
01	<i>M</i>	<i>O</i>
23	<i>C</i>	<i>O</i>
33	<i>C</i>	<i>M</i>
02	<i>C</i>	<i>O</i>

Jednostavni bitmap indeks

- Prednost
 - efikasna podrška realizacije upita sa skupovnim funkcijama (SUM, COUNT, itd.) i/ili logičkim operatorima AND i OR
- Razlozi
 - CPU direktno podržava operacije nad bit vektorima, putem posebnih instrukcija
 - pogodnost pri realizaciji operacija unije i preseka skupova
 - omogućena je prethodna priprema upita nad indeksima, bez potrebe pristupanja torkama tabele
 - smanjuje se potreban broj pristupa disku

Jednostavni bitmap indeks

- Primer

- realizacija upita uz pomoć bitmap indeksa

- selektovati studente doktorskih ili master studija koji su upisali studijski program matematike ili statistike

```
SELECT *
```

```
FROM Student
```

```
WHERE (StTip = 'D' OR StTip = 'M')
```

```
      AND (StPrg = 'M' OR StPrg = 'S')
```

Jednostavni bitmap indeks

- Primer

- realizacija upita uz pomoć bitmap indeksa

<i>StTip</i>	<i>D</i>	0	1	1	0	0	0	0	0	0	0
	<i>M</i>	0	0	0	1	0	1	0	0	1	0
	OR1	0	1	1	1	0	1	0	0	1	0

<i>StPrg</i>	<i>M</i>	0	1	0	0	0	0	1	0	0	0
	<i>S</i>	0	0	1	0	0	1	0	0	0	0
	OR2	0	1	1	0	0	1	1	0	0	0

	OR1	0	1	1	1	0	1	0	0	1	0
	OR2	0	1	1	0	0	1	1	0	0	0
	AND	0	1	1	0	0	1	0	0	0	0
	RowId	1	2	3	4	5	6	7	8	9	10

Jednostavni bitmap indeks

- Primer

– realizacija upita uz pomoć bitmap indeksa

Indeks za *StPrg*

<i>C</i>	<i>M</i>	<i>S</i>
1	0	0
0	1	0
0	0	1
1	0	0
1	0	0
0	0	1
0	1	0
1	0	0
1	0	0
1	0	0

Indeks za *StTip*

<i>D</i>	<i>M</i>	<i>O</i>
0	0	1
1	0	0
1	0	0
0	1	0
0	0	1
0	1	0
0	0	1
0	0	1
0	1	0
0	0	1

Student

<i>StId</i>	<i>StPrg</i>	<i>StTip</i>
07	<i>C</i>	<i>O</i>
13	<i>M</i>	<i>D</i>
55	<i>S</i>	<i>D</i>
10	<i>C</i>	<i>M</i>
97	<i>C</i>	<i>O</i>
11	<i>S</i>	<i>M</i>
01	<i>M</i>	<i>O</i>
23	<i>C</i>	<i>O</i>
33	<i>C</i>	<i>M</i>
02	<i>C</i>	<i>O</i>

Jednostavni bitmap indeks

- Primer

- realizacija upita uz pomoć bitmap indeksa
 - selektovati ukupan broj studenata osnovnih studija

```
SELECT COUNT(*)  
FROM Student  
WHERE StPrg = 'O'
```

0	1	0	0	0	1	0	1	1	0	1
---	---	---	---	---	---	---	---	---	---	---

- kompletan upit realizuje se nad bitmap indeksom
 - ne pristupa se uopšte tabeli Student

Jednostavni bitmap indeks

- Nedostaci
 - nepogodan za indeksiranje atributa visoke selektivnosti, kada je broj torki indeksirane tabele veliki
- Razlozi
 - linearan porast dužine bit vektora s porastom broja torki indeksirane tabele n
 - linearan porast broja bit vektora s porastom broja različitih vrednosti indeksiranog atributa, d

Jednostavni bitmap indeks

- Upotrebljava se u slučaju
 - atributa s niskim stepenom selektivnosti
 - selektivnost = d / n (broj različitih vrednosti / broj torki tabele)
 - selektivnost se smatra niskom, ako je $\leq 20\%$
 - u DW, uobičajeno, atributa s vrlo malim brojem različitih vrednosti
- Mogućnost izračunavanja statistike (Oracle)
 - raspodele aktuelnih vrednosti indeksa u domenu, sa aktuelnim brojem torki koje zadovoljavaju datu vrednost
 - **CREATE [BITMAP] INDEX** index **ON** table (...)
COMPUTE STATISTICS

Sadržaj

- Projektovanje DW sistema
- Tehnike indeksiranja
- Jednostavni bitmap indeks
- **Bit-slice indeks**
- Kodirani bitmap indeks
- Projekcioni indeks
- Komparacija i upotreba bitmap indeksa
- Spojni indeks
- Fizička organizacija i dimenzionisanje BP
- Fizičko particioniranje BP

Bit-slice indeks

- **Bit-slice indeks**

- bitmap matrica od n vrsta i m kolona
 - za n torki u indeksiranoj tabeli
 - i -ta vrsta ($i \in \{1, \dots, n\}$) reprezentuje jednu indeksiranu vrednost u binarno kodiranom obliku, koja se pojavljuje u i -toj torci tabele
- analogija sa bitmap indeksom
 - koji bi bio formiran sa d vektora, za d **moćući vrednosti iz domena** indeksa, svaki dužine n
- svaki niz od po d bitova prikazuje se u binarno kodiranom ("upakovanom") obliku
 - umesto što bi se formiralo d bitmap vektora

Bit-slice indeks

- **Bit-slice indeks**

- svaka i -ta vrsta ($i \in \{1, \dots, n\}$), kodira niz od d bitova, putem niza bitova dužine m
 - umesto d "raspakovanih" vektora, svaki dužine n
 - $d = |\text{dom}(X)|$, gde je X indeksirani niz atributa
 - formira se m kodiranih vektora, svaki dužine n
 - $m = \lceil \log_2 d \rceil$
- posebno pogodna struktura ukoliko se indeksira atribut čije su vrednosti celobrojnog tipa

Bit-slice indeks

- Primer

– $d = |\text{dom}(\text{BrPollspita})| = |\{0, \dots, 31\}| = 32, m = 5$

Student

Std	Ime	BrPollspita
07	Pavle	23
13	Jovana	16
10	Dragana	09
55	Zvonko	05
23	Matija	20
27	Daniel	25
01	Goran	04
77	Daniel	13
66	Miloš	19
33	Vladislava	31

Bit-slice indeks nad *BrPollspita*

2^4	2^3	2^2	2^1	2^0
1	0	1	1	1
1	0	0	0	0
0	1	0	0	1
0	0	1	0	1
1	0	1	0	0
1	1	0	0	1
0	0	1	0	0
0	1	1	0	1
1	0	0	1	1
1	1	1	1	1

Bit-slice indeks

- Prednosti
 - pogodan za indeksiranje atributa sa većim brojem mogućih vrednosti iz domena
 - u tom slučaju, zauzima manje memorijskog prostora od običnog bitmap indeksa
 - efikasnost upita zadržana na istom nivou kao kod običnih bitmap indeksa
- Nedostatak
 - složeniji algoritmi za korišćenje bit-slice indeksa

Bit-slice indeks

- Primer

- realizacija upita uz pomoć bit-slice indeksa

- selektovati studente sa brojem položenih ispita većim od 15

```
SELECT *
```

```
FROM Student
```

```
WHERE BrPollspita > 15
```

- dovoljno je samo selektovati torke za koje bitovi vektora 2^4 imaju vrednost 1

2^4	1	1	0	0	1	1	0	0	1	1
RowId	1	2	3	4	5	6	7	8	9	10

Bit-slice indeks

- Primer
 - realizacija upita uz pomoć bit-slice indeksa

Student

<i>Stld</i>	<i>Ime</i>	<i>BrPollspita</i>	<i>Rowid</i>	2^4
07	Pavle	23	1	1
13	Jovana	16	2	1
10	Dragana	09	3	0
55	Zvonko	05	4	0
23	Matija	20	5	1
27	Daniel	25	6	1
01	Goran	04	7	0
77	Daniel	13	8	0
66	Miloš	19	9	1
33	Vladislava	31	10	1

Bit-slice indeks

- Primer

- realizacija upita uz pomoć bit-slice indeksa

- selektovati ukupan broj položenih ispita studenata

```
SELECT SUM(BrPollspita)
```

```
FROM Student
```

- algoritam

- Korak 1: $SUM \leftarrow 0$, $n \leftarrow 5$

- Korak 2: For $i = 0$, $n - 1$ do

- FETCH i-ti bit vektor indeksa nad BrPollspita na poziciji 2^i INTO $V(i)$

- $SUM \leftarrow SUM + 2^i * SUM(V(i))$

- // $SUM(V(i))$ sabira sve članove vektora – vraća broj pojava 1 u vektoru

- upit se u potpunosti realizuje nad indeksom

- bez pristupa torkama tabele

Bit-slice indeks

- Primer

– realizacija upita uz pomoć bit-slice indeksa

2^0	1	0	1	1	0	1	0	1	1	1
-------	---	---	---	---	---	---	---	---	---	---

$$\text{SUM}(V(0)) = 7, \text{SUM} = 0 + 7 \cdot 2^0 = 7$$

2^1	1	0	0	0	0	0	0	0	1	1
-------	---	---	---	---	---	---	---	---	---	---

$$\text{SUM}(V(1)) = 3, \text{SUM} = 7 + 3 \cdot 2^1 = 13$$

2^2	1	0	0	1	1	0	1	1	0	1
-------	---	---	---	---	---	---	---	---	---	---

$$\text{SUM}(V(2)) = 6, \text{SUM} = 13 + 6 \cdot 2^2 = 37$$

2^3	0	0	1	0	0	1	0	1	0	1
-------	---	---	---	---	---	---	---	---	---	---

$$\text{SUM}(V(3)) = 4, \text{SUM} = 37 + 4 \cdot 2^3 = 69$$

2^4	1	1	0	0	1	1	0	0	1	1
-------	---	---	---	---	---	---	---	---	---	---

$$\text{SUM}(V(4)) = 6, \text{SUM} = 69 + 6 \cdot 2^4 = 165$$

Sadržaj

- Projektovanje DW sistema
- Tehnike indeksiranja
- Jednostavni bitmap indeks
- Bit-slice indeks
- Kodirani bitmap indeks
- Projekcioni indeks
- Komparacija i upotreba bitmap indeksa
- Spojni indeks
- Fizička organizacija i dimenzionisanje BP
- Fizičko particioniranje BP

Kodirani bitmap indeks

- **Kodirani bitmap indeks**

- generalizacija bit-slice indeksa
- primenljiv nad atributima bilo kog tipa
- bitmap matrica od n vrsta i m kolona
 - za n torci u indeksiranoj tabeli
 - svaka i -ta vrsta ($i \in \{1, \dots, n\}$), kodira indeksiranu vrednost u i -toj torci putem niza bitova dužine m
 - d je broj **različitih aktuelnih vrednosti** indeksiranog atributa
 - formira se m kodiranih vektora, svaki dužine n
 - $m = \lceil \log_2 d \rceil$
- zajedno sa tabelom kodiranih vrednosti
 - sadrži d vrsta, za svaku aktuelnu vrednost indeksa po jednu
 - svakoj vrednosti indeksa pridružuje se binarni kod dužine m

Kodirani bitmap indeks

- Primer

- $d(\text{Ime}) = 9$, $m = \lceil \log_2 9 \rceil = 4$

<i>Ime</i>	2^3	2^2	2^1	2^0
<i>Daniel</i>	0	0	0	0
<i>Dragana</i>	0	0	0	1
<i>Goran</i>	0	0	1	0
<i>Jovana</i>	0	0	1	1
<i>Matija</i>	0	1	0	0
<i>Miloš</i>	0	1	0	1
<i>Pavle</i>	0	1	1	0
<i>Vladislava</i>	0	1	1	1
<i>Zvonko</i>	1	0	0	0

Kodirani bitmap indeks

- Primer

– $d(\text{Ime}) = 9$, $m = \lceil \log_2 9 \rceil = 4$

Student

<i>StId</i>	<i>Ime</i>	<i>BrPollspita</i>
07	Pavle	23
13	Jovana	16
10	Dragana	09
55	Zvonko	05
23	Matija	20
27	Daniel	25
01	Goran	04
77	Daniel	13
66	Miloš	19
33	Vladislava	31

Kodirani bitmap indeks nad *Ime*

2^3	2^2	2^1	2^0
0	1	1	0
0	0	1	1
0	0	0	1
1	0	0	0
0	1	0	0
0	0	0	0
0	0	1	0
0	0	0	0
0	1	0	1
0	1	1	1

Kodirani bitmap indeks

- Primer

- realizacija upita uz pomoć kodiranog bitmap indeksa

```
SELECT *  
FROM Student  
WHERE Ime LIKE 'D%'
```

- da bi upit bio realizovan, neophodno je koristiti bitmap matricu i tabelu kodiranih vrednosti indeksa
- opšti zahtev
 - da procesor upita bude osposobljen da, u što većem broju slučajeva, obavi sve pripremne radnje nad indeksom, pa tek onda da pristupa indeksiranoj tabeli

Kodirani bitmap indeks

- Primer

- realizacija upita uz pomoć kodiranog bitmap indeksa
- Korak 1: pristup tabeli kodiranih vrednosti
 - zaključak: pretražuje se tako da bude
 - $\neg 2^3 \wedge \neg 2^2 \wedge \neg 2^1 \wedge (\neg 2^0 \vee 2^0) \Leftrightarrow \neg 2^3 \wedge \neg 2^2 \wedge \neg 2^1$
- Korak 2: pristup bitmap matrici, vektorima 2^1 , 2^2 i 2^3
 - primenjuje se invertovanje vrednosti i logički AND

$\neg 2^1$	0	0	1	1	1	1	0	1	1	1
$\neg 2^2$	0	1	1	1	0	1	1	1	0	1
$\neg 2^3$	1	1	1	0	1	1	1	1	1	0
AND	0	0	1	0	0	1	0	1	0	0
RowId	1	2	3	4	5	6	7	8	9	10

Kodirani bitmap indeks

- Primer

- realizacija upita uz pomoć kodiranog bitmap indeksa

Student

<i>Std</i>	<i>Ime</i>	<i>BrPollspita</i>	<i>RowId</i>
07	Pavle	23	1
13	Jovana	16	2
10	Dragana	09	3
55	Zvonko	05	4
23	Matija	20	5
27	Daniel	25	6
01	Goran	04	7
77	Daniel	13	8
66	Miloš	19	9
33	Vladislava	31	10

Kodirani bitmap indeks nad *Ime*

2^3	2^2	2^1	2^0
0	1	1	0
0	0	1	1
0	0	0	1
1	0	0	0
0	1	0	0
0	0	0	0
0	0	1	0
0	0	0	0
0	1	0	1
0	1	1	1

Kodirani bitmap indeks

- **Ažuriranje kodiranog bitmap indeksa**

- dodavanje novih toriki u tabelu činjenica dovodi do produžavanja svih vektora bitmap indeksa
- broj vektora bitmap indeksa nad atributom A zavisi od broja aktuelnih vrednosti d
- ažuriranje tabele činjenica može dovesti do povećanja broja aktuelnih vrednosti, i time izazvati
 - potrebu ažuriranja tabele kodiranih vrednosti i
 - uvođenje novog bitmap vektora
 - neka je aktuelni broj bitmap vektora $m = \lceil \log_2 d_0 \rceil$, za početni broj aktuelnih vrednosti indeksa d_0
 - novi bitmap vektor se uvodi kada broj aktuelnih vrednosti $d > d_0$ prevaziđe vrednost

$$d > 2^{\lceil \log_2 d_0 \rceil}$$

Sadržaj

- Projektovanje DW sistema
- Tehnike indeksiranja
- Jednostavni bitmap indeks
- Bit-slice indeks
- Kodirani bitmap indeks
- Projekcioni indeks
- Komparacija i upotreba bitmap indeksa
- Spojni indeks
- Fizička organizacija i dimenzionisanje BP
- Fizičko particioniranje BP

Projekcioni indeks

- **Projekcioni indeks (Projection Index)**
 - predstavlja niz vrednosti indeksiranih atributa
 - takav da uključuje duplikate istih vrednosti
 - pozicioniranih tačno u onom redosledu kako se indeksirane vrednosti pojavljuju u tabeli
 - relativna pozicija indeksirane vrednosti u nizu predstavlja i redni broj torke koja sadrži datu vrednost
 - razlike u odnosu na kodirani bitmap indeks
 - ne postoji tabela kodiranih vrednosti
 - indeksirane vrednosti se direkno koriste u indeksu

Projekcioni indeks

- Primer

Student

<i>Std</i>	<i>Ime</i>	<i>BrPollspita</i>
07	<i>Pavle</i>	23
13	<i>Jovana</i>	16
10	<i>Dragana</i>	09
55	<i>Zvonko</i>	05
23	<i>Matija</i>	20
27	<i>Daniel</i>	25
01	<i>Goran</i>	04
77	<i>Daniel</i>	13
66	<i>Miloš</i>	19
33	<i>Vladislava</i>	31

Projekcioni indeks nad *BrPollspita*

<i>BrPollspita</i>
23
16
09
05
20
25
04
13
19
31

Projekcioni indeks

- Primer

- posmatra se tabela činjenica Realizacija

- obim: $100 \cdot 10^6$ torki
- kapacitet torke: 100 B
- kapacitet atributa Iznos: 4 B
- kapacitet bloka SUBP: 4 KB
- broj torki po bloku: $\lfloor 4 \text{ KB} / 100 \text{ B} \rfloor = 40$
- kapacitet tabele: **$2,5 \cdot 10^6$ blokova**

- posmatra se projekcioni indeks nad atributom Iznos

- broj vrednosti tipa Iznos u bloku: $\lfloor 4 \text{ KB} / 4 \text{ B} \rfloor = 1024$
- kapacitet projekcionog indeksa: **$\sim 0,1 \cdot 10^6$ blokova**

Projekcioni indeks

- Primer

- posmatra se upit

```
SELECT SUM(Iznos) FROM Realizacija
```

- zahtevani broj pristupa disku

- bez projekcionog indeksa: $2,5 \cdot 10^6$

- sa projekcionim indeksom: $0,1 \cdot 10^6$

- upotreba projekcionog indeksa $\Rightarrow 2,5 / 0,1 = 25$ puta manje pristupa

Sadržaj

- Projektovanje DW sistema
- Tehnike indeksiranja
- Jednostavni bitmap indeks
- Bit-slice indeks
- Kodirani bitmap indeks
- Projekcioni indeks
- Komparacija i upotreba bitmap indeksa
- Spojni indeks
- Fizička organizacija i dimenzionisanje BP
- Fizičko particioniranje BP

Komparacija i upotreba bitmap indeksa



- Komparacija tehnika indeksiranja
 - načini adresiranja
 - **B/B+ indeks**
 - koristi relativno adresiranje, zasnovano na eksplicitnom memorisanju pointera u čvorovima/listovima B stabla
 - indeksirane vrednosti ne ponavljaju se kompletno u indeksu
 - moguće je zadržati ili izbeći ponavljanje delimičnih vrednosti ("prefiksa" vrednosti), u slučaju indeksa složenih iz više atributa
 - **bitmap, bit-slice, kodirani bitmap i projekcioni indeks**
 - koriste relativno adresiranje, zasnovano na pozicioniranju indeksirane vrednosti u nizu indeksa
 - dužina niza indeksiranih vrednosti odgovara broju torki indeksirane tabele n
 - ponavljanje isitih vrednosti u indeksu, po pravilu, dozvoljeno je

Komparacija i upotreba bitmap indeksa



- Komparacija tehnika indeksiranja
 - načini pakovanja indeksiranih vrednosti
 - **bitmap indeks**
 - onoliko bitmap vektora, koliko različitih indeksiranih vrednosti
 - redundantno korišćenje bitova (d bitova) za indeksiranje iste relativne pozicije
 - "raspakovana" ("nekodirana", "sparse") indeksna struktura
 - **bit-slice, kodirani bitmap i projekcioni indeks**
 - onoliko vrednosti u indeksnom nizu, koliko indeksiranih vrednosti u tabeli
 - ponavljanje istih indeksiranih vrednosti
 - "upakovana" ("kodirana", "dense") indeksna struktura

Komparacija i upotreba bitmap indeksa



- Komparacija tehnika indeksiranja
 - kodiranje indeksiranih vrednosti
 - **bit-slice i kodirani bitmap indeks**
 - kodirane vrednosti sadržane u indeksu
 - **projekcioni indeks**
 - originalne vrednosti sadržane u indeksu
 - korišćenje pomoćnih struktura
 - **kodirani bitmap indeks**
 - zahteva upotrebu tabele kodiranih vrednosti
 - **bit-slice i projekcioni indeks**
 - ne zahtevaju upotrebu tabele kodiranih vrednosti

Komparacija i upotreba bitmap indeksa

- Komparacija tehnika indeksiranja
 - s obzirom na upite nad zvezdastom šemom
 - 5 – odlične performanse
 - 4 – vrlo dobre performanse
 - 3 – dobre performanse
 - 2 – zadovoljavajuće (ne loše) performanse
 - 1 – loše performanse
 - 0 – vrlo loše performanse

Vrsta agregacije	B stablo	Projekcioni indeks	Bit-slice indeks
COUNT	3	4	5
SUM	2	3	5
AVG	2	3	5
MAX i MIN	5	1	1
GROUP BY (lista)	0	5	0

Komparacija i upotreba bitmap indeksa

- **Bitmap indeksi nad zvezdastom šemom**
- Primer
 - data zvezdasta šema sa denormalizovanim dimenzijama
 - dimenzije: *Vreme, Proizvod i Prodavnica*
 - atributi
 - » *Vreme.Vremeld, Vreme.Tromesec*
 - » *Proizvod.Proizld, Proizvod.TipProizvoda*
 - » *Prodavnica.Prold, Prodavnica.Grad*
 - činjenice: *Realizacija*, sa atributom *Iznos*
 - **bitmap indeksi nad atributima – kolonama činjenice**
 - *Realizacija.Prold,*
 - *Realizacija.Proizld,*
 - *Realizacija.Vremeld*

Komparacija i upotreba bitmap indeksa



- Primer

- realizacija upita nad zvezdastom šemom

```
SELECT Grad, Tromesec, SUM(Iznos)
FROM Realizacija
    NATURAL JOIN Prodavnica
    NATURAL JOIN Proizvod
    NATURAL JOIN Vreme
WHERE Grad IN ('BG', 'NS', 'KG')
    AND TipProizvoda = 'Automobil'
    AND Tromesec IN ('1Q2003', '2Q2003', '3Q2003')
GROUP BY Grad, Tromesec
```

Komparacija i upotreba bitmap indeksa



- Primer
 - realizacija upita nad zvezdastom šemom
 - realizacija koja bi vršila spajanja svih činjenica sa svim selektovanim torkama dimenzija bila bi veoma neefikasna
 - cilj
 - iskoristiti indekse nad stranim ključevima u tabeli činjenica i pristupati samo neophodnim torkama činjenica
 - mogućnost koja postoji i sa B/B⁺ indeksima
 - iskoristiti posebne pogodnosti primene bitmap indeksa
 - za upite konjuktivnog i disjunktivnog tipa

Komparacija i upotreba bitmap indeksa



- Primer
 - realizacija upita nad zvezdastom šemom
 - **Korak 1**
 - **(1A)** selekcija skupova vrednosti ključeva dimenzija, prema kriterijumima
 - *Prodavnica.ProdId* (*Grad* IN ('BG', 'NS', 'KG'))
 - *Proizvod.ProizId* (*TipProizvoda* = 'Automobil')
 - *Vreme.VremId* (*Tromesec* IN ('1Q2003', '2Q2003', '3Q2003'))
 - **(1B)** optimizator sada može upotrebiti bitmap indekse nad stranim ključevima u tabeli činjenica
 - zato da se izdvoje samo one torke činjenica koje učestvuju u izračunavanju rezultata
 - » umesto što bi se, po svakom indeksu, pojedinačno pristupalo tabeli činjenica
 - izvršavanjem odgovarajućih bit orijentisanih AND i/ili OR operacija nad bitmap vektorima potrebnih indeksa

Komparacija i upotreba bitmap indeksa



- Primer
 - realizacija upita nad zvezdastom šemom
 - **Korak 2**
 - spajanje selektovanih torki činjenica sa torkama dimenzija
 - izračunavanje vrednosti skupovnih funkcija

Komparacija i upotreba bitmap indeksa



- Primer

- realizacija upita nad zvezdastom šemom

- **Korak 1**

```
CREATE VIEW Medjurezultat_Cinjenice AS (  
  SELECT * FROM Realizacija  
  WHERE  
    ProId IN ( SELECT ProId FROM Prodavnica  
              WHERE Grad IN ('BG', 'NS', 'KG')) AND  
    ProizId IN (SELECT ProizId FROM Proizvod  
              WHERE TipProizvoda = 'Automobil') AND  
    Vremeld IN (SELECT Vremeld FROM Vreme  
              WHERE Tromesec IN  
                ('1Q2003', '2Q2003', '3Q2003'))  
)
```

Komparacija i upotreba bitmap indeksa

- Primer

- realizacija upita nad zvezdastom šemom

- **Korak 1**

```
SELECT * FROM Realizacija
```

```
WHERE
```

```
    ProdId IN ( SELECT ProdId FROM Prodavnica  
                WHERE Grad IN ('BG', 'NS', 'KG')) AND
```

```
...
```

- Način izvršavanja ugnježdenog upita
 - za svaki selektovani ProdId, u bitmap indeksu *Realizacija.ProdId* selektuju se odgovarajući bit vektori
 - vrši se bit orijentisani OR svih selektovanih vektora
 - rezultujući bit vektor ukazuje na sve torke činjenica koje zadovoljavaju uslov *Grad IN ('BG', 'NS', 'KG')*
- Izvršavanje ostalih ugnježdenih upita - analogno

Komparacija i upotreba bitmap indeksa

- Primer

- realizacija upita nad zvezdastom šemom

- **Korak 1**

```
SELECT * FROM Realizacija
```

```
WHERE
```

```
  ProdId IN ( SELECT ... ) AND
```

```
  ProizId IN ( SELECT ... ) AND
```

```
  VremId IN (SELECT ...)
```

- Način izvršavanja glavnog upita

- bit orijentisani AND tri prethodno dobijena, rezultujuća bit vektora

- » formiranje konačnog bit vektora

- selektovanje samo onih torke činjenica koje dogovaraju pozicijama 1 u konačno dobijenom bit vektoru

Komparacija i upotreba bitmap indeksa



- Primer

- realizacija upita nad zvezdastom šemom
- **Korak 2**

```
SELECT Grad, Tromesec, SUM(Iznos)
FROM Medjurezultat_Cinjenice
    NATURAL JOIN Prodavnica
    NATURAL JOIN Vreme
GROUP BY Grad, Tromesec
```

- Način izvršavanja upita
 - spajanjem prethodno selektovanih torke tabela zvezdaste šeme

Sadržaj

- Projektovanje DW sistema
- Tehnike indeksiranja
- Jednostavni bitmap indeks
- Bit-slice indeks
- Kodirani bitmap indeks
- Projekcioni indeks
- Komparacija i upotreba bitmap indeksa
- **Spojni indeks**
- Fizička organizacija i dimenzionisanje BP
- Fizičko particioniranje BP

Spojni indeks

- **Spojni indeks (Join index)**
 - indeks koji obuhvata više od jedne tabele
 - za razliku od tradicionalnih indeksa koji se kreiraju nad jednom tabelom
 - sa unapred pripremljenim podacima koji su rezultat spajanja nad zvezdastom šemom
 - indeksirane vrednosti povezuju se sa relativnim adresama torki tabele koje bi trebalo spajati
 - motivacija
 - obezbeđenje dobrih performansi upita koji zahtevaju spajanje vrlo velikih relacija
 - time što bi rezultati spajanja bili unapred pripremljeni

Spojni indeks

- **Spojni indeks (Join index)**
 - može biti organizovan kao
 - bitmap indeks ili
 - klasični indeks s B⁺ stablom
 - predstavlja strukturu uređenog stabla (stabla traženja)
 - uz indeksiranu vrednost u listu stabla nalazi se
 - niz relativnih adresa oblika (r_1, r_2, \dots, r_n) , gde je svaki r_i relativna adresa indeksirane torke ili
 - bitmap vektor

Spojni indeks

- **Spojni indeks u kontekstu zvezdaste šeme**
 - povezuje vrednosti indeksiranih atributa dimenzija sa relativnim adresama odgovarajućih torke tabele činjenica
 - primer
 - dimenzija: $Prodavnica(\{Prodlid, Grad, \dots\}, \{Prodlid\})$
 - činjenica: $Realizacija(\{Releid, \dots, Prodlid, \dots, Iznos\}, \{Releid\})$
 - spojni indeks
 - indeksirane vrednosti atributa $Prodavnica.Grad$
 - u tabelama $Realizacija$ i $Prodavnica$
 - uz svaku indeksiranu vrednost pojavljuje se lista parova relativnih adresa (p, r)
 - » p - relativna adresa torke dimenzije $Prodavnica$
 - » r - relativna adresa torke tabele činjenica $Realizacija$
 - takvih, da važi $Prodavnica.Prodlid = Realizacija.Prodlid$

Spojni indeks

- Primer

Prodavnica

Prodl	Grad	Rowld
s_1	NS	1
s_2	NS	2
s_3	KG	3

Spojni indeks nad Grad

Grad	(Rowld(Prodavnica), Rowld(Realizacija))
NS	(1, 1), (1, 2), (1, 7), (1, 8), (2, 3), (2, 4), (2, 9), (2, 10)
KG	(3, 5), (3, 6), (3, 11), (3, 12)

Rowld	Prodl	Vremeld	Proizld	Iznos
1	s_1	t_1	p_1	230
2	s_1	t_1	p_2	300
3	s_2	t_1	p_1	550
4	s_2	t_1	p_2	100
5	s_3	t_1	p_1	60
6	s_3	t_1	p_2	140
7	s_1	t_2	p_1	310
8	s_1	t_2	p_2	50
9	s_2	t_2	p_1	880
10	s_2	t_2	p_2	60
11	s_3	t_2	p_1	350
12	s_3	t_2	p_2	350

Spojni indeks

- Primer

- SUBP Oracle: kreiranje bitmap join indeksa

```
CREATE BITMAP INDEX IDX_prodavn_realiz_grd  
ON Realizacija(Prodavnica.Grad)  
FROM Realizacija, Prodavnica  
WHERE Realizacija.ProdId = Prodavnica.ProdId
```

- Upit koji se realizuje uz pomoć spojnog indeksa

```
SELECT SUM(Iznos)  
FROM Realizacija, Prodavnica  
WHERE Realizacija.ProdId = Prodavnica.ProdId  
      AND Prodavnica.Grad = 'NS'
```

Spojni indeks

- **Spojni indeks u kontekstu zvezdaste šeme**
 - spojni indeks može biti kreiran kao generalizovana struktura nad više atributa različitih dimenzija
 - posmatra se zvezdasta šema
 - dimenzija D1, sa atributom A1,
 - dimenzija D2, sa atributom A2 i
 - dimenzija D3 i tabela činjenica F
 - često upotrebljavana šema upita nad zvezdastom šemom

```
SELECT <lista_atributa>, <skupovna_funkcija>
FROM F, D1, D2
WHERE <uslov_spajanja> AND A1 = a1 AND A2 = a2
GROUP BY <lista_atributa>
```

Spojni indeks

- **Spojni indeks u kontekstu zvezdaste šeme**
 - moguće je formirati spojni indeks nad (A1, A2)
 - u cilju obezbeđenja dobrih performansi navedenog upita
 - uz svaku indeksiranu vrednost (A1, A2) pojavljuje se lista uređenih trojki relativnih adresa (r_1, r_2, r_3)
 - r_1 – RowId torke dimenzije D1
 - r_2 – RowId torke dimenzije D2
 - r_3 – RowId torke tabele činjenica F
 - takvih, da važi uslov spajanja
 - $F.K1 = D1.K1$ AND $F.K2 = D2.K2$
 - gde je K_i primarni ključ dimenzije $D_i, i \in \{1, 2\}$

Spojni indeks

- **Spojni indeks u kontekstu zvezdaste šeme**
 - nedostatak prethodnog rešenja
 - porast broja dimenzija dovodi do "eksplozije" potrebnog broja takvih indeksa
 - za zvezdastu šemu sa n dimenzija potrebno je $n(n-1)/2$ spojna indeksa
 - da pokriju kombinacije dimenzija "svaka sa svakom"
 - a i dalje je pokrivena selekcija podataka iz svake dimenzije samo po jednom atributu
 - alternativno rešenje
 - kreirati po jedan spojni indeks za svaku dimenziju i izabrane attribute
 - pri realizaciji upita, selektovati skupove traženih RowId-ova i primeniti skupovne operacije (predizračunavanje) pre pristupa torkama u tabelama

Spojni indeks

- Primer

Prodavnica

<i>Prodl</i>	<i>Grad</i>	<i>Rowld</i>
s_1	NS	1
s_2	NS	2
s_3	KG	3

Vreme

<i>Vremeld</i>	<i>Godina</i>	<i>Rowld</i>
t_1	2000	1
t_2	2001	2

Proizvod

<i>Proizld</i>	<i>Naziv</i>	<i>Rowld</i>
p_1	šporet	1
p_2	pegla	2

<i>Rowld</i>	<i>Prodl</i>	<i>Vremeld</i>	<i>Proizld</i>	<i>Iznos</i>
1	s_1	t_1	p_1	230
2	s_1	t_1	p_2	300
3	s_2	t_1	p_1	550
4	s_2	t_1	p_2	100
5	s_3	t_1	p_1	60
6	s_3	t_1	p_2	140
7	s_1	t_2	p_1	310
8	s_1	t_2	p_2	50
9	s_2	t_2	p_1	880
10	s_2	t_2	p_2	60
11	s_3	t_2	p_1	350
12	s_3	t_2	p_2	350

Spojni indeks

- Primer

- realizacija upita nad zvezdastom šemom
- upit

```
SELECT p.Naziv, SUM(f.Iznos) AS Sum
FROM Realizacija f, Proizvod p, Vreme t, Prodavnica r
WHERE f.Vremeld = t.Vremeld
      AND f.ProdId = r.ProdId
      AND f.ProizId = p.ProizId
      AND p.Grad = 'NS' AND t.Godina = 2000
GROUP BY p.Naziv
```

Spojni indeks

- Primer
 - realizacija upita nad zvezdastom šemom
 - potrebni spojni indeksi

Spojni indeks nad *Grad*

<i>Grad</i>	Lista RowId(<i>Prodavnica</i>)	Lista RowId(<i>Realizacija</i>)
<i>NS</i>	1, 2	1, 2, 3, 4, 7, 8, 9, 10
<i>KG</i>	3	5, 6, 11, 12

Spojni indeks nad *Godina*

<i>Godina</i>	Lista RowId(<i>Vreme</i>)	Lista RowId(<i>Realizacija</i>)
<i>2000</i>	1	1, 2, 3, 4, 5, 6
<i>2001</i>	2	7, 8, 9, 10, 11, 12

Spojni indeks

- Primer
 - realizacija upita nad zvezdastom šemom
 - selektovanje vrednosti za *RowId* na osnovu spojnog indeksa nad *Grad*
 - $RowId(Prodavnica)(NS) = \{1, 2\}$
 - $RowId(Realizacija)(NS) = \{1, 2, 3, 4, 7, 8, 9, 10\}$
 - selektovanje vrednosti za *RowId* na osnovu spojnog indeksa nad *Vreme*
 - $RowId(Vreme)(2000) = \{1\}$
 - $RowId(Realizacija)(2000) = \{1, 2, 3, 4, 5, 6\}$

Spojni indeks

- Primer

- realizacija upita nad zvezdastom šemom

- predizračunavanje – formiranje skupa traženih *RowId* u tabeli činjenica

- $RowId(Realizacija) =$

- $RowId(Realizacija)(NS) \cap RowId(Realizacija)(2000) =$

- $\{1, 2, 3, 4, 7, 8, 9, 10\} \cap \{1, 2, 3, 4, 5, 6\} = \{1, 2, 3, 4\}$

- pribavljanje iz tabela torki

- iz tabele *Prodavnica*, $RowId(Prodavnica)(NS) = \{1, 2\}$

- iz tabele *Vreme*, $RowId(Vreme)(2000) = \{1\}$

- iz tabele *Realizacija*, $RowId(Realizacija) = \{1, 2, 3, 4\}$

- spajanje pribavljenih torki

Spojni indeks

- Primer

Prodavnica

<i>Prodl</i>	<i>Grad</i>	<i>Rowld</i>
s_1	NS	1
s_2	NS	2
s_3	KG	3

Vreme

<i>Vremeld</i>	<i>Godina</i>	<i>Rowld</i>
t_1	2000	1
t_2	2001	2

Proizvod

<i>Proizld</i>	<i>Naziv</i>	<i>Rowld</i>
p_1	šporet	1
p_2	pegla	2

<i>Rowld</i>	<i>Prodl</i>	<i>Vremeld</i>	<i>Proizld</i>	<i>Iznos</i>
1	s_1	t_1	p_1	230
2	s_1	t_1	p_2	300
3	s_2	t_1	p_1	550
4	s_2	t_1	p_2	100
5	s_3	t_1	p_1	60
6	s_3	t_1	p_2	140
7	s_1	t_2	p_1	310
8	s_1	t_2	p_2	50
9	s_2	t_2	p_1	880
10	s_2	t_2	p_2	60
11	s_3	t_2	p_1	350
12	s_3	t_2	p_2	350

Sadržaj

- Projektovanje DW sistema
- Tehnike indeksiranja
- Jednostavni bitmap indeks
- Bit-slice indeks
- Kodirani bitmap indeks
- Projekcioni indeks
- Komparacija i upotreba bitmap indeksa
- Spojni indeks
- Fizička organizacija i dimenzionisanje BP
- Fizičko particioniranje BP

Fizička organizacija i dimenzionisanje BP



- Upravljanje fizičkom organizacijom BP
 - u okvirima projektovane arhitekture DW sistema
 - zadaci
 - projektovanje fizičke organizacije BP
 - implementacija fizičke organizacije BP
 - praćenje, analiza i izmene fizičke organizacije BP

Fizička organizacija i dimenzionisanje BP



- Specifikacija fizičke organizacije BP
 - dimenzionisanje BP
 - specifikacija vrsta i organizacije datoteka SUBP
 - specifikacija fizičkog prostora BP
 - specifikacija radnog prostora OM za SUBP



- **Dimenzionisanje BP**

- procena potrebnog kapaciteta BP DW sistema
 - za DSA i za DW bazu podataka
 - uključuje procene potrebnih kapaciteta za
 - tabele s podacima
 - indekse za tabele
 - materijalizovane poglede sa agregiranim podacima
 - ostale potrebne kopije podataka (npr. za replikaciju)
 - ostale pomoćne fizičke strukture
 - potrebna privremena područja
 - repozitorijum meta podataka (MDR)
 - rečnik podataka SUBP
 - arhiviranje podataka



- **Dimenzionisanje BP**

- procena potrebnog kapaciteta BP DW sistema
 - procena prosečne i maksimalne veličine torke u tabeli činjenica i u tabelama dimenzija
 - procena broja torke dimenzija, saglasno zadatoj granularnosti
 - maksimalistička procena potrebnog kapaciteta za podatke
 - proizvod procenjenih brojeva torke svih dimenzija, pomnožen s procenjenom veličinom torke u tabeli činjenica
 - redukovanje maksimalističke procene potrebnog kapaciteta
 - saglasno proceni gustine broja torke činjenica, s obzirom na sve moguće kombinacije torke dimenzija
 - procena dinamike zauzeća potrebnog kapaciteta
 - procena inicijalno potrebnog kapaciteta
 - procena brzine povećavanja obima podataka i maksimalno potrebnog kapaciteta



- **Dimenzionisanje BP**

- validacija i testiranje procene potrebnog kapaciteta
 - ekstrahovanje uzoraka podataka iz projektovanih izvora
 - izbor statistički relevantnih uzoraka
 - učitavanje uzoraka podataka u DW BP
 - formiranje svih potrebnih struktura podataka nad učitanim uzorkom
 - indeksa, materijalizovanih pogleda, ostalih pomoćnih struktura
 - poređenje stvarnih vrednosti i procenjenih vrednosti koje se odnose na kapacitet BP



- **Specifikacija vrsta i organizacije datoteka SUBP**

- specifikacija organizacije datoteka podataka
- specifikacija organizacije datoteka rečnika podataka
- specifikacija organizacije log datoteka
- specifikacija organizacije sistemskih i konfiguracionih datoteka
- uključuje
 - specifikaciju inicijalne veličine i načina širenja – alokacije potrebnog prostora
 - specifikaciju disk jedinica na koje se smeštaju
 - informacije o multiplikaciji sadržaja datoteke u više OS fajlova



- **Specifikacija fizičkog prostora BP**
 - specifikacija organizacije fizičkog prostora BP
 - s obzirom na raspoložive koncepte koje nudi SUBP
 - specifikacija načina raspoređivanja podataka iz BP i rečnika podataka po datotekama
 - specifikacija načina alokacije i dealokacije raspoloživog prostora
 - pri inicijalnom punjenju podataka i
 - pri kasnijim ažuriranjima podataka



- **Specifikacija fizičkog prostora BP**

- primer: SUBP Oracle

- podržani koncepti

- DB blok

- osnovna jedinica upravljanja fizičkim strukturama podataka

- extent

- kontinualni niz alociranih blokova

- segment

- za podatke, indeksni, LOB, rollback segment

- tablespace

- sistemski, za osnovne i pomoćne podatke, privremeni, za rollback segmente



- **Specifikacija fizičkog prostora BP**

- primer: SUBP Oracle

- specifikacije

- kapacitet bloka BP

- fizički parametri tablespace-a

- vrsta, svrha i mogući načini upotrebe (RO, RW, Offline/Online)

- inicijalna i maksimalna veličina

- način širenja - alokacije prostora u jedinicama extent-ova

- preslikavanje u fajlove operativnog sistema

- fizički parametri tabele, indeksa i materijalizovanog pogleda

- inicijalna i maksimalna veličina

- preslikavanje u tablespace-ove

- način širenja u prostoru pridruženih tablespace-ova

- način alokacije prostora u dodeljenim blokovima

- fizička organizacija podataka



- **Specifikacija fizičkog prostora BP**

- primer: SUBP Oracle
- specifikacija parametara tablespace-a

```
CREATE SMALLFILE TABLESPACE TBS_DATA
```

```
DATAFILE
```

```
'C:\ORACLE\PRODUCT\10.2.0\ORADATA\ORC0\TBS_DATA'
```

```
SIZE 100M
```

```
AUTOEXTEND ON NEXT 10M MAXSIZE 200M
```

```
LOGGING
```

```
EXTENT MANAGEMENT LOCAL
```

```
SEGMENT SPACE MANAGEMENT AUTO
```




- **Specifikacija fizičkog prostora BP**

- primer: SUBP Oracle
- specifikacija parametara tabele

```
CREATE TABLE SCHEMA.TAB_DATAFACTS  
(A NUMBER(5) DEFAULT 0 NOT NULL,  
B VARCHAR2(25) NOT NULL,  
PRIMARY KEY (A) VALIDATE  
)  
TABLESPACE USER PCTFREE 20 PCTUSED 60  
STORAGE  
( INITIAL 10M  
  NEXT 1M  
  MINEXTENTS 5 PCTINCREASE 10)
```



- **Specifikacija radnog prostora OM za SUBP**
 - specifikacija potrebnog broja bafera, odnosno potrebnog kapaciteta u radnom prostoru
 - za osnovne i pomoćne podatke iz BP
 - za log podatke
 - za podatke iz rečnika podataka
 - za programske kodove
 - naredbi ili procedura koje treba parsirati i izvršavati
 - specifikacija načina korišćenja radnog prostora
 - broja i vrsta "run-time" procesa koji opslužuju SUBP
 - frekvencije "pražnjenja" bafera, itd.

Sadržaj

- Projektovanje DW sistema
- Tehnike indeksiranja
- Jednostavni bitmap indeks
- Bit-slice indeks
- Kodirani bitmap indeks
- Projekcioni indeks
- Komparacija i upotreba bitmap indeksa
- Spojni indeks
- Fizička organizacija i dimenzionisanje BP
- Fizičko partitioniranje BP

Fizičko particioniranje BP

- Motivacija
 - problem performansi kod vrlo velikih BP (VLDB)
 - neprihvatljivo duga vremena za upite koji zahtevaju pristup svim ili skoro svim torkama velike tabele
 - karakteristično za tabele činjenica u DW BP
 - takođe, problem s performansama pri drugim operacijama nad velikim tabelama
 - učitavanja velikih količina podataka
 - ažuriranja podataka
 - transporta celokupnog sadržaja
 - arhiviranja podataka
 - restauracije i oporavka sadržaja

Fizičko particioniranje BP

- Ideja
 - particioniranje sadržaja tabele na disjunktne delove
 - po kriterijumu koji se vezuje za attribute sa statičkim vrednostima
 - smeštanje particija u različite tablespace-ove (Oracle)
 - fizički locirane u različite datoteke
 - smeštene na različitim disk jedinicama računara
 - sa mogućnošću podešavanja parametara fizičke organizacije za svaki tablespace, tj. particiju pojedinačno
 - sa mogućnošću particioniranja indeksa i indeksiranja particioniranih torki tabele
 - uz očuvanje logičke jedinstvenosti tabele
 - transakcioni programi "vide" tabelu kao jedinstvenu strukturu
 - ne vode računa o fizičkom particionisanju njenog sadržaja

Fizičko particioniranje BP

- Ciljevi
 - poboljšati performanse upita i drugih operacija nad podacima
 - tako što će operacije, u velikom broju slučajeva, biti "lokalizovane" samo na jednu particiju, ili bar manji broj particija
 - čime se smanjuje obim podataka koji je predmet obrade u jednoj operaciji
 - omogućiti paralelizaciju izvođenja operacija
 - tako što se operacije, lokalizovane na pojedinačne particije, mogu izvršavati paralelno
 - svaka particija je smeštena na posebnu disk jedinicu, tako da se i različite disk jedinice koriste u paralelnom režimu rada
 - » a ne samo različite CPU, u višeprosesorskom sistemu

Fizičko particioniranje BP

- Vrste particioniranja tabela
 - vertikalno
 - horizontalno
 - intervalno (range) particioniranje
 - rasuto (hash) particioniranje
 - nabrojano (list) particioniranje
 - kombinovano particioniranje
 - intervalno-rasuto
 - intervalno-nabrojano

Fizičko particioniranje BP

- **Vertikalno particioniranje**

- particioniranje po atributima (kolonama) tabele

- svaka particija obuhvata primarni ključ i izabrane attribute tabele
- svaki atribut tabele izvan primarnog ključa pripada tačno jednoj particiji
- realizuje se kreiranjem posebnih tabela za svaku particiju

- motivacija

- atributi tabele se grupišu na one čije se vrednosti
 - retko menjaju, ili se uopšte ne menjaju i uniformne su veličine
 - često se menjaju, neuniformne su veličine

- u praksi se vrlo retko koristi

- upiti nad različitim particijama zahtevaju spajanje torki iz različitih tabela

Fizičko particioniranje BP

- **Horizontalno particioniranje**
 - particioniranje po torkama tabele, saglasno zadatim kriterijumima
 - **intervalno (range) particioniranje**
 - po opsezima vrednosti zadatih atributa
 - **rasuto (hash) particioniranje**
 - primenom hash transformacije vrednosti zadatih atributa
 - **nabrojano (list) particioniranje**
 - po zadatim listama vrednosti datog atributa
 - **kombinovano particioniranje**
 - intervalno-rasuto
 - » intervalno glavno particioniranje i hash potparticioniranje
 - intervalno-nabrojano
 - » intervalno glavno particioniranje i nabrojano potparticioniranje

Fizičko particioniranje BP

- **Intervalno particioniranje**

- primer: SUBP Oracle

```
CREATE TABLE TAB_DATAFACTS
```

```
(A NUMBER(5),
```

```
  B VARCHAR2(30)
```

```
)
```

```
NOLOGGING
```

```
PARTITION BY RANGE (A)
```

```
(PARTITION P1 VALUES LESS THAN (10001)
```

```
TABLESPACE TBS1,
```

```
PARTITION P2 VALUES LESS THAN (20001)
```

```
TABLESPACE TBS2,
```

```
PARTITION P3 VALUES LESS THAN (MAXVALUE)
```

```
TABLESPACE TBS3)
```

Fizičko particioniranje BP

- **Rasuto particioniranje**

- primer: SUBP Oracle

```
CREATE TABLE TAB_DATAFACTS
```

```
(A NUMBER(5),
```

```
  B VARCHAR2(30)
```

```
)
```

```
NOLOGGING
```

```
PARTITION BY HASH (A)
```

```
(PARTITION P1 TABLESPACE TBS1,
```

```
  PARTITION P2 TABLESPACE TBS2,
```

```
  PARTITION P3 TABLESPACE TBS3,
```

```
  PARTITION P4 TABLESPACE TBS4
```

```
)
```

Fizičko particioniranje BP

- **Nabrojano particioniranje**

- primer: SUBP Oracle

```
CREATE TABLE TAB_DATAFACTS
```

```
  (A CHAR(2),
```

```
   B VARCHAR2(30)
```

```
)
```

```
NOLOGGING
```

```
PARTITION BY LIST (A)
```

```
  (PARTITION P1 VALUES ('BG', 'NS', 'KG', 'NI', 'PR')
```

```
                                     TABLESPACE TBS1,
```

```
  PARTITION P2 VALUES ('SU', 'ZR', 'KV', 'LE', 'UE')
```

```
                                     TABLESPACE TBS2,
```

```
  PARTITION P3 VALUES (DEFAULT) TABLESPACE TBS3
```

```
)
```

Fizičko particioniranje BP

- Particioniranje indeksa
 - moguće kreiranje particioniranog indeksa nad
 - neparticioniranom tabelom
 - particioniranom tabelom
 - moguće kreiranje particioniranog indeksa kao
 - klasičnog B⁺ stabla
 - bitmap indeksa
 - moguće kreiranje neparticioniranog indeksa nad particioniranom tabelom

Fizičko particioniranje BP

- Vrste particioniranja indeksa
 - **globalno particionirani indeks**
 - podvrste
 - **globalno intervalno particionirani indeks**
 - **globalno rasuto particionirani indeks**
 - može se kreirati i nad particioniranom i nad neparticioniranom tabelom
 - šema particioniranja indeksa ne zavisi od šeme particioniranja indeksirane tabele
 - primenljiv samo za klasične B⁺ indekse
 - **lokalno particionirani indeks**
 - može se kreirati isključivo nad particioniranom tabelom
 - šema particioniranja indeksa odgovara šemi particioniranja indeksirane tabele
 - primenljiv i za klasične B⁺ i za bitmap indekse

Fizičko particioniranje BP

- **Lokalno particioniranje indeksa**

- primer: SUBP Oracle

```
CREATE TABLE TAB_DATAFACTS
```

```
(A NUMBER(5),
```

```
  B VARCHAR2(5),
```

```
  C NUMBER(5)
```

```
) NOLOGGING
```

```
PARTITION BY RANGE (C)
```

```
(PARTITION P1 VALUES LESS THAN (10001)
```

```
TABLESPACE TBS1,
```

```
PARTITION P2 VALUES LESS THAN (20001)
```

```
TABLESPACE TBS2,
```

```
PARTITION P3 VALUES LESS THAN (MAXVALUE)
```

```
TABLESPACE TBS3)
```

Fizičko particioniranje BP

- **Lokalno particioniranje indeksa**
 - primer: SUBP Oracle

```
CREATE BITMAP INDEX IND_DATAFACTS  
ON TAB_DATAFACTS (C)  
NOLOGGING  
LOCAL  
(PARTITION P1 TABLESPACE IBS1,  
PARTITION P2 TABLESPACE IBS2,  
PARTITION P3 TABLESPACE IBS3  
)
```


Fizičko particioniranje BP

- **Globalno intervalno particioniranje indeksa**
 - primer: SUBP Oracle

```
CREATE INDEX IND_DATAFACTS  
ON TAB_DATAFACTS (B, C)  
NOLOGGING  
GLOBAL PARTITION BY RANGE (B)  
(PARTITION P1 VALUES LESS THAN ('N')  
TABLESPACE IBS1,  
PARTITION P2 VALUES LESS THAN ('T')  
TABLESPACE IBS2,  
PARTITION P3 VALUES LESS THAN (MAXVALUE)  
TABLESPACE IBS3  
)
```

Sadržaj

- Projektovanje DW sistema
- Tehnike indeksiranja
- Jednostavni bitmap indeks
- Bit-slice indeks
- Kodirani bitmap indeks
- Projekcioni indeks
- Komparacija i upotreba bitmap indeksa
- Spojni indeks
- Fizička organizacija i dimenzionisanje BP
- Fizičko particioniranje BP

Pitanja i komentari



Fizička organizacija BP Data Warehouse sistema

*Modeli fizičke strukture BP
DW sistema*