

Indeksi

Mehanizmi za ubrzavanje operacija

Indeksi

- Indeksi predstavljaju dodatne strukture koje omogućavaju neskevincijalni pristup podacima.
- Jednom fajlu može biti pridruženo više indeksa. Indeks se defininiše nad atributom ili skupom atributa, tzv. indeksnim poljem, koji predstavlja ključ po kojem se vrši pretraga indeksne strukture umesto celih fajlova sa podacima.
- Indeksni fajl se sastoji iz **indeksnih zapisa** (*index entry*) u formi

search-key	pointer
------------	---------

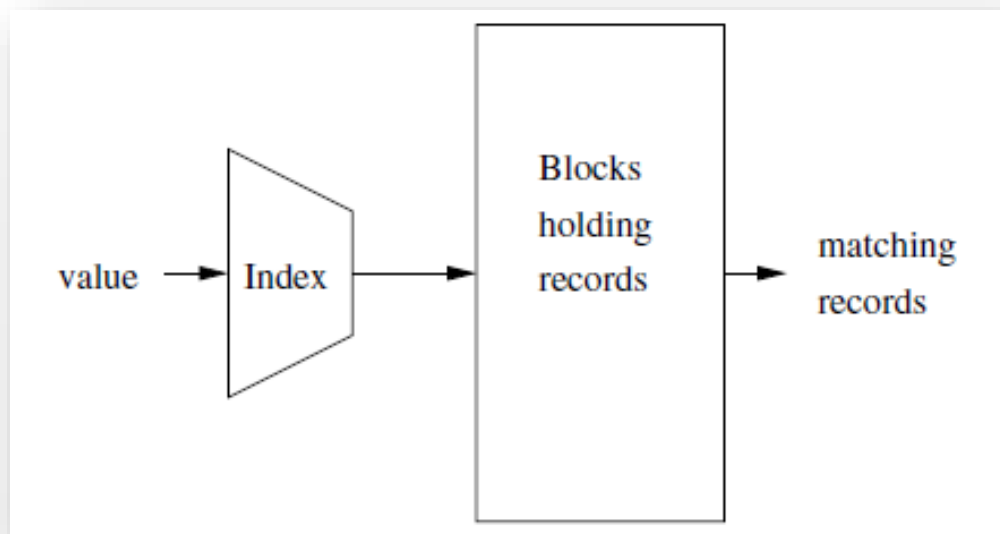
kojima su povezane vrednosti ključa/indeksnog polja sa informacijom o lokaciji slogova podataka (*data entry*) sa navedenom vrednošću ključa.

Informacija o lokaciji sloga – najčešće predstavlja vrstu reference na slog (RID) ili adresa bloka.

- Prednost upotrebe
veća količina potrebnih informacija se učitava u manjem broju blokova -> brža pretraga po indeksnom polju
-

Prednost upotrebe indeksa

- Ključni za efikasnu obradu upita u bazama podataka. Veća količina potrebnih informacija se učitava u manjem broju blokova -> brža pretraga po indeksnom polju
- Bez indeksa, svaki upit bi morao da pročita celokupan sadržaj svake relacije koju koristi; što je za upite koji zahtevaju dohvaćanje samo nekoliko zapisa nepotrebno resursno skupo.



Ključni za efikasnu obradu upita u bazama podataka. Veća količina potrebnih informacija se učitava u manjem broju blokova

Vrste i merenje kvaliteta indeksa

- Dve osnovne vrste indeksa:
 - **Uređeni**. Uređeni prema vrednosti indeksnog polja.
 - **Hash indeksi**. Zasnovani na uniformnoj distribuciji vrednosti u kolekciju korpi (*bucket-a*). Određivanje pripadnosti korpi se vrši na osnovu vrednosti odabrane heš funkcije.
 - Ne postoji univerzalni odabir vrste indeksa. Svaka vrsta može biti manje ili više dobra u zavisnosti od načina upotrebe.
Procena performansi neke indeksne tehnike se vrši analizom:
 - Vrste pristupa/pretrage koje efikasno podržava (jedna vrednost, opseg vrednosti)
 - Vreme dobavljanja traženog/ih podatka
 - Vreme dodavanja – uključuje upis u fajl sa podacima i ažuriranje indeksa
 - Vreme brisanja - uključuje upis u fajl sa podacima i ažuriranje indeksa
 - Prostorno prekoračenje – dodatni prostor koji se rezerviše za sam indeks
-

Podela

- Primarni vs. Sekundarni

Da li uređenje slogova u fajlu sa podacima prati uređenje indeksiranih polja u indeksu?

- Klasterovani vs. Neklasterovani (Grupišući/Negrupišući)

Da li se su podaci u fajlu uređeni prema vrednosti ključa u indeksu ili ne?
(klasterovani indeks nije isto što i klasterovana fajl oranizacija)

- Gusti vs. Retki

Da li se u indeksu nalaze sve vrednosti indeksnog polja koje se pojavljuju i u fajlu sa podacima?

- Jednonivoiski vs. Višenivoiski

Da li se u indeksnoj strukturi definiše 'indeks nad indeksom' ili ne?

Uređeni indeksi

O uređenim indeksima i uređenim fajlovima

Uređeni indeksi

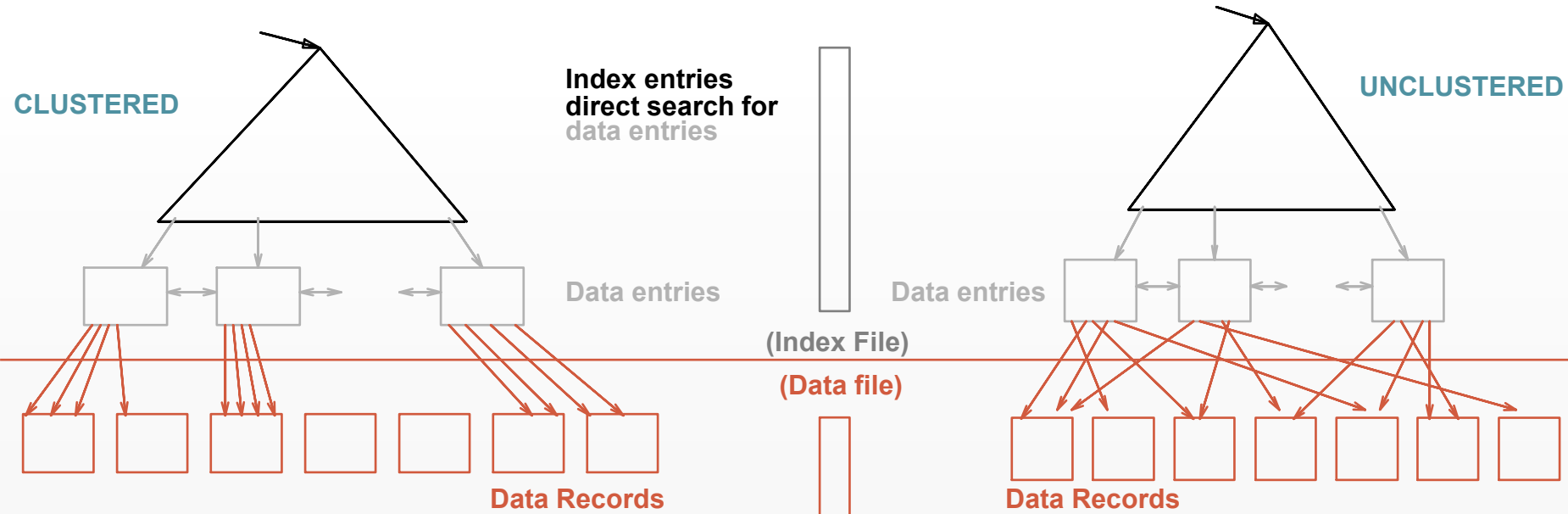
Osnovni koncept i vrste

Uređeni indeksi

- Struktura podatka u kojoj su parovi key, data entry uređeni prema vrednosti ključa.
Key - može biti jedan ili više atributa tabele nad kojima je postavljen indeks.
Data entry – ukazuje na torku u fajlu (tabeli).
 - Struktura se beleži tako da omogućiti brzu pretragu.
 - DBMS ima obavezu da pri izmeni sadržaja tabele sinhronizovano vrši i izmene u pridruženim indeksima.
 - Tokom izvršavanja upita DBMS procenjuje koje indekse je moguće iskoristiti za ubrzavanje izvršavanja, tj. smanjenje broja čitanja/pisanja po disku.
 - Ključni troškovi:
 - Memorijski prostor
 - Održavanje
-

Primarni i sekundarni indeksi

- **Klasterišući indeks**. Indeksno polje odgovara kriterijumu po kojem je fajl sa podacima uređen. Klasterišući indeks se često naziva i **primarnim**. Teorijski gledano primarni indeks ne mora biti definisan nad primarnim ključem relacije, mada najčešće i jeste tako implementirano.
- **Neklasterišući indeks**. Indeks u kojem indeksno polje nameće uređenje koje ne odgovara sekvencijalnom uređenju fajla. Neklasterišući indeks se naziva i **sekundarnim indeksom**.



Indeks-sekvencijalni fajlovi

- Fajl sa sekvencijalnim uređenjem prema ključu nad kojim je postavljen i uređeni indeks se naziva **indeks-sekvencijalnim fajlom**.

10101		10101	Srinivasan	Comp. Sci.	65000	
12121		12121	Wu	Finance	90000	
15151		15151	Mozart	Music	40000	
22222		22222	Einstein	Physics	95000	
32343		32343	El Said	History	60000	
33456		33456	Gold	Physics	87000	
45565		45565	Katz	Comp. Sci.	75000	
58583		58583	Califieri	History	62000	
76543		76543	Singh	Finance	80000	
76766		76766	Crick	Biology	72000	
83821		83821	Brandt	Comp. Sci.	92000	
98345		98345	Kim	Elec. Eng.	80000	

sekvencijalni fajl

10101	Srinivasan	Comp. Sci.	65000	
12121	Wu	Finance	90000	
15151	Mozart	Music	40000	
22222	Einstein	Physics	95000	
32343	El Said	History	60000	
33456	Gold	Physics	87000	
45565	Katz	Comp. Sci.	75000	
58583	Califieri	History	62000	
76543	Singh	Finance	80000	
76766	Crick	Biology	72000	
83821	Brandt	Comp. Sci.	92000	
98345	Kim	Elec. Eng.	80000	

indeks-sekvencijalni fajl

Primarni indeksi

- **Gusti indeks.**

U indeksu postoji slog za svaku vrednost indeksnog polja.

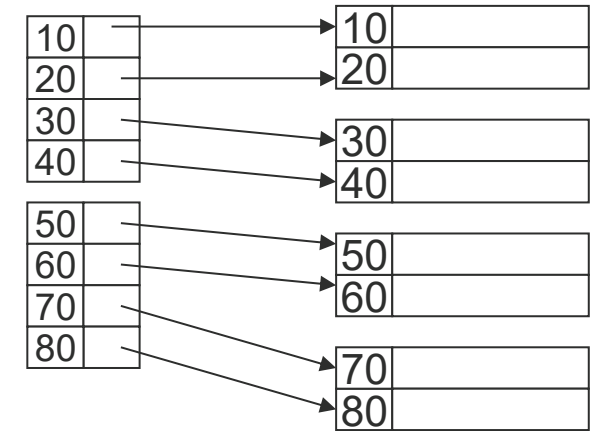
- **Redak indeks.**

U indeksu postoje samo neke vrednosti indeksnog polja.

Obično sadrže jedan ključ-pokazivač par po bloku.

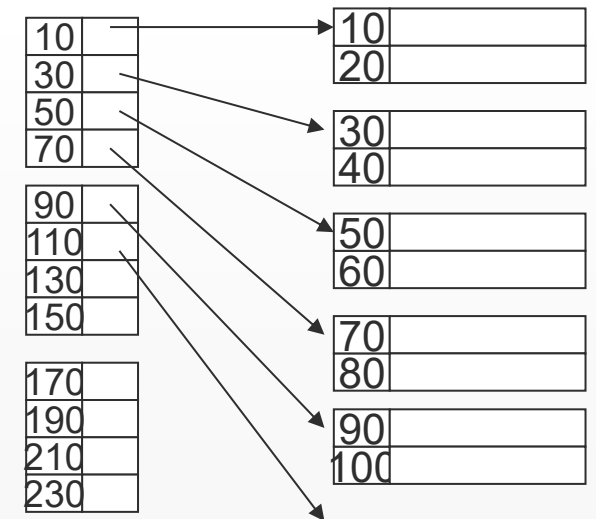
Primenljiv samo u slučaju sekvencijalnog uređenja prema indeksnom polju.

Dense Index



Sparse Index

Sequential File



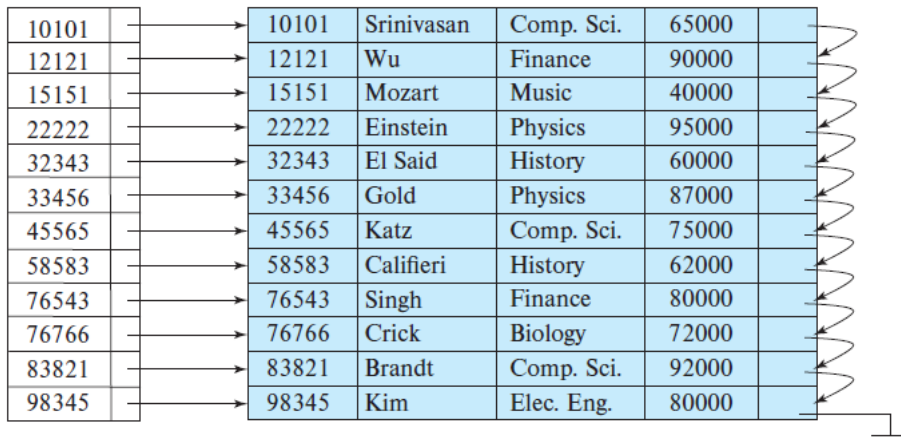
Gusti vs. Retki

- Gusti indeksi.

Može se utvrditi da li neka vrednost indeksnog polja postoji bez učitavanja fajla sa podacima.

Zauzima više prostora.

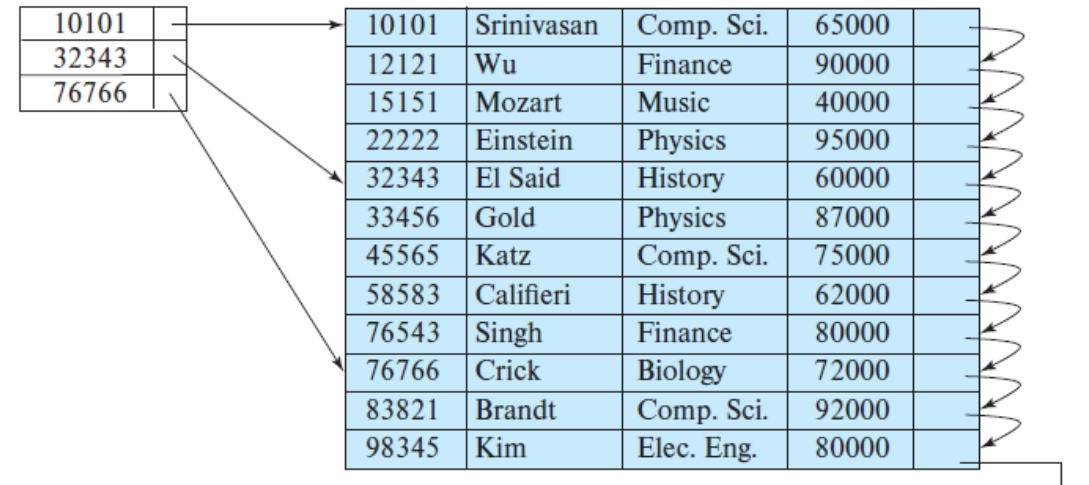
Sporije se pretražuje sam indeks.



- Retki indeksi.

Veći deo indeksa može biti učitán u memoriju.

Lakši za implementaciju dodavanja novih slogova.



Duplikati / gust indeks

10	
10	

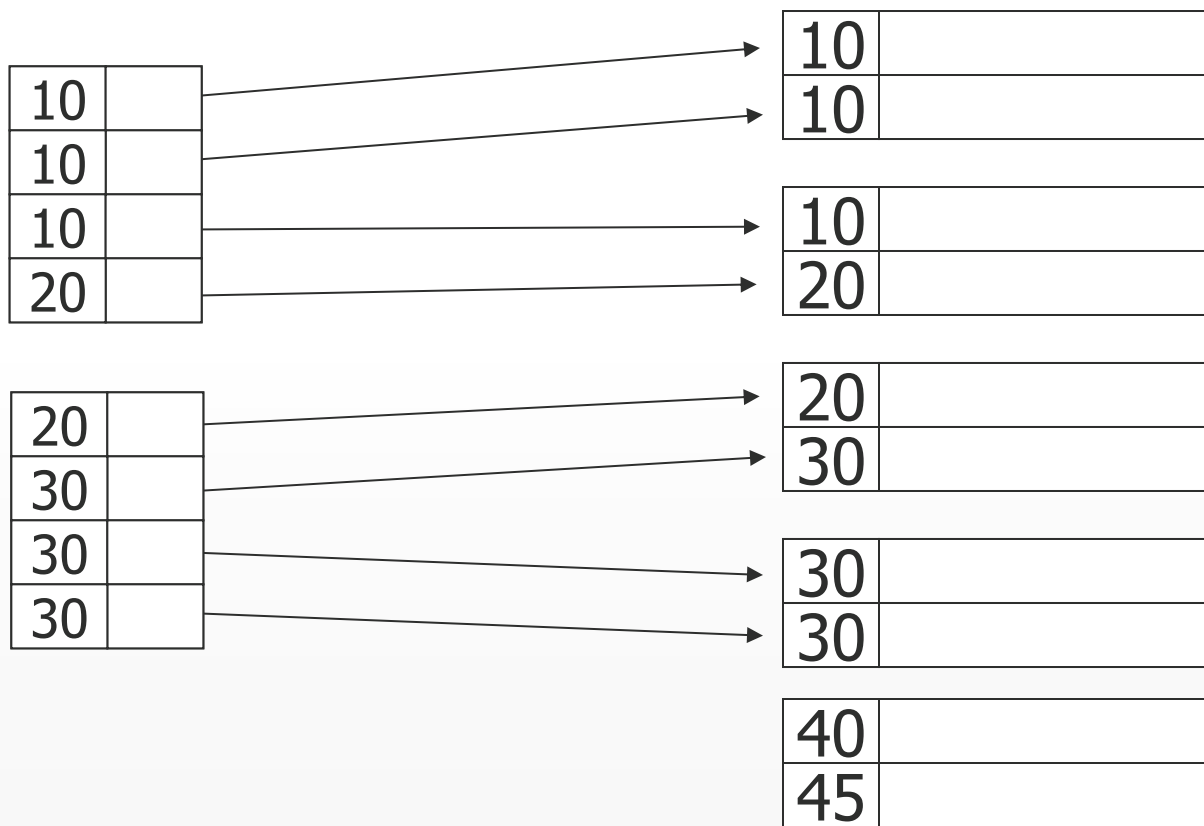
10	
20	

20	
30	

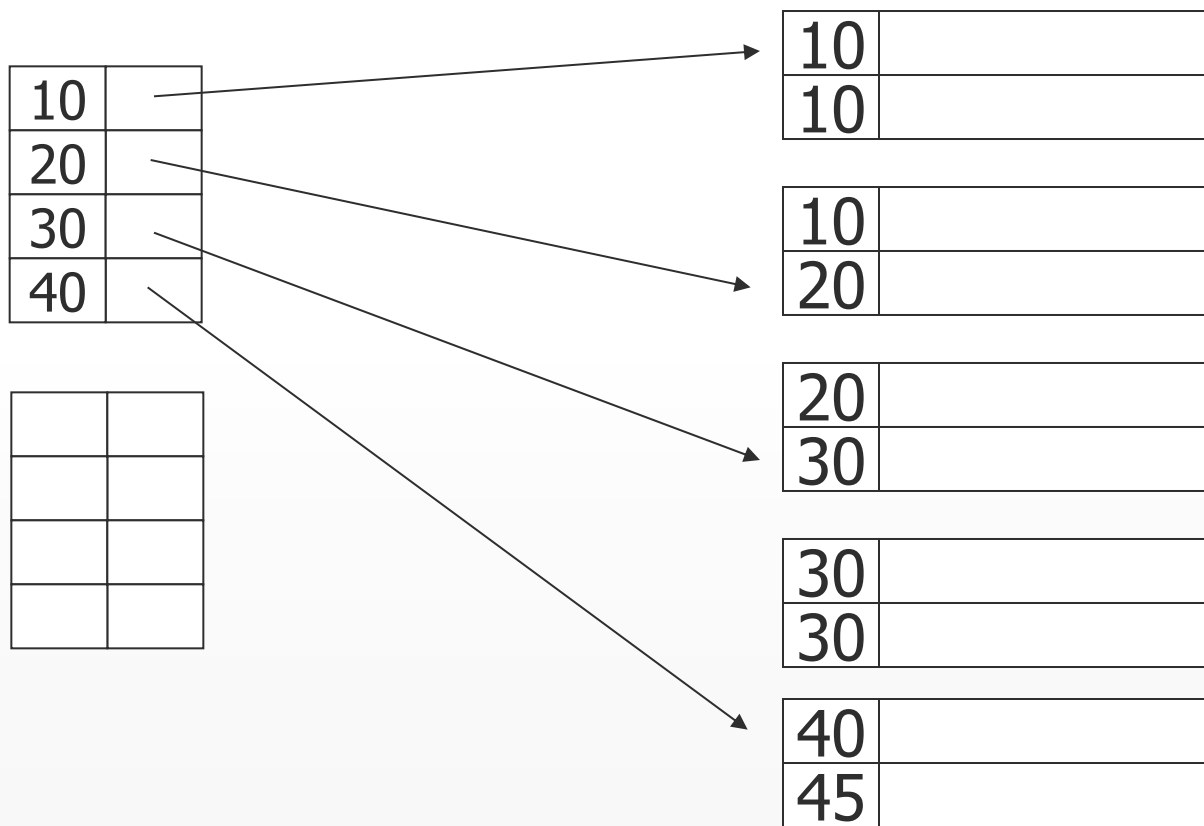
30	
30	

40	
45	

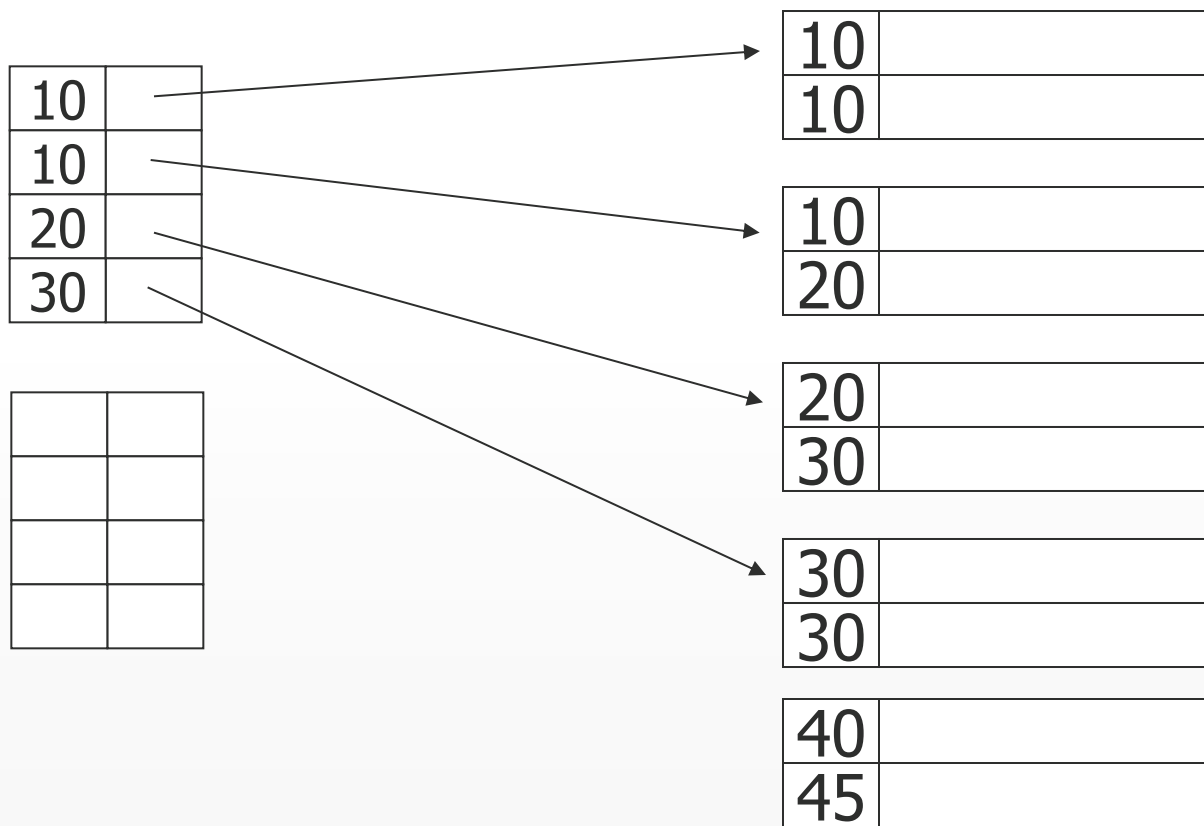
Duplikati / gust indeks – prvi način



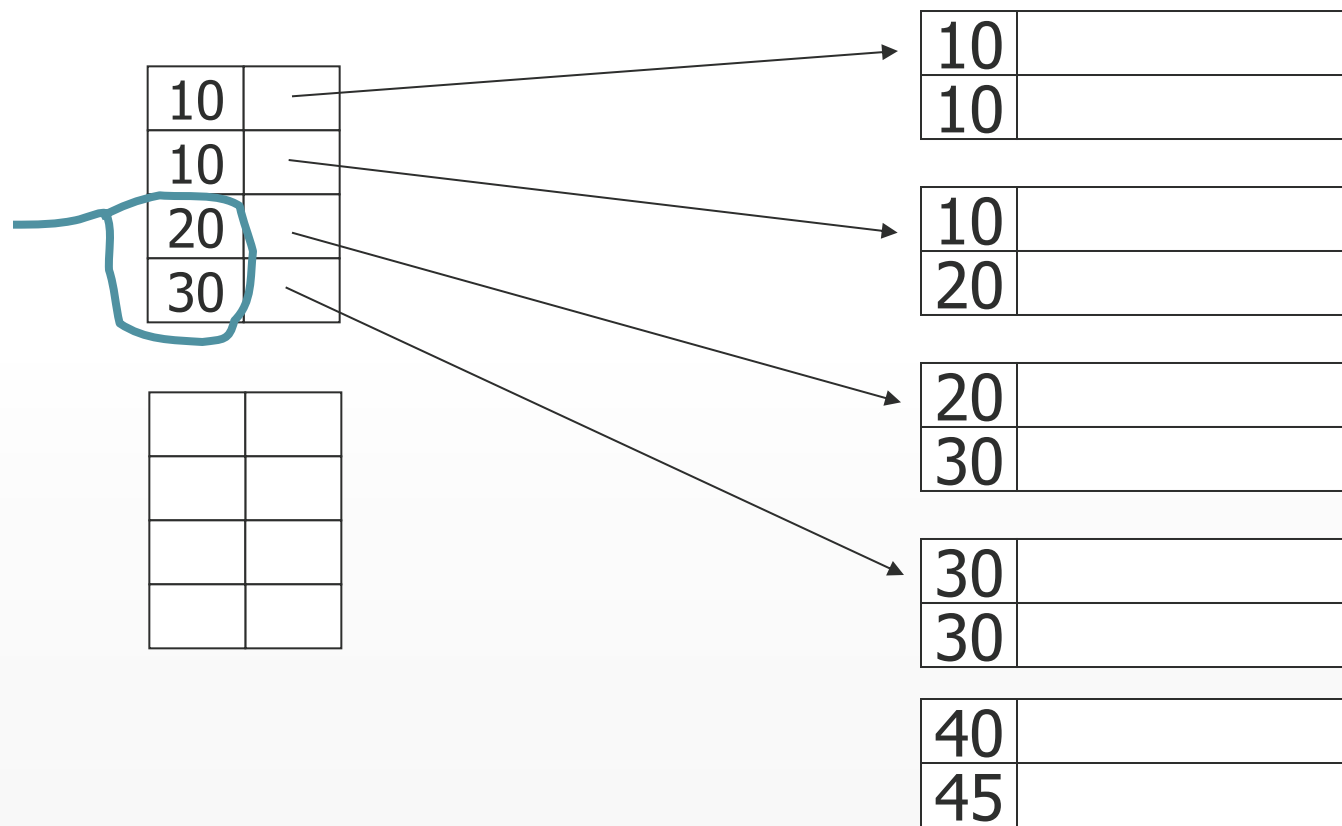
Duplikati / gust indeks – drugi način



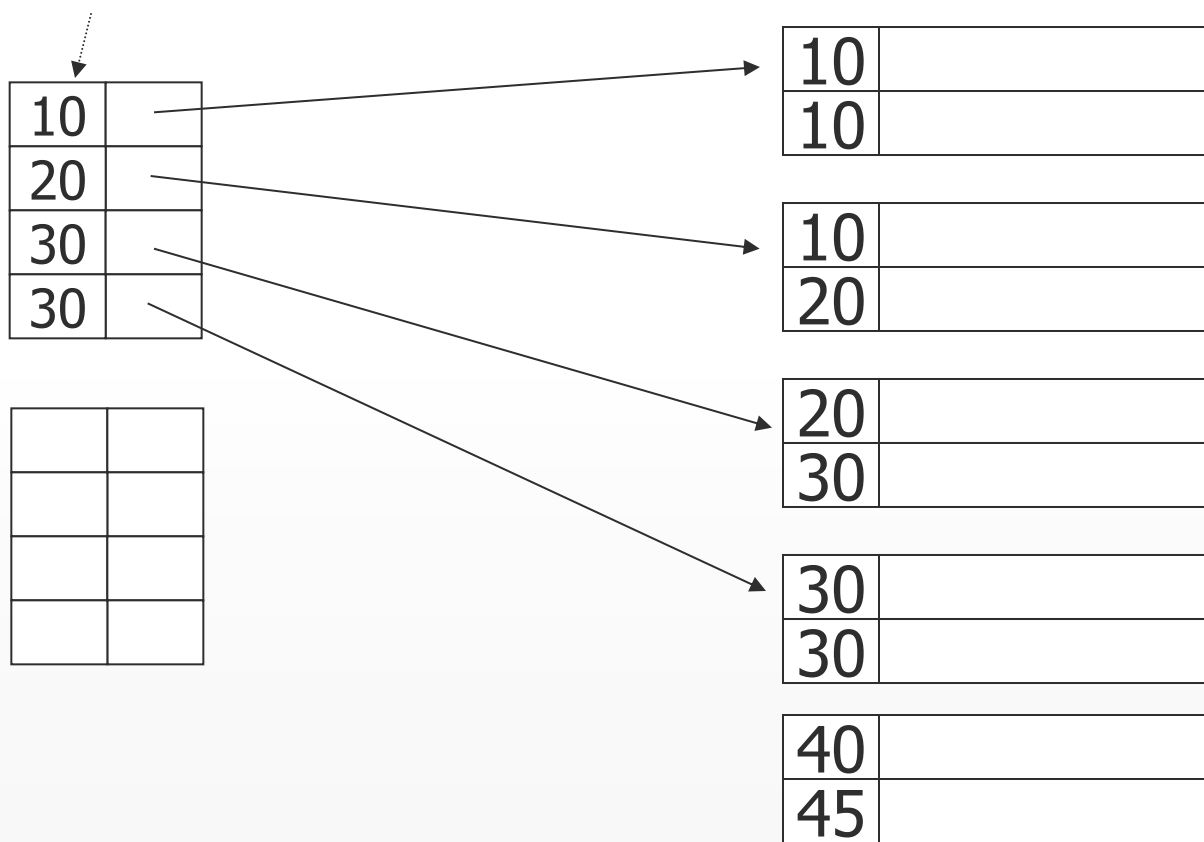
Duplikati / redak



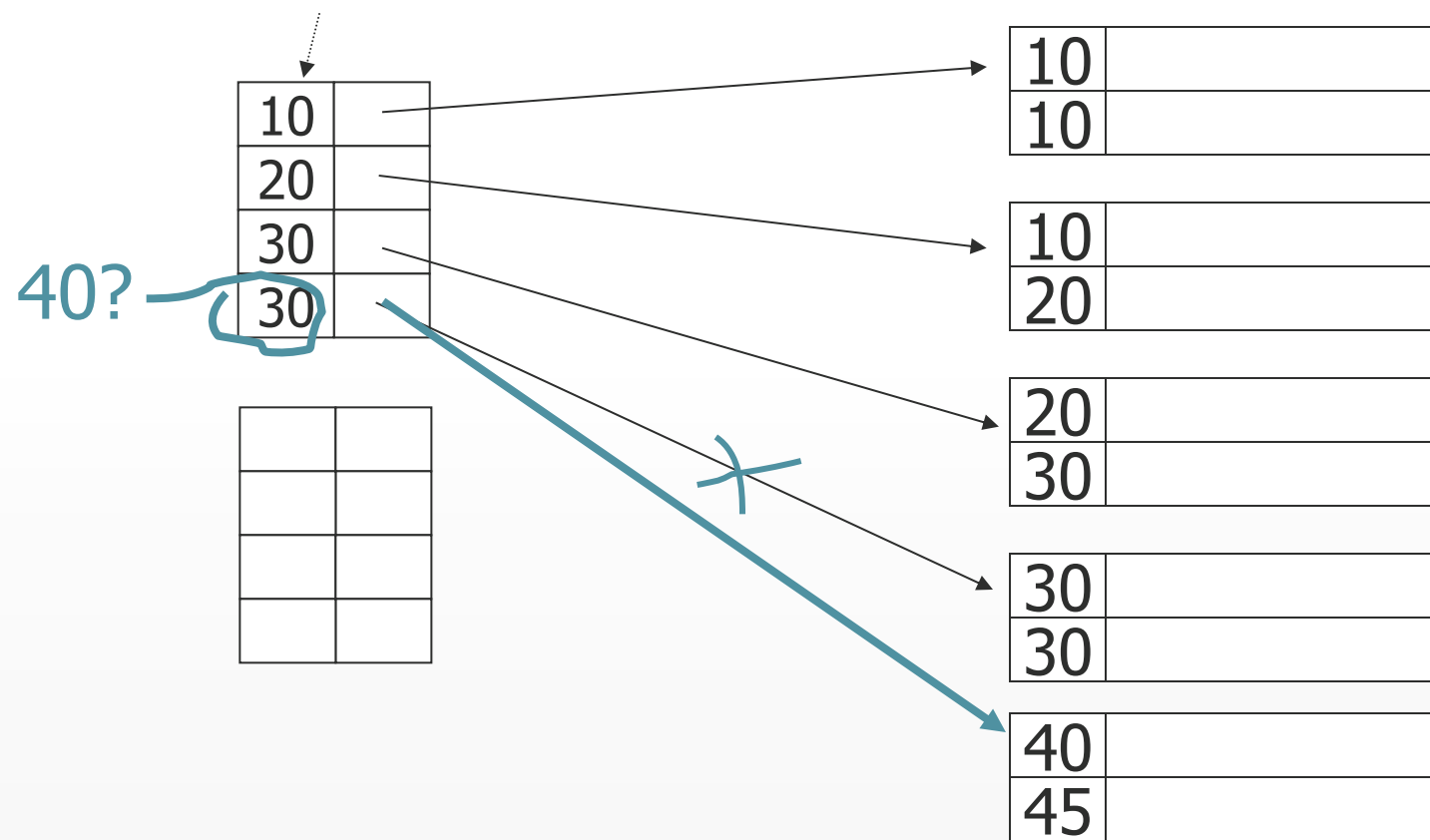
Duplikati / redak



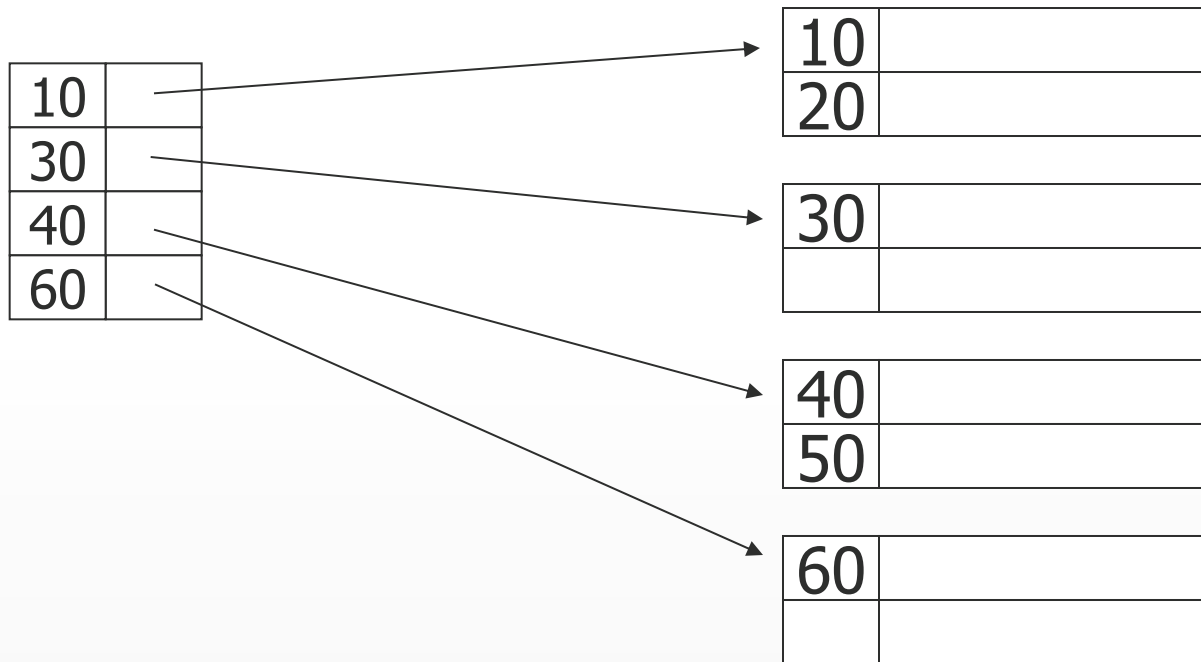
Duplikati / redak



Duplikati / redak

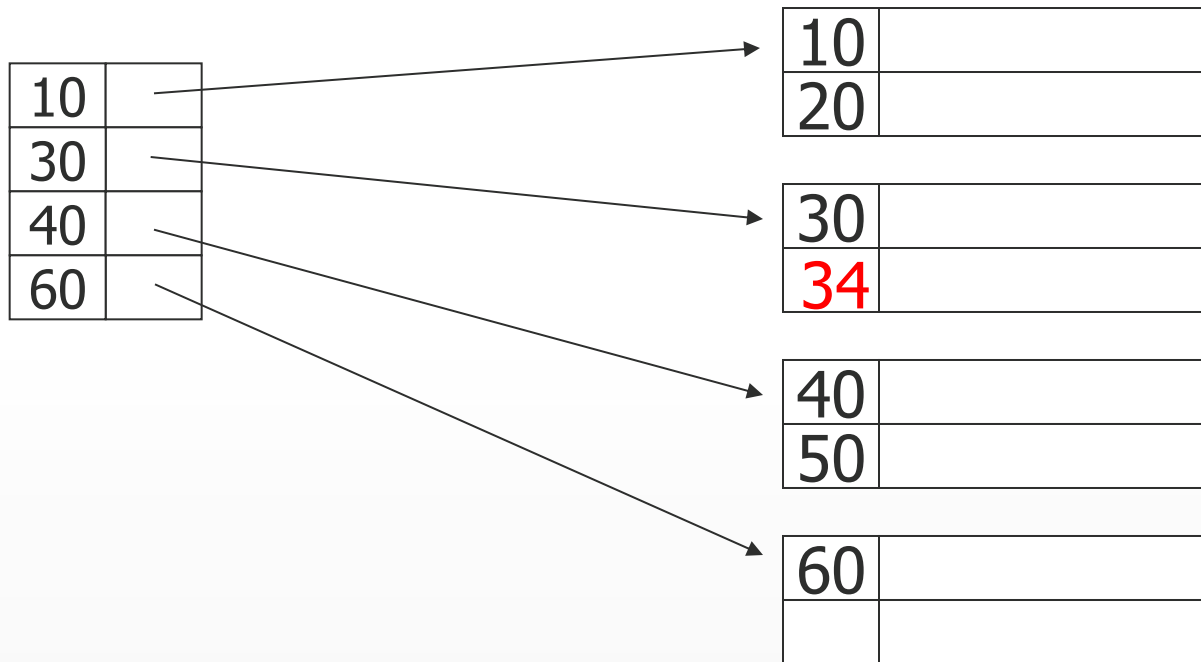


Dodavanje redak indeks



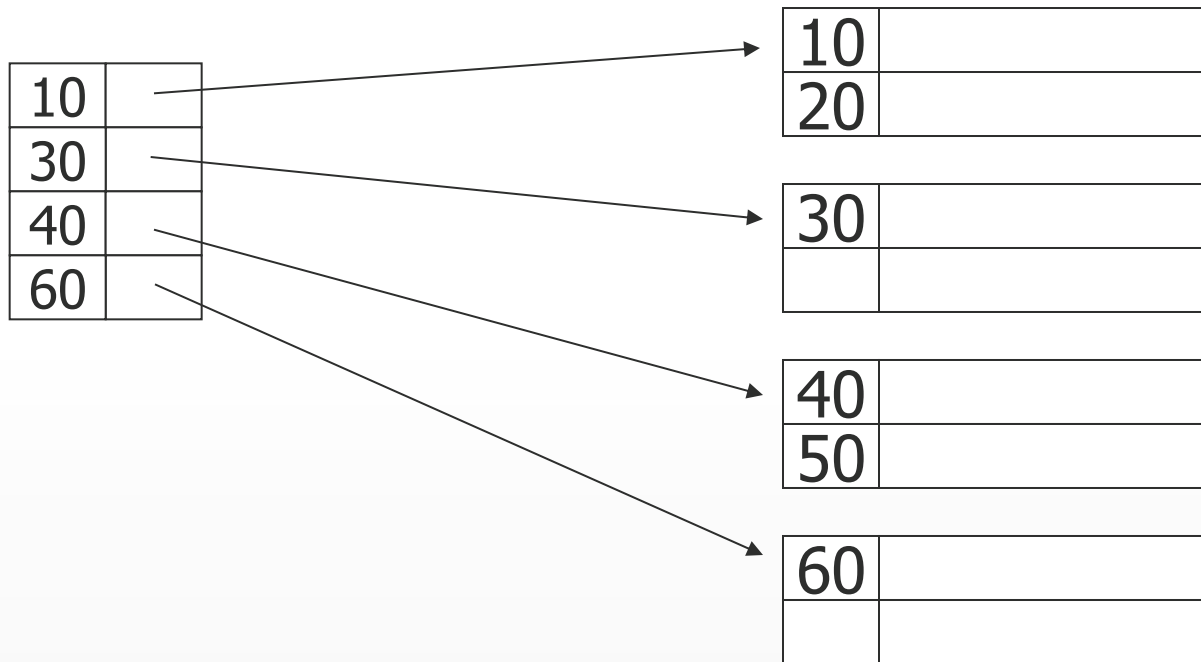
—insert record 34

Dodavanje redak indeks



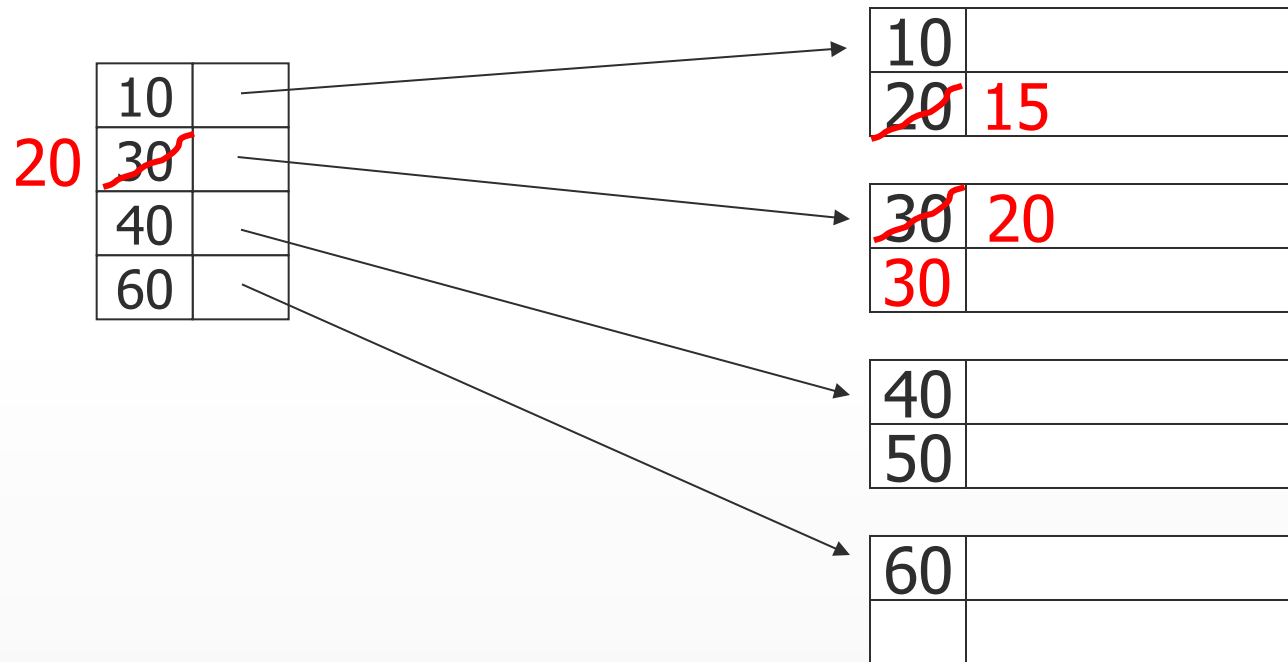
–insert record 34

Dodavanje redak indeks



—insert record 15

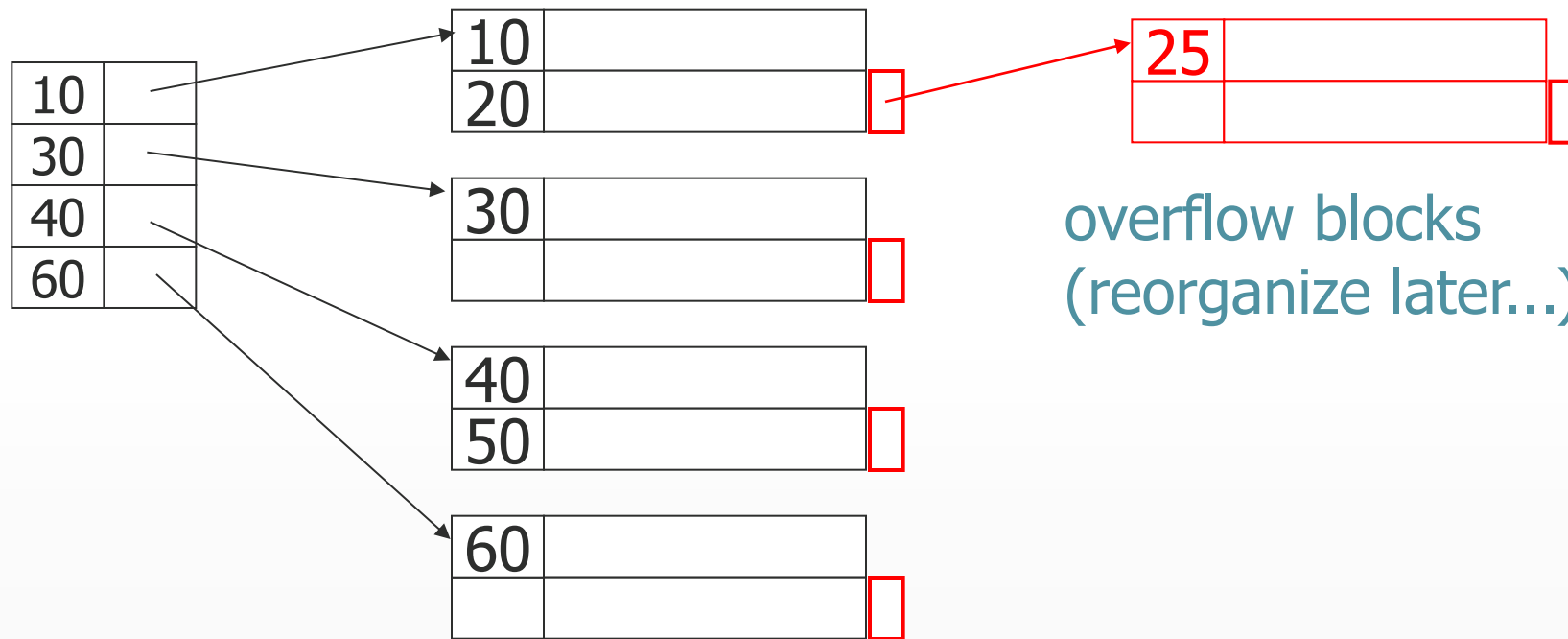
Dodavanje redak indeks



- Illustrated: Immediate reorganization
- Variation:
 - insert new block (chained file)
 - update index

–insert record 15

Dodavanje redak indeks



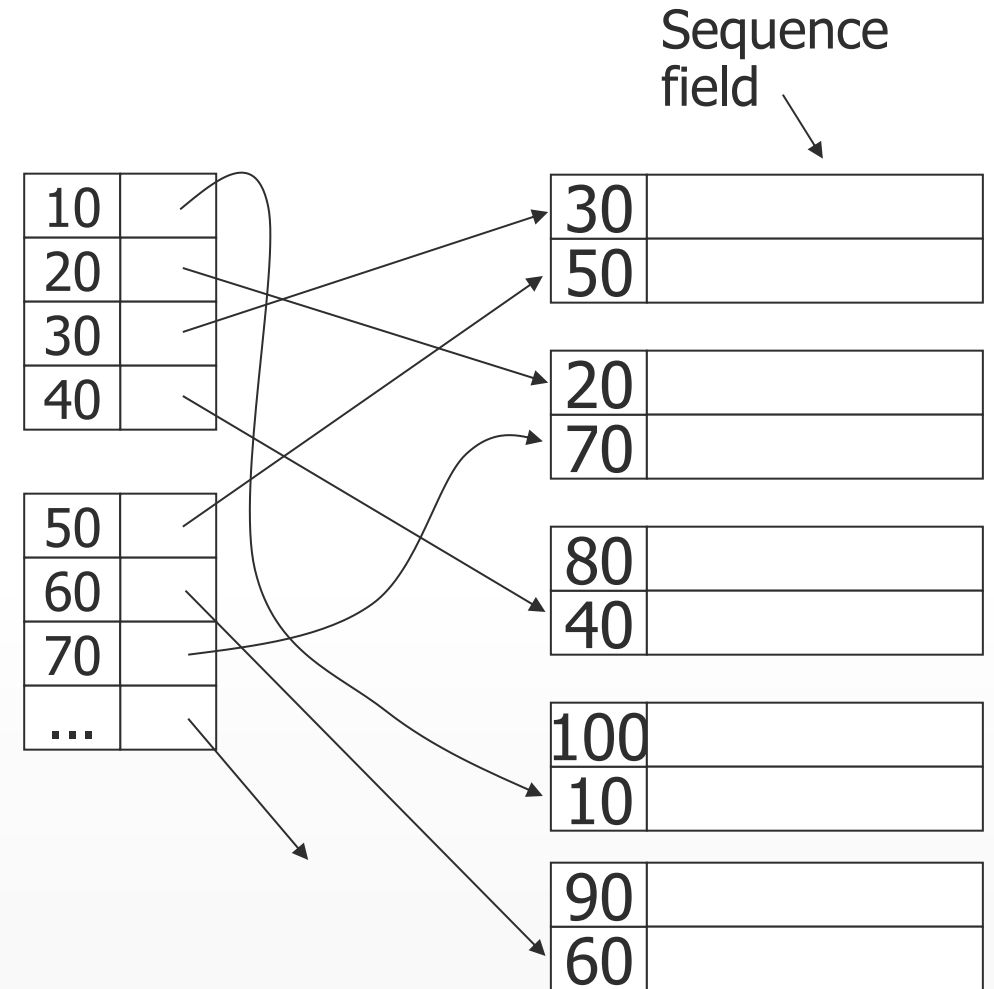
–insert record 25

Dodavanje gust indeks

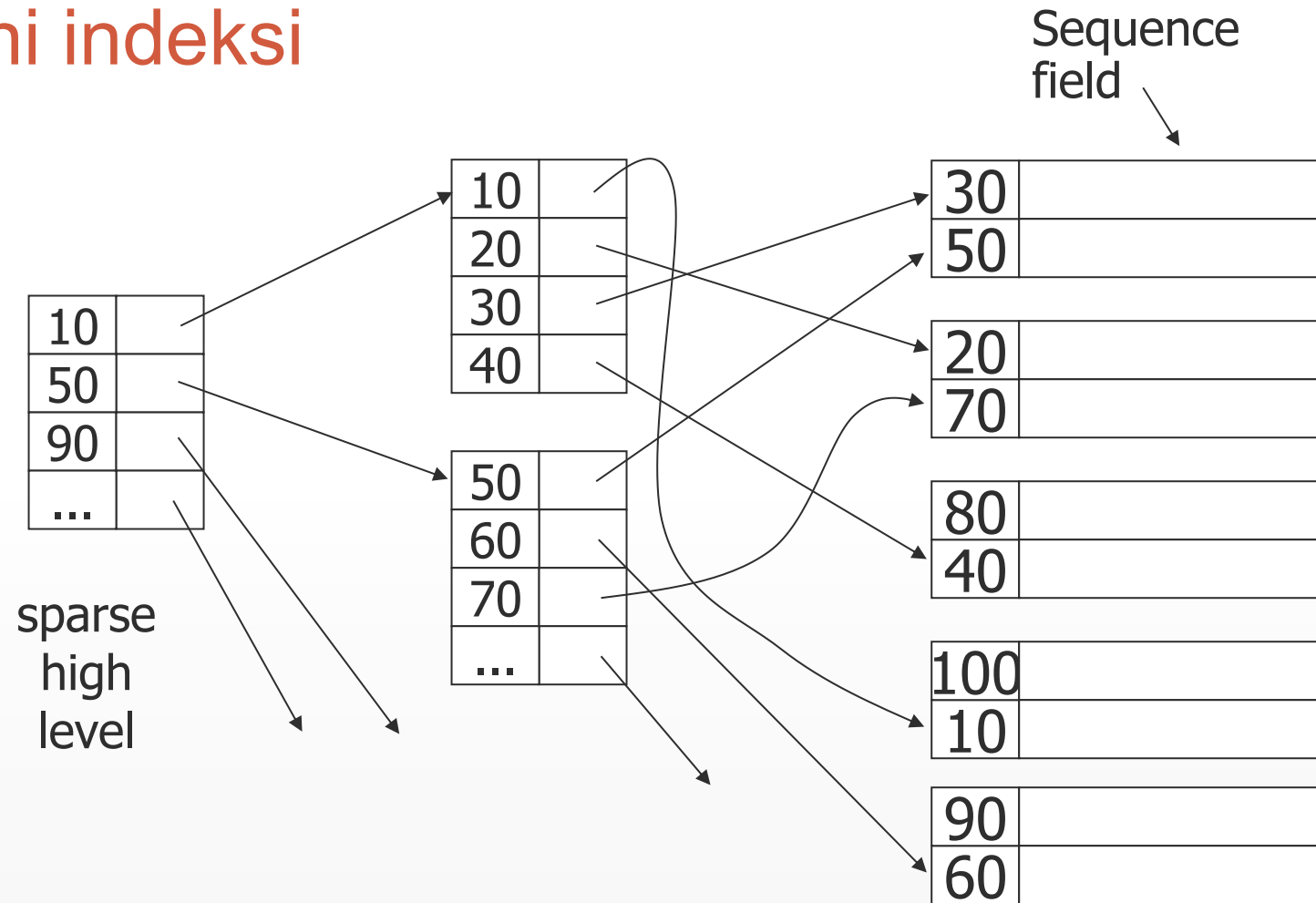
Analogno retkom

Sekundarni indeksi

- Indeksirano polje ne odgovara kriterijumu uređenja fajla sa podacima.
- Indeks mora biti **gust**.
- Pointer u slogovima indeksa mora biti **pointer ka slogu podataka**.



Sekundarni indeks

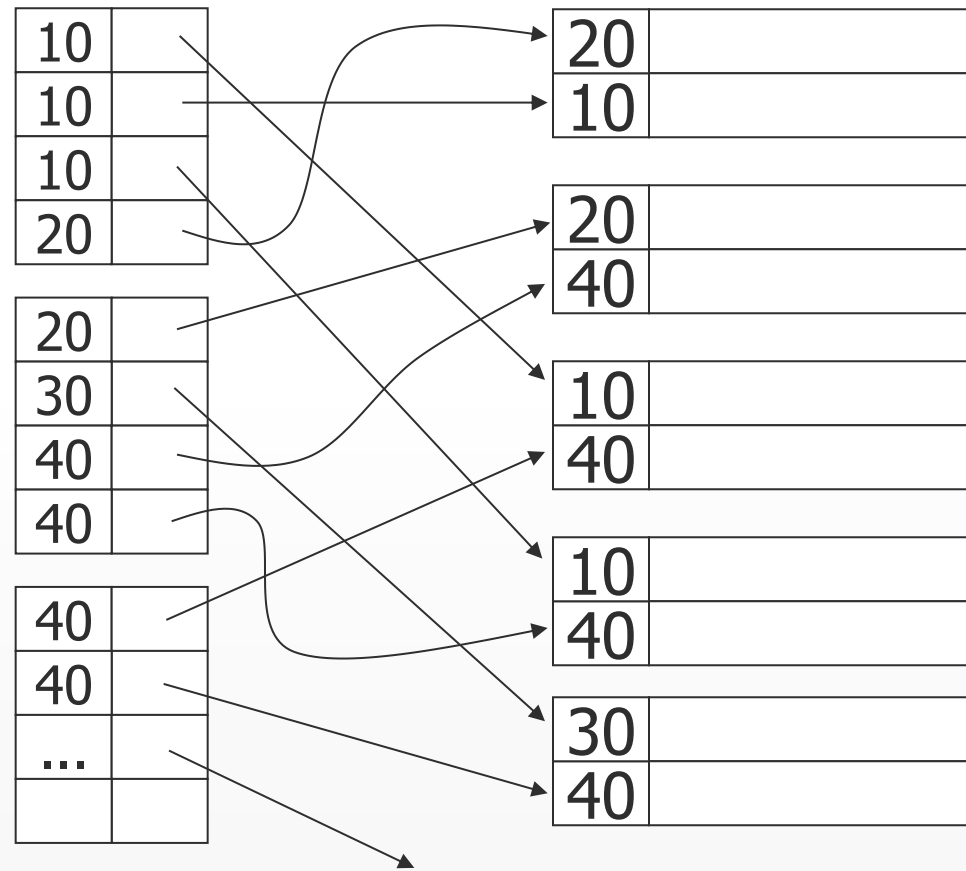


Duplikati i sekundarni indeksi

Problem:

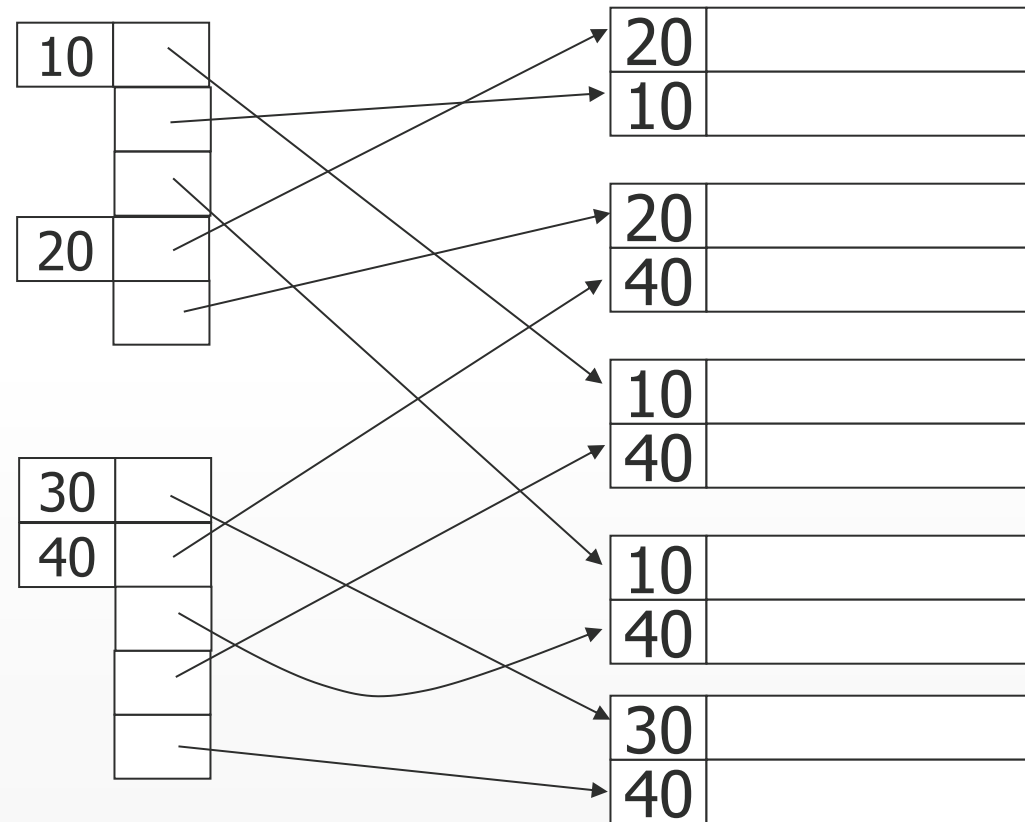
excess overhead!

- disk space
- search time

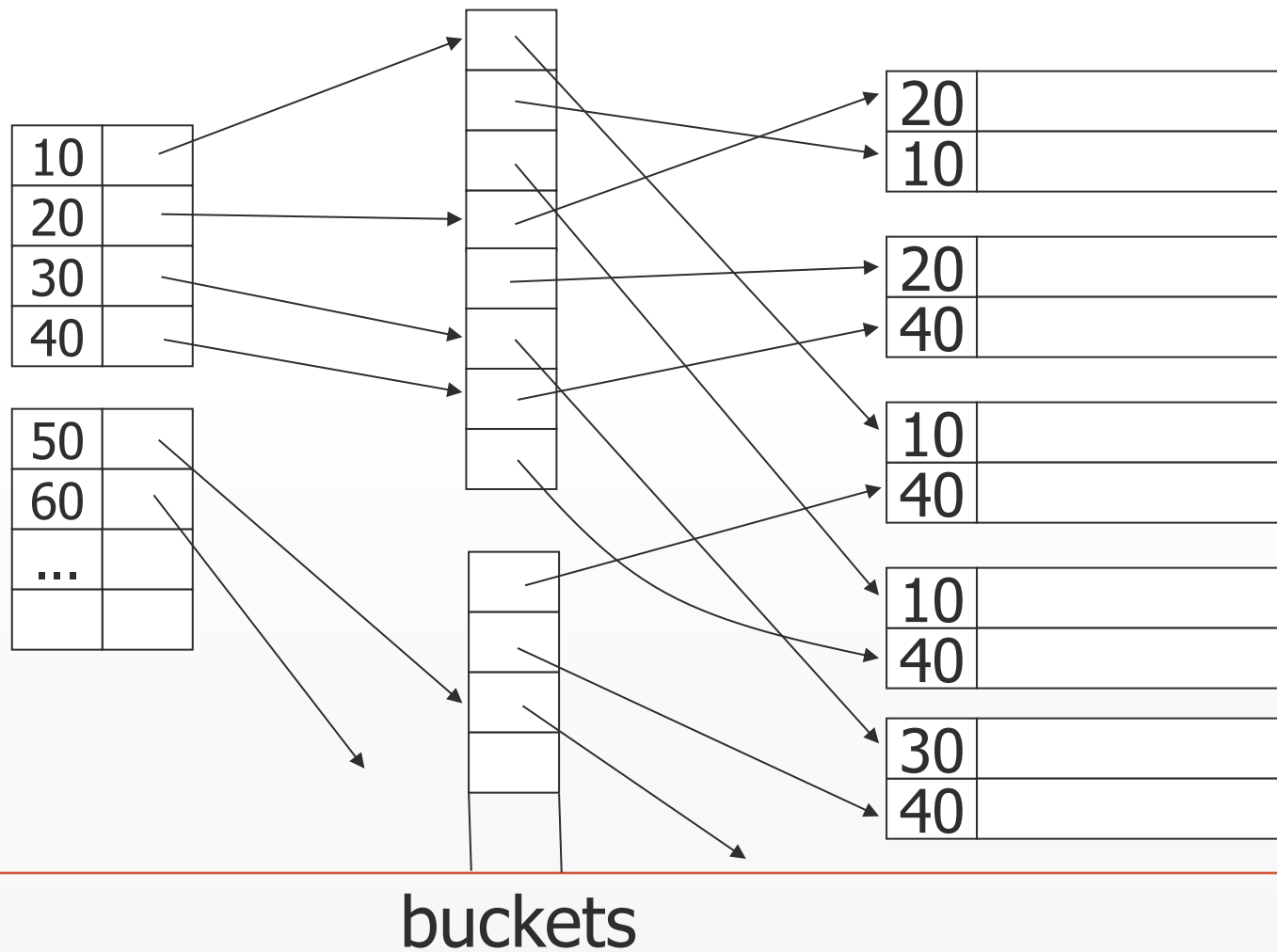


Duplikati i sekundarni indeksi

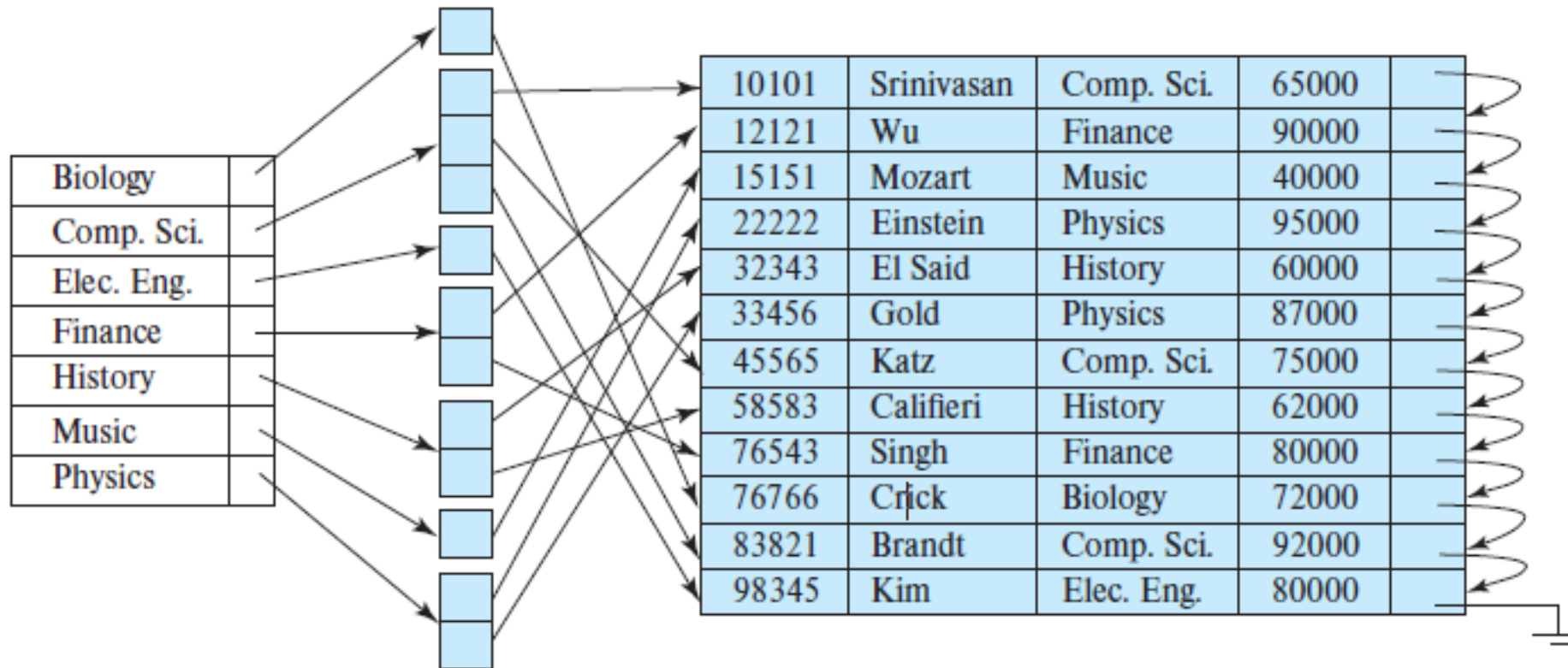
Problem:
variable size
records in
index!



Duplikati i sekundarni indeksi

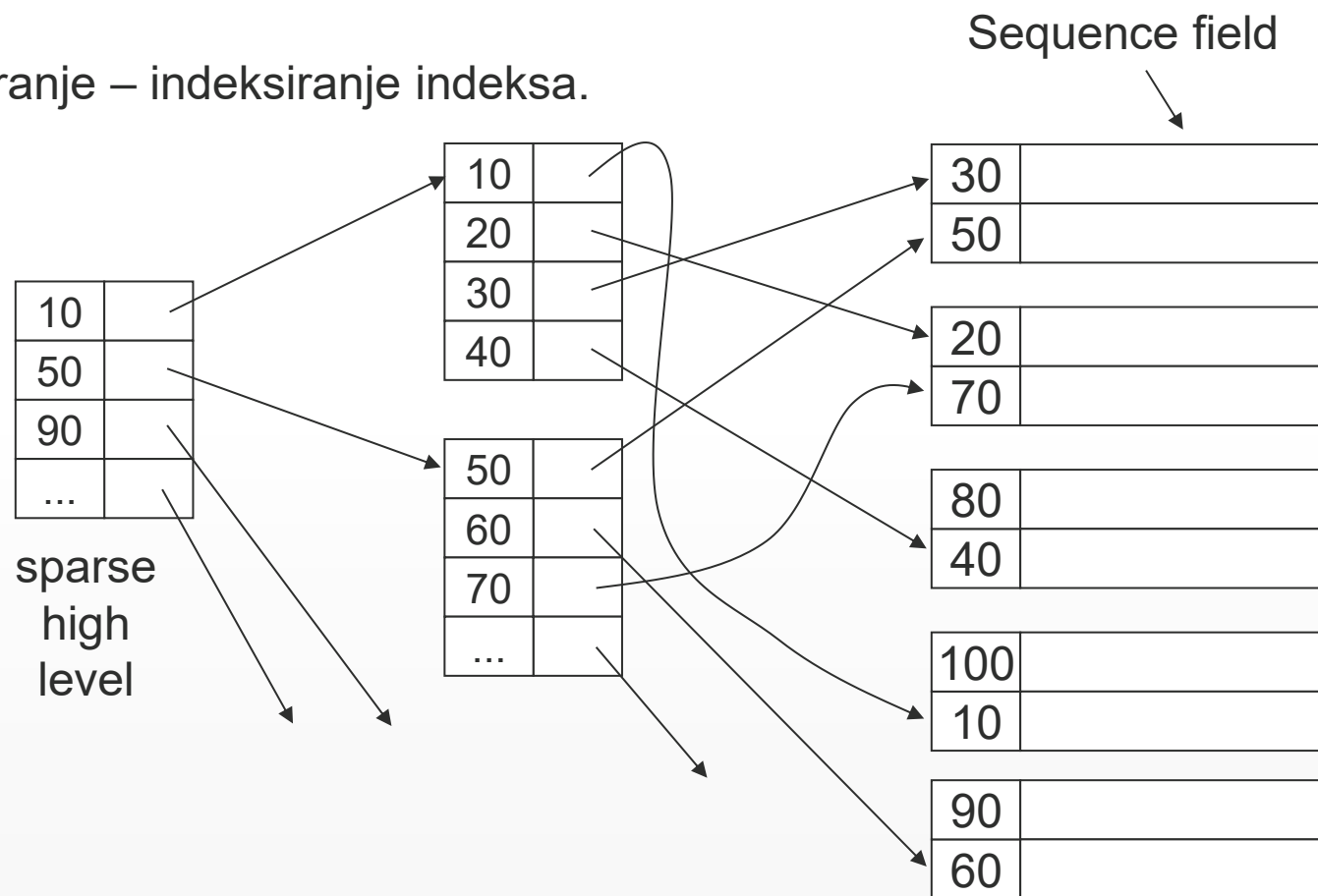


Duplikati i sekundarni indeksi



Višenivojski indeksi

- Višenivojsko, hijerarhijsko indeksiranje – indeksiranje indeksa.



B+ stablo

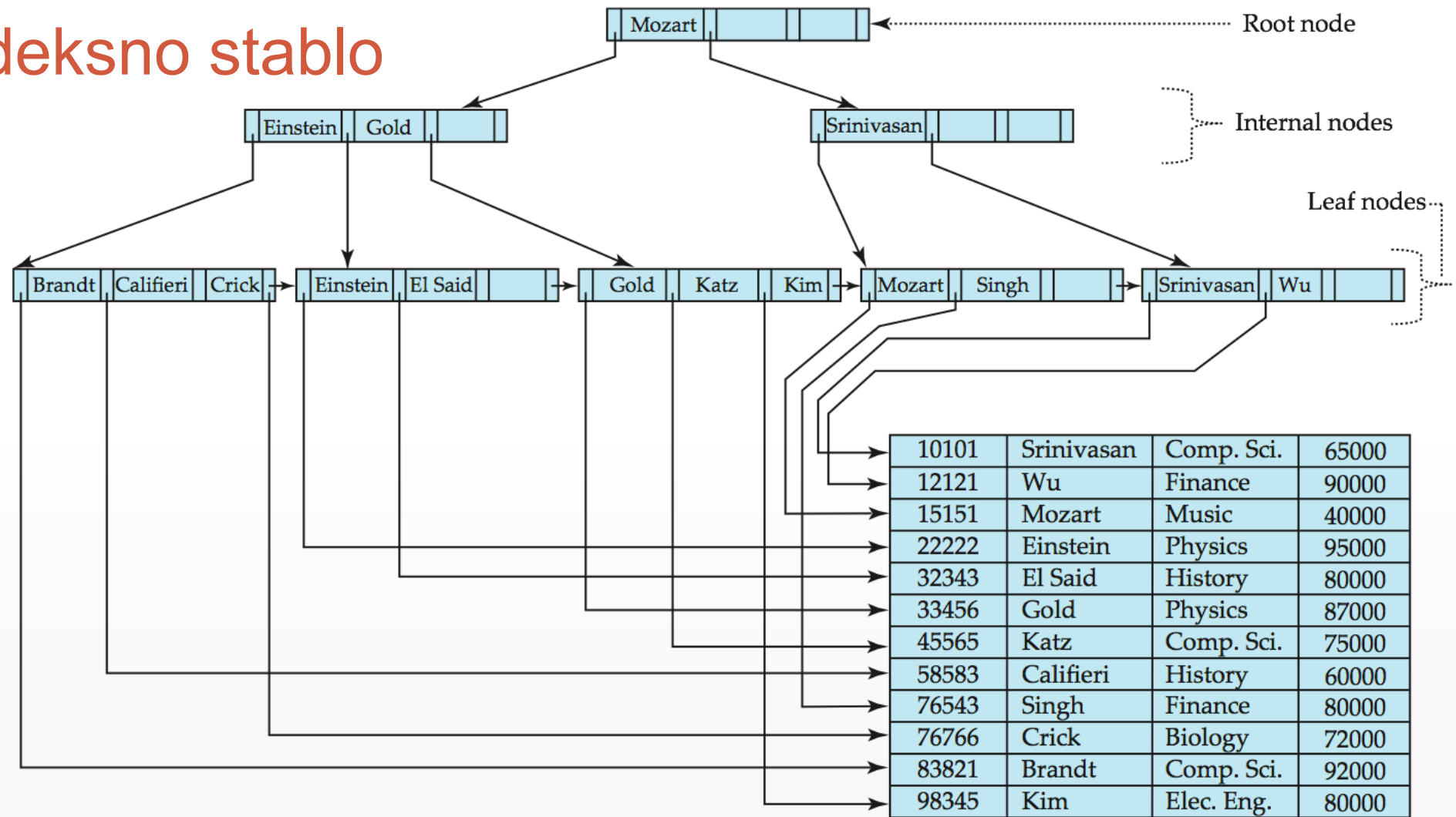
Uređeni indeksi

Koncept B+ stabla

B+ indeksno stablo

- **B stablo** - balansirano stablo sa fiksno definisanim maksimalnim brojem potomaka (*fan-out*); u njemu se svaka vrednost ključa pojavljuje jednom u stablu na nekom njegovom nivou zajedno sa pokazivačem na ostale podatke odgovarajućeg sloga ili bloka.
 - **B+ stablo** – B stablo u kome se pokazivači na ostale podatke nalaze samo u listovima stabla.
 - Svi listovi B+ stabla imaju isti nivo i međusobno su spregnuti pokazivačima.
 - Balansirana struktura.
 - Pristup, dodavanje i brisanje imaju logaritamsku kompleksnost
 - Optimizovan za čitanje/pisanje velikih blokova podataka
-

B+ indeksno stablo



B+ indeksno stablo

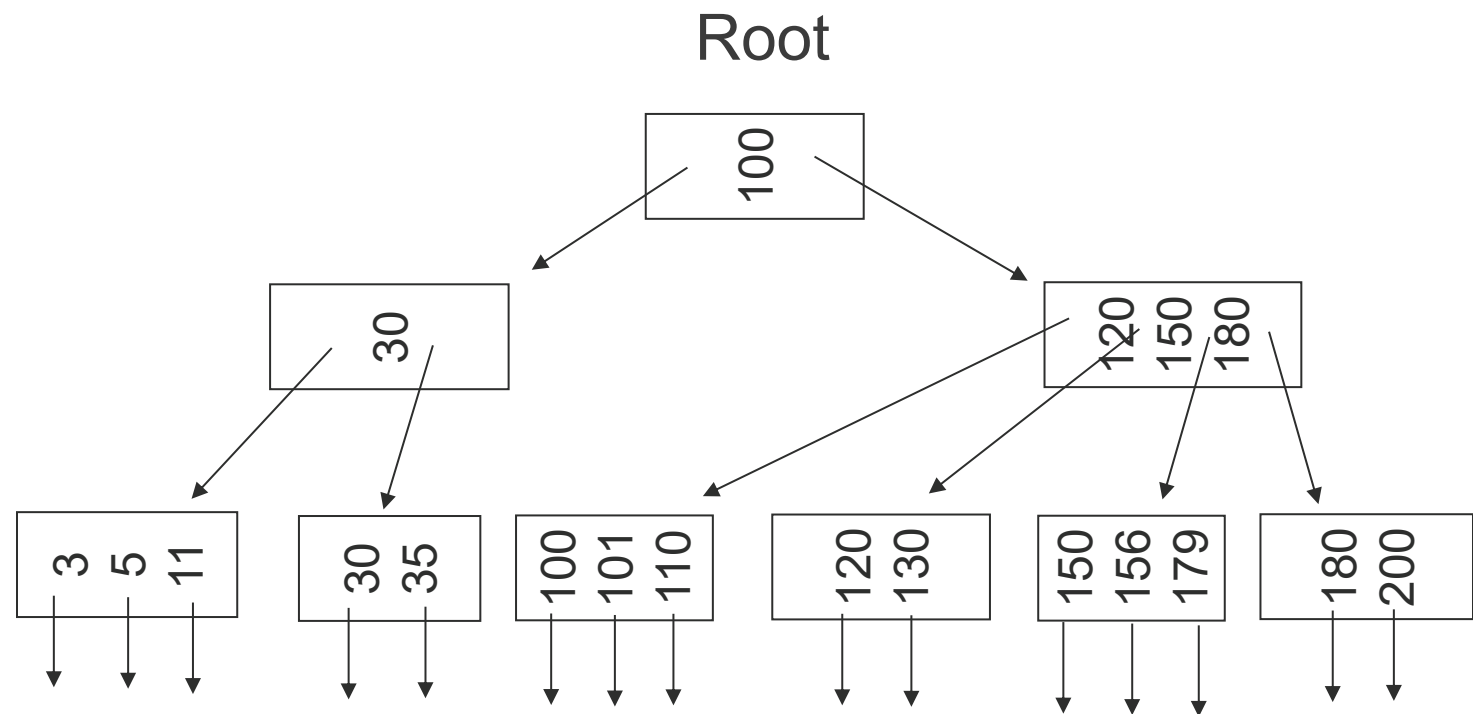
P_1	K_1	P_2	$\dots$	P_{n-1}	K_{n-1}	P_n
-------	-------	-------	---------	-----------	-----------	-------

- K_i su vrednosti indeksiranog polja

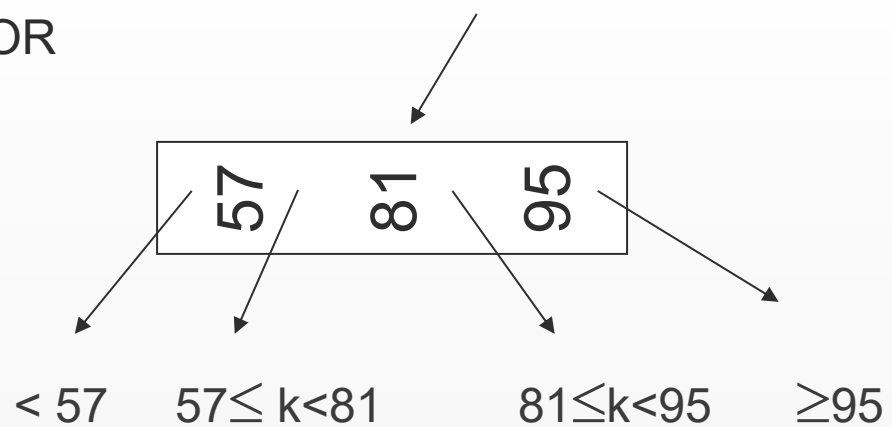
P_i pokazivači na decu (ako nisu listovi u pitanju) ili pointeri na slogove u fajlu sa podacima.

- B+ je balansirano stablo u kojem svaki put pretrage od korena do lista ima istu dužinu
 - Svaki unutrašnji čvor je najmanje do pola pun ($n/2 - 1 \leq \text{num of keys} \leq n - 1$).
 - Svaki unutrašnji čvor sa k ključeva ima $k+1$ ne-null dete.
 - Listovi sadrže:
 - Pokazivače na lokacije torki
 - Kompletne torke (ređi slučaj).
-

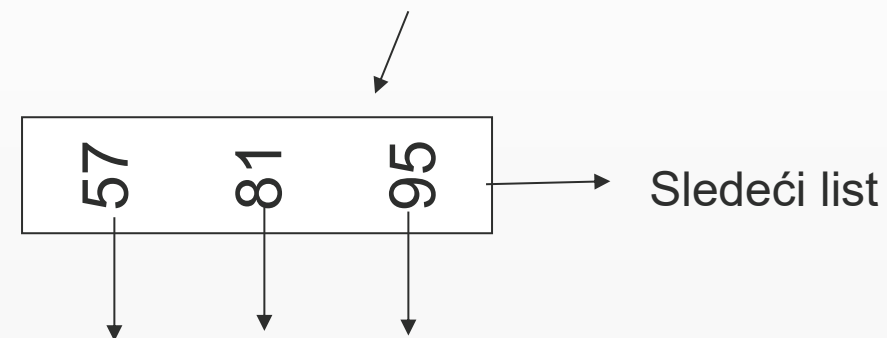
B+ indeksno stablo



ČVOR



LIST

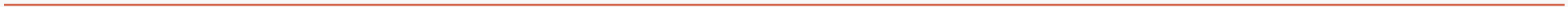


Pretraga, dodavanje, masovno učitavanje (*bulk load*)

Obavezno pogledati predavavanja

[B+ Trees \(Lecture 6\)](#)

[Refinement on Indices \(Lecture 7\)](#)



B+ indeksno stablo

Dodavanje / brisanje

[B+ Tree Visualization \(usfca.edu\)](http://usfca.edu)

