INDEKSI

Predmet: Administriranje Baze Podataka

Predavač: dr Dušan Stefanović

ULOGA INDEKSA

Dizajn baze podataka

Kako baza postaje sve veća dizajn baze postaje sve važniji. Podaci u tabelama nisu struktuirani i nisu poređani.

Uloga indeksa

Indeksi omogućavaju brži pristup određenim redovima unutar velikih tabela

Sličnost sa indeksima u knjizi

Ukoliko nas nešto posebno interesuje u knjizi ne želimo da pročitamo celu knjigu da bi smo pronašli deo koji nas interesuje

Osnovni mehanizam performansi

Indeksi ubrzavaju pristup podacima smanjenjem broja redova koje je potrebno pretražiti.

Trajna struktura podataka

Indeks je persistenta struktura podataka, automatski održavana od strane DBMS-a. Implementira se korišćenjem struktura: B+ stabla (najčešće korišćena), Hash tabele ili Bitmap.

Problemi implementacije

Implementacija može izazvati sporije INSERT/UPDATE/DELETE operacije i veću potrošnju memorije.

Fokus na upotrebu aplikacije

Aplikacija koristi indekse za povećanje performansi sistema na osnovu Analiza upita, Pravilan dizajn upita, Praćenje performansi, da bi se uočilo koji se indeksi koriste, a koji ne.

Utility

Index = razlika izmedju kompletnog skeniranja tabele i neposredne lokacije

Kompletno skeniranje tabele

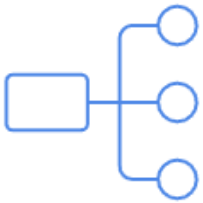
Sporo, naročito na velikim skupovima podataka.

Neposredni pristup podacima

Brža pretraga pomoću specifične strukture.

Utility

Osnovna struktura podataka kod indeksiranja



Uravnoteženo stablo

B stablo: Omogućava logaritamski pristup podacima. B+ stablo (najčešći tip u SQL bazama), brže za sekvencijalni pristup..

Hash tabela

Efikasna za tačno pronalaženje vrednosti, nije pogodna za rasponske upite.

Karakteristika	**B+ stablo**	**Hash tabela**
Tip upita	Idealno za **range upite** (<, >, BETWEEN)	Idealno za **tačne pretrage** (=)
Poredak podataka	Čuva podatke u **sortiranom** redosledu	Podaci **nisu sortirani**
Pretraga	Brza, **logaritamska složenost**	Veoma brza, **konstantna složenost** (idealno)
Navigacija kroz podatke	Omogućava **sekvencijalno skeniranje**	Nije pogodna za sekvencijalnu pretragu
Korišćenje u JOIN-ovima	Da (često efikasno za sortirane JOIN-e)	Neefikasno za JOIN-ove bez tačnih ključeva
Tipični slučajevi upotrebe	WHERE uslovi sa opsezima, ORDER BY, MIN/MAX	WHERE uslovi sa tačnim ključem (id = 123)
Podrška u DBMS-ima	**Standardno u većini DBMS-a**	Ograničeno; **npr. PostgreSQL podržava hash**
Održavanje indeksa	Više troška pri INSERT/DELETE	Manje fleksibilna, može zahtevati rehashing

```
Select sName
From Student
Where sID=18942
```

Mnogi DBMS kreiraju indekse automatski nad atributima koji su PRIMARNI KLJUČEVI (ponekad nad UNIQUE atributima)

```
Select sID
From Student
Where Sime = 'Marko' And Prosek > 3.9
```

Karakteristika	Hash tabela	B+ stablo (kompozitni indeks)
Tačno poređenje (`sIme = 'Marko'`)	Odlično	Podržano
Rasponski uslov (`Prosek > 3.9`)	Loše	Podržano
Kombinovani uslovi	Nije pogodna	Optimalno

```
Select Sime, Fime
From Student, Prijava
Where Student.sID = Prijava.sID
```

Karakteristika	Indeks Prijava.sID	Indeks Student.sID	Oba sID indeksirana
Analiza	Za svaki sID iz tabele Student, odmah se pronalazi odgovarajući red u Prijava pomoću indeksa.	Za svaki red u tabeli Prijava, mora se skenirati cela tabela Student da bi se pronašao sID	Kada su oba sID atributa indeksirana, mogu se efikasno koristiti kod spjanja tabela:
Brzina	Brzo i efikasno	Sporije	Najbrže
Efikasnost	Visoko	Nisko	Optimalno

Zadatak

Razmotriti sledeći upit:

```
Select *
From Prijava, Fakultet
Where   Prijava.Fime = Fakultet.Fime and
            Prijava.smer = 'SRT' and
            Fakultet.BrojMesta < 5000
```

Koji od sledećih indeksa ne bi ubrzali izvršenje upita?

a) Stablo indeks na atribut Prijava.Fime
b) Hash indeks na atribut Prijava.Smer
c) Hash indeks na atribut Fakultet.BrojMesta
d) Hash indeks na atribut Fakultet.Fime

Zadatak

Razmotriti sledeći upit:

```
Select *
From  Student, Prijava, Fakultet
Where   Student.Sid=Prijava.Sid and
        Prijava.Fime = Fakultet.Fime and
        Student.Prosek > 1.5 and
        Fakultet.Fime = 'VTS'
```

Pretpostavimo da nam je dozvoljeno da kreiramo dva indeksa i da su svi indeksi tipa stablo.Koja od dva indeksa bi najviše ubrzala izvršenje upita?

a) Student.Sid, Fakultet.Fime
b) Student.Sid, Student.Prosek
c) Prijava.Fime,Fakultet.Fime
d) Prijava.Sid, Student.Prosek

NEDOSTACI INDEKSA

Karakteristika	Prostor na disku	Vreme kreiranja	Ažuriranje
Uticaj	Dodatna fizička struktura koja se čuva u bazi	Kreiranje indeksa nad velikim tabelama može zahtevati značajno vreme, posebno ako tabela ima milione redova.	Kod svake INSERT, UPDATE ili DELETE operacije indeks mora biti ažuriran u realnom vremenu
Razmatranja	Marginalni nedostatak u odnosu na brzinu	Tabela može biti zaključana ili usporena	Trošak može biti veći od koristi

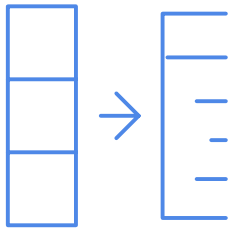
PREDNOSTI UPOTREBE INDEKSA ZAVISE OD:

Veličina tabela

Što je tabela veća, to indeksiranje postaje značajnije za performanse.
Bez indeksa, pretraga kroz velike tabele zahteva full table scan, što je izrazito sporo.
Indeks omogućava logaritamski pristup podacima umesto linearnog.

Distribucija podataka

Kolona ima visoku selektivnost (mnoge različite vrednosti), indeks je efikasan.
Primer: JMBG, ID korisnika.
Ako kolona ima nisku selektivnost (malo različitih vrednosti), indeks neće značajno pomoći (pol, status sa 2–3 vrednosti* → koristimo bitmap indeks umesto B+ stabla



Upiti vs. Ažuriranja

Ako se tabela češće čita nego što se menja (read-heavy): Indeksiranje ima velike prednosti. Ako se tabela često menja (write-heavy): Održavanje indeksa postaje značajan trošak.

CLUSTER INDEKS

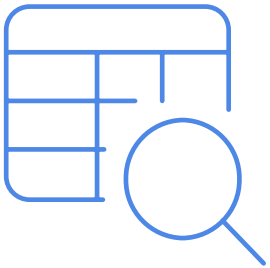
Osobina	Opis
Fizička organizacija	Sortirani redovi po koloni
Jedinstvenost	Obično primarni ključ
Broj po tabeli	Samo jedan cluster indeks
Brz pristup	Efikasan pretraživač po koloni
Promenljivo	Može se redefinisati na drugu kolonu

SifraMusterije	Ime	Prezime	Adresa	Email
238	Milan	Stanko	SZK	ms@it.rs
234	Dejan	Mitic	BB	dm@ni.rs
237	Marina	Pantic	VK9	mp@vts.rs
...	...	...	...	...

SifraMusterije	Ime	Prezime	Adresa	Email
234	Dejan	Mitic	BB	dm@ni.rs
237	Marina	Pantic	VK9	mp@vts.rs
238	Milan	Stanko	SZK	ms@it.rs
...	...	...	...	...

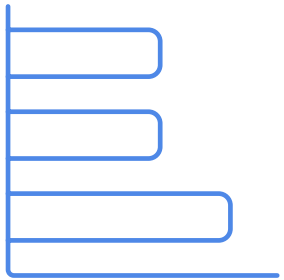
Cluster indeks

SCENARIJI KORIŠĆENJA CLUSTER INDEKSA



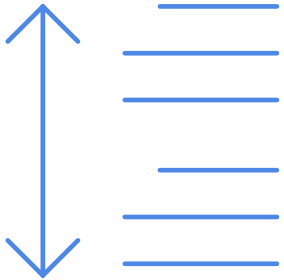
Pristup podacima

Kada je pristup podacima dominantno po vrednosti određene kolone (npr. ID, datum).



Range upiti

Česta upotreba range upita je prisutna (BETWEEN, >, <).



Redosled podataka

Redosled podataka omogućava efikasno grupisanje i sortiranje.

CLUSTER INDEKS - PRIMER

SifraMusterije	Ime	Prezime	Adresa	Email
234	Dejan	Mitic	BB	dm@ni.rs
237	Marina	Pantic	VK9	mp@vts.rs
238	Milan	Stanko	SZK	ms@it.rs
...	...	...	...	...

Cluster indeks

```
SELECT *

FROM Klijent

WHERE SifraMusterije=237;
```

Pretraga će biti brža jer pretraživanje radimo po primarnom ključu koji je indeksiran

NON CLUSTER INDEKS

SifraMusterije	Ime	Prezime	Adresa	Email
234	Dejan	Mitic	BB	dm@ni.rs
237	Marina	Pantic	VK9	mp@vts.rs
238	Milan	Stanko	SZK	ms@it.rs
...	...	...	...	...

Može da ih bude **više u tabeli**

```
SELECT *
FROM Klijent
WHERE Prezime='Pantic';
```

- Pošto DBMS ne zna gde se nalazi traženo prezime on vrši pretragu redom od prvog do poslednjeg reda
- Full table scan, neefikasno pretraživanje traženog podatka
- Neefikasnost raste sa brojem redova u tabeli

NON CLUSTER INDEKS

- Rešenje je sekundarni indeks (none clustered index) koji se može kreirati.
- Kreira se posebna tabela koja je sortirana na osnovu Prezimena
- Pretraživanje na osnovu prezimena je sada ubrzano
- Ubrzanje nije ekvivaletno cluster indeksu jer je potrebno pretraživanje sprovesti u dve tabele

Non-Cluster indeks za Prezime

Prezime	SifraMusterije
Aleksic	112
Antic	246
Bobic	78
...	...
Pantic	237

Cluster indeks

SifraMusterije	Ime	Prezime	Adresa	Email
234	Dejan	Stevic	BB	dm@ni.rs
237	Marina	Pantic	VK9	mp@vts.rs
238	Milan	Stanko	SZK	ms@it.rs
...	...	...	...	...

NON CLUSTER INDEKS

Non-Cluster indeks za Prezime

Prezime	SifraMusterije
Aleksic	112
Antic	246
Bobic	78
...	...
Pantic	237

Cluster indeks

SifraMusterije	Ime	Prezime	Adresa	Email
234	Dejan	Stevic	BB	dm@ni.rs
237	Marina	Pantic	VK9	mp@vts.rs
238	Milan	Stanko	SZK	ms@it.rs
...	...	...	...	...

Preporuka je koristi ovaj indeks samo za kolone za koje znamo da će se dosta koristiti u upitima

```
SELECT *
FROM Klijent
WHERE Prezime='Pantic';
```

NON CLUSTER INDEKSI - NEDOSTATAK

Postavlja se pitanje zašto ne indeksiramo sve kolone u tabeli i obezbedimo brži prikaz podataka

Problem je što svaki indeks ima svoju cenu.

Prednost indeksa je u čitanju (select) podataka

Nedostatak je u upisu(insert), promeni(update) I brisanju (delete) podataka

BALANS IZMEĐU BRZINE ČITANJA I TROŠKOVA ODRŽAVANJA

NON CLUSTER INDEKSI - NEDOSTATAK

U našem slučaju pretraživanje na osnovu imena ili prezimena biće brzo jer smo kreirali dva none cluser indeksa nad ovim kolonama

Ukoliko kreiramo novog zaposlenog umesto jedne operacije fizičkog upisa na disku to će zahtevati tri upisa jer imamo tri tabele (cluster index i dva none cluster indeksa).

Što je broj indeksnih kolona veći, duže je vreme potrebno za fizički upis podataka što dovodi do neefikasnosti baze.

clustered index

EmployeeID	FirstName	MiddleInitial	LastName	...
2	Aaron	F	Cooper	...
4	Lou	(null)	Donoghue	
5	Alice	B	Bailey	
6	Oswald	T	Hall	
7	John	(null)	Velasquez	
8	Brenda	A	Daniels	

non-clustered index on FirstName

FirstName	EmployeeID
Aaron	2
Alice	5
Brenda	8
John	7
Lou	4
Oswald	6

non-clustered index on LastName

LastName	EmployeeID
Bailey	5
Cooper	2
Daniels	8
Donoghue	4
Hall	6
Velasquez	7

INDEKSIRANJE - ZAKLJUČAK

Indeksiranje zahteva od administratora baze podataka konstatno nadgledanje performansi sistema

Indeksiranje je kopmpromis između:

- Bržeg čitanja podataka
- Sporijeg upisa podataka

INDEKSIRANJE - PRIMERI

Uputstvo

Tabelu users_details koja sadrži milion redova sa sledećeg linka ubaciti u unapred kreiranu bazu db_over_1M

```
desc user_details
```

Field	Type	Null	Key	Default	Extra
user_id	int	NO	PRI	NULL	auto_increment
username	varchar(255)	YES		NULL	
first_name	varchar(50)	YES		NULL	
last_name	varchar(50)	YES		NULL	
gender	varchar(10)	YES		NULL	
password	varchar(50)	YES		NULL	
status	tinyint	YES		NULL	

```
select count(*) as 'Ukupan broj redova' from user_details
```

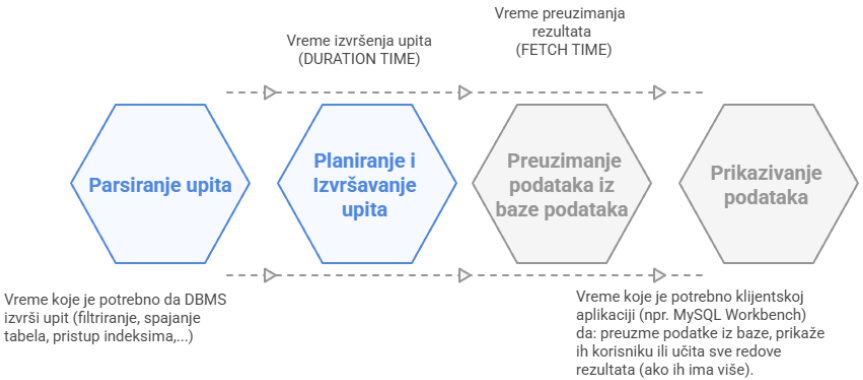
Ukupan broj redova
1000000

INDEKSIRANJE – SLUČAJ 1

Ispitati koliko je vremena potrebno da se prikažu svi podaci za imena kod kojih je user_id između 100000 i 500000.

```
select *
 from user_details
where user_id between 100000 and 500000
```

INDEKSIRANJE – SLUČAJ 1



Duration time – je vreme potrebno da se upit izvrši

Fetch time – vreme za preuzimanje i prikaz razultata

Action	Message	Duration / Fetch
select * from user_details where user_id between 100000 and 50...	1000 row(s) returned	0.000 sec / 0.016 sec
select * from user_details where user_id between 100000 and 50...	1000 row(s) returned	0.000 sec / 0.015 sec

Pretraga je rađena po koloni user_details koja je primarni ključ i indeks

INDEKSIRANJE – SLUČAJ 1a

Ispitati koliko je vremena potrebno da se prikažu svi podaci za imena kod kojih je user_id između 100000 i 500000 i sortirati po koloni last_name

```
select * from user_details
 where user_id between 100000 and 500000
order by last_name asc
```

Sortirati po user_id koloni ?

Action	Message	Duration / Fetch
select * from user_details where last_name='John' and gender='m...	1000 row(s) returned	0.063 sec / 0.234 sec
select * from user_details where user_id between 100000 and 50...	1000 row(s) returned	6.313 sec / 0.000 sec

Pretraga je rađena po koloni user_details koja je primarni ključ i indeks ali je sortiranje rađeno po koloni last_name koja nije indeksirana.

INDEKSIRANJE – SLUČAJ 2

Ispitati koliko je vremena potrebno da se prikažu svi podaci za imena koja počinju sa John.

```
select *
from user_details
 where last_name='John'
```

Action	Message	Duration / Fetch
select count(*) as 'Ukupan broj redova' from user_details LIMIT 0,...	1 row(s) returned	0.000 sec / 0.000 sec
select * from user_details where last_name='John' LIMIT 0, 1000	1000 row(s) returned	0.031 sec / 0.125 sec

Pretraga je rađena po koloni last_name koja nije indeksirana

Potrebno je bilo 31ms da se prikaže odgovor

INDEKSIRANJE – SLUČAJ 3

Ispitati koliko je vremena potrebno da se prikažu svi podaci za imena koja počinju sa John i da su muškog pola

```
select *
from user_details
 where last_name='John' and gender='male'
```

Action	Message	Duration / Fetch
select * from user_details where last_name='John' LIMIT 0, 1000	1000 row(s) returned	0.031 sec / 0.125 sec
select * from user_details where last_name='John' and gender='m...	1000 row(s) returned	0.063 sec / 0.234 sec

Pretraga je rađena po koloni last_name i koloni pol koje nisu indeksirane

INDEKSIRANJE – SLUČAJ 4

- Prikazati ukupan broj različitih prezimena u bazi

```
select count(distinct last_name)
from user_details
```

Action	Message	Duration / Fetch
alter table user_details drop index ind_last	1000000 row(s) affected Records: 1000000 Duplicates: 0 Warni...	11.687 sec
select count(distinct last_name) from user_details LIMIT 0, 1000	1 row(s) returned	5.922 sec / 0.000 sec

Eliminisanje duplikata po koloni koja nije indeksirana zahteva znatno više vremena

INDEKSIRANJE – SLUČAJ 5

Indeksiramo kolonu last_name

```
alter table user_details add index ind_last(last_name)
```

Ispitati koliko je vremena potrebno da se prikažu svi podaci za imena koja počinju sa John.

```
select *
from user_details
where last_name='John'
```

Action	Message	Duration / Fetch
select count(*) as 'Ukupan broj redova' from user_details LIMIT 0,...	1 row(s) returned	0.000 sec / 0.000 sec
select * from user_details where last_name='John' LIMIT 0, 1000	1000 row(s) returned	0.031 sec / 0.125 sec

Nije indeksirana kolona last_name

	Action	Message	Duration / Fetch
8	select * from user_details where last_name='John' LIMIT 0, 1000	1000 row(s) returned	0.000 sec / 0.000 sec
2	select * from user_details where last_name='John' LIMIT 0, 1000	1000 row(s) returned	0.000 sec / 0.015 sec

Indeksirana kolona last_name

INDEKSIRANJE – SLUČAJ 6

Ispitati koliko je vremena potrebno da se prikažu svi podaci za imena kod kojih je user_id između 100000 i 500000 i sortirati po koloni last_name

```
select * from user_details
where user_id between 100000 and 500000
order by last_name asc
```

Action	Message	Duration / Fetch
select * from user_details where last_name='John' and gender='m...	1000 row(s) returned	0.063 sec / 0.234 sec
select * from user_details where user_id between 100000 and 50...	1000 row(s) returned	6.313 sec / 0.000 sec

Nije indeksirana kolona last_name

Action	Message	Duration / Fetch
select * from user_details where user_id between 100000 and 50...	1000 row(s) returned	0.031 sec / 0.000 sec
select * from user_details where user_id between 100000 and 500...	1000 row(s) returned	0.031 sec / 0.000 sec

Indeksirana kolona last_name

INDEKSIRANJE – SLUČAJ 7

Prikazati ukupan broj različitih prezimena u bazi

```
select count(distinct last_name)
from user_details
```

Action	Message	Duration / Fetch
alter table user_details drop index ind_last	1000000 row(s) affected Records: 1000000 Duplicates: 0 Warni...	11.687 sec
select count(distinct last_name) from user_details LIMIT 0, 1000	1 row(s) returned	5.922 sec / 0.000 sec

Nije indeksirana kolona last_name

Action	Message	Duration / Fetch
select * from user_details where user_id between 100000 and 500...	1000 row(s) returned	0.031 sec / 0.000 sec
select count(distinct last_name) from user_details LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec

Indeksirana kolona last_name

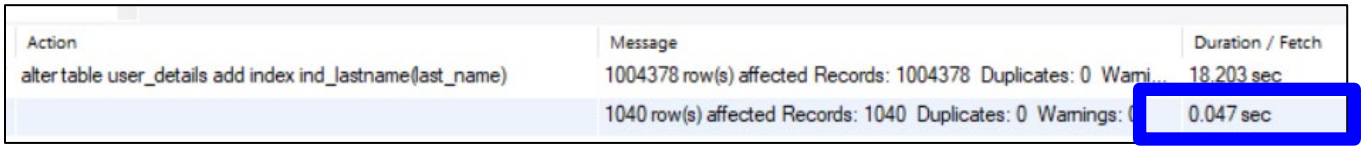
INDEKSIRANJE – SLUČAJ 8

Upis podataka u bazu

Upisano je 1040 redova odjednom

	Action	Message	Duration / Fetch
2	insert into user_details (username,first_name,last_name,gender,password,sta...	908 row(s) affected Records: 908 Duplicates: 0 Warni...	0.015 sec
1		1040 row(s) affected Records: 1040 Duplicates: 0 War...	0.015 sec

Upis 1040 redova bez indeksirane kolone last_name



Action	Message	Duration / Fetch
alter table user_details add index ind_lastname(last_name)	1004378 row(s) affected Records: 1004378 Duplicates: 0 Warni...	18.203 sec
	1040 row(s) affected Records: 1040 Duplicates: 0 Warnings: 0	0.047 sec

Upis 1040 redova sa Indeksiranom kolonom last_name

PRIKAZ INDEKSIRANIH KOLONA

```
use information_schema;
SELECT * FROM statistics;
```

Prikaz svih indeksa u svim bazama

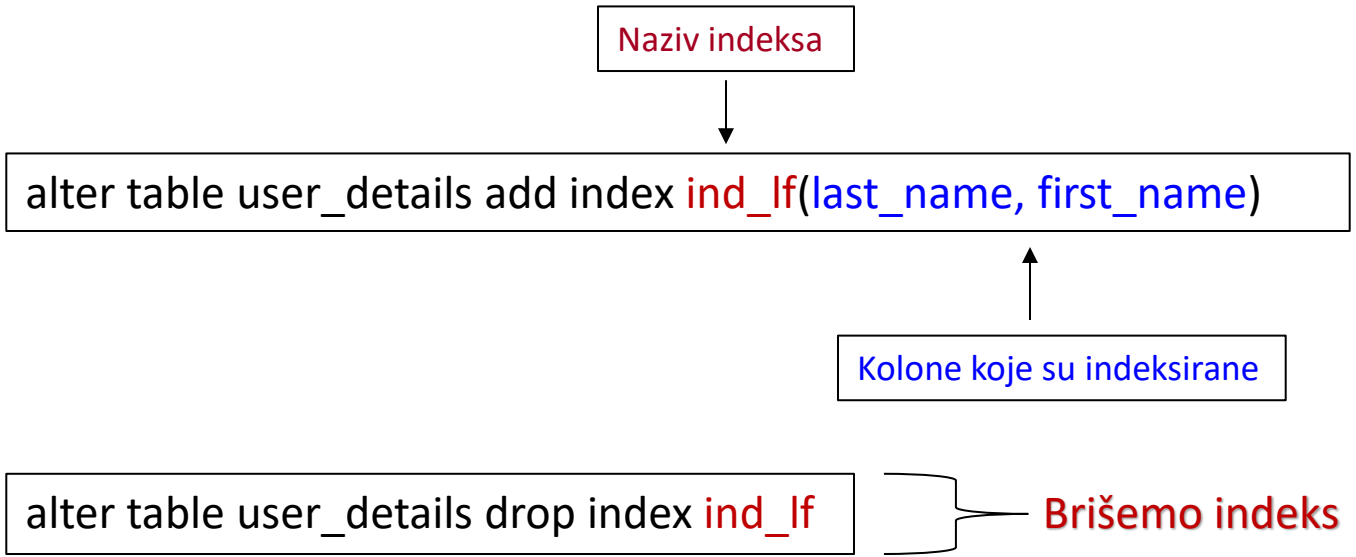
```
SHOW INDEX FROM user_details
ili
SHOW INDEX IN user_details
```

```
SHOW INDEX FROM db_over_1m.user_details
```

Prikaz svih indeksa u tabeli user_details u bazi db_over_1m

INDEKSIRANJE VIŠE KOLONA

Na ovaj način indeksiramo kolonu last_name i kolone last_name i first_name ali ne i kolonu first_name.



ANALIZA IZVRŠENJA UPITA

Explain komanda ne izvršava upit već objašnjava kako će izvršiti upit

```
explain select username
from user_details
where username='Dusan'
```

Ne koristi indekse

Broj pregledanih redova

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	user_details	NULL	ALL	NULL	NULL	NULL	NULL	1005418	10.00	Using where

Skenira se kompletna tabela

ANALIZA IZVRŠENJA UPITA

Explain komanda ne izvršava upit već objašnjava kako će izvršiti upit

```
explain select username
from user_details
where username='Dusan'
```

koristi indeks u pretrazi

Broj pregledanih redova

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	user_details	NULL	ref	ind_username	ind_username	258	const	1	100.00	Using index

Ne skenira se kompletna tabela