

# SQL

---

## Indeksi

# Uloga Indeksa

---

- Kako baza postaje sve veća dizajn baze postaje sve važniji
- Indeksi igraju značajnu ulogu u brzini rada same baze jer pomažu DBMS da brže pronađe ono što tražimo
- Slični su indeksima koji se koriste u knjizi.
  - Ukoliko nas nešto posebno interesuje u knjizi ne želimo da pročitamo celu knjigu da bi smo pronašli deo koji nas interesuje
- Podaci u tabeli nisu strukturirani i nisu poređani
- Indeksi nam omogućavaju da brže pristupimo određenom redu u velikoj tabeli

# Cluster Indeks

- Primarni indeks koji se kreira u svakoj tabeli je cluster indeks
  - Redovi u tabeli se organizuju po toj koloni
  - Cluster indeks je sličan prezimenu u telefonskom imeniku
- Cluster indeks je u tabeli primarni ključ i većina DBMS će to uraditi automatski jer je primarni ključ označen kao jedinstven preko koga pristupamo svakom redu
- Ukoliko pristupamo podacima koristeći neku drugu kolonu možemo da promenimo cluster indeks.
- Svaka tabela može da ima samo jedan cluster indeks isto kao što telefonski imenik može da bude sortiran na samo jedan način

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**

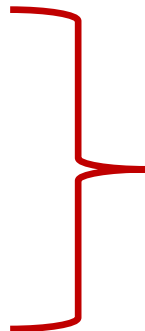
# Cluster Indeks - Primer

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 Prezime
- Pretraživanje na osnovu prezimena je sada ubrzano
- Ubrzanje **nije ekvivalentno 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
...	...	...	...	...

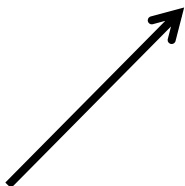
# Non Cluster Indeks

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 koristiti 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 pisanju(**insert**) i promeni(**update**) podataka



# Non Cluster Indeksi - Nedostatak

- U našem slučaju pretraživanje na osnovu imena ili prezimena biće brzo jer smo kreirali dva none cluster 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 dve none cluster indeks).
- 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 **trgovina** između:
  - **Bržeg čitanja** podataka
  - **Sporijeg upisa** podataka

# Indeksiranje - Primer

---

- 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
```

*Fetch time – prenos i prikazivanje rezultata*

*Duration time – je vreme potrebno da se upit izvrši*

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 Wami...	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
insert into user_details (username,first_name,last_name,gender,password,sta...	908 row(s) affected Records: 908 Duplicates: 0 Wami...	0.015 sec
	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 Wami...	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.

Naziv indeksa

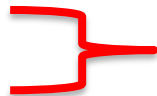


```
alter table user_details add index ind_lf(last_name, first_name)
```



Kolone koje su indeksirane

```
alter table user_details drop index ind_lf
```



Brišemo indeks

# 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	<small>NULL</small>	ref	ind username	ind username	258	const	1	100.00	Using index

Ne skenira se kompletna tabela