

Baze podataka II

Modul 8 – Jezik SQL

DDL komande

Rad sa indeksima



Summary

- Uvod u indekse;
- Tipovi podataka i indeksi;
- Planiranje indeksa;
- Kreiranje indeksa;
- Optimizacija indeksa;



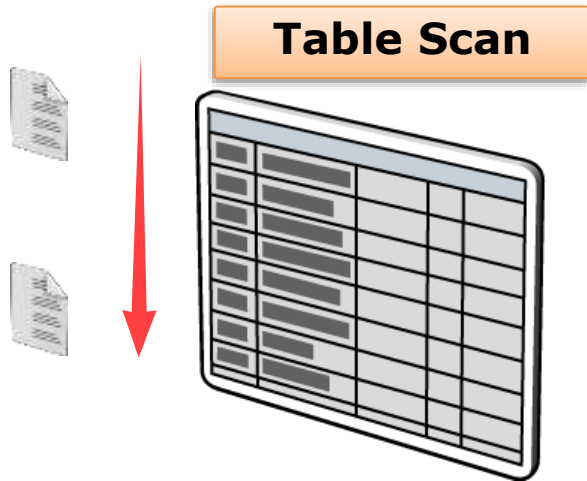
Lekcija 1: Uvod u indekse

- Kako SQL Server pristupa podacima;
- Potreba za indeksima;
- Osnovni koncepti;
- Struktura indeksa;
- Osobine indeksa;

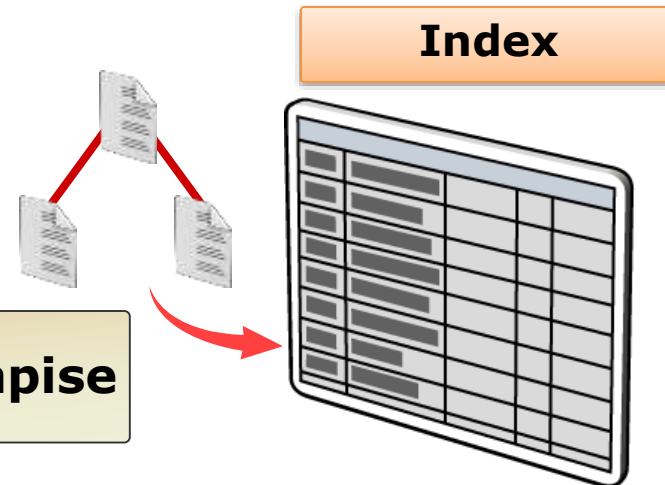


Kako SQL Server pristupa podacima ?

SQL Server čita sve stranice u tabeli



SQL Server koristi indeks kako bi pronašao zapise



Potreba za indeksima

- **ANSI SQL nije definisao indekse**
 - Smatraju se odvojenim od logičkog modela podataka
- **Svi upiti se mogu izvršiti bez indeksa**
 - Osnovni razlog zašto postoje su performanse
- **Nekada su ograničenja implementirana putem indeksa**
 - Naravno u teoriji/praksi postoji mnogo načina za ograničenja bez upotrebe indeksa
- **Analogija: Biblioteka**
 - Indeks po autorima je izuzetno poželjan
 - Dodatni indeksi bi također bili dobri
 - Oblasti, naslovi, ključne riječi i sl.



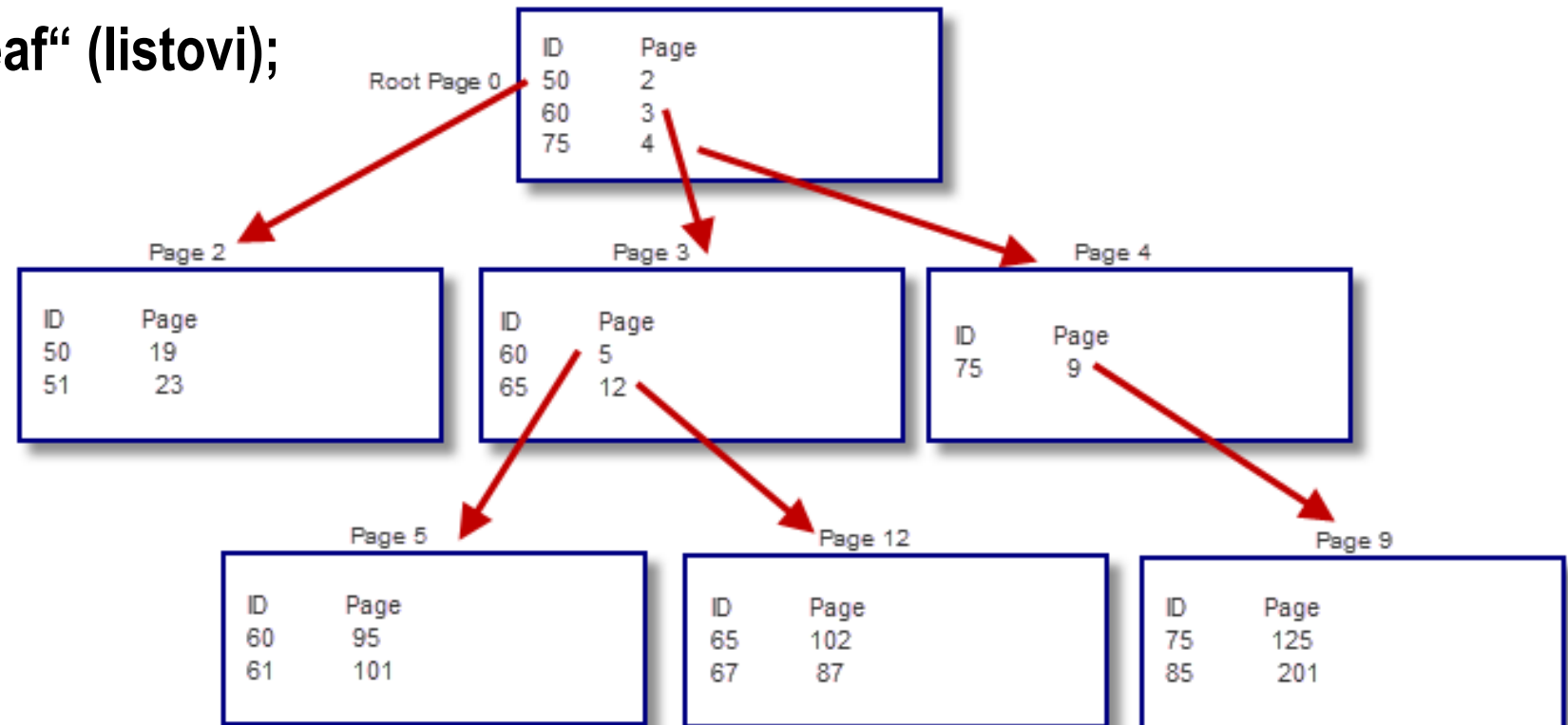
Osnovni koncepti

- **Indeksi ubrzavaju pristup željenim podacima**
 - npr. Lista studenata iz HNK
- **Ključ pretrage (Search Key)** – atribut koji se sastoji od jednog ili više atributa i koristi se za lociranje zapisa u fajlovima.
- **Indeks fajl** se sastoji od zapisa u obliku:

```
graph LR; A[Ključ pretrage] --- B[Pokazivač]
```
- **Indeks fajlovi su obično mnogo manji nego originalni fajlovi**
- **Postoje dva osnovna tipa indeksa:**
 - Poredani (ordered) indeksi: ključevi pretrage su poredani po određenom redoslijedu
 - Hash indeksi: ključevi pretrage su raspoređeni ravnomjerno po ćelijama upotrebom hash funkcije

Struktura indeksa

- Uglavnom su bazirani na strukturi stabla (tree);
- Prvi čvor se naziva „root“;
- Svi naredni su „leaf“ (listovi);



Osobine indeksa

- **Tri osnovna koncepta pri radu s indeksima:**
- **Selektivnost - *Selectivity***
 - Mjera, koliko redova se vratilo u odnosu na ukupan broj redova
 - Visoka selektivnost znači mali broj redova u odnosu na ukupan broj redova
- **Gustoća - *Density***
 - Mjera nedostatka jedinstvenosti podataka u tabeli
 - Visoka gustoća ukazuje na veliki broj duplikata
- **Dubina indeksa – *Index Depth***
 - Broj nivoa unutar indeksa
- **Parametri evaluacije**
 - Tip pristupa
 - Brzina pristupa
 - Vrijeme dodavanja
 - Vrijeme brisanja

Lekcija 2: Tipovi podataka i indeksi

- Numerički indeksi;
- Indeksi bazirani na tekstualnim tipovima;
- Indeksi bazirani na datumu;
- GUID indeksi;
- BIT indeksi;
- Indeks nad izračunatim kolonama;



Numerički indeksi

- **Indeksi sa numeričkim podacima rade efikasnije**
 - Više vrijednosti se nalazi u manjem broju stranica indeksa
 - Sortiranja i komparacije su brže
- **Egzaktni numerički tipovi su najefikasniji**
 - Cijeli brojevi su najefikasniji u grupi egzaktnih tipova
 - INT i BIGINT se najčešće koriste
- **Aproksimacijski tipovi podataka (float i real) su manje efikasni**

Indeksi bazirani na tekstualnim tipovima

- Tekstualni podaci su manje efikasni kada se koriste kao ključevi indeksa
- Tekstualne vrijednosti imaju tendenciju da budu mnogo veće od numeričkih vrijednosti
- Čak i kratka tekstualna vrijednosti je spora za poređenje, osim ako nije binarna komparacija
 - Većina aplikacija koristiti upoređivanja koje nije binarno
 - Pravila za uspoređivanja se trebaju primijeniti prilikom svakog poređenja

Indeksi bazirani na datumu

- **Podaci bazirani na datum su generalno dobri kandidati za indekse**
 - Vrlo često se koriste u poslovnim aplikacijama
- **Tek nešto manje efikasniji nego indeksi sa cijelim brojevima**
- **Veličina podataka je važna**
 - Date je efikasniji od DateTime

GUID indeksi

- **Postali su vrlo česti u novim poslovnim aplikacijama**
- **Umjerena efikasnost**
 - Veličina je 128 bita ili 16 bajta
 - Performanse poređenja su prihvatljive
- **Problemi nastaju zbog:**
 - Slučajnost (random generator) stvara probleme fragmentacije
 - Vrlo čest problem u mnogim aplikacijama

BIT indeksi

- BIT kolone imaju samo dvije moguća vrijednosti
- BIT kolone su efikasne kao ključevi indeksa
- Zabluda je da BIT kolone nisu korisni u procesu indeksiranja

Indeks nad izračunatim kolonama

- **Izračunate kolone su bazirane osnovu izraza**
 - Vrijednosti se obično izvode iz drugih kolona
- **Indeksiranje izračunatih vrijednosti, a ne onih na osnovu kojih nastaje, može biti korisno**
 - Može pomoći u poboljšanju performansi na loše dizajniranim bazama podataka
 - Primjer: kolona koja se koristi za držanje vrijednosti koje su se trebale pohraniti u posebnim kolonama
- **Persisted opcija**
 - Ubrzava SELECT
 - Izračunava se prilikom INSERT i UPDATE procedura

Lekcija 3: Planiranje indeksa

- Šta je „Clustered Index“ ?
- Šta je gomila „heap“ ?
- Šta je „Nonclustered Index“ ?
- Jedna kolona vs. kompozitni indeksi;
- ASC vs. DESC;

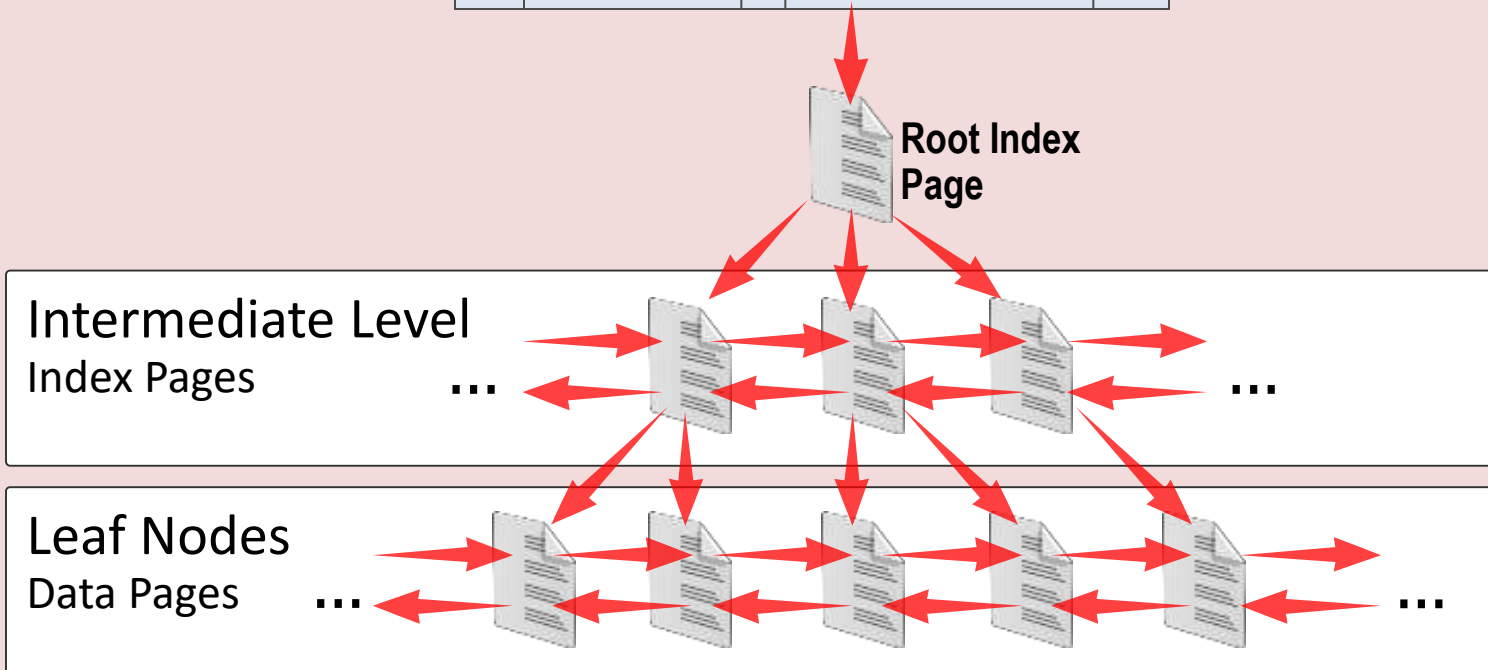


Šta je „Clustered Index“?

- Samo jedan clustered index po tabeli
- B-tree pohranjuje data stranice u redosljedu indexa

sys.partitions

id	index_id = 1	root_page	
----	--------------	-----------	--



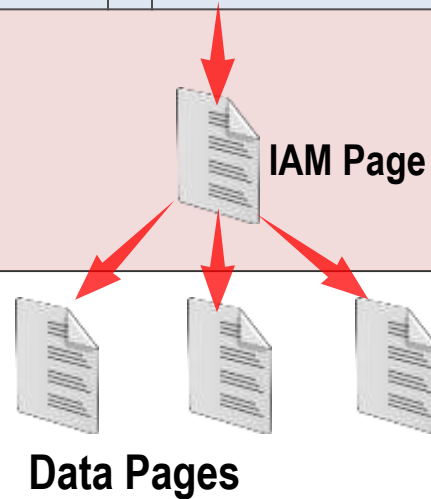
Šta je gomila „heap“?

- To je tabela bez clustered indexa
- Stranice se pohranjuju bez posebnog redosljeda

sys.partitions

id	index_id = 0	first_iam_page	
----	--------------	----------------	--

Heap

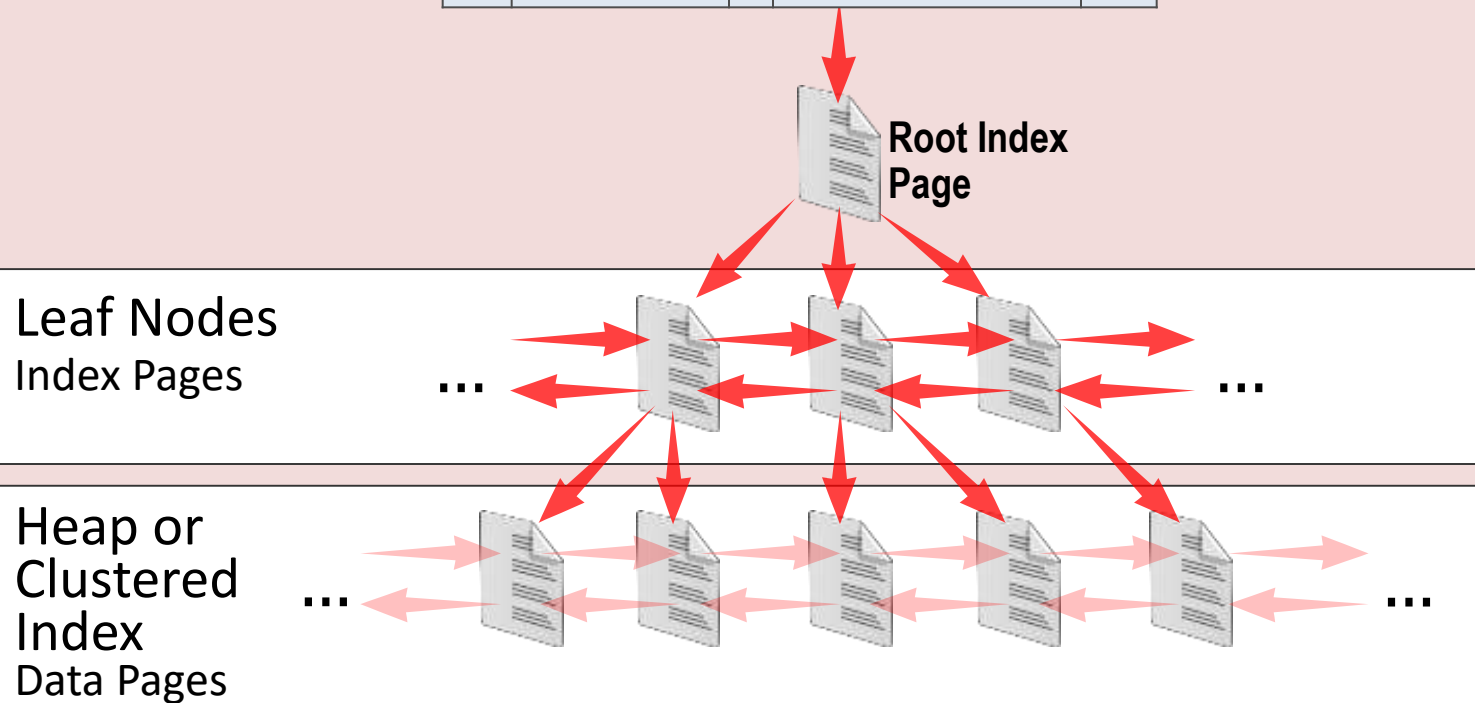


Šta je „Nonclustered Index“?

- B-tree struktura
- Do 249 non-clustered indeksa po tabeli

sys.partitions

id	index_id > 1	root_page	
----	--------------	-----------	--



Jedna kolona vs. kompozitni indeksi

- **Indeksi nisu uvijek bazirani na jednoj koloni**
 - Indeksi na više kolona se nazivaju "kompozitni" indeksi
- **Kompozitni indeksi su često korisni:**
 - Imaju tendenciju da budu korisniji od indeksa na jednoj koloni u većini tipičnih poslovnih aplikacija
 - Indeks poredan prvo po kupcu, a nakon toga po datumu ga čini efikasnim u pronalaženju narudžbi za određenog kupca na određeni datum.
 - Upit može uključivati više predikata u pretrazi
- **Indeks na A, B nije isto kao indeks na B, A**
 - Obično se prvo indeksira više restriktivnija kolona

ASC vs. DESC

- **Indeksi se mogu konstruisati u uzlaznom (ASC) ili silaznom (DESC) redoslijedu**
- **U principu, za indekse nad jednom kolonom, nema razlike**
 - Svaki sloj indeksa na SQL Serveru je dvostruko povezan (u oba smjera)
 - SQL Server može početi na jednom od dva i završiti na drugom kraju
- **Svaka komponenta kompozitnih indeks može biti ASC ili DESC**
 - Korisno za izbjegavanje SORT operacija

Lekcija 4: Kreiranje indeksa

- Kako kreirati indeks?
- Kreiranje Nonclustered indeksa
- Kompozitni indksi
- Klauzula INCLUDE
- Šta su UNIQUE indksi?
- Brisanje i izmjena indeksa



Kako kreirati indeks?

● SQL kod

```
CREATE [ UNIQUE ] [ CLUSTERED | NONCLUSTERED ]  
INDEX index_name ON { table | view } ( column [ ASC | DESC ] [ ,...n ]  
)  
INCLUDE ( column [ ,...n ] )  
[WITH option [ ,...n ] ][ON {partition_scheme (column) | filegroup |  
"default" } ]
```

● SSMS GUI

Kreiranje Nonclustered indeksa

```
CREATE TABLE dbo.Book
( ISBN nvarchar(20) PRIMARY KEY,
  PublisherID int NOT NULL,
  Title nvarchar(50) NOT NULL,
  ReleaseDate date NOT NULL
);
GO
CREATE NONCLUSTERED INDEX IX_Book_Publisher
  ON dbo.Book (PublisherID, ReleaseDate DESC);
GO
```


Kompozitni indeksi

- Do 16 kolona
- Neka prve kolone budu one koju su najviše unique

```
CREATE NONCLUSTERED INDEX Contact_LastName_FirstName  
ON Person.Contact ( LastName ASC, FirstName ASC)
```

Klauzula INCLUDE

- Omogućuje pohranu odabranih kolona podataka, ali samo na nivou lista nonclustered indeksa


```
CREATE NONCLUSTERED INDEX IX_Book_Publisher
ON dbo.Book (PublisherID, ReleaseDate DESC)
INCLUDE (Title);
GO

SELECT PublisherID, Title, ReleaseDate
FROM dbo.Book
WHERE ReleaseDate > DATEADD(year, -1, SYSDATETIME())
ORDER BY PublisherID, ReleaseDate DESC;
GO
```

Šta su UNIQUE indeksi?

Osigurava da nema duplih vrijednosti unutar indeksa

```
CREATE UNIQUE NONCLUSTERED INDEX [UQ_Employee_LoginID]  
ON [HumanResources].[Employee] ( [LoginID] ASC)
```



EmployeeID	LoginID	Gender	MaritalStatus	...
216	mike0	M	S	...
231	fukiko0	M	M	...
242	pat0	M	S	...

...



291

pat0

F

S

...

Vrijednost nije
dozvoljena

Brisanje i izmjena indeksa

- Indeksi se brišu putem **DROP INDEX** komande
 - Brisanje Clustered indeksa, tabela se konvertuje u heap
- Izmjena nad indeksom se radi putem **ALTER INDEX** komande

```
ALTER INDEX IX_Book_Publisher ON dbo.Book  
    DISABLE;  
GO
```

```
ALTER INDEX IX_Book_Publisher ON dbo.Book  
    REBUILD WITH ONLINE = ON;  
GO
```

```
DROP INDEX IX_Book_Publisher ON dbo.Book;  
GO
```

Lekcija 5: Optimizacija indeksa

- **SQL Server Profiler**
- **Database Engine Tuning Advisor**
- **Fragmentacija indeksa**
- **Opcije za fragmentaciju**

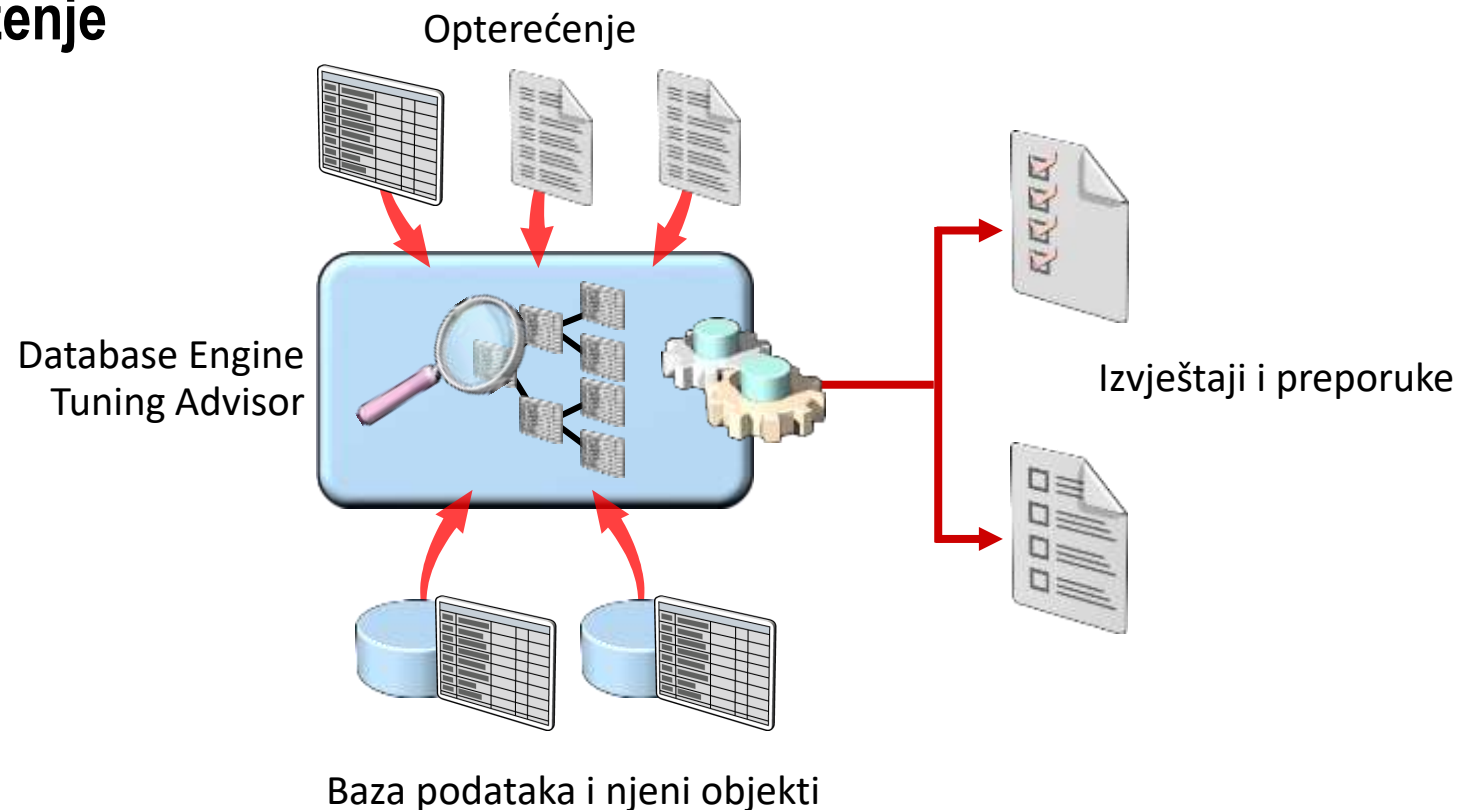


SQL Server Profiler

- **Prati sve komande koje se šalju prema SQL Serveru**
- **Prikuplja podatke o određenim događajima**
 - Zavisno od vrste praćenja (trace)
- **Trace se može koristiti prilikom:**
 - Dijagnostike
 - Utvrđivanja uskih grla sistema
 - Poboljšanja performansi
 - Audit
 - ...

Database Engine Tuning Advisor

- Analizira performanse sistema na osnovu podataka koje je prikupio SQL Server Profiler
- GUI i cmd okruženje



Fragmentacija indeksa

- **Kako se javlja fragmentacija?**

- SQL Server reorganizuje stranice kada se podaci modifikuju i time uzrokuje fragmentaciju

- **Vrste fragmentacije**

- Interna - stranice nisu pune
 - Vanjska - stranice nisu u logičkom redoslijedu

- **Otkrivanje fragmentacija**

- SQL Server Management Studio - Indeks Properties
 - Systemska funkcija sys.dm_db_index_physical_stats

```
SELECT * FROM sys.dm_db_index_physical_stats (NULL, NULL,  
NULL, NULL, NULL)  
ORDER BY avg_fragmentation_in_percent DESC  
GO
```


Opcije za fragmentaciju

- **<= 30% fragmentation = Reorganize**

```
ALTER INDEX IX_Product_Name ON Production.Product  
REORGANIZE
```

- **> 30% fragmentation = Rebuild**

```
ALTER INDEX IX_Product_Name ON Production.Product  
REBUILD
```

Pitanja

