



Univerzitet u Nišu  
Elektronski fakultet  
Katedra za računarstvo



Aleksandar Stamenković

# **Interna statistika koju Microsoft SQL Server održava**

**Seminarski rad**

**Master akademske studije**

**Predmet: Sistemi za upravljanje bazama podataka**

Smer: Računarstvo i informatika

Modul: Bezbednost računarskih sistema

Student:

Aleksandar Stamenković

Br. indeksa: 1403

Niš, april 2022. god.

# Sadržaj

<b>1</b>	<b>Uvod .....</b>	<b>3</b>
<b>2</b>	<b>Kreiranje statistike .....</b>	<b>3</b>
2.1	Histogram u statistici .....	4
<b>3</b>	<b>Kreiranje tabele i popunjavanje podacima .....</b>	<b>4</b>
<b>4</b>	<b>Korišćenje SQL Server statistike.....</b>	<b>5</b>
<b>5</b>	<b>Automatsko ažuriranje statistike .....</b>	<b>7</b>
<b>6</b>	<b>Korisnički definisane statistike .....</b>	<b>9</b>
6.1	Kreiranje statistike .....	9
6.2	Modifikovanje statistike.....	9
6.3	Brisanje statistike .....	10
<b>7</b>	<b>Zaključak .....</b>	<b>10</b>
<b>8</b>	<b>Literatura.....</b>	<b>11</b>

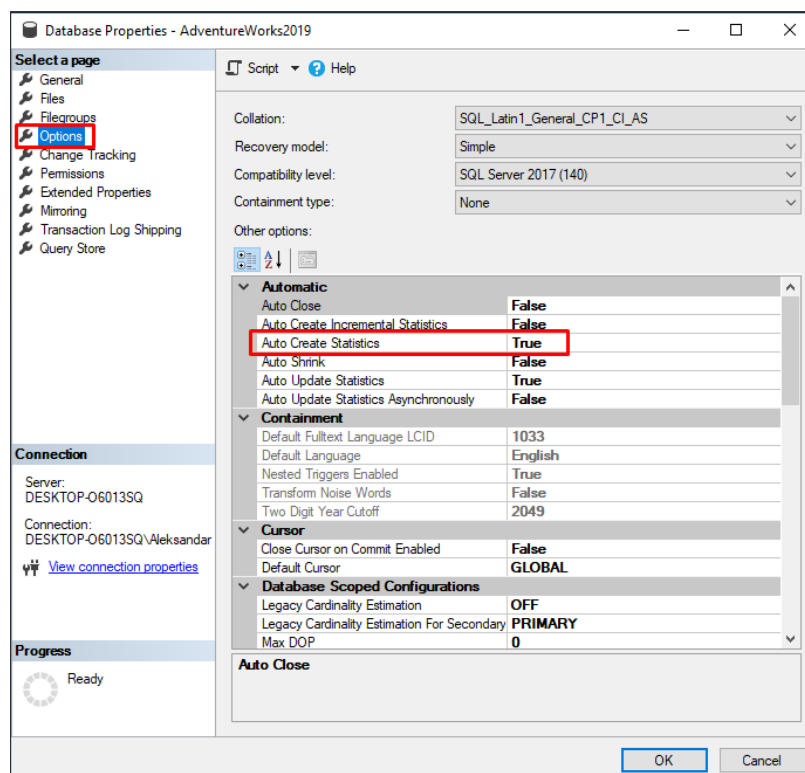
# 1 Uvod

Statistika koju Microsoft SQL Server održava je jedna od glavnih stvari za optimizatora upita koji se izvršava za vreme generisanja plana upita. Statistika se koristi od strane optimizatora da bi procenila koliko redova će upit vratiti pa na taj način izračunava cenu plana upita koristeći tu procenu. Zahtevi CPU, U/I i memorije se kreiraju na osnovu ovih procena, tako da precizna i ažurirana statistika igra važnu ulogu u kreiranju efikasnijih planova upita. Sem toga, preciznost statistike pomaže u odgovarajućim planiranjem resursa za upite. Tako da može se reći da statistika igra ključnu ulogu u performansama izvršenja upita.

Kao klijent preko koga će se vršiti konekcija na bazu, koristiće se SQL Server Management Studio, a baza koja će se koristiti u svrhu demonstracije je opšte poznata baza Adventure works koja se nalazi na sledećem linku: <https://docs.microsoft.com/en-us/sql/samples/adventureworks-install-configure?view=sql-server-ver15&tabs=ssms>.

## 2 Kreiranje statistike

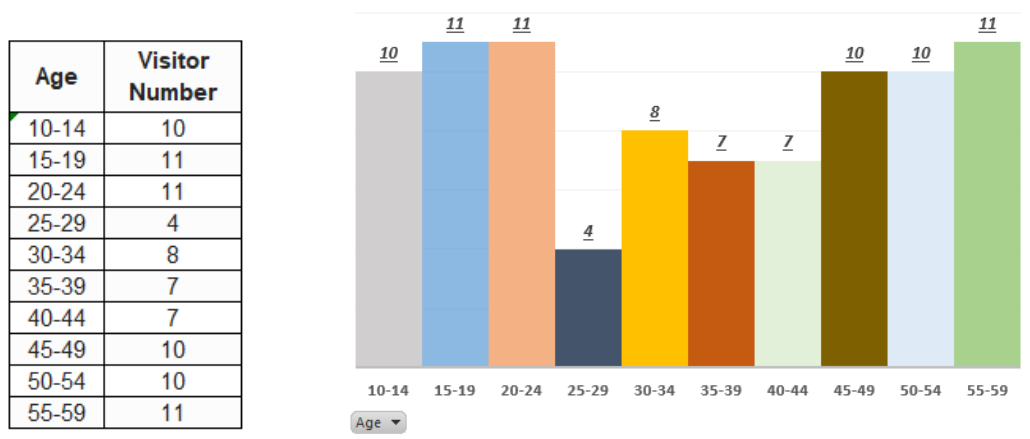
SQL Server automatski kreira statistiku kada se kreira indeks nad tabelom ili indeksirani pogled. Istovremeno, optimizator upita može odlučiti da li će da kreira statistiku za neindeksirane individualne kolone za vreme izvršenja upita. Kako god, ova opcija može da radi ukoliko omogućeno automatsko kreiranje statistike (eng. Auto Create Statistics). Opcija za automatsko kreiranje statistike se može naći pod tabom Options u svojstvima konkretne baze podataka, kao što je prikazano na sledećoj slici:



Slika 1 Automatsko kreiranje statistike

## 2.1 Histogram u statistici

Histogram je klasifikacija vrednosti u grupi podataka. Histogram omogućava da se jasno vide učestanosti distribucije skupa podataka. Na primer, na sledećoj slici je prikazan histogram koji je kreiran na osnovu raspodele godina starosti među ljudima koji posećuje neki muzej u jednom danu. Histogrami se generalno koriste radi vizualizacije podataka.



Slika 2 Histogram

SQL server statistika se skladišti kao distribucija vrednosti kolona u histogram i čuva jedinstvene vrednosti kao vektor gustine. Ova dva meta podatka koristi optimizator upita da bi izračunao koliko redova će upit vratiti. Razumevanje koncepta statistike SQL Servera će biti mnogo jasnije uz primere. Tako da, kao što je rečeno i u uvodu ovog rada, u nastavku će biti korišćena opšte poznata t+baza podataka Adventure works.

## 3 Kreiranje tabele i popunjavanje podacima

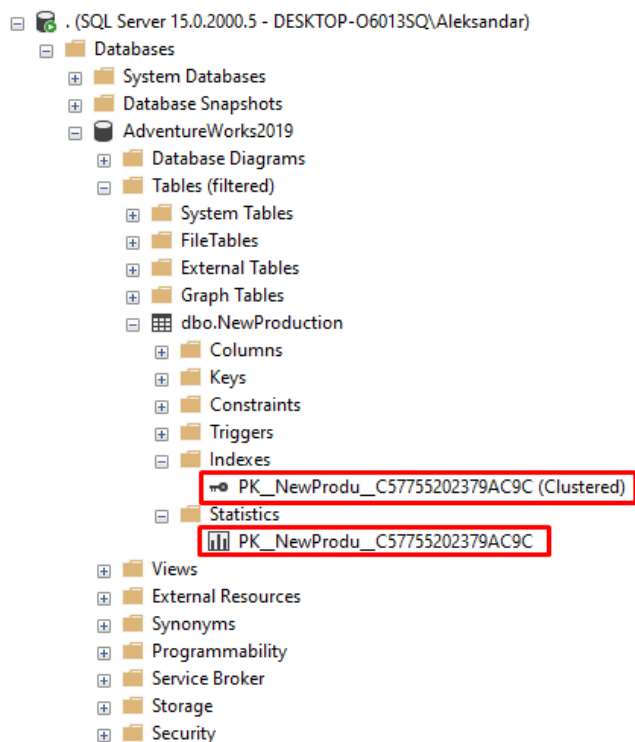
Konkretno će biti korišćena tabela Product iz ove baze podataka. Korišćićemo sledeći upit da bismo napravili novu tabelu NewProduction i napunili je podacima:

```
1 CREATE TABLE NewProduction
2 (
3     PID INT PRIMARY KEY IDENTITY(1,1),
4     Name VARCHAR(50),
5     ProductNumber VARCHAR(50),
6     SafetyStockLevel INT,
7     ReorderPoint INT
8 );
9
10 INSERT INTO NewProduction (Name, ProductNumber, SafetyStockLevel, ReorderPoint)
11 SELECT TOP 100 Name, ProductNumber, SafetyStockLevel, ReorderPoint
12 FROM Production.Product;
```

Slika 3 SQL upit za kreiranje tabele i njeno popunjavanje

## 4 Korišćenje SQL Server statistike

Kao što je već rečeno, nakon kreiranja indeksa, SQL Server automatski kreira statistiku za taj indeks. Na osnovu ovoga, možemo očekivati da SQL Server kreira statistiku za primarni ključ kreirane tabele. Podrazumevano, kada se kreira ograničenje primarnog ključa, SQL Sever automatski kreira jedinstveni klasterovani indeks. Koristeći SQL Server Management Studio, možemo videti SQL Server statistiku pod folderom sa tabelama (eng. Tables) kao što je prikazano na sledećoj slici:

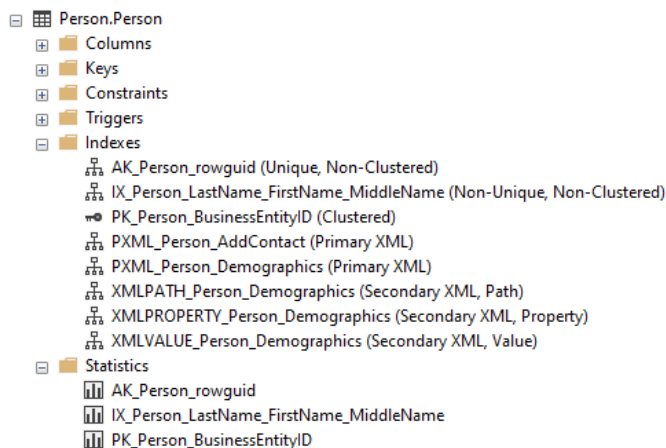


Slika 4 Indeksi i statistike tabele dbo.NewProduction

Komanda koja se koristi da bi se prikazala statistika je sledeća:

```
DBCC SHOW_STATISTICS ('Object_Name', 'Statistic_Name')
```

Pogledajmo najpre statistiku za već postojeću tabelu, recimo neka je to tabela Person.Person. Ona ima sledeće kreirane indekse i njihove odgovarajuće statistike:



Slika 5 Indeksi i statistike tabele Person.Person

Da bismo pogledali statistiku nad indeksom za ime, prezime i srednje ime iskoristićemo sledeću komandu:

```
DBCC SHOW_STATISTICS ('Person.Person', 'IX_Person_LastName_FirstName_MiddleName')
```

Nakon čega dobijamo sledeći izlaz:

Name	Updated	Rows	Rows Sampled	Steps	Density	Average key length	String Index	Filter Expression	Unfiltered Rows	Persisted Sample Percent
IX_Person_LastName_FirstName_MiddleName	Oct 27 2017 2:33PM	19972	19972	199	0.5505741	28,23934	YES	NULL	19972	0

All density	Average Length	Columns
0.0008291874	11,2126	LastName
5.124001E-05	23.02013	LastName, FirstName
5.05025E-05	24,23934	LastName, FirstName, MiddleName
5.00701E-05	28,23934	LastName, FirstName, MiddleName, BusinessEntityID

RANGE_HI_KEY	RANGE_ROWS	EQ_ROWS	DISTINCT_RANGE_ROWS	AVG_RANGE_ROWS
1 Abbas	0	1	0	1
2 Adams	10	86	6	1,666667
3 Alexander	28	123	15	1,866667
4 Allen	0	82	0	1
5 Alonso	1	93	1	1
6 Alvarez	5	99	4	1,25
7 Anand	2	74	2	1
8 Andersen	0	95	0	1
9 Anderson	0	85	0	1
10 Arthur	9	24	8	1,125
11 Arun	0	58	0	1
12 Ashe	0	30	0	1
13 Bailey	9	95	7	1,285714
14 Baker	0	84	0	1
15 Barnes	13	77	11	1,181818
16 Beck	17	36	13	1,307692
17 Bell	31	62	2	15,5
18 Bennett	4	68	3	1,333333
19 Black	33	35	20	1,65
20 Blanco	1	89	1	1
21 Blue	1	30	1	1
22 Bradley	12	30	10	1,2
23 Brooks	10	100	8	1,25
24 Brown	0	92	0	1
25 Bryant	23	120	9	2,555556
26 Butler	16	121	11	1,454545
27 Cai	3	64	3	1
28 Campbell	7	87	6	1,166667
29 Carlson	11	104	9	1,222222
30 Carter	23	76	9	2,555556

Slika 6 Izgled statistike

Izlaz koji se dobija se sastoji iz 3 dela, a to su redom s vrha naniže:

- Zaglavlje statistike (eng. Statistic header)
- Vektor gustine (eng. Density vector)
- Histogram

Ukoliko bismo želeli da pogledamo statistiku za našu novokreiranu tabelu dobili bismo prazan izlaz, kao što je to prikazano na slici ispod (iako tabela već ima podatke):

**Komanda:** `DBCC SHOW_STATISTICS ('dbo.NewProduction', 'PK__NewProdu__C57755202379AC9C')`

Name	Updated	Rows	Rows Sampled	Steps	Density	Average key length	String Index	Filter Expression	Unfiltered Rows	Persisted Sample Percent
PK__NewProdu__C57755202379AC9C	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

All density	Average Length	Columns

RANGE_HI_KEY	RANGE_ROWS	EQ_ROWS	DISTINCT_RANGE_ROWS	AVG_RANGE_ROWS

Slika 7 Prazna statistika

Statistički podaci će biti ažurirani kada se izvrši upit koji zahteva statističke podatke za klasterovani indeks. Primer takvog upita je:

```
SELECT * FROM NewProduction WHERE PID > 24
```

Dakle, dovoljno je u where klauzuli navesti kolonu koja predstavlja podatak nad kojim je kreirana statistika. Nakon ovoga možemo da pogledamo statistiku:

Results Messages

	Name	Updated	Rows	Rows Sampled	Steps	Density	Average key length	String Index	Filter Expression	Unfiltered Rows	Persisted Sample Percent
1	PK__NewProdu__C57755202379AC9C	Apr 26 2022 9:12PM	100	100	51	1	4	NO	NULL	100	0

	All density	Average Length	Columns
1	0.01	4	PID

	RANGE_HI_KEY	RANGE_ROWS	EQ_ROWS	DISTINCT_RANGE_ROWS	AVG_RANGE_ROWS
1	1	0	1	0	1
2	3	1	1	1	1
3	5	1	1	1	1
4	7	1	1	1	1
5	9	1	1	1	1
6	11	1	1	1	1
7	13	1	1	1	1
8	15	1	1	1	1
9	17	1	1	1	1
10	19	1	1	1	1
11	21	1	1	1	1
12	23	1	1	1	1
13	25	1	1	1	1
14	27	1	1	1	1
15	29	1	1	1	1
16	31	1	1	1	1
17	33	1	1	1	1

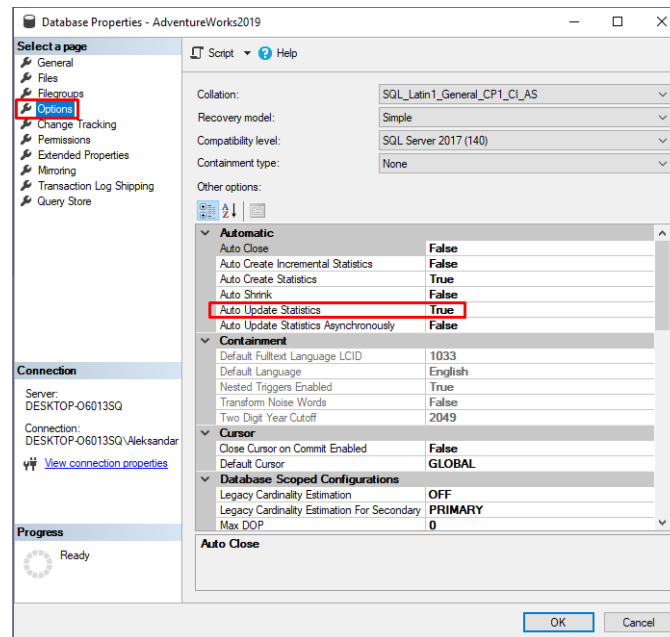
Slika 8 Statistika novokreirane tabele dbo.NewProduction

S druge strane, SQL Server statistika može postati zastarela nakon neke modifikacije (insert, update, delete) tabele zato što će se distribucija vrednosti kolona promeniti. U ovom slučaju, optimizator upita proverava brojač modifikacija i ukoliko on pređe vrednost praga, statistika će se ažurirati. Nakon ove operacije, optimizator generiše novi plan upita zbog toga što je moguće da se povećala šansa da će se generisati efikasniji plan upita sa ažuriranom statistikom. Optimizator upita odlučuje da li će ažurirati statistiku na osnovu sledećih pravila:

- Broj vrsta u tabeli je prešao sa 0 na broj veći od 0,
- Tabela je imala 500 ili manje redova kada je statistika poslednji put bila uzorkovana i od tada ima više od 500 modifikacija,
- Tabela je imala više od 500 redova kada se statistika poslednji put ažurirala i broj modifikacija redova je veći od  $\min(500 + \frac{n}{5}, \sqrt{1000n})$  nakon što je statistika poslednji put uzorkovana.

## 5 Automatsko ažuriranje statistike

Opcija automatskog ažuriranja statistike se može naći pod tabom Options u svojstvima baze podataka., kao što je prikazano na sledećoj slici:



Slika 9 Automatsko ažuriranje statistike

Sada ćemo da pokušamo da ubacimo 500.000 novih redova u našoj tabeli, a onda ćemo izvršiti isti upit. Tako da očekujemo da će se statistika ažurirati.

```
INSERT INTO NewProduction ( Name,ProductNumber,SafetyStockLevel,ReorderPoint)
SELECT Name,ProductNumber,SafetyStockLevel,ReorderPoint FROM Production.Product
GO 1000
SELECT * FROM NewProduction WHERE PID > 24
```

Kao što se sa sledeće slike može videti, statistika je ažurirana (u zaglavlju statistike se povećala vrednost za Rows i Rows Sampled):

Results Messages											
Name	Updated	Rows	Rows Sampled	Steps	Density	Average key length	String Index	Filter Expression	Unfiltered Rows	Persisted Sample Percent	
PK__NewProdu__C57755202379AC9C	Apr 26 2022 10:15PM	504100	171335	177	0.9931116	4	NO	NULL	504100	0	
All density	Average Length	Columns									
2.236866E-06	4	PID									
RANGE_HI_KEY	RANGE_ROWS	EQ_ROWS	DISTINCT_RANGE_ROWS	AVG_RANGE_ROWS							
1	0	1	0	1							
2	735	1260,115	4,685684	733	1,71912						
3	1406	1083,463	1	670	1,617109						
4	6341	3003,077	1	2982	1,006942						
5	7811	3008,966	1	1469	2,048309						
6	7814	8,832581	1	2	4,41629						
7	10196	3003,077	1	2381	1,261267						
8	14205	3011,91	1	2991	1,006924						
9	14208	5,888387	1	2	2,944193						
10	17497	3011,91	1	2991	1,006924						
11	20044	3011,91	1	2546	1,182997						
12	23622	3003,077	1	2982	1,006942						
13	25418	3011,91	1	1795	1,677944						
14	28563	3011,91	1	2991	1,006924						
15	36104	6026,764	1	5983	1,007248						
16	39394	3011,91	1	2991	1,006924						
17	39397	5,888387	1	2	2,944193						
18	42698	3011,91	1	2991	1,006924						
19	44758	3011,91	1	2059	1,462802						

Slika 10 Izgled statistike nakon dodavanja podataka



## 6 Korisnički definisane statistike

### 6.1 Kreiranje statistike

Osim što SQL Server sam može kreirati statistike, moguće je i definisanje statistika od strane korisnika. Postoji poseban SQL upit kojim se to može uraditi. Sintaksa je sledeća:

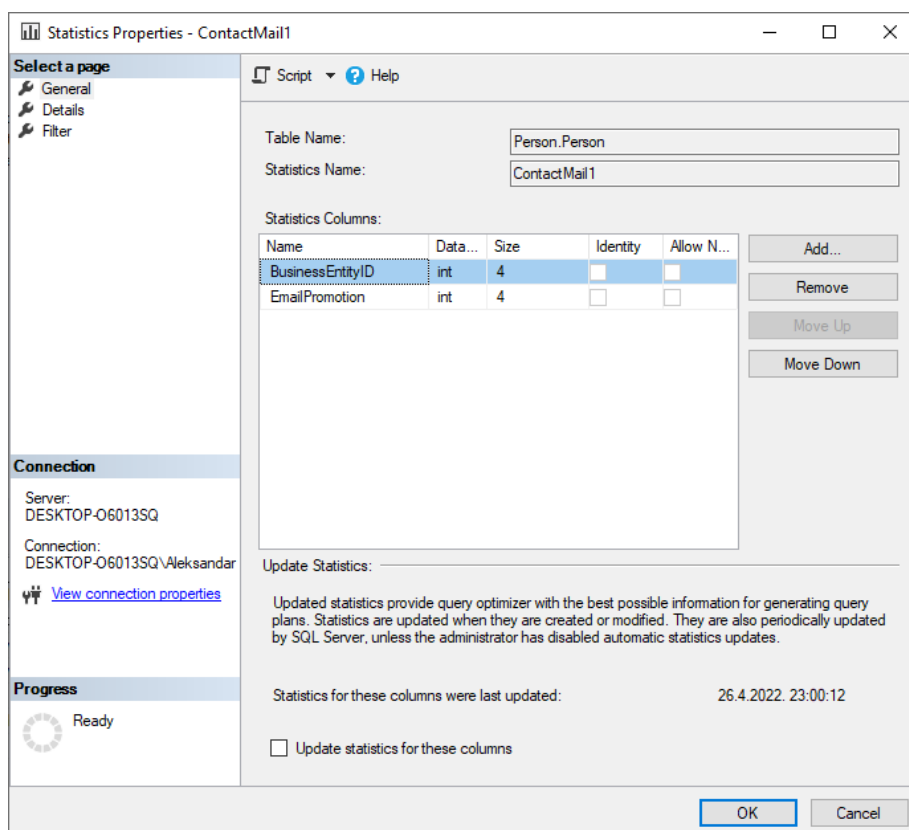
```
CREATE STATISTICS <ImeStatistike>  
ON Person.Person (<ImeKolone> [,<ImeKolone>])
```

Na primer, napravićemo statistiku nad tabelom Person.Person na sledeći način:

```
CREATE STATISTICS ContactMail  
ON Person.Person (BusinessEntityID, EmailPromotion)
```

### 6.2 Modifikovanje statistike

Da bi se modifikovala postojeća statistika neophodno je desnim klikom miša kliknuti na statistiku koja se želi modifikovati, pa zatim se klikne na Properties. Na taj način se otvara sledeći prozor gde je moguće izvršiti modifikaciju:



Slika 11 Modifikovanje statistike

Modifikacija statistie preko SQL-a nije moguća. Alternativni pristup je da se najpre obriše statistika koja se želi pomeniti i zatim se kreira nova.

## 6.3 Brisanje statistike

Sintaksa za brisanje postojeće statistike je sledeća:

```
DROP STATISTICS <ImeTabele>.<ImeStatistike> [,<ImeTabele>.<ImeStatistike>]
```

Primer brisanja kreirane statistike je:

```
DROP STATISTICS Person.Person.ContactMail1
```

## 7 Zaključak

Optimizator upita koristi statistiku da bi napravio plan upita (eng. query plan) koji će poboljšati performanse upita. Za najveći broj upita optimizator već generiše neophodnu statistiku za visoko kvalitetan plan upita. Međutim, u nekim slučajevima poželjno je kreiranje dodatne statistike ili izmenu upita da bi se dobili najbolji rezultati.

Statistika za optimizator predstavlja velike binarne objekte (eng. Binary large object – BLOB) koji sadrže informacije o distribuciji vrednosti u jednoj ili više kolona tabele ili indeksiranog pogleda. Optimizator koristi ovi statistiku da bi procenio kardinalnost (broj redova) u rezultatu upita. Kardinalnost omogućava optimizatoru da kreira upite visokog kvaliteta. Tako na primer, u zavisnosti od predikata koji se koriste u upitu, optimizator može iskoristiti kardinalnost i izabrati indeks seek operator umesto indeks scan operatora koji je zahtevniji u pogledu resursa.

## 8 Literatura

- [1] Microsoft Docs Statistics <https://docs.microsoft.com/en-us/sql/relational-databases/statistics/statistics?view=sql-server-ver15>
- [2] AdventureWorks sample databases <https://docs.microsoft.com/en-us/sql/samples/adventureworks-install-configure?view=sql-server-ver15&tabs=ssms>
- [3] Fundamentals of SQL Server Statistics <https://www.sqlshack.com/fundamentals-of-sql-server-statistics/>
- [4] Deep Dive into SQL Server Statistics <https://www.sqlshack.com/deep-dive-into-sql-server-statistics/>