

**Univerzitet u Nišu**

**Elektronski fakultet**

**Katedra za računarstvo**

**Seminarski rad**

**Steganografija slika**

**Predmet: Studijsko-istraživački rad**

|  |  |  |
| --- | --- | --- |
| **Mentor**: |  | **Student**: |
| Dr Aleksandar Dimitrijević |  | Miloš Veljanovski 1559 |

**Niš, novembar 2023.**

**Sadržaj**

[1. Uvod 3](#_Toc154654509)

[2. Steganografija slika 4](#_Toc154654510)

[2.1. Datoteke slika 4](#_Toc154654511)

[2.2. Opšti koncepti 4](#_Toc154654512)

[3. Istorija steganografije 5](#_Toc154654513)

[4. Taksonomija steganografskih tehnika 5](#_Toc154654514)

[4.1. Tehnika prostornog domena 6](#_Toc154654515)

[4.2. Opcija AUTO\_UPDATE\_STATISTICS 8](#_Toc154654516)

[4.3. Opcija AUTO\_UPDATE\_STATISTICS\_ASYNC 9](#_Toc154654517)

[4.4. Opcija AUTO\_DROP 11](#_Toc154654518)

[4.5. Opcija INCREMENTAL 11](#_Toc154654519)

[5. Korisnički definisane statistike 12](#_Toc154654520)

[5.1. Kreiranje statistike 12](#_Toc154654521)

[5.2. Ažuriranje statistike 14](#_Toc154654522)

[5.3. Brisanje statistike 16](#_Toc154654523)

[5.4. Opcije za pregled statistika kod SQL Servera 16](#_Toc154654524)

[5.4.1. Korišćenje SSMS-a za predleg statistika 16](#_Toc154654525)

[5.4.2. Pregled statistika korišćenjem T-SQL upita 18](#_Toc154654526)

[6. Literatura 20](#_Toc154654527)

[7. Listing 21](#_Toc154654528)

# 1. Uvod

U ovom modernom dobu, računari i internet su glavni mediji komunikacije koji povezuju različite delove sveta u jedan globalni virtuelni svet. Kao rezultat toga, ljudi mogu lako razmenjivati informacije i udaljenost više nije prepreka komunikaciji. Međutim, bezbednost komunikacije na velike udaljenosti ostaje pitanje od značaja. To je posebno važno u slučaju poverljivih podataka. Potreba za rešavanjem ovog problema dovela je do razvoja steganografije. Steganografija je moćno sredstvo zaštite koje pruža visok nivo bezbednosti, posebno kada se kombinuje sa enkripcijom.

Steganografija nije isto što i kriptografija. Cilj kriptografije je osigurati komunikaciju tako što se podaci pretvaraju u oblik koji prisluškivač ne može razumeti. Tehnike steganografije, s druge strane, imaju tendenciju da sakriju postojanje same poruke, što otežava posmatraču da otkrije gde se poruka nalazi. U nekim slučajevima, slanje šifrovanih informacija može privući pažnju, dok nevidljive informacije to neće. Prema tome, kriptografija nije najbolje rešenje za sigurnu komunikaciju; to je samo deo rešenja. Obe nauke mogu se koristiti zajedno kako bi se bolje zaštitile informacije. U ovom slučaju, čak i ako steganografija ne uspe, poruka se ne može otkriti jer se koristi i tehnika kriptografije.

Vodeni žigovi i otisci prstiju, među tehnologijama povezanim sa steganografijom, uglavnom se koriste za zaštitu intelektualne svojine. Digitalni vodeni žig je signal koji je trajno ugrađen u digitalne podatke (zvuk, slike, video i tekst) koji se kasnije može otkriti ili izvući kako bi se potvrdila autentičnost podataka. Vodeni žig je skriven u podacima nosača (hosta) na način da se ne može ukloniti bez promene samog nosača. Iako ovaj metod omogućava dostupnost podataka, oni su trajno obeleženi. Skrivena informacija u objektu sa vodenim žigom je potpis koji se odnosi na poreklo ili stvarnog vlasnika podataka kako bi se obezbedila zaštita autorskih prava. U slučaju otisaka prstiju, različiti i specifični znakovi su ugrađeni u kopije dela koje različiti korisnici treba da dobiju. U ovom slučaju, vlasniku intelektualne svojine postaje lako da identifikuje takve korisnike koji sebi daju za pravo da prekrše svoj ugovor o licenciranju kada nezakonito prenose svojinu drugim grupama. Pogledajte sliku 1, koja ilustruje vrste steganografije.

Slika

Slika 1. Optimizator upita

Performanse steganografskog sistema mogu se meriti pomoću nekoliko svojstava. Najvažnije svojstvo je statistička neuočljivost (neprimetnost) podataka, koje pokazuje koliko je teško odrediti postojanje skrivene poruke. Druge povezane mere uključuju steganografski kapacitet, koji predstavlja maksimalnu količinu informacija koje se mogu bezbedno ugraditi u delo bez statistički detektabilnih objekata, i otpornost, koja se odnosi na to koliko dobro steganografski sistem odoleva izvlačenju skrivenih podataka.

Gotovo svi digitalni formati datoteka, sa visokim stepenom redundantnosti, poznati su po tome što se koriste za steganografiju, a redundantni delovi se odnose na one delove koji su podložni promenama bez ikakve mogućnosti da se otkrije izmena. Slike i audio datoteke posebno dobro zadovoljavaju ovaj zahtev. Zapravo, digitalne slike su najčešće korišćeni formati datoteka nosača zbog njihove popularnosti na internetu. Postoji niz steganografskih tehnika koje omogućavaju skrivanje tajne poruke u datoteci slike, a sve imaju odgovarajuće prednosti i mane. Različite steganografske tehnike koriste se za različite namene. Moderna steganografija kategorizuje dve glavne klasifikacione šeme za taksonomiju algoritama. Prvi istaknuti algoritam se bazira na tipu datoteke. Druga je šema koja se široko koristi, gde se klasifikacija bazira na tehnici ugrađivanja, što je i glavni fokus ovog rada.

# 2. Steganografija slika

Kao što je ranije navedeno, slike se smatraju najpopularnijim formatima datoteka koji se koriste u steganografiji. Poznate su po tome što predstavljaju nekauzalni medijum, zbog mogućnosti proizvoljnog pristupa bilo kojem pikselu slike. Pored toga, skrivena informacija može ostati nevidljiva golom oku, zahvaljujući tehnikama steganografije koje koriste „rupe" u ljudskom vizuelnom sistemu.

## **2.1. Datoteke slika**

Slika se definiše kao raspored brojeva, pri čemu ti brojevi obično predstavljaju različite intenzitete svetlosti na različitim delovima slike. Numerički opis uzima oblik mreže (matrice) gde pojedinačne tačke nose naziv "pikseli". Pikseli se prikazuju horizontalno, red po red. U šemi boja, broj bitova poznat je kao dubina bita, i to se uglavnom odnosi na broj bitova dodeljen svakom pikselu. Štaviše, najmanja dubina bita u šemi boja je 8, tj. 8 bitova se koristi za predstavljanje boje svakog piksela. Čak i monohromatske i sivonijanse slike obično koriste 8 bitova za svaki piksel, i takvi bitovi su sposobni da prikažu do 256 različitih boja ili nijansi sive. Još jedan detalj koji treba dodati je da se digitalne slike u boji obično čuvaju u datotekama od 24 bita i koriste RGB model boja. Gotovo sve varijacije boja piksela na slici od 24 bita proizlaze iz tri osnovna termina boja: crvene, zelene i plave, i svaka od ovih boja je predstavljena sa 8 bitova. Dakle, u bilo kojem datom pikselu, broj različitih nijansi crvene, zelene i plave može dostići 256, što dovodi do više od 16 miliona kombinacija i na kraju rezultira sa više od 16 miliona boja. Najistaknutiji formati slika, posebno na internetu, jesu format grafike sa izmenom (GIF), format zajednice za stručnjake u oblasti fotografije (JPEG) i, u manjoj meri, format prenosivih mrežnih grafika (PNG). Važno je napomenuti da većina steganografskih tehnika pokušava iskoristiti strukturu ovih formata. Međutim, neki književni prilozi koriste format bitmape (BMP) zbog jednostavne i nekomplikovane strukture podataka.

## **2.2. Opšti koncepti**

* Poznato je da je kompresija bez gubitaka poželjnija kada originalni podaci treba da ostanu u celini. Na ovaj način, informacije o originalnoj slici nikada neće biti uklonjene, što omogućava rekonstrukciju originalnih podataka iz kompresovanih podataka. Ovo je tipično za slike u GIF i BMP formatu.
* Kompresija sa gubicima štedi prostor za skladištenje tako što odbacuje tačke koje su teške za ljudsko oko da identifikuje. U ovom slučaju, rezultirajuća slika se očekuje da bude slična originalnoj slici, ali ne ista kao original. JPEG kompresija koristi ovu tehniku. Početna slika je slika odabrana da nosi ugrađene bitove ili tajne informacije.
* Stego slika se odnosi na sliku koja nosi skrivenu poruku.
* Stego ključ je tajna informacija potrebna za dobijanje skrivene poruke iz stego slike.

# 3. Istorija steganografije

Kroz istoriju, ljudi su skrivali informacije na različite načine. Reč "steganografija" potiče od grčkih reči koje znače "pokriveno pisanje". Istraživači su je koristili hiljadama godina na različite načine. Tokom 5. veka pre nove ere, grčkog tiranina Histieja zarobio je kralj Darije u Suzi. Histiej je trebao da pošalje tajnu poruku svom zetu Aristagorasu, koji je bio u Miletu, i kako bi to postigao, Histiej je obrijao glavu roba i tetovirao poruku na njegovoj koži glave. Čim je kosa roba dovoljno porasla da bi sakrila tetovažu, poslat je u Miletu sa porukom. U antičkoj Grčkoj, još jedan metod bio je skidanje voska sa voskom prekrivene ploče, zatim pisanje poruke i ponovno nanošenje voska. Osoba koja je trebala primiti poruku samo bi trebala ukloniti vosak sa tablete da bi videla poruku. Nevidljivo mastilo je bila još jedna popularna forma steganografije. Antički Rimljani su pisali između redova koristeći nevidljivo mastilo i supstance poput voćnog soka, urina i mleka. Korišćenjem nevidljivog mastila, pismo bi moglo otkriti potpuno drugačiju poruku napisanu između redova. Nevidljivo mastilo se koristilo čak sve i do kraja Drugog svetskog rata.

Pored nevidljivog mastila, Nemci su tokom Drugog svetskog rata koristili tehniku "Mikrotačka". Informacije, posebno fotografije, bile su napravljene tako malim da su bile veoma teške za otkrivanje.

Godine 1550, Đirolamo Kardano, italijanski matematičar, predložio je šemu tajnog pisanja u kojoj se koristi maska od papira sa rupama. Korisnik takvih papira treba samo da napiše svoju tajnu poruku u tim rupama nakon što postavi masku preko praznog lista papira. Sledeći korak je ukloniti masku kako bi se popunili prazni delovi stranice, na taj način poruka izgleda kao bezopasan tekst.

Ova tehnika, steganografija, danas se široko koristi u digitalnim datotekama sa digitalnim podacima kao nosačem, a mreže se smatraju brzim kanalima za prenos. Odeljci koji slede ilustruju taksonomiju steganografskih tehnika za slikovne datoteke, uključujući pregled najvažnijih steganografskih tehnika za digitalne slike.

# 4. Taksonomija steganografskih tehnika

Postoji dosta pristupa za klasifikaciju steganografskih tehnika. Ovi pristupi se mogu klasifikovati u skladu sa vrstom nosilaca koji se koriste za skrivene komunikacije. Druga mogućnost je da se pristupi sortiraju u zavisnosti od vrste promene nosilaca već primenjene u procesu ugrađivanja. Drugi pristup je usvojen u ovom radu, iako u nekim slučajevima tačna klasifikacija nije moguća. Uopšteno, proces ugrađivanja se može definisati na sledeći način:

Neka označava nosioca, a stego-sliku. Neka predstavlja opcioni ključ (kao seme koje se koristi za šifrovanje poruke ili za generisanje pseudoslučajnog šuma, koji se može postaviti na radi jednostavnosti), a neka bude poruka koja se šalje. Tada predstavlja ugrađenu poruku, a predstavlja ekstrahovanu poruku. Dakle,

Da bi se razlikovale različite steganografske tehnike u širem smislu, moraju se uzeti u obzir kako metode koje menjaju sliku, tako i one koje menjaju format datoteke slike. Međutim, modifikacije formata datoteke su manje robusne. Važno je napomenuti da kompresija igra ključnu ulogu kada je reč o odlučivanju koja steganografska tehnika je bolja. Iako metodi sa kompresijom sa gubicima rezultiraju manjim veličinama datoteka slike, povećavaju mogućnost delimičnog gubitka ugrađene poruke jer se višak podataka slike mora eliminisati ovim tehnikama. Kompresija bez gubitaka ne kompresuje datoteku slike značajno. Kao rezultat toga, istraživači su razvili različite steganografske algoritme koji odgovaraju ovakvim vrstama kompresije. Steganografske tehnike koje menjaju datoteke slika kako bi sakrile informacije su sledeće:

* Tehnika prostornog domena
* Tehnike transformacije domena
* Tehnika širenja spektra
* Statističke metode
* Tehnike distorzije

Steganografske tehnike koje menjaju format datoteke slike uključuju ugrađivanje datoteke i ugrađivanje palete. Osim toga, postoje tehnike koje modifikuju elemente u vizuelnoj slici, a to su:

* Tehnika generisanja slike; i
* tehnika modifikacije elemenata slike.

Napokon, postoji posebna vrsta tehničkih i transformacijskih metoda prostornog domena, nazvana adaptivna steganografija, koju takođe opisujem radi potpunosti. Sledeća sekcija detaljnije objašnjava svaki steganografski pristup.

## **4.1. Tehnika prostornog domena**

Steganografske tehnike u prostornom domenu, takođe poznate kao tehnike zamene, čine grupu relativno jednostavnih tehnika koje stvaraju skriveni kanal u delovima nosača slike u kojima su promene vrlo male u poređenju sa ljudskim vizuelnim sistemom. Jedan od načina da se to postigne je skrivanje informacija u najmanje značajnom bitu (*LSB*) podataka slike. Ovaj način ugrađivanja se uglavnom zasniva na činjenici da se najmanje značajni bitovi u slici mogu smatrati slučajnom bukom, i stoga postaju neosetljivi na bilo kakve promene na slici.

Operacija ugrađivanja u LSB steganografiji opisana je sledećom jednačinom:

gde su , , i i-ti bit poruke, i-ta izabrana vrednost piksela pre i posle ugrađivanja, redom. Mnogi alati kao što su Steghide, S-tools, Steganos i drugi koji koriste steganografiju zasnovanu na LSB bitovima dostupni su na internetu.

Neka označava distribuciju najmanje značajnih bitova slike nosilaca, a distribuciju bitova tajne binarne poruke.

Poruka treba da se kompresuje ili šifruje pre nego što se ugradi, kako bi se zaštitila tajnost. Prema ovome, može se pretpostaviti da je distribucija poruke jednaka prosečnoj distribuciji, tako da je,

Osim toga, slika-nosilac i poruka takođe se mogu smatrati nezavisnim. Stoga, šum uveden u sliku može se modelovati kao:

Gde je stopa ugrađivanja, izražena u bitovima po pikselu (bpp). Gore opisani proces ugrađivanja jasno pokazuje u kojoj meri je moguće direktno izvući bitove tajne poruke iz najmanje značajnih bitova ovih piksela koji su već izabrani tokom ovog procesa.

Prilikom skrivanja bitova poruke u slici korišćenjem LSB algoritama, postoje dva sistema tj. šeme, i to sekvencijalni i rasuti. Najmanje značajni bitovi slike u sekvencijalnom sistemu zamenjuju se bitovima poruke, dok se u slučaju sistema sa rasutim ugrađivanjem bitovi poruke nasumično raspoređuju po celoj slici koristeći nasumični niz za kontrolu sekvence ugrađivanja.

Dobro poznati steganografski alati zasnovani na ugrađivanju LSB-ova različiti su u pogledu načina na koji sakrivaju informacije. Neki od njih nasumično menjaju LSB piksela, drugi modifikuju piksele ne na celoj slici već na odabranim delovima, a neki treći povećavaju ili smanjuju vrednost piksela LSB-a, umesto da menjaju vrednost.

Katzenbeisser i Petitcolas [21] opisuju nekoliko varijacija osnovnih LSB tehnika. Takođe opisuju tehniku zamene za ugrađivanje tajne poruke u najmanje značajne bitove palete GIF ili BMP formata slike koristeći steganografiju.

Bailey i Curran pružaju procenu različitih tehnika u vezi sa prostornom steganografijom, a takve tehnike se principijelno mogu primeniti na GIF slike [22]. Iz gore navedenog zaključujemo da su promene rezultirane upotrebom LSB tehnika na nosačkoj slici veoma teške za prepoznavanje ljudskim okom zbog svoje male veličine. Osim toga, takve tehnike su jednostavne i popularne. Nedostatak ove tehnike je što koristi svaki piksel u slici. Kao rezultat toga, ako se koristi kompresija sa gubicima, neke od skrivenih informacija mogu biti izgubljene [23].

## **4.2. Opcija AUTO\_UPDATE\_STATISTICS**

Kada je automatska opcija ažuriranja statistika, *AUTO\_UPDATE\_STATISTICS*, uključena, Optimizator upita utvrđuje kada bi statistike mogle biti zastarele, a zatim ih ažurira kada ih upit koristi. Ova akcija se takođe naziva rekompilacija statistike. Statistike postaju zastarele nakon što izmene operacija umetanja, ažuriranja, brisanja ili spajanja promene distribuciju podataka u tabeli ili indeksiranom pogledu. Optimizator upita utvrđuje kada statistike mogu biti zastarele brojanjem broja izmena redova od poslednjeg ažuriranja statistika i upoređivanjem broja izmena redova sa pragom. Prag se bazira na kardinalnosti tabele, koja se može definisati kao broj redova u tabeli ili indeksiranom pogledu.

Obeležavanje statistika kao zastarelih na osnovu izmena redova se dešava čak i kada je opcija *AUTO\_UPDATE\_STATISTICS* isključena. Kada je ova opcija isključena, statistike se ne ažuriraju, čak i kada su obeležene kao zastarele. Planovi će i dalje nastaviti da koriste zastarele statističke objekte. Postavljanje opcije *AUTO\_UPDATE\_STATISTICS* na isključeno može uzrokovati suboptimalne planove upita i smanjenje performansi upita, zato se uvek preporučuje postavljanje ove opcije na uključeno.

## **4.3. Opcija AUTO\_UPDATE\_STATISTICS\_ASYNC**

Opcija asinhronog ažuriranja statistika, *AUTO\_UPDATE\_STATISTICS\_ASYNC*, određuje da li optimizator upita koristi sinhrona ili asinhrona ažuriranja statistike. Podrazumevano, opcija asinhronog ažuriranja statistika je isključena, i optimizator upita ažurira statistike sinhrono. Ova opcija se primenjuje na statističke objekte kreirane za indekse, pojedinačne kolone u predikatima upita i statistike kreirane pomoću *CREATE STATISTICS* naredbe.

Da bismo postavili opciju asinhronog ažuriranja statistika u SQL Server Management Studiju, na stranici *Options* (opcije) prozora *Properties* (svojstva) baze podataka, obe opcije *Auto Update Statistics* (Automatsko ažuriranje statistika) i *Auto Update Statistics Asynchronously* (Asinhrono ažuriranje statistika) treba postaviti na *True* (tačno). Na sledećoj slici je prikazan izgled tog prozora.

A screenshot of a computer

Description automatically generated

Slika 7. Podešavanje statističkih opcija kroz dijalog svojstava baze podataka

Ažuriranja statistika mogu biti ili sinhrona (podrazumevano) ili asinhrona.

* Sa sinhronim ažuriranjem statistika, upiti se uvek kompajliraju i izvršavaju sa ažuriranim statistikama. Kada su statistike zastarele, optimizator upita čeka na ažurirane statistike pre nego što kompajlira i izvršava upit.
* Sa asinhronim ažuriranjem statistika, upiti se kompajliraju sa postojećim statistikama čak i ako su postojeće statistike zastarele. Optimizator upita može izabrati suboptimalni plan upita ako su statistike zastarele kada se upit kompajlira. Statistike se obično ažuriraju ubrzo nakon toga. Upiti koji se kompajliraju nakon što se ažuriranja statistika završe će imati koristi od korišćenja ažuriranih statistika.

Korišćenje sinhronih statistika treba razmotriti kada vršimo operacije koje menjaju raspodelu podataka, kao što je brisanje tabele ili izvođenje grupnog ažuriranja velikog procenta redova. Ako ne ažuriramo statistike ručno nakon završetka operacije, korišćenje sinhronih statistika će osigurati da su statistike ažurirane pre nego što se izvrše upiti nad promenjenim podacima.

S druge strane, treba razmotriti korišćenje asinhronih statistika kako bismo postigli predvidive vremene odziva upita za sledeće scenarije:

* Ako naša aplikacija često izvršava isti upit, slične upite ili slične keširane planove upita. Vremena odziva upita mogu biti predvidljivija sa asinhronim ažuriranjem statistika nego sa sinhronim ažuriranjem statistika, jer optimizator upita može izvršavati dolazne upite bez čekanja ažuriranih statistika. To izbegava odlaganje nekih upita.
* Ako je naša aplikacija doživela vremensko ograničenje čekanja zahteva klijenta koji su uzrokovani time što jedan ili više upita čeka na ažurirane statistike. U nekim slučajevima, čekanje na sinhrone statistikame može dovesti do neuspeha aplikacija sa agresivnim tajm-autima.

Asinhrono ažuriranje statistika se vrši putem pozadinskog zahteva. Kada je zahtev spreman da upiše ažurirane statistike u bazu podataka, pokušava da dobije bravu (*lock*) modifikacije šeme (*schema modification lock*) na objektu metapodataka statistike. Ako druga sesija već drži bravu (*lock*) na istom objektu, asinhrono ažuriranje statistika se blokira dok se brava modifikacije šeme ne dobije. Slično tome, sesije koje treba da steknu bravu stabilnosti šeme (*Sch-S*) na objektu metapodataka statistika kako bi kompajlirale upit mogu biti blokirane od strane pozadinske sesije asinhronog ažuriranja statistika, koja već drži ili čeka da stekne bravu modifikacije šeme. Zbog toga, za radne zadatke sa veoma čestim kompajliranjem upita i čestim ažuriranjem statistika, korišćenje asinhronih statistika može povećati verovatnoću problema sa konkurencijom usled blokiranja brava.

Međutim, od verzije 16.x SQL Servera 2022, mogući problemi sa konkurencijom pri korišćenju asinhronog ažuriranja statistika mogu biti izbegnuti uz pomoć opcije *ASYNC\_STATS\_UPDATE\_WAIT\_AT\_LOW\_PRIORITY* koja se postavlja na nivou baze podataka. Kada je ova opcija omogućena, pozadinski zahtev će čekati da stekne bravu modifikacije šeme (*Sch-M*) i sačuva ažuriranu statistiku u posebnom redu niskog prioriteta, omogućavajući drugim zahtevima da nastave kompajliranje upita sa postojećim statistikama. Kada nijedna druga sesija ne drži zaključavanje (bravu) na objektu metapodataka statistike, pozadinski zahtev će dobiti bravu modifikacije šeme i ažurirati statistike. U retkom slučaju da pozadinski zahtev ne može da stekne bravu u okviru vremenskog ograničenja od nekoliko minuta, asinhrono ažuriranje statistika će biti prekinuto, i statistika se neće ažurirati dok se ne pokrene sledeće automatsko ažuriranje statistika ili dok statistike ne budu ažurirane ručno.

## **4.4. Opcija AUTO\_DROP**

U SQL Serveru, pre verzije 16 (SQL Servera 2022), ukoliko su statistike ručno kreirane od strane korisnika ili nekog *third-party* alata na korisničkoj bazi podataka, te statistike mogu blokirati ili ometati šematske promene koje korisnik možda želi.

Počevši od SQL Servera 2022 (verzija 16.x), opcija za automatsko brisanje (*AUTO\_DROP*) je podrazumevano omogućena na svim novim i migriranim bazama podataka. *AUTO\_DROP* svojstvo omogućava kreiranje statističkih objekata u modu u kojem kasnije promene šeme neće biti blokirane od strane statističkog objekta, već će statistike biti obrisane po potrebi. Na ovaj način, ručno kreirane statistike sa omogućenim automatskim brisanjem se ponašaju kao auto-kreirane statistike.

Pokušaji postavljanja ili uklanjanja svojstva automatskog brisanja na auto-kreiranim statistikama mogu izazvati greške. Auto-kreirane statistike uvek koriste automatsko brisanje. Neke rezervne kopije (*backup*-ovi) baza, kada se vraćaju (*restore*-uju), mogu imati ovo svojstvo netačno postavljeno sve dok sledeći put ne budu ažurirane statistike (ručno ili automatski). S toga, prilikom vraćanja baze podataka na SQL Server 2022 (verzija 16.x) sa prethodne verzije, preporučuje se izvršavanje procedure *sp\_updatestats* na bazi podataka kako bi se postavili odgovarajući metapodaci za funkciju automatskog brisanja statistika.

## **4.5. Opcija INCREMENTAL**

Kada je opcija *INCREMENTAL* pri kreiranju statistika postavljena na *ON*, kreirane statistike su statistike po particijama. Kada je postavljena na *OFF*, statističko stablo se briše i SQL Server ponovo računa statistike. Ova opcija je podrazumevano isključena.

Kada se novi segmenti (particije) dodaju velikoj tabeli, statistike bi trebalo ažurirati da bi obuhvatile nove segmente. Međutim, vreme potrebno za skeniranje cele tabele (opcija *FULLSCAN* ili *SAMPLE*) može potrajati dosta dugo. Takođe, skeniranje cele tabele nije neophodno jer su možda potrebne samo statistike za nove segmente. Opcija za inkrementalno ažuriranje kreira i skladišti statistike na nivou svake particije i kada se ažuriraju, osvežava samo statistike na onim particijama koje zahtevaju nove statistike.

# 5. Korisnički definisane statistike

## **5.1. Kreiranje statistike**

Ponekad statistike automatski generisane od strane SQL Servera neće biti optimalne za vrste upita koje izvodimo, u kom slučaju možemo kreirati svoje sopstvene statistike. Microsoft preporučuje da kreiramo statistike u sledećim okolnostima:

* Ako *Database Engine Tuning Advisor* to preporučuje.
* Ako upit vraća samo deo podataka iz ciljane kolone ili indeksa.
* Ako se uslov upita odnosi na više povezanih kolona u odvojenim indeksima.
* Ako optimizator upita nema potrebne statistike da bi generisao efikasan plan upita.

Na primer, pretpostavimo da želimo da postavimo upit za podatke u tabeli *AWSales* na osnovu vrednosti u koloni *LineTotal*, ali naši upiti uglavnom ciljaju samo redove sa vrednošću *LineTotal* većom od 100 dolara. Možemo kreirati filtrirani statistički objekat na osnovu tih vrednosti, kao što je prikazano u sledećem primeru:

IF EXISTS(SELECT name FROM sys.stats

WHERE name = N'TotalStats'

AND object\_id = OBJECT\_ID(N'dbo.AWSales'))

DROP STATISTICS dbo.AWSales.TotalStats;

GO

CREATE STATISTICS TotalStats ON dbo.AWSales(LineTotal)

WHERE LineTotal > 100

WITH FULLSCAN;

Primer 9. Upit za kreiranje statistike

Naredba *CREATE STATISTICS* omogućava kreiranje statističkog objekta na jednoj ili više kolona u tabeli ili indeksiranom pogledu. U ovom slučaju, ciljamo na kolonu *LineTotal* u tabeli *AWSales*, ali samo za vrednosti veće od 100, kako je naznačeno u *WHERE* klauzuli. Ovo predstavlja naš filter, pa se često koristi i termin "filtrirane statistike".

Naredba *CREATE STATISTICS* takođe uključuje klauzulu *WITH FULLSCAN*, koja specificira da bi svi redovi trebalo da budu pregledani prilikom kreiranja statistika. Alternativno, možemo specificirati da se određeni procenat podataka uzorkuje, umesto da se pregledaju svi redovi.

Ako bismo sada izvršili upit nad sistemskim pogledima *sys.stats* i *sys.stats\_columns*, koristeći istu *SELECT* naredbu kao pre, primetili bismo da je četvrti objekat statistika, *TotalStats*, dodat tabeli *AWSales*, kao što je prikazano u sledećoj tabeli koja predstavlja rezultate upita.

|  |  |  |  |
| --- | --- | --- | --- |
| StatsID | StatsName | StatsColID | ColumnName |
| 1 | PK\_\_AWSales\_\_C3905BAF9C6C0B09 | 1 | OrderID |
| 2 | \_WA\_Sys\_00000004\_39AD8A7F | 1 | ProductID |
| 3 | ixOrders | 1 | SalesOrderID |
| 3 | ixOrders | 2 | OrderDetailID |
| 4 | TotalStats | 1 | LineTotal |

Takođe, sada možemo izvršiti *DBCC SHOW\_STATISTICS* naredbu nad novim statističkim objektom da bismo videli kako je histogram mapiran:

DBCC SHOW\_STATISTICS (AWSales, TotalStats);

Sledeća tabela prikazuje rezultate izvršavanja naredbe. Primetite da linija 7 (označena na slici histograma), ima vrednost za *RANGE\_HI\_KEY* od 120 i vrednost za *EQ\_ROWS* od 99.

A screenshot of a computer

Description automatically generated

Slika 8. Rezultat izvršavanja *DBCC SHOW STATISTICS* naredbe za novi statistički objekat

Sada ćemo ponovo pokrenuti upit za tabelu *AWSales*, ovog puta uključujući *WHERE* klauzulu koja specificira da vrednost *LineTotal* treba da bude jednaka 120:

SELECT \*

FROM AWSales

WHERE LineTotal = 120

OPTION(RECOMPILE);

Primer 10. Upit za analizu histograma za graničnu vrednost novog statističkog objekta

Ovaj upit vraća 99 redova, što je ista vrednost u koloni *EQ\_ROWS* histograma i ista vrednost kao vrednost procenjenog broja redova (*Estimated Number of Rows*)u planu izvršenja, kao što je prikazano na sledećoj slici.

A screenshot of a computer screen

Description automatically generated

Slika 9. Procenjeni plan izvršenja za novi statistički objekat

Naravno, možemo eksperimentisati sa svojim upitima i upoređivati procene plana izvršenja sa histogramom. Zapravo, ovo je dobar način za sticanje boljeg osećaja za to kako se planovi izvršenja odnose na statistike. Samo, treba imati na umu da neki od proračuna optimizatora mogu biti malo složeniji, sa manje očiglednom logikom kako je došao od tačke A do tačke B.

## **5.2. Ažuriranje statistike**

Kao i prilikom generisanja statistika, SQL Server obično dobro održava njihovu ažuriranost. Međutim, u nekim slučajevima, moraćemo da intervenišemo i ažuriramo statistike ručno. Microsoft preporučuje da to radimo u sledećim okolnostima:

* Ako se naši upiti izvode sporije nego što bismo očekivali.
* Ako ubacujemo podatke u rastuće ili opadajuće kolone koje predstavljaju ključeve, kao što su *IDENTITY* kolone.
* Ako vršimo određene operacije održavanja, kao što je brisanje svih podataka iz tabele ili izvođenje grupnog umetanja podataka.

Na primer, pretpostavimo da dodajemo još jedan blok podataka u tabelu *AWSales* koristeći sledeću *INSERT* naredbu:

INSERT INTO dbo.AWSales

(SalesOrderID, OrderDetailID, ProductID, LineTotal)

SELECT SalesOrderID, SalesOrderDetailID, ProductID, LineTotal

FROM AdventureWorks2019.Sales.SalesOrderDetail

WHERE SalesOrderDetailID >= 60000;

Primer 11. Upit za unošenje dodatnih podataka u AWSales tabelu

Ako bismo ponovo izvršili sledeću *DBCC SHOW\_STATISTICS* naredbu, dobili bismo iste rezultate kao kada smo je izvršili u prethodnom odeljku:

DBCC SHOW\_STATISTICS (AWSales, TotalStats);

Na mom sistemu, *Updated* kolona u rezultatima zaglavlja prikazuje vrednost za datum 20. avgust 2023. godine i vreme kao 20:53, što predstavlja vreme pre nego što sam izvršio prethodnu *INSERT* naredbu, što ukazuje na to da histogram nije promenjen, uprkos dodavanju preko 60.000 novih redova.

Da bismo osigurali da naši upiti imaju najnovije statistike, možemo izvršiti *UPDATE STATISTICS* naredbu nad tabelom *AWSales*:

UPDATE STATISTICS dbo.AWSales

WITH FULLSCAN;

Primer 12. Upit za ažuriranje statistike

Takođe sam uključio i *WITH FULLSCAN* klauzulu da bih SQL Serveru rekao da skenira celu kolonu u svakom statističkom objektu prilikom ažuriranja statistika. Prema podrazumevanoj postavci, SQL Server koristi samo uzorak podataka umesto da izračunava statistike na osnovu svih podataka. Puna analiza obično pruža tačnije statistike, ali može rezultirati operacijama koje zahtevaju više resursa.

Nakon ažuriranja statistika, možemo ponovo pokrenuti *DBCC SHOW\_STATISTICS* naredbu, koja daje rezultate prikazane na sledećoj slici.

A screenshot of a computer

Description automatically generated

Slika 10. Rezultat izvršavanja *DBCC SHOW STATISTICS* naredbe za ažurirani statistički objekat

Možemo primetiti da je vreme ažurirano i da su koraci histograma definisali nešto drugačije opsege. Za malen skup podataka, ove razlike možda neće delovati značajno, ali kada počnemo da radimo sa milionima redova, ažuriranje statistika nakon određenih operacija može biti ključno za održavanje performansi našeg upita.

## **5.3. Brisanje statistike**

Naravno, korisnički definisane statistike se mogu i obrisati. Da bismo obrisali statistički objekat u SQL Serveru, možemo koristiti naredbu *DROP STATISTICS*. Ova naredba uklanja statistički objekat povezan sa navedenom kolonom ili skupom kolona u tabeli ili indeksiranom pogledu. Osnovna T-SQL sintaksa je sledeća:

DROP STATISTICS <ImeTabele>.<ImeStatistike>

## **5.4. Opcije za pregled statistika kod SQL Servera**

Sve ove sistemske i korisnički definisane statistike kod SQL Servera za određenu tabelu možemo pregledati kako pomoću SQL Server Management Studio-a, tako i pomoću T-SQL upita.

### 5.4.1. Korišćenje SSMS-a za predleg statistika

Za ovaj pristup se podrazumeva da smo povezani na instancu SQL Servera kroz SSMS alat, nakon čega možemo odabrati konkretnu bazu i tabelu (u mom slučaju *AdventureWorks2019.AWSales*) i videti sve dostupne statističke objekte u okviru *Statistics* direktorijuma.

A screenshot of a computer

Description automatically generated

Slika 11. Pregled statističkih objekata tabele kroz SSMS

Odavde možemo dobiti detalje o bilo kojoj određenoj statistici iz liste, odabirom opcije za pregled svojstava (desnim klikom na statistiku biramo *Properties*). To otvara svojstva statistike i prikazuje statističke kolone i datum poslednjeg ažuriranja za određenu statistiku.

A screenshot of a computer

Description automatically generated

Slika 12. Pregled svojstava određenog statističkog objekta

Zatim, klikom na *Details* opciju iz menija možemo dobiti prikaz distribucije vrednosti i učestalost svakog pojavljivanja različite vrednosti (histogram) za navedeni objekat.

A screenshot of a computer

Description automatically generated

Slika 13. Pregled detalja određenog statističkog objekta

### 5.4.2. Pregled statistika korišćenjem T-SQL upita

Za ovaj pristup možemo koristiti dinamički upravljački pogled (*DMV*) *sys.dm\_db\_stats\_properties* kako bismo videli osobine statistika za određeni objekat u trenutnoj bazi podataka. Uz pomoć sledećeg upita proverili smo statistike za tabelu *AdventureWorks2019.AWSales*.

SELECT sp.stats\_id,

name,

filter\_definition,

last\_updated,

rows,

rows\_sampled,

steps,

unfiltered\_rows,

modification\_counter

FROM sys.stats AS stat

CROSS APPLY sys.dm\_db\_stats\_properties(stat.object\_id, stat.stats\_id) AS sp

WHERE stat.object\_id = OBJECT\_ID('dbo.AWSales')

Primer 13. Upit za prikaz liste statističkih objekata

Dati upit vraća očekivane rezultate:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| stats\_id | name | filter\_definition | last\_updated | rows | rows\_sampled | steps | unfiltered\_rows | modification\_counter |
| 1 | PK\_\_AWSales\_\_C3905BAF9C6C0B09 | NULL | 2023-08-20 21:47:16.6500000 | 121317 | 121317 | 3 | 121317 | 0 |
| 2 | \_WA\_Sys\_00000004\_39AD8A7F | NULL | 2023-08-20 21:47:16.6633333 | 121317 | 121317 | 200 | 121317 | 0 |
| 3 | ixOrders | NULL | 2023-08-20 21:47:16.7100000 | 121317 | 121317 | 146 | 121317 | 0 |
| 4 | TotalStats | ([LineTotal]>(100)) | 2023-08-20 21:47:16.7300000 | 63605 | 63605 | 200 | 121317 | 0 |
| 5 | \_WA\_Sys\_00000005\_39AD8A7F | NULL | 2023-08-20 21:47:16.7633333 | 121317 | 121317 | 200 | 121317 | 0 |

A evo i šta kolone iz rezultata predstavljaju:

* stats\_ID: Jedinstveni ID statističkog objekta.
* name: Ime statistike.
* last\_updated: Datum i vreme poslednjeg ažuriranja statistike.
* rows: Prikazuje ukupan broj redova u vreme poslednjeg ažuriranja statistike.
* rows\_sampled: Daje ukupan broj uzoraka redova za statistiku.
* unfiltered\_rows: Ova kolona predstavlja ukupan broj redova u tabeli za koje se prikupljaju statistike, bez uzimanja u obzir bilo kog uslova filtera ili predikata.
* modification\_counter: Predstavlja vrednost koja prati broj promena napravljenih na statističkom objektu od poslednjeg ažuriranja statistike.

# 6. Literatura

1. *TutorialsPoint. "MS SQL Server - Overview". [*[*Online*](https://www.tutorialspoint.com/ms_sql_server/ms_sql_server_quick_guide.htm)*], [Accessed: 12-Aug-2023]*
2. *Microsoft Learn. "Editions and supported features of SQL Server 2022". [*[*Online*](https://learn.microsoft.com/en-us/sql/sql-server/editions-and-components-of-sql-server-2022?view=sql-server-ver16)*], [Accessed: 12-Aug-2023]*
3. *Microsoft Learn. "Introduction to SQL Server 2022". [*[*Online*](https://learn.microsoft.com/en-us/training/modules/introduction-to-sql-server-2022/)*], [Accessed: 12-Aug-2023]*
4. *Microsoft Learn. "Query processing architecture guide". [*[*Online*](https://learn.microsoft.com/en-us/sql/relational-databases/query-processing-architecture-guide?view=sql-server-ver16)*], [Accessed: 12-Aug-2023]*
5. *Sheldon, Robert (2016). "SQL Server Statistics Basics". Redgate Hub, [*[*Online*](https://www.red-gate.com/simple-talk/databases/sql-server/performance-sql-server/sql-server-statistics-basics/)*], [Accessed: 15-Aug-2023]*
6. *Microsoft Learn (2023). "Statistics". [*[*Online*](https://learn.microsoft.com/en-us/sql/relational-databases/statistics/statistics?view=sql-server-ver16)*], [Accessed: 16-Aug-2023]*
7. *Eset, Erkec (2021). "Fundamentals of SQL Server Statistics". [*[*Online*](https://www.sqlshack.com/fundamentals-of-sql-server-statistics/)*], [Accessed: 16-Aug-2023]*
8. *Gupta, Rajendra (2019). "SQL Server Statistics and how to perform Update Statistics in SQL". [*[*Online*](https://www.sqlshack.com/sql-server-statistics-and-how-to-perform-update-statistics-in-sql/)*], [Accessed: 17-Aug-2023]*

# 7. Listing

Slika 1. Optimizator upita 4

Slika 2. Karakteristične vrednosti u histogramu 9

Slika 3. Rezultat izvršavanja *DBCC SHOW STATISTICS* naredbe 11

Slika 4. Procenjeni plan izvršenja za vrednost koja jeste svoj red 12

Slika 5. Procenjeni plan izvršenja za vrednost koja nije svoj red 13

Slika 6. Procenjeni plan izvršenja za opseg vrednosti 14

Slika 7. Podešavanje statističkih opcija kroz dijalog svojstava baze podataka 16

Slika 8. Rezultat izvršavanja *DBCC SHOW STATISTICS* naredbe za novi statistički objekat 20

Slika 9. Procenjeni plan izvršenja za novi statistički objekat 21

Slika 10. Rezultat izvršavanja *DBCC SHOW STATISTICS* naredbe za ažurirani statistički objekat 22

Slika 11. Pregled statističkih objekata tabele kroz SSMS 23

Slika 12. Pregled svojstava određenog statističkog objekta 24

Slika 13. Pregled detalja određenog statističkog objekta 25

Primer 1. Kreiranje AWSales tabele 5

Primer 2. Dobijanje informacija o statističkim objektima pomoću sistemskih pogleda 6

Primer 3. Uključivanje uslova u upit pomoću *WHERE* klauzule 7

Primer 4. Kreiranje indeksa u *AWSales* tabeli 8

Primer 5. Korišćenje *DBCC SHOW STATISTICS* naredbe 11

Primer 6. Upit za analizu histograma za graničnu vrednost 12

Primer 7. Upit za analizu histograma za ne-graničnu vrednost 13

Primer 8. Upit za analizu histograma za opseg vrednosti 14

Primer 9. Upit za kreiranje statistike 19

Primer 10. Upit za analizu histograma za graničnu vrednost novog statističkog objekta 20

Primer 11. Upit za unošenje dodatnih podataka u AWSales tabelu 22

Primer 12. Upit za ažuriranje statistike 22

Primer 13. Upit za prikaz liste statističkih objekata 25