

## **ADMINISTRACIJA BAZA PODATAKA**

### **1. Objasniti tok izvršenja SQL naredbe u okviru SUBP-a od renutka zahteva za isvršenjem (Click na Execute) do dobijanja rezultata SQL naredbe/Upita (po fazama i modulima SUBP-a)**

- (1) Formiranje kanoničkog stabla izvršenja upita
- (2) Perturbacija kanoničkog stabla izvršenja (radi unapređenja performanse izvršenja upita).
- (3) Odabir fizičke rutine (algoritma) izvršavanja operacija relacione algebre.
- (4) Formiranje različitih PLANOVA IZVRŠENJA
- (5) Izbor OPTIMALNOG PLANA IZVRŠENJA
- (6) Izvršavanje Plana izvršenja
- (7) Formiranje skupa rezultata (odgovora na upit).

### **2. Zadaci administratora baza podataka**

- (1) Dizajn baze podataka
- (2) Praćenje performansi i podešavanje
- (3) Obezbeđenje raspoloživosti baze podataka
- (4) Sigurnost baze podataka
- (5) Uzimanje rezervnih kopija baze podataka (Backup) i oporavak baze podataka (Restore)
- (6) Integritet podataka
- (7) Migracija podataka na druge verzije SUBP-a
- (8) Obezbeđuje proceduralnu funkcionalnost
  - i. Pogledi
  - ii. Uskladištene procedure
  - iii. Okidači (trigeri)
  - iv. Korisnički definisane funkcije
- (9) Administracija mobilnih uređaja

### **3. Pristupi u administraciji baza podataka**

- (1) Reaktivan (reaguje se kada se problemi pojave)
- (2) Proaktivan (predvideti probleme i reagovati pravovremeno)

### **4. SUBP – Odgovoran za**

- (1) Integritet podataka
- (2) Sigurnost podataka
- (3) Kontrolu pristupa
- (4) Optimizaciju
- (5) Automatsko sažimanje
- (6) Restartovanje i oporavljanje

### **5. Fizičko projektovanje BP**

Fizičko projektovanje je poslednja faza u procesu projektovanja baze podataka. Određuje se način na koji se baza podataka konkretno implementira na izabranoj SUBP platformi. Fizičko projektovanje je proces razvoja efikasno primenljive fizičke strukture baze podataka iz date logičke strukture u cilju zadovoljenja zahteva korisnika za informacijama. Projektovanje fizičke strukture baze podataka obuhvata aktivnosti koje se tiču organizaciji prostora namenjenog bazi podataka

### **6. Objasniti pojam fizičkog bloka, logičkog bloka i stranice baze podataka**

Fizički blok – osnovna jedinica skladištenja podataka na disku. Predstavlja segment prostora na fizičkom uređaju za skladištenje koji SUBP koristi za čuvanje podataka. Veličina varira, u rasponu od nekoliko kilobajta do nekoliko megabajta.

Logički blok – jedinica podataka koju SUBP koristi na logičkom nivou za organizaciju i upravljanje podacima. Može se sastojati od jednog ili više fizičkih blokova. Logički blokovi su konceptualni entiteti koje SUBP koristi za optimizaciju performansi i efikasnije upravljanje podacima.

Stranica baze podataka – osnovna jedinica podataka na nivou baze podataka koju SUBP koristi za čitanje i pisanje podataka. Može biti jednaka jednom ili više fizičkih blokova. Veličina stranice je obično fiksna (4KB, 8KB, 16KB, itd). SUBP čita i piše podatke o stranici, čime optimizuje pristup podacima i smanjuje učestalost IO operacija.

## **7. Karakteristike relacionih SUBP bitne za fizičko projektovanje**

Relacioni sistemi za upravljanje bazama podataka (SUBP) koriste neproceduralni upitni jezik koji omogućava korisnicima da lako dođu do traženih podataka. Oni obezbeđuju različite algoritme obrade podataka, uključujući implementaciju relacionih operacija, funkcija agregacije, grupisanja i eliminacije duplikata. Takođe, zadatak sistema je da izabere optimalan način izvršavanja upita uzimajući u obzir dostupne algoritme obrade i fizičku strukturu baze podataka.

## **8. Zahtevi pri fizičkom projektovanju**

Da bi se minimizirali troškovi U/I operacija u sistemu baze podataka važno je:

- (1) da strukture na sekundarnoj memoriji obezbeđuju pretraživanje relevantnih podataka preko efikasnih pristupnih putanja;
- (2) da su podaci organizovani i smešteni na sekundarnu memoriju na način koji minimizira U/I troškove pri pristupu podacima.

## **9. Karakterističan skup aplikacija**

- (1) Spisak upita i frekvencije njihovog izvršavanja;
- (2) Spisak ažuriranja i njihove frekvencije izvršavanja;
- (3) Ciljnu performansu za svaki tip upita i ažuriranja.

## **10. Prostor baza podataka (BP i datoteke BP, njihov odnos, pojam fajl grupe)**

Podaci su korisnicima dostupni preko tabela, ali se iza tih tabela nalaze datoteke odnosno Data setovi. Tokom procesa Fizičkog projektovanja ABP mora mapirati svaku tabelu na odgovarajuću fizičku strukturu koja će skladištiti podatke – table space, data space, file group... Svaki Tablespace (File Group) je prostor na disku rezervisan za skladištenje podataka u formi datoteka. Fizička organizacija baze podataka se sastoji od jednog ili više Tablespace-ova (File Group-a), a svaki Tablespace (File Group) sadrži podatke jedne ili više tabela. ABP, na osnovu korišćenja podataka, odlučuje kako će mapirati tabele na Tablespace-ove (File group-e) zavisno od tipova Tablespace-ova (File group-a) koje SUBP podržava i specifičnosti samog SUBP-a.

## **11. Log transakcija, Dnevnik ažuriranja (njihov odnos)**

Vrste žurnalizacije:

- (1) Before Image Žurnalizacija
- (2) After Image Žurnalizacija

Before image žurnalizacija se koristi u starijim SUBP sistemima:

- Kod SUBP zasnovanih na mrežnom modelu podataka i pre toga,

- Kod SUBP zasnovanih na hijerarhijskom modelu podataka.
- U log transakcija se zapisuju stanja stranica BP pre ažuriranja, dok se ažuriranje samih podataka obavlja direktno u BP.
- Za potrebe ROLLBACK-a, stranice se vraćaju u BP na stanje koje je zabeleženo pre ažuriranja, čime se poništavaju izvršena ažuriranja.
- DNEVNIK AŽURIRANJA se vodi u zasebnoj datoteci, koja je odvojena od loga transakcija.

After image žurnalizacija se danas praktikuje kod svih relacionih BP:

- Ažuriranja se beleže u toku transakcije.
- U slučaju potrebe za ROLLBACK-om, prostor u Logu Transakcija se označava slobodnim.
- Ako je transakcija uspešno završena, izdata je COMMIT naredba (bilo eksplicitno ili implicitno).
- BP se ažurira na osnovu sadržaja LOG-a transakcije koja je uspešno završena.
- Logovi uspešno izvršenih transakcija se povezuju i formiraju DNEVNIK Ažuriranja.
- Nema posebnog fajla za dnevnik ažuriranja.
- Prilikom uzimanja rezervne kopije BP (BACKUP-a), prostor LOG-a Transakcija je logički prazan, iako fizički prostor na disku ostaje zauzet.

## 12. Opcije pri fizičkom projektovanju BP

- Izbor formata datoteka: Definiše se broj datoteka BP, njihova veličina, veličina stranica datoteka, način grupisanja stranica u jedinice I/O prenosa, kao i veličina kontinualnih proširenja.
- Korišćenje pokazivača: Pokazivači omogućavaju navigaciju kroz BP i unapređuju performanse uparivanja skupova podataka, implementirajući se kao identifikatori n-torki (IDN-ovi).
- Vertikalna podela relacija: Smanjuje broj pristupa sekundarnoj memoriji grupisanjem obeležja iste relacije u različite datoteke BP.
- Kompresija podataka: Smanjuje potreban prostor na sekundarnoj memoriji i poboljšava performanse obrade.
- Asocijativno pretraživanje: Koristi se za smanjenje broja pristupa sekundarnoj memoriji. Najpoznatija struktura je B-stablo, a redosled n-torki u datoteci podataka može se kontrolisati organizacijom indeksa.
- Kontrolisano uvođenje redundantnih podataka: Opcije uključuju replikaciju za čuvanje istih delova baze podataka na više uređaja, kao i izvedene informacije koje se materijalizuju kao rezultat operacija spoja na sekundarnu memoriju.
- Join indeksi: U izvedene relacije smeštaju se samo neophodne informacije za efikasno izvršavanje operacija spoja, što eliminiše problem ažuriranja.
- Denormalizacija: Replikacija pojedinačnih vrednosti obeležja radi bržeg pristupa, posebno korisna za sisteme koji ne podržavaju grupisanje po različitim tipovima podataka.
- Način smeštanja i fizički redosled podataka: Organizacija elemenata podataka na sekundarnoj memoriji kako bi se maksimizirala količina relevantnih podataka čitanih jednom I/O operacijom, s ciljem smanjenja broja potrebnih I/O operacija za obradu BP.
- Razdvajanje relacija: Kod nekih SUBP, stranice datoteka BP mogu se rasporediti po više uređaja sekundarne memorije radi povećanja I/O propusnosti.

## 13. Indeksi

Indeksi su pomoćne strukture koje se kreiraju da bi se ubrzao pristupa podacima u Relacijama (Tabelama) prilikom pretraživanja. Kreira ih Administrator baze Podataka (ABP) preko SUBP-a. SUBP ih automatski održava posle njihovog kreiranja. Indeksni ključem u kontekstu baza podataka označavamo strukturu podataka koja se koristi za efikasno pretraživanje ili sortiranje podataka unutar baze. Indeksni ključ može biti prost i složen.

#### 14. Vrste indeksa

Sa aspekta grupisanja podataka postoji dve vrste indeksa:

- (1) Grupišući (Cluster) indeksi (Nad tabelom može biti samo jedan grupišući indeks.)
- (2) Negrupišući (Non Cluster) indeksi (Nad tabelom može biti više negrupišućih indeksa.)

Sa aspekta jedinstvenosti vrednosti indeksnog ključa postoji takođe dve vrste indeksa:

- (1) Jedinstveni indeksi (Nad jednom tabelom može biti definisano više jedinstvenih indeksa).
- (2) Nejedinstveni indeksi (Nad tabelom takođe može biti definisano više nejedinstvenih indeksa).

#### 15. Izbor indeksa – kriterijumi za izbor indeksa, alati za izbor

- (1) Kada indeksirati;
- (2) Izbor ključa pretraživanja;
- (3) Složeni ključ pretraživanja;
- (4) Grupisanje;
- (5) Heširanje ili stabla indeksa;
- (6) Uravnoteženje troškova održavanja indeksa;

#### 16. Optimizacija upita – Optimizator

Zadatak optimizatora upita:

- (1) određivanje alternativnih planova za izvršavanje upita;
- (2) estimacija troškova alternativnih planova i izbor plana sa najnižim troškovima;

Osnovne tehnike optimizacije:

- (1) Heuristička optimizacija - Heuristička optimizacija transformiše stablo upita korišćenjem skupa pravila koja tipično (ali ne u svim slučajevima) poboljšavaju performanse izvršenja.
- (2) Optimizacija estimacijom troškova izvršavanja upita - Optimizacija zasnovana na troškovima je "skupa", SUBP-ovi mogu koristiti heuristiku optimizaciju da smanje broj izbora koji se moraju napraviti estimacijom troškovima.
- (3) Semantička optimizacija - Proces transformacije upita koji je izdao korisnik u drugi upit koji, zbog semantike daje isti odgovor.

Heuristička optimizacija

- Zasniva se na transformacionim pravilima
- Polazi se od kanoničkog stabla upita
- Vrš se transformacija polaznog, kanoničkog, stabla upita u konačno stablo upita koje ima bolju performansu;
- Vrš se izbor pristupnih rutina i algoritama za operacije upita koje je moguće primeniti za raspoložive pristupne putanje baze podataka
- Osnovno pravilo je unarne operacije izvršavati pre binarnih

Optimizacija estimacijom troškova izvršavanja upita

- Vrš se procena i poređenje troškova različitih planova izvršavanja upita;
- Odabira se plan sa najnižim očekivanim troškovima;
- Obično se ograničava broj planova koji se pri optimizaciji razmatra da bi se trošilo manje vremena;
- Način optimizacije pogodan za kompilirane upite;
- Kod nekih SUBP se vrši "potpuna" optimizacija kompiliranih i "delimična" optimizacija interpretativnih upita, koja zahteva manje vremena;
- Funkcija troškova je aproksimativna, pa je moguć i izbor plana izvršenja koji nije optimalan;

## 17. Indeksi i ažuriranje baze podataka

- Neophodno je analizirati troškove pristupa podacima i troškove samog ažuriranja
- Ograničavajući faktor na broj pristupnih struktura je odnos smanjenja troškova koje neki indeks donosi i troškova njegovog održavanja;
- Troškovi ažuriranja n-torki i indeksa imaju važan uticaj na rezultat procesa selekcije indeksa i na odluke optimizatora;
- U zavisnosti od toga da li se objektima BP pristupa istim redosledom kojim su smešteni u bazu podataka ili ne primenjuju se različite formule za računanje troškova;

## 18. Šta je i čemu služi: Database Engine Tuning Advisor

Database Engine Tuning Advisor (DET) je alat koji se koristi u Microsoft SQL Serveru za optimizaciju performansi baza podataka. Osnovna svrha ovog alata je da preporuči indekse, statistike i fizičku strukturu baze podataka kako bi se poboljšao rad i efikasnost upita.

## 19. SQL Server – sistemske baze podataka

Posle instalacije SQL Servera na njemu se nalaze samo njegove sistemske baze podataka: master, model, msdb i tempdb. Praktično se rečnik podataka (system catalog) RSUBP MS SQL Servera sastoji iz te četiri baze.

Master baza podataka:

- Sadrži informacije potrebne samom SQL Serveru.
- U njoj se memorišu korisnički nalozi, svi parametri konfiguracije sistema.
- U master bazi podataka se memorišu i podaci o svim korisničkim bazama podataka, uključujući lokaciju datoteka svake baze podataka.
- U master bazu podataka se upisuju informacije za inicijalizaciju SQL Servera.
- SQL Server se ne može pokrenuti ako nije raspoloživa master baza podataka (uzeti u obzir prilikom uzimanja rezervnih kopija BP).
- Zbog toga je uvek potrebno imati raspoloživu kopiju (backup) master baze podataka.

Tempdb baza podataka:

- Privremene tabele i privremene uskladištene procedure
- Koristi se kada je potreban prostor za privremene tabele, međurezultate i radne tabele koje generiše sam SQL Server.
- Tempdb se ponovo kreira svaki put kad se startuje SQL Server, tako da se sistem startuje s praznom tempdb bazom podataka.
- Tokom rada se vrši auto-proširenje tempdb baze podataka kad je potrebno više prostora u toku rada SQL Servera, ali se za razliku od ostalih baza podataka reinicijalizuje na početnu veličinu kad god se SQL Server ponovo startuje.
- Ako je inicijalno definisan prostor za tempdb mali, sistem će u toku rada deo vremena trošiti na povećanje prostora tempdb baze podataka koji je potreban za podršku obimu posla koji SQL Server izvršava.
- Da bi se izbeglo trošenje vremena Administrator Baze Podataka (ABP) može koristeći naredbu ALTER DATABASE povećati prostor tempdb sistemske baze podataka

Model baza podataka:

- Modele (template) za sve baze podataka koje se kreiraju na sistemu.
- Kada se izda CREATE DATABASE naredba, prvi deo nove baze podataka koja se kreira se kopiranjem sadržaja model sistemske baze podataka.
- Ostatak nove baze podataka se popunjava praznim stranicama.

- Zbog toga što se tempdb sistemska baza podataka ponovo kreira svaki put kad se startuje SQL Server, model sistemska baza mora uvek postojati na SQL Serveru
- Svi korisnički definisani objekti u model bazi podataka se kopiraju u svaku novu bazu podataka kada se ona kreira.

Msdb baza podataka:

- od strane Agenata SQL Servera (njegove komponente se pokreću kao servisi operativnog sistema) za upravljanje upozorenjima, poslovima i za memorisanje (čuvanje) operatora

## 20. Information Schema

Definicija Information Schema-e:

- Information Schema je standardizovan skup sistemskih pogleda u Relacionim SUBP-ovima.
- Omogućava pristup metapodacima o strukturi i sadržaju baze podataka.

Svrha Information Schema-e:

- Glavna svrha je standardizovan pristup rečniku različitih SUBP-ova.
- Pruža korisnicima mogućnost pregleda informacija o objektima baze podataka bez obzira na specifičnosti svakog SUBP-a.

Strukturne Komponente koje se Mogu Pregledati:

- Pogledi poput Tables, Columns, Views, Indexes, Constraints pružaju detaljne informacije o strukturi baze podataka.
- Podaci uključuju katalog, šemu, naziv objekta i tip objekta.

Funkcionalnosti i Ograničenja:

- Information Schema omogućava samo pregled metapodataka, a ne i njihovu izmenu putem upita.
- Skriva implementacione detalje SUBP-ova, osiguravajući konzistentan pristup sistemskim informacijama.

Podrška i Rasprostranjenost:

- Podržana je u većini modernih Relacionih SUBP-ova kao što su SQL Server, MySQL, PostgreSQL i drugi.
- Omogućava administraciji baza podataka da efikasno upravlja strukturom i analizira kompleksna okruženja.

## 21. Transakcija nad bazom podataka

- Transakcija je osnovna, nedeljiva jedinica rada nad bazom podataka koja u celini ne remeti uslove integriteta.
- Transformiše jedno konzistentno stanje baze podataka u drugo, takođe konzistentno stanje.
- Dozvoljeno je da baza bude u nekonzistentnom stanju između dva konzistentna stanja.
- Ključno je da transakcija ne ostavi bazu trajno nekonzistentnom.
- Ako se transakcija ne može uspešno izvršiti (npr. zbog greške hardvera ili softvera), treba obezbediti mehanizam za poništavanje svih njenih dejstava i vraćanje baze u stanje pre početka transakcije.
- SUBP koji podržavaju transakcije moraju garantovati da se, u slučaju neuspešne transakcije, sva ažuriranja ponište.
- Transakcije se odvijaju pod kontrolom administratora transakcija (transaction manager-a) unutar SUBP-a, sa naredbama COMMIT i ROLLBACK.
- Mogu biti izvršene ručno ili automatski.

- Obuhvataju promene poput inserta, updata ili brisanja vrednosti iz baze.
- Upravljanje transakcijama je ključno za obezbeđivanje integriteta podataka i upravljanje greškama.
- Više SQL upita se grupiše i izvršava kao jedna logička nedeljiva celina, odnosno transakcija.

## 22. Paralelno izvršavanje transakcija

- SUBP treba da obezbedi adekvatno upravljanje paralelnim izvršavanjem transakcija kako one nepoželjno mešale svoja dejstva jedna na drugu.
- U osnovi postoje tri slučaja kada je transakcija samostalno korektna, ali zbog mešanja transakcija može nastati pogrešan rezultat:
  - Izgubljena ažuriranja
  - Zavisnost od privremenih ažuriranja
  - Narušavanje serijabilnosti

## 23. Problem izgubljenog ažuriranja

Problem izgubljenog ažuriranja (lost update problem) nastaje kada dve ili više transakcija čitaju istu vrednost podataka, izvrše neku promenu nad tom vrednošću i zatim je zapisuju nazad u bazu podataka. U ovom scenariju, poslednja transakcija koja zapiše svoju vrednost nadjača prethodne promene, čime se gube svi prethodni ažurirani podaci. Ovaj problem može dovesti do gubitka konzistentnosti podataka i neželjenih efekata na integritet podataka u sistemu.

## 24. Problem zavisnosti od privremenih ažuriranja

Problem zavisnosti od privremenih ažuriranja (temporary update dependency problem) se javlja kada jedna transakcija privremeno ažurira podatak, ali to ažuriranje nije trajno zbog rollback-a transakcije. Druga transakcija može da pročita ažuriranu vrednost pre nego što se rollback dešava, što može dovesti do nekonzistentnog stanja. Kada se prva transakcija rollback-uje, druga transakcija može zasnovati svoje odluke na privremenoj vrednosti koja više ne postoji, što može uzrokovati probleme u integritetu podataka i tačnosti rezultata transakcija.

## 25. Osobine transakcija (ACID)

- **Atomarnost (Atomicity)** u transakcijama osigurava da se sve operacije u transakciji izvrše u potpunosti ili se nijedna ne izvrši.
- Ako dođe do greške u bilo kojoj operaciji transakcije, sve promene se automatski poništavaju (rollback).
- Bitno je da ne postoji stanje baze podataka u kojem su delovi transakcije parcijalno primenjeni, što bi moglo dovesti do nekonzistentnog stanja.
- Transakcija će ili biti izvršena u celosti ili će se sva njena dejstva poništiti, što osigurava konzistentnost podataka.
- **Konzistentnost (Consistency)** – obezbeđuje da će baza pravilno promeniti stanje nakon uspešno izvršene transakcije.
- Nijedna transakcija ne sme ostaviti negativan efekat na podatke koji se nalaze u bazi podataka.
- Ako je baza bila u konzistentnom stanju pre izvršene transakcije, mora ostati u konzistentnom stanju i nakon izvršene transakcije.
- **Izolacija (Isolation)** – obezbeđuje da nijedna transakcija neće ugroziti izvršavanje neke druge transakcije.
- U sistemu baza podataka, gde se jedna ili više transakcija izvršava redom, ili paralelno, izolacija omogućava da svaka transakcija bude izvršena kao da je jedina transakcija u sistemu.
- **Trajnost (Durability)** – obezbeđuje da će rezultat izvršene transakcije postojati i u slučaju otkaza sistema.
- Baza podataka bi trebalo da bude trajna toliko da zadrži i poslednje napravljene izmene čak i u slučaju otkaza sistema ili ponovnog pokretanja.
- Ako transakcija promeni deo podataka u bazi i ako je izvršena u potpunosti, promene nad bazom podataka će biti trajne čak id da neposredno posle toga dođe do "ispada" bilo kog dela sistema.

- U slučaju da transakcija komituje, ali da baza otkáže pre nego što se podaci upišu na disk, podaci će biti ažurirani čim se sistem vrati u operativno stanje.

## 26. Protokoli zaključavanja

Proveru serijabilnosti i preduzimanje odgovarajućih akcija planer izvršenja teško može da obavi u realnom vremenu. Zbog toga se serijabilnosti izvršavanje skupa transakcija najčešće ostvaruju forsirano primenjujući mehanizam zaključavanja.

Ekskluzivno zaključavanje - ako neka transakcija postavi ekskluzivni lokot na objekat baze podataka, nijedna druga transakcija ne može na taj objekat da postavi bilo koji drugi lokot.

Deljivo zaključavanje - ako neka transakcija postavi deljivi lokot na objekat baze podataka, neka druga transakcija takođe može da postavi deljivi lokot na isti objekat baze, ali nijedna druga ne može da postavi ekskluzivni lokot na taj objekat.

## 27. Restauracija konzistentnog stanja baze podataka

Restauracija konzistentnog stanja baze podataka podrazumeva vraćanje baze podataka u stanje u kojem su svi podaci ispravni i međusobno usklađeni nakon neočekivanih grešaka ili prekida rada. Ovaj proces uključuje upotrebu bekap kopija i log fajlova kako bi se baza podataka vratila u prethodno poznato dobro stanje, obnavljajući sve transakcije koje su bile u toku ili nedovršene do trenutka greške. Cilj je osigurati integritet i tačnost podataka.

## 28. Pogledi

Pogled je „prozor“ kroz koji se vide podaci baze podataka. Pojednostavljuje korišćenje baze podataka, a može se koristiti za zaštitu od neovlašćenog pristupa. Sa aspekta performansi, pogledi se čuvaju u kompiliranom obliku. Pogled se može posmatrati kao podšema neke šeme relacione baze podataka. Koriste se kao bazne tabele a korisnik ne zna da li koristi baznu tabelu ili pogled (odnosno izvedenu tabelu).

## 29. Information Schema

INFORMATION\_SCHEMA- je skup sistemskih pogleda. Nalazi se u većini modernih Relacionih SUBP. Ona omogućava korisnicima da pristupe metapodacima o strukturi i sadržaju baze podataka. Koristi za prikazivanje informacija o samoj bazi i objektima baze podataka kao što su tabele, pogledi, ograničenja, funkcije, indeksi itd. Obezbeđuje se da se, skrivajući implementacione detalje same Sistemske baze konkretnog SUBP-a, rečniku pristupa na potpuno isti funkcionalan način. Pristup sistemskim podacima preko pogleda Information Schema je potpuno isti kod svih SUBP-ovakoji je podržavaju.

## 30. Korisničke funkcije

Kao i funkcije u programskim jezicima, SQL Server korisnički definisane funkcije su rutine koje prihvataju parametre, izvršavaju različite iskaze i vraćaju rezultate tih iskaza kao vrednosti.

- Povratna vrednost može biti jedna skalarna vrednost ili skup rezultata.
- Postoje sistemske i korisnički definisane funkcije baze podataka.
- Sistemske funkcije su uskladištene u master bazi i ne mogu se menjati.
- Kao primeri sistemskih funkcija mogu se navesti funkcije COUNT, SUM, ROUND, GETDATE,...

Bilo da se radi o sistemskim ili korisničkim definisanim funkcijama, funkcije se mogu podeliti na:



- Skalarne - korisnički definisane skalarne funkcije vraćaju jednu vrednost podataka kao rezultat funkcije. Ukoliko telo funkcije sadrži više od jednog iskaza isti se moraju navesti kao blok iskaza. Povratna vrednost može biti bilo koji tip podataka, osim teksta, nteksta, slika, kursora i vremenskih tipova podataka.
- Table valued funkcije – predstavljaju korisnički definisane funkcije koje vraćaju tabele kao rezultat rada funkcije.

### 31. Uskladištene procedure

Uskladištene procedure se mogu definisati kao kolekcija Transact – SQL iskaza koje mogu da vrate određene rezultate na osnovu prosledjenih parametara od strane korisnika.

- Uskladištene procedure se prevashodno koriste za neke obrade koje najčešće podrazumevaju čitanje podataka iz jednog dela BP, izračunavanja i ažuriranje sadržaja nekog drugog dela BP (obračun kamata na bankarske račune, obračun zarada,...).
- Ovi postupci obrade se mogu vršiti i pomoću programa pisanim u nekom od programskih jezika koji se izvršavaju na klijentima ali je prednost korišćenja uskladištenih procedura u tome što se one u nekim situacijama obrade (ali ne svim) mnogo brže izvršavaju.

Kod procedura baza podataka postoje dve vrste parametara:

- Ulazni (input) i
- Izlazni (output) parametri.

Ulazni parametri su korisnički prosledjeni parametri koji služe za zadavanje uslova u samom telu procedure. Pomoću njih zadaju se određeni uslovi koji će izdvojiti n-torke za obradu. Izlazni parametri su parametri koji se vraćaju izvršavanjem tela procedure. Oni predstavljaju prikaz rezultata obrade podataka pomoću uskladištenih procedura. Postoje dve osnovne vrste procedura: sistemske procedure i korisnički definisane procedure baze podataka.

Sistemske procedure:

- U SQL Serveru, mnoge administrativne aktivnosti mogu se obavljati pomoću sistemskih procedura.
- Svaki put kada se doda ili izmeni tabela, napravi rezervna kopija baze podataka (backup), kreira login ili korisnik, dodeljuju korisnička prava ili obavljaju druge administratorske aktivnosti, zapravo se pozivaju sistemske procedure specijalno napisane da izvrše željenu aktivnost.
- Sistemske procedure imaju prefiks sp\_ i čine deo SQL Server instalacije. Čuvaju se u master bazi podataka i dostupne su za korišćenje u bilo kojoj bazi podataka konkretne instance SQL Servera.

Korisnički definisane procedure:

- Korisnički definisana procedura je svaki program napisan od strane Administratora baze podataka u okruženju SUBP-a uz korišćenje SQL naredbi I njegovih proceduralnih proširenja.
- Najčešći zadatak korisničkih Uskladištenih procedura je ažuriranje sadržaja jednih tabela na osnovu obrađenog sadržaja drugih tabela baze podataka.
- Za razliku od sistemskih procedura, korisnički definisane procedure se čuvaju u lokalnoj bazi i samo su u njoj dostupne za korišćenje.

### 32. Cursor

Sistemi za upravljanje bazama podataka u svojim proceduralnim proširenjima SQL-a omogućavaju deklarisanje kursora ili kursorskih promenljivih koje obezbeđuju prihvatanje rezultata kada on sadrži više od jedne n-torke. Otvaranje kursora nad skupom rezultata omogućava obradu podataka red po red.

Rad sa cursorima obično podrazumeva sledeće korake:

- DECLARE: Deklaracija kursora i pridruživanje SELECT naredbe.
- OPEN: Izvršavanje SELECT naredbe i popunjavanje kursora rezultujućim n-torkama.
- FETCH: Pristupanje n-torkama u kursoru.
- Opcionalno: Modifikacija tekuće n-torke.
- CLOSE: Zatvaranje kursora.

#### Tipovi Kursora

- INSENSITIVE: Pravi privremenu kopiju podataka; izmene u baznim tabelama se ne reflektuju. Ne dozvoljava modifikacije.
- SCROLL: Omogućava sve FETCH opcije (FIRST, LAST, PRIOR, NEXT, RELATIVE, ABSOLUTE).

#### Mogućnosti FETCH naredbe

- NEXT: Pristupa svakoj sledećoj n-torci (default).
- PRIOR: Pristupa prethodnoj n-torci.
- FIRST: Pozicionira se na prvu n-torku.
- LAST: Pozicionira se na poslednju n-torku.

#### Dodatne Opcije

- READ ONLY: Onemogućava modifikacije podataka.
- UPDATE [OF column\_name[,...n]]: Definiše kolone koje se mogu modifikovati.

#### Zatvaranje i Uklanjanje Kursora

- CLOSE: Zatvara kursor i oslobađa memorijski prostor, ali zadržava deklaraciju.
- DEALLOCATE: Poništava deklaraciju kursora.

#### Sistemska Varijabla

@@FETCH\_STATUS: Vraća status poslednjeg FETCH iskaza.

0: Uspešan FETCH.

-1: FETCH nije uspeo ili je n-torka van rezultujućeg skupa.

-2: Fetch-ovana n-torka ne postoji.

### 33. Trigeri

Okidač (triger) je automatski pokretan mehanizam u SQL-u koji se aktivira pri unosu, modifikaciji ili brisanju podataka u tabeli. Koristi se za proveru integriteta podataka i obezbeđenje konzistentnosti baze podataka. Postoje dve glavne vrste okidača: AFTER (aktivira se posle uspešnog ažuriranja) i INSTEAD OF (izvršava se umesto ažuriranja). Okidači mogu koristiti logičke tabele inserted i deleted za rad sa starim i novim vrednostima podataka. Za rad sa okidačima koriste se DDL naredbe: CREATE TRIGGER, ALTER TRIGGER i DROP TRIGGER.

### 34. Sigurnost i zaštita baza podataka

Administracija sigurnosti je jedan od ključnih zadataka administracije baza podataka. Razumevanje mogućnosti zaštite koje neki SUBP podržava je ključan za obezbeđivanje integriteta kompletnog okruženja baze podataka. Dobra implementacija i rukovanje modelom zaštite pomaže u smanjenju ranjivosti osetljivih podataka u bazi, a povećavaju ukupnu skalabilnost i pouzdanost kompletnog okruženja baze podataka.

### 35. Koncepti sigurnosti baze podataka

Zaštita baze podataka od neovlašćenog korišćenja odvija se na više nivoa:

- Zaštita celog sistema;
- Zaštita na nivou mreže;

- Zaštita na nivou operativnog sistema;
- Zaštita na nivou aplikacije;
- Zaštita unutar same baze podataka.

### 36. Preporuke u zaštiti baze podataka

- Administracija zaštite baze podataka treba da bude deo procesa standardne administracije baze podataka.
- Koristiti princip najmanjih privilegija.
- Implementirati dubinsku zaštitu na više nivoa.
- Dozvoliti (enejblovati) samo potrebne servise i komponente.
- Redovno pratiti podešavanja zaštite.
- Obučavati korisnike o važnosti zaštite.
- Definisati zaštitne uloge u sistemu na osnovu poslovnih pravila koja važe u sistemu.

### 37. Načelo minimalnih ovlašćenja

Princip minimalnih ovlašćenja predviđa da korisnicima treba dodeliti samo minimum ovlašćenja potrebnih za obavljanje njihovih poslova nad bazom. To takođe predviđa:

- Upotrebu uloga, koje sadrže grupe (skup) ovlašćenja i olakšavaju administraciju;
- Upotrebu pogleda, koji ograničavaju pristup na definisane podskupove postojećih podataka;
- Upotrebu uskladištenih procedura čijom se upotrebom može izbeći dodela konkretnih prava nad baznim tabelama korisnicima.

### 38. Mehanizmi zaštite na nivou SUBP-a

- Identifikacija i dokazivanje autentičnosti (predstavljanje onoga ko pristupa sistemu);
- Autorizacija i kontrola pristupa (ko nešto može da uradi);
- Enkripcija podataka (određivanje ko podatke može videti);
- Praćenje pristupa podacima i zapisivanje (prećenje ko je šta i kada uradio):
  - Upis izvornog teksta korisničkog zahteva;
  - Upis identifikatora klijanta/terminala sa kojeg je pristupljeno;
  - Upis datuma i vremena pristupa bazi;
  - Upis relacije, n-torke i atributa kojima je pristupano;
  - Upis starih i novih vrednosti podataka.

### 39. Kontrola pristupa

- Kontrola pristupa je deo sigurnosne politike koja je prisutna u gotovo svakom sistemu.
- Kad se spominje kontrola pristupa, uzimaju se u obzir četiri situacije: sprečavanje pristupa, ograničavanje pristupa, dozvola pristupa, oduzimanje prava pristupa
- Korisnicima se dodeljuju ovlašćenja (eng. privileges) za povezivanje na bazu i rad sa njenim objektima.
- Ovlašćenja korisnicima može dodeljivati administrator baze, vlasnik objekata ili neki drugi autorizovani korisnik kome je dato to pravo.
- Ovlašćenja omogućavaju korisnicima da obavljaju određene akcije nad serverom baze i samom bazom podataka (sistemska ovlašćenja) ili objektima baze (objektni ovlašćenja).

### 40. Modeli kontrole pristupa

Modeli kontrole pristupa u bazama podataka dele se na modele zasnovane na mogućnostima i modele zasnovane na listama kontrole pristupa (ACL), koji definišu dozvole nad objektima. Prema metodama implementacije, postoje diskrecioni model (DAC), mandatorni model (MAC), i model grupa i uloga (RBAC). Diskrecioni model

omogućava vlasnicima podataka kontrolu pristupa, mandatorni model koristi striktno sigurnosne politike, dok model grupa i uloga upravlja pristupom putem definisanih uloga.

#### 41. Model kontrole pristupa zasnovan na ulogama

- Role Based Access Control (RBAC) – Način kontrole pristupa zasnovan na ulogama i pravima.
- U ovom sigurnosnom modelu, pristup resursima je baziran na ulozi korisnika koju je dobio od administratora.
- Administrator dodeljuje ulogu koja dolazi sa već određenim pravima i privilegijama, i korisnik može da izvršimo akcije dodeljene pravilima uloge.
- RBAC je novija alternativa DAC i MAC.
- RBAC model nudi mogućnost implementacije i MAC modela i DAC modela kontrole pristupa.

#### 42. SQL Server - režimi bezbednosti

- SQL Server poseduje dva načina na koje se korisnik (takođe i aplikacija ili Web sajt), može povezati.
  - SQL Server autentifikacija - Za povezivanje je potrebno uneti korisničko ime i šifru. Ova metoda se koristi u peer-to-peer mrežama bez centralizovane lokacije za korisničke račune, kao što je aktivni direktorijum.
  - Windows autentifikacija - Koristi se korisničko ime i šifra sa Windows prijave. Korisnik koji je prijavljen na Windows mrežu koristi iste kredencijale za SQL Server. Prava pristupa SQL Serveru mogu se dodeliti korisnicima ili grupama na lokalnom računaru ili u domenskoj mreži.
- Način autentifikacije se definiše pri instalaciji ali se može promeniti promenom parametara konfiguracije SQL Servera.

#### 43. Korisnik baze podataka (Database User)

Korisnik baze podataka (User) je principal na nivou baze, i svakom korisniku se dodeljuje public role. Pri kreiranju baze podataka automatski se kreira guest korisnik, čija prava koriste korisnici bez naloga u bazi. Guest korisnika nije moguće izbrisati, ali mu se mogu ukinuti prava konekcije naredbom REVOKE CONNECT FROM GUEST, osim u master ili tempdb bazama.

#### 44. Uloge na nivou servera (Server-Level Roles)

SQL Server koristi uloge (roles) za jednostavniju dodelu prava na nivou servera, slično grupama u Windows operativnom sistemu. Svaka serverska uloga ima specifična prava pristupa na nivou instance SQL Servera.

##### Glavne Server-Level Roles

- sysadmin: Potpuna kontrola nad serverom.
- serveradmin: Menjanje konfiguracija i zaustavljanje servera.
- securityadmin: Upravljanje login nalogima i njihovim pravima, uključujući GRANT, DENY, i REVOKE naredbe.
- processadmin: Upravljanje procesima na serveru.
- setupadmin: Dodavanje i uklanjanje povezanih servera.
- bulkadmin: Izvršavanje BULK INSERT naredbi.
- diskadmin: Upravljanje datotekama baze podataka.
- dbcreator: Kreiranje, izmena, brisanje i restore baza podataka.
- public: Osnovna prava dodeljena svakom login nalogu.

Svaki login nalog SQL Servera automatski pripada public ulozi, koja daje osnovna prava pristupa kada specifična prava nisu dodeljena ili zabranjena.

#### 45. Uloge na nivou baze podataka (Database-Level Roles)

Dve grupe uloga:

1. Fiksne uloge: Unapred definisane uloge u svakoj bazi podataka.
2. Promenljive uloge: Uloge koje kreira administrator baze.

Fiksne uloge:

- **db\_owner**: Potpuna kontrola nad bazom, uključujući brisanje baze.
- **db\_securityadmin**: Upravljanje korisnicima i njihovim pravima pristupa.
- **db\_accessadmin**: Dodavanje ili uklanjanje pristupa bazi za Windows i SQL Server login naloge.
- **db\_backupoperator**: Pravljenje rezervnih kopija baze.
- **db\_ddladmin**: Izvršavanje bilo koje DDL naredbe (kao što su CREATE, ALTER, DROP).
- **db\_datawriter**: Dodavanje, brisanje ili menjanje podataka u svim tabelama baze.
- **db\_datareader**: Čitanje podataka iz svih tabela u bazi.

Članovi uloga **db\_owner** i **db\_securityadmin** mogu dodeljivati fiksne uloge drugim korisnicima, ali samo **db\_owner** može dodavati članove u **db\_owner** ulogu.

#### 46. Pristupi koji BP mogu učiniti sigurnijom

Neki od pristupa koji bazu podataka mogu učiniti sigurnijom:

- Ograničiti pristup važnim resursima koji mogu biti pogrešno korišćeni – zlonamerno ili slučajno;
- Onemogućiti nepotrebne komponente i servise sistema za upravljanje bazom podataka;
- Ukloniti ili onemogućiti predefinisane korisničke naloge;
- Izvršavati procese baze podataka pod namenskim neprivilegovanim nalogima.

#### 47. Kontrola pristupa objektima baze podataka

Objektna ovlašćenja korisniku omogućavaju da izvrši operacije nad konkretnim objektima baze (kao što su određena baza, tabele, obeležja, pogledi...). Ako korisnik treba da vidi podatke neke tabele, potrebno mu je dodeliti SELECT ovlašćenje nad tom tabelom (isto važi za INSERT, UPDATE, DELETE, ...).

#### 48. Auditing BP (alati, opcije...)

Auditing BP uključuje nadzor baza podataka na taj način da smo svesni svih aktivnosti na bazi podataka koje iniciraju korisnici iste. Administratori baze podataka ili sigurnosni eksperti često postavljaju audit za sigurnosne namene.

Database auditing pomaže da odgovori na pitanja kao:

- Ko pristupa podacima ili menja podatke?
- Kada su ti podaci promenjeni?
- Koji je stari sadržaj bio pre promene?

#### 49. Migracija baze podataka

Definicija: Migracija baze podataka je proces prenosa podataka iz jednog sistema za upravljanje bazama podataka (SUBP) u drugi. Ovaj postupak je složen i zahteva pažljivo planiranje, implementaciju i testiranje kako bi tranzicija bila uspešna. Migracija može uključivati prenos podataka, struktura baze, logiku i aplikacije koje koriste bazu podataka. Cilj je osigurati kontinuitet poslovnih operacija i minimizirati prekide rada tokom tranzicije.

Razlozi za migraciju:

- a. Ažuriranje SUBP-a na noviju verziju.

- b. Prebacivanje platformi (npr. sa Oracle na MS SQL Server).
- c. Konsolidacija baza podataka radi smanjenja troškova i pojednostavljenja upravljanja.
- d. Spajanje ili deljenje baza zbog promena u poslovnim zahtevima.

Proces migracije: Migracija podataka obuhvata više aktivnosti koje prenose podatke iz starog u novi sistem, uz sve potrebne transformacije prema zahtevima novog sistema. Razlozi uključuju modernizaciju, smanjenje troškova održavanja zastarelih sistema i prilagođavanje novim poslovnim strategijama.

Rizici i teškoće: Migracija baze podataka nosi rizike kao što su:

- Oštećeni podaci
- Nepoznata zavisnost podataka
- Direktne izmene bez adekvatnih alata
- Greške u šemama i alatima za migraciju
- Problemi sa kvalitetom podataka i razumevanjem poslovne logike
- Kompleksnost alata za Export, transformaciju i Import podataka

Tipovi migracija:

- e. Migracija na SUBP istog proizvođača i verzije:
  - Najjednostavnija migracija.
  - Koriste se backup i restore procedure.
- f. Migracija na SUBP istog proizvođača različite verzije:
  - Potencijalne komplikacije zbog razlika u verzijama.
- g. Migracija na SUBP drugog proizvođača:
  - Najkomplikovanija migracija.
  - Zahteva specifične alate i potencijalno pisanje aplikacija za Export i Import podataka.

Alati za migraciju:

- IMPORT/EXPORT: Pojedinačnih tabela, različitih formata (tekstualne datoteke, .XLS), direktno preuzimanje tabela s drugog servera, dela baze podataka.
- Copy baze podataka

Migracija baze podataka je kompleksan proces koji zahteva detaljno planiranje, korišćenje odgovarajućih alata i pažljivo izvršenje kako bi se osigurao uspeh.

## 50. Koraci migracije baze podataka

Svaka od metodologija ukazuje da je potrebno razmotriti sledeće korake u migraciji podataka:

1. Analiza poslovnog uticaja
2. Prikupljanje informacija
3. Mapiranje, dizajniranje
4. Planiranje migracije
5. Testiranje (plana) migracije
6. Migracija
7. Validacija

## 51. Sigurnosne kopije baze podataka

- Sigurnosna kopija (Backup) je proces formiranja kopija podataka koje se mogu koristiti za oporavak podataka usličaju njihovog gubitka, iz bilo kojih razloga.

- Postoji potreba za stalnim kreiranjem rezervnih kopija podataka da bi se sprečio njihov gubitak;
- Nije dovoljno samo posedovanje rezervnih kopija već je imperativ da one budu na drugoj lokaciji van opasnosti od uništenja.
- Sigurnosne kopije su osiguranje od gubitka podataka

Sigurnosna kopija je važna:

- Prevencija gubitka podataka
- Poslovni kontinuitet
- Regulatorni zahtevi i usklađenost

## **52. Vrste sigurnosnih kopija baze podataka**

- Potpune sigurnosne kopije,
- Diferencijalne sigurnosne kopije,
- Inkrementalne sigurnosne kopije,
- Sigurnosne kopije Loga transakcija, itd.

## **53. Oporavak baze podataka od ispada**

- Oporavak baze podataka (Restore) je proces vraćanja podataka iz sigurnosne kopije. To se obično radi za oporavak podataka koji su izgubljeni, oštećeni ili izbrisani.

Koraci procesa oporavka BP:

- Izbor Sigurnosne Kopije: Odabrati odgovarajuću sigurnosnu kopiju za oporavak na osnovu tipa gubitka podataka i vremenske tačke do koje se želi obnoviti BP.
- Priprema za oporavak: Proveriti da li je ciljna lokacija spremna za oporavak, uključujući dovoljan prostor za smeštanje i odgovarajuće dozvole.
- Proces oporavka: Inicirati proces oporavka korišćenjem alata za upravljanje sigurnosnim kopijama ili komandama baze podataka.
- Provera: Nakon oporavka, proveriti da li su podaci uspešno obnovljeni i da li sistem pravilno funkcioniše.

Najbolje prakse za oporavak BP:

- Testiranje Oporavka: Redovno testirati proces obnavljanja kako bi se uverilo da su sigurnosne kopije ispravne i da se mogu uspešno obnoviti podaci.
- Plan Oporavka: Izraditi i održavati detaljan plan obnavljanja koji uključuje procedure i prioritete.
- Edukacija: Obezbediti da osoblje razume proces oporavka BP i da zna kako reagovati u slučaju gubitka podataka.

## **54. Uzroci koji mogu dovesti do potrebe za oporavkom BP**

- Hardver
- Greške u samim aplikacijama
- Padovi operativnog sistema
- Korisničke greške
- Sigurnosni propusti
- Vanredne situacije

## **55. Backup naredba**

Naredba za Backup baze podataka omogućuje:

- Backing up an entire database:

- Backing up specific files or filegroups:
- Backing up a transaction log-a:

## 56. Restore naredba

Restauracija baze podataka na osnovu sigurnosne kopije omogućuje RESTORE naredba. Restore naredbom moguće je:

- Restore an entire database:
- Restore part of a database:
- Restore specific files or filegroups:
- Restore a transaction log-a:

## 57. Automatizacija Backup-a i Restore-a

Značaj automatizacije: Smanjuje mogućnost ljudske greške, obezbeđuje redovnost sigurnosnih kopija.

Alati za automatizaciju: SQL Server Agent i Maintenance Plans

SQL Server Agent

- Servis za planiranje i automatizaciju zadataka u okviru MS SQL Server-a,
- Kreiranje Job-a za Backup,
- Definisanje Step-a za izvršavanje T-SQL komande BACKUP DATABASE,
- Postavljanje Schedule-a za Job.

Maintenance Plans

- Alat unutar SQL Server Management Studio-a (SSMS) za automatizaciju rutinskih zadataka.
- Kreiranje novog Maintenance Plan-a,
- Dodavanje Backup Database Task-a,
- Definisanje Schedule-a za Plan.

Najbolje prakse za automatizaciju

- Monitoring i upozorenja: Konfiguracija SQL Server Agent Job-a ili Maintenance Plan-a da šalju obaveštenja u slučaju grešaka.
- Testiranje: Redovno testiranje procesa Backup-a kako bi se obezbedilo da automatizacija funkcioniše kako treba.
- Dokumentacija: Potrebno je detaljno dokumentovanje svih kreiranih Job-ova i Planova.

## 58. Performansa baze podataka

Performansa baze podataka je brzina kojom SUBP obezbeđuje informacije korisnicima.

## 59. Faktori koji utiču na performansu baze podataka

Faktori koji utiču na performansu baze podataka:

- Obim posla
- Protok
- Resursi
- Optimizacija
- Konkurentnost

## 60. Podešavanje performansi

- Obim posla varira tokom vremena, nekad je predvidiv, a nekad ne. Sastoji se od:



- Onlajn transakcija
- Batch poslova
- Ad hoc upita
- Data Warehouse analiza
- Sistemskih poslova
- Protok je ukupna sposobnost računara da obradi podatke. Faktori protoka su:
  - Brzina ulaza/izlaza
  - Brzina procesora
  - Mogućnost paralelizma
  - Efikasnost operativnog sistema
  - Sistemskog softvera
- Resursi – hardverski i softverski alati
  - Kernel BP
  - Diskovi
  - RAM
  - Kontroleri
- Optimizacija upita se vrši unutar SUBP-a. Potrebno je optimizovati i druge faktore za efikasan rad baze podataka.
- Konkurencija nad resursima: pojavljuje se kada više elemenata pokušava da koristi isti resurs istovremeno, povećana konkurencija smanjuje protok
- Vrste podešavanja performanse: podešavanje sistema, podešavanje BP i podešavanje aplikacije

#### **61. Plan upravljanja performansom**

- Plan upravljanja performansama ima za cilj upravljanje i analizu performansi aplikacija baza podataka. Sastoji se od alata za praćenje performansi i finog podešavanja BP i SQL koda.
- Po pravilu 80/20, prvo se identifikuju problematična područja. Neefikasan SQL i programski kod uzrokuju većinu problema s performansama (75-80%).
- Loš SQL kod može biti rezultat neodgovarajućih indeksa, nepoštovanja postojećih indeksa, zastarelih statistika ili lošeg aplikativnog koda. Resursno intenzivne SQL iskaze identifikujte pomoću monitora i administrativnih alata, te fokusirajte podešavanje na njih.
- Proces upravljanja performansama uključuje praćenje, analizu i korekciju. Pristup može biti proaktivan (rešavanje problema unapred) ili reaktivan (rešavanje neočekivanih problema).

#### **62. Neki od razloga degradacije performanse baze podataka**

Performanse mogu degradirati tokom vremena zbog:

- Rasta baze podataka.
- Novih pristupnih putanja podacima.
- Dodatnih korisnika.
- Promena u poslovanju.

#### **63. Alati za praćenje i podešavanje performanse baze podataka**

- Monitori performanse
- Alati za planiranje kapaciteta
- Alati za SQL analizu i podešavanje
- Alati konkretnog SUBP-a
  - Rade samo s jednim sistemom
- Heterogeni alati
  - Rade s različitim BP i Operativnim sistemima

- Kod MS SQL Server-a:
  - Database Engine Tuning Advisor
  - System Stored Procedures
  - DBCC (Database Console Command)
  - Built-in functions