Administriranje baza podataka

Predavanje 03

Sadržaj

- Upravljanje performansama
- Korištenje memorije
- Tehnike optimiziranja baze

1. Upravljanje performansama

- Reaktivno i proaktivno upravljanje performansama
- Nadgledanje, analiza i optimizacija
- Tipovi alata za upravljanje performansama
- Povezanost performansi baze i ostatka sustava

Reaktivno i proaktivno upravljanje performansama

- Reaktivno upravljanje
  - Na problem reagiramo tek nakon što se on dogodi
  - Treba ga izbjegavati, premda se ne može sasvim izbjeći

- Proaktivno upravljanje
  - Nadgledanje
  - Uočavanje problema u njihovom nastanku
  - Sprečavanje eskalacije problema

Faze u proaktivnom upravljanju performansama

Nadgledanje

- Promatranje sustava dok on radi

- Može uključivati:
  - Nadgledanje stanja komponenti sustava (***Windows Performance Monitor***)
  - Bilježenje procesa koji blokiraju rad drugih procesa (***SQL Activity Monitor***)
  - Praćenje SQL naredbi koje se izvode nad bazom (***SQL Profiler***)
- Potrebno paziti da se nadgledanjem ne preoptereti sustav
  - Preporuka: Nadgledati samo mali skup dobro odabranih parametara

Analiza

- Otkrivanje uzroka eventualnih problema
- Dovođenje prikupljenih podataka u međusobnu vezu
  - Izolirano promatranje samo jedne komponente može navesti na krive zaključke
  - Primjer:
    - Malo slobodne memorije → treba povećati RAM?
    - Možda su uzrok loši upiti
  - *Baseline* – lista s referentnim vrijednostima promatranih aspekata sustava (*performance countera*)
  - Rezultate nadgledanja uputno je pohraniti
    - Mogu se izvoditi zaključci o trendovima

Optimizacija

- Izmjene dijelova sustava s ciljem poboljšanja performansi baze
- Primjeri:
  - Zamjena diskova bržima
  - Korekcije loše napisanih upita
  - Alociranje više memorijskog prostora za međuspremnik

Tipovi alata za upravljanje performansama

- Alati za nadgledanje stanja komponenti sustava
- Alati za praćenje procesa u bazama
- Alati za SQL analizu
- Alati za praćenje izvođenja SQL naredbi
- Alati za ugađanje performansi
- Alati za reorganizaciju

Alati za nadgledanje stanja komponenti sustava

- *Windows Performance Monitor*
  - Administrativni alat u Windowsima za nadgledanje performansi računala
  - Mogućnosti:
    - Praćenje trenutnih vrijednosti
    - Pregledavanje povijesnih podataka iz prošlih mjerenja
    - Automatsko pokretanje nadgledanja u unaprijed zadanom terminu
  - *System Summary* - tekstualno su prikazani pokazatelji stanja najvažnijih komponenti sustava – memorije, mreže, diska i procesora

Alati za nadgledanje stanja komponenti sustava

- Mogu mjeriti vrijednosti *performance countera*
- *Performance counteri* - parametri koji opisuju stanje komponenti sustava
  - Primjeri *performance countera*:
    - Za opterećenost procesora ili diskova
      - Physical Disk: Avg. Disk Queue Length
    - Za zauzetost memorije
      - Memory: Available Bytes
    - Za korištenje međuspremnika
      - SQL Server Buffer Manager: Buffer Cache Hit Ratio

Alati za nadgledanje stanja komponenti sustava

- Puno bolji pregled može se dobiti uz pomoć *Resource Monitora*
- Za svaku od navedenih komponenti rezerviran je poseban panel:
  - Na panelu CPU za svaki se proces može vidjeti u kojoj mjeri on opterećuje procesor
  - Na preostalim panelima vidi se koliko koji proces doprinosi opterećenju diska, mrežnom opterećenju i zauzeću memorije
- Možemo nadgledati rad različitih hardverskih komponenti na računalu, ali i performanse softvera koji izlažu neke svoje objekte ovakvim alatima
- Za SQL Server možemo promatrati npr. koliko efikasno koristi međuspremnik ili koliko ima otvorenih konekcija na određenoj instanci

Alati za nadgledanje stanja komponenti sustava

- *Reliability and Performance Monitor* > *Performance Monitor* - pokretanje nadgledanja trenutačnog stanja računala
- Predefinirano je da se pokazuje samo opterećenost procesora, a po potrebi možemo dodati i neke druge pokazatelje

- Za razumijevanje rada *Performance Monitora* potrebno je znati što su objekti, brojači i instance te kako su oni međusobno povezani

Alati za nadgledanje stanja komponenti sustava

- **Objekti (*objects*)**
  - Glavne komponente računalnog sustava
  - Mogu biti dijelovi hardvera (npr. diskovi ili procesori), sistemski procesi ili softverske komponente poput instanci SQL Servera
  - U Windowsima postoji fiksan broj objekata koji se mogu pratiti preko *Performance Monitora*, a neki softveri poput SQL Servera prilikom instalacije dodaju i neke svoje objekte, što omogućuje da se njegove performanse također mogu nadgledati kroz *Performance Monitor*

Alati za nadgledanje stanja komponenti sustava

- **Brojači (*performance counters*)**
  - Brojači skupljaju podatke o različitim aspektima objekata
  - Npr. za **Memory** postoje brojači **Available Mbytes**, **Pages/sec** i drugi
  - Možemo ih dodavati i sami
- **Instance (*instances*)**
  - Ako neki objekt ima više instanci, brojači mogu pratiti vrijednosti za sve instance zajedno ili za svaku posebno
  - Npr. možemo vidjeti koliko je opterećen svaki pojedini procesor na poslužitelju ili koliko je njihovo ukupno opterećenje

Alati za nadgledanje stanja komponenti sustava

- Vrijednosti brojača mogu se prikazati kao graf, histogram ili izvještaj
- Često se kod redovnih održavanja sustava vrijednosti brojača snimaju u log datoteke za kasniju analizu - na taj način dobivamo mogućnost praćenja trendova u vrijednostima brojača, što može pomoći pri otklanjanju problema ili u planiranju promjena na sustavu
- Da ne bismo za svako nadgledanje morali iznova definirati popis brojača koje želimo pratiti, možemo koristiti *Data Collector Setove*
- *Data Collector Setove* možemo automatizirati, tako da se pokreću i zaustavljaju u točno određeno vrijeme

Alati za nadgledanje stanja komponenti sustava

- Nadgledanje SQL Servera izuzetno je važno jer se na taj način identificiraju komponente sustava koje ne rade dobro, a analizom rezultata mogu se utvrditi uzroci loših performansi

- Optimizacija može uključivati nadogradnju hardvera, distribuciju opterećenja na druge poslužitelje, restrukturiranje baze, postavljanje indeksa ili modifikaciju upita

Alati za nadgledanje stanja komponenti sustava

- Primjer mogućih zaključaka:
  - Ako su vrijednosti brojača *Current Disk Queue Length* i *% Disk Time* neprestano vrlo visoke, treba razmisliti o poduzimanju neke od sljedećih akcija:
    - Zamijeniti diskove ili diskovne sustave bržima
    - Premjestiti neke datoteke na dodatne diskove
  - Ako je neprestano visok postotak korištenja procesora, to može ukazivati na potrebu da se doda još procesora ili da se postojeći procesor zamijeni jačim, ali ponekad su uzrok velikog opterećenja procesora loše napisani upiti koji se mogu optimizirati

Alati za praćenje procesa u bazama

- *SQL Server Activity Monitor*

- Mogu prikazati:
  - Koji se procesi izvode u bazama
  - Koji su korisnici pokrenuli koji proces
  - Koje se naredbe izvode pod kojim procesom
  - Ima li međusobnog blokiranja
  - Koliko procesi čekaju na dobivanje pojedinih resursa

Activity Monitor paneli

- ***Overview* panel**
  - grafički se prikazuju podaci iz kojih možemo dobiti brzi uvid u opterećenost SQL Server instance koju nadgledamo:
    - ***% Processor Time*** - opterećenje procesora od SQL Server procesa
    - ***Waiting Tasks*** - zadaci koji stoje u redu čekanja da se oslobodi određeni resurs
    - ***Database I/O*** - opterećenje diska zbog dohvaćanja ili zapisivanja podataka
    - ***Batch requests/sec*** - broj skupina SQL naredbi (*batches*) koje se obrađuju po sekundi

Activity Monitor paneli

- ***Processes* panel**
  - Ovdje se prikazuju osnovne informacije o svakom procesu koji se izvodi na odabranoj instanci SQL Servera
    - identifikacijski broj (*Session ID*)
    - korisnik koji je pokrenuo proces
    - baza nad kojom se izvodi proces
    - naziv aplikacije iz koje je pokrenut proces
    - naziv računala s kojeg je pokrenut proces
  - Vrlo korisni mogu biti i podaci u stupcima *Blocked by* i *Head blocker*, koji pokazuju je li proces blokiran od strane nekog drugog procesa ili možda upravo on uzrokuje blokiranje drugih

Activity Monitor paneli

- ***Resource Waits* panel**
  - Na ovom panelu može se vidjeti postoje li procesi koji čekaju da se oslobode određeni resursi
  - Može se vidjeti koliko procesa čeka na pojedini resurs
  - Prikazuje se i koliko je ukupno čekanje na pojedine resurse od početka mjerenja

Activity Monitor paneli

- ***Data File I/O* panel**
  - Za svaku podatkovnu i log datoteku u bazama na odabranoj instanci prikazano je kojom se brzinom podaci iz nje čitaju ili u nju zapisuju
  - Vidi se i koliko je vrijeme potrebno da se posluži I/O zahtjev

- ***Recent /Active Expensive Queries* panel**
  - Mogu se vidjeti „najskuplji“ upiti koji su se izvodili u posljednjih 30 s
  - Za svakog od njih prikazani su važni pokazatelji njegovih performansi poput trajanja, korištenja procesora te broja logičkih i fizičkih čitanja i zapisivanja

Alati za SQL analizu

- *Estimated / Actual Execution Plan*

- Grafički prikaz plana izvođenja neke SQL naredbe
  - Primjer:
    - Kako će se tablice spajati
    - Koji će se indeksi koristiti
    - Koji korak je „najskuplji”

Alati za praćenje izvođenja SQL naredbi

- *SQL Server Profiler*
- Nadgledanje izvođenja upita nad bazama
- Mogu prikazati koje se naredbe izvode na odabranim bazama
- Naredbe se mogu filtrirati
  - Naredbe koje dugo traju
  - Naredbe koje izvodi određeni korisnik
  - Naredbe koje sadrže određeni tekst
- Pomažu u otkrivanju problematičnih upita
- Pomažu programerima kod *debugginga*

SQL Server Profiler

- Između ostalog, moguće izdvojiti:
  - Upite koji traju dulje od zadanog vremena
  - Upite koje izvodi određeni korisnik
  - Upite koji se izvode nad određenom bazom
  - Upite koji sadrže određeni tekst
  - Upite koji mijenjaju definiciju nekih objekata (npr. ALTER TABLE, DROP INDEX i slično)
  - …

SQL Server Profiler

– …
– Koliko su pojedini upiti potrošili procesorskog vremena
– Koliko su pojedini upiti napravili čitanja/pisanja po disku
– Pojavu *deadlockova* i njihove detalje
– Pokušaje logiranja
– Slučajeve korištenja značajki za koje je najavljeno da će biti ukinute u nekoj od sljedećih verzija
– Izvedbene planove po kojima su se upiti izvodili
– Skeniranja tablica ili indeksa

SQL Server Profiler

- Kod pokretanja snimanja, između ostalog, treba odabrati koji će se događaji pratiti te kamo će se oni snimiti
- Nadgledanje stvara dodatno opterećenje na poslužitelju pa ako se snima puno događaja, to može značajno degradirati performanse
- Postoji mogućnost da se za praćenje odaberu samo događaji koji su nam najvažniji – najlakše odabrati jedan od postojećih predložaka za snimanje
- Na primjer, postoje predlošci za izdvajanje dugotrajnih upita, za praćenje *lockova* ili za optimizaciju upita
- Odabrane predloške možemo modificirati prema vlastitim potrebama i iz njih kreirati vlastite

SQL Server Profiler

- Prikupljene podatke moguće je pohraniti u datoteku ili tablicu u bazi da bi se kasnije mogli analizirati ili da bi se događaji mogli ponovo izvesti na testnoj okolini (*replay*)
- *Trace* datoteka ili tablica može se iskoristiti za ulazne podatke u *Database Engine Tuning Advisor*
- Analizirajući podatke, *Tuning Advisor* može utvrditi koje su promjene nad bazom potrebne da bi se njezine performanse poboljšale
- Dodatno, *trace* datoteka može se poslati osobama koje nam pružaju tehničku podršku

SQL Server Profiler

- Informacije koje se bilježe u *traceu* podijeljene su u kategorije, kategorije sadrže događaje, a svaki događaj sadrži atribute koji se nazivaju stupci

- *Trace* kategorije
  - Sadrži međusobno povezane tipove događaja

SQL Server Profiler

– Neke od kategorija:

- *Security Audit*: uključuje događaje poput prijave i odjave s instance SQL Servera, neuspješne prijave, dodavanja *logina* i *usera* ili izvođenja GRANT, REVOKE i DENY naredbi
- *Sessions*: uključuje događaj *ExistingConnection*, koji za svaku konekciju na instancu SQL Servera prikazuje njezina svojstva
- *Stored Procedures*: uključuje događaje koji se okidaju pri izvođenju pohranjenih procedura
- *TSQL*: uključuje događaje koji se okidaju pri izvođenju T-SQL naredbi proslijeđenih s klijenta instanci SQL Servera

SQL Server Profiler

- Stupci su atributi koji detaljnije opisuju događaje uhvaćene za vrijeme rada *tracea*
- Postavljanjem filtra na stupce možemo postaviti kriterije za podatke koje želimo prikupljati
- Na primjer, korištenjem filtra na koloni *Application Name* po defaultu se isključuju podaci generirani od samog *Profilera*

Alati za ugađanje (*tuning*)

- *SQL Server Database Engine Tuning Advisor*

- Mogu zaključiti što bi u bazi trebalo promijeniti da bi performanse bile bolje
  - Primjer: alat može zaključiti da je potreban indeks i automatski ga dodati
- Rade u sprezi s alatima za praćenje izvođenja SQL naredbi
- To su invazivni alati (mijenjaju strukturu baze) – preporučljivo ih je pokrenuti najprije na testnoj okolini

Alati za reorganizaciju

- Za reorganizaciju tablica ili indeksa
  - Da bi se izbjegli problemi poput fragmentacije ili nesortiranih redaka
  - `ALTER INDEX ... REORGANIZE`

Povezanost performansi baze i ostatka sustava

- Aplikacija može biti dobro napisana, ali radi loše zbog loše baze
- Baza može biti dobro dizajnirana, ali radi loše zbog hardverskih problema ili sistemskih postavki
- Baza može raditi loše zbog loših upita koji dolaze iz aplikacije

- Briga o dobrim performansama aplikacije briga je svih koji održavaju sustav – aplikacijskih programera, administratora baza i sistemskih administratora

2. Korištenje memorije

- Korištenje podatkovnog međuspremnika
- Korištenje programskog međuspremnika
- Ostali potrošači memorije

Korištenje podatkovnog međuspremnika (1)

- Da bi međuspremnik bio efikasan, najvažnije je odrediti njegovu optimalnu veličinu
  - Prevelik međuspremnik nepotrebno troši memoriju
  - Kod premalog međuspremnika mogu se dogoditi forsirana zapisivanja stranica na disk
- Logičko čitanje (*logical read*) – svako traženje podatka u podatkovnom međuspremniku
- Fizičko čitanje (*physical read*) – odlazak po podatak na disk jer nije pronađen u međuspremniku

Korištenje podatkovnog međuspremnika (2)

- Efikasnost čitanja =
  (broj logičkih čitanja – broj fizičkih čitanja) / broj logičkih čitanja
  - Pokazuje u kolikom se postotku tražene stranice već nalaze u međuspremniku
- Ugrubo: efikasnost čitanja veća od 80% je dobra
- Efikasnost ovisi o tipu procesiranja
  - Ako se neki procesi rijetko izvode, moguće je da će imati malu efikasnost čitanja
  - Efikasnost čitanja nakon skeniranja velikih tablica može pasti
- Efikasnost čitanja može se saznati čitanjem odgovarajućeg *performance countera*

Korištenje programskog međuspremnika

- Programski međuspremnik – pohranjuje izvedbene planove SQL naredbi
- Jednom izrađeni izvedbeni planovi mogu se ponovo iskoristiti ako se SQL naredbe ponove
- Efikasnost programskog međuspremnika
  - Koliko često DBMS mora pripremati izvedbene planove jer ih nema gotove u međuspremniku
  - Ugrubo: dobro je ako je efikasnost veća od 60%

Ostali potrošači memorije (1)

- Otvorene baze
  - Za svaku bazu koja je otvorena za pristup (*online* baza) DBMS zahtijeva određenu količinu memorije za čuvanje metapodataka o njoj
- Otvoreni objekti
  - Svaka tablica ili indeks treba mjesto u memoriji za svoje metapodatke
- Konekcije na DBMS
  - Svaka korisnička konekcija smještena je kao objekt u memoriju

Ostali potrošači memorije (2)

- Lokoti
  - Moderni DBMS-ovi obično implementiraju višekorisnički rad preko “zaključavanja” određenih dijelova baze
  - Ako neka transakcija mijenja neki podatak, ona ga zaključa (drži nad njim X lokot) pa ga niti jedna druga transakcija ne može promijeniti dok se prva ne potvrdi ili poništi
  - Lokoti su također objekti u memoriji

3. Tehnike optimiziranja baze

- Indeksiranje
- Ostavljanje slobodnog prostora
- Raspoređivanje datoteka po diskovima
- Reorganizacija
- …

Indeksiranje (1)

- Jedna od najčešće korištenih tehnika za optimiziranje baze
- Indeksi mogu ubrzati sljedeće operacije:
  - Pronalaženje redaka prema zadanoj vrijednosti kolona
  - Spajanje (*join*) tablica
  - Agregiranje podataka
  - Sortiranje podataka
- Indeksi ubrzavaju samo određeni upit ili skup sličnih upita; nekim drugim upitima isti indeks neće biti od koristi

Indeksiranje (2)

- Ako je neki upit izrazito važan za poslovanje, indeksi se mogu postaviti da ubrzaju samo taj jedan upit
- Za različite upite nad nekom tablicom treba imati različite indekse da bi većina upita imala dobre performanse
- Previše indeksa može biti loše za performanse
  - Kod ažuriranja osnovnih zapisa treba ažurirati i indekse
  - Indeksi zauzimaju prostor
- Potreba za reindeksiranjem
  - S vremenom se aplikacije mogu nadograđivati
  - Mogu se pojaviti novi upiti koje treba optimizirati

Indeksiranje (3)

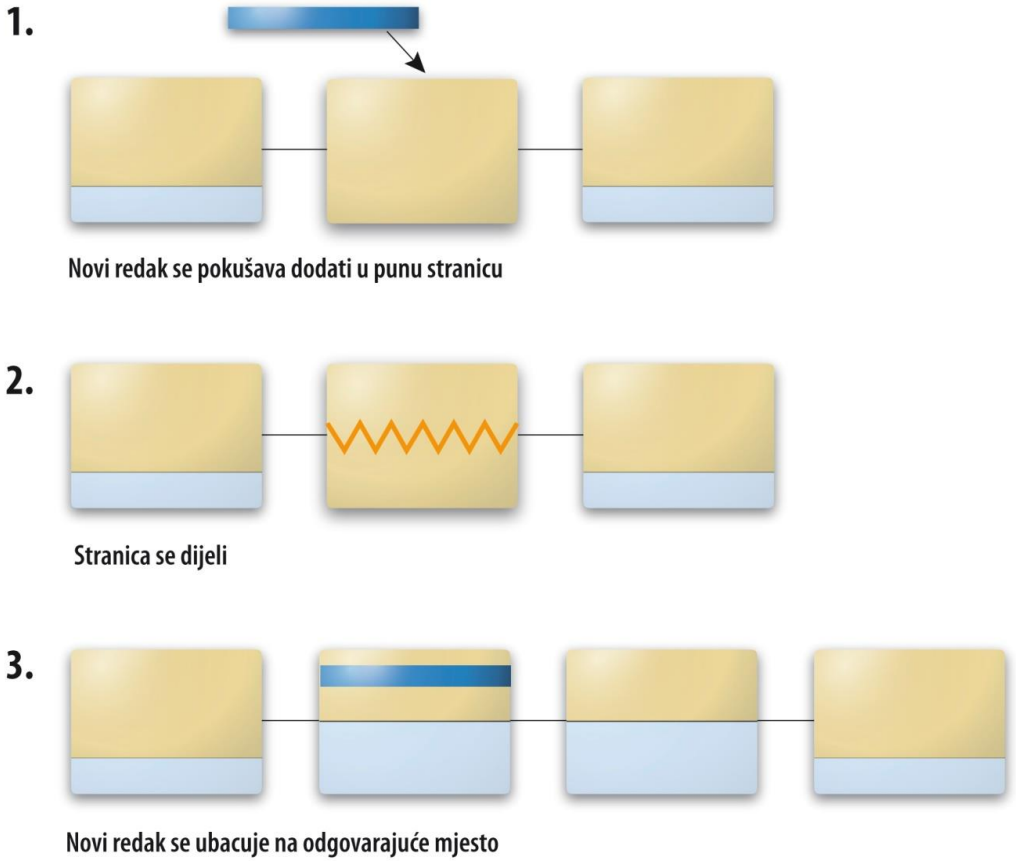
- DBMS sam odlučuje hoće li u određenoj situaciji koristiti indekse
  - Postoje načini da se DBMS prisili da koristi određeni indeks
  - Optimizatori upita su dobri i možemo biti poprilično sigurni da su generirali najbolji izvedbeni plan
  - Forsiranja korištenja indeksa u praksi se rijetko koriste

Ostavljanje slobodnog prostora (1)

- Kod dodavanja novih podataka u indekse i klasterirane tablice ne smije se narušiti sortiranost
- Ako više nema mjesta u stranici kojoj bi redak trebao pripadati, doći će do podjele stranice (*page splitting*)

Podjela stranice (*page splitting*)

- Polovica zapisa se iz pune stranice premjesti u novu
- Novi redak se nakon toga ubaci na odgovarajuće mjesto
- Fizički poredak zapisa više ne odgovara logičkom



Ostavljanje slobodnog prostora (2)

- Normalno je da se u transakcijskoj bazi ponekad dogodi podjela stranica
- Prečesta podjela stranica može značajno degradirati performanse baze
  - Te se situacije mogu izbjeći definiranjem faktora popunjenosti stranica (*fill factor*)
    - Koliki postotak stranice će ostati prazan prilikom izgradnje indeksa da bi se mogao iskoristiti za buduća ubacivanja novih zapisa
- Postoje *performance counteri* koji prate broj podjela stranica

Ostavljanje slobodnog prostora (3)

- Prednosti ostavljanja slobodnog prostora:
  - Ubacivanja su brža jer se podjele stranica rijetko događaju
  - Manje redaka na stranici rezultira boljom konkurentnošću
    - Kod zaključavanja stranica zaključava se manji broj redaka
- Nedostaci ostavljanja slobodnog prostora:
  - Baza zauzima više prostora
  - Skeniranja podataka traju dulje jer tablica ima više stranica
  - Manje redaka na stranici može povećati broj odlazaka na disk da bi se dohvatili svi potrebni podaci

Reorganizacija baze (1)

- Obilježja dezorganiziranosti
  - Fragmentiranost – tablice i indeksi sastoje se od puno malih i “raštrkanih” komadića prostora
  - Određeni tipovi operacija pretrage mogu se značajno usporiti

- Uzroci dezorganiziranosti
  - Podjele stranica
    - Fizički redoslijed stranica u indeksima postaje drugačiji od logičkog

Reorganizacija baze (2)

- Uzroci dezorganiziranosti (nastavak):
  - Ulančavanje i migracija redaka
    - Ako redak sadrži kolonu varijabilne duljine i ažuriramo ga, može se dogoditi da postane dulji nego što je prvotno bio
    - Budući da se ne može “stisnuti” između svojih susjeda, dio retka se može prebaciti u neki prazan prostor (ulančavanje redaka, *row chaining*)
    - Ako se prebaci cijeli redak, govori se o migraciji retka (*row migration*)

Reorganizacija baze (3)

- Reorganizacija baze
  - Većina DBMS-ova ima pomoćni program za reorganizaciju
  - Reorganizacija obično uključuje izradu kopije zadanog objekta i uklanjanje anomalija za vrijeme kopiranja. Na kraju se original i reorganizirana kopija zamijene
  - *Online* reorganizacija – u sustavima s velikim brojem transakcija može trajati dugo
  - Reorganizacija zahtijeva dodatan diskovni prostor

Reorganizacija baze (4)

- Prikupljanje informacija o dezorganiziranosti

  – Informacije o dezorganiziranosti objekata obično se smještaju u sistemske kataloge ili unutar posebnih stranica samog objekta

  https://learn.microsoft.com/en-us/sql/relational-databases/system-dynamic-management-views/sys-dm-db-index-physical-stats-transact-sql?view=sql-server-ver16

  – *Primjer:*

```
SELECT * FROM sys.dm_db_index_physical_stats
    (DB_ID(N'AdventureWorksOBP'), NULL, NULL, NULL , 'DETAILED');
```