

SQL SERVERDA MA'LUMOTLARNI SAQLASH, INDEKSLASH VA NUSXALASH ISHLARI

Topvoldiyev Islomjon Ikromjon o'g'li

Farg'ona davlat universiteti

Islombektopvoldiyev564@gmail.com

Anotatsiya : *Ushbu maqolada Microsoft SQL Server'da ma'lumotlarni fiziki saqlash mexanizmlari, turli indekslash usullari va ma'lumotlarni zaxiralash (backup) strategiyalari ko'rib chiqiladi. Ma'lumotlar fayllari, fayl guruhlar, sahifalar va ekstensiyalar orqali ma'lumotlar qanday tashkil etilishi tahlil qilinib, klaster va nodavriy indekslar hamda ularning ishlash samaradorligi muhokama qilinadi. Shuningdek, to'liq, differentsial, tranzaksiya jurnal va maxsus nusxalar turlari haqida ma'lumot berilib, ular yordamida ma'lumotlarni yo'qotishdan himoyalash strategiyalari yoritiladi. Har bir bo'limda nazariy ma'lumotlar amaliy misollar va iqtiboslar bilan mustahkamlanadi.*

Kalit so'zlar: *SQL Server, Ma'lumotlar bazasi, Ma'lumotlarni saqlash, Sahifa ,Ekstensiya Fayl guruhi , Indekslish,Klaster indeks ,Nodavriy indeks , Kolonka indeksi , Nusxalash, To'liq nusxa, Differentsial nusxa, Tranzaksiya jurnal nusxasi, Qayta tiklash rejimi.*

Аннотация : *В данной статье рассматриваются механизмы хранения данных, индексации и резервного копирования в Microsoft SQL Server. Подробно анализируется физическая структура хранения данных, включая страницы, экстенды, файловые группы и основные типы файлов (.mdf, .ndf, .ldf). Описаны различные виды индексов: кластерные, некластерные и колоночные, их влияние на производительность запросов и методы эффективного использования. Кроме того, рассматриваются стратегии резервного копирования, включая полное, дифференциальное, журнальное и специализированное копирование. Представлены рекомендации по выбору модели восстановления и обеспечению надежности хранения данных в крупных информационных системах. Теоретические положения сопровождаются практическими примерами.*

Ключевые слова: *SQL Server, база данных, хранение данных, страница, расширение, группа файлов, индексирование, кластерный индекс, некластерный индекс, столбцовый индекс, репликация, полная резервная*

копия, дифференциальная копия, резервная копия журнала транзакций, режим восстановления.

Abstract : *This article discusses data storage, indexing, and backup mechanisms in Microsoft SQL Server. It provides a detailed analysis of the physical data storage structure, including pages, extents, filegroups, and primary file types (.mdf, .ndf, .ldf). The article describes various indexing types such as clustered, non-clustered, and columnstore indexes, emphasizing their impact on query performance and practical usage. Backup strategies, including full, differential, transaction log, and specialized backups, are thoroughly examined. The paper offers guidance on choosing the appropriate recovery model to ensure data reliability in large-scale information systems. Theoretical discussions are supported by practical examples.*

Keywords: *SQL Server, database, data storage, page, extent, filegroup, indexing, clustered index, non-clustered index, columnstore index, replication, full backup, differential backup, transaction log backup, recovery model.*

Kirish : Korxonalar va tashkilotlar ommaviy va muhim ma'lumotlarni boshqarishda yuqori tezlik va ishonchlilikni talab qiladi. Shu bois, ma'lumotlar bazasida ma'lumotlarni to'g'ri saqlash, ularni tezkor qidirish uchun indekslash va yo'qotishdan himoyalash uchun zaxira nusxalari muhim ahamiyat kasb etadi. SQL Server shunday muhitda keng qo'llaniladigan tizim bo'lib, uning arxitekturasi ma'lumotlarni tartib bilan saqlash va ishlashini optimallashtirishga qaratilgan. Ma'lumotlar bazasini mustahkam boshqarish uchun samarali saqlash usullari, murakkab indekslash mexanizmlari va to'g'ri ishlab chiqilgan zaxiralash strategiyalari kerak bo'ladi. Ma'lumotlarni saqlash, indekslash va zaxiralash bo'yicha to'liq strategiya biznes talablariga muvofiq ma'lumotlarga doimiy kirish imkoniyatini ta'minlab, ma'lumotlarni yo'qotish xavfini minimallashtiradi. Quyida birinchi bo'limda SQL Serverda ma'lumotlarning fizik joylashuvi tushuntiriladi, so'ngra indekslash turlari va ularning samaradorligi ko'rib chiqiladi, oxirida esa nusxalash turlari va ulardan samarali foydalanish strategiyalari keltiriladi. Ma'lumotlarni saqlash mexanizmlari: SQL Server ma'lumotlarni diskda va xotirada saqlash uchun **ma'lumot bazasi fayllari** tizimidan foydalanadi. Har bir ma'lumotlar bazasi kamida bitta **asosiy** (PRIMARY) ma'lumot fayli va bitta tranzaksiya jurnal (LOG) fayliga ega bo'ladi. Asosiy fayl (.mdf) bazaning ishga tushishi uchun zarur bo'lgan ma'lumotlarni o'zida saqlaydi, yordamchi (yoki ikkinchi darajali) ma'lumot fayllari (.ndf) esa katta hajmdagi ma'lumotlarni bir nechta disk ustida taqsimlash

imkonini beradi. Har bir ma'lumot fayli ichida jadvallar, indekslar, saqlangan protseduralar, ko'rinishlar va boshqa obyektlar malumotlari saqlanadi. Tranzaksiya jurnal fayli (.ldf) esa har bir o'zgartirish operatsiyasi (INSERT, UPDATE, DELETE va h.k.) tarixini yozib boradi va ma'lumotlar bazasini har qanday vaqtda tiklash uchun zarur bo'lgan malumotlar ketma-ketligini ta'minlaydi. Fayllarni boshqarishni soddalashtirish va disk resurslaridan foydalash samaradorligini oshirish uchun SQL Server'da **fayl guruhleri** (filegroups) tushunchasi joriy etilgan. Asosiy filegroup (PRIMARY) birlamchi ma'lumot faylini o'z ichiga oladi va agar boshqa filegroup'lar ko'rsatilmasa, barcha yangi obyektlar ushbu guruhga yoziladi. Foydalanuvchi tomonidan yaratiladigan qo'shimcha filegroup'lar yordamida ma'lumot fayllari ma'lum guruhlarga ajratib qo'yilishi mumkin. Masalan, bir necha yordamchi .ndf faylni turli fizik diskda joylashtirib, ularni bitta filegroupga (fgroup1) biriktirish mumkin. Shu filegroupga bag'ishlangan jadvaldan ma'lumot o'qilganda, so'rovlar hosil bo'lgan kiritmalar uchta disk bo'ylab teng taqsimlanadi va bu ma'lumotlarni parallel ravishda o'qish bilan tizim samaradorligini oshiradi. Fayl guruhlarini yaratish va kengaytirish orqali disklar qo'shimcha tarzda qo'llaniladi, natijada ma'lumotlar hajmi o'sgan sayin yangi disklarni qo'shish orqali tizimni kengaytirish osonlashadi. SQL Serverda ma'lumotlar **sahifalar** (pages) bo'yicha saqlanadi. Har bir sahifa doimo 8 KB hajmga ega bo'lib, ma'lumotlar faylidagi bo'sh joylar mantiqan sahifalarga ajratiladi. Disk operatsiyalari ham sahifa birligi bo'yicha bajariladi — ya'ni tizim har safar 8 KB o'lchamdagi sahifani to'liq yozadi yoki o'qiydi. Ma'lumotli sahifalar (data pages) ichida jadvallardagi qatorlar yozilib boriladi, indeks sahifalari (index pages) esa qatorlar manzillarini belgilaydi. Shuningdek, tizim tarkibi haqidagi ma'lumotlarni saqlovchi maxsus boshqaruv sahifalari ham mavjud. Ma'lumotni samarali boshqarish uchun sakkizta ketma-ket sahifa bitta **ekstensiya** (extent) ga birlashadi va bo'sh sahifalar shu bloklarga bo'ysundiriladi. Bitta ekstensiyada birlashtirilgan sahifalar yordamida SQL Server disk maydonini tartib bilan bo'lish va bo'sh joyni tezda topish imkoniga ega bo'ladi. Shu tariqa, ma'lumotlar fayli, fayl guruhleri, sahifalar va ekstensiyalar tuzilmasi ma'lumotlarni kattaroq hajmda samarali saqlash va boshqarish uchun asosiy mexanizmni tashkil etadi. Indekslash turlari va samaradorligi: SQL Server'da indekslar jadval yoki ko'rinishdagi ma'lumotlarga tezkor kirishni ta'minlaydigan maxsus tuzilmalar bo'lib, ular ustun(lar)dan olinadigan kalit qiymatlaridan tashkil topgan B+daraxtlarni ishga soladi. Indeks kalitlari yordamida tizim kerakli qator(lar)ni tezda topa oladi, natijada filtrlangan yoki tartiblangan so'rovlar katta hajmdagi ma'lumotni skanerlash o'rniga indeks orqali

amalga oshadi. **Klaster indeks** tufayli jadval ma'lumotlari indeks kalitlari bo'yicha tartiblab saqlanadi. Bitta jadval faqat bitta klaster indeksga ega bo'lishi mumkin, chunki ma'lumotlarni faqat bir xil tartibda joylashtirish mumkin. Klaster indeks mavjud bo'lsa, jadvalga kiritilgan qatorlar aniq tartibda saqlanadi va so'rov amalga oshirilganda to'g'ridan-to'g'ri shu tartib asosida qatorlarga murojaat qilinadi. Agar jadvalda hech qanday klaster indeks bo'lmasa, u paytda uning ma'lumot qatorlari **heap** deb nomlangan tartibsiz tuzilmada joylashadi (ya'ni ma'lumotlar diskda xronologik tartibda o'zgarib boradi). **Nodavriy indeks(lar)** esa alohida ma'lumot tuzilmasida saqlanadi va har bir indeks qatori ma'lumot qatori joylashuviga ko'rsatkichni (row locator) o'z ichiga oladi. Agar asosiy jadval heap bo'lsa, bu ko'rsatkich bevosita qatorning joylashgan manziliga ishora qiladi; agar jadval klaster indeksli bo'lsa, unda ko'rsatkich klaster indeks kalitini (ya'ni indeks kalit bo'yicha qayta yozilgan manzilini) belgilaydi. Shu tariqa, nodavriy indeks ma'lumotlar bazasi uchun qo'shimcha yo'nalish — ikkinchi indeks daraxtini ta'minlab, klasterlash bo'lmaganda ham qatorlarni aniqlashni tezlashtiradi. Klaster va nodavriy indekslar unikal (unique) yoki unikal bo'lmagan ko'rinishda bo'lishi mumkin. Unikal indeks holatida bir xil indeks kaliti bilan bir nechta qator bo'lmaydi. Indekslar ma'lumotlar bazasi tuzilishiga qo'yilgan asosiy va noyob cheklovlar bilan avtomatik bog'liq bo'lib, masalan PRIMARY KEY yaratishda avtomatik ravishda klaster indeks yaratiladi. Ko'pchilik so'rovlar va kattaroq ma'lumotlar bazalarida indekslarning to'g'ri tuzilishi ish samaradorligiga katta ta'sir ko'rsatadi. Ma'lumotlar bazasida juda yaxshi loyihalashtirilgan indekslar diskdan o'qiladigan ma'lumot hajmini sezilarli darajada kamaytiradi va tizim resurslarini tejaydi. Natijada, so'rovlar, ayniqsa katta jadval yoki bir nechta murakkab filtrlash va tartiblash talab etiladiganlar, indekslar bilan bir necha barobarga tezroq bajarilishi mumkin. Biroq indeks yaratish va yangilash ham qo'shimcha resurs sarflaydi: ma'lumotlarga har bir INSERT/UPDATE/DELETE amali indeks daraxtini yangilashni talab qiladi. Shu sababli indekslar sonini va tarkibini ehtiyotkorlik bilan rejalashtirish muhimdir. Yaqinda kiritilgan **kolonka indeksi** (columnstore) esa katta hajmdagi analitik hisob-kitoblar uchun an'anaviy B-daraxt indekslarga nisbatan sezilarli yuqori tezlik va zichroq siqishni ta'minlaydi. Microsoft hujjatlariga ko'ra, kolonka indekslari ma'lumotlarni ustunlar bo'yicha qayta joylash orqali analitik so'rovlarni 100 barobarga tezroq bajarishi va an'anaviy indekslarga qaraganda o'rtacha 10 barobarga katta siqishni ta'minlashi mumkin. Shu sababli katta hajmdagi ma'lumotlar ustida tahliliy so'rovlar bajarishda kolonka indekslari ish unumdorligini keskin oshiradi. Indeksning samaradorligini ta'minlash uchun

ularni muntazam tekshirish va kerak bo'lsa qayta qurish (rebuild) yoki qayta tashkil etish (reorganize) kabi operatsiyalarni bajarish tavsiya etiladi. Buning natijasida indekslardagi parchalanishlar (fragmentation) kamayadi va so'rov tezkorligi barqarorlanadi. Xulosa qilib aytganda, indekslash SQL Server'da so'rovlarni optimallashtirishning asosiy vositasi bo'lib, u yaxshi loyihalashtirilganda tizimni sezilarli darajada samaraliroq qiladi. Nusxalash turlari va strategiyalari: Ma'lumotlar bazasidagi o'zgartirishlarni xavfsiz saqlab qolish va istalgan vaqtda qayta tiklash uchun SQL Server bir qator nusxalash turlarini taklif etadi. Ular orasida eng asosiy va keng ishlatiladiganlar – **to'liq nusxa (full backup)**, **differentzial nusxa (differential backup)** hamda **tranzaksiya jurnal nusxasi (transaction log backup)**. Bundan tashqari **oxirgi jurnal nusxasi (tail-log backup)** va **copy-only nusxasi** kabi maxsus turlar ham mavjud. **To'liq nusxa** ma'lumotlar bazasining barcha obyektlarini o'z ichiga olgan yagona nusxadir. U jadval va indekslardan tortib protseduralar va triggerlargacha barcha ma'lumotlarni birdaniga zaxiralaydi. To'liq nusxa olish eng sodda va to'liq tiklash imkoniyatini beradi, shuningdek, boshqa barcha nusxalash turlarining asosiy poydevorini tashkil etadi. Amaliyotda har bir backup strategiyasida dastlab bitta to'liq nusxa olinishi shart bo'ladi. **Differentzial nusxa** esa oxirgi to'liq nusxadan beri yuz bergan barcha o'zgartirishlarni o'z ichiga oladi. Ya'ni, agar bir jadvalga yangi yozuvlar qo'shilgan bo'lsa yoki ba'zi ma'lumotlar o'zgartirilgan bo'lsa, differentzial nusxa faqat shu sahifalarda bo'lgan o'zgarishlarni qamrab oladi. Shu bois, differentzial nusxa to'liq nusxaga qaraganda ancha kam vaqt va joyni oladi. Ammo ma'lumotlar bazasida ko'p sonli tranzaksiya bo'lgani sayin differentzial nusxa o'lchami ham kattalashadi. Agar juda uzoq vaqt davomida bir nechta differentzial nusxa olingan bo'lsa, oxirgi differentzial nusxa kattaligi to'liq nusxa hajmiga yaqinlashib qolar va tiklash vaqti oshishi mumkin. Shu sababdan, ma'lumotlar og'ishlari intensiv bo'lgan tizimlarda har haftada yoki muayyan interval bilan yangi to'liq nusxa olib, differentzial nusxalarni shu so'nggi to'liq nusxaga nisbatan olish tavsiya etiladi. Baza tiklash kerak bo'lsa, odatda oxirgi to'liq nusxa va undan keyingi eng so'ngi differentzial nusxa ketma-ket tiklanadi, bu orqali ma'lumotlarni zarur vaqtga qadar tiklash osonlashadi. **Tranzaksiya jurnal nusxasi** (transaction log backup) esa faqat ma'lumotlar bazasining tranzaksiya jurnalidagi so'nggi nusxalashdan beri yozilgan o'zgarishlarni o'z ichiga oladi. Tranzaksiya jurnali butun bazaning barcha o'zgartirishlarini saqlab borishini inobatga olsak, jurnal nusxasi yordamida bazani ma'lum bir vaqtgacha — xato yoki avariya sodir bo'lgan paytga yaqin holatgacha tiklash mumkin. SQL Server'da tranzaksiya jurnali nusxalarini juda tez-tez, masalan har bir necha

daqiqada olish mumkin, bu esa biror noxush holat yuz berganda ma'lumotlarni yo'qotishni minimallashtirishga xizmat qiladi. Tranzaksiya jurnal nusxalari zanjiri buzilmagan bo'lishi sharti bilan, ularni ketma-ket tiklash orqali eng aniq point-in-time (vaqt nuqtasi) tiklash amalga oshiriladi. **Oxirgi jurnal nusxasi** (tail-log backup) ayniqsa muhim — u ma'lumotlar ombori avariya oldin joriy tranzaksiya jurnalining holatini zaxiralash uchun olinadi. Masalan, server qulashidan yoki replikatsiya nosozligidan so'ng darhol oxirgi jurnal nusxasi olinib, tiklash jarayonini minimal ma'lumot yo'qotilgan holda boshlash mumkin.

Copy-only nusxasi – bu to'liq nusxaning maxsus turlari bo'lib, u oddiy nusxalash zanjiriga ta'sir qilmaydi. Bunday nusxa asosan qo'shimcha, tashqi zaxira olish zarur bo'lganda qo'llaniladi. Masalan, agar test maqsadida ma'lumotlar bazasining nusxasini olmoqchi bo'lsak, oddiy differentsial zaxiraga asos bo'lishini istamasak, copy-only full backup qilishimiz mumkin. Copy-only nusxasi oddiy to'liq nusxalarning asosiy (differentsial bazasiga aylanish) oqimiga aralashmagan holda, alohida mustaqil zaxira sifatida saqlanadi. Samarali nusxalash strategiyasini tuzishda quyidagilar e'tiborga olinishi kerak: ma'lumot va jurnal fayllarini iloji boricha alohida fizik disk yoki disk guruhida saqlash tavsiya etiladi, chunki disk nosozligi taqdirida bitta joydagi nuqson butun tizimga ta'sir qildirishi mumkin. Shuningdek, zaxira fayllarini maxfiylik uchun shifrlash va joyni tejash uchun siqib olish mumkin. Zaxiralash jadvali tashkil etib, masalan, haftada bir marta to'liq, har kuni yoki bir necha marta differentsial, hamda soatiga yoki daqiqasiga bir necha marta tranzaksiya jurnal nusxalarini olish orqali biznes talablari asosida tiklash nuqtasi (RPO) vaqti minimallashtiriladi. Buning uchun ma'lumotlar bazasi tiklash rejimini to'g'ri tanlash ham muhimdir: oddiy (simple) rejimda faqat to'liq va differentsial nusxalar olinishi mumkin, tranzaksiya jurnalini zaxiralash qo'llanilmaydi; to'liq (full) rejim esa tranzaksiya jurnal nusxalarini muntazam olib borish orqali ma'lumotlarni istalgan vaqtgacha tiklash imkonini beradi. Zaxiralash strategiyasini ishlab chiqishda, shuningdek, ma'lumotlarni tiklash jarayonini sinash va uni kim bajarishini belgilab qo'yish lozim. Yaxshi strategiya biznes talablariga mos ravishda maksimal ma'lumot mavjudligi va minimal yo'qotishni ta'minlashi lozim. Masalan, tez-tez o'zgarib turuvchi ma'lumotlar uchun to'liq tiklash rejimida kichik intervalda tranzaksiya jurnal nusxalari olinishi tavsiya etiladi; agar ma'lumot kamroq o'zgaradigan bo'lsa, oddiy rejimda muntazam to'liq va differentsial nusxalar bilan ishni ko'rish mumkin. Nusxalar har doim asosiy diskdan alohida joyda saqlanishi kerak – shunga ko'ra, agar asosiy disk ishlamay qolsa, zaxira nusxasi yordamida ma'lumot tiklanadi.

Xulosa

Xulosa qilib aytganda, SQL Server ma'lumotlar bazasida samarali saqlash, indekslash va nusxalashni ta'minlash ma'lumotlar bazasining ishlash tezligi va xavfsizligini belgilovchi asosiy omillardir. Ma'lumotlarni **disk fayllari**, sahifalar va ekstensiyalar orqali tuzilgan mantiqiy bloklarda saqlash, hamda **fileguruh** yordamida fayllarni turli disklar bo'ylab taqsimlash saqlashni optimal qiladi. **Klaster** va **nodavriy indekslar** ma'lumotlarga tezkor kirish imkonini oshirib, so'rovlarni sezilarli darajada tezlashtiradi. Katta hajmdagi analitik so'rovlar uchun **kolonka indeksleri** an'anaviy indekslarga qaraganda yuqori samaradorlik ko'rsatadi. Ma'lumotlarni yo'qotishdan himoya qilish maqsadida esa muntazam va rejalashtirilgan nusxalash strategiyasi tuzilishi zarur. To'liq, differentsial va tranzaksiya jurnal nusxalari yordamida ma'lumotlar kerakli vaqtda tiklanish imkoniyati yaratiladi. Tavsiya etiladiki, ma'lumotlar va jurnal fayllari alohida diskda saqlansin, zaxira nusxalari muntazam olinib, ularning tiklanishini test qilish amalga oshirilsin. Shu yo'l bilan ma'lumotlar bazasi yuqori darajada barqaror, xavfsiz va tez ishlovchi bo'ladi.

Foydalanilgan adabiyotlar

1. Microsoft Learn. *Pages and extents architecture guide*. SQL Server hajmi va sahifalar tuzilishi haqida. [learn.microsoft.com](https://learn.microsoft.com/learn.microsoft.com)
2. Microsoft Learn. *Database files and filegroups*. SQL Server ma'lumot va log fayllari, shuningdek file guruhlar tuzilishi haqida. [learn.microsoft.com](https://learn.microsoft.com/learn.microsoft.com)
3. Microsoft Learn. *Clustered and nonclustered indexes*. Klaster va nodavriy indekslarning ta'riflari va xususiyatlari. [learn.microsoft.com](https://learn.microsoft.com/learn.microsoft.com)
4. Microsoft Learn. *Columnstore indexes – query performance*. Kolonka indeksleri analitik so'rovlarni tezlashtirish va siqish haqida. learn.microsoft.com
5. SQLShack. *Understanding SQL Server Backup Types* (Prashanth Jayaram, 2018). SQL Serverda to'liq, differentsial va jurnal nusxalarining tavsifi. [sqlshack.com](https://sqlshack.com/sqlshack.com)
6. MSP360 Blog. *SQL Server Backup and Restore Best Practices*. To'liq, differentsial va log nusxalarini o'z vaqtida olishning ahamiyati. msp360.com