

TRANSAKTSIYALARDA ACID XOSSALARINING BAJARILISHINI TA'MINLASH MEXANIZMLARI

Tojimamatov Israiljon Nurmamatovich

Farg'ona davlat universiteti

israiltojimatov@gmail.com

Numonova Malika Abduxakim qizi

Farg'ona davlat universiteti

numonovamalika38@gmail.com

Anotatsiya: Ushbu maqolada transaktsiyalarni boshqarish tizimlarida ACID xossalari — ya'ni Atomiklik (Atomicity), Izchillik (Consistency), Izolyatsiya (Isolation) va Barqarorlik (Durability) — bajarilishini ta'minlovchi asosiy mexanizmlar atroflicha tahlil qilinadi. Har bir xossaning nazariy asoslari bilan birga, amaliy jihatlari va zamonaviy ma'lumotlar bazasi boshqaruv tizimlarida ularni amalga oshirishda qo'llaniladigan texnologiyalar o'rganiladi.

Kalit so'zlar: Transaksiya, ACID xossalari, atomiklik, izchillik, izolyatsiya, barqarorlik, ma'lumotlar bazasi, DBMS, blokirovka, jurnal yuritish, versiyalash, tranzaksiyani boshqarish, izolyatsiya darajalari, xavfsizlik, uzluksizlik

Annotation: This article thoroughly analyzes the key mechanisms that ensure the fulfillment of ACID properties — namely Atomicity, Consistency, Isolation, and Durability — in transaction management systems. The study explores both the theoretical foundations of each property and their practical applications, as well as the technologies used to implement them in modern database management systems.

Keywords: Transaction, ACID properties, atomicity, consistency, isolation, durability, database, DBMS, locking, logging, versioning, transaction management, isolation levels, security, availability

Аннотация: В данной статье всесторонне анализируются основные механизмы обеспечения выполнения свойств ACID — атомарности, согласованности, изоляции и долговечности — в системах управления транзакциями. Рассматриваются как теоретические основы каждого свойства, так и практические аспекты их реализации с применением современных технологий в системах управления базами данных.

Ключевые слова: Транзакция, свойства ACID, атомарность, согласованность, изоляция, долговечность, база данных, СУБД, блокировка, ведение журнала, версионирование, управление транзакциями, уровни изоляции, безопасность, доступность.

Transaksiya bu — ma'lumotlar bazasida bajariladigan va yagona mantiqiy amal sifatida ko'riladigan operatsiyalar ketma-ketligidir. Boshqacha aytganda, bir yoki bir nechta SQL buyruqlari transaktsiya deb ataluvchi yagona birlik sifatida bajariladi. Bunda asosiy maqsad, ma'lumotlar bazasida o'zgarishlar kiritilganda tizimni ishonchli va izchil holda saqlab qolishdir.

Transaksiyalar dastlab bank tizimlari kabi moliyaviy operatsiyalarni bajarishda katta ahamiyat kasb etgan. Masalan, bir hisobdan pul yechib, boshqasiga o'tkazish jarayonida ikkala qadam ham muvaffaqiyatli bajarilishi yoki butunlay bekor qilinishi lozim. Aks holda, pul noto'g'ri o'tkaziladi yoki umuman yo'qolib ketishi mumkin. Shunday nozik vaziyatlarda transaktsiyalarning ishonchiligi va xatolarga bardoshliligi eng muhim

masalaga aylanadi. Bu esa, o'z navbatida, transaktsiyalarning sifatli ishlashini ta'minlovchi ACID xossalari chuqur o'rganishni talab etadi. ACID atamasi ingliz tilidagi "Atomicity", "Consistency", "Isolation" va "Durability" so'zlarining bosh harflaridan tashkil topgan bo'lib, transaktsiyalarni bajarish jarayonida ma'lumotlarning ishonchli saqlanishini ta'minlab beradi. Har bir xossa o'ziga xos maqsadga ega va ular birgalikda ishlashi orqali transaktsiya natijasi har doim aniq va ishonchli bo'lishiga erishiladi.

Atomicity – atomiklik degani, transaktsiya doirasidagi barcha amallar butunlay bajarilishi yoki umuman bajarilmasligi kerakligini anglatadi. Bu xususiyat ma'lumotlar bazasida noto'liq yoki yarim bajarilgan o'zgarishlarning oldini oladi.

Consistency – izchillik esa, transaktsiya yakunlangach, ma'lumotlar bazasi o'zining oldindan belgilangan mantiqiy holatiga mos bo'lishini ta'minlaydi. Ya'ni, transaktsiya boshlanishidan oldin bazada qanday mantiqiy bog'lanishlar va qoidalar mavjud bo'lsa, u yakunlangach ham ular buzilmasligi lozim.

Isolation – izolyatsiya xossasi esa bir vaqtning o'zida bir nechta foydalanuvchi yoki tizim transaktsiyalarini mustaqil tarzda bajarishga imkon beradi. Bunda har bir transaktsiya boshqalardan mustaqil ravishda bajariladi, ya'ni ularning o'zaro ta'siri kamaytiriladi.

Durability – mustahkamlik degani esa, transaktsiya muvaffaqiyatli yakunlangach, undagi barcha o'zgarishlar ma'lumotlar bazasida doimiy saqlanishini bildiradi. Hatto tizimda elektr uzilishi, server nosozligi yoki boshqa texnik muammolar yuz bersa ham, oldin bajarilgan va saqlangan o'zgarishlar yo'qolmaydi. Bu xossa ko'pincha maxsus log fayllari, zaxira nusxalari yoki jurnal tizimlari yordamida ta'minlanadi. Mustahkamlik tufayli foydalanuvchilar ma'lumotlarning doimiyligini ishonch bilan kafolatlashlari mumkin bo'ladi.

Shunday qilib, ACID xossalari birgalikda ishlaganida transaktsiyaning mukammal bajarilishi uchun zarur bo'lgan barcha shart-sharoitlarni yaratadi. Har bir xossa alohida muhim bo'lsa-da, ular faqat birgalikda transaktsiyani to'liq va xavfsiz bajarish uchun xizmat qiladi. Hozirgi zamon dasturiy tizimlarida, ayniqsa moliyaviy, sog'liqni saqlash, onlayn xarid qilish va ijtimoiy tarmoqlar kabi sohalarda minglab foydalanuvchilar bir vaqtning o'zida tizim bilan ishlaydi. Bunday holatlarda har bir foydalanuvchining ma'lumotlari o'zaro chalkashmasligi, noto'g'ri o'zgarishlar kiritilmasligi va tizim ishdan chiqsa ham ma'lumotlar tiklanishi juda muhimdir. Shu sababli ACID xossalari transaktsiyalar uchun fundamental mezon sifatida qaraladi va har bir jiddiy ma'lumotlar bazasi boshqaruv tizimi ushbu xossalarning bajarilishini ta'minlaydigan mexanizmlarni o'z ichiga oladi. Zamonaviy ma'lumotlar bazasi boshqaruv tizimlari, masalan, Microsoft SQL Server, Oracle Database, MySQL, PostgreSQL va boshqalar, ACID xossalari saqlash uchun turli texnikalar va arxitekturalarni taklif qilgan.

Atomiklikni ta'minlash uchun zamonaviy ma'lumotlar bazasi boshqaruv tizimlari turli texnik vositalar va mexanizmlardan foydalanadi. Eng asosiy va keng tarqalgan mexanizmlardan biri - bu loglash tizimidir. Loglash (yoki yozuv jurnalini yuritish) jarayoni ma'lumotlar bazasida bajarilgan har bir o'zgarish haqida alohida log faylga yozib borishni anglatadi. Bu log fayllar, ayniqsa tranzaksiyalar bo'yicha yozuvlarni saqlaydi. Transaktsiya boshlanganda, amalga oshirilayotgan har bir amal haqida ma'lumot log faylga yoziladi. Agar transaktsiya muvaffaqiyatli yakunlansa, log faylga COMMIT yozuvi qo'shiladi. Agar esa transaktsiya bajarilayotgan vaqtda tizimda xatolik yuz bersa, log fayldagi ma'lumotlar asosida transaktsiya davomida kiritilgan barcha o'zgarishlar ortga qaytariladi. Bu jarayon "ROLLBACK" deb ataladi.

Atomiklikni ta'minlashda yana bir usul — bu transaction manager deb ataluvchi mexanizmdir. Transaction manager — bu ma'lumotlar bazasida transaksiyalarni kuzatib boradigan va ularning holatini nazorat qiladigan modul yoki tizimdir. Ushbu mexanizm orqali har bir transaksiyaning boshlanishi, davom etishi va tugash bosqichlari aniqlanadi. Ular orqali BEGIN, COMMIT, va ROLLBACK amallari muvofiqlashtiriladi. Transaction manager tizimdagi barcha transaksiyalarni yagona tartibda boshqaradi va ularning o'zaro to'qnashmasligini ta'minlaydi. Agar biror transaksiya noto'g'ri bajarilsa yoki foydalanuvchi tomonidan bekor qilinsa, manager avtomatik tarzda unga bog'liq bo'lgan o'zgarishlarni orqaga qaytaradi. Bu jarayon avtomatlashtirilgan va foydalanuvchidan maxsus harakat talab qilmaydi.

Izchillikni ta'minlashda birinchi navbatda ma'lumotlar bazasi modelining to'g'ri va puxta ishlab chiqilishi muhim rol o'ynaydi. Bu degani — bazadagi har bir jadval, ustun, cheklov (constraints), kalit (keys) va bog'lanishlar (relationships) o'zaro mantiqiy tarzda uyg'unlashtirilgan bo'lishi kerak. Buning uchun PRIMARY KEY, FOREIGN KEY, NOT NULL, UNIQUE, va CHECK kabi cheklovlar yordamida bazaning strukturasi qat'iy nazorat ostida bo'ladi. Masalan, FOREIGN KEY yordamida ikki jadval o'rtasida bog'liqlik o'rnatiladi va bu bog'liqlikni buzuvchi har qanday transaksiya avtomatik ravishda bekor qilinadi. Bu bilan izchillik saqlanib qoladi.

Izchillikka erishishda ma'lumotlar bazasining tranzaksiyalarga oid isolation darajalari ham bevosita ta'sir ko'rsatadi. Izolyatsiya darajalari transaksiyalar bir-biriga qanday ta'sir qilishini belgilaydi. Agar bir transaksiya hali tugallanmagan bo'lsa va u boshqa transaksiyalarga ko'rinadigan bo'lsa, bu holatda "dirty read" yoki noto'g'ri o'qish yuz beradi. Bu esa izchillikni buzadi. Shu sababli ko'plab bazalarda SERIALIZABLE, REPEATABLE READ, READ COMMITTED va READ UNCOMMITTED kabi izolyatsiya darajalari mavjud bo'lib, ular transaksiyalar o'rtasidagi to'qnashuvlarning oldini olish orqali izchillikni saqlashga xizmat qiladi.

Zamonaviy ma'lumotlar bazasida izchillikni saqlash faqat ichki mexanizmlar bilan cheklanib qolmaydi, balki foydalanuvchi interfeysi orqali kiritilayotgan ma'lumotlarning ham tekshirilishini talab qiladi. Bu degani — foydalanuvchi ma'lumot kiritayotgan vaqtda xatolik yuz berishining oldini olish kerak. Masalan, forma validatsiyasi orqali foydalanuvchidan noto'g'ri email yoki manfiy son kiritilishi cheklanadi. Bu dasturiy qatlamda bajariladigan izchillikni ta'minlash usullaridan biridir va u umumiy tizimning barqarorligiga bevosita ta'sir qiladi.

Izolyatsiya (Isolation) — bu ma'lumotlar bazasida bir vaqtda bir nechta transaksiyalar bajarilayotgan holatlarda ularning bir-biriga ta'sirini boshqarish, har bir transaksiyaning mustaqilligini ta'minlash tamoyilidir. ACID xossalarining uchinchi elementi bo'lgan izolyatsiya, ayniqsa ko'p foydalanuvchili muhitda, ya'ni parallel ishlovchi axborot tizimlarida nihoyatda muhim hisoblanadi. U orqali har bir transaksiyaning boshqa transaksiyalar tomonidan ta'sirga uchramasdan o'z ishini yakunlashiga kafolat beriladi. Bunday izolyatsiyaning yo'qligi tizimda noaniqliklar, noto'g'ri natijalar yoki mantiqsiz holatlarning yuzaga kelishiga sabab bo'lishi mumkin.

Izolyatsiya konsepsiyasining chuqur mohiyati shundaki, tizim bir vaqtning o'zida bir nechta foydalanuvchining transaksiyalarini boshqarishi, ammo ularni bir-biridan mustaqil tarzda amalga oshirishi kerak. Masalan, agar bir foydalanuvchi mahsulot narxini yangilayotgan bo'lsa, boshqa foydalanuvchi shu mahsulotga buyurtma berayotgan bo'lishi mumkin. Agar izolyatsiya ta'minlanmasa, birinchi foydalanuvchi hali o'z transaksiyasini yakunlamay turib, ikkinchi foydalanuvchining transaksiyasi yangilanayotgan narxni o'qib yuboradi. Bu holatda "dirty read" deb ataluvchi hodisa yuz

beradi. Eng past daraja — READ UNCOMMITTED bo'lib, bu darajada transaksiyalar bir-birining hali yakunlanmagan o'zgarishlarini ham ko'ra oladi. Bu daraja eng tezkor ishlov beradi, ammo eng past izchillikni ta'minlaydi. READ COMMITTED darajasida esa transaksiyalar faqat boshqa transaksiyalar tomonidan allaqachon COMMIT qilingan o'zgarishlarni ko'ra oladi, ya'ni "dirty read" holati bartaraf etiladi. Bu ko'proq amaliy tizimlarda ishlatiladigan daraja bo'lib, izchillik va unumdorlik o'rtasida muvozanatni saqlaydi.

Yana bir daraja — REPEATABLE READ, u takroran o'qish holatlarida barqaror natijani kafolatlaydi. Bu darajada transaksiya biror ma'lumotni birinchi marta o'qiganidan so'ng boshqa transaksiyalar bu qiymatni o'zgartira olmaydi. Ya'ni, biror transaksiya bir xil ma'lumotni necha marta o'qimasin, har safar aynan bir xil qiymatni ko'radi. Bu orqali "non-repeatable read" deb ataladigan holat oldi olinadi. Eng yuqori daraja esa SERIALIZABLE bo'lib, u barcha transaksiyalarni ketma-ketlikda bajarayotgandek simulyatsiya qiladi. Bu daraja eng yuqori izchillikni ta'minlaydi, biroq tizimning unumdorligiga salbiy ta'sir ko'rsatadi, chunki transaksiyalar o'rtasidagi blokirovka holatlari ko'p bo'ladi.

Parallel transaksiyalarni boshqarishda blokirovka (locking) mexanizmi muhim o'rin tutadi. Ma'lumotlar bazasi biror transaksiya ma'lumotni o'zgartirishga uringanda, bu ma'lumot ustidan "lock" qo'yadi. Bu boshqa transaksiyalarning shu ma'lumotga murojaat qilishini vaqtincha to'xtatadi. Blokirovka ikki xil bo'ladi: o'qish uchun va yozish uchun. O'qish blokirovkasi boshqa o'qishlarni to'xtatmaydi, lekin yozishga ruxsat bermaydi; yozish blokirovkasi esa boshqa har qanday operatsiyani cheklaydi. Blokirovka mexanizmi orqali "phantom read", "dirty read", va "non-repeatable read" holatlarining oldi olinadi. Biroq haddan tashqari ko'p blokirovkalar "deadlock" holatiga olib kelishi mumkin, ya'ni ikkita yoki undan ortiq transaksiya bir-birining bloklarini kutib qoladi va hech biri yakunlanmaydi. Bu holatda ma'lumotlar bazasi avtomatik tarzda transaksiyalardan birini bekor qiladi.

Izolyatsiya faqat texnik mexanizmlar bilan emas, balki tizim arxitekturasini va dasturlash amaliyotlari orqali ham ta'minlanadi. Masalan, Unit of Work yoki Repository dizayn naqshlari yordamida transaksiyalarni ma'lum bir kontekst ichida boshqarish mumkin. Dasturiy darajada transaksiyalarni faollashtirish, izolyatsiya darajalarini sozlash va kerakli hollarda BEGIN TRANSACTION, COMMIT, yoki ROLLBACK amallarini to'g'ri ishlatish orqali izolyatsiyaga erishiladi. Shuningdek, foydalanuvchi interfeysida ham bir vaqtning o'zida ikki foydalanuvchi bir ma'lumotni tahrir qilayotgan bo'lsa, bu holatni aniqlab, ularga ogohlantirish berish mumkin. Zamonaviy ma'lumotlar bazalarida izolyatsiya nafaqat blokirovka asosida, balki versiyalash (MVCC — Multi-Version Concurrency Control) texnologiyasi orqali ham ta'minlanadi. Bu texnologiyada har bir yozuvning bir necha versiyasi saqlanadi va har bir transaksiya o'z versiyasi bilan ishlaydi. Bu orqali blokirovka ehtiyoji kamayadi va tizimning parallel ishlash qobiliyati oshadi. MVCC PostgreSQL, Oracle va ba'zi boshqa zamonaviy MBTlarda keng qo'llaniladi. Bu usul yordamida foydalanuvchilar bir-biriga xalaqit bermasdan yozuvlarni o'qishlari va yozishlari mumkin, va ayni paytda izolyatsiya darajasi yuqori darajada ta'minlanadi.

Transaksiyalar bilan ishlovchi har qanday axborot tizimi uchun ACID xossalari - ya'ni Atomiklik, Izchillik, Izolyatsiya va Barqarorlik — eng muhim tamoyillardan hisoblanadi. Ushbu xossalarni foydalanuvchi tomonidan bajariladigan har bir operatsiyaning aniqligi, ishonchliligi va butun tizim barqarorligini kafolatlashga xizmat qiladi. Atomiklik orqali transaksiya ichidagi barcha amallar yakunlanadi yoki hech biri

bajarilmaydi, bu esa ma'lumotlar bazasida yarim holatlarning oldini oladi. Izchillik xossasi ma'lumotlar bazasining qat'iy qoidalar asosida ishlashini ta'minlab, ma'lumotlar yaxlitligini saqlaydi. Izolyatsiya esa parallel transaktsiyalarning bir-biriga ta'sir qilmasligini kafolatlab, ko'p foydalanuvchili tizimlarda to'g'ri ishlashni ta'minlaydi. Barqarorlik esa transaktsiya muvaffaqiyatli yakunlangach, uning natijalari doimiy ravishda saqlanib. Bularning har biri o'z vazifasini bajarib, tizimning ishonchli va uzluksiz ishlashini kafolatlaydi. Shu bilan birga, real hayotda bu xossalarni muvozanatda ushlab zarurati yuzaga keladi, chunki haddan tashqari qat'iy izolyatsiya samaradorlikni pasaytirishi mumkin, yoki yuqori samaradorlik esa ba'zan barqarorlikka xavf tug'dirishi mumkin.

Transaktsiyalarda ACID xossalarning bajarilishini ta'minlash — bu faqat texnik muammo emas, balki murakkab muvozanatli boshqaruv vazifasidir. Har bir dasturchi va ma'lumotlar bazasi mutaxassisi bu tamoyillarni chuqur tushunishi, ularni amaliyotda to'g'ri qo'llay olishi, tizim ehtiyojlariga qarab optimal yondashuvni tanlashi lozim. Aynan shu bilim va tajriba zamonaviy axborot tizimlarining ishonchliligi, xavfsizligi va foydalanuvchi ishonchini ta'minlashga xizmat qiladi.

Foydalanilgan adabiyotlar:

1. Silberschatz, A., Korth, H.F., & Sudarshan, S. Database System Concepts. McGraw-Hill, 7th Edition, 2020.
2. Elmasri, R., & Navathe, S.B. Fundamentals of Database Systems. Pearson, 7th Edition, 2016.
3. Date, C.J. An Introduction to Database Systems. Addison-Wesley, 8th Edition, 2003.
4. Coronel, C., & Morris, S. Database Systems: Design, Implementation, and Management. Cengage Learning, 13th Edition, 2019.
5. Thomas Connolly, Carolyn Begg. Database Systems: A Practical Approach to Design, Implementation, and Management. Pearson, 6th Edition, 2015.
6. Garcia-Molina, H., Ullman, J.D., & Widom, J. Database Systems: The Complete Book. Pearson, 2nd Edition, 2008.
7. Jim Gray, Andreas Reuter. Transaction Processing: Concepts and Techniques. Morgan Kaufmann, 1993.
8. Nurmatovich, T. I., & Azizjon o'g, N. A. Z. (2024). The SQL server language and its structure. American Journal of Open University Education.
9. Nurmatovich, T. I. (2024). MY SQL MISOLIDA LOYIHA YARATISH. Ta'limda raqamli texnologiyalarni tadbqiq etishning zamonaviy tendensiyalari va rivojlanish omillari