



SQL-TARTIBLANGAN SO'ROVLAR TILINING ASOSIY BUYRUQLARI, ULARDAN FOYDALANISH TARTIBI.

Alisher Ismailov

alisherismailov1991@gmail.com

Choriyev Mahmudjon

choriyevmahmudjon658@gmail.com

Toshkent Davlat Iqtisodiyot Universiteti

<https://doi.org/10.5281/zenodo.18889189>

Annotatsiya:

Ushbu mavzuda SQL (Structured Query Language)-ma'lumotlar bazasi bilan ishlash uchun mo'ljallangan asosiy so'rovlar tili haqida tushuncha beriladi. Unda SQL tilining asosiy buyruqlari, jumladan **SELECT, INSERT, UPDATE, DELETE, CREATE, ALTER, DROP** kabi operatorlarning vazifalari va ulardan foydalanish tartibi yoritiladi. Shuningdek, ma'lumotlar bazasida jadval yaratish, ma'lumot qo'shish, o'zgartirish, o'chirish hamda kerakli ma'lumotlarni saralab olish jarayonlari misollar orqali tushuntiriladi. Ushbu mavzu orqali foydalanuvchilar SQL buyruqlarining sintaksisi va ularni amaliyotda to'g'ri qo'llash ko'nikmalariga ega bo'ladilar.

Kalit so'zlar:

SQL (Structured Query Language), Ma'lumotlar bazasi (Database), Jadval (Table), DDL (Data Definition Language).

Zamonaviy axborot texnologiyalari dunyosida ma'lumotlar eng qimmatli resurslardan biriga aylandi. Har kuni milliardlab foydalanuvchilar internet orqali turli xizmatlardan foydalanadilar, xarid qiladilar, ijtimoiy tarmoqlarda muloqot qiladilar va boshqa ko'plab amallarni bajaradilar. Bu jarayonlarning har birida katta hajmdagi ma'lumotlar yaratiladi, saqlanadi va qayta ishlanadi. Ushbu ma'lumotlarni tartibli, xavfsiz va samarali saqlash hamda ulardan kerakli paytda tez foydalana bilish-zamonaviy dasturiy ta'minot ishlab chiqarishning asosiy muammolaridan biridir.

Aynan shu ehtiyojni qondirish maqsadida 1970-yillarda relatsion ma'lumotlar bazasi tushunchasi va uni boshqarish uchun maxsus til-SQL (Structured Query Language, ya'ni Tuzilgan So'rovlar Tili) ishlab chiqildi. O'sha davrdan beri SQL dunyodagi eng keng tarqalgan va eng ko'p ishlatiladigan dasturlash tillari qatorida turib kelmoqda. Bugungi kunda MySQL, PostgreSQL, Microsoft SQL Server, Oracle, SQLite kabi o'nlab mashhur ma'lumotlar bazasi tizimlari SQL tilida ishlaydi va bu til amalda hech qanday muqobilsiz standartga aylangan.

DDL-Ma'lumotlar Strukturasini Boshqarish Buyruqlari

DDL (Data Definition Language-Ma'lumotlarni Aniqlash Tili)-bu ma'lumotlar bazasining strukturasini, ya'ni jadvallar, indekslar, ko'rinishlar va boshqa ob'ektlarni yaratish, o'zgartirish va o'chirishga mo'ljallangan buyruqlar to'plamidir. DDL buyruqlari bajarilgandan so'ng o'zgarishlar darhol kuchga kiradi va odatda qaytarib bo'lmaydi, shuning uchun ularni ehtiyotkorlik bilan ishlatish zarur.

CREATE Buyrug'i

CREATE buyrug'i yangi ma'lumotlar bazasi ob'ektlarini yaratish uchun ishlatiladi. Bu ob'ektlar orasida eng ko'p ishlatiladigan-jadval (TABLE) yaratish hisoblanadi. Jadval yaratishda uning nomi, ustunlar ro'yxati, har bir ustunning ma'lumot turi va cheklovlar (constraints) ko'rsatiladi. Masalan, talabalar jadvalini yaratishda id (birlamchi kalit), ism, familiya, tug'ilgan sana, manzil kabi ustunlar aniqlanadi. Birlamchi kalit (PRIMARY KEY) jadvalning har bir qatorini yagona tarzda identifikatsiya qiladi va takrorlanmasligi hamda bo'sh bo'lmasligi kafolatlaydi. Tashqi kalit (FOREIGN KEY) esa ikki jadval o'rtasida mantiqiy bog'liqlik o'rnatadi va ma'lumotlar yaxlitligini ta'minlaydi.

CREATE buyrug'i bilan nafaqat jadvallar, balki ma'lumotlar bazasi (DATABASE), ko'rinishlar (VIEW), saqlanadigan protsedurar (STORED PROCEDURE), triggerlar (TRIGGER), indekslar (INDEX) va boshqa ob'ektlar ham yaratiladi. Har bir ob'ekt turi o'ziga xos sintaksisga ega bo'lib, lekin ularning barchasi CREATE kalit so'zi bilan boshlanadi.

ALTER Buyrug'i

ALTER buyrug'i mavjud ob'ektlarning strukturasini o'zgartirish uchun mo'ljallangan. Jadvalga yangi ustun qo'shish, mavjud ustunni o'chirish, ustun nomini yoki ma'lumot turini o'zgartirish, cheklov qo'shish yoki olib tashlash-bularning barchasi ALTER TABLE buyrug'i orqali amalga oshiriladi. Katta hajmdagi jadvallar uchun ALTER operatsiyasi uzoq vaqt olishi mumkin, chunki ba'zi hollarda tizim barcha mavjud qatorlarni qayta ko'rib chiqishi zarur bo'ladi. Shuning uchun ishlab chiqarish muhitlarida (production environment) ALTER buyrug'larini qo'llashda jadval qulflanish muddatini hisobga olish lozim.

DROP va TRUNCATE Buyruqlari

DROP buyrug'i ob'ektni butunlay o'chirib tashlaydi-jadval ham, uning ichidagi barcha ma'lumotlar ham butunlay yo'qoladi. Bu operatsiya qaytarib bo'lmaydi va juda ehtiyotkorlik talab qiladi. Ishlab chiqarish bazalarida DROP ni tasodifan ishlatib yubormaslik uchun maxsus ruxsat mexanizmlari qo'llaniladi.

TRUNCATE buyrug'i DROP dan farqli ravishda faqat jadvaldagi ma'lumotlarni o'chiradi, jadvalning o'zini esa saqlab qoladi. U DELETE buyrug'iga

o'xshashaydi, lekin ancha tezroq ishlaydi, chunki har bir qatorni alohida qayd etmaydi. TRUNCATE ham odatda qaytarib bo'lmaydigan operatsiya hisoblanadi.

DML-Ma'lumotlar bilan ishlash buyruqlari

DML (Data Manipulation Language-Ma'lumotlarni Boshqarish Tili)-bu mavjud jadvallar ichidagi ma'lumotlarni qo'shish, o'qish, yangilash va o'chirish uchun mo'ljallangan buyruqlar guruhi. SQL dasturchilarning kundalik ishida eng ko'p murojaat qiladigan buyruqlari aynan DML ga tegishli. DML buyruqlari tranzaksiya doirasida bajarilishi mumkin, bu esa xato bo'lgan taqdirda o'zgarishlarni bekor qilish imkonini beradi.

INSERT - Ma'lumot qo'shish

INSERT INTO buyrug'i jadvalga yangi qatorlar qo'shish uchun ishlatiladi. Eng oddiy shaklda bitta qator qo'shiladi: ustunlar ro'yxati va ularga mos qiymatlar VALUES kalit so'zi bilan ko'rsatiladi. Agar barcha ustunlarga qiymat berilayotgan bo'lsa, ustunlar ro'yxatini ko'rsatmaslik ham mumkin, lekin bu usul xavfli-kelajakda jadval tuzilishi o'zgarsa, so'rov ishlamay qolishi mumkin. Shuning uchun har doim ustunlar ro'yxatini aniq ko'rsatish tavsiya etiladi.

INSERT buyrug'ining kengaytirilgan shakli bir vaqtda bir nechta qator qo'shish imkonini beradi-bu yagona VALUES blokida vergul bilan ajratilgan bir nechta qiymatlar to'plami sifatida yoziladi. Bundan tashqari, INSERT INTO ... SELECT konstruksiyasi boshqa jadvaldan olingan ma'lumotlarni yangi jadvalga ko'chirish uchun ishlatiladi, bu esa katta hajmdagi ma'lumotlarni ko'chirish jarayonini tezlashtiradi.

SELECT – Ma'lumot o'qish

SELECT buyrug'i SQL ning eng muhim va eng ko'p ishlatiladigan buyrug'idir. U jadvaldan ma'lumotlarni o'qib olish uchun xizmat qiladi. SELECT ning asosiy qismlari quyidagilar: FROM (qaysi jadvaldan o'qilsin), WHERE (qanday shart bo'yicha filtrlansin), GROUP BY (qanday guruhlangin), HAVING (guruhlar qanday filtrlansin), ORDER BY (qanday tartiblansin) va LIMIT (nechta qator qaytarilsin). Bu qismlarning har biri ixtiyoriy bo'lib, faqat SELECT va FROM majburiy hisoblanadi.

WHERE shartida turli operatorlar ishlatilishi mumkin: tenglik (=), tengsizlik (<>, !=), taqqoslash (<, >, <=, >=), oraliq (BETWEEN), ro'yxatda borligini tekshirish (IN), bo'sh qiymatni tekshirish (IS NULL), matn namunasiga moslashish (LIKE). LIKE operatorida foiz belgisi (%) nol yoki undan ko'p belgilarni, pastki chiziq (_) esa aynan bitta belgini ifodalaydi. Bu operatorlarni AND, OR, NOT mantiqiy operatorlari bilan birlashtirib, murakkab filtr shartlarini yozish mumkin.



ORDER BY qismi natijalarni bir yoki bir nechta ustun bo'yicha tartiblash imkonini beradi. ASC (ascending-o'sish tartibida) sukut bo'yicha ishlaydi, DESC (descending-kamayish tartibida) esa qarama-qarshi tartibni beradi. Bir nechta ustun bo'yicha tartiblashda ular vergul bilan ajratiladi: avval birinchi ustun bo'yicha, so'ng teng qiymatlar ichida ikkinchi ustun bo'yicha tartiblanadi.

UPDATE – Ma'lumot yangilash

UPDATE buyrug'i mavjud qatorlardagi qiymatlarni o'zgartirish uchun ishlatiladi. SET kalit so'zi bilan o'zgartirilayotgan ustunlar va ularning yangi qiymatlari ko'rsatiladi, WHERE bilan esa qaysi qatorlar o'zgartirilishi aniqlanadi. WHERE shartini ko'rsatmaslik jiddiy xato hisoblanadi-bunday holda jadvalning barcha qatorlari yangilanib ketadi. Shuning uchun UPDATE ni ishlatishdan oldin avval SELECT bilan WHERE shartini tekshirib ko'rish yaxshi amaliyot hisoblanadi.

UPDATE buyrug'ida SET qismida hisob-kitob ifodalaridan foydalanish ham mumkin. Masalan, narxni 10 foizga oshirish uchun $\text{narx} = \text{narx} * 1.10$ yoziladi, yoki hisoblagichni bittaga oshirish uchun $\text{counter} = \text{counter} + 1$ ishlatiladi. Bundan tashqari, UPDATE ... JOIN konstruksiyasi boshqa jadval ma'lumotlariga asoslanib yangilash imkonini beradi.

DELETE – Ma'lumot o'chirish

DELETE FROM buyrug'i jadvaldan qatorlarni o'chirish uchun mo'ljallangan. UPDATE kabi, bu yerda ham WHERE sharti juda muhim-WHERE siz barcha qatorlar o'chiriladi. O'chirishdan oldin avval SELECT bilan nechta qator o'chirilishini tekshirish tavsiya etiladi. DELETE tranzaksiya ichida bajarilsa, ROLLBACK buyrug'i bilan bekor qilish mumkin, bu esa tasodifiy o'chirishlardan himoyalash imkonini beradi.

JOIN – Jadvallarni birlashtirish

Relation ma'lumotlar bazalarida ma'lumotlar normalizatsiya tamoyili asosida bir nechta jadvallarga bo'lib saqlanadi. Bu takrorlanishni kamaytiradi va ma'lumotlar yaxlitligini saqlaydi. Ammo so'rov natijasida turli jadvallardan ma'lumotlarni birlashtirib ko'rsatish kerak bo'lganda JOIN operatsiyalari qo'l keladi. JOIN -bu ikki yoki undan ortiq jadvallarni ularning bir-biriga mos keladigan ustunlari orqali birlashtirish amali.

INNER JOIN-eng ko'p ishlatiladigan birlashtirish turi. U faqat ikki jadvalda ham mos qiymat mavjud bo'lgan qatorlarni qaytaradi. Agar bir jadvalda mos qiymat bo'lmasa, o'sha qator natijaga kirmaydi. Masalan, talabalar va kurslar jadvalini birlashtiranda faqat biror kursga yozilgan talabalar ko'rsatiladi.

LEFT JOIN (yoki LEFT OUTER JOIN)-chap jadvaldagi barcha qatorlarni va o'ng jadvaldagi mos qatorlarni qaytaradi. O'ng jadvalda mos qiymat bo'lmasa,



o'sha ustunlarda NULL qiymati ko'rsatiladi. Bu, masalan, hech qaysi kursga yozilmagan talabalarni ham ro'yxatga kiritish kerak bo'lganda ishlatiladi. RIGHT JOIN esa buning aksi-o'ng jadvaldagi barcha qatorlarni qaytaradi

FULL OUTER JOIN ikki jadvalning barcha qatorlarini birlashtiradi-mos qiymat bo'lmagan joylarda NULL ko'rsatiladi. CROSS JOIN esa birinchi jadvalning har bir qatorini ikkinchi jadvalning har bir qatori bilan birlashtiradi (kartezian ko'paytma), natijada qatorlar soni birinchi jadval qatorlari soni bilan ikkinchi jadval qatorlari sonining ko'paytmasiga teng bo'ladi.

Agregat funksiyalar va guruhlash

SQL da agregat funksiyalar bir nechta qatorlar bo'yicha hisob-kitob natijasini bitta qiymat sifatida qaytaradi. Ular ko'pincha GROUP BY bilan birgalikda ishlatiladi. Asosiy agregat funksiyalar quyidagilardir: COUNT()-qatorlar sonini hisoblaydi; SUM()-sonli qiymatlar yig'indisini topadi; AVG()-o'rtacha qiymatni hisoblaydi; MAX()-eng katta qiymatni topadi; MIN() — eng kichik qiymatni topadi.

GROUP BY buyrug'i ma'lumotlarni bir yoki bir nechta ustun bo'yicha guruhlaydi va har guruh uchun agregat funksiya natijasini beradi. Masalan, har bir shahardagi talabalar sonini topish uchun shahar bo'yicha guruhlash va COUNT() funksiyasidan foydalaniladi. HAVING qismi esa WHERE ga o'xshab ishlaydi, lekin u guruhlar uchun filtr o'rnatadi. WHERE agregat funksiyalar bilan ishlamaydi, HAVING esa ishlaydi.

Agregat funksiyalarda DISTINCT kalit so'zini ishlatish mumkin-bu takrorlanuvchi qiymatlarni hisobga olmaslik imkonini beradi. Masalan, COUNT(DISTINCT shahar) bir jadvalda nechta har xil shahar borligini hisoblaydi, COUNT(*) esa barcha qatorlarni sanaydi. NULL qiymatlar COUNT(*) dan tashqari barcha agregat funksiyalar tomonidan e'tiborga olinmaydi.

Kichik so'rovlar

Kichik so'rov (subquery yoki ichki so'rov)-bu boshqa so'rovning ichida joylashgan SQL so'rovidir. U qavslar ichida yoziladi va tashqi so'rovga qiymat yoki qiymatlar to'plami beradi. Kichik so'rovlar SELECT, INSERT, UPDATE, DELETE buyruqlarida ishlatilishi mumkin, shuningdek, WHERE, FROM, HAVING, SELECT qismlarida ham uchraydi.

Masalan, o'rtacha yoshdan katta bo'lgan barcha talabalarni topish uchun WHERE yosh > (SELECT AVG(yosh) FROM talabalar) kichik so'rovidan foydalaniladi. Bu yerda ichki so'rov avval o'rtacha yoshni hisoblaydi, so'ng tashqi so'rov shu qiymatdan katta yoshli talabalarni filtrlaydi. Kichik so'rovlar EXISTS



operatori bilan birgalikda ham keng qo'llaniladi -u kichik so'rov kamida bitta qator qaytarsa TRUE, qaytarmasa FALSE beradi.

Korrelyatsiyalangan kichik so'rovlar (correlated subquery) tashqi so'rovning har bir qatori uchun qayta bajariladi. Bu ularni sekinroq qiladi, lekin ba'zi muammolarni boshqa yo'l bilan hal qilib bo'lmaydi. Katta hajmdagi ma'lumotlarda korrelyatsiyalangan kichik so'rovlar o'rniga JOIN ishlatish samaraliroq hisoblanadi.

Tranzaksiyalar va ma'lumotlar yaxlitligi

Tranzaksiya -bu bir butun sifatida bajarilishi yoki butunlay bekor qilinishi kerak bo'lgan SQL operatsiyalari ketma-ketligidir. Masalan, bank o'tkazmasi ikki operatsiyadan iborat: birinchi hisobdan pul yechish va ikkinchi hisobga pul qo'shish. Agar birinchi operatsiya muvaffaqiyatli bajarilsa, lekin ikkinchisida xato yuz bersa, pul yo'qolib ketmasligi uchun birinchi operatsiya ham bekor qilinishi kerak. Aynan shu maqsadda tranzaksiyalar xizmat qiladi.

Tranzaksiyalar ACID tamoyillariga asoslanadi: Atomarlik (Atomicity) - tranzaksiya yo to'liq bajariladi, yo umuman bajarilmaydi; Izchillik (Consistency) - tranzaksiya ma'lumotlar bazasini bir izchil holatdan boshqa izchil holatga o'tkazadi; Izolyatsiya (Isolation) -parallel tranzaksiyalar bir-biriga ta'sir qilmaydi; Chidamlilik (Durability)-muvaffaqiyatli tugagan tranzaksiya natijalari doimiy saqlanadi.

BEGIN TRANSACTION (yoki START TRANSACTION) buyrug'i tranzaksiyani boshlaydi, COMMIT barcha o'zgarishlarni saqlaydi, ROLLBACK esa tranzaksiya boshlanganidan beri qilingan barcha o'zgarishlarni bekor qiladi. SAVEPOINT orqali tranzaksiya ichida oraliq nuqtalar o'rnatish mumkin va ROLLBACK TO SAVEPOINT bilan faqat ma'lum nuqtaga qadar qaytish imkoniyati mavjud.

Indekslar va Optimizatsiya

Indeks - bu ma'lumotlar bazasida qidiruv tezligini oshirish uchun mo'ljallangan maxsus tuzilma. Kitobdagi mazmun sahifasiga o'xshab, indeks ma'lumotlar bazasi tizimiga kerakli qatorlarni butun jadvalni ko'zdan kechirmasdan tez topish imkonini beradi. Indekssiz so'rovlarda tizim jadvalning barcha qatorlarini ketma-ket ko'rib chiqadi (full table scan) - bu kichik jadvallar uchun muammo bo'lmasa-da, millionlab qatorli jadvallarda jiddiy sekinlashuvga olib keladi.

Indekslar CREATE INDEX buyrug'i bilan yaratiladi. Ko'p ishlatiladigan WHERE shartlari, JOIN birlashtirishlari va ORDER BY tartiblashlarida ishtirok etuvchi ustunlarga indeks qo'yish so'rovlarni sezilarli tezlashtiradi. Ammo indekslar ham disk joyini egallaydi va INSERT, UPDATE, DELETE operatsiyalarini



biroz sekinlashtiradi, chunki ma'lumot o'zgarganda indeks ham yangilanishi kerak. Shuning uchun barcha ustunlarga emas, balki haqiqatan keraklilarga indeks qo'yiladi.

So'rov bajarilish rejasini ko'rish uchun EXPLAIN (PostgreSQL, MySQL) yoki EXPLAIN PLAN (Oracle) buyrug'i ishlatiladi. Bu buyruq tizimning so'rovni qanday bajarishini ko'rsatadi: qaysi indekslardan foydalanilayotganini, qancha qatorlar ko'rib chiqilayotganini, qaysi birlashtirish usuli tanlanganini. So'rovni optimallashtirish jarayonida EXPLAIN natijasini tahlil qilish muhim ahamiyatga ega.

Xulosa

SQL tili - bu yarim asrlik tarixi bor, lekin bugungi kunda ham dolzarbligi so'nib bo'lmayotgan, balki yanada kengayib borayotgan texnologiya. Dastlab faqat ma'lumotlar bazasi ma'murlari uchun mo'ljallangan bu til bugun dasturchilar, ma'lumotlar tahlilchilari, biznes-tahlilchilar, ilmiy tadqiqotchilar va boshqa ko'plab mutaxassislar uchun asosiy vositaga aylandi.

Ushbu maqolada ko'rib chiqqan DDL, DML buyruqlari, JOIN operatsiyalari, agregat funksiyalar, kichik so'rovlar, tranzaksiyalar va indekslar - bularning barchasi SQL ning asosini tashkil etadi. Ularni chuqur o'zlashtirgan mutaxassis istalgan ma'lumotlar bazasi tizimida muvaffaqiyatli ishlash imkoniyatiga ega bo'ladi.

Bilimlarni mustahkamlash uchun amaliy mashqlar, real loyihalar va turli ma'lumotlar bazasi tizimlarida tajriba orttirishni tavsiya etaman. SQL ni o'rganish -bu nafaqat texnik ko'nikma, balki mantiqiy fikrlash va muammolarni tizimli yechish qobiliyatini ham rivojlantiradi. Bu ko'nikmalar esa zamonaviy texnologiyalar dunyosida muvaffaqiyatli karera qurishda beqiyos yordam beradi.

Foydalanilgan adabiyotlar:

- 1.E. F. Codd (1970). Katta umumiy ma'lumotlar banklari uchun ma'lumotlarning relyatsion modeli. Communications of the ACM jurnali.
- 2.C. J. Date (2004). Ma'lumotlar bazasi tizimlariga kirish.
- 3.A. Silberschatz, H. Korth, S. Sudarshan (2019). Ma'lumotlar bazasi tizimlari tushunchalari.
- 4.PostgreSQL hujjatlari (2023). PostgreSQL rasmiy texnik qo'llanmasi.
- 5.MySQL ma'lumotnoma qo'llanmasi (2023). MySQL rasmiy ma'lumotnoma va hujjatlari

