

**UZLILAR VA JADVALLAR YORDAMIDA KENGAYTIRILGAN  
MA'LUMOTLAR BAZASINI BOSHQARISH**

**Tojimatov Israiljon Nurmamatovich**

Farg'ona davlat universiteti

[israiltojimatov@gmail.com](mailto:israiltojimatov@gmail.com)

**Nishonova Ruhshona Otajon qizi**

Farg'ona davlat universiteti

[nishonovaruhshonaxon@gmail.com](mailto:nishonovaruhshonaxon@gmail.com)

**Adilbekova Raxila**

Botken Davlat Universitetining

**Qizil-Qiya Gumanitar-pedagogika Instituti**

**Annotatsiya:** Ushbu maqolada uzlilar (views) va jadvallar (tables) yordamida ma'lumotlar bazasini samarali boshqarish usullari ko'rib chiqiladi. Asosiy e'tibor PostgreSQL tizimida kengaytirilgan tuzilmalarning qanday tashkil etilishi va ularning yirik hajmdagi ma'lumotlarni boshqarishdagi o'rniga qaratiladi. Maqolada uzlilar orqali ko'p bosqichli so'rovlar, xavfsizlikni oshirish va ma'lumotlarni optimallashtirish kabi masalalar tahlil qilinadi.

**Kalit so'zlar:** PostgreSQL, uzlilar, jadvallar, ma'lumotlar bazasi, so'rovlar, optimallashtirish, xavfsizlik

**Abstract:** This article explores effective methods of managing databases using views and tables. The focus is on how extended data structures are organized in PostgreSQL and their role in managing large-scale datasets. The paper analyzes the use of views for multi-layered queries, enhancing security, and optimizing data handling.

**Keywords:** PostgreSQL, views, tables, database, queries, optimization, security

**Аннотация:** В данной статье рассматриваются эффективные методы управления базами данных с использованием представлений (views) и таблиц. Основное внимание уделяется тому, как в системе PostgreSQL организованы расширенные структуры данных и как они помогают управлять большими объемами информации. В статье анализируется использование представлений для многоуровневых запросов, повышения безопасности и оптимизации обработки данных.

**Ключевые слова:** PostgreSQL, представления, таблицы, база данных, запросы, оптимизация, безопасность

## Kirish

Axborot texnologiyalari jadal rivojlanayotgan hozirgi davrda ma'lumotlarni to'plash, saqlash, tahlil qilish va ulardan foydalanish deyarli barcha sohalarda -moliya, sog'liqni saqlash, ta'lim, sanoat, tijorat va boshqalarda - asosiy ehtiyojlardan biriga aylangan. Ma'lumotlar bazasi boshqaruv tizimlari ushbu jarayonlarni samarali tashkil qilish va avtomatlashtirishga xizmat qiladi. Ular foydalanuvchilarga strukturalangan ma'lumotlar ustida amallar bajarish, turli tahliliy va operatsion so'rovlarni tezkorlik bilan amalga oshirish, hamda ma'lumotlar xavfsizligini ta'minlash imkonini beradi. Bugungi kunda tashkilotlar ko'p miqdordagi murakkab va o'zgaruvchan ma'lumotlar bilan ishlaydi. Bunday sharoitda MBBTlarning tezkorlik, ishonchlilik, kengayuvchanlik, va funksional qulaylik kabi xususiyatlari ularning amaliy qiymatini belgilab beradi. Shu sababli, ma'lumotlar bazasi texnologiyalari har bir zamonaviy IT infratuzilmasining markaziy komponentlaridan biri hisoblanadi. PostgreSQL - bu kuchli, kengaytiriladigan va ochiq manbali MBBT bo'lib, ko'plab murakkab tizimlarda asosiy vosita sifatida qo'llaniladi. U dastlab Kaliforniya universitetida ishlab chiqilgan bo'lib, hozirda butun dunyo bo'ylab faol hamjamiyat tomonidan qo'llab-quvvatlanadi va rivojlantirilmoqda. PostgreSQL quyidagi ustunliklarga ega: Kengayuvchan arxitektura: foydalanuvchi funksiyalari, ma'lumot turlari, indeks turlari va operatorlar qo'shish imkoniyati. Yuqori darajadagi so'rovlar optimallashtirish: kompleks so'rovlar tez va samarali bajariladi. Xavfsizlik: foydalanuvchi huquqlari, audit, tarmoq xavfsizligi kuchli darajada ta'minlanadi. Aynan shu sababli PostgreSQL ko'plab katta hajmdagi axborot tizimlarida asosiy MBBT sifatida tanlanadi. Ushbu maqola maqsadi - PostgreSQL tizimi misolida uzlilar (views) va jadvallar yordamida ma'lumotlar bazasini samarali boshqarish usullarini chuqur tahlil qilishdan iborat. Ayniqsa, katta hajmdagi ma'lumotlar bilan ishlashda uzlilar orqali so'rovlarni soddalashtirish, foydalanuvchilarning huquqlarini chegaralash, va ma'lumotlar strukturasi optimallashtirish muhim ahamiyatga ega. Tadqiqot dolzarbligi shundaki, hozirgi vaqtda ko'plab tashkilotlar katta va murakkab ma'lumotlar to'plamlarini boshqarishga majbur bo'lib, samarali arxitektura tanlovi ularning ish faoliyatiga bevosita ta'sir qiladi. Uzlilar va jadvallarni to'g'ri tashkil qilish esa ma'lumotlar oqimining tezligi, sifatli saqlanishi va xavfsizligini ta'minlashga xizmat qiladi. Zamonaviy ma'lumotlar bazasi boshqaruv tizimlari ichida PostgreSQL alohida o'rin egallaydi. Bu tizim ochiq manbali, ob'ektga yo'naltirilgan va kuchli SQL imkoniyatlariga ega bo'lgan relatsion MBBT hisoblanadi. PostgreSQL birinchi bor 1986-yilda Kaliforniya universitetining Inqilobiy Axborot Tizimlari loyihasi doirasida ishlab chiqilgan bo'lib, keyinchalik keng jamoa tomonidan faol rivojlantirilib kelinmoqda.

PostgreSQL ANSI SQL:2011 va ACID (Atomicity, Consistency, Isolation, Durability) tamoyillariga to'liq mos keladi. U tranzaksiyalar ustida aniq nazorat, ma'lumotlar butligini kafolatlash va foydalanuvchi uchun keng sozlash imkoniyatlarini taklif etadi. PostgreSQL quyidagi asosiy texnik va arxitektura xususiyatlari bilan ajralib turadi: Modulli arxitektur: PostgreSQL foydalanuvchilarga o'z funksiyalarini, ma'lumot

turlarini, indekslash algoritmlarini va operatorlarini yaratish imkonini beradi. Pluginlar orqali kengaytirish: C yoki boshqa dasturlash tillarida yozilgan modullar yordamida tizimga yangi imkoniyatlar qo‘shish mumkin. Yadroga integratsiyalashgan PL/pgSQL tili: bu til yordamida foydalanuvchi tomonidan yaratilgan funksiyalarni yozish va triggerlarni avtomatlashtirish mumkin. PostgreSQL ma’lumotlar bazasida asosiy tushunchalar quyidagilardan iborat: ma’lumotlar satr va ustunlar ko‘rinishida saqlanadi. Ustun : har bir ustun o‘ziga xos ma’lumot turiga ega, masalan INTEGER, TEXT, DATE, BOOLEAN, yoki JSONB. Satrs: jadvaldagi har bir yozuv yoki yozilgan ma’lumot birligi. So‘rov : SQL yordamida ma’lumotlarni olish, tahrirlash yoki o‘chirish amallari. Indekslar : tezkor qidiruv va tahlilni ta’minlaydigan tuzilmalar. Bog‘lanishlar : chet kalitlar orqali jadvallar orasida bog‘lanish o‘rnatiladi. PostgreSQL an’anaviy int, text, float, date kabi ma’lumot turlaridan tashqari quyidagi zamonaviy turlarni ham qo‘llab-quvvatlaydi: JSON va JSONB: yarim-strukturalangan ma’lumotlar bilan ishlash uchun ARRAY: bir ustunda bir nechta qiymatlarni saqlash imkoniyati UUID, INET, CIDR, XML: maxsus identifikatorlar va tarmoqlar bilan ishlash uchun Composite turlar: foydalanuvchi tomonidan aniqlanadigan murakkab tuzilmalar Bu turlar ma’lumotlar modelini yuqori darajada moslashtirishga imkon beradi, bu esa turli xil tarmoq va veb-texnologiyalar bilan integratsiyani soddalashtiradi.

PostgreSQL -bu zamonaviy, kuchli va ishonchli ochiq manbali ma’lumotlar bazasi boshqaruv tizimi bo‘lib, relatsion model va ob’ektga yo‘naltirilgan funksiyalarni o‘zida mujassam etgan. Ushbu tizim keng jamoa tomonidan ishlab chiqilgan va muntazam yangilanib boriladi. Bugungi kunda PostgreSQL turli sohalarda, jumladan bank tizimlari, telekommunikatsiya, sog‘liqni saqlash, e-tijorat va ta’lim kabi yo‘nalishlarda keng qo‘llaniladi. PostgreSQL'ning yadro tamoyillari - barqarorlik, ma’lumotlar yaxlitligi, tranzaksiyalar ustidan aniq nazorat va kengaytiriluvchanlik hisoblanadi. Bu esa uni kichik loyihalardan tortib korporativ darajadagi yirik axborot tizimlarigacha bo‘lgan barcha sohalarda qo‘llash imkonini beradi. PostgreSQL tizimining eng muhim afzalliklaridan biri bu -ochiq manbali va modulli arxitekturaga ega bo‘lishidir. Bu degani, dasturchilar PostgreSQL manba kodiga bemalol murojaat qilishi, tizimni o‘z ehtiyojlariga moslab o‘zgartirishi va kengaytirishi mumkin. Bu ochiqlik uni ilmiy tadqiqotlarda, yangi texnologiyalar bilan integratsiyada va maxsus sohalarda ideal vositaga aylantiradi. Zamonaviy ma’lumotlar bazasida murakkab so‘rovlarni boshqarish, ko‘rinish soddalashtirish va xavfsizlikni oshirishda uzlilar muhim vosita hisoblanadi. PostgreSQL tizimida uzlilar yordamida foydalanuvchiga real jadvallarga bevosita murojaat qilmasdan, oldindan aniqlangan ma’lumotlar natijasini ko‘rsatish imkoniyati yaratiladi. Uzlilar foydalanuvchi va asosiy ma’lumotlar o‘rtasida oraliq qatlam vazifasini bajaradi. Ular ko‘p bosqichli, murakkab so‘rovlarni soddalashtirish, ma’lumotlarni mantiqiy guruhlash, filtrlar qo‘llash va xavfsizlikni ta’minlash uchun keng qo‘llaniladi. Uzli - bu virtual jadval bo‘lib, real ma’lumotlarni saqlamaydi, balki oldindan yozilgan SQL so‘rov natijasi asosida ishlaydi. Har gal SELECT orqali uzliga murojaat qilinganda, u ortida yotgan

so'rov qayta bajariladi. Yaratish sintaksisi: `CREATE VIEW view_nomi AS, SELECT ustun1, ustun2, FROM jadval_nomi WHERE shartlar;` View foydalanuvchilarga ma'lumotlar bazasining murakkab qismlarini soddalashtirilgan shaklda taqdim etadi. Asosiy jadvallar ichki ishlarini yashirgan holda, foydalanuvchiga faqat kerakli va cheklangan ma'lumotlarni ko'rsatadi. Aytaylik, sizda mijozlar, buyurtmalar va to'lovlar haqidagi bir nechta jadvallar mavjud. Har safar bu jadvallar asosida foydalanuvchi faoliyati statistikasi olinishi kerak bo'lsa, har gal ko'p bosqichli JOIN va GROUP BY ishlatish murakkablik tug'diradi. Uzlilar xavfsizlik darajasini oshirishda ham muhim rol o'ynaydi. Real jadvallarda mavjud nozik ma'lumotlarga to'g'ridan-to'g'ri kirish o'rniga, foydalanuvchiga faqat kerakli ustunlar yoki satrlar ko'rinadigan uzli orqali kirish huquqi beriladi. Ma'lumotlar bazasi tuzilmasi samaradorligini oshirish, ma'lumotlar yaxlitligini ta'minlash hamda tizim ish faoliyatini optimallashtirishda jadvallar bilan to'g'ri ishlash muhim ahamiyatga ega. PostgreSQL kabi ilg'or ma'lumotlar bazasi boshqaruv tizimlarida jadval tuzilmalari ustida chuqur tahliliy yondashuv talab etiladi. Quyida bu boradagi asosiy usullar yoritiladi. Normalizatsiya - bu ma'lumotlar bazasi jadvallarini ortiqcha ma'lumotlardan tozalash, yaxlitlikni ta'minlash va ma'lumotlarni mantiqan mantiqli bo'limlarga ajratish jarayonidir. Bu jarayon orqali: ma'lumotlar takrorlanishini kamaytirish; yangilash va o'chirishdagi ziddiyatlarni oldini olish; ma'lumotlar yaxlitligini saqlash mumkin. Normalizatsiya odatda birinchi (1NF), ikkinchi (2NF), uchinchi (3NF) va undan yuqori shakllarda amalga oshiriladi. Har bir bosqichda jadval ma'lum qoidalarga moslashtiriladi. Denormalizatsiya esa normalizatsiyaning aksi bo'lib, ma'lumotlar bazasining o'qish tezligini oshirish maqsadida ma'lumotlarni ayrim hollarda takrorlashga yo'l qo'yiladi. Bu ayniqsa, analitik so'rovlar, hisobotlar yoki katta hajmli JOIN operatsiyalardan qochish zarur bo'lgan hollarda foydali bo'ladi. Misol: Normalizatsiya - mijoz va manzil alohida jadvallarda, Denormalizatsiya - mijoz jadvalida manzil ustuni mavjud

Relation ma'lumotlar bazasida jadvallar orasidagi bog'lanishlar ma'lumotlar strukturasi va bog'liq so'rovlar uchun asos bo'ladi. Bu bog'lanishda har bir yozuv faqat bitta boshqa yozuv bilan bog'lanadi. Masalan, foydalanuvchi jadvali va foydalanuvchi\_profil jadvali. Eng keng tarqalgan bog'lanish turi. Masalan, bir mijoz bir nechta buyurtmaga ega bo'lishi mumkin. het kalit - bu jadvaldagi ustun orqali boshqa jadvaldagi asosiy kalitga bog'lanishni bildiradi. Bu vosita orqali ma'lumotlar yaxlitligi ta'minlanadi, ya'ni mavjud bo'lmagan ob'ektga bog'lanishning oldi olinadi. isol: `ALTER TABLE buyurtma, ADD CONSTRAINT fk_foydalanuvchi, FOREIGN KEY (foydalanuvchi_id), REFERENCES foydalanuvchi(id), ON DELETE CASCADE;` Bu yerda ON DELETE CASCADE sharti orqali asosiy yozuv o'chirilganda unga bog'langan yozuvlar ham o'chiriladi. Indekslar esa jadvallarda qidiruv tezligini oshiradi. PostgreSQL avtomatik tarzda primary key va unique ustunlar uchun indeks yaratadi. Ammo tez-tez ishlatiladigan so'rov ustunlariga qo'lda indeks qo'shish katta hajmdagi ma'lumotlar bilan ishlashda samaradorlikni oshiradi. Katta hajmdagi ma'lumotlar ustida samarali ishlash

uchun quyidagi yondashuvlar muhim: Indeksflash - so'rovni tezlashtiradi, lekin ortiqcha indeks xotira sarfini oshiradi. Partitioning - katta jadvallarni vaqt, kategoriya yoki boshqa mezonlar asosida kichik bo'limlarga ajratish. Ilmiy va amaliy ishlar samaradorligini ta'minlash uchun nazariy yondashuvlarni real loyihalarda qo'llash muhim ahamiyatga ega. Quyida PostgreSQL ma'lumotlar bazasi tizimida uzlilar va jadvallar asosida amalga oshirilgan loyiha misolida samaradorlik, xavfsizlik va monitoring vositalari tahlil qilinadi. Tasavvur qilaylik, siz elektron savdo platformasi uchun ma'lumotlar bazasi ishlab chiqmoqdasiz. Platformada quyidagi asosiy jadvallar mavjud: foydalanuvchilar - mijozlar ma'lumotlari, buyurtmalar - foydalanuvchi buyurtmalari, mahsulotlar - mavjud mahsulotlar, tolovlar - to'lovlar tarixi. Murakkab hisob-kitobli so'rovlar va hisobotlar uchun quyidagi uzlilar yaratildi: foydalanuvchi\_statistikasi - foydalanuvchining umumiy xarajatlari va buyurtma soni, kunlik\_savdo - har kuni bo'yicha jami savdo summasi. Uzlilar yordamida samaradorlik sezilarli darajada oshdi. Hisobotlar uchun har gal murakkab JOIN va GROUP BY ishlatish o'rniga, bir nechta uzlilar orqali natijalar tezkor ko'rinishda olinmoqda. Xavfsizlik nuqtai nazaridan esa, uzlilar real jadvallarda saqlanuvchi nozik (maxfiy) ma'lumotlarni cheklashda foydali bo'ldi. PostgreSQL tizimining ishlashini doimiy nazorat qilish va optimallashtirish uchun bir qancha vositalar mavjud. Ulardan ba'zilari quyidagilar: Bazadagi faol so'rovlar va ulanishlarni ko'rish: `SELECT * FROM pg_stat_activity;` So'rov rejasini va undan foydalanilayotgan indekslarni ko'rsatadi: `EXPLAIN ANALYZE SELECT * FROM buyurtmalar WHERE mijoz_id = 123;` Tadqiqot davomida PostgreSQL ma'lumotlar bazasi boshqaruv tizimining uzlilar va jadvallar bilan ishlashdagi keng imkoniyatlari va ularning amaliy ahamiyati chuqur tahlil qilindi. Asosiy natijalar quyidagilardan iborat: PostgreSQL ochiq manbali, yuqori darajada kengaytiriladigan va professional ma'lumotlar bazasi sifatida murakkab tizimlar uchun eng maqbul yechimlardan biri ekanligi isbotlandi. Uzlilar yordamida murakkab so'rovlar soddalashtirildi, foydalanuvchi xavfsizligi oshirildi va tizim ishlashi yengillashtirildi. Jadvallar orasidagi bog'lanishlar, chet kalitlar, indekslash va normalizatsiya orqali ma'lumotlar yaxlitligi va ishlov berish samaradorligi ta'minlandi. Amaliy loyiha misolida uzlilar va jadvallar o'rtasidagi strategik yondashuv tizim ish faoliyatini sezilarli darajada yaxshilaganligi kuzatildi. Ushbu natijalar PostgreSQL'dan foydalanayotgan yoki uni joriy etishni rejalashtirayotgan tashkilotlar uchun amaliy qo'llanma bo'lib xizmat qiladi.

### **Xulosa**

Uzlilar va jadvallarni strategik darajada to'g'ri qo'llash PostgreSQL asosidagi har qanday ma'lumotlar bazasining mustahkam, xavfsiz va samarali ishlashini ta'minlaydi. Uzlilar: murakkab analitik so'rovlarni soddalashtirish, turli foydalanuvchilar uchun cheklangan ko'rinishlar yaratish, hisobotlarni tezlashtirishda muhim vositadir. Ushbu elementlarning o'zaro muvofiqlashtirilgan ishlatilishi har qanday zamonaviy axborot tizimi uchun poydevor hisoblanadi. Tadqiqot davomida aniqlangan muhim jihatlar asosida kelajakda quyidagi yo'nalishlar bo'yicha chuqur ilmiy izlanishlar olib

borish maqsadga muvofiqdir: PostgreSQL replikasiya va klasterlash texnologiyalarini chuqur tahlil qilish va yirik tizimlarda qo'llash bo'yicha metodik yondashuvlar ishlab chiqish; PostgreSQL bilan integratsiyalashgan mikroxizmatlar arxitekturasi bo'yicha amaliy yechimlarni ishlab chiqish.

### Foydalanilgan adabiyotlar

1. PostgreSQL Global Development Group. *PostgreSQL Documentation*. <https://www.postgresql.org/docs/>
2. Klaas, P., & Baader, D. *PostgreSQL: Up and Running*. O'Reilly Media, 2019.
3. Chaudhuri, S., & Dayal, U. "An overview of data warehousing and OLAP technology." *ACM SIGMOD Record*, vol. 26, no. 1, 1997, pp. 65-74.
4. PostgreSQL Development Group. *PostgreSQL Query Optimization and Performance Tuning*. Packt Publishing, 2021.
5. Tojimamatov, I., & Sahobiddinov, A. (2025). BERILGANLAR BAZISI USTIDA HISOBOTLARNI SHAKILLANTIRISH. *BRIDGING THE GAP: EDUCATION AND SCIENCE FOR A SUSTAINABLE FUTURE*, 1(1), 83-90.
6. Tojimamatov, I., & Qirg'izboyev, D. (2025). SQL SO 'ROVLAR TILI STRUKTURASI. SQL-STRUKTURALANGAN SOROVLAR TILI. *BRIDGING THE GAP: EDUCATION AND SCIENCE FOR A SUSTAINABLE FUTURE*, 1(1), 64-71.