Engineer

PostgreSQL

база даних, на якій побудовано більшість цих робіт

Що це таке

PostgreSQL — це реляційна система керування базами даних, безкоштовна і з відкритим кодом, яка зберігає дані в таблицях, відповідає на запити мовою SQL і виконує кожну транзакцію за принципом «усе або нічого». Більшість тих, хто з нею працює, називає її Postgres, і проєкт вважає обидві назви своїми. На запитання «що таке PostgreSQL?» досвідчений інженер відповість просто: це універсальна база даних, з якої починається нова система, коли ні в кого немає конкретної причини обрати іншу.

Вона старша за більшість програмного забезпечення, яке нині на ній працює. Проєкт POSTGRES почався в Каліфорнійському університеті в Берклі під керівництвом професора Майкла Стоунбрейкера, а його реалізацію розпочали 1986 року. 1994 року Ендрю Ю та Джоллі Чен додали інтерпретатор SQL і випустили результат під назвою Postgres95. Через два роки систему перейменували на PostgreSQL, а нумерація версій почалася з 6.0, продовжуючи послідовність, розпочату проєктом у Берклі. Сам проєкт описує систему як об'єктно-реляційну: крім таблиць і з'єднань, користувач може додавати нові типи даних, функції, оператори, агрегатні функції, методи індексування та процедурні мови, і сервер працює з ними як із власними.

Архітектурно це клієнт і сервер. Один серверний процес володіє файлами даних, а кожне клієнтське з'єднання отримує власний процес, який сервер породжує саме для нього. Застосунки надсилають цим з'єднанням SQL через драйвер для тієї мови, якою вони написані. Паралельна робота тримається на багатоверсійному керуванні конкурентним доступом, MVCC: оновлення записує нову версію рядка, а не перезаписує стару, тож кожна інструкція читає узгоджений знімок, читання ніколи не чекає на запис, а запис ніколи не чекає на читання.

Яке місце вона посідає серед альтернатив? Поруч із MySQL і MariaDB як іншими реляційними серверами з відкритим кодом, поруч з Oracle і Microsoft SQL Server як комерційними, поруч із SQLite як вбудованою базою даних, що живе всередині однієї програми, а на противагу їм — документні сховища на кшталт MongoDB, які зберігають записи без фіксованої схеми.

Застосуйте поняття до рішення

Застосуйте «Що таке PostgreSQL?» до конкретного рішення. Ми можемо пов'язати поняття із системою, яку ви створюєте або купуєте. Пошук запитів і відсутніх індексів у системі, що рік за роком важчає, та виправлення тих, що справді мають значення.

Переглянути наші послуги Зв'язатися з нами →

Перевірте припущення за терміном

Використайте пояснення та пов'язані роботи, щоб сформулювати запитання для проєкту. Ми можемо перевірити припущення разом до вибору інструмента. 30 днів даних про електроенергію за 180 мс — 180 ms на агрегацію місяця показань лічильників

30 днів даних про електроенергію за 180 мс Зв'язатися з нами →

Чому це важливо

Вибір бази даних — це рішення, з яким команда живе роками, бо дані переживають код, написаний для роботи з ними. PostgreSQL — вибір за замовчуванням, про який рідко шкодують, і на це є три причини, які покупець може перевірити сам.

Перша — вартість і прив’язка до постачальника. PostgreSQL поширюється за ліцензією PostgreSQL, короткою дозвільною ліцензією з тієї ж родини, що BSD і MIT, тож немає плати за ядро, аудиту ліцензій чи постачальника, який може змінити умови. Amazon, Microsoft і Google продають її як керований сервіс, а та сама база даних працює на ноутбуці, тож переїзд між хостингами — це перенесення даних, а не переписування застосунку.

Друга — широта. Один сервер охоплює реляційні таблиці, JSON-документи в стовпцях `jsonb`, повнотекстовий пошук, геопросторові запити через розширення PostGIS і заплановані завдання через `pg_cron`. Кожне з цього — окремий продукт, який команді не доводиться розгортати, захищати, резервувати й під який не треба наймати людей. Для більшості продуктів вибір між PostgreSQL і MySQL вирішується саме тут: зі звичайним читанням і записом обидві справляються добре, а різниця проявляється, коли робота потребує багатших типів, суворіших обмежень або розширення.

Третя — довговічність. Проєкт випускає одну основну версію на рік і підтримує кожну п'ять років, тож оновлення можна планувати, а не проводити в аварійному порядку. PostgreSQL 18, що вийшов у вересні 2025 року, додав асинхронне введення-виведення та вбудовану функцію для впорядкованих за часом UUID, а SQL, написаний для версій, старших на десятиліття, зазвичай працює на ньому без змін.

Компроміси реальні. Окремий процес на кожне з'єднання робить з'єднання дорогими, тож навантаженому застосунку потрібен пулер. PostgreSQL легше росте на одній потужнішій машині, ніж на кількох: щоб розподілити запис між багатьма серверами, потрібні партиціонування, реплікація або розширення, і кожен із цих шляхів додає експлуатаційного навантаження. А налаштування за замовчуванням розраховані на невелику машину, тож серйозну інсталяцію доводиться конфігурувати свідомо.

Як це виглядає на практиці

Чимало проєктів на цьому сайті побудовано на PostgreSQL, і три з них показують три різні способи на неї спиратися.

На одній платформі база даних стала всім рівнем даних. Кожне читання й кожен запис проходять через одну з 1 275 функцій PostgreSQL у 34 схемах, і Go API не має прямого доступу до жодної таблиці. Роль, під якою API входить у базу, `mariner`, має EXECUTE на функціях і USAGE на схемах — і нічого більше, це приблизно 4 000 явних грантів. Функції виконуються як SECURITY DEFINER від імені окремої ролі-власника без права входу та із зафіксованим `search_path`. Межу доступу забезпечує сам Postgres; ціна в тому, що нове поле означає міграцію та зміну функції.

У застосунку з орендарями, локаціями та щоденними показниками електроспоживання робота полягала в налаштуванні, а не в проєктуванні, а завданням було підвищити продуктивність у 10 разів без капітальної переробки наявної архітектури. Індекси GiST для геопросторових даних, композитні індекси за орендарем і діапазоном дат, партиціонування таблиці електроенергії за днями й місяцями і запити, переписані після аналізу їхніх планів через `EXPLAIN ANALYZE`, скоротили час відповіді на запити у 10 разів і знизили використання ресурсів на 40%.

А на платформі з повільною фоновою роботою черга задач живе в тій самій базі даних. River тримає свою чергу в PostgreSQL, тож задача і рядки, яких вона торкається, комітяться або відкочуються разом. За чергою стоять 15 модулів-воркерів, 8 запланованих задач і 20 задач `pg_cron`, які підчищають партиції, ротують солі та оновлюють агрегати. Сторінка проєкту прямо називає межу: черга в Postgres на певному обсязі перестає бути правильною відповіддю, а на одну рухому частину менше виявилося ціннішим за запас, яким однаково ніхто не скористався б.

Де виникають проблеми

PostgreSQL рідко втрачає дані. Зате вона вміє дивувати, і ці несподіванки мають кілька знайомих форм.

Таблиця росте, а кількість рядків стоїть на місці. `UPDATE` або `DELETE` залишає стару версію рядка, доки її не прибере `VACUUM`, бо інша транзакція ще може її читати. Транзакція, залишена відкритою на години, стримує це прибирання для всієї бази даних, а налаштування autovacuum за замовчуванням розраховані на невеликі таблиці. Закривайте неактивні транзакції, налаштовуйте autovacuum окремо для великих таблиць і партиціонуйте дані, що застарівають, щоб старі місяці видалялися цілими партиціями, а не рядок за рядком. Vacuum має й другу роботу: ідентифікатори транзакцій 32-бітні, кожна таблиця має проходити через vacuum принаймні раз на два мільярди транзакцій, а сервер, що надто відстав, перестає приймати нові записи, щоб захистити дані.

Пам’ять закінчується на першому ж піку трафіку. Кожне з’єднання — це серверний процес із власною пам’яттю, а ліміт за замовчуванням — 100 з’єднань. Двадцять екземплярів застосунку з пулом на п’ятдесят з’єднань кожен просять тисячу процесів. Поставте перед сервером пулер на кшталт PgBouncer і задайте розмір пулу кожного застосунку за обсягом роботи — зазвичай вистачає кількох з’єднань на екземпляр.

Запит, який виконувався за мілісекунди, починає тривати секунди. Із ростом даних планувальник почав обирати інший план, часто тому, що статистика застаріла або індекси будували під торішні запити. Прочитайте план, перш ніж щось змінювати, оновіть статистику командою `ANALYZE` і проіндексуйте той предикат, за яким запит насправді фільтрує:

```
EXPLAIN (ANALYZE, BUFFERS)
SELECT id, total FROM orders
WHERE customer_id = 42 AND placed_at >= '2026-01-01';
```

Вставки сповільнюються, коли таблиця стає великою. Випадкові UUID як первинні ключі потрапляють у довільні місця B-tree-індексу, тож кожна вставка зачіпає іншу сторінку, і індекс фрагментується. Ідентифікатори, впорядковані за часом, цього уникають: одна платформа зробила UUID v7 ключем за замовчуванням через `uuidv7()` з PostgreSQL 18, і нові рядки вбудовуються в індекс, а не розсипаються по ньому.

Що почитати далі

- What Is PostgreSQL? — власне визначення проєкту з можливостями та точками розширення, які він називає.
- A Brief History of PostgreSQL — Берклі, Postgres95 і перейменування, розказані самим проєктом.
- Concurrency Control: Introduction — як знімки дають читанню й запису відбуватися, не блокуючи одне одного.
- Routine Vacuuming — мертві версії рядків, autovacuum і переповнення лічильника транзакцій у подробицях.
- Using EXPLAIN — як читати план запиту, з розібраними прикладами.
- Versioning Policy — щорічна основна версія і п’ять років підтримки кожної.

Словник

Обирайте за фактичними умовами

Надайте репрезентативний набір даних, запитання до нього та поточне вузьке місце. Узгодимо вимірювання якості, пропускної здатності та відновлення.

Дивитися, скільки це коштує Зв'язатися з нами →

Зробіть термінологію практичною

Запитайте, як «Що таке PostgreSQL?» стосується вашої організації.
Конкретного прикладу з поточної системи достатньо для початку розмови.

Переглянути наші послуги

welcome@engineer.company +45 50 10 61 84 Копенгаген, Данія

Візьміть цю сторінку із собою

Візитівка

Дбайливо зроблено з любов'ю — D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

Авторське право © 2025 – дотепер · Engineer ApS · Усі права захищено.

Надруковано з https://engineer.company/uk/glossary/postgresql/