Engineer

PostgreSQL

28 досягнень

Поєднайте частини системи

Роботи за темою «PostgreSQL» показують одну частину дієвої системи. Ми можемо допомогти пов'язати її з рештою вашої діяльності. 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 →

Знайдіть схоже завдання

Відкрийте приклад, схожий на вашу ситуацію. Огляд пояснює рішення, що привели до результату, а не лише називає технологію. 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 →

30 днів даних про електроенергію за 180 мс

Розробила варіанти погодинної агрегації споживання електроенергії на Python, SQL і Bash + Jq; під час вимірювання Python обробив один 30-денний набір даних JSONL за 180 мс.

Прискорила пайплайн географічних даних у 50 разів

Зафіксувала прискорення географічного пайплайна у 50 разів у роботі засновниці для Utiligize завдяки змінам SQL і моделей даних у PostgreSQL, MS SQL та Google Cloud BigQuery.

Пришвидшила GIS-карти в 10 разів на PostgreSQL

Підвищила продуктивність застосунку географічних карт у 10 разів завдяки стратегічному переходу бази даних з MSSQL на PostgreSQL, включно з фінальним перенесенням даних (cutover), оптимізувавши обробку та безпеку даних.

Усунула 1 000 проблем у географічних даних

Усунула 1 000 проблем у географічних даних та часових рядах, використовуючи GDAL, ArcGIS, PostGIS, Mapbox, QGIS, SQL (PL/pgSQL, Transact-SQL), Bash, забезпечивши високоякісну обробку великих даних.

Спроєктувала й підтримувала 6 пайплайнів ETL/ELT

Спроєктувала, реалізувала та адмініструвала 6 пайплайнів ETL/ELT, використовуючи Google BigQuery, MSSQL, PostgreSQL, shell-скрипти, PL/pgSQL та Transact-SQL, інтегрувавши дані для ефективної обробки через Python API.

Випустила 500 звітів про електроенергію та GIS

Випустила 500 звітів з аналізу даних електроенергетики та GIS, застосовуючи поглиблені дослідження та усунення несправностей для забезпечення точної аналітики географічних і часових великих даних.

Вела 100 сховищ PostgreSQL, MS SQL і BigQuery

Спроєктувала та створила 100 баз даних для сховищ даних на PostgreSQL, MS SQL та Google BigQuery, переважно з GIS-даними та часовими рядами, і керувала ними, оптимізувавши продуктивність і масштабованість.

Підтримувала 10 серверів PostgreSQL і MS SQL

Спроєктувала, розгорнула та підтримувала 10 серверів PostgreSQL і MS SQL на Ubuntu Linux VPS, забезпечивши оптимальну продуктивність і надійність серверів.

Прискорила PostgreSQL у 10 разів

Прискорила PostgreSQL у 10 разів завдяки стратегічній індексації, партиціонуванню та оптимізації запитів, підвищивши ефективність бази даних для користувацьких, тенантних, геопросторових та часових електричних даних.

Автоматизувала 100 критичних резервних копій

Автоматизувала 100 критично важливих резервних копій даних за допомогою Barman, Google Cloud, Bash та Python, забезпечивши цілісність даних в усіх базах.

Керувала full-stack-розробкою GIS-карт

Керувала full-stack-розробкою GIS-карт, наглядаючи за роботою з PostgreSQL, Mapbox, ReactJS та NodeJS, щоб створити інтегроване рішення.

Очолила понад 30 GIS-проєктів

Очолила розробку, розгортання та підтримку понад 30 GIS-проєктів, продемонструвавши фахові знання технологій PostgreSQL, Bash, Python, JavaScript, GDAL, ArcGIS, PostGIS та Mapbox.

Автоматизувала обробку даних на Python і PL/pgSQL

Автоматизувала завдання обробки даних за допомогою shell-скриптів, PL/pgSQL, Python та Transact-SQL, підвищивши продуктивність і ефективність.

Розробила 600 пайплайнів ETL/ELT на PL/pgSQL

Розробила 600 пайплайнів ETL/ELT на основі PL/pgSQL для оптимізації складних робочих процесів обробки даних у кількох середовищах розробки та продакшну PostgreSQL.

Морська платформа з чіткими межами систем

Спроєктувала багаторівневу морську платформу, розділивши фронтенд Next.js PWA, API на Go (Huma/Fiber) та рівень функцій PostgreSQL, зберігши всю бізнес-логіку в базі даних.

Спроєктувала наскрізний JSON від Postgres до API

Спроєктувала архітектуру наскрізної передачі JSON, у якій функції PostgreSQL повертають повний JSON, що дослівно пересилається через Go API, усунувши проміжну десеріалізацію (unmarshalling) та унезалежнивши фронтенд від змін схеми.

Спроєктувала функційний рівень даних PostgreSQL

Спроєктувала рівень даних PostgreSQL, побудований насамперед на функціях — 1 275 збережених функцій у 34 схемах — щоб кожне читання й кожен запис проходили через функцію, на яку база даних може видати права, а не через таблицю.

Впорядковані за часом ключі для моделі даних

Запровадила впорядковані за часом ідентифікатори UUID v7 (PostgreSQL 18) як ключі сутностей, щоб зменшити фрагментацію індексів B-tree та пришвидшити запити.

Впровадила deep-merge на основі каталогу

Впровадила deep-merge на основі каталогу для збережених JSON-налаштувань, запобігши збоям через відсутні ключі в міру розвитку схеми.

Змоделювала 348 таблиць у 34 схемах PostgreSQL

Змоделювала морську предметну галузь — фахівців, компанії, судна, вакансії, відгуки та решту — у 348 нормалізованих таблицях у межах 34 схем PostgreSQL, з довідниками SMALLINT та ключами UUID v7.

Налаштувала резервні копії та відновлення як код

Налаштувала зберігання резервних копій бази даних у вигляді інфраструктури як коду (infrastructure-as-code), провела аудит готовності до відновлення й задокументувала процедуру відновлення — чітко назвавши наявні прогалини, а не залишивши їх на час інциденту.

Побудувала тести для Go, SQL і браузера

Побудувала багаторівневий набір автоматизованих тестів — 981 тест Go, 543 фронтендні та браузерні специфікації, 494 поведінкові тести SQL — з мутаційним тестуванням, property-based тестами та обов'язковою перевіркою доступності.

Доступ через Stripe та покупки в застосунку

Побудувала рівень платежів і прав доступу — Stripe поряд із внутрішньозастосунковими покупками Apple і Google — обмеживши каталог, пошук та експорт моделлю доступу з 11 таблиць, яку перевіряє сервер.

Перенесла повільні операції в чергу завдань River

Перенесла повільну роботу зі шляху запиту в чергу завдань River — 15 модулів воркерів, 8 запланованих задач та 20 завдань pg_cron — щоб запит повертався, поки робота за ним триває.

Тримала схему узгодженою впродовж 1 022 міграцій

Тримала схему узгодженою впродовж 1 022 міграцій за допомогою перевірки CI, яка збирає базу даних обома шляхами — чиста інсталяція та інсталяція плюс усі міграції — і падає, коли вони розходяться.

Побудувала власну аналітику в PostgreSQL

Побудувала власну продуктову аналітику в PostgreSQL — 47 функцій над партиціонованими таблицями подій, які самі себе очищають, — псевдонімізовану за допомогою солі, що регулярно змінюється, і ввімкнену лише за згодою відвідувача.

Побудувала систему лояльності та репутації

Побудувала систему лояльності та репутації — 67 функцій на основі реєстру з 31 таблиці, з лігами, бейджами та крамницею обміну, — і блокувала рядок балансу, щоб закрити вікно для подвійного витрачання.

Побудувала розділи найму й кар'єри для моряків

Побудувала робочі простори найму й кар'єри моряків; зафіксований стан реалізації містить 151 збережену функцію у 48 таблицях для вакансій, заявок, сертифікатів, просування за посадами й морського стажу.

Спрямуйте інженерні зусилля

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 →

Розгляньте тему у своєму контексті

Розкажіть, як «PostgreSQL» стосується вашої системи. Ми допоможемо обрати відповідну послугу та визначити обсяг роботи.

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/categories/postgresql/