Engineer

BigQuery

сховище Google для запитів до величезних таблиць

Що це таке

BigQuery — це керований SQL-сервіс для аналітики в Google Cloud: серверів для нього ніхто не розгортає, таблиці зберігаються стовпець за стовпцем, а кожен запит оплачується або за обсяг прочитаних байтів, або з обчислювальної потужності, зарезервованої наперед. Його також називають Google BigQuery, а діалект SQL, яким він користується, — GoogleSQL: він дотримується стандарту ANSI і додає власні функції.

В основі архітектури — розділення зберігання й обчислень. Таблиці лежать у сховищі Google, а запит позичає обчислювальні ресурси лише на час виконання. Оскільки кожен стовпець зберігається окремо, запит, що називає три стовпці з п'ятдесяти, читає саме ці три, а решти не торкається. Тому й налаштування продуктивності тут інше, ніж на сервері PostgreSQL. Замість проєктування B-tree-індексів і розрахунку пам'яті робота зводиться до форми таблиці: її можна поділити на партиції за стовпцем дати чи часової мітки, за часом завантаження або за діапазоном цілих чисел, а також кластеризувати за кількома стовпцями, до чотирьох, щоб пов'язані рядки лежали в тих самих блоках.

BigQuery належить до класу сховищ даних, і його зазвичай порівнюють зі Snowflake та Amazon Redshift, а також із варіантом залишити аналітику на транзакційній базі даних, яку компанія вже має. Він створений для запитів, що проходять мільйони чи мільярди рядків і повертають кілька підсумкових чисел. Пошук і оновлення окремих рядків, які застосунок виконує на кожен запит користувача, належать базі даних на кшталт PostgreSQL чи SQL Server. Поширена схема — тримати обидві системи й переносити дані з однієї в іншу конвеєром.

Застосуйте поняття до рішення

Застосуйте «Що таке BigQuery?» до конкретного рішення. Ми можемо пов'язати поняття із системою, яку ви створюєте або купуєте. Дані продажів, операцій і фінансів зведені разом, щоб два відділи не приносили на одну нараду дві різні відповіді.

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. Прискорила пайплайн географічних даних у 50 разів — 50x швидші пайплайни географічних даних

Прискорила пайплайн географічних даних у 50 разів Зв'язатися з нами →

Чому це важливо

Рішення, яке BigQuery ставить перед командою, стосується того, як платити за аналітику. Рахунок залежить від запитів, які пишуть люди, а не від розміру машини, яку колись хтось обрав, і це має два боки.

За тарифом on-demand кожен запит оплачується за обсяг оброблених байтів. Це зручно для нерегулярної роботи: команда, яка кілька разів на тиждень запускає важкі звіти, платить за ці звіти й більше ні за що, а сервери не простоюють ночами. Водночас недбалий запит отримує свою ціну. `SELECT *` за рік показників датчиків коштує однаково, хоч аналітикові потрібен був один рядок, хоч усі. Ємнісний тариф замінює це слотами — одиницями обчислювальної потужності, які купують через резервування. Тоді рахунок відомий наперед, а інтенсивне й рівномірне використання легше планувати. Проєкт може переходити між двома моделями, тож перший вибір можна переглянути, коли стане видно реальне навантаження.

Зберігання має власні правила. Таблиця чи партиція, яку не змінювали 90 днів поспіль, автоматично переходить на тариф довгострокового зберігання, нижчий на 50%. Кожен набір даних тарифікується або за логічним, нестиснутим обсягом даних, або за фізичним, стиснутим, і для даних, що добре стискаються, друга модель може виявитися дешевшою.

Перед тим як погоджуватися, варто зважити три речі. Розташування набору даних — окремий регіон або мультирегіон US чи EU — задається під час створення і потім не змінюється, тож це рішення про відповідність вимогам, яке ухвалюють у перший же день. Діалект, інструмент командного рядка `bq` і клієнтські бібліотеки належать Google, тому запити переходять до іншого сховища даних із правками, а не як є. Самі дані водночас вивантажити легко:

таблиці експортуються в CSV, Avro і Parquet. А питання найму м’якше, ніж здається: фахівці з аналізу даних, які знають SQL, стають продуктивними за кілька днів, а рідкісніша навичка — читати запит у байтах, а не лише в рядках.

Тож коли BigQuery — правильний вибір? Коли дані надходять великими обсягами й здебільшого лише доповнюються, як-от події, журнали чи показники лічильників; коли питання до них змінюються з тижня на тиждень; і коли команда не хоче адмініструвати сервери баз даних для аналітики. Для даних, які спокійно вміщує один сервер PostgreSQL, сховище даних додає ще одну систему, ще один рахунок і конвеєр між ними. Аргументи на його користь зростають разом з обсягом і з кількістю людей, які ставлять запитання.

Як це виглядає на практиці

Понад 100 баз даних в одному середовищі працювали на PostgreSQL, Microsoft SQL Server і Google BigQuery, і BigQuery стояв там поряд із PostgreSQL, а не замість нього. Більшість даних були географічними або часовими рядами: координати, показники датчиків, журнали й метрики в реальному часі. З боку BigQuery великі таблиці часових рядів було поділено на партиції за часом і кластеризовано, а стиснення й стовпцеве зберігання застосовано, щоб тримати витрати на зберігання низькими, а сканування швидким. Разом з індексуванням, матеріалізованими представленнями й кешуванням у всьому середовищі це на 40–60% скоротило час, за який повертаються GIS-запити й запити до часових рядів.

У геопросторовому конвеєрі даних, який став повільним, спершу проаналізували SQL-код і моделі даних у PostgreSQL, MS SQL і Google Cloud BigQuery, що виявило слабке індексування, неефективні запити й відсутні обмеження цілісності. Потім для кожного рушія застосували доступний йому інструмент: B-tree- та GiST-індекси в PostgreSQL, фільтровані індекси в MS SQL і кластеризацію в BigQuery, а також нормалізовані моделі, партиціювання й переписані запити. В описі досягнення засновниці зазначено, що конвеєр загалом працював у 50 разів швидше; набір даних для бенчмарку, обладнання й процедуру порівняння там не вказано.

А на даних розумних лічильників із кількох регіонів понад 500 аналітичних звітів було написано чистим SQL у PostgreSQL і BigQuery: з CTE, віконними функціями, підзапитами й геопросторовими з’єднаннями за ідентифікаторами лічильників, часовими мітками, координатами й

метаданими про довкілля. До кожного звіту додавали прокоментований SQL, щоб колега міг перезапустити й перевірити його, а також нотатки про відомі проблеми якості даних, як-от відсутні GPS-координати чи пошкоджені показники лічильників. Звіти стали стандартним довідником для багатьох відділів і сприяли підвищенню ефективності операційного планування на 10–15%.

Де виникають проблеми

Запит із `LIMIT 10` оплачується за всю таблицю. На некластеризованій таблиці `LIMIT` обрізає результат, а не прочитані байти, а `SELECT *` читає кожен стовпець, який є. Швидкий погляд на дані перетворюється на повне сканування. Називайте лише ті стовпці, які потрібні для питання, а зразки рядків дивіться через попередній перегляд таблиці в консолі або `bq head` — це безкоштовно.

Партиційовану таблицю все одно сканують від початку до кінця. Партиції пропускаються лише тоді, коли фільтр за стовпцем партиціювання — константа, яку рушій може обчислити ще до читання. Порівняння стовпця з результатом підзапиту, загортання його у функцію чи арифметика над ним змушують запит пройти всі партиції. Допомагають два запобіжники. Таблиця може вимагати фільтр за партицією, і тоді запит без придатного фільтра зупиняється з помилкою, замість того щоб виконатися. А пробний запуск (dry run) показує, скільки байтів обробив би запит, не виконуючи його і не беручи за нього плату:

```
bq query --use_legacy_sql=false --dry_run \
  'SELECT meter_id, kwh
   FROM energy.readings
   WHERE reading_date = DATE "2026-10-01"'
```

Кластеризацію додали, а кількість прочитаних байтів майже не змінилася. Кластеризація впорядковує блоки сховища за стовпцями, до чотирьох, у заданому порядку, і цей порядок важить: таблиця, кластеризована за датою замовлення, країною і статусом, не впорядкована для фільтра лише за країною і статусом. Таблиці, менші за 64 МБ, виграють від цього мало, а на кластеризованій таблиці кількість фактично просканованих байтів стає відомою лише після завершення запиту. Розташовуйте стовпці кластеризації в тому порядку, у якому фільтрують запити. Якщо витрати треба обмежити ще до запуску, покладайтеся на

партиціювання, відсікання партицій якого пробний запуск уже враховує, і задайте максимальну кількість оплачуваних байтів, щоб запит понад ліміт зупинявся до того, як за нього виставлять рахунок.

Запит, що з'єднує два набори даних, відхиляється. Кожен набір даних розміщений в одному місці, і запит виконується в одному місці, тож усі набори, яких він торкається, мусять мати спільне розташування. Команди, що дають кожному проєкту взяти власне значення за замовчуванням, отримують таблиці, які можна з'єднати лише після копіювання однієї з них. Оберіть розташування один раз — з огляду на вимоги відповідності й на те, звідки надходять дані, — і свідомо створюйте кожен набір даних саме там.

Що почитати далі

- BigQuery overview — власний опис сервісу від Google: окремі шари зберігання й обчислень і те, чим керує сам сервіс.
- Introduction to partitioned tables — типи партиціювання, їхня гранулярність і обмеження.
- Query partitioned tables — які фільтри відсікають партиції, а які змушують запит пройти їх усі.
- Introduction to clustered tables — як порядок стовпців визначає, які блоки буде пропущено, і що це означає для оцінки вартості.
- Estimate and control costs — пробні запуски, ліміт оплачуваних байтів і безкоштовні способи переглянути таблицю.
- BigQuery locations — регіони, мультирегіони й правила для запитів, що зачіпають кілька наборів даних.

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 BigQuery?» 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/bigquery/