Engineer

Набір даних

сукупність даних, оприлюднена як одне ціле

Що це таке

Набір даних, англійською dataset, — це обмежений масив даних, з яким поводяться як з однією річчю: він має назву, джерело, період, який охоплює, і форму, спільну для всіх його записів, і ці чотири ознаки визначають, де він починається й де закінчується. Показання лічильників за місяць в одному файлі — набір даних; так само й нічний експорт трьох пов'язаних таблиць. Слово нічого не каже про розмір чи формат, лише про те, що дані всередині належать одне до одного.

Сторінки цього сайту вживають слово саме в такому робочому значенні. Одна говорить про історичний набір даних у форматі JSONL і про набори даних за 1 день і за 30 днів для тестів продуктивності, друга — про впорядковані набори даних і таблиці, експортовані для завантаження деінде, третя — про величезні набори даних від розумних лічильників. Щоразу слово проводить межу навколо даних, перш ніж із ними щось робитимуть.

Словник W3C Data Catalog Vocabulary, DCAT, дає слову формальну рамку. Версія 3 стала рекомендацією W3C 22 серпня 2024 року й визначає набір даних як зібрання даних, оприлюднене або впорядковане одним агентом і доступне для отримання чи завантаження в одній або кількох серіалізаціях або форматах. Поняття навмисно широке: числа, текст, зображення й звук — усе підходить. Далі DCAT розводить три речі, які в щоденному мовленні змішуються:

- **Dataset**: дані як ідея, наприклад погодинне споживання за зонами за 2024 рік.
- **Distribution**: одна доступна форма набору, наприклад CSV-файл для завантаження; в одного набору даних їх може бути кілька.

- **Dataset series**: набори даних, які оприлюднюють окремо, але які мають досить спільного, щоб їх згрупувати, наприклад по одному файлу на місяць.

Тож два файли в різних форматах можуть бути одним набором даних, а два місячні файли під однією назвою — двома членами серії.

У Google BigQuery слово має друге, вужче значення. Там набір даних — це контейнер для таблиць, подань та інших ресурсів, і він міститься всередині проєкту. Таблиця мусить належати до набору даних, тож набір створюють до завантаження будь-яких даних, а звертаються до нього як `project.dataset` у GoogleSQL і як `project:dataset` в інструменті командного рядка bq. Яке значення мається на увазі, вирішує контекст.

Застосуйте поняття до рішення

Застосуйте «Що таке набір даних?» до конкретного рішення. Ми можемо пов'язати поняття із системою, яку ви створюєте або купуєте. Дашборди і звіти, що показують ті кілька цифр, від яких залежить рішення, й оновлюються без нагадувань.

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. Комунікація бренду з урахуванням реакцій аудиторії — комунікація бренду з урахуванням реакцій аудиторії

Комунікація бренду з урахуванням реакцій аудиторії Зв'язатися з нами →

Чому це важливо

Межа робить дані порівнянними. Два масиви показань можна порівняти лише тоді, коли обидва охоплюють той самий період, ті самі місця й той самий крок між записами, а назва набору даних і є способом закріпити ці умови. DCAT виділяє для кроку окрему властивість, temporal resolution, тобто мінімальну відстань між елементами в наборі даних, записану як тривалість, наприклад `P1D` для одного дня. Набір, що вказує свою роздільність, ще до першої суми каже читачеві, чи його рядки — це години, чверті години чи щось нерегулярне.

Опис має подорожувати разом із даними. DCAT перелічує, що дистрибуція може сказати про себе: тип медіа, формат, формат стиснення, формат пакування, розмір у байтах і контрольну суму, остання з яких дає отримувачеві змогу перевірити, що вміст лишився незмінним. Власний

приклад формату стиснення в специфікації — gzip. Файл, що приходить із цими відомостями, можна перевірити й відкрити; файл, що приходить голим, доводиться розгадувати.

Формат дистрибуції визначає, хто зможе її прочитати. Поширені три:

- **CSV.** RFC 4180 від 2005 року описує розкладку, якої, схоже, дотримується більшість реалізацій: один запис на рядок, необов'язковий рядок заголовка, подвійні лапки навколо будь-якого поля з комою, розривом рядка чи подвійною лапкою, а подвійна лапка всередині поля записується двічі. Файли CSV можна відкрити майже будь-яким інструментом.
- **JSON Lines.** Одне коректне значення JSON на рядок у кодуванні UTF-8, із `\n` між рядками. Файл зберігають як `.jsonl` або, стиснувши gzip, як `.jsonl.gz`. Після останнього значення JSON у файлі настійно рекомендовано додати символ нового рядка, але це не обов'язково. Програми читають його по одному запису, а текстові інструменти в стилі Unix і конвеєри командної оболонки добре з ним працюють.
- **Parquet.** Відкритий стовпцево орієнтований формат файлів даних, створений для ефективного зберігання й отримання даних, зі схемами стиснення й кодування для роботи зі складними даними у великих обсягах.

Ось один запис у перших двох форматах; кома в примітці змушує взяти поле CSV в лапки:

```
id,read_at,kwh,note
M-17,2024-01-08T10:00:00Z,3.21,"late, manual"

{"id":"M-17","read_at":"2024-01-08T10:00:00Z","kwh":3.21,"note":"late, manual"}
```

Як це виглядає на практиці

Набір даних про електроенергію спершу треба було сформувати: дані надходили з кількох джерел, кожне з власною частотою звітності, від погодинних до 15-хвилинних інтервалів, а подекуди з нерегулярними часовими мітками, тож скласти цілісний, порівнянний набір даних було важко. Звідси й завдання: перетворити сирі дані про події з історичного набору JSONL на погодинне споживання, по одному рядку на кожну годину й зону: підсумувати мікс виробництва в кожному інтервалі, додати імпорт і

відняти експорт. Написано три варіанти — на Python, JQ і SQL. Пайплайн на Python парсив файли JSONL у датафрейми, зводив часові ряди до погодинних інтервалів і експортував результати у CSV або завантажував їх у легку базу даних SQLite для перевірки. На історичних наборах даних за 1 день і за 30 днів він виявився найшвидшим із трьох: 34 мс на один день і 180 мс на тридцять. Робота між входом і виходом — це зміна зернистості: від власного кроку кожного джерела до одного фіксованого рядка на годину й зону.

Переміщення — окрема робота: шість ETL/ELT-пайплайнів переносили дані з віддаленого сховища даних за IPsec VPN у Google BigQuery. Дані надходили як стиснуті архіви з файлами Parquet, CSV і .bak, а скрипти на Bash і Python класифікували кожен файл за типом і схемою. Коли процедури очистили, нормалізували й збагатили рядки, впорядковані набори даних і таблиці експортували в проміжні формати, стиснули gzip і передали в бакет Google Cloud Storage, звідки bq CLI і скрипти на Python завантажували їх у BigQuery. Понад 3 години ручної роботи перетворилися на менш ніж 20 хвилин на весь цикл, а синхронізація стала щоденною замість щотижневої.

Розумні лічильники в кількох географічних регіонах давали величезні набори даних, на яких трималися понад 500 аналітичних звітів на чистому SQL у PostgreSQL і BigQuery. Дані охоплювали геолокаційні координати, ідентифікатори лічильників, дані про споживання енергії з часовими мітками та метадані про довкілля. Кожен звіт містив анотований SQL, щоб колеги могли відтворити й перевірити його, а нотатки з усунення несправностей документували типові проблеми якості даних, як-от відсутні GPS-координати або пошкоджені значення лічильників, разом із рекомендованою обробкою. Сторінка пов’язує цю роботу з підвищенням ефективності операційного планування на 10–15%.

Де виникають проблеми

Більшість проблем із наборами даних — це факт про дані, що залишився в голові однієї людини.

Зернистість лишається незаписаною. Якщо скласти погодинні рядки з п’ятнадцятихвилинними, частина інтервалів порахується вчетверо. Запишіть зернистість одним реченням, наприклад один рядок на годину на зону в UTC, і вирівняйте за нею кожне джерело, перш ніж підсумовувати.

Пропуск читається як нуль. Порожнє поле, null і 0 означають три різні речі, а сума непомітно вважає відсутнє показання нульовим споживанням. Відсутні GPS-координати й пошкоджені значення лічильників були в проєкті з лічильниками настільки типовими, що для них задокументували процедури обробки. Для кожного стовпця вирішіть, чи пропуск відкидають, заповнюють чи позначають, і запишіть вибір у примітках до набору даних.

Діалект CSV дрейфує. RFC 4180 описує поширену розкладку, а реальні файли від неї відходять: інший роздільник, пропущений заголовок, поле з комою, але без лапок. RFC робить рядок заголовка необов'язковим і дозволяє сигналізувати про нього параметром MIME, який губиться, щойно файл скопіювали на диск. Перед завантаженням звіряйте перший рядок кожного нового файлу з очікуваним заголовком.

Копія забуває, звідки взялася. Впорядкований витяг, завантажений в іншу систему, за місяць уже виглядає точнісінько як оригінал. Біля кожної копії записуйте джерело, період, час вивантаження й контрольну суму, щоб звіт міг назвати витяг, який він прочитав.

Набір даних BigQuery опиняється не там. Розташування набору даних задають під час створення, і змінити його потім можна лише копіюванням або повторним створенням набору. Усі таблиці в одному запиті мають лежати в наборах даних з однаковим розташуванням, тож обирайте розташування до першого завантаження.

Що почитати далі

- Data Catalog Vocabulary (DCAT) - Version 3 — визначення набору даних, дистрибуції й серії наборів даних від W3C разом із властивостями, що описують кожне з них.
- BigQuery: Introduction to datasets — значення слова як контейнера: проєкти, розташування, іменування й обмеження.
- JSON Lines — три правила формату та домовленості про назви файлів.
- RFC 4180: Common Format and MIME Type for CSV Files — розкладка CSV, якої дотримується більшість інструментів, правила лапок і тип text/csv.
- Apache Parquet: Overview — що таке цей стовпцевий формат файлів і де лежить його специфікація.

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 набір даних?» 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/dataset/