Engineer

Озеро даних

≋ сховище сирих даних у їхньому початковому вигляді

Що це таке

Озеро даних — англійською data lake — це сховище, яке зберігає сирі дані в тому вигляді, в якому вони надходять із джерела, і на найдетальнішому рівні, який це джерело дає, щоб пізніше з них можна було відповісти й на питання, яких ніхто заздалегідь не ставив. На практиці дані лежать у файлах, часто в стовпцевому форматі на кшталт Apache Parquet, у розподіленій файловій системі або в об'єктному сховищі, а рушії запитів читають ці файли безпосередньо.

Термін запропонував Джеймс Діксон із Pentaho. У дописі в блозі від 14 жовтня 2010 року, написаному того тижня, коли Pentaho оголосила свій перший реліз для Hadoop, він підсумував те, що почув від компаній, які користувалися Hadoop. Вісім-дев'ять компаній із десяти мали справу зі структурованими або напівструктурованими, а не неструктурованими даними, і ці дані зазвичай надходили з одного застосунку чи однієї системи. Кілька питань до них були відомі заздалегідь, значно більше мало з'явитися згодом, ставити їх хотіли кілька груп користувачів, а обсяг не вміщувався в реляційну базу даних ні технічно, ні за розумну ціну. Звичною відповіддю було вибрати найцікавіші атрибути й агрегувати їх у вітрину даних (data mart). Так можна відповісти лише на питання, обрані наперед, а агрегування назавжди ховає найнижчий рівень деталізації.

Діксон запропонував натомість образ. Вітрина даних — як бутильована вода: очищена, упакована й подана у формі, яку легко спожити. Озеро даних, за його словами, — це «велика водойма в природнішому стані». Джерело наповнює його, і кожен, хто має питання, може прийти подивитися, пірнути чи набрати пробу.

Через чотири роки Діксон повернувся до терміна, бо його широко вживали, але визначали дуже по-різному. Він повторив своє визначення: озеро містить дані з одного джерела, не агреговані заздалегідь, і в організації може бути кілька озер. Якщо зберігати дані з багатьох систем і поєднувати їх між собою, писав він, вийде водний сад (water garden), а не озеро. Так само чітко він наголошував, що в 2010 році озеро не замінювало ні бази даних, ні вітрини даних, ні сховища даних. Кілька визначень, на які він відповідав, описували одне спільне сховище сирих даних із багатьох систем. Це ширше значення слова, і варто знати, яке з двох має на увазі колега.

Застосуйте поняття до рішення

Застосуйте «Що таке озеро даних?» 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

Кожен підсумок — це вибір того, які питання важливі. Щоденна сума продажів по кожному магазину показує, скільки продав кожен магазин. Вона не скаже, яка година була найнапруженішою, які товари люди переглядали, але не купили, чи повертався той самий покупець тричі, перш ніж замовити. Коли зберігається лише сума, ці питання зникають назавжди, бо рядки, які могли б на них відповісти, викинули, щойно суму обчислили. Діксон доводив, що питання з'являються після даних, тож сирі записи варто зберігати на найдетальнішому рівні, а агрегувати тоді, коли питання вже відоме.

Сховище даних від цього не стає зайвим. Діксон описував, як усталені вітрини й сховища даних організації наповнюють агрегатами, взятими з озера, а нерегламентовані (ad hoc) запити йдуть прямо до озера. Сховище лишається місцем для цифр, щодо яких організація домовилася. До озера звертаються, коли нове питання потребує деталей, яких сховище ніколи не зберігало, і саме з нього перебудовують таблицю сховища, коли змінюється її визначення.

Формат файлів важить не менше за саму ідею. Parquet зберігає кожен стовпець групи рядків (row group) окремим блоком, а метадані в кінці файлу фіксують, де починається кожен блок. Програма, яка читає файл, спершу переглядає ці метадані, а потім завантажує лише ті блоки стовпців, які їй потрібні, тож запит, що зачіпає три стовпці із сорока, не читає решти тридцяти семи. Оскільки Parquet — відкритий формат і кожен файл несе власні метадані, будь-який рушій, що читає Parquet, може робити запити до тієї самої теки, зокрема й DuckDB на ноутбуці, без жодного кроку експорту.

Як це виглядає на практиці

Інтернет-магазин реєструє кожен перегляд товару, кожне додавання в кошик і кожну покупку. Замість того щоб зводити їх у щоденні суми, він зберігає кожну подію окремо й записує події у файли Parquet, по одній теці на день. У DuckDB 2.0 обидва кроки — звичайний SQL:

```
CREATE TABLE shop_events AS
SELECT * FROM (VALUES
  (TIMESTAMP '2026-10-26 09:14:03', 'view',     'boots-41', NULL::INTEGER),
  (TIMESTAMP '2026-10-26 09:15:40', 'cart',     'boots-41', NULL),
  (TIMESTAMP '2026-10-26 09:17:12', 'purchase', 'boots-41', 1200),
  (TIMESTAMP '2026-10-27 11:02:55', 'view',     'jacket-m', NULL),
  (TIMESTAMP '2026-10-27 11:03:30', 'view',     'boots-41', NULL),
  (TIMESTAMP '2026-10-27 11:05:01', 'cart',     'jacket-m', NULL)
) t(occurred_at, event, product, amount);

COPY (SELECT *, CAST(occurred_at AS DATE) AS day FROM shop_events)
TO 'events' (FORMAT parquet, PARTITION_BY (day));
```

`PARTITION_BY` приймає лише назви стовпців, а не вирази, тому день обчислюється в `SELECT` . DuckDB записує по одній теці на кожне значення й дає їй назву у форматі key=value, який документація називає Hive partitioning:

```
events/day=2026-10-26/data_0.parquet
events/day=2026-10-27/data_0.parquet
```

Питання про один день фільтрує за цим стовпцем:

```
SELECT event, count(*) AS n
FROM   read_parquet('events/*/*.parquet', hive_partitioning = true)
WHERE  day = DATE '2026-10-27'
GROUP  BY event
ORDER  BY n DESC;
-- view 2, cart 1
```

DuckDB передає фільтр за стовпцем партиції аж до назв тек і пропускає файли, які не можуть дати збігу. `EXPLAIN ANALYZE` для цього запиту показує, що з двох файлів просканована один. Якби тек було за цілий рік, те саме питання відкрило б одну теку з 365. Стовпець `day` повертається як `DATE` без жодних вказівок, бо DuckDB за замовчуванням розпізнає в назвах тек дати, часові мітки та великі цілі числа. Він також сам вмикає Hive partitioning, коли теки відповідають шаблону key=value, тож прапорець у цих запитах лише робить намір явним.

Тепер питання, якого ніхто не планував: які товари переглядали найчастіше?

```
SELECT product, count(*) AS views
FROM   read_parquet('events/*/*.parquet', hive_partitioning = true)
WHERE  event = 'view'
GROUP  BY product
ORDER  BY views DESC, product;
-- boots-41 2, jacket-m 1
```

Таблиця щоденних сум покупок ніколи б на це не відповіла. Сирі події відповідають на нього ще одним запитом до тієї самої теки, бо кожну подію зберегли ще на вході.

Послуга компанії з проєктування сховищ даних описує мету як одне місце, де всі цифри збігаються, щоб два відділи не приносили на одну нараду дві різні відповіді. Озеро на кшталт цього розташоване вище за течією від такого сховища: сховище містить узгоджені цифри, а сирі події лишаються під рукою для наступного питання.

Де виникають проблеми

Файли, які ніхто не може пояснити. Тека з файлами Parquet без жодного запису про те, звідки вони взялися, хто ними володіє і що означає кожен стовпець, — це озеро, яким може користуватися лише той, хто його збудував. Коли критики у 2014 році стверджували, що в озера даних немає метаданих, Діксон відповів, що в самій ідеї нічого такого немає, а згодом додав, що метадані знадобляться кожному, хто поєднує дані з різних джерел або відстежує їхнє походження (lineage). Ведіть у каталозі даних по одному запису на кожен набір даних — джерело, власник, стовпці, час завантаження й строк зберігання — і зробіть створення цього запису частиною завдання завантаження.

Тисячі дрібних файлів. Кожен файл Parquet має власний блок метаданих, і програма, що його читає, мусить прочитати цей блок, перш ніж дістатися до даних. Потік, який щохвилини записує один невеликий файл, за 30-денний місяць залишає понад сорок тисяч файлів, і кожен із них відкривається й розбирається під час кожного повного сканування. Збирайте записи в меншу кількість більших файлів і за розкладом об'єднуйте старі партиції (compaction). Табличний формат Apache Iceberg фіксує таке об'єднання як знімок типу `replace` : кілька файлів замінено одним, що містить ті самі рядки.

Запити, які бачать незавершене завантаження. Завдання, яке записує десять файлів і зупиняється після шостого, залишає в теці шість файлів, і все, що переглядає вміст теки, їх прочитає. Звичайна тека не має поняття коміту. Цю проблему розв'язує табличний формат. Iceberg відстежує в метаданих окремі файли даних, а не цілі теки, і фіксує зміну, атомарно замінюючи один файл метаданих наступним. Читання завжди відбувається з уже зафіксованого знімка, записи ніколи не видно частково, а для читання не потрібні жодні блокування.

Перейменовані стовпці, що розколюють історію. Старі файли зберігають стару назву стовпця, і кожен запит мусить знати обидві. Iceberg обирає стовпці у файлах даних за ідентифікатором поля, а не за назвою, тож стовпець можна перейменувати, додати, видалити чи перемістити, навіть усередині вкладених структур, не переписуючи вже записаних файлів.

Одне озеро для всіх систем. Злити всі бази даних організації в одне місце й поєднувати все між собою — це те, що Діксон назвав водним садом. Він радив підприємству використовувати різні види сховищ, добираючи кожен

під дані й запити, які воно обробляє. Його порада — брати найкращий інструмент для кожного завдання: транзакційну базу даних для транзакційних цілей, а аналітичну — для аналітичних. Починайте озеро з одного джерела, на дані якого вже чекають реальні питання, а узгоджені зведені цифри будуйте в сховищі даних, яке наповнюється з озера.

Персональні дані, що зберігаються вічно. Сховище, задумане для того, щоб зберігати все, стикається з вимогами видалити дані та з обмеженнями строку зберігання. Iceberg версії 2 може видаляти окремі рядки, записуючи файли видалень (delete files), без переписування файлів даних, а отже, початкові байти лишаються в сховищі. Вони зникають тільки тоді, коли файли даних переписано, а старіші знімки, які ще на них посилаються, втратили чинність (expire), після чого файли можна фізично видалити. Не тримайте прямих ідентифікаторів у сирих подіях, де це можливо, партиціонуйте так, щоб дані з вичерпаним строком зберігання видалялися цілими теками, і плануйте об’єднання файлів та закінчення строку дії знімків після видалень.

Що почитати далі

- Pentaho, Hadoop, and Data Lakes — допис Джеймса Діксона за жовтень 2010 року, у якому з’явилися термін і порівняння з бутильованою водою.
- Data Lakes Revisited — продовження Діксона 2014 року: одне джерело на озеро, водний сад і його відповіді критикам.
- File Format — будова Parquet: групи рядків, блоки стовпців і метадані файлу, записані після даних.
- Iceberg Table Spec — специфікація Apache Iceberg: знімки, атомарні коміти, еволюція схеми за ідентифікаторами полів, файли видалень і закінчення строку дії знімків.
- Hive Partitioning — документація DuckDB про теки key=value, filter pushdown і визначення типів партицій.

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 озеро 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/data-lake/