Engineer

Сховище даних

база даних, що збирає дані організації для аналізу

Що це таке

Сховище даних, англійською data warehouse, — це база даних, яку будують, щоб відповідати на запитання про бізнес, а не щоб вести його. Системи, які приймають замовлення, оформлюють відправлення, виставляють рахунки й реєструють звернення до підтримки, тримають кожна свої дані у своїй формі. Сховище даних отримує копії з усіх них, зберігає їх роками й упорядковує так, щоб запитання, яке зачіпає кілька систем, ставало одним запитом, а не тижнем вивантажень у електронні таблиці. Саму роботу з побудови й супроводу такої системи англійською називають data warehousing.

Відмінність від баз, з яких копіюють дані, — у навантаженні. Система замовлень читає й записує кілька рядків за раз, тисячі разів на хвилину, і тримає лише теперішнє: коли клієнт переїжджає, стару адресу перезаписують. Сховище даних наповнюють великими порціями, читають запитами, що проглядають мільйони рядків за раз, а минуле в ньому зберігають навмисно, щоб торішні цифри можна було розкласти так, як бізнес виглядав торік.

Ральф Кімбол приніс вимірне моделювання в галузь 1996 року, і Kimball Group ділить таку систему на дві частини: технічну (back room) та користувацьку (front room). У back room працює ETL-система, яку вони розкладають на 34 підсистеми за чотирма завданнями: видобути дані з джерел, очистити й узгодити їх, доставити до презентаційної зони (presentation area) і керувати всім процесом. Front room — це сама презентаційна зона, тобто те, до чого люди справді звертаються із запитами. Вона впорядкована за бізнес-процесами, тримає дані на найдетальнішому рівні, який фіксує кожен процес, має агреговані підсумки для швидкості й побудована з таблиць фактів і вимірів. Навколо неї стоять BI-застосунки:

довільні (ad hoc) запити, стандартні звіти, дашборди, картки показників і аналітичні застосунки. Front room, за словами Kimball Group, — це те, чим уся система повернута до людей.

Це не єдиний варіант устрою. Сторінка Kimball Group про архітектуру описує й Corporate Information Factory — побудову за принципом hub-and-spoke, де атомарні дані лежать у нормалізованому корпоративному сховищі в третій нормальній формі, а другий крок ETL завантажує з нього вимірні таблиці для звітності, — і гібрид, у якому наявне нормалізоване сховище постачає дані до вимірної презентаційної зони. Спільне для всіх варіантів — розділення: системи, що ведуть бізнес, не є тим місцем, де його аналізують.

Сховище даних — це роль, а не продукт. Ним може бути база PostgreSQL чи Microsoft SQL Server на сервері, який команда обслуговує сама, або керований сервіс на кшталт Google BigQuery, який Google описує як повністю керовану платформу даних, що поєднує хмарне сховище даних з аналітичними інструментами. BigQuery тримає зберігання й обчислення окремо, тож кожне масштабується незалежно, зберігає дані за стовпцями, а не за рядками, і приймає запити стандартним ANSI SQL.

Застосуйте поняття до рішення

Застосуйте «Що таке сховище 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

Бізнес, який ставить те саме запитання двом системам, отримує дві відповіді. Відділ продажів рахує клієнта, коли оформлено замовлення, фінанси — коли оплачено рахунок, підтримка — щоразу, коли надходить звернення. Ніхто з них не помиляється щодо власних цілей, і ніхто не сходиться з рештою. Сховище даних — місце, де одне визначення погоджують і записують у завантаження, щоб кожен звіт, побудований на ньому, рахував однаково.

Інструмент Kimball Group для цього — узгоджений вимір (conformed dimension): один клієнт, один продукт, один календар, які централізовано підтримують в ETL-системі й спільно використовують усі таблиці фактів, яким вони потрібні. Таблицю фактів продажів і таблицю фактів звернень, що посилаються на той самий вимір клієнтів, можна порівнювати рядок за рядком. Їхня шинна архітектура (bus architecture) зводить бізнес-процеси й спільні для них виміри в матрицю, яку Kimball Group називає архітектурним кресленням, і дає змогу будувати сховище по одному бізнес-процесу за раз, що, за їхніми словами, скорочує час до виходу на ринок. Команда може цього кварталу здати замовлення, а наступного — повернення, не переробляючи вже зробленого, доки обидва процеси використовують ті самі виміри.

Історія — друга причина. Операційні системи тримають поточний стан; сховище даних може тримати кожен. Техніка Kimball для атрибута, що змінюється, відома як тип 2, додає до виміру новий рядок із новими значеннями й новим сурогатним ключем, і кожен факт, завантажений від цієї миті, вказує на новий рядок. Продажі до переїзду клієнта лишаються за старим регіоном, продажі після нього йдуть до нового. Щонайменше три додаткові стовпці — дата початку дії, дата завершення дії й позначка поточного рядка — дають змогу спитати, що було правдою на будь-яку дату.

Третя причина — звичайне навантаження. Звітові, який проглядає замовлення за три роки, не місце в базі, що приймає сьогоднішні замовлення: там він змагається з оформленням покупок за ту саму пам’ять, ті самі диски й ті самі блокування. Коли аналітика переїжджає в окрему базу, кожну сторону можна налаштувати під її власну роботу.

Ціна в тому, що сховище даних — копія. Воно таке свіже, як останнє завантаження, і таке правильне, як правила в ньому, тож хтось мусить відповідати за обидва: за розклад запусків і за визначення продажу, клієнта й місяця, про які домовився бізнес.

Як це виглядає на практиці

Серце вимірного сховища — таблиця фактів, чию гранулярність (grain) визначають першою, тут один запис на кожну позицію замовлення, а навколо неї — виміри, що її описують. Вимір клієнтів має стовпці для типу 2, тож клієнт, який переїхав, отримує другий рядок, а не перезаписаний.

```
CREATE TABLE dim_customer (
    customer_key bigint PRIMARY KEY,   -- сурогатний ключ, належить сховищу
    customer_id  text NOT NULL,        -- ключ у системі-джерелі
    region       text NOT NULL,
    valid_from   date NOT NULL,
    valid_to     date,                 -- порожньо, поки рядок поточний
    is_current   boolean NOT NULL
);

CREATE TABLE fact_order_line (
    date_key     integer NOT NULL REFERENCES dim_date,
    customer_key bigint  NOT NULL REFERENCES dim_customer,
    product_key  bigint  NOT NULL REFERENCES dim_product,
    quantity     integer NOT NULL,
    net_amount   numeric(12, 2) NOT NULL
);

SELECT d.year, c.region, sum(f.net_amount) AS revenue
FROM fact_order_line AS f
JOIN dim_date AS d USING (date_key)
JOIN dim_customer AS c USING (customer_key)
GROUP BY d.year, c.region
ORDER BY d.year, c.region;
```

Кожну позицію замовлення завантажено з тим рядком клієнта, який був поточним на той момент, тому запит показує виручку за регіоном, де клієнт жив, коли купував. Якщо натомість з'єднувати за `customer_id` і фільтрувати за `is_current`, уся історія перераховується за сьогоднішніми регіонами. Обидва запитання доречні, і модель відповідає на кожне без другої копії даних.

Обіцянка послуги Проєктування сховищ даних — саме ця перша причина: дані продажів, операцій і фінансів зведені разом, щоб два відділи не приносили на одну нараду дві різні відповіді. У більшому масштабі з 2022 по 2024 рік засновниця спроєктувала й експлуатувала 100 сховищ даних на PostgreSQL, Microsoft SQL Server і Google BigQuery, переважно з GIS-даними та часовими рядами, як-от просторові координати й показання сенсорів. Робота поєднувала нормалізовані й денормалізовані структури таблиць із просторовою індексацією (PostGIS у PostgreSQL) і партиціонуванням часових рядів на перших двох платформах, а в BigQuery — партиціоновані за часом і кластеризовані таблиці зі стисненням і колонковим зберіганням. Час відповіді на запити для GIS-даних і часових рядів скоротився на 40–60%, а оптимізована інфраструктура дала змогу організації, яка використовувала ці бази даних, запустити нові продукти на основі даних, наприклад дашборди аналітики в реальному часі.

Де виникають проблеми

Одна таблиця фактів змішує дві гранулярності, і кожен підсумок рахує частину грошей двічі. Замовлення має позиції, а ще має плату за доставку, яка належить замовленню загалом. Якщо записати плату в кожну позицію, сума за позиціями врахує її стільки разів, скільки є позицій. У чотирикроковому процесі проєктування Kimball Group визначення гранулярності стоїть другим, одразу після вибору бізнес-процесу й до того, як названо бодай один вимір чи факт, а їхнє правило таке: різні гранулярності не можна змішувати в одній таблиці фактів. Тримайте окрему таблицю фактів на рівні замовлення для сум, що належать усьому замовленню, або розподіліть плату між позиціями за правилом, яке погодить бізнес.

Клієнт переїжджає, і торішній звіт змінюється. Коли регіон у рядку клієнта просто перезаписують, усі минулі продажі переходять разом із клієнтом до нового регіону, і звіт, затверджений у січні, у червні розповідає іншу історію. Атрибути, історія яких має значення, ведіть за типом 2: закрийте поточний рядок і додайте новий із новим сурогатним ключем в одній транзакції бази даних, щоб жодне завантаження не побачило двох поточних рядків одночасно.

```
BEGIN;
UPDATE dim_customer
SET valid_to = current_date, is_current = false
WHERE customer_id = 'C-1042' AND is_current;
INSERT INTO dim_customer
    (customer_key, customer_id, region, valid_from, valid_to, is_current)
VALUES (90731, 'C-1042', 'Північ', current_date, NULL, true);
COMMIT;
```

Дві команди будують два виміри клієнтів, і сховище повторює суперечку, яку мало завершити. Кожну нову предметну область швидше здати з власною копією списку клієнтів чи продуктів, і кожна копія поступово розходиться з іншими: одна відкидає закриті облікові записи, інша інакше зливає дублікати. Побудуйте спільні виміри один раз в ETL-системі й подбайте, щоб кожна нова таблиця фактів використовувала саме їх. Матриця шини і є контрольним списком: перш ніж додати бізнес-процес, вона показує, на які наявні виміри має посилатися нова таблиця фактів.

Звіти спрямовано на проміжні (staging) таблиці. Сирі копії джерел спершу потрапляють у back room, і коли вони вже там, кортить надсилати запити просто до них, бо дані вже завантажено. Ці таблиці несуть власні коди кожного джерела, дублікати й прогалини та змінюються щоразу, коли змінюється джерело, тож кожен звіт на них успадковує саме ту проблему, яку сховище даних мало розв'язати. Залиште back room тільки для ETL-системи, а звітам, дашбордам і аналітикам дайте лише презентаційну зону, де узгоджені виміри й визначена гранулярність і змушують цифри сходитися.

Що почитати далі

- Kimball Technical DW/BI System Architecture — back room і front room, підсистеми ETL, метадані та альтернативи: hub-and-spoke і гібрид.
- Enterprise Data Warehouse Bus Architecture — узгоджені виміри, матриця шини й побудова по одному бізнес-процесу за раз.
- Four-Step Dimensional Design Process — бізнес-процес, гранулярність, виміри й факти, саме в такому порядку.
- Type 2: Add New Row — історія атрибута, що змінюється, через новий рядок, новий сурогатний ключ і три додаткові стовпці.
- BigQuery overview — кероване сховище даних Google, у якому зберігання й обчислення розділено, а дані зберігаються за стовпцями.

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 сховище 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-warehouse/