Engineer

Datavarehus

en database som samler virksomhetens data for analyse

Hva det er

Et datavarehus er en database som er bygget for å svare på spørsmål om en virksomhet, ikke for å drive den. Systemene som tar imot bestillinger, booker forsendelser, sender ut fakturaer og registrerer supporthenvendelser, har hver sine data i hver sin form. Et datavarehus får kopier fra alle sammen, tar vare på dem i årevis og ordner dem slik at et spørsmål som går på tvers av flere systemer, blir én spørring i stedet for en uke med eksport til regneark. På engelsk heter det data warehouse, et ord som også brukes i norske fagtekster, og arbeidet med å bygge og drive det kalles data warehousing.

Forskjellen fra databasene det kopierer fra, ligger i arbeidsmengden. Et ordresystem leser og skriver noen få rader om gangen, tusenvis av ganger i minuttet, og holder bare på nåtiden: når en kunde flytter, blir den gamle adressen overskrevet. Et datavarehus skrives i store porsjoner, leses av spørringer som går gjennom millioner av rader på én gang, og tar vare på fortiden med vilje, slik at fjorårets tall fortsatt kan brytes ned slik virksomheten så ut i fjor.

Ralph Kimball tok dimensjonsmodellering inn i bransjen i 1996, og Kimball Group deler et slikt system i to rom. Back room rommer ETL-systemet, som de deler i 34 delsystemer fordelt på fire oppgaver: å hente data ut av kildene, å rense og avstemme dem, å levere dem til presentasjonsområdet og å styre hele prosessen. Front room rommer presentasjonsområdet, som er det folk faktisk sender spørringer mot. Det er organisert etter forretningsprosess, holdt på det mest detaljerte nivået hver prosess registrerer, supplert med aggregerte sammendrag for hastighetens skyld og bygget opp av fakta- og dimensjonstabeller. Rundt det står BI-applikasjonene: ad hoc-spørringer, standardrapporter, dashbord, målstyringskort (scorecards) og analyseapplikasjoner. Kimball Group beskriver front room som den siden av systemet brukerne møter.

Det er ikke den eneste oppbygningen. Kimball Groups egen arkitekturside beskriver Corporate Information Factory, et hub-and-spoke-design der atomære data ligger i et normalisert datavarehus for hele virksomheten i tredje normalform, og der et andre ETL-steg fyller dimensjonelle tabeller til rapportering, og en hybrid der et eksisterende normalisert datavarehus forsyner et dimensjonelt presentasjonsområde. Felles for alle er skillet: systemene som driver virksomheten, er ikke stedet der den analyseres.

Et datavarehus er en rolle, ikke et produkt. Det kan være en PostgreSQL- eller Microsoft SQL Server-database på en server teamet drifter selv, eller en administrert tjeneste som Google BigQuery, som Google beskriver som en fullt administrert dataplattform som kombinerer et datavarehus i skyen med analyseverktøy. BigQuery holder lagring og regnekraft adskilt, slik at hver del skalerer for seg, lagrer data kolonnevis i stedet for radvis og tar imot spørringer i ANSI-standard-SQL.

Bruk begrepet i en beslutning

Bruk «Hva er et datavarehus?» i en konkret beslutning. Vi kan knytte begrepet til systemet dere bygger eller kjøper. Tall fra salg, drift og økonomi ført sammen, slik at to avdelinger slutter å komme med to ulike svar til det samme møtet.

Se tjenestene våre Ta kontakt →

Undersøk antakelsene bak ordet

Bruk forklaringen og det tilknyttede arbeidet til å formulere spørsmål for prosjektet. Vi kan undersøke antakelsene sammen før et verktøy velges. Gjorde en geodatapipeline 50 ganger raskere — 50x raskere pipelines for geografiske data

Gjorde en geodatapipeline 50 ganger raskere Ta kontakt →

Hvorfor det er viktig

En virksomhet som stiller det samme spørsmålet til to systemer, får to svar. Salg teller en kunde når bestillingen legges inn, økonomi når fakturaen er betalt, og support hver gang det kommer en henvendelse. Ingen av dem tar feil ut fra sitt eget formål, og ingen av dem stemmer med de andre. Datavarehuset er stedet der én definisjon blir avtalt og skrevet ned i lastingen, slik at hver rapport som bygger på den, teller på samme måte.

Kimball Groups verktøy for dette er den felles, avstemte dimensjonen, på engelsk conformed dimension: én kunde, ett produkt, én kalender, vedlikeholdt sentralt i ETL-systemet og delt av hver faktatabell som trenger den. En faktatabell for salg og en for supporthenvendelser som peker på den samme kundedimensjonen, kan sammenlignes linje for linje. Bussarkitekturen deres stiller forretningsprosessene opp mot dimensjonene de deler, i en matrise de

kaller den arkitektoniske tegningen, og den gjør det mulig å bygge datavarehuset én forretningsprosess om gangen, noe som ifølge Kimball Group gir kortere tid til markedet. Et team kan levere bestillinger dette kvartalet og returer det neste uten å bygge om det som allerede er levert, så lenge begge bruker de samme dimensjonene.

Historikk er den andre grunnen. Driftssystemer holder den gjeldende tilstanden; et datavarehus kan holde hver tilstand. Kimballs teknikk for en egenskap som endrer seg, kalt type 2, legger til en ny rad i dimensjonen med de nye verdiene og en ny surrogatnøkkel, og hver faktarad som lastes fra det øyeblikket, peker på den nye raden. Salg fra før kunden flyttet, blir værende i den gamle regionen; salg etterpå havner i den nye. Minst tre ekstra kolonner, en gyldig fra-dato, en gyldig til-dato og en markør for gjeldende rad, gjør det mulig å spørre hva som var sant på en gitt dato.

Den tredje grunnen er ren belastning. En rapport som går gjennom tre års bestillinger, hører ikke hjemme på databasen som tar imot dagens bestillinger, der den konkurrerer med kassen om det samme minnet, de samme diskene og de samme låsene. Når analysen flytter til en egen database, kan hver side innstilles for sitt eget arbeid.

Prisen er at et datavarehus er en kopi. Det er så ferskt som den siste lastingen og så riktig som reglene i den, og noen må eie begge deler: kjøreplanen og de definisjonene av et salg, en kunde og en måned som virksomheten er blitt enig om.

Slik ser det ut i praksis

Kjernen i et dimensjonelt datavarehus er en faktatabell der granulariteten, på engelsk grain, er fastsatt først, her én rad per ordrelinje, omgitt av dimensjonene som beskriver den. Kundedimensjonen har kolonnene for type 2, slik at en kunde som flytter, får en ny rad i stedet for en overskrevet.

```
CREATE TABLE dim_customer (
    customer_key bigint PRIMARY KEY,   -- surrogatnøkkel, eid av varehuset
    customer_id  text NOT NULL,        -- nøkkelen i kildesystemet
    region       text NOT NULL,
    valid_from   date NOT NULL,
    valid_to     date,                 -- tom så lenge raden er gjeldende
    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;
```

Fordi hver ordrelinje ble lastet med kunderaden som var gjeldende da, viser spørringen omsetningen fordelt på regionen kunden bodde i da kjøpet skjedde. Joiner man i stedet på `customer_id` og filtrerer på `is_current`, blir hele historikken gjort opp etter dagens regioner. Begge spørsmålene er legitime, og modellen svarer på begge uten en ekstra kopi av dataene.

Løftet i tjenesten Design av datavarehus er nettopp den første grunnen: tall fra salg, drift og økonomi ført sammen, slik at to avdelinger slutter å komme med to ulike svar til det samme møtet. I større skala utformet og driftet grunnleggeren 100 datavarehus i PostgreSQL, Microsoft SQL Server og Google BigQuery fra 2022 til 2024, hovedsakelig med GIS- og tidsseriedata som romlige koordinater og sensoravlesninger. Arbeidet kombinerte normaliserte og denormaliserte tabelloppsett med romlig indeksering (PostGIS i PostgreSQL) og tidsseriepartisjonering på de to første plattformene, og tidspartisjonerte, klyngede tabeller med komprimering og kolonnebasert lagring i BigQuery. Svartidene for spørringer mot GIS- og tidsseriedata gikk ned med 40–60 %, og den optimaliserte infrastrukturen gjorde det mulig for organisasjonen som brukte disse databasene, å lansere nye datadrevne produkter som dashbord med sanntidsanalyse.

Hvor det går galt

Én faktatabell blander to granulariteter, og hver sum teller noen penger to ganger. En ordre har linjer, og den har også et fraktgebyr som hører til ordren som helhet. Legges gebyret på hver linje, teller en sum over linjene det én gang per linje. I Kimball Groups designprosess i fire steg kommer fastsettelsen av granulariteten som nummer to, rett etter valget av forretningsprosess og før

noen dimensjon eller noe faktum navngis, og regelen deres er at ulike granulariteter ikke skal dele én faktatabell. Hold en egen faktatabell på ordrenivå for beløp som hører til hele ordren, eller fordel gebyret på linjene etter en regel virksomheten godkjenner.

En kunde flytter, og fjorårets rapport endrer seg. Når regionen på kunderaden bare overskrives, følger alt tidligere salg kunden til den nye regionen, og en rapport som ble godkjent i januar, forteller en annen historie i juni. Behandle egenskaper der historikken betyr noe, med type 2: lukk den gjeldende raden og legg til en ny med en ny surrogatnøkkel, i én databasetransaksjon, slik at ingen lasting noen gang ser to gjeldende rader.

```
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', 'Nord', current_date, NULL, true);
COMMIT;
```

To team bygger hver sin kundedimensjon, og diskusjonen datavarehuset skulle avslutte, begynner på nytt. Et nytt fagområde er raskere å levere med sin egen kopi av kunde- eller produktlisten, og hver kopi glir bort fra de andre: én utelater avsluttede kontoer, en annen slår sammen duplikater på en annen måte. Bygg de felles dimensjonene én gang i ETL-systemet, og la hver ny faktatabell bruke dem. Bussmatrisen er sjekklisten: før en forretningsprosess tas med, viser den hvilke eksisterende dimensjoner den nye faktatabellen skal peke på.

Rapportene peker på staging-tabellene. Råkopier av kildene havner først i back room, og når de først ligger der, er det fristende å spørre rett mot dem, fordi de allerede er lastet. De tabellene bærer hver kildes egne koder, duplikater og hull og endrer seg hver gang en kilde gjør det, så hver rapport som bygger på dem, arver akkurat det problemet datavarehuset skulle løse. Hold back room for ETL-systemet alene, og gi rapporter, dashbord og analytikere bare presentasjonsområdet, der de felles dimensjonene og den fastsatte granulariteten er det som får tallene til å stemme.

Les mer

- Kimball Technical DW/BI System Architecture — back room og front room, ETL-delsystemene, metadata og alternativene hub-and-spoke og hybrid.
- Enterprise Data Warehouse Bus Architecture — conformed dimensions, bussmatrisen og oppbygging én forretningsprosess om gangen.
- Four-Step Dimensional Design Process — forretningsprosess, granularitet, dimensjoner og fakta, i den rekkefølgen.
- Type 2: Add New Row — historikk for en egenskap som endrer seg, med en ny rad, en ny surrogatnøkkel og tre ekstra kolonner.
- BigQuery overview — Googles administrerte datavarehus, med lagring og regnekraft holdt adskilt og data lagret kolonnevis.

Ordliste

Velg ut fra deres faktiske rammer

Ta med et representativt datasett, spørsmålene det skal besvare og dagens flaskehals. Vi kan avtale målinger av kvalitet, gjennomstrømning og gjenoppretting.

Se hva det koster Ta kontakt →

Gjør begrepene praktiske

Spør hvordan «Hva er et datavarehus?» gjelder i organisasjonen deres. Et konkret eksempel fra dagens system er nok til å starte samtalen.

Se tjenestene våre

welcome@engineer.company +45 50 10 61 84 København, Danmark

Ta med deg denne siden

Visittkort

Laget med kjærlighet av D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

Opphavsrett © 2025 – nå · Engineer ApS · Alle rettigheter forbeholdt.

Skrevet ut fra https://engineer.company/nb/glossary/data-warehouse/