**Engineer**

# Data warehouse

en database, der samler virksomhedens data til analyse

## Hvad det er

Et data warehouse er en database, der er bygget til at besvare spørgsmål om en virksomhed, ikke til at drive den. De systemer, der tager imod ordrer, booker forsendelser, udsteder fakturaer og registrerer supportsager, har hver deres data i hver deres form. Et data warehouse får kopier fra dem alle, gemmer dem i årevis og ordner dem, så et spørgsmål, der går på tværs af flere systemer, bliver til én forespørgsel i stedet for en uges eksport til regneark. Ordet skrives også i ét, datawarehouse, og nogle siger datavarehus; arbejdet med at bygge og drive det kaldes data warehousing.

Forskellen fra de databaser, der kopieres fra, ligger i arbejdsbyrden. Et ordresystem læser og skriver få rækker ad gangen, tusindvis af gange i minuttet, og holder kun nutiden: flytter en kunde, overskrives den gamle adresse. Et data warehouse skrives i store portioner, læses af forespørgsler, der gennemløber millioner af rækker på én gang, og gemmer fortiden med vilje, så sidste års tal stadig kan brydes ned, sådan som forretningen så ud sidste år.

Ralph Kimball introducerede dimensionel modellering i branchen i 1996, og Kimball Group deler et sådant system i to rum. Back room rummer ETL-systemet, som de opdeler i 34 delsystemer under fire opgaver: at hente data ud af kilderne, at rense og afstemme dem, at levere dem til præsentationslaget og at styre hele processen. Front room rummer præsentationslaget, som er det, folk faktisk spørger i. Det er organiseret efter forretningsproces, holdt på det mest detaljerede niveau, hver proces registrerer, suppleret med aggregerede opsummeringer for hastighedens skyld og bygget op af fakta- og dimensionstabeller. Rundt om det står BI-applikationerne: ad hoc-forespørgsler, standardrapporter, dashboards, scorecards og analyseapplikationer. Kimball Group kalder front room for det ansigt, systemet viser udadtil.

Det er ikke den eneste opbygning. Kimball Groups egen arkitekturside beskriver Corporate Information Factory, et hub-and-spoke-design, der gemmer atomare data i et normaliseret virksomhedsdækkende data warehouse på tredje normalform og lader et andet ETL-trin fylde dimensionelle tabeller til rapportering, og en hybrid, hvor et eksisterende normaliseret data warehouse forsyner et dimensionelt præsentationslag. Fælles for dem alle er adskillelsen: de systemer, der driver forretningen, er ikke det sted, hvor den analyseres.

Et data warehouse er en rolle, ikke et produkt. Det kan være en PostgreSQL- eller Microsoft SQL Server-database på en server, teamet selv driver, eller en administreret tjeneste som Google BigQuery, som Google beskriver som en fuldt administreret dataplatform, der kombinerer et data warehouse i skyen med analyseværktøjer. BigQuery holder lager og regnekraft adskilt, så hver del skalerer for sig, gemmer data kolonnevis i stedet for rækkevis og tager imod forespørgsler i ANSI-standard-SQL.

Brug begrebet i en beslutning

Brug »Hvad er et data warehouse?« i en konkret beslutning. Vi kan knytte begrebet til det system, I bygger eller køber. Tal fra salg, drift og økonomi samlet ét sted, så to afdelinger ikke længere kommer til det samme møde med hver sit svar.

**Se vores ydelser** Kontakt os →

Undersøg antagelserne bag ordet

Brug forklaringen og det tilknyttede arbejde til at formulere spørgsmål til jeres projekt. Vi kan undersøge antagelserne sammen, før et værktøj vælges. Har gjort en geodatapipeline 50 gange hurtigere — 50x hurtigere geografiske datapipelines

**Har gjort en geodatapipeline 50 gange hurtigere** Kontakt os →

## Hvorfor det betyder noget

En virksomhed, der stiller det samme spørgsmål til to systemer, får to svar. Salg tæller en kunde, når ordren afgives, økonomi, når fakturaen er betalt, og support, hver gang der kommer en sag. Ingen af dem tager fejl til deres eget formål, og ingen af dem stemmer med de andre. Et data warehouse er stedet, hvor én definition bliver aftalt og skrevet ned i indlæsningen, så hver rapport, der bygger på den, tæller på samme måde.

Kimball Groups redskab til det er den fælles, afstemte dimension, på engelsk conformed dimension: én kunde, ét produkt, én kalender, vedligeholdt centralt i ETL-systemet og delt af hver faktatabel, der har brug for den. En faktatabel for salg og en for supportsager, der peger på den samme kundedimension, kan sammenlignes linje for linje. Deres busarkitektur stiller forretningsprocesserne op mod de dimensioner, de deler, i en matrix, som de kalder den arkitektoniske

tegning, og den gør det muligt at bygge et data warehouse én forretningsproces ad gangen, hvilket ifølge Kimball Group forkorter tiden, før noget kan tages i brug. Et team kan levere ordrer i dette kvartal og returvarer i næste uden at bygge om på det, der allerede er leveret, så længe begge bruger de samme dimensioner.

Historik er den anden grund. Driftssystemer holder den aktuelle tilstand; et data warehouse kan holde hver tilstand. Kimballs teknik til en egenskab, der ændrer sig, kaldet type 2, tilføjer en ny række i dimensionen med de nye værdier og en ny surrogatnøgle, og hver faktarække, der indlæses fra det øjeblik, peger på den nye række. Salg fra før kunden flyttede, bliver i den gamle region; salg bagefter havner i den nye. Mindst tre ekstra kolonner, en gyldig fra-dato, en gyldig til-dato og et flag for den aktuelle række, gør det muligt at spørge, hvad der var sandt på en given dato.

Den tredje grund er ren belastning. En rapport, der gennemløber tre års ordrer, hører ikke hjemme på den database, der tager imod dagens ordrer, hvor den konkurrerer med kassen om den samme hukommelse, de samme diske og de samme låse. Når analysen flytter til sin egen database, kan hver side indstilles til sit eget arbejde.

Prisen er, at et data warehouse er en kopi. Det er så friskt som den seneste indlæsning og så korrekt som reglerne i den, og nogen skal eje begge dele: kørselsplanen og de definitioner af et salg, en kunde og en måned, som forretningen er blevet enig om.

## Sådan ser det ud i praksis

Kernen i et dimensionelt data warehouse er en faktatabel, hvis granularitet, på engelsk grain, er fastlagt først, her én række pr. ordrelinje, omgivet af de dimensioner, der beskriver den. Kundedimensionen har kolonnerne til type 2, så en kunde, der flytter, får en ny række i stedet for en overskrevet.

```
CREATE TABLE dim_customer (
    customer_key bigint PRIMARY KEY,   -- surrogatnøgle, ejet af warehouset
    customer_id  text NOT NULL,        -- nøglen i kildesystemet
    region       text NOT NULL,
    valid_from   date NOT NULL,
    valid_to     date,                 -- tom, så længe rækken er aktuel
    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 blev indlæst med den kunderække, der var aktuel dengang, viser forespørgslen omsætningen fordelt på den region, kunden boede i, da købet skete. Joines der i stedet på `customer_id` med filter på `is_current`, bliver hele historikken gjort op efter nutidens regioner. Begge spørgsmål er legitime, og modellen svarer på dem begge uden en ekstra kopi af dataene.

Løftet i ydelsen Design af data warehouse er netop den første grund: tal fra salg, drift og økonomi samlet ét sted, så to afdelinger ikke længere kommer til det samme møde med hver sit svar. I større skala stod stifteren for arkitektur og drift af 100 data warehouse-databaser i PostgreSQL, Microsoft SQL Server og Google BigQuery fra 2022 til 2024, primært med GIS- og tidsseriedata som spatiale koordinater og sensoraflæsninger. Arbejdet kombinerede normaliserede og denormaliserede tabeldesign med spatial indeksering (PostGIS i PostgreSQL) og tidsseriepartitionering i de to første systemer og tidspartitionerede, clustrede tabeller med komprimering og kolonnebaseret lagring i BigQuery. Svartiderne for forespørgsler på GIS- og tidsseriedata faldt med 40–60 %, og infrastrukturen gjorde det muligt for organisationen, der brugte disse databaser, at lancere nye datadrevne produkter som dashboards med realtidsanalyser.

## Hvor det går galt

**En faktatabel blander to granulariteter, og hver total tæller nogle penge to gange.** En ordre har linjer, og den har også et fragtgebyr, der hører til ordren som helhed. Sættes gebyret på hver linje, tæller en sum over linjerne det én gang pr. linje. I Kimball Groups designproces i fire trin kommer fastlæggelsen af granulariteten som nummer to, lige efter valget af forretningsproces og før nogen dimension eller noget faktum navngives, og reglen er, at forskellige

granulariteter ikke må dele én faktatabel. Hold en separat faktatabel på ordreniveau til beløb, der hører til hele ordren, eller fordel gebyret på linjerne efter en regel, forretningen godkender.

**En kunde flytter, og sidste års rapport ændrer sig.** Når regionen på kunderækken bare overskrives, følger alle tidligere salg kunden til den nye region, og en rapport, der blev godkendt i januar, fortæller en anden historie i juni. Behandl egenskaber, hvis historik betyder noget, med type 2: luk den aktuelle række, og tilføj en ny med en ny surrogatnøgle, i én databasetransaktion, så ingen indlæsning nogensinde ser to aktuelle rækker.

```
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 teams bygger hver sin kundedimension, og diskussionen, et data warehouse skulle afslutte, starter forfra.** Et nyt fagområde er hurtigere at levere med sin egen kopi af kunde- eller produktlisten, og hver kopi glider væk fra de andre: den ene udelader lukkede konti, den anden slår dubletter sammen på en anden måde. Byg de fælles dimensioner én gang i ETL-systemet, og lad hver ny faktatabel bruge dem. Busmatricen er tjeklisten: før en forretningsproces kommer med, viser den, hvilke eksisterende dimensioner den nye faktatabel skal pege på.

**Rapporterne peger på staging-tabellerne.** Råkopier af kilderne lander først i back room, og når de først er der, er det fristende at spørge direkte i dem, fordi de allerede er indlæst. De tabeller bærer hver kildes egne koder, dubletter og huller og ændrer sig, hver gang en kilde gør, så hver rapport, der bygger på dem, arver netop det problem, et data warehouse skulle løse. Hold back room til ETL-systemet alene, og giv rapporter, dashboards og analytikere præsentationslaget, hvor de fælles dimensioner og den fastlagte granularitet er det, der får tallene til at stemme.

# Læs mere

- Kimball Technical DW/BI System Architecture — back room og front room, ETL-delsystemerne, metadata og alternativerne med hub-and-spoke og hybrid.
- Enterprise Data Warehouse Bus Architecture — conformed dimensions, busmatricen og opbygning én forretningsproces ad gangen.
- Four-Step Dimensional Design Process — forretningsproces, granularitet, dimensioner og fakta, i den rækkefølge.
- Type 2: Add New Row — historik for en egenskab, der ændrer sig, med en ny række, en ny surrogatnøgle og tre ekstra kolonner.
- BigQuery overview — Googles administrerede data warehouse med lager og regnekraft holdt adskilt og data gemt kolonnevis.

Ordliste

Vælg ud fra jeres faktiske vilkår

Medbring et repræsentativt datasæt, de spørgsmål det skal besvare, og den aktuelle flaskehals. Vi kan aftale målinger af kvalitet, gennemløb og gendannelse.

**Se, hvad det koster** Kontakt os →

## Gør begreberne praktiske

Spørg, hvordan »Hvad er et data warehouse?« gælder i jeres organisation. Et konkret eksempel fra jeres nuværende system er nok til at begynde samtalen.

**Se vores ydelser**

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

Tag denne side med dig

Visitkort

Lavet med kærlighed af D.

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

Copyright © 2025 – nu · Engineer ApS · Alle rettigheder forbeholdes.

Udskrevet fra https://engineer.company/da/glossary/data-warehouse/