

Handleiding: Een Datawarehouse Opzetten van A tot Z

1. Wat is een Datawarehouse?

Een datawarehouse is een gecentraliseerd systeem dat data uit verschillende bronnen samenbrengt, opslaat en optimaliseert voor analyse en rapportage. In tegenstelling tot operationele databases, die geoptimaliseerd zijn voor transactieverwerking (OLTP), is een datawarehouse ontworpen voor analytische query's (OLAP).

Het fungeert als één betrouwbare bron van waarheid ('single source of truth') voor organisaties die op basis van data strategische beslissingen willen nemen. Een goed opgezet datawarehouse bevat gestructureerde, tijdsgebonden en gevalideerde data, vaak in een genormaliseerd of dimensioneel model.

2. Waarom een Datawarehouse?

Zonder datawarehouse moet je telkens data ophalen uit losse bronnen (Excel, ERP, CRM), wat foutgevoelig en tijdrovend is. Een datawarehouse zorgt voor centralisatie en uniformiteit van je gegevens.

Het voordeel is dat je historische analyses kunt uitvoeren, consistente dashboards kunt bouwen, en meerdere teams dezelfde definities gebruiken (bijv. wat is 'omzet?'). Een datawarehouse biedt ook schaalbaarheid en prestaties: query's worden sneller omdat de structuur is geoptimaliseerd voor leesbewerkingen.

Daarnaast maakt het monitoring en auditing eenvoudiger, bijvoorbeeld met logging van wijzigingen of toegangsbeheer.

3. Cloud vs On-Premise

Cloud-oplossingen zoals Azure Synapse, AWS Redshift en Google BigQuery zijn populair vanwege hun flexibiliteit, schaalbaarheid, lage instapkosten en automatische updates. Voorbeelden zijn: een startup die begint in Google BigQuery en automatisch opschaaft als de data groeit.

On-premise oplossingen (zoals SQL Server, Oracle of Teradata) vereisen vaak hogere initiële investeringen, maar bieden meer controle en worden soms gekozen vanwege wet- of regelgeving (denk aan datalocatie of privacy-eisen).

Een hybride vorm is ook mogelijk: data wordt on-premise opgeslagen, maar verwerking of rapportage gebeurt in de cloud. Kies bewust op basis van budget, privacy en schaalbaarheid.

4. Typen Datawarehouses

Er bestaan verschillende architecturen voor datawarehouses, elk met specifieke toepassingen:

- **Relationeel datawarehouse**: Klassiek model, vaak in SQL Server of Oracle, met gestructureerde tabellen. - **Star schema**: Simpel en effectief voor BI-tools; je hebt één feitentabel met meerdere

dimensies. - **Snowflake schema**: Meer genormaliseerd dan een star schema, zorgt voor minder redundantie maar is complexer. - **Data Vault**: Robuuste en flexibele structuur, geschikt voor auditeerbare omgevingen. Je hebt hubs, links en satellites voor historisering. - **Lakehouse**: Combineert het beste van een datalake en datawarehouse. Voorbeeld: Delta Lake op Databricks met structured + semi-structured data.

5. Bronnen en Integratie

Begin met het in kaart brengen van je bronsystemen: denk aan ERP (SAP, Dynamics), CRM (Salesforce, Hubspot), platte bestanden (CSV, Excel) en externe API's. Bepaal per bron hoe vaak je updates wilt (real-time, dagelijks, wekelijks) en hoe je data op kwaliteit controleert.

Gebruik extractietools zoals Azure Data Factory, Fivetran, Stitch of zelfgeschreven Python-scripts om deze data op te halen en op te slaan in een staging-omgeving.

Daarna transformeer je de data naar een uniforme structuur. Logging en error handling zijn cruciaal: als een API faalt, wil je daar meteen van op de hoogte zijn.

6. Ontwerp van het Datamodel

Het datamodel is het fundament van je datawarehouse. Werk meestal met een star schema waarin je één centrale feitentabel hebt (zoals fact_sales) en daaromheen meerdere dimensies zoals dim_date, dim_klant, dim_product. Dit maakt rapporteren eenvoudiger en begrijpelijker voor gebruikers.

Zorg voor duidelijke primary en foreign keys. Maak ook ruimte voor auditvelden zoals created_at en updated_by. Gebruik surrogate keys in plaats van natuurlijke sleutels (zoals klantnummers).

Een goed datamodel zorgt dat BI-tools zoals Power BI of Tableau zonder extra logica data kunnen visualiseren.

7. ETL vs ELT

Bij ETL (Extract-Transform-Load) transformeer je de data voordat je het opslaat. Dit werd vroeger vooral gebruikt bij traditionele tools. Bij ELT (Extract-Load-Transform) laad je eerst ruwe data in je warehouse, en transformeer je die pas met SQL of dbt.

In moderne cloudomgevingen heeft ELT vaak de voorkeur, omdat opslag goedkoop is en compute on-demand beschikbaar is. Je kunt eenvoudig herhalen, rollbacken en sneller itereren in je data pipelines.

8. Voorbeeld in Python: API naar CSV

Met Python kun je een simpele ETL bouwen:

```
import requests
import pandas as pd
```

```
url = "https://api.example.com/sales" response = requests.get(url) data = pd.DataFrame(response.json())
data.to_csv("brondata.csv", index=False)
```

Deze CSV kan dan worden ingelezen in je staging-omgeving of direct in je lakehouse geplaatst.

9. Voorbeeld in SQL: Star Join

Een join tussen feitentabel en dimensies:

```
SELECT f.sales_amount, d.date, c.customer_name FROM fact_sales f JOIN dim_date d ON f.date_id =
d.date_id JOIN dim_customer c ON f.customer_id = c.customer_id
```

Zorg ervoor dat deze relaties ook visueel beschikbaar zijn in je BI-tool zodat filters goed werken.

10. Dataopslag en Scheiding

Scheid je lagen: RAW bevat ruwe data, STAGING bevat opgeschoonde data, en CURATED is de definitieve versie voor rapportages.

Gebruik Delta Lake, PostgreSQL, SQL Server of Parquet-bestanden. Structuur helpt je om fouten op te sporen en herbruikbare modellen op te bouwen. Gebruik ook metadata tagging, zodat je later snel weet wat de gevoeligheid, eigenaar en bron is.

11. Visualisatie en Rapportage

Koppel tools als Power BI, Looker of Tableau op de CURATED laag van je datawarehouse.

Zorg voor consistente KPI-definities, gebruikersrollen, en filters. Bouw dashboards die antwoord geven op echte businessvragen, niet alleen 'mooie plaatjes'. Gebruik bookmarks, drillthrough en tooltip-pages om gebruikerservaring te verbeteren.

12. Governance en Beveiliging

Governance is geen luxe maar noodzaak. Leg rollen vast (wie mag wat aanpassen), versleutel gevoelige data (zoals BSN of e-mail) en implementeer monitoring.

Gebruik bijvoorbeeld Azure Purview of Collibra voor metadata management. Versiebeheer (bijv. via Git) en CI/CD zijn essentieel voor het betrouwbaar uitrollen van veranderingen in ETL-pipelines.

13. Tools & Technologieën

- **Azure Data Factory**: visuele ETL-tool - **dbt (Data Build Tool)**: voor SQL-modellen en data lineage - **Power BI / Looker / Tableau**: visualisatie - **Databricks / Spark / Delta Lake**: schaalbare compute en opslag - **GitHub + CI/CD**: versiebeheer en automatische deployment - **SQL, Python, YAML, Jinja2**:

belangrijke talen in je stack

14. Conclusie

Een datawarehouse is de ruggengraat van datagedreven organisaties. Of je nu klein begint of enterprise werkt, de principes blijven hetzelfde: duidelijke architectuur, betrouwbare dataflows en goede documentatie.

Gebruik deze handleiding als uitgangspunt voor jouw project of klantcase. Succes!