Engineer

Datasæt

en samling data, der udgives som én enhed

Hvad det er

Et datasæt, på engelsk dataset, er en afgrænset mængde data, som man håndterer som én ting: det har et navn, en kilde, en periode, det dækker, og en form, som alle dets poster deler, og de fire forhold siger, hvor det begynder og slutter. En måneds måleraflæsninger i én fil er et datasæt; det samme er en natlig eksport af tre relaterede tabeller. Ordet siger intet om størrelse eller format, kun at dataene indeni hører sammen.

Siderne på dette site bruger ordet i den praktiske betydning. Én taler om et historisk datasæt i JSONL-format og om datasæt på 1 dag og 30 dage brugt til benchmarks, en anden om kuraterede datasæt og tabeller, der eksporteres for at blive indlæst andetsteds, en tredje om enorme datasæt fra fjernaflæste elmålere. Hver gang trækker ordet en grænse om data, før der gøres noget ved dem.

W3C's Data Catalog Vocabulary, DCAT, giver ordet en formel ramme. Version 3 blev W3C-anbefaling den 22. august 2024 og definerer et datasæt som en samling data, offentliggjort eller kurateret af én aktør og tilgængelig til adgang eller download i en eller flere serialiseringer eller formater. Begrebet er bredt med vilje: tal, tekst, billeder og lyd kan alle indgå. DCAT skiller derefter tre ting ad, som daglig tale blander sammen:

- **Dataset**: dataene som idé, for eksempel timeforbrug pr. zone for 2024.
- **Distribution**: én tilgængelig form af datasættet, for eksempel en CSV-fil til download; ét datasæt kan have flere.
- **Dataset series**: datasæt, der udgives hver for sig, men har nok til fælles til at blive samlet, for eksempel én fil pr. måned.

To filer i forskellige formater kan altså være ét datasæt, og to månedsfiler under samme navn kan være to medlemmer af en serie.

I Google BigQuery har ordet en anden, snævrere betydning. Der er et datasæt en beholder til tabeller, views og andre ressourcer, og det ligger inde i et projekt. En tabel skal høre til et datasæt, så datasættet oprettes, før der indlæses data, og det adresseres som `project.dataset` i GoogleSQL og som `project:dataset` i kommandolinjeværktøjet bq. Sammenhængen afgør, hvilken betydning der er i spil.

Brug begrebet i en beslutning

Brug »Hvad er et datasæt?« i en konkret beslutning. Vi kan knytte begrebet til det system, I bygger eller køber. Dashboards og rapporter, der viser de få tal, en beslutning faktisk afhænger af, opdateret uden at nogen beder om det.

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. Brandkommunikation med afsæt i publikums reaktioner — brandkommunikation med afsæt i publikums reaktioner

Brandkommunikation med afsæt i publikums reaktioner Kontakt os →

Hvorfor det betyder noget

En grænse gør data sammenlignelige. To sæt aflæsninger kan kun sammenlignes, når begge dækker samme periode, samme steder og samme afstand mellem posterne, og at navngive et datasæt er måden, de vilkår fastlægges på. DCAT giver afstanden sin egen egenskab, temporal resolution, den mindste afstand mellem elementer i datasættet, skrevet som en varighed som `P1D` for én dag. Et datasæt, der oplyser sin opløsning, fortæller læseren, om rækkerne er timer, kvarterer eller noget uregelmæssigt, før den første sum.

Beskrivelsen skal følge med dataene. DCAT opregner, hvad en distribution kan sige om sig selv: medietype, format, komprimeringsformat, pakkeformat, størrelse i byte og en checksum, og den sidste lader modtageren kontrollere, at indholdet er uændret. Specifikationens eget eksempel på et komprimeringsformat er gzip. En fil, der ankommer med de oplysninger, kan kontrolleres og åbnes; en fil, der ankommer nøgen, må man gætte sig til.

Formatet på en distribution afgør, hvem der kan læse den. Tre er almindelige:

- **CSV.** RFC 4180 fra 2005 beskriver det layout, de fleste implementeringer ser ud til at følge: én post pr. linje, en valgfri overskriftslinje, dobbelte anførselstegn om ethvert felt, der rummer et komma, et linjeskift eller et dobbelt anførselstegn, og et dobbelt anførselstegn inde i et felt skrevet to gange. Næsten alle værktøjer kan åbne CSV-filer.
- **JSON Lines.** Én gyldig JSON-værdi pr. linje, UTF-8, hvor linjerne adskilles med `\n`, gemt som `.jsonl` eller, komprimeret med gzip, `.jsonl.gz`. Et afsluttende linjeskift anbefales kraftigt, men er ikke påkrævet. Programmer læser filen én post ad gangen, og tekstværktøjer i Unix-stil og pipelines i en shell håndterer formatet godt.
- **Parquet.** Et open source-filformat, der er orienteret efter kolonner og bygget til effektiv lagring og hentning af data, med komprimerings- og kodningsmetoder til at håndtere komplekse data i store mængder.

Her er én post i de to første formater, hvor kommaet i noten tvinger anførselstegn frem i CSV-udgaven:

```
id,read_at,kwh,note
M-17,2024-01-08T10:00:00Z,3.21,"late, manual"

{"id":"M-17","read_at":"2024-01-08T10:00:00Z","kwh":3.21,"note":"late, manual"}
```

Sådan ser det ud i praksis

Eldata kom fra flere datakilder, som hver havde sin egen rapporteringsfrekvens, fra timeintervaller til intervaller på 15 minutter og i nogle tilfælde uregelmæssige tidsstempler, og det gjorde det svært at skabe et sammenhængende og sammenligneligt datasæt. Arbejdet tog rå hændelsesdata fra et historisk JSONL-datasæt og gjorde dem til timeforbrug med én række pr. time og pr. zone ved at summere produktionsmikset i hvert interval, lægge import til og trække eksport fra. Der blev skrevet tre varianter, i Python, JQ og SQL. Python-pipelinen parsede JSONL-filerne til dataframes, samplede tidsserierne om til timeintervaller og eksporterede resultaterne som CSV eller indlæste dem i en let SQLite-database, så de kunne efterses. I benchmarks med historiske datasæt på 1 dag og 30 dage var den den hurtigste af de tre: 34 ms for én dag og 180 ms for tredive. Arbejdet mellem input og output er et skift i granularitet, fra hver kildes egen takt til én fast række pr. time og zone.

Et datasæt bliver også flyttet, og et sæt på seks ETL/ELT-pipelines flyttede data fra et data warehouse bag en IPsec VPN til Google BigQuery. Dataene kom som komprimerede arkiver med Parquet-, CSV- og .bak-filer, og Bash- og Python-scripts klassificerede hver fil efter type og skema. Da procedurer havde renset, normaliseret og beriget rækkerne, blev kuraterede datasæt og tabeller eksporteret til mellemformater, komprimeret med gzip og overført til en Google Cloud Storage-bucket, hvorfra bq CLI og Python-scripts indlæste dem i BigQuery. For datasættene i BigQuery betød det friske data hver dag i stedet for hver uge, og hele turen fra arkiv til tabel tog under 20 minutter, hvor den før havde krævet over 3 timers manuelt arbejde.

Andre steder er datasættet noget, der skal læses. Fjernaflæste elmålere i flere geografiske regioner leverede enorme datasæt bag over 500 analyserapporter skrevet i ren SQL i PostgreSQL og BigQuery. Dataene omfattede geografiske koordinater, måler-id'er, tidsstemplet energiforbrug og miljømetadata. Hver rapport havde kommenteret SQL, så kolleger kunne reproducere og efterprøve den, og noter om fejlfinding dokumenterede almindelige datakvalitetsproblemer, for eksempel manglende GPS-koordinater eller korrupte målerværdier, med anbefalet håndtering. Siden tilskriver arbejdet en forbedring af effektiviteten i driftsplanlægningen på 10–15 %.

Hvor det går galt

Det meste besvær med datasæt er et faktum om dataene, der blev i ét menneskes hoved.

Granulariteten står ingen steder. Timerækker lagt sammen med kvartersrækker tæller nogle intervaller fire gange. Skriv granulariteten ned i én sætning, for eksempel én række pr. time pr. zone i UTC, og afstem hver kilde efter den, før der summeres.

Et hul læses som nul. Et tomt felt, en null og et 0 betyder tre forskellige ting, og en sum behandler stille en manglende aflæsning som intet forbrug. Manglende GPS-koordinater og korrupte målerværdier var almindelige nok i måleprojektet til at få dokumenteret håndtering. Afgør for hver kolonne, om et hul droppes, udfyldes eller markeres, og skriv valget i datasættets noter.

CSV-dialekten glider. RFC 4180 beskriver det almindelige layout, og virkelige filer afviger fra det: en anden separator, en manglende overskrift, et felt med komma, hvor anførselstegnene er glemt. RFC'en gør overskriftslinjen valgfri og

angiver dens tilstedeværelse med en MIME-parameter, som ikke følger med, når filen kopieres til disk. Kontrollér hver ny fils første linje mod den forventede overskrift før indlæsning.

En kopi glemmer, hvor den kom fra. Et kurateret udtræk, der er indlæst i et andet system, ligner originalen til forveksling en måned senere. Notér kilde, periode, udtrækstidspunkt og en checksum ved hver kopi, så en rapport kan sige, hvilket udtræk den læste.

Et BigQuery-datasæt havner det forkerte sted. Et datasæts placering fastlægges ved oprettelsen; at flytte det betyder at kopiere eller genskabe datasættet. Alle tabeller i én SQL-forespørgsel skal ligge i datasæt med samme placering, så vælg placeringen før den første indlæsning.

Læs mere

- Data Catalog Vocabulary (DCAT) - Version 3 — W3C's definitioner af datasæt, distribution og datasætserie med de egenskaber, der beskriver hver af dem.
- BigQuery: Introduction to datasets — ordets betydning som beholder: projekter, placeringer, navngivning og begrænsninger.
- JSON Lines — formatets tre regler og dets konventioner for filnavne.
- RFC 4180: Common Format and MIME Type for CSV Files — det CSV-layout, de fleste værktøjer følger, reglerne for anførselstegn og typen text/csv.
- Apache Parquet: Overview — hvad det kolonneorienterede filformat er, og hvor specifikationen ligger.

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 datasæt?« 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/dataset/