Refleksion over dokument databaser

Martin Sandgaard Rasmussen

Nu har jeg ikke skrevet så mange ”refleksioner”. Men jeg forestiller mig at det der bedes om, er et uddrag af en indre dialog.

Jeg er et menneske der tænker mange retninger på samme tid, og drager løbende konklusioner ud fra, hvor meget af en ”given verden” jeg kan se mening i.

Heldigvis har min lærer Kris, været tilgængelig til sparring det meste af dagen – og her har jeg fået en masse indsigt og eksempler i, hvordan jeg kan koble min viden- og skabe broer til forståelse på andre områder.

Min gruppekammerat Frederik har ligeledes været en glimrende sparringspartner, en værdifuld medspiller i dagens debat.

Martinus, gav nogle gode eksempler på, hvordan man kunne simplificere sin forståelse af dokument databaser.

Alt i alt, har det her været en rigtig fed øvelse. Og meget velkommen.

Derfor kommer der her nogle forskellige vinkler.

* Filosofisk vinkel
  + Perspektivering til videnskaben
* Teknisk vinkel
  + Perspektivering til fremtiden
  + Perspektivering til videnskaben
  + Anvendelse i praksis
  + Performance?
  + Konklusion
  + Meta evaluering
  + Fremtidsaspektet

# Filosofiske tanker

## Verdenen som en Samling af Data

Verdenen er utroligt kompleks og fyldt med forskellige typer af information, lige fra strukturerede data som kalenderdatoer og telefonnumre til ustrukturerede data som billeder, videoer og tekst. Denne mangfoldighed i data kan sammenlignes med de forskellige måder, databaser organiserer og håndterer information på.

## Relationelle Databaser som Lov

Relationelle databaser følger strenge regler og strukturer, hvor data er organiseret i tabeller, og relationer mellem disse tabeller er klart definerede. Denne tilgang minder om "lov" i den forstand, at alt har sin plads, og der er klare regler for, hvordan data interagerer. Det skaber orden og forudsigelighed, hvilket er essentielt for mange traditionelle forretningsapplikationer, hvor integritet og nøjagtighed af data er afgørende.

## Dokumentdatabaser som Kaos

Ikke kaos i traditionel forstand, men mere som natur-elementet, som Fyodor Dostojevskij beskriver det [[1]](#footnote-1)

På den anden side har vi dokumentdatabaser, som er mere fleksible med hensyn til datastruktur. I denne sammenhæng kan "kaos" ses som en kreativ kraft, der tillader en mere dynamisk og mindre foruddefineret tilgang til datalagring og -interaktion. Dokumentdatabaser tillader varierede dataformater inden for samme "samling" eller database og understøtter hurtig udvikling og iteration, da kravene til applikationens data kan ændre sig over tid.

## Harmoni Mellem Lov og Kaos

Selvom metaforen med lov og kaos kan synes at stille relationelle databaser og dokumentdatabaser som modsætninger, kan virkeligheden lige så vel være, at moderne dataløsninger indebærer en harmoni mellem de to. Mange virksomheder og applikationer bruger en hybrid tilgang, hvor de udnytter styrkerne ved både relationelle databaser og dokumentdatabaser for at opnå de mest effektive og skalerbare løsninger.

Tænk på irriterende relationer såsom interesser og tags. Det ville da være en brise, hvis man kunne slippe afsted med, blot at have et array liggende, til disse.

Denne balance mellem "lov" og "kaos" afspejler en forståelse af, at forskellige typer data og applikationsbehov kræver forskellige tilgange. Ved at vælge den rigtige database for opgaven kan udviklere og virksomheder sikre, at de effektivt kan håndtere den kompleksitet og mangfoldighed af data, som vores "verden" af information indebærer.

# Perspektivering til videnskaben og dennes cyklus

## Kan de filosofiske tanker tryktestes imod videnskabens udvikling?

Hvis man skulle beskrive udviklingen af videnskab, der drages paralleller mellem lov og kaos. Men andre ord, en overgang fra en mere ustruktureret til en struktureret tilgang til datahåndtering, som vi ser i dokumentdatabaser kontra relationelle databaser.

## Perspektiv

I videnskabens verden starter opdagelsesrejsen ofte med enkelte, ustrukturerede observationer – et utal af datapunkter indsamlet fra naturen uden umiddelbare sammenhænge eller foruddefinerede strukturer. Disse observationer repræsenterer den rå, ufiltrerede viden om vores verden, ligesom data lagret i en dokumentdatabase, hvor hvert dokument kan være unikt og ikke nødvendigvis følger en fast struktur.

## Hvor starter det

I tidernes morgen, kan vi jo se alle vor forfædres hjerner som dokument databaser. Men igennem tiden og som uddannelsen har vundet frem, har vi opnået en mulighed/evne for at tænke mere og mere relationelt, mht. vores viden og de paralleller vi drager i introspektion.

## Transformeringen fra dokumenter til relations-organiseret viden

Som videnskabsfolk dykker dybere ned i disse data, begynder de at opdage mønstre og sammenhænge mellem forskellige observationer. De begynder at stille disse observationer i relation til hinanden, identificere årsag og virkning, og formulere hypoteser. Denne proces minder om, hvordan data i relationelle databaser er organiseret, hvor relationer mellem tabeller (eller observationer) tillader en dybere forståelse af, hvordan forskellige elementer påvirker hinanden.

Efterhånden som hypoteser testes gennem eksperimenter, og resultaterne analyseres, bliver de oprindeligt kaotiske og ustrukturerede data transformeret til veldefinerede sæt af viden, der er i relation til andre sandheder. Denne nye, strukturerede viden bliver grundlaget for teorier og modeller, der kan forudsige nye fænomener og føre til yderligere opdagelser.

## Ender vi op med fuldstændig relations baseret viden?

Overgangen fra ustruktureret til struktureret viden i videnskaben afspejler den balance og harmoni mellem dokumentdatabaser og relationelle databaser i databehandling. Og fra observationer til fakta. Begge tilgange er værdifulde og nødvendige for deres respektive formål: Dokumentdatabaser for at indfange og håndtere den brede mangfoldighed af data i sine mange former, og relationelle databaser for at etablere dybe forbindelser og forståelser mellem data, som muliggør kompleks analyse og indsigt.

Så ja, man kan sige, at videnskabens udvikling fra ustrukturerede observationer til strukturerede teorier og modeller afspejler dynamikken mellem at håndtere kaotiske data og skabe orden gennem relationer, hvilket er fundamentalt for både datalogi og videnskabelig forskning.

Og nej, man kan tværtimod sige at det er en Ouroboros situation. Jo mere vi ved, jo flere spørgsmål har vi.

# Teknisk vinkel

Med de filosofiske tanker ude af vejen, kan vi nu dykke ned i de mere tekniske tanker. Jeg holder fokus på sagens kerne, altså dokument databaser, og dykker ikke dybere ned i relationelle databaser.

## Dokument databaser i deres rene form

Er designet til, ikke at holde fuldstændigt solide relationer til andre data, men mere til at enkapsulere ”det vi ved”. Ergo, ”observationerne”, der blev beskrevet i det videnskabelige perspektiv. Observationer er i evig forandring og sjældent hverken fyldestgørende eller ens.

Når man tænker på eksemplerne med videoer, blog indlæg og chat beskeder, er de i sig selv, glimrende til at beskrive den kaotiske, forandrende struktur, der ikke ”bare lige” kan tæmmes med relationer.

Dokumentdatabaser er derfor det værktøj vi har til rådighed, til at tæmme den evige datastrøm. Og udfra disse, kan vi så forsøge at se, om vi kan danne noget orden. Det kunne i hvert fald være én strategi.

En anden strategi, kunne være at forlige sig med, at vi aldrig får 100% orden i kaos. Der vil altid være ting der stikker ud – og er vigtige nok, til at huske eller gemme.

Med andre ord, er vi nu i stand til at forberede en transformation fra kaotiske data, til at bygge relationer imellem dem. Dette gøres muligt af dokumentdatabasernes mulighed for, at søge rundt i de forskellige dokumenter(observationer) vi har. Og lede efter sammenhænge og paralleller.

Vi i denne slags database, mulighed for at sammenkoble frie data, dynamisk, uden forcerede krav om sammenhænge. Lidt ligesom videnskabens vej fra observation, til hypotese, til teori og til fakta.

Når vi rammer ”fakta”, er vi langt nok i vores forståelse af relationerne mellem observationerne og det øvrige billede vi har defineret som ”sandt” eller ”virkelighedens fakta”. Og kan nu overveje at overføre data til en relationel database.

Menneskets kaotiske natur bliver aldrig hundrede procent faktuel. Så vi er nødt til at forholde os til, at det vil være utopi at tro, at vores forståelse af hele verdenen nogensinde vil komme til at passe ind i en relationel database. Vi er derfor nødt til at slutte fred med, at det bedste ”fit” i denne kontekst, er at indgå et kompromis.

Vi kan dele de aspekter af verdenen op der ligger fuldstændigt fast(indtil videre), såsom:

* Den periodiske tabel
* Samfundets love
* Pi
* Alfabetet

Og lægge dem i en relationel database, og bruge disse til at foretage ”sikre” operationer. Eller antagelser. Baser hvorfra vi kan studere og drage nye konklusioner.

Men ting som:

* Nyheder
* Bøgers indhold
* Og ja – du gættede det – dokumenter

Hører til i dokument databaser. Det gør de fordi, at deres indhold er forskelligt og ikke egner sig til dybdegående adskillelse og indeksering. Fordi der sjældent er klokkeklare mønstre i, hvordan de tager sig ud.

# Praktiske hybrid-eksempler

Hvis vi laver en mini-case, hvor hovedemnet er IoT devices, kan vi dele det data vi forventer, op i nogle kategorier, ud fra hvad vi ved.

* Vi ved hvor mange IoT devices vi har derhjemme.
* Vi ved at deres hardware ikke altid er klippe stabil.
* At de somregl kommunikerer over wifi, hvilket resulterer i latency og nogle gange, data der ikke når frem.

Hvis vi skulle danne os et overblik over disse enheder, og det de leverer af data, kan det se således ud:

Det der er relationelt

* Antallet af enheder
* Hvilken kategori af data de leverer
* Enheds modeller
* Netværks konfigurationer

Det der er kaotisk – og med andre ord, fint kunne passe ned i en dynamisk dokument database collection:

* Sensor data
* Enheds status / tilstand
* Hændelses logs
* Bruger-definerede tags

# Dokument databasers performance

Performance har ikke været påstået at være en key-tennet af dokument-databaser. Det syntes mere at være en ”best-effort” approach til at levere ”Det de andre vil have”. Forstået på den måde, at Dokument databaser leverer dybdegående forespørgsels metodik, og cacher de svar der gives.

Det betyder dog ikke, at dokument databaser er notorisk langsomme. De er blot anderledes, og har en **helt** anden workload og præmis, end traditionelle, relationelle databaser.

På denne måde - ved at analysere de forespørgsler der kommer ind - kan der uddrages antagelser om, at de specifikke mønstre der bedes om, er interessante for systemerne der benytter sig af dokument databaserne(jep, det sagde sig selv). Man kan derfor med rette antage, at der foretages en optimisering internt i dokument databasen, så den fremover vil kunne levere resultaterne af de observerede mønstre, hurtigere i fremtiden.

Dokument-datbabaser læner sig hæftigt op af caching-teknikker, der fungerer som en slags hukommelse, for hvad der blev svaret, sidst den fik ”samme spørgsmål”.

Af de performance metoder jeg har kunnet finde, står nogle stykker særligt ud

## Dokument Design

Opbygning af JSON-objekter: Selvom dokumentdatabaser er fleksible og kan håndtere ustrukturerede data, er det stadig vigtigt at opretholde en vis struktur inden for dine dokumenter. En velstruktureret JSON kan forbedre læsbarheden og effektiviteten af databasen.

### Organisering af Collections

Gruppering af relaterede dokumenter i samlinger (collections) kan hjælpe med at organisere dine data bedre. Dette skaber en mere intuitiv struktur og gør det lettere at forstå, hvilke typer data man kan forvente at finde.

## Begrænsning af store arrays

Selvom det er en fordel at kunne indeholde arrays direkte i dokumenterne, skal store arrays håndteres med forsigtighed. De kan nemt blive en flaskehals for performance, især hvis de skal scannes ofte. Det er derfor vigtigt at overveje størrelsen af disse arrays og potentielt opdele dem eller bruge andre optimeringsteknikker.

### Indeksering / Opbygning af Indekser

Indekser fungerer som en indholdsfortegnelse og gør det muligt at finde data hurtigere. Ved at indeksere de felter, der oftest forespørges, kan du markant reducere søgetiden og forbedre databasens ydeevne.

### Caching og brugen af den

Caching lagrer ofte forespurgte data i en hurtigt tilgængelig form. Dette betyder, at når de samme data forespørges igen, kan databasen levere svaret hurtigere, da det hentes fra cachen i stedet for at skulle beregnes eller hentes fra en langsommere datalager.

#### Eksempler på Caching i praksis

Teknologier som Cloudflare’s website caching, udnytter caching til at forbedre loadtider for websteder ved at gemme kopier af websteder på servere tættere på brugeren. Metoden de benytter til at levere det cache data, er opbygget med content-delivery network designet.

I en infrastrukturkontekst anvendes konceptet med "Hot/Cold" storage til at skelne mellem ofte anvendte data ("hot"), som opbevares på hurtigere medier som SSD'er, og sjældent anvendte data ("cold"), som opbevares mere økonomisk på langsommere medier såsom magnetiske diske eller endda bånd i tilfælde af Amazons S3 Glacier. Den parallel jeg taler om her er, at ”Hot storage” således en form for ”cache”, hvor ”cold storage” er vores dokument databases øvrige data(som i den grad ikke er så ”cold” alligevel).

# Konklusion

Det står tydeligt for mig nu, at idéen om, at alt data kan og bør være delt op, nydeligt og adskilt i relationelle databaser, er utopi.

Debatten i klassen i dag, samt min egen refleksion, har ført mig hen til et sted, hvor jeg nu står med en fod i ”hver verden”. Dokument databaser er ikke længere en mulighed. De er værktøjer der kan være ”best fit” i en lang række usecases, og ikke bør undervurderes.

Min forståelse af database opbygning og design som en helhed, har nu endelig fået en behagelig og blød side. Hvor der er plads til at gøre tingene lidt anderledes, og hvor det er okay, at nogle ting er lidt anderledes.

Jeg ser dokument databaser som et sindsygt stærkt add-on, til min forståelse af relationelle databaser. Og jeg glæder mig til at bygge systemer, hvor jeg gør brug af begge approaches.

Jeg har linet en usecase op til mig selv, hvor jeg i en dokument database, vil forsøge at importere det fantasy-univers jeg har bygget i json, og se hvordan det kan tage sig ud. Og jeg er mildest talt vildt begejstret for udsigten til at komme i gang med det.

Det muliggør en søgen i json objekter, som jeg hidtil kun har kunnet tilgå via. AI. Nu kommer jeg til at kunne gøre det med min kode også. Og DÉT er fedt!

### Meta evaluering

Jeg er meget taknemlig overfor al den sparring jeg fik i klassen. Det hører sig til sjældenhederne, at lærene er så tilgængelige for debat, som Kris var i dag. Jeg ser Kris som en ”ordensmand”, i ordets bedste betydning. Og at han sad og forklarede og vinklede forståelsen af dokument database så indgående, nysgerrigt og tålmodigt, accelererede min forståelse helt vildt.

Hvis selv Kris, kan se positive punkter i en kaos-database, så kan jeg også, blev mit udgangspunkt. I mine filosofiske reflekteringer, blev det hurtigt endnu tydeligere, hvor stor en rolle dokument databaser spiller i den tekniske verden jeg uddanner mig til at kunne finde vej i, og forhåbentligt, en skønne dag, kunne mestre en niche eller to, i.

## Fremtidsaspektet

Som jeg også beskrev i det filosofiske afsnit, mener jeg aldrig at den ene database type kommer til at erstatte den anden. Jeg mener der er behov for begge dele. Bankerne ønsker databaser med fakta. Og illustreret Videnskab ønsker abstraktioner af ”sandheden som vi kender den i dag”.

Dokumentdatabaser bliver helt sikkert stærkere med tiden. Og kommer højest sandsynligt til at danne ”backbone” for Object databaser, når vi en dag kommer så langt, at vi(menneskeheden) har fundet en ”bulletproof” opskrift på disse.

1. Fyodor Dostoejsvskij beskriver ”naturen” som kaos. Det er en kraft der er upålidelig, og ikke kan tæmmes. I hans perspektiv, ses mennesker som et glimrende eksempel på dette. Man kan se det(det iboende kaos i mennesket) som en kilde til kreativitet og innovation. Og også som en kilde til uforudsigelighed og evigt i forandring. [↑](#footnote-ref-1)