Figur 1 – Graf [1]

Datainnsetting i grafdatabasen Neo4j

Prosjektrapport 7.5.2018

OPPDRAGSGIVER

arkitektum

SAMMENDRAG

Arkitektum AS har under sin databehandling vurdert behovet for å benytte grafdatabaser i produksjon. De har av den grunn valgt å kjøre et bachelorprosjekt som skal resultere i en erfarings-rapport. Rapporten skal være en del av beslutningsgrunnlaget for eventuelt videre arbeid.

Bachelorprosjekt ved USN

KAN Solutions

Kristian Robertsen 151546
Atle Amundsen 916406
Nikolai Fosså 151703

Innholdsfortegnelse

Forord

Prosjektrapporten er dokumentasjon på vårt Bachelorprosjekt og vil beskrive arbeidsmetodikk, gjennomføring, måloppnåelse og erfaringer vi har gjort oss gjennom prosessen. Vi valgte dette oppdraget fordi arbeidserfaring for en etablert oppdragsgiver var viktig. En annen avgjørende faktor for valget er at vi kunne bygge videre på ervervet kunnskap. Samtidig ville vi kunne tilegne oss kunnskap om noe helt nytt. Vi har hatt stort utbytte av oppdraget og vi tar med oss erfaringene inn i arbeidslivet og eventuelle videre studier.

Vi er tre studenter i siste semester av Bachelorstudium i informatikk, som har utarbeidet erfaringsrapporten for Arkitektum AS. Studiet består av både teoretiske forelesninger og praktisk arbeid på lab. Gruppen, KAN Solutions, består av:

- Kristian Robertsen
- Atle Amundsen
- Nikolai Fosså

Arbeidet er gjennomført i samarbeid med Arkitektum AS som et bachelorprosjekt ved Universitetet i Sørøst-Norge. Prosjektarbeidet gjøres innenfor ett semester, og har et omfang på 15 studiepoeng. Bachelorprosjektet er siste og avsluttende del av studiet.

Beskrivelse av prosjektet

I vårt prosjekt skal vi tilegne oss ny kunnskap om grafdatabaser. I tillegg skal vi utarbeide en erfaringsrapport som dokumenterer innsetting av et datasett i valgt grafdatabase.

Erfaringsrapporten skal i detalj ta for seg prosessen med valg av grafdatabase og implementasjon av denne. Videre skal rapporten dokumentere arbeidet med å ta inn datasettet og eventuelt omforme det, slik dataene kan representeres i valgt grafdatabase. Rapporten skal spesielt belyse problemområder og løsningene på disse skal beskrives i detalj. Erfaringsrapporten er vel så viktig dokumentasjon av vårt arbeid som denne prosjektrapporten.

Arkitektum AS har i sin databehandling vurdert behovet for å benytte grafdatabaser i sin produksjon. Erfaringsrapporten skal være en del beslutningsgrunnlaget for eventuelt videre arbeid.

Etter forslag fra Arkitektum har vi valgt å ta jobbe med det åpne datasettet «Administrative enheter kommuner»

Prosjektweb
Her får du tilgang til prosjektets ressurser: https://robertsen.xyz/BachelorWeb/.

Kort om datasettet

Administrative enheter kommuner[2] viser kommuneinndelingen i Norge. Grensene som er registrert digitalt er er samlet i ett datasett. Kommunene avgrenses av følgende grensetyper: riksgrense, territorialgrense, avgrensningslinje i sjø, fylkesgrense og kommunegrense.

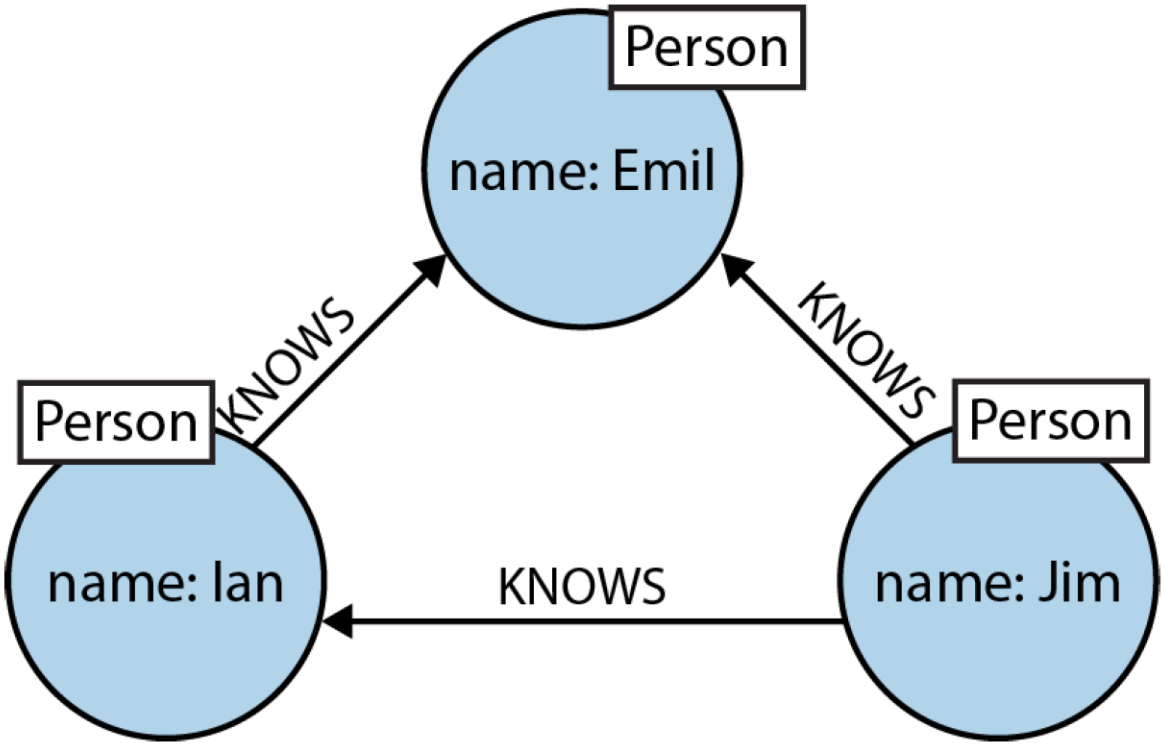
Kort om grafdatabaser

Tradisjonelle relasjonsdatabaser baserer seg på tabeller med rader og kolonner. Dette har etter flere tiår med utvikling gitt disse systemene flere styrker, men det er også en del svakheter. Grafdatabaser er et svar på en noe ironisk svakhet ved relasjonsdatabaser, nemlig relasjoner. Mange-til-mange

forhold i relasjonsdatabaser blir et problem for store datasamlinger, systemet blir tungrodd og ender opp med å ta svært mye plass [12].

Grafdatabaser tar tradisjonelt i bruk en såkalt "Property Graph" [13]. Denne implementasjonen består av tre komponenter.

- Noder
- Relasjoner, også kalt kanter (edges).
- Egenskaper (properties)



Figur 6 - Enkel graf, uttrykt ved hjelp av et diagram [3]

I diagrammet over er disse tre komponentene representert. En sirkel er en node, disse er alle gitt typen "Person", og en egenskap "name". En pil representerer en relasjon, en slik relasjon må ha en fra-node og en til-node, og er som nodene gitt en type ("KNOWS"). Relasjoner kan også ha egenskaper på lik linje med noder.

Egenskaper tilsvarer på sett og vis data i en relasjonsdatabase. I grafdatabaser er riktignok egenskaper lagret som "dokumenter" slik som i NoSQL databaser, og ikke som tabeller.

Å lagre disse egenskapene i form av tabeller er selvsagt mulig, dette gjøres for eksempel i en database kalt MongoDB.

Oppdragsgiver

Arkitektum[16] er lokalisert i Bø i Telemark og et firma som påtar seg oppdrag for både privat og offentlig sektor, og tilbyr følgende tjenester:

- Systemutvikling
- Systemarkitektur
- Informasjonsforvaltning
- Standardisering
- SOSI Produktspesifikasjoner
- Modelldrevet utvikling
- Prosjektledelse

Arkitektum har også en avdeling som jobber med webløsninger for andre bedrifter og organisasjoner. I mars 2018 består bedriften av 14 ansatte. Våre kontaktpersoner hos Arkitektum har vært daglig leder Hallstein Søvik og prosjektleder og systemutvikler Henning Jensen.

Mål

Dette gjengir målene definert i prosjektbeskrivelsen, eventuelle endringer blir belyst senere i rapporten.

Resultatmål

Etter endt prosjektarbeid skal det foreligge en erfaringsrapport som vurderer en grafdatabase sin egnethet for innsetting av data og eventuell kobling mellom flere datasett. Rapporten skal inneholde grundige beskrivelser av problemområder og løsninger på disse som oppdages underveis. Rapporten skal overleveres til oppdragsgiver 27.04.2018.

Effektmål

Beslutningsgrunnlag på om Arkitektum AS ønsker å ta i bruk grafdatabaser (den grafdatabasen vi har valgt å arbeide med i prosjektarbeidet).

Tids- og kostnadsbesparelser for Arkitektum AS.

Prosessmål

Erfaring med å arbeide i prosjekter ved bruk av smidige metoder; Kanban.

Erfaring med et prosjektstyringsverktøyet Jira. Programvaren gjøres tilgjengelig gjennom Arkitektum AS i prosjektperioden og støtter opp under valgt utviklingsmetode (Kanban).

Kompetanseoppbygging i en eller flere grafdatabaser.

Utviklingsmiljø

Verktøyene som har vært i bruk under arbeidet med erfaringsrapporten består av:

- draw.io
- Visual Paradigm
- IntelliJ IDEA
- Neo4j
- Google Docs
- Google Drive
- Git

Draw.io

draw.io[5] er en gratis web-applikasjon. Det er et utmerket verktøy for modellering som gjør det enkelt for brukeren å lage diagrammer og andre modeller.

Visual Paradigm

Visual Paradigm[6] er et modelleringsverktøy for Unified Modeling Language (UML). Det er et avansert og kraftig verktøy som egner seg spesielt godt til store prosjekter med høye krav til verktøyene, men også til våre formål.

Både Visual Paradigm og draw.io er brukt gjennom prosjektet til å modellere diagrammer brukt i erfaringsrapporten.

IntelliJ IDEA

IntelliJ IDEA[7] er en IDE (Integrated Development Environment) for utvikling av programvare. Det er tilgjengelig i to versjoner, en "Community Edition" som er lisensiert under Apache 2 lisensen, og en proprietær, kommersiell utgave kalt "Ultimate Edition".

Ultimate Edition tilbys gratis til studenter på årsbasis, og vi har benyttet oss av denne til utviklingen av Java-applikasjonen.

Neo4j v3.3.3 (Community Edition)

Neo4j[8] er en grafdatabase utviklet av Neo4j Inc. og har vært hovedkomponenten og fokus i arbeidet vårt.

Neo4j er utviklet kun med tanke på graflagring og behandling grafen som formål, i motsetning til mange andre grafdatabaser som for eksempel kan være en relasjonsdatabase med et "graf-lag" på toppen [4].

En typisk egenskap for relasjonsdatabaser er ACID-egenskapene og transaksjons-fokus. Neo4j Inc. har overført dette til sitt grafbaserte system for å få "det beste av to verdener".

Neo4j tilbyr en rekke metoder for å interagere med selve databasen, av disse har vi tatt i bruk:

Cypher Query Language (CQL, uttalt 'sequel'). Dette er et språk utviklet av Neo4j Inc. spesifikt for å jobbe mot Neo4j databaser. Det er "grafdatabasers SQL". CQL kan tas i bruk gjennom Neo4j sitt web-interface, eller gjennom "Cypher Shell".

Java Embedded Framework (JEF). Dette er et omfattende bibliotek til Java for å jobbe mot Neo4j grafdatabaser.

Begge metodene tilbyr brukeren kraftige verktøy for arbeid mot grafdatabasen. Vår erfaring er at JEF utmerker seg for innsetting av data, mens CQL sin styrke ligger i å hente ut data, da særlig i kombinasjon med Neo4j sitt webgrensesnitt.

Neo4j sin "Community Edition" er lisensiert under GPLv3 og er altså open source. Dette var en av de avgjørende faktorene for vårt valg av Neo4j som databaseteknologi.

Google Docs

Når gruppen har hatt behov for å jobbe på samme dokument, enten vi sitter samlet eller hver for oss, har vi brukt Google Docs[9]. Erfaringsrapporten har primært blitt utarbeidet i dette verktøyet.

Google Drive

Til å lagre og dele dokumenter og øvrige prosjektressurser som ikke har behov for versjonskontroll har vi gjennom prosjektet brukt Google Drive[10].

Git

Git har blitt brukt som verktøy for versjonskontroll under applikasjonsutvikling i prosjektet, vi har tidligere jobbet med dette verktøyet og finner det uvurderlig i utviklingssammenheng.

Vi har spesifikt tatt i bruk GitHub sin tjeneste for dette prosjektet, da de tilbyr gratis private [repos] til studenter.

Arbeidsfordeling og samarbeid

Inn i prosjektet har vi tatt med oss mange erfaringer fra forskjellige prosjekter/obligatoriske oppgaver tidligere i studiet. Vi har som gruppe jobbet tett sammen fra første semester. Som en naturlig følge av dette kjenner vi hverandres styrker, svakheter, hvordan den enkelte jobber, og ikke minst liker å jobbe.

I gruppen er vi tre personer hvor Atle ble tildelt rollen som prosjektleder og kontaktperson mot oppdragsgiver. Grunnet sykdom har Atle, i kortere perioder, vært forhindret fra å delta direkte i lab-arbeidet. Dette har også medført periodiske endringer i arbeidsfordelingen for prosjektet.

Alle har jobbet med tilegning av ny kunnskap og programmering. Kristian har hatt hovedansvaret for utviklingen av Java-applikasjonen.

Nikolai har hatt hovedansvaret for erfaringsrapporten, som er gruppens akkumulerte erfaringer gjennom prosjektet. Atle har også brukt erfaringsrapporten for å holde seg oppdatert kunnskapsmessig og på utviklingen i prosjektet i sykdomsperioder.

Atle har hatt hovedansvaret for utarbeidelsen av prosjektrapporten, hvor alle i gruppen har bidratt.

En fast arbeidsplass og faste arbeidstider på lab har medført økt effektivitet og, naturligvis, bedret direkte kommunikasjon blant deltakerne.

Felles arbeidstider på lab						
Mandag	**Tirsdag**	**Onsdag**	**Torsdag**	**Fredag**	**Lørdag**	**Søndag**
	0930-1500	1200-1500		0930-1500		

Figur 7 - Oversikt over felles arbeidstider på lab

Testing

Vårt oppdrag er å levere en erfaringsrapport på innsetting av et datasett med relasjoner/forhold i en grafdatabase, derfor har testing vært en kontinuerlig prosess med dokumentasjon av problemer og eventuelle løsninger som resultat. Vi har ikke tatt i bruk eksterne rammeverk o.l for testing i de utviklingsverktøyene vi har benyttet.

Arbeidsmetode

I beskrivelsen av prosessmål er valgt arbeidsmetode Kanban, men den ble ikke brukt i prosjektet. Avviket fra opprinnelig plan skyldes en rekke faktorer:

- Fast arbeidsplass, ingen behov for booking av rom ved HSN
  - Mindre behov for planlegging
- Faste felles arbeidstider gjennom uken
- Sammensveiset prosjektgruppe, jobbet sammen gjennom hele studiet
  - Kommuniserer godt
  - Kjenner hverandres styrker og svakheter
  - «Nesten automatisk oppgavefordeling»
- Lengre tids sykdom for prosjektleder

Fremdriftsplanen og gjennomføring

Arbeidet med prosjektet startet 9.1.2018 og avsluttes 8.5.2018. Fremdriftsplanen er delt inn i 6 faser, der hver fase er tildelt en tidsramme på et gitt antall uker. Figur 8 - Fremdriftsplan før justering, se Figur 10 - Milepælsplan før justering, for detaljer. viser en detaljert beskrivelse av hvilke faser planen består av og eventuelle avhengigheter, samt planlagt oppstart og avslutning for hver enkelt fase.

	Ukenummer													
Faser	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Prosjektbeskrivelse														
Datakonvertering og innsetting														
Koble datasett														
Ferdigstilling/overlevering erfaringsrapport														
Rapportlevering m/vedlegg														
Dokumentasjon (erfaringsrapport)														

Figur 8 - Fremdriftsplan før justering, se Figur 10 - Milepælsplan før justering, for detaljer.

Frem til og med fase 2 ble planen fulgt, men sykdom, statusmøte med oppdragsgiver og nye ønsker til rapporten medførte endringer i prosjektets fremdriftsplan. Følgende endringer ble gjort og disse vises i både Figur 9 - Fremdriftsplan etter justering, se Figur 11 - Milepælsplan etter justering, for detaljer. og Figur 11 - Milepælsplan etter justering.

- Mål som ble fjernet:
  - Mål om å koble nytt datasett til eksisterende (Administrative enheter kommuner)
  - Mål om oversikt og teoretisk sammenligning mellom flere grafdatabaser.
  - Prosjektleveransen til oppdragsgiver 27.04.2018
  - Mål om å ta i bruk en annen grafdatabase enn den vi har valgt (Neo4j)
- Mål som ble lagt til:
  - Benytte CQL i innsetting
  - Erfaringsrapporten skal sammenligne NoSQL og SQL databaser for innsetting av data.
  - Erfaringsrapporten skal ha en liten del om vedlikeholdbarhet av Neo4j databaser.
  - Prosjektleveranse til oppdragsgiver utsatt til 08.05.2018, etter avtale.

	Ukenummer													
Faser	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**
Prosjektbeskrivelse														
Datakonvertering														
Koble datasett														
Overlevering til oppdragsgiver														
Utarbeiding og overlevering prosjektrapport														
Dokumentasjon (erfaringsrapport)														

Figur 9 - Fremdriftsplan etter justering, se Figur 11 - Milepælsplan etter justering, for detaljer.

Milepælsplan(er)

#	Oppgave	Forventet fullført	Beskrivelse/aktiviteter
1	Prosjektbeskrivelse	06.02.2018	Kontraktsignering og presentasjon av prosjektbeskrivelse
2	Datakonvertering	06.03.2018	Tilegning av kunnskap om grafdatabaser
			Finne/lage teknologi for å konvertere datasett til ønsket format
			Koble innbyrdes data i datasettet (geografiske punkter)
			Implementere løsningen i grafdatabasen
3	Koble datasett	06.04.2018	Koble et annet datasett med datasettet Administrative Enheter
			Modellere/identifisere relasjoner mellom datasettene
			Implementere løsningen i grafdatabasen
4	Overlevering til oppdragsgiver	27.04.2018	Overlevering av prosjektresultat til Arkitektum AS.
5	Rapportlevering	08.05.2018	Overlevering av prosjektrapport med erfaringsrapport til HSN.
			Overlevering av prosjektrapport til Arkitektum AS dersom de ønsker den.
6	Erfaringsrapport	23.04.2018	Dokumentere:
			Utfordringer og løsninger ved alle aktiviteter under milepæl 2 og 3.
			Hva har vi gjort (kortfattet).
			Hvordan vi har gjort det.
			Hvorfor vi gjorde det slik.

Figur 10 - Milepælsplan før justering

#	Oppgave	Forventet fullført	Beskrivelse/aktiviteter
1	Prosjektbeskrivelse	06.02.2018	Kontraktsignering og presentasjon av prosjektbeskrivelse
2	Datakonvertering	06.03.2018	• Tilegning av kunnskap om grafdatabaser • Finne/lage teknologi for å konvertere datasett til ønsket format • Koble innbyrdes data i datasettet (geografiske punkter) • Implementere løsningen i grafdatabasen
3	Benytte CQL i innsetting	06.04.2018	Nytt forsøk med å sette inn data ved bruk av Cypher Query Language
4	Overlevering til oppdragsgiver	08.05.2018	Overlevering av prosjektrapport og erfaringsrapport til Arkitektum AS.
5	Utarbeiding og overlevering prosjektrapport	08.05.2018	• Utarbeidelse og ferdigstilling av prosjektrapport • Overlevering av prosjektrapport med erfaringsrapport til HSN
6	Erfaringsrapport	08.05.2018	Dokumentere: • Utfordringer og løsninger ved alle aktiviteter under milepæl 2 og 3. • Hva har vi gjort (kortfattet). • Hvordan vi har gjort det. • Hvorfor vi gjorde det slik. Tilleggspunkter: • Sammenligning avNoSQL og SQL databaser for innsetting av data. • Erfaringsrapporten skal ha en liten del om vedlikeholdbarhet av Neo4j databaser.

Figur 11 - Milepælsplan etter justering

#	Faser					Totalt
	1	6		4	5	
		2	3			
Timer	10	240	240	180	120	790

Figur 12 - Faseinndelt initielt tidsestimat

Fase 0 – Prosjektskisse/forprosjekt

Dette er ikke en fase i prosjektbeskrivelsen, men vi synes det er nyttig som dokumentasjon på forberedelsene forut for prosjektet.

Vi begynte med forundersøkelser og kommunikasjon med oppdragsgiver høsten 2017. Hallstein Søvik holdt en presentasjon ved HSN høsten 2017 og vi meldte vår interesse umiddelbart. Dette resulterte i møte med Hallstein Søvik og Tor Kjetil Nilsen 5. oktober 2017. På oppfordring fra Arkitektum og som et undersøkende forprosjekt reiste vi til Hønefoss og deltok på Hack4no 27. og 28. oktober 2017, sammen med Benjamin Dehli (Systemutvikler/interaksjonsdesigner) og Tor Kjetil Nilsen (Utviklingsleder og eier av Arkitektum AS).

Vår deltakelse på Hack4no lot oss diskutere bruk og nytteverdi av datasett med dataeierne, samt knytte kontakter. Vi lærte oss også hva som kreves av forarbeid og forkunnskaper for å få et best mulig utbytte av å delta.

Basert på møtet med oppdragsgiver og erfaringene fra hack4no utarbeidet vi og leverte prosjektskissen som obligatorisk oppgave i faget Prosjektstyring i november 2017.

Vi avholdt et nytt møte med Hallstein Søvik og Henning Jensen 18. januar 2018. Her gikk vi gjennom prosjektskissen og la rammene for prosjektet. Resultatet av møtet, prosjektskisse og forprosjektet ga oss et godt grunnlag for utarbeidelsen av prosjektbeskrivelsen.

Fase 1 – Prosjektbeskrivelse

Plan for fasen

Planlegge og detaljere prosjektets mål og rammer i en prosjektbeskrivelse. Det er satt av 2 uker til å gjennomføre arbeidet (uke 6 og 7) med innlevering/publisering av arbeidet 13. februar 2018.

Med høstens møter/samtaler med Arkitektum (heretter kalt oppdragsgiver), erfaringene fra Hack4no og våre tanker rundt prosjektet hadde vi grunnlaget for å gå i gang med en detaljert planlegging av prosjektet og sementere gjennomføringen av prosjektet.

I denne fasen skal vi også utforme og signere samarbeidsavtale med oppdragsgiver.

Gjennomføring

Vi gikk i gang med å definere oppgaven som skal løses i en klar problembeskrivelse, deretter detaljerte vi resultatmål, effektmål og prosessmål for prosjektet. Basert på målene som skulle nås beskrev vi prosjektets omfang gjennom fremdriftsplan, milepælsplan, felles arbeidstider, tidsbudsjett og rammebetingelser (herunder utviklingsmiljø). Eventuelle risikoer og tiltak er ble også belyst, samt opplysninger om rollene til deltakerne og interessentene i prosjektet og deres roller og ansvarsområder.

Planleggingen av prosjektet som resulterte i prosjektbeskrivelsen[22] var svært viktig for gjennomføringen av prosjektet. Prosjektbeskrivelsen skulle fungere som styringsdokument og det var derfor viktig at vi hadde tydelige og klare mål, en realistisk fremdriftsplan og milepælsplan, risikoanalyse og rammebetingelser, som er forstått av alle i prosjektgruppen.

Utarbeidelsen av prosjektbeskrivelsen er et resultat av erfaringene fra Hack4no, våre tanker om prosjektet og møter/samtaler med oppdragsgiver.

Prosjektbeskrivelsen har hjulpet oss å holde frister, men også hjulpet oss å gjennomføre endringer/justeringer når det har vist seg nødvendig.

Samarbeidsavtaler hadde ingen av oss erfaring med fra før, vi tok derfor kontakt med Runar Gundersen som er rådgiver og foreleser i emnene Prosjektstyring og Immateriell Rett ved USN, Campus Bø. Han ga oss svar og veiledning i utarbeidelsen av avtalen, og samarbeidsavtalen[17] med oppdragsgiver ble signert 5. februar 2018.

Fase 2 - Datakonvertering

Plan for fasen

Grafdatabaser er emne vi i utgangspunktet ikke har kunnskap om, men vi kjenner til grafer fra faget «Algoritmer og Datastrukturer». Uten å opparbeide oss kunnskap om grafdatabaser generelt og Neo4j spesielt vil vi ikke kunne gå videre med prosjektarbeidet.

Basert på tilegnet kunnskap om grafdatabaser skal vi så gå i gang med å konvertere/legge inn datasettet vi har blitt enige med oppdragsgiver om å bruke, «Administrative enheter kommuner», heretter kalt «datasettet», inn i grafdatabasen neo4j.

Deretter, basert på datasettets oppbygning, koble innbyrdes data i datasettet (Geografiske punker, angitt som koordinater).

Til slutt skal vi implementere løsningen i databasen.

Gjennomføring

Etter innledende fase med utarbeidelse av prosjektbeskrivelsen gikk vi i gang med å tilegne oss nødvendig kunnskap gjennom boken «Graph Databases» [18]. Tidligere tilegnet kunnskap om relasjonsdatabaser og boken «Graph Databases» kombinert med læremidler i Neo4j i form av eksempeldatabaser og grunnleggende eksempelspørringer ga oss en god plattform for tilegne oss ønsket kunnskap. Gjennom lesing, diskusjoner internt i prosjektgruppen og testing i Neo4js webgrensesnitt tilegnet vi oss den kunnskapen vi hadde trengte for å gå videre med neste aktivitet i fase 2.

Allerede under aktiviteten med å tilegne oss kunnskap gikk vi i gang med å se på datasettet og hvilke formater det var tilgjengelig i, vi valgte pga. tidligere kjennskap til formatet JSON[14], å gå for GeoJSON[1]. «Datasettet» kan lastes ned helt, eller stykkevis og delt; hele Norge, flere fylker, enkeltfylker, flere kommuner eller enkeltkommuner.

Vi spesifiserte krav til behandling av datasettet på egenhånd ved å modellere en ideell graf med de noder og forhold vi ønsket å sette inn i grafdatabasen.

Vi valgte å begynne stort og se på datasettet i sin helhet og som omfatter hele Norge. Det er enorme mengder data, og vi innså raskt at det ville være formålstjenlig å skalere ned størrelsen på datasettet vi skulle jobbe med. Vi gikk derfor ned på fylkesnivå; Telemark fylke.

For å kunne lese inn datasettet for Telemark fylke i grafdatabasen tok vi i bruk CQL og APOC[2][15], etter litt prøving og feiling virket det som om datainnsetting ved bruk av CQL kombinert med APOC ikke helt egner seg. Vi har behov for mer finmasket kontroll ved innlesing av datasettet enn det CQL/APOC tilbyr, i tillegg var datasettets størrelse og lang prosesseringstid en stor faktor ved innsetting når vi brukte CQL. Vi begynner derfor å se på muligheten for å ta i bruk et Java API[3] som Neo4j tilbyr.

[1] GeoJSON er et åpent standardformat laget for å representere enkle geografiske egenskaper, utvidelse av JSON.
[2] APOC er ressursbibliotek for Neo4j, som tilbyr brukerdefinerte prosedyrer.
[3] Application Programming Interface.

Neo4j sitt Java API kalles «Embedded Database»[19] og kan integreres i IntelliJ IDEA som vi bruker til programmere Java-applikasjoner, i tillegg bruker vi et JSON API, «org.json[4]»[21] for å kunne tolke GeoJSON. Applikasjonen vi utviklet, Neo4Java[20], ga oss den kontrollen vi ønsket under innlesing av datasettet som skal settes inn i grafdatabasen. Når vi har tilgang på data- og kontrollstrukturene som finnes i Java, reduseres tidsordenen for innsetting på bekostning av økt minnebruk. Med andre ord så flyttes logikken som Neo4j utfører ved oppretting av noder og relasjoner til i Java-applikasjonen.

Vi begynte å jobbe med Telemark fylke og økte skopet til å omfatte hele datasettet.

Relasjoner mellom koordinatnodene er dobbelt opp, men peker i hver sin retning. Dette skal i teorien ikke kunne skje, men vi har ikke funnet noen løsning på dette. Årsaken til at de peker i hver sin retning avhenger av rekkefølgen på koordinatene i datasettet ved gjennomløpning.

Innsetting av koordinater i grafdatabasen forløp uten problemer; ingen duplikater, ingen løse noder.

Relasjoner mellom fylkesnoder og deres grensenoder dupliseres til tider. Utfordringen med disse duplikatene kan løses på samme som for relasjoner for øvrig; å ta i bruk enda en datastruktur. Dette også medføre økt minnebruk. Vi valgte å ikke løse dette i applikasjonen da det går ut over rammene for prosjektet, samtidig som vi må innse at tiden ikke strekker til.

Mot slutten av fasen skrev vi om koden og tok i bruk Object Relational Mapping (ORM). Dette ble i utgangspunktet bare brukt på koordinater, men ble utvidet til å gjelde samtlige objekter som skulle settes inn i grafdatabasen. Dette har heller ikke løst utfordring med at relasjoner dupliseres.

Erfaringsrapporten detaljerer hvordan man kan kjøre prosjektet på egen maskin i delkapittel «5.4: Kjøre prosjekt på egen maskin». Du må også laste ned et IntelliJ-prosjekt herfra: https://github.com/gitsieg/Neo4Java

Datasettet er korrekt satt inn i grafdatabasen, selv om applikasjonen dupliserer relasjoner. Vi anser derfor milepæl 2 som nådd.

Fase 3 – Benytte CQL i innsetting

Plan for fasen

Selv om milepæl for innsetting av data er nådd, ønsker vi å gjøre et nytt forsøk på innsetting ved hjelp av CQL. Erfaringsrapporten skal omhandle innsetting av et datasett i en grafdatabase. I fase 2 demonstrerte vi en teknikk for å gjøre dette, og vi anser det som interessant for oppdragsgiver å bli vist en alternativ fremgangsmåte for innsetting av data. Vi har på bakgrunn av dette gått bort i fra mål om lenking av et annet datasett, og valgt å gjøre innsetting av data i CQL. Vi vil forsøke dette på tross av at en slik operasjon er svært tung og at prosesseringstiden vil gå opp (se vedlagt erfaringsrapport «Datainnsetting i Neo4j» 3.4.5 Innsetting av JSON-data for alle kommuner i et fylke).

Gjennomføring

Kontinuerlig og systematisk testing av spørringer og dokumentasjon av problemer bar frukter. Med større kunnskap opparbeidet gjennom prosjektet ble datasettet nå korrekt satt inn med de relasjoner/forhold vi ønsket. Vi har nådd milepæl 3.

[4] Org.json har klasser og metoder som tillater interaksjon med JSON-objekter og tabeller i datasettet.

Fase 4 – Overlevering til oppdragsgiver

Plan

Ferdigstille erfaringsrapport og overlevere ferdig produkt til oppdragsgiver 8. mai 2018.

Gjennomføring

Erfaringsrapporten gjennomgås i sin helhet, med fokus på økt lesbarhet, korrektur og gode eksempler på kode, forklarende figurer og diagrammer.

Fase 5 – Utarbeiding og overlevering prosjektrapport

Plan

Over 4 uker skal vi utarbeide og ferdigstille prosjektrapporten og overlevere prosjektrapport med vedlegg til HSN. Prosjektrapporten overleveres også til oppdragsgiver, etter ønske.

Gjennomføring

Arbeidet med prosjektrapporten skal dokumentere hvordan vi har jobbet med prosjektet, og som prosjektgruppe. Grunnlaget for rapporten er prosjektbeskrivelsen og erfaringsrapporten, hvor prosjektbeskrivelsen angir ønsket løp og notater/erfaringsrapport er med på å forklare endringer i prosessen. Alle har bidratt i utarbeidelsen av prosjektrapporten som er hovedgrunnlaget for evalueringen av prosjektet. Arbeidet er gitt høy prioritet.

Milepæl 5 anses som nådd ved overlevering.

Fase 6 – Dokumentasjon (erfaringsrapport)

Plan

Erfaringsrapporten skal dokumentere alle utfordringer og løsninger ved alle aktiviteter under fase 2 og 3 i fremdriftsplanen.

Etter ønske fra oppdragsgiver, fremsatt 11. april 2018, skal erfaringsrapporten sammenligne NoSQL- og SQL- databaser med tanke på innsetting av datasettet. I tillegg ønsket oppdragsgiver at erfaringsrapporten ha en egen del om vedlikehold av Neo4j databaser.

Krav til rapporten, utover overnevnte:

- Høy lesbarhet
- Gode eksempler på kode, figurer og diagrammer

Gjennomføring

Til å begynne med brukte vi egne dagbøker som dokumentasjon på arbeidet, men vi så snart at det ville være enklere å samle våre kollektive erfaringer i et og samme dokument; erfaringsrapporten. Erfaringsrapporten ble således til gjennom prosjektet, med bidrag fra alle i gruppen.

Dokumentasjon på forskjeller mellom NoSQL- og SQL-databaser ved innsetting av datasettet blir utarbeidet og illustrert og lagt til i erfaringsrapporten samt en egen del om vedlikehold av Neo4j databaser. Andre krav og valg for behandling av datasettet er beskrevet i detalj i vedlagt erfaringsrapport.

Alle aktiviteter er fullført i henhold til planen, milepæl 6 anses derfor som nådd.

Konklusjoner og erfaringer

Gjennom arbeidet med prosjektet har vi jobbet som en autonom gruppe og i praksis uten veiledning fra oppdragsgiver eller veileder, utover møter nevnt i rapporten. Vi har planlagt prosjektet, etablert vårt eget utviklingsmiljø, fulgt planen satt i prosjektbeskrivelsen og foretatt de endringer vi så som nødvendige.

Prosjektet er gjennomført i henhold til initiell fremdriftsplan til og med fase 2. Grunnet sykdom og statusmøte med oppdragsgiver ble fase 3 sine opprinnelige mål fjernet og nye innført. Følgende faktorer gjorde fortsatt gjennomføring mulig:

- Prosjektbeskrivelsen er helt klar på at hvordan vi skal behandle sykdomstilfeller
- Godt sammensveiset gruppe med god evne til omstilling

Foreløpig erfaringsrapport ble presentert for oppdragsgiver 3.april 2018 og foreløpig erfaringsrapport ble overlevert 6. april 2018. Vi fikk gode tilbakemeldinger på rapporten og tok med oss oppdragsgivers ønsker inn i det videre arbeidet med erfaringsrapporten.

Måloppnåelse

Resultatmål

Vi mener at erfaringsrapporten oppfyller resultatmålene, samt de ønsker fremsatt av oppdragsgiver, på en god måte. Vi gjort mer enn å bare sette inn data fra datasettet i grafdatabasen, vi har også opprettet forhold/relasjoner mellom objektene, dette er utover rammene gitt i prosjektbeskrivelsen.

Effektmål

Når det gjelder effektmål av erfaringsrapporten kan vi med sikkerhet si at oppdragsgiver har fått tids- og kostnadsbesparelser som følge av vårt arbeid. Om oppdragsgiver ønsker å ta bruk erfaringsrapporten som beslutningsgrunnlag på om de ønsker å ta i bruk grafdatabaser i sin produksjon, gjenstår å se.

Prosessmål

Kanban ble ikke tatt i bruk som arbeidsmetode i prosjektet fordi vi ikke så behovet for dette i vårt prosjekt.

Vi ser i ettertid at dette ville hjulpet oss med oppgaveidentifisering og gjennomføring da sykdomsfravær ble en faktor. For en gruppe som ikke er så samkjørt vil valg av arbeidsmetodikk og oppfølging av den være avgjørende.

Vi hadde også tilgang til Jira, men valgte å ikke benytte oss av dette verktøyet. For vårt prosjekt tok vi heller i bruk Google Docs (erfaringsrapport) og Google Drive (øvrige prosjektressurser) som verktøy. Dette var et viktig og riktig valg som har hjulpet oss godt gjennom prosjektet.

Å sette oss inn ny databaseteknologi og nytt spørrespråk har gitt oss godt faglig utbytte, selv om datastrukturen ikke er ukjent for oss. Forkunnskap om datastrukturen som Neo4j bruker fikk vi gjennom faget Algoritmer og Datastrukturer. Dette var en klar fordel og hjalp oss betraktelig i å tilegne oss den kunnskapen som var nødvendig for å gå videre i prosjektarbeidet.

Samarbeid

Vår styrke gjennom prosjektet var, og er, at vi har jobbet tett sammen gjennom hele studieløpet og det har vært avgjørende for gjennomføringen. Samarbeidet har, på tross av lang fartstid, lidd liten eller ingen slitasje. Vi har derfor vært i stand til å takle de utfordringene vi har møtt på en løsningsorientert måte og samtidig vært i stand til å endre arbeidsfordeling og mål som følge av

sykdom og ønsker fra oppdragsgiver. Gjennom de 4 siste fasene, uke 11 – 19, har vi nok overskredet tidsbudsjettet en del som følge av tidligere sykdom i prosjektperioden.

Avslutningsvis vil vi presisere at denne rapporten dokumenterer prosessen og erfaringsrapporten dokumenterer resultatet av arbeidet. Det er viktig å se disse to dokumentene i sammenheng.

Kilder og referanser

[1] By Martin Grandjean [CC BY-SA 3.0 (https://creativecommons.org/licenses/by-sa/3.0)], via Wikimedia Commons

[2] https://kartkatalog.geonorge.no/metadata/kartverket/administrative-enheter-kommuner/041f1e6e-bdbc-4091-b48f-8a5990f3cc5b

[3] Ian Robinson, Jim Webber & Emil Eifrem, «Chapter 3, Data Modeling with Graphs», side 28: «Figure 3-1» i «Graph Databases» 2nd Edition, O'Reilly Media, California, USA, O'Reilly 2015.

[4] Ian Robinson, Jim Webber & Emil Eifrem, «Chapter 1, Introduction», side 5: «The underlying storage» i «Graph Databases» 2nd Edition, O'Reilly Media, California, USA, O'Reilly 2015

[5] https://www.draw.io/

[6] https://www.visual-paradigm.com

[7] https://www.jetbrains.com/idea/

[8] https://neo4j.com/

[9] https://www.google.com/docs/about/

[10] https://www.google.com/drive/

[11] https://git-scm.com/about/

[12] Ian Robinson, Jim Webber & Emil Eifrem, «Chapter 2, Options for Storing Connected Data», side 11: «Relational Databases Lack Relationships» i «Graph Databases» 2nd Edition, O'Reilly Media, California, USA, O'Reilly 2015

[13] Ian Robinson, Jim Webber & Emil Eifrem, «Chapter 3, Data Modeling with Graphs», side 26: «The Labeled Property Graph Model» i «Graph Databases» 2nd Edition, O'Reilly Media, California, USA, O'Reilly 2015

[14] https://www.json.org/

[15] https://neo4j.com/blog/intro-user-defined-procedures-apoc/

[16] https://www.arkitektum.no/

[17] https://robertsen.xyz/static/docs/Samarbeidsavtale.pdf

[18] https://neo4j.com/lp/book-graph-databases/

[19] https://neo4j.com/docs/java-reference/current/tutorials-java-embedded/

[20] https://github.com/gitsieg/Neo4Java

[21] https://stleary.github.io/JSON-java/

[22] https://robertsen.xyz/BachelorWeb/beskrivelse/