

DAT107 Databaser – våren 2025

Obligatorisk øvelse 4: NoSQL

Lasse Jenssen, 11.04.2025

Rammer

Oppgaven løses i grupper på inntil 4 deltagere. Du kan også levere alene.

NB! Oppgaven skal leveres via en av gruppene i gruppesettet 'Oblig 4 Gr'. Dette gjelder også om du leverer alene (Finn et ledig gruppenummer og legg dere inn selv).

Innleveringer utenfor dette gruppesettet risikerer å ikke bli rettet!!

Vurdering og godkjenning

Skriftlig innlevering i Canvas innen **onsdag 30. april kl.23:59**.

Leveranse i Canvas skal være:

- 1) **Et PDF-dokument** med navn «dat107-oblig4.pdf»:
 - a) NB! I starten av PDF-dokumentet skal det stå
 - gruppenummer
 - navn og studentnummer til alle som er med på innleveringen
 - Navn på database i PostgreSQL (der tabeller i oppgave 1 og 2 er opprettet).(NB! Dere trenger bare å utføre oppgavene i en av databasene for gruppen)
 - b) Hver besvarelse/deloppgave skal ha overskrift med «Oppgave <nr> <bokstav>»
 - c) Skjermdump(er) av løst oppgave (der oppgaven spesifiserer dette)
 - d) Tekst. F.eks. SQL-er, JSON eller lignende (der oppgaven spesifiserer dette)
- 2) **En zip-fil** som inneholder fullført Eclipse-prosjekt, inkl. data filer fra oppgave 3c.

Bakgrunn

I denne oppgaven skal dere ta utgangspunkt i Hobbyhuset databasen som dere (kanskje) har benyttet i relasjonsdatabase kurset (se vedlagt fil «hobbyhuset.sql»).

Ledelsen i Hobbyhuset AS har besluttet at de ønsker å teste ut andre datamodeller, eventuelt andre database systemer, for å lagre sine data. De har derfor besluttet å gjennomføre et lite prosjekt for å teste ut ulike datamodeller og database systemer.

Dette prosjektet består av to separate deler. I den ene delen ønsker de å teste ut en hybrid løsning i PostgreSQL, der de migrerer noe av data til XML (altså til XML kolonne(r)), samt lager noen SQL-er for å hente ut data igjen. Denne delen finner dere i oppgave 2 under.

I den andre delen ønsker bedriften å teste ut en løsning med en MongoDB (NoSQL) database, der de flytter noe av data inn i en MongoDB database, samt å utføre noen enkle CRUD operasjoner i Java. Denne delen finner dere i Oppgave 3 og 4 under.

Databaser

Database	Beskrivelse
PostgreSQL	Databasen som brukes i prosjektet for oppgave 1 og 2 må være på databasetjeneren vår "ider-database.westeurope.cloudapp.azure.com:5432" slik at innlevert prosjekt kan verifiseres fra hvilken som helst PC (f.eks. læreren sin PC ved retting). Ved innlevering må databasen være opprettet og inneholde de nødvendige data. Eventuelle Java-prosjektet må inneholde riktig oppkoblingsinfo, dvs. datavasenavn, brukernavn og passord. NB! Husk å gjør COMMIT, sånn at data blir bekreftet lagret i databasen.
MongoDB	Databasen som brukes i prosjektet for oppgave 3 kan kjøres mot en valgfri MongoDB database (versjon 7 eller nyere. Sjekk med «db.version()».) Når der utfører oppgaven(e) så velger dere hvem sin MongoDB database dere vil benytte.

Oppgave 1 – Opprettelse av Hobbyhuset

- a) **Note! Svar på denne oppgaven ved å utføre oppgaven i gruppens PostgreSQL database.**
Lag et nytt skjema «oblig4» i din PostgreSQL database, og opprett tabellene (med data) for Hobbyhuset (se «hobbyhuset.sql»).

(Oppgaven fortsetter på neste side)

Oppgave 2 – XML og XPATH

Kunde

a) **NB! Svar på denne oppgaven i PDF-dokumentet.**

Skriv en SQL som oppretter en tabell «KUNDE_NY» med to kolonner: «knr» (datatype: integer) og «kunde_xml» (datatype: xml). La «knr» være primær nøkkel.

b) **NB! Svar på denne oppgaven ved å utføre oppgaven i gruppens PostgreSQL database.**

Opprett tabellen KUNDE_NY fra oppgave 2a) i «oblig4» skjemaet.

c) **NB! Svar på denne oppgaven i PDF-dokumentet.**

Skriv en INSERT spørring som migrerer data fra KUNDE tabellen til KUNDE_NY tabellen.

Tips! - INSERT INTO kunde_ny (knr, kunde_xml) SELECT

- Bruk XPath() (og eventuelt andre) funksjoner i SELECT uttrykket.

d) **NB! Svar på denne oppgaven ved å utføre oppgaven i gruppens PostgreSQL database.**

Kjør INSERT spørringen fra oppgaven over, og migrer kunde informasjonen inn i KUNDE_NY tabellen.

e) **NB! Svar på denne oppgaven i PDF-dokumentet.**

Skriv en spørring som benytter XPATH funksjonen (og eventuelt andre funksjoner), og som henter ut data om alle kunder fra KUNDE_NY som representert i bildet under.

knr [PK] integer	navn text	adresse text	postnr text
5002	Paal Aass	Lindemans gate79	1711
5007	Joakim Laursen	Thomas Heftyes gate 39	0015
5009	Laurits Eckhoff	Solheimgata 81	0654
5011	Åshild Sætran	Erling Skjalgssons gate 56	3750

f) **NB! Svar på denne oppgaven i PDF-dokumentet.**

Skriv en spørring som henter ut kunder med etternavn som starter på «A» fra KUNDE_NY som vist i tabellen under, og sorter på etternavn:

knr [PK] integer	navn text	adresse text	postnr text
5527	Alex Akselsen	Arnoldus Reimers' gate 24	5859
5585	Amalie Albrigtsen	Askøyveien 27	5258
5813	Amina Alexandersen	Asvlnlassen 2	3199

(Oppgaven fortsetter på neste side)

Ordre (og Ordrelinje)**g) NB! Svar på denne oppgaven i PDF-dokumentet.**

Skriv en SQL som oppretter en tabell «ORDRE_NY» med følgende kolonner:

«ordrenr» (datatype: integer), «kundenr» (datatype: integer), og «ordre_xml»(datatype: xml).

La «ordrenr» være primær nøkkel, og definer «knr» med referanseintegritet (FK) mot «kunde_ny» tabellen.

h) NB! Svar på denne oppgaven ved å utføre oppgaven i gruppens PostgreSQL database.

Opprett tabellen ORDRE_NY fra oppgaven over i «oblig4» skjemaet.

i) NB! Svar på denne oppgaven i PDF-dokumentet.

Skriv en INSERT spørring som migrerer data fra ORDRE tabellen til ORDRE_NY tabellen.

Tips! INSERT INTO ordre_ny (ordrenr, kundenr, ordre_xml) SELECT

j) NB! Svar på denne oppgaven ved å utføre oppgaven i gruppens PostgreSQL database.

Kjør INSERT spørringen fra oppgave over for å migrere ordre informasjonen (kun fra ORDRE - tabellen) inn i ORDRE_NY tabellen.

k) NB! Svar på denne oppgaven i PDF-dokumentet.

Lag en ny INSERT spørring som migrerer data fra ORDRE og ORDRELINJE tabellene, og inn i ORDRE_NY tabellen. Kolonnen «ordre_xml» skal følge XML strukturen vist i eksempelet under:

```
<ordre>
  <ordredato>2019-11-17</ordredato>
  <sendtdato>2019-11-22</sendtdato>
  <betaltdato>2019-12-03</betaltdato>
  <ordrelinjer>
    <ordrelinje>
      <vnr>32069</vnr>
      <prisprenh>177.00</prisprenh>
      <antall>2</antall>
    </ordrelinje>
    <ordrelinje>
      <vnr>33044</vnr>
      <prisprenh>16.50</prisprenh>
      <antall>1</antall>
    </ordrelinje>
    <ordrelinje>
      <vnr>45923</vnr>
      <prisprenh>48.00</prisprenh>
      <antall>2</antall>
    </ordrelinje>
  </ordrelinjer>
</ordre>
```

Tips!

- INSERT INTO ordre_ny (ordrenr, kundenr, ordre_xml) SELECT

- Start med å skrive den indre spørringen som henter <ordrelinjer> først.

Her må du bruke GROUP BY, og XMLAGG funksjonen.

- Bruk en LEFT JOIN (mot den indre spørringen du laget over):

FROM ordre LEFT JOIN (SELECT ... FROM ordrelinje GROUP BY ordrenr)

(Oppgaven fortsetter på neste side)

l) **NB! Svar på denne oppgaven ved å utføre oppgaven i gruppens PostgreSQL database.**

MEN ... hvis dere ikke får den til, så legg inn rader med kun data fra ORDRE-tabellen.

Slett først alle data i tabellen ORDRE_NY (fra forrige oppgave).

Deretter skal dere kjøre INSERT spørringen fra oppgaven over for å migrere ordre informasjonen (fra ORDRE og ORDRELINJE tabellene) inn i ORDRE_NY tabellen.

m) **NB! Svar på denne oppgaven i PDF-dokumentet.**

Denne oppgaven forutsetter at du har fått til oppgave 2k) og 2l). Hvis ikke, kan du allikevel prøve å skrive SQL-en, men du vil ikke få samme resultat som vist under.

Skriv en spørring som henter ut data fra ORDRE_NY tabellen. Spørringen skal hente alle ordre for kunde nummer (knr) 5643 som ble sendt («sendtdato») i august 2019 («2019-08»). Spørringen skal hente ut «ordrenr», «knr», «prisprøhet» og «antall» som vist under:

ordrenr [PK] integer	knr integer	prisprøhet double precision	antall integer
22020	5643	329.5	3
22020	5643	177	5
22020	5643	21	1
21943	5643	163.5	5
21943	5643	757	1

(Oppgaven fortsetter på neste side)

Oppgave 3 - MongoDB

I denne oppgaven skal vi migrere data fra PostgreSQL og inn i vår MongoDB database.

a) **NB! Svar på denne oppgaven i PDF-dokumentet.**

Hobbyhuset har besluttet at de ønsker tre samlinger (Eng: «collections») i MongoDB databasen:

«kunde», «vare» og «ordre».

Samlingen «kunde» skal inneholde data fra KUNDE og POSTSTED tabellen, og samlingen «vare» skal inneholde data fra VARE tabellen (og eventuelt KATEGORI tabellen. Valgfritt). Samlingen «ordre» skal inneholde informasjon både fra ORDRE og ORDRELINJE tabellen, der data fra ORDRELINJE blir innebygde (Eng: «embedded») dokumenter i «ordre» dokumentene.

Lag et eksempel på et JSON dokument for samlingene «vare» og «ordre».

Tips! I oppgave 3 b) kan du se eksempel på et JSON dokument for samlingen «kunde».

b) **NB! Svar på denne oppgaven i PDF-dokumentet.**

Skriv de 3 SQL-ene for å hente ut data som samsvarer med «datamodellen» som ble grovt skissert i «oppgave 3a» (dvs. en SQL per samling «kunde», «vare» og «ordre»).

SQL-ene skal skape et JSON dokument per rad. **NB! Dette er viktig ettersom vi senere skal skrive Java kode som laster fil data inn basert på disse SQL-ene inn i en MongoDB database.**

Tips!

SQL-en for samlingen «kunde» skal eksempelvis opprette tilsvarende rader for hver kunde:

{knr: "5002", fornavn: "Paal", etternavn: "Aass", adresse: "...", postnr: "...", poststed: "..."}"

c) **NB! Filene du oppretter i denne oppgaven skal legges ved i Eclipse prosjektet i oppgave 4 (i mappen: «data»).**

Kjør SQL-ene fra oppgaven over (3b), og lagre resultatet fra hver SQL i en fil med følgende format: «<collection>.json». Et forslag til filen «kunde.json» finnes allerede i mappen «data» i vedlagt Eclipse prosjekt. Gruppen kan velge om du/dere vil bruke denne filen eller gruppens egen.

Tips! Se video om obligatorisk oppgave 4. Her viser vi hvordan vi kan lagre resultatet av en SQL kjøring i pgAdmin og Query Tool.

(Oppgaven fortsetter på neste side)

Oppgave 4 – MongoDB og Java

I denne oppgaven skal du fullføre et eksisterende Java prosjekt i Eclipse.

For å fullføre oppgave 4 b – e må du legge til/ endre kode i følgende filer:

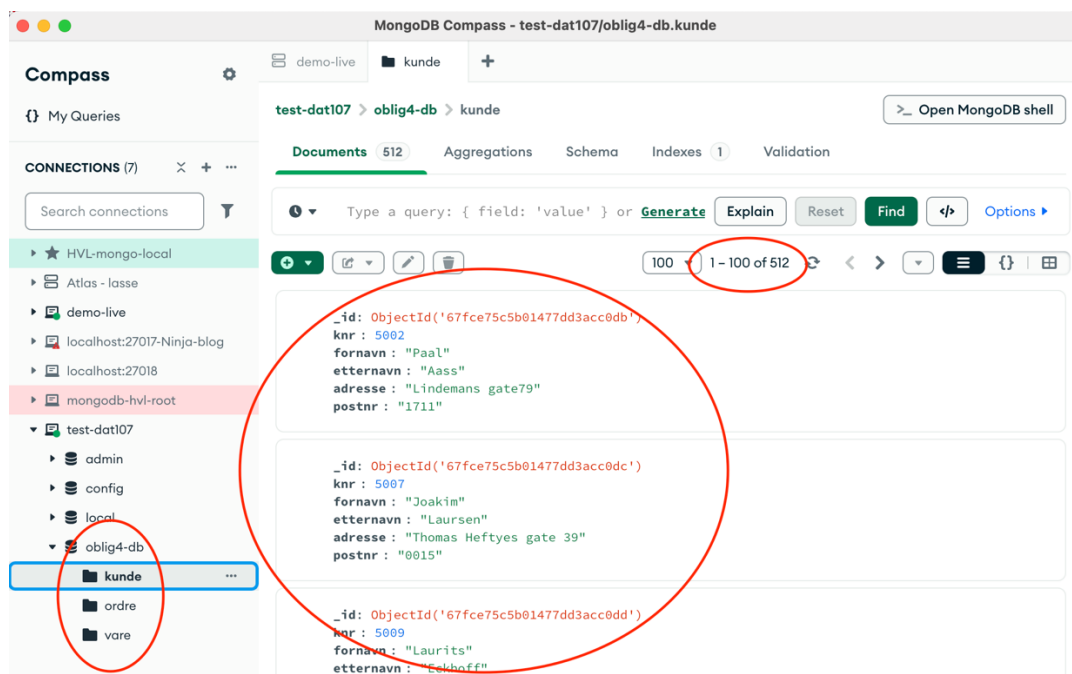
- MongoDBOblig4.java
- KundeRepository.java

- Pakk ut filen «eclipse-oblig4-template.zip» som er vedlagt i oppgaven. Pakk ut filen og åpne som nytt prosjekt i Eclipse.
- Legg til kode i «main» metoden i «MongoDbOblig4.java» for å laste data inn i gruppens MongoDB database (søk etter «oppgave 4 b»). Bruk LoadDataDao.loadCollection("<collection>", "<filename>") til å laste filene fra oppgave 3c.

Tips! Husk å kopiere filene fra oppgave 3c inn i «data» mappen før du kjører programmet. Filen «kunde.json» finnes allerede, så her kan du velge om du vil benytte denne eller gruppens egen.

- NB! Svar på denne oppgaven i PDF-dokumentet ved å lime inn skjermdump av innlest data i MongoDB. Skjermdumpen skal inneholde database med collection, noen dokumenter, samt oversikt over hvor mange dokumenter som finnes i collection. En skjermdump for hver samling/ «collection».**

Eksempel:



(Oppgaven fortsetter på neste side)

- d) Metoden «findByKnr» i KundeRepository.java skal gi oss mulighet til å søke frem en kunde basert på kunde nummer (knr). Fullfør metoden «findByKnr» slik at følgende kodelinjer fungerer i «main» -metoden i MongoDBOblig4.java:

```
Kunde kunde = repoKunde.findByKnr(5002);
```

- e) Metoden «save» i KundeRepository.java skal gi oss mulighet til å legge til en kunde i samlingen. Fullfør metoden «save» slik at følgende kodelinje fungerer i «main» -metoden i MongoDBOblig4.java:

```
Kunde aKunde = new Kunde(5001,"Ola", "Hansen", "Lia 2", "1234");
Kunde nyKunde = repoKunde.save(aKunde);
```

- f) Metoden «delete» i KundeRepository.java skal gi oss mulighet til å slette en kunde i samlingen. Fullfør metoden «delete» slik at følgende kodelinje(r) fungerer i «main» -metoden i MongoDBOblig4.java:

```
Kunde slettetKunde = repoKunde.delete(5007);
```

- g) I denne oppgaven skal vi oppdatere kunden vi hentet i «oppgave 4 d» (dvs knr 5002). Metoden «update» i KundeRepository.java skal gi oss mulighet til å lagre oppdatert informasjon i et kunde-objekt. Fullfør metoden «update» slik at følgende kodelinje(r) fungerer i «main» -metoden i MongoDBOblig4.java:

```
kunde.setFornavn("Pål");
Kunde endretKunde = repoKunde.update(kunde.getId(), kunde);
```

- h) **NB! Svar på denne oppgaven i PDF-dokumentet.**

Kjør det ferdige programmet, og kopier utskrift (Eng: output) inn i PDF-dokumentet.

God jobbet!