

# Obligatorisk arbeidskrav DAT2000 – Database 2

Læringsutbytte - Etter fullført emne skal studenten inneha følgende:

## Kunnskap

- sentrale oppgaver i databaseadministrasjon og -sikkerhet, som autorisasjon og tilgangskontroll, sikkerhetskopiering (backup) og gjenoppretting (recovery), replikering og synkronisering
- bruksområder og virkemåte for lagrede prosedyrer og triggere
- transaksjonshåndtering, samtidighet og låsing i en flerbrukerdatabase
- fysiske lagringsmåter for relasjonelle data
- hensikten med indekser og denormalisering, og hvordan dette kan oppnås
- hvordan databasesystemet gjør optimalisering av spørringer, og utfører disse
- andre, ikke-relasjonelle måter å lagre data på, f.eks. objektorienterte databaser og NoSQL databaser

## Ferdigheter

- utføre grunnleggende administrasjon og drift av et databasesystem
- programmere lagrede prosedyrer, triggere og transaksjonshåndtering med et prosedyrespråk i et relasjonsdatabasesystem (RDBMS)
- grunnleggende bruk av ikke-relasjonell datalagring og semistrukturerte data

## Generell kompetanse

- kunne tilegne seg oppdatert kunnskap innenfor ovennevnte kunnskaps- og ferdighetsområder
- forstå hvordan databaser inngår som del av applikasjonsutvikling og -drift
- ha erfaring med å jobbe i gruppe

## Praktiske opplysninger og innlevering

### Innleveringsfrist

Besvarelsen skal leveres digitalt i én pakket fil (zip) i Canvas innen **torsdag 29/10 kl 15:30**.

### Hjelpemidler og fusk

Reglene for fusk for obligatoriske arbeidskrav er de samme som for eksamen, og er beskrevet i §8-8 i Forskrift om studier og eksamen ved USN (<https://lovdata.no/dokument/SF/forskrift/2018-06-15-1090#shareModal> ).

Alle trykte og skrevne hjelpemidler er tillatt, men bruk av ferdige løsninger og/eller nettsider regnes som fusk. Det bruk av ressurser utviklet av andre (faglitteratur, internettsider etc.) gjelder kravene til kildehenvisning.

## Gruppeinnlevering

Arbeidskravet skal besvares og leveres inn i grupper. Dersom du ennå ikke har en gruppe må du melde fra til emneansvarlig i løpet av torsdag 15/10. Gruppestørrelsen er 3-5, men det vil kunne bli gitt dispensasjon etter søknad til emneansvarlig. Gruppene skal registrere seg i Canvas innen tirsdag 19/10. Bruk gruppene under overskriften Arbeidskrav DAT2000.

Arbeidskravet er en gruppeoppgave og det innebærer at gruppen skal løse oppgaven sammen, **uten** hjelp fra personer utenfor gruppen. Gruppen skal levere en selvstendig besvarelse, uten bidrag fra andre utenfor gruppen. Slikt samarbeid er å regne som fusk i henhold til forskriften over.

Alle gruppemedlemmer skal delta i løsning av arbeidskravet – **gruppemedlemmer som ikke bidrar i arbeidet vil ikke få godkjent arbeidskravet.**

## Hva skal leveres

Det innleverte arbeidet (zip-fil) skal inneholde:

1. Et dokument med de delene som ikke leveres som SQL-script (se punkt 2):
  - Navn og studentnummer på alle gruppemedlemmene
  - Kort presentasjon av informasjonsområdet gruppen har valgt
  - Ansvarskart over hva de ulike gruppemedlemmer har bidratt med
  - Datamodell for databasen (lages i f.eks DBeaver)
  - SQL-spørringer mot databasen og tilhørende eksekveringsplaner og kommentarer
2. En samling SQL-script som:
  - Bygger databasen
  - Fyller databasen med testdata
  - Definerer brukergrupper/roller, brukere og rettigheter
  - Definerer lagrede rutiner (prosedyrer og funksjoner) i databasen
  - Definerer triggere og tilhørende funksjoner som utløses før og etter behandling av data
  - Tester på lagrede rutiner og triggere i databasen
  - Eksempel på transaksjoner med COMMIT og ROLLBACK.

## Oppgavetekst

Arbeidskravet går ut på å modellere, designe og realisere en database i PostgreSQL som innfrir kravene under. Dere bestemmer selv hva databasen skal representere og hvilke relasjoner den skal inneholde. Det er viktig at dere ikke lager databasen for omfattende, men den må minimum inneholde 3 relasjoner (les: tabeller). Ta gjerne utgangspunkt i et område dere som gruppe har kjennskap til.

## Datamodell

Det skal utarbeides en datamodell (ER) som danner grunnlag for den realiserde databasen. Datamodellen skal vise primær- og fremmednøkler.

## Realisert database

PostgreSQL databasen skal innfri følgende krav:

- Databasen skal inneholde minimum 3 tabeller – dere bør som ikke lage den for omfattende.
- Valg av datatyper skal begrunnes og ikke medføre at databasen krever mer lagringsplass enn nødvendig.
- Valg av primærnøkler skal grunngis og både primærnøkler og fremmednøkler skal være definert.
- Databasen skal definere relevante/nyttige indekser der det er fornuftig – utover primær/fremmednøkler

## Datainnhold

Lag script som fyller databasen med testdata. Dersom det benyttes «reelle» data så er det viktig at disse anonymiseres. For å generere større datasett kan dere bruke <https://www.mockaroo.com/>, eventuelt søke på nettet etter måter å genere testdata – oppgi uansett hvordan dere har generert testdata.

## Spørringer og spørreoptimalisering

Dere skal lage ulike spørringer mot databasen med ulik *kompleksitet*. Dere bør ha med spørringer som henter data fra flere tabeller og som viser bruk av både JOIN og predikat. For hver spørring skal eksekveringsplanen analyseres med EXPLAIN og besvarelsen skal vise spørringene med tilhørende eksekveringsplan (grafisk og tabellform). Spørringen og planene bør vise ulike aksessmetoder. Det skal også gis en kort forklaring på hver av kjøreplanene, med forklaring av aksessmetodene optimalisatoren har nyttet og hvorfor den har valgt disse.

## Brukergrupper og brukeradministrasjon

Besvarelsen skal inneholde forslag til brukere og roller, og rettigheter til databasen. Det skal minimum lages en administrator gruppe, en gruppe for å legge inn og oppdatere data i noen gitte tabeller og en gruppe/rolle som har leserretter til gitte tabeller/view i databasen. Dere skal også lage en rolle som skal ha rettigheter til å eksekvere en eller flere lagrede rutiner. Alle SQL-kommandoene i denne deloppgaven skal samles i et SQL-script. Dette skal kunne kjøres flere ganger uten å resultere i feil.

## Lagrede rutiner og triggere

Dere skal også vise bruk av prosedyrer, funksjoner og triggere i databasen, og dere bestemmer selv bruksområder for disse, men som et minimum skal dere lage følgende:

- En lagret prosedyre for å sette inn en rad i minst en av tabellene
- En lagret prosedyre for å slette en rad i minst en av tabellene
- En lagret prosedyre som setter inn/oppdaterer data i minst to tabeller ved hjelp av en transaksjon
- En lagret prosedyre som sikrer «forretningsregler» i databasen
- En lagret funksjon som nytter data fra minst to kolonner i en tabell
- En lagret funksjon som summerer/kombinerer data fra mer enn én rad i en tabell.
- Triggere som sikrer konsistens i databasen, f.eks. ved å oppdatere en tabell som følge av endringer i en annen tabell, eller sette defaultverdier i nye rader.
- En trigger som tar vare på historiske data ved endring/sletting av rader i en tabell.

Besvarelsen skal inneholde sql-script som tester(kjører) de lagrede rutinene.

## Transaksjoner

I denne delen av oppgaven skal dere vurdere hvorvidt forløpene som er skissert er konflikt serialiserbare eller ikke. Dere skal dokumentere med transaksjonstabell og presedensgraf. Det er også lurt av dere å bygge opp presedensgrafene stegvis.  $R(x)$  og  $W(x)$  står for henholdsvis Read(x) og Write(x).

- Gitt følgende forløp  $S_a$  for to forløp  $T_1$  og  $T_2$  på element  $x$ .  
(  $T_1, R(x)$  ), (  $T_2, R(x)$  ), (  $T_1, W(x)$  ), (  $T_2, W(x)$  )
- Gitt følgende forløp  $S_b$  for to forløp  $T_1$ ,  $T_2$  og  $T_3$  på elementene  $x$  og  $y$ .  
(  $T_1, R(x)$  ), (  $T_1, R(y)$  ), (  $T_1, W(x)$  ), (  $T_2, R(y)$  ), (  $T_3, W(y)$  ), (  $T_1, W(x)$  ), (  $T_2, R(y)$  ),
- Gitt følgende forløp  $S$  for fem transaksjoner  $T_1$ ,  $T_2$ ,  $T_3$ ,  $T_4$  og  $T_5$  på elementene  $v$ ,  $x$ ,  $y$  og  $z$ :  
(  $T_1, R(x)$  ), (  $T_2, W(x)$  ), (  $T_3, R(x)$  ), (  $T_1, R(y)$  ), (  $T_4, R(z)$  ), (  $T_2, W(y)$  ), (  $T_1, R(v)$  ), (  $T_3, W(v)$  ), (  $T_4, R(v)$  ),  
(  $T_4, W(y)$  ), (  $T_5, W(y)$  ), (  $T_5, W(z)$  )

## Backup og Recovery

Dere skal foreta en backup både i cmd og i et GUI verktøy. Ta med skjermbilder for oppretting av arkivfilene og gjenoppretting av arkivfilen til en ny database.