

# Prosjekt 2024 – TDT4145

Selve oppgaven er beskrevet i eget dokumentet. Dette dokumentet beskriver kravene til leveransene. Dere velger selv om dere vil skrive på norsk eller engelsk, men ikke bland dem.

## Innlevering, del 1: ER-modell og relasjonsdatabaseskjema

**DB1: Frist 8. mars 23:59 (11:59 pm) på Blackboard.**

- a) En ER-modell som viser deres fullstendige datamodell. Dere står fritt til å bruke alle ER- og EER-konsepter som er gjennomgått i emnet. Dokumenter de forutsetningene dere har gjort og eventuelle restriksjonene som ikke (kan) uttrykkes gjennom ER-modellen.
- b) ER-modellen oversatt til relasjonsdatabaseskjema (tabeller). Spesifiser nøkler og fremmednøkler for hver tabell. For hver tabell skal du forklare hvorfor tabellen er på fjerde normalform, eventuelt forklar hvorfor du velger å ha den på en lavere normalform (du må forklare hvilken).
- c) Et SQL-script som konstruerer databasen med tabellene. Husk å spesifiser primær- og fremmednøkler, samt andre nødvendige restriksjoner. Dokumenter restriksjoner som ikke uttrykkes i relasjonsdatabaseskjemaet og derfor må håndteres i applikasjonsprogrammene.

Lever besvarelsen som PDF-fil. SQL-scriptet skal legges ved som en .sql-fil (tekstfil). Dokumentet skal være oversiktlig og konsist, og figurene skal være enkle å forstå. Husk å ta med navn på alle gruppemedlemmene og lever på BlackBoard.

## Innlevering, del 2: Realisert databasesystem

**DB2: Frist 21. mars 23:59 (11:59 pm) på Blackboard**

**TeaterDB implementert i Python med bruk av sqlite3.**

Databaseapplikasjonen skal implementeres i Python (noen brukerhistorier skrives som SQL-script, se beskrivelsen) basert på skjemaet fra første delinnlevering. Brukerhistoriene må være realisert og tilfredsstilt. Enkleste løsning er å lage et tekstbasert grensesnitt som kjører i et terminalvindu (eksempelvis cmd, bash, o.l.). Husk at poenget med oppgaven er å lage modeller, skrive SQL og gjøre databaseaksess fra Python. Følgende skal leveres:

- a) Python kildekode med SQL og SQL-scripts pakket i en zip-fil.
- b) Databasefilen til prosjektets SQLite-database. Denne skal være tom, slik at sensor kan kjøre alle initialiseringsprogrammene dere leverer. Det skal være med en oppskrift på hvordan sensor skal kjøre programmet, sammen med eksempeldata som brukes som input. Det er viktig at sensor kan kjøre programmet uten å streve.
- c) De tekstlige resultatene (output) fra brukerhistorienes spørringer.

Dokumentet skal være konsist og evt. figurer enkle å forstå. Inkluder navn på alle gruppemedlemmene og lever på BlackBoard.

## Bruk av KI – kunstig intelligens

Hvis dere gjør bruk av kunstig intelligens i prosjektet, må dere dokumentere hvordan og hvor dere har brukt dette. Blant annet må dere vise hvilken del av koden som er generert av KI og hvilken del som dere har programmert selv.

## Evalueringskriterier

Følgende kriterier ligger til grunn for vurderingen:

### DB1

1. Bruk av entiteter, relasjoner og attributter. Disse konseptene skal anvendes på en hensiktsmessig måte, dvs. det skal være lett å forstå hva de modellerer.
2. Bruk av nøkler, herunder naturlige og genererte.
3. Bruk av restriksjoner, f.eks. kardinaliteter, i modellen. Er disse anvendt på korrekt vis?
4. Riktig oversetting til relasjonsmodellen. Korrekt beskrivelse av normalformer.
5. Oversetting til SQL-tabeller. Korrekt bruk av SQL, herunder attributtdomener, (fremmed-)nøkkelrestriksjoner og UNIQUE.
6. Dokumentene skal være konsise og figurene enkle å forstå.

Følgende poenger brukes på de tre delene av leveransen.

- a) Inntil 30 poeng
- b) Inntil 10 poeng
- c) Inntil 10 poeng.

### DB2

1. Korrekt bruk av SQL i Python. Vi favoriserer SQL framfor Python, slik at hvis det er greit å løse noe i SQL, ønsker vi det, framfor at det løses i Python.
2. Forståelig og lesbar kode.
3. Det skal være mulig å reprodusere de leverte resultatene ved hjelp av programmet og databasen som er levert. Oppskriften på hvordan programmet skal kjøres, skal fungere.

Karaktergivningen baserer seg på de generelle beskrivelsene man finner her:

<https://i.ntnu.no/wiki/-/wiki/Norsk/Karakterbeskrivelser+for+teknologiske+fag>

Følgende poenger brukes på de tre delene av leveransen.

- a) Inntil 35 poeng. 25 poeng på SQL og 10 poeng på tilhørende Python.
- b) Inntil 10 poeng. Reproduserbarhet av resultatene er viktig.
- c) Inntil 5 poeng på riktig output.