

TDT4145 Datamodellering og databasesystemer:

Prosjektinnleveringer

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 språk.

Del 1 teller 60 % og del 2 teller 40 %.

Innlevering, del 1: ER-modell og relasjonsdatabaseskjema

DB1: Frist mandag 10. mars 14:00 (02:00 pm) på Blackboard

(NTNU krever at vi leverer innen arbeidstid).

Innleveringen består av tre deler:

- 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. Det gis inntil 40 poeng for denne delen.
- 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). Det gis inntil 10 poeng for denne delen.
- 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. Det gis inntil 10 poeng for denne delen.

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 gruppelemmene og lever på Blackboard.

Innlevering, del 2: Realisert databasesystem

DB2: Frist fredag 21. mars 14:00 (02:00 pm) på Blackboard

(NTNU krever at vi leverer i arbeidstid).

FlyDB implementert i Python med bruk av sqlite3.

Databaseapplikasjonen skal implementeres i Python (noen brukstilfeller skrives som SQL-script, se beskrivelsen) basert på skjemaet fra første delinnlevering. Brukstilfellene 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:

- Python kildekode med SQL og SQL-scripts pakket i en zip-fil.
- 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 ev. figurer enkle å forstå. Husk å ta med navn på alle gruppe-medlemmene 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) Struktur og sammenheng i datamodellen – bruk av entitetsklasser, relasjonsklasser og attributter. Disse modellkonseptene skal anvendes konsekvent og hensiktsmessig, det skal være lett å forstå hva de modellerer.
- 2) Bruk av nøkler, herunder naturlige og genererte nøkler.
- 3) Bruk av restriksjoner, som for eksempel strukturelle restriksjoner, i modellen. Disse skal anvendes konsekvent og korrekt.
- 4) Riktig oversetting til relasjonsmodellen. Korrekt beskrivelse og vurdering av normal-former.
- 5) Oversetting til SQL-tabeller. Korrekt bruk av SQL, herunder attributtdomener, (fremmed-)nøkkelrestriksjoner og UNIQUE.
- 6) Dokumentene skal være konsise og figurene skal være 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. Alle data som er nevnt i oppgaven skal settes inn.
- 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.

Følgende poeng gis på de tre delene av leveransen:

- a) Inntil 30 poeng. 20 poeng på SQL og 10 poeng på tilhørende Python.
- b) Inntil 5 poeng. Reproduserbarhet av resultatene er viktig.
- c) Inntil 5 poeng på riktig output.

Generelt:

Karaktergivningen baserer seg på de generelle beskrivelsene man finner her:

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