

EKSAMENSFORSIDE

Skoleeksamen med tilsyn

Generell informasjon om eksamen:

Emnekode: DAT1000

Emnenavn: Database 1

Dato: 14.12.2023

Start klokkeslett: 09:00

Antall timer: 4

Emneansvarlig: Bjørn Kristoffersen

Campus: Bø

Fakultetet: HH

Antall oppgaver: 4

Antall vedlegg: 1

Totalt antall sider (inkludert forside): 8

Tillatte hjelpemidler (jfr. emneplan):

Tillatte hjelpemidler er definert som «alle trykte og skrevne». Du kan altså ta med bøker og alle slags notater på papir.

Du kan i tillegg bruke følgende programmer/verktøy for å gjennomføre eksamen:

- MySQL Workbench koblet opp mot enten lokal database eller itfag.usn.no
- phpMyAdmin via nettleseren på adresse <https://itfag.usn.no/phpmyadmin>
- Visual Paradigm (eller bare et enkelt tegneprogram som Paint)
- DBeaver koblet opp mot enten lokal database eller itfag.usn.no
- redigere egen besvarelse i Word (med mulighet for å generere PDF-filer)
- ta skjermbilder av datamodell og lime inn i Word (innebygd verktøy i både Windows og på Mac)

KANDIDATEN MÅ SELV KONTROLLERE AT OPPGAVESETTET ER FULLSTENDIG

- se på PDF-filer lagret på egen maskin
- se på tekstfiler/notater lagret på egen maskin (ikke OneDrive) i Word eller en enkel editor.

Du kan bruke nettleseren for å laste ned oppgaven og laste opp besvarelsen via Wiseflow.

Du kan også bruke nettleseren på følgende adresser:

- <https://dev.mysql.com/doc> (for å lese MySQL-dokumentasjonen)
- <https://allvit.no> (hvis du har skaffet deg læreboken som e-bok)

Det er ikke lov å samarbeide med andre om hvordan du skal skrive besvarelsen. Tekst som leveres inn på eksamen skal være ditt eget arbeid. Det er ikke tillatt å bruke kunstig intelligens (som f.eks. ChatGPT eller Copilot) under eksamen, heller ikke hvis slike funksjoner er integrert i verktøyene listet over (f.eks. Word). Merk at WiseMonitor gjør opptak av skjermen underveis.

Vi gjør oppmerksom på at det kjøres plagiatskontroll på alle besvarelser, og unaturlige likheter, ulovlig bruk av hjelpemidler etc. – eller at du ikke har skrevet besvarelsen selv – medfører at USN oppretter sak om fusk. Fusk kan resultere i annullering av besvarelsen, tap av retten til å ta eksamen ved USN og andre universiteter og høyskoler, eller midlertidig utestengelse fra USN. Beskrivelse av individuell eksamen og ulovlig samarbeid finner du på USN Intranett.

Opplysninger om vedlegg:

SQL-skript som oppretter databasetabeller med eksempeldata.

Merknader:

Vekting av oppgavene ved sensur:

- Oppgave 1: 40%
- Oppgave 2: 35%
- Oppgave 3: 10%
- Oppgave 4: 15%

Vektingen gir også en god pekepinn til hvor mye tid du bør sette av til hver oppgave. Legg ved dine egne forutsetninger / tolkinger hvis oppgaveteksten er uklar eller tvetydig. Prøv å svare på så mange spørsmål som mulig. Lykke til!

Oppgave 1 (40%)

Tabellene under er en del av databasen til Fotturforeningen i Utopia. Primærnøkler er understreket og fremmednøkler er merket med en stjerne.

- Deltaker(DNr, Fornavn, Etternavn, EPost)
- Hytte(HNr, Navn, Telefon, AntSenger)
- Fottur(FNr, StartDato, Turtype, AntDager, StartHytte*, Pris)
- Påmelding(FNr*, DNr*, RegDato)

Vedlegget inneholder et SQL-skript som lager disse tabellene og setter inn noen få rader med eksempeldata. Gjør deg kjent med databasen og kjør gjerne skriptet for å teste spørringer. Det er forutsatt at du kjører skriptet i en tom database, men øverst i skriptet finner du tips til hvordan du kan lage en ny tom database (på din lokale maskin).

Du kan få bruk for funksjonene YEAR, MONTH, CURDATE og CONCAT, i tillegg til de vanlige mengdefunksjonene. Se MySQL-dokumentasjonen for forklaring til disse og andre funksjoner.

1-a (5%)

Skriv en SQL-spørring som viser alle fotturer i 2023 med varighet 3 dager.

1-b (5%)

Lag en SQL-spørring som viser alle hytter med flere enn 10 senger og et navn som inneholder teksten «bu». Sorter utskriften synkende etter antall senger.

1-c (5%)

Bruk SQL for å vise deltakerlisten for fottur nummer 1. Utskriften skal inneholde navn på deltakere i én kolonne (fornavnet og et blankt tegn og etternavnet).

1-d (5%)

Lag en SQL-spørring som viser antall deltakere på hver fottur i 2023.

1-e (5%)

Forklar hvilke endringer som må gjøres i SQL-skriptet for å sikre at følgende gjelder:

- E-post og navn på deltaker skal alltid fylles ut og e-post skal dessuten være unik.
- Turtype skal ha én av følgende verdier: «Enkel», «Middels» og «Krevende».
- Pris (for en fottur) skal være mellom 0 og 50000 kroner.

1-f (5%)

Lag en SQL-spørring som viser deltakere som ikke er påmeldt noen fotturer.

1-g (5%)

Turledere skal ha mulighet for å se innholdet i alle tabellene og dessuten kunne sette inn, oppdatere og slette opplysninger i tabellen Fottur. Bruk SQL for å opprette en rolle turleder med de nevnte rettighetene og lag deretter en databasebruker for Kari og lag henne rollen som turleder.

1-h (5%)

Lag et view som viser påmeldinger til turer i inneværende måned (som nå er desember 2023). Viewet skal inneholde alle kolonner fra Påmelding, samt navn på deltakere og startdato for turen.

KANDIDATEN MÅ SELV KONTROLLERE AT OPPGAVESETTET ER FULLSTENDIG

Oppgave 2 (35%)

I denne oppgaven skal du tegne en logisk datamodell (ER-diagram). Diagrammet skal vise entiteter og attributter med datatyper, forhold med kardinaliteter samt primærnøkler og fremmednøkler. Du skal tegne identifiserende forhold med heltrukne linjer og ikke-identifiserende forhold med stiplede linjer. Du kan tegne diagrammene i MySQL Workbench, Visual Paradigm, Paint, eller du kan tegne for hånd (bruk da notasjonen i læreboken) og deretter ta et bilde av modellen, som du limer inn i besvarelsen.

Beskrivelse av systemet

SykloFab er en produksjonsbedrift som bygger sykler. Bedriften ønsker å utvikle et IT-system for håndtering av leverandører, delelager, produksjon og oppsett av jobbvakter. Du skal foreslå en database for løsningen.

SykloFab har et lager med deler til de ulike sykkelmodellene. En del har et unikt delenummer og et navn. En sykkelmodell har en unik modellkode, som består av to bokstaver og tre sifre, samt et beskrivende navn. Databasen skal også lagre antall enheter på lager av hver del, og hvor mange enheter av hver del som trengs for å lage én sykkel av de ulike sykkelmodellene. En bestemt del kan brukes i flere sykkelmodeller og en sykkelmodell kan være bygd opp av flere deler.

Delene kjøpes inn fra et antall leverandører. Om hver leverandør skal navn og telefonnummer lagres. En leverandør kan levere flere deler og samme del kan kjøpes fra flere leverandører. Dato for siste gang en bestemt del ble kjøpt inn av hver leverandør skal lagres i databasen. (Håndtering av bestillinger gjøres på nettsidene til leverandørene og er ikke en del av systemet.)

Produksjonen foregår på et antall monteringsstasjoner. Det kan jobbe flere ansatte på hver stasjon samtidig, men en ansatt jobber kun på én stasjon en bestemt dag. Om hver ansatt skal det lagres fornavn, etternavn, stilling og dato for ansettelse.

På en stasjon blir det kun bygd én sykkelmodell en bestemt dag, men en sykkelmodell kan bygges på flere stasjoner og over flere dager. Systemet skal kunne skrive ut jobblister som for hver stasjon viser hvilken sykkelmodell som skal bygges på denne stasjonen og hvilke ansatte som skal jobbe der.

Oppgave 3 (10%)

Fotturforeningen i Utopia vil utvide databasen fra oppgave 1 med opplysninger om beholdningen av kioskarer på hver hytte. Det er foreslått å utvide tabellen Hytte med noen nye kolonner:

- HytteUtvidet(HNr, Navn, Telefon, AntSenger, PNr, PNavn, AntEnheter)

Eksempelrad:

- (3, 'Uppigarden', '22334455', 12, 27, 'Kvikk Lunch', 38)

Denne raden beskriver at hytte nummer 3 heter Uppigarden og har telefonnummer 22334455. Uppigarden har 12 sengeplasser. Produkt 27 har navn Kvikk Lunch og på lageret til Uppigarden er det 38 enheter av dette produktet.

Forklar kort med bruk av konkrete eksempler hva som er uheldig med den nye tabellen. Skriv deretter ned funksjonelle avhengigheter, bestem kandidatnøkkel og utfør normalisering til BCNF. Få fram hvert normaliseringssteg. Du skal ikke legge til flere kolonner.

Vis primærnøkler med understreking og merk fremmednøkler med en stjerne i sluttresultatet.

KANDIDATEN MÅ SELV KONTROLLERE AT OPPGAVESETTET ER FULLSTENDIG

Oppgave 4 (15%)

Denne oppgaven består av 15 flervalgsspørsmål som hver teller 1% ved sensur.

- Hvert spørsmål har 4 svaralternativer.
- Kun 1 alternativ er riktig.
- Du kan velge å «gardere» ved å svare flere enn ett alternativ.
- Du får 3 poeng for riktig svar, -1 poeng for hvert gale svar og 0 poeng for ubesvarte spørsmål.
- Oppgi svar på formen: Spørsmål 1: a, d
 - Hvis a er riktig og d er feil, får du $3 - 1 = 2$ poeng på dette spørsmålet.
- **Merk forutsetninger:** Noen spørsmål viser til tabellene fra oppgave 1, men du skal nå gå ut fra at disse tabellene inneholder adskillig *flere* rader enn de har i vedlegget, nærmere bestemt:
 - Deltaker: 200 rader
 - Hytte: 40 rader
 - Fottur: 70 rader
 - Påmelding: 300 rader
 - Kolonnen Påmelding.DNr inneholder 180 ulike verdier.
 - Kolonnen Påmelding.FNr inneholder 60 ulike verdier.
 - Du skal også gå ut fra at *ingen* av tabellene inneholder **nullmerker**.
- Noen av spørsmålene forutsetter at du regner ut antall rader i ulike spørreresultater. Et tips er da at du kan bruke MySQL Workbench som kalkulator, f.eks.: `SELECT 2+2;`

Spørsmål 1

Hvilken tekst passer **ikke** med (matcher ikke) mønsteret 'x_y%' ?

- a) x1yyz
- b) xxy
- c) x12yz
- d) xyyz

Spørsmål 2

Hvilken WHERE-betingelse betyr det samme som `Pris <= 1500`?

- a) `NOT (DøgnPris > 1500) OR FALSE`
- b) `NOT (DøgnPris = 1500)`
- c) `NOT (DøgnPris < 1500) AND NULL`
- d) `DøgnPris < 1500 AND DøgnPris = 1500`

Spørsmål 3

Hvor mange rader gir følgende spørring – gitt **forutsetningene** forklart i innledningen til oppgave 4?

```
SELECT * FROM Deltaker, Fottur, Påmelding;
```

- a) Færre enn 100
- b) Mellom 50 og 250
- c) Mellom 200 og 1000
- d) Flere enn 2000

KANDIDATEN MÅ SELV KONTROLLERE AT OPPGAVESETTET ER FULLSTENDIG

Spørsmål 4

Hvor mange rader gir følgende spørring – gitt **forutsetningene** forklart i innledningen til oppgave 4?

```
SELECT * FROM Deltaker INNER JOIN Påmelding ON Deltaker.DNr = Påmelding.DNr;
```

- a) 200
- b) 300
- c) 500
- d) 6 000

Spørsmål 5

Hvor mange rader gir følgende spørring – gitt **forutsetningene** forklart i innledningen til oppgave 4?

```
SELECT * FROM Deltaker LEFT OUTER JOIN Påmelding ON Deltaker.DNr = Påmelding.DNr;
```

- a) Færre enn 100
- b) Mellom 50 og 200
- c) Mellom 150 og 400
- d) Flere enn 500

Spørsmål 6

Hvilken spørring gir flest rader – gitt **forutsetningene** forklart i innledningen til oppgave 4?

- a) SELECT COUNT(*) FROM Påmelding;
- b) SELECT DNr, COUNT(*) FROM Påmelding GROUP BY DNr;
- c) SELECT FNr, COUNT(*) FROM Påmelding GROUP BY FNr;
- d) SELECT DISTINCT FNr FROM Påmelding;

Spørsmål 7

Hva slags forhold vil det være mellom Deltaker og Påmelding i en datamodell for databasen fra oppgave 1?

- a) Ikke-identifiserende en-til-mange med kråkefot ved Deltaker
- b) Identifiserende en-til-mange med kråkefot ved Deltaker
- c) Ikke-identifiserende en-til-mange med kråkefot ved Påmelding
- d) Identifiserende en-til-mange med kråkefot ved Påmelding

Spørsmål 8

Hvilket alternativ er sant?

- a) Telefon + AntSenger er en supernøkkel for tabellen Hytte.
- b) HNr + AntSenger er en kandidatnøkkel for tabellen Hytte.
- c) HNr er en kandidatnøkkel for tabellen Hytte.
- d) Telefon + AntSenger er en primærnøkkel for tabellen Hytte.

Spørsmål 9

Nevn en teknikk for å sikre at alle verdiene i en kolonne alltid er én av verdiene 7, 11, 15 eller 18?

- a) Bruk ytre koblinger (LEFT OUTER JOIN) når du skriver spørringer som leser fra denne kolonnen.
- b) Sørg for at du alltid bruker INSERT-kommandoen for å sette inn disse verdiene.
- c) La kolonnen være fremmednøkkel mot en annen kolonne som inneholder disse 4 verdiene.
- d) Definer den aktuelle kolonnen som primærnøkkel med datatype INTEGER.

Spørsmål 10

Hvilket uttrykk betyr det samme som TRUE ?

- a) TRUE AND FALSE
- b) TRUE OR NULL
- c) NULL AND TRUE
- d) TRUE AND NULL

Spørsmål 11

Du skal velge datatype for en kolonne som skal inneholde romkoder på formen 9-999, altså et siffer etterfulgt av en bindestrek etterfulgt av tre sifre. Velg en datatype som gjør det mulig å lagre slike verdier og som samtidig vil kreve minst lagringsplass.

- a) VARCHAR(100)
- b) INTEGER
- c) DECIMAL(3, 1)
- d) CHAR(50)

Spørsmål 12

Hvor mange bytes kreves for å representere 8 millioner forskjellige verdier?

- a) 3
- b) 4
- c) 128
- d) 256

Spørsmål 13

Hvilket view er oppdaterbart?

- a) CREATE VIEW A AS SELECT Hyttetype, COUNT(*) FROM Hytte GROUP BY Hyttetype;
- b) CREATE VIEW B AS SELECT AVG(AntSenger) FROM Hytte;
- c) CREATE VIEW C AS SELECT DISTINCT Hyttetype FROM Hytte;
- d) CREATE VIEW D AS SELECT HNr, AntSenger FROM Hytte;

Spørsmål 14

Hvor mange rader gir følgende spørring – gitt **forutsetningene** forklart i innledningen til oppgave 4?

```
SELECT * FROM Deltaker WHERE DNr NOT IN (SELECT DISTINCT DNr FROM Påmelding);
```

- a) 0
- b) 20
- c) 200
- d) 300

Spørsmål 15

Hvordan blir et mange-til-mange forhold representert i databasen?

- a) Som en koblingstabell.
- b) Primærnøkklene på mange-siden blir kopiert inn på en-siden og blir til en fremmednøkkel.
- c) Begge tabellene får fremmednøkler til hverandre.
- d) Som en kombinert primærnøkkel og fremmednøkkel.