

EKSAMENSFORSIDE

Skoleeksamen med tilsyn

Generell informasjon om eksamen:

Emnekode: DAT1000 / DAT1000N / INF105E

Emnenavn: Database 1

Dato: 06.12.2024

Start klokkeslett: 10:00

Antall timer: 4

Emneansvarlig: Bjørn Kristoffersen

Campus: Bø / Gol / Drammen

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 skjermskudd 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)
- <https://dbsys.info/1000> (med undersider, for å lese fagstoffet til leksjonene)

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!

KANDIDATEN MÅ SELV KONTROLLERE AT OPPGAVESETTET ER FULLSTENDIG

Oppgave 1 (40%)

Tabellene under er en del av databasen til et rengjøringsbyrå som retter seg mot private boliger. Primærnøkler er understreket og fremmednøkler er merket med en stjerne.

- Kunde(KNr, Fornavn, Etternavn, Mobil)
- Bolig(BNr, Adresse, PostNr, Kvm, AntBad, KNr*)
- Avtaletype(TNr, Beskrivelse, KvmPris)
- Avtale(ANr, BNr*, TNr*, FraDato, Ukedag, DagerPrMnd)

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).

1-a (5%)

Skriv en SQL-spørring som viser kundenummer og navn til alle kunder som har fornavn eller etternavn som starter med 'O'.

1-b (5%)

Skriv en SQL-spørring som viser alle boliger med størrelse mellom 80 og 100 kvadratmeter og sorter dem etter adresse.

1-c (5%)

Skriv en SQL-spørring som viser alle avtaler med vaskedag på onsdager. Resultatet skal vise alle kolonner fra Avtale, men også adressen til boligen og navnet til kunden.

1-d (5%)

Skriv en SQL-spørring som viser antall boliger hver enkelt kunde eier, men kun de som eier flere enn 1 bolig skal tas med i resultatet. Det holder å vise kundenummer og antall boliger i resultatet.

1-e (5%)

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

- Adressen til boliger må alltid fylles ut.
- Mobilnummer til kunder skal være unikt.
- Antall kvadratmeter skal være et positivt tall.

1-f (5%)

Skriv SQL-kode som oppretter en bruker og gir brukeren rettigheter til å lese innholdet i tabellene Avtale og Avtaletype.

1-g (5%)

Skriv en SQL-spørring som viser alle kunder som ikke har en avtale.

1-h (5%)

Opprett et view som viser månedsprisen til alle avtaler. Viewet skal vise avtalenummer, kundens navn, avtaletype og månedspris. Kolonnen Avtaletype.KvmPris inneholder prisen pr. kvadratmeter for én dag.

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

Matkasse AS er en bedrift som leverer matkasser til privatpersoner. Bedriften tilbyr ulike typer matkasser som kundene kan abonnere på. En matkasse inneholder råvarer for å lage ulike middagsretter for et antall personer gjennom ei uke.

For å kunne tilby denne tjenesten, trenger Matkasse AS å lagre en rekke opplysninger i en database. Først og fremst trenger de å lagre informasjon om de ulike matvarene som brukes i matrettene. Dette inkluderer navn på matvaren, pris og om den selges i løsvekt eller ikke.

Videre trenger de å lagre informasjon om de ulike matrettene. Hver matrett har et navn og man lagrer også hvilke matvarer som inngår som ingrediens i de ulike matrettene. Alle oppskriftene er laget for én person.

En matkasse er en sammensetning av et antall matretter som kan være ukemeny for én person. Hver matkasse har et navn og en kort beskrivelse.

Matkasse AS må også å lagre informasjon om kundene sine. Dette inkluderer fornavn, etternavn, adresse, postnummer og mobilnummer. Hver kunde kan abonnere på en eller flere matkasser, og informasjon om abonnementene må også lagres. Dette inkluderer startdato for abonnementet og antall personer som matkassen skal lages for.

Når kundene skal velge matkasser, kan de søke etter matretter innenfor ulike kategorier, som f.eks. «italiensk», «norsk tradisjonsmat», «glutenfritt», «fisk», «kjøtt» og «vegetar». En matrett kan tilhøre én eller flere kategorier, og en kunde kan få lagret et antall slike kategorier som sine preferanser.

Oppgave 3 (10%)

Tabellen Turisme inneholder informasjon om kjente turistattraksjoner og personer som har besøkt disse stedene:

- Turisme(KNr, KNavn, ANr, ANavn, PID, PNavn, Dato)

Eksempelrad:

- ('0301', 'Oslo', 12, 'Operahuset', 103, 'Ole Mo', '2024-07-02')

Denne raden beskriver at Operahuset er attraksjon 12 og ligger i Oslo. Oslo har kommunenummer 0301. Person med person-ID lik 103 heter Ole Mo og besøkte Operahuset 2. juli 2024.

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:
 - Kunde: 500 rader
 - Bolig: 600 rader
 - AvtaleType: 10 rader
 - Avtale: 1000 rader
 - Kolonnen Avtale.BNr inneholder 580 ulike verdier.
 - Kolonnen Avtale.TNr inneholder 8 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 WHERE-betingelse betyr det samme som `KvmPris <> 100`?

- a) `KvmPris < 100 AND KvmPris > 100`
- b) `NOT (KvmPris < 100 AND KvmPris > 100)`
- c) `NOT (KvmPris = 100)`
- d) `NOT (KvmPris < 100 OR KvmPris > 100)`

Spørsmål 2

Hvilken tekst passer med (matcher) mønsteret `'%y_z_'` ?

- a) aybczd
- b) aybzcd
- c) yazb
- d) aybczd

Spørsmål 3

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

```
SELECT * FROM Avtale INNER JOIN Bolig ON Avtale.BNr = Bolig.BNr;
```

- a) 600
- b) 1 000
- c) 2 000
- d) Flere enn 10 000

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 Avtale, AvtaleType;
```

- a) Færre enn 100
- b) Mellom 500 og 2 000
- c) Mellom 5 000 og 15 000
- d) Flere enn 20 000

Spørsmål 5

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

```
SELECT * FROM Bolig LEFT OUTER JOIN Avtale ON Avtale.BNr = Bolig.BNr;
```

- a) Færre enn 100
- b) Mellom 500 og 2 000
- c) Mellom 3 000 og 4 000
- d) Flere enn 5 000

Spørsmål 6

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

- a) SELECT BNr, COUNT(*) FROM Avtale GROUP BY BNr;
- b) SELECT COUNT(*) FROM Kunde;
- c) SELECT TNr, COUNT(*) FROM Avtale GROUP BY TNr;
- d) SELECT DISTINCT TNr FROM Avtale;

Spørsmål 7

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

- a) Ikke-identifiserende en-til-mange med kråkefot ved Avtale
- b) Identifiserende en-til-en
- c) Ikke-identifiserende en-til-mange med kråkefot ved AvtaleType
- d) Mange-til-mange

Spørsmål 8

Hvilket uttrykk betyr det samme som TRUE i MySQL?

- a) TRUE AND FALSE
- b) TRUE AND NULL
- c) NOT NULL
- d) NULL OR TRUE

Spørsmål 9

Hvilket alternativ er sant?

- a) Fornavn + Etternavn er en supernøkkel for tabellen Kunde.
- b) KNr er en kandidatnøkkel for tabellen Kunde.
- c) PostNr + Kvm er en supernøkkel for tabellen Bolig.
- d) BNr er en primærnøkkel for tabellen Avtale.

Spørsmål 10

Nevn en teknikk for å sikre at alle verdiene i en tallkolonne er et heltall?

- a) Gi kolonnen datatype FLOAT.
- b) Definer kolonnen som primærnøkkel.
- c) Definer kolonnen med både UNIQUE og NOT NULL.
- d) Gi kolonnen en talldatatype som SMALLINT eller INTEGER.

Spørsmål 11

Du skal velge datatype for en kolonne som skal inneholde to bokstaver etterfulgt av to sifre. Velg en datatype som gjør det mulig å lagre slike verdier og som samtidig vil kreve minst lagringsplass.

- a) SMALLINT
- b) CHAR(4)
- c) DECIMAL(2, 2)
- d) LETTER_AND_DIGIT

Spørsmål 12

Hvor mange bytes kreves for å representere 300 forskjellige verdier?

- a) 2
- b) 4
- c) 8
- d) 150

Spørsmål 13

Hvilket view er oppdaterbart?

- a) CREATE VIEW A AS SELECT BNr, COUNT(*) FROM Avtale GROUP BY BNr;
- b) CREATE VIEW B AS SELECT AVG(KvmPris) FROM Avtale;
- c) CREATE VIEW C AS SELECT KNr, Fornavn FROM Kunde;
- d) CREATE VIEW D AS SELECT DISTINCT BNr FROM Avtale;

Spørsmål 14

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

```
SELECT * FROM AvtaleType WHERE TNr NOT IN (SELECT DISTINCT TNr FROM Avtale);
```

- a) 0
- b) 1
- c) 2
- d) 4

Spørsmål 15

Hva er hensikten med å opprette en indeks?

- a) Spare lagringsplass.
- b) Sørge for å redusere feilregistreringer.
- c) Sikre at databasen ikke mister data ved diskkrasj.
- d) Få søk til å bli utført raskere.