

EKSAMENSFORSIDE

Skriftlig eksamen med tilsyn

Emnekode: DAT1000/DAT1000N	Emnenavn: Database 1	
Dato: 04.05.2020	Tid fra / til: 10.00 – 14.00	Ant. timer: 4 (+ 30 minutter for opplasting)
Ansvarlig faglærer: Bjørn Kristoffersen		
Campus: Bø/Nett	Fakultet: Handelshøyskolen	
Antall oppgaver: 3	Antall vedlegg: 0	Ant. sider inkl. forside og vedlegg: 4
Tillatte hjelpemidler (jfr. emnebeskrivelse): Alle hjelpemidler er tillatt.		
Opplysninger om vedlegg:		
Merknader: Vekting av oppgavene ved sensur: <ul style="list-style-type: none"> • Oppgave 1: 35 % • Oppgave 2: 35 % • Oppgave 3: 30 % <p>NB! Du bør løse oppgave 1 før oppgave 2, fordi SQL-spørsmålene i oppgave 2 bygger på datamodellen du skal utvikle i oppgave 1.</p> <p>Vektingen gir også en god pekepinn til hvor mye tid du bør sette av til hver oppgave. Prøv å svare på så mange spørsmål som mulig. Legg ved dine egne forutsetninger hvis oppgaveteksten er uklar. Lykke til!</p>		

Kryss av for type eksamenspapir

Ruter Linjer

Oppgave 1

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 med MySQL Workbench, eller du kan tegne for hånd (bruk da notasjonen i læreboken) og deretter ta et bilde av modellen, som du limer inn i besvarelsen. Merk: Oppgaven er litt større enn normalt. Prøv å få til så mye som mulig!

Krav til systemet

Det skal utvikles en webapplikasjon for håndtering av konferanser og du skal lage en datamodell for dette. En konferanse kan strekke seg over et antall dager og det blir gjennomført presentasjoner i flere rom samtidig. Databasen skal kunne håndtere ulike typer av konferanser, men skal bare brukes for å gjennomføre én konferanse av gangen. Vi tenker oss altså at vi for hver konferanse starter med en tom database.

Hver konferanse blir delt inn i et antall faglige temaer. Konferansen har gjerne noen fellespresentasjoner, som blir avviklet i store fellesrom. De fleste presentasjonene blir imidlertid lagt inn under ett faglig tema. Slike presentasjoner blir gjennomført i mindre rom, flere samtidig. Vi kan f.eks. ha presentasjoner om tema A i rom 1 og presentasjoner om tema B i rom 2 på samme tid.

Deltakere kan melde seg på konferansen på nettet. De må da fylle inn navn og epost og får tildelt et unikt deltakernummer. For å planlegge bruk av rom, blir deltakere også spurt om hvilke temaer de er mest interesserte i. Under konferansen kan deltakere likevel bevege seg fritt på konferanseområdet og delta på ulike presentasjoner i den grad det er ledige plasser.

Databasen skal inneholde opplysninger om alle presentasjoner som skal holdes på konferansen. Hver presentasjon gis et unikt nummer, og man skal dessuten lagre tittel på presentasjonen, hvilket tema presentasjonen tilhører, et kort sammendrag samt varighet i minutter. Personer som skal holde en presentasjon melder seg også på konferansen via det samme nettskjemaet. De krysser av for at de skal holde presentasjon og blir da spurt om tittel på presentasjonen, tema, sammendrag og varighet (i minutter). Det er alltid kun én person som holder en bestemt presentasjon, men en person kan holde flere presentasjoner.

Noen av de som holder presentasjoner sender samtidig inn en bakgrunnsartikkel. Artikkelen har alltid samme tittel som presentasjonen, men ofte vil det være flere forfattere som samarbeider. Navn og epost til alle forfattere skal registreres, samt URL til selve artikkelen. Også forfattere blir tildelt et deltakernummer, selv om de ikke fysisk er med på konferansen.

Konferansen går over et antall dager, og i forkant av konferansen blir det laget et program der hver enkelt presentasjon blir plassert i et rom og får et starttidspunkt (dato og klokkeslett). Rommene som konferansen disponerer blir lagt inn i systemet, med et romnummer og antall plasser.

Det blir servert lunch og middag hver konferansedag. Ved påmelding krysser deltakere av for hvilke måltider de ønsker. Det skal f.eks. være mulig å velge kun lunch på dag 1, både lunch og middag på dag 2 og ingen måltider på dag 3.

Systemet skal også kunne produsere faktura til de som deltar på konferansen. Arrangøren skal kunne sette ulike priser på hvert enkelt måltid. Samlet faktura blir dermed deltakeravgift pluss summen av måltidene man har krysset av for.

Datamodellen skal videre lages på en slik måte at SQL-spørringene beskrevet i oppgave 2 lar seg løse.

KANDIDATEN MÅ SELV KONTROLLERE AT OPPGAVESETTET ER FULLSTENDIG

Oppgave 2

I denne oppgaven skal du skrive SQL-spørringer mot databasen du laget datamodell for i oppgave 1. Hvis du har laget modellen med MySQL Workbench, så bør du vurdere å lage en database med «forward engineer», slik at du kan teste ut SQL-spørringene. Men hvis du ikke er trygg på hvordan man gjør dette, så er det kanskje bedre bare å skrive SQL-spørringer uten å teste. Pass også på at du ikke bruker for lang tid på å teste en enkelt spørring, kanskje det bare er en liten detalj som er feil.

Hvis du oppdager at datamodellen du laget i oppgave 1 ikke inneholder nok opplysninger eller er uegnet for å svare på en av SQL-oppgavene under, og du ikke har tid til å rette opp datamodellen, så kan du på et slikt spørsmål beskrive et tillegg eller en endring i datamodellen, f.eks. ved å skrive: «Jeg antar i dette spørsmålet at databasen inneholder følgende tabeller...». For full uttelling må slike tillegg gi en hensiktsmessig og normalisert database.

Tips: MySQL-dokumentasjonen har forklaring med eksempler til ulike funksjoner. På denne samlesiden er alle funksjonene ordnet etter datatype: <https://dev.mysql.com/doc/refman/8.0/en/functions.html> .

2-a (5 %)

Skriv en SQL-spørring som viser listen med presentasjoner i ett bestemt rom på én bestemt dag, f.eks. den første dagen av konferansen. Ta kun med klokkeslett og tittel på foredraget i utskriften, og sorter utskriften med hensyn på tittel.

2-b (5 %)

Skriv en SQL-spørring som viser antall deltakere som har vist interesse for hvert tema.

2-c (5 %)

Skriv en SQL-spørring som viser alle artikkeltitler som inneholder et bestemt ord (velg ordet selv), sammen med navn på alle forfattere av artikkelen. Vis både fornavn og etternavn til forfatterne i én kolonne adskilt av et blankt tegn.

2-d (5 %)

Skriv en SQL-spørring som flytter en eksisterende presentasjon til et annet rom med et annet starttidspunkt. Du kan altså forutsette at presentasjonen allerede er registrert. Velg nytt romnummer og nytt starttidspunkt selv.

2-e (5 %)

Lag en SQL-oppgave mot databasen du har laget som tester evne til å bruke delspørringer, og løs deretter denne oppgaven. Prøv å lage en oppgave som er naturlig og nyttig. Legg vekt på at du formulerer spørsmålet så presist som mulig. Krevende SQL-oppgaver vil gi mer uttelling enn enkle.

2-f (5 %)

Skriv SQL-spørringer for å registrere Peder Aas som deltaker. Han har epost `paas@gmail.com`, og ønsker lunch alle konferansedager, men middag kun på den siste dagen.

2-g (5%)

Skriv SQL-kode som oppretter et view som viser samlet fakturabeløp for hver deltaker. Ta med navnet på deltakeren i resultatet. For full uttelling må beløpet regnes ut fra prisen på enkeltmåltider og hvilke måltider som deltakeren har bestilt.

Oppgave 3

3-a (10 %)

Databasen fra oppgave 1 skal utvides med mulighet for deltakerne å vurdere hver enkel presentasjon med et tall fra 1 (dårlig) til 10 (bra). For å få til dette så blir det foreslått å legge til følgende tabell:

- Tilbakemelding(Dato, Klokkeslett, RomNr, Tittel, Tema, DNr, Fornavn, Etternavn, Vurdering)

Tittel er tittelen på presentasjonen, Tema er temaet for presentasjonen, mens Dato, Klokkeslett og RomNr forteller hvor og når presentasjonen ble holdt. DNr, Fornavn og Etternavn er henholdsvis deltakernummer og navnet til den som gjør vurderingen, mens Vurdering inneholder selve vurderingen (et tall mellom 1 og 10).

Forklar kort hva som er uheldig med den nye tabellen. Skriv deretter ned funksjonelle avhengigheter, bestem kandidatnøkkel og utfør normalisering til BCNF. Du skal ikke legge til flere kolonner. Vis primærnøkler med understreking og merk fremmednøkler med en stjerne i resultatet.

3-b (5 %)

Forklar hvorfor følgende to uttrykk betyr det samme:

- RomNr = 101 AND Vurdering > 5
- NOT (RomNr <> 101 OR Vurdering <= 5)

Vi tenker oss at uttrykkene blir brukt i en spørring mot tabellen gitt til oppgave 3-a.

3-c (5 %)

Forklar hensikten med å opprette primærnøkler og fremmednøkler.

3-d (5 %)

Informasjonen i en databasetabell lar seg representere i et Excel regneark. Man kunne tenke seg å lagre en database i et antall slike regnearkfiler. Hvilke fordeler og ulemper ser du ved å lagre informasjon i et databasesystem i forhold til regnearkfiler?

3-e (5 %)

Se på følgende spørring mot eksempeltabellen gitt til oppgave 3-a:

```
SELECT RomNr, Tema, AVG(Vurdering) AS X
FROM Tilbakemelding
GROUP BY RomNr, Tema
```

Forklar hva som blir resultatet av å utføre spørringen, sammen med en figur som viser konkrete eksempeldata i tabellen og spørreresultatet.