Eksamen DB1102 H2020

Hjemmeeksamen 1.desember - 2. desember 2020.

Teller 100% av karakteren i emnet. Vurdering A-F.

Alle hjelpemidler tillatt.

Når du leverer eksamen, skal du samle besvarelsen din i ett pdf-dokument. Dokumentet inneholder svar på alle oppgavene.

Oppgave 1 – modellering (20%)

Som et ledd i smittesporing ønsker et universitet å utvikle løsning for å holde rede på hvilke personer som er til stede i en fysisk økt på campus. I den anledning trenger de en database.

Du får i oppgave å modellere databasen. Dataene beskrives slik:

Vi trenger å lagre informasjon om personer i databasen. Vi trenger å lagre både navn og kontaktinformasjon (e-post og telefonnummer). Videre må vi ta vare på informasjon om at denne personen var til stede i en spesifikk fysisk økt. En fysisk økt kan for eksempel være en forelesning eller øving. En fysisk økt vil ha et starttidspunkt, et sluttidspunkt og vil gjelde ett spesifikt emne i ett spesifikt rom. Vi trenger å lagre både emnekode, emnenavn og hvor mange oppmeldte studenter det er i emnet. Vi ønsker å lagre en beskrivelse av rommet og gjeldende maksimale kapasitet (antall personer). Det er ikke nødvendig å skille på hvilke roller personene har i den fysiske økten. Hvis en student, veileder eller foreleser får Korona, så kan vi nå finne ut hvilke personer som har vært i samme fysiske rom som den smittede.

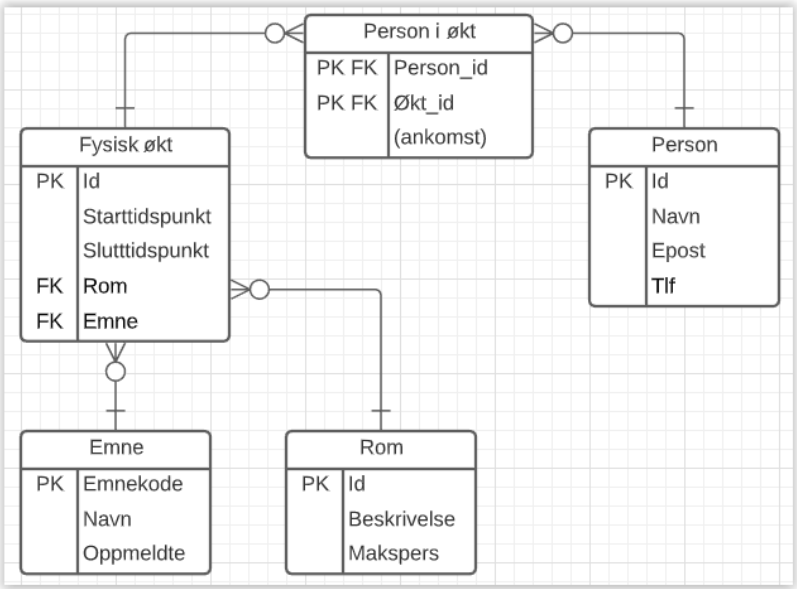
Oppgave: Tegn en modell for din foreslåtte løsning. Du kan selv velge om du vil benytte kråkefot eller UML notasjon. Velger du kråkefot trenger du ikke skille mellom identifiserende og ikke-identifiserende forhold. (UML notasjon har uansett ikke skille på dette.) Modellen din skal inneholde:

- Entitetene og deres attributter.
- Primærnøkler og fremmednøkler.
- Relasjonene mellom entitetene.
- Multiplisiteten (deltagelse og kardinalitet) for relasjonene.
- Hvis nødvendig, koblingsentiteter.

Hvis du synes noe er uklart, så gjør dine egne antagelser. Husk i så fall å gjøre rede for disse. Ja, når vi lagrer personopplysninger, så er det visse regler som kommer i spill, men det er ikke noe du trenger å ta hensyn til i denne oppgaven.

Vi anbefaler å benytte et program, som for eksempel Lucidchart, til å tegne modellen. Du kan også velge å tegne for hånd og lime inn bildet av tegningen din, men det kan da være vanskeligere for sensor å tyde den.

Løsningsforslag:



Oppgave 2 – SQL (50%)

I eksamensoppgaven vil du finne et SQL-script (konferanse.sql) som opprettet en database (konferanse) og fyller tabeller med data. Kjør SQL-scriptet mot din egen MySQL-server. Deretter løser du oppgavene som beskrevet nedenfor. Svaret på hver deloppgave er todelt:

1. SQL som løser oppgaven.
2. En skjermdump som viser resultatet ditt når SQL er kjørt.

Eksempel:

Hvis jeg hadde hatt en oppgave fra world-databasen: «Hent ut navn på alle land som begynner på bokstaven D», så ville et svar kunne sett slik ut:

SELECT name from country WHERE name LIKE 'D%';

name
Djibouti
Dominica
Denmark
Dominican Republic

VIKTIG! At studentene tar skjermdump er ment for å gjøre det lettere for sensor. Hvis noen ikke har med skjermdump som dokumentasjon av svar så skal det IKKE trekke ned.

Men du skal altså benytte **konferanse-databasen** som er vedlagt eksamensoppgaven. Husk at det er bedre å gjøre et forsøk på å besvare en oppgave enn å svare blankt.

a) Skriv en SQL som henter ut all informasjon om registrerte deltakere. Sorter resultatet basert på etternavn først, deretter fornavn.

```
SELECT * FROM Deltaker
 ORDER BY etternavn, fornavn;
```

	DNr	Fornavn	Etternavn	EPost
▶	15	Anders	Andersen	anders@andersen.no
	6	Benny	Ball	benny@benny.no
	10	Billy	Betong	billy@ppbb.no
	4	Eva	Dahl	eva@dahl.no
	12	Frida	Frosk	frida@ppbb.no
	1	Hans	Hansen	hans@hansen.no
	18	Svetlana	Iversen	svetlana@iversen.no
	8	Hans	Jensen	hj@jensen.no
	3	Jens	Jensen	jens@jensen.no
	16	Julie	Jensen	julie@jensen.no
	7	Oline	Jensen	o-j@jensen.no
	13	Leon	Latex	leon@ppbb.no
	2	Kari	Normann	kari@normann.no
	17	Igor	Olsen	igor@olsen.no
	5	Ole	Olsen	ole@olsen.no
	11	Pelle	Parafin	pelle@ppbb.no
	14	Ragna	Rekkverk	ragna@ppbb.no
	9	Sandra	Salamander	sandra@ppbb.no

OBS! Svetlana har opprinnelig en annen e-postadresse, men det er uviktig.

b) Skriv en SQL som henter ut fornavn og etternavn til registrerte deltakere som har en e-postadresse som ender med '@ppbb.no'.

```
SELECT fornavn, etternavn FROM Deltaker
 WHERE EPost LIKE '%ppbb.no';
```

	fornavn	etternavn
▶	Sandra	Salamander
	Billy	Betong
	Pelle	Parafin
	Frida	Frosk
	Leon	Latex
	Ragna	Rekkverk

c) Skriv en SQL som henter ut informasjon om hvor mye maten (alle måltider) koster til sammen på de ulike konferansedagene.

```
SELECT DagNR, SUM(MåltidPris) AS Totalsum
 FROM måltid
 GROUP BY DagNR;
```

	DagNR	Totalsum
▶	1	278
	2	278

d) Skriv en SQL som henter ut all informasjon om deltakere som ikke er forfattere på konferansen.

```
SELECT * FROM deltaker WHERE DNr NOT IN(SELECT DNr from forfatter);
```

	DNr	Fornavn	Etternavn	EPost
▶	15	Anders	Andersen	anders@andersen.no
	16	Julie	Jensen	julie@jensen.no
	17	Igor	Olsen	igor@olsen.no
	18	Svetlana	Iversen	svetlana@iversen.no

OBS! Svetlana har opprinnelig en annen e-postadresse, men det er uviktig.

e) Skriv en SQL som henter ut følgende informasjon om deltakerne: Fornavn, etternavn og hvor mange temaer de har markert som interessante. Gjør det slik at svaret listes ut med antall temaer i synkende rekkefølge.

```
SELECT fornavn, etternavn, (SELECT COUNT(*) FROM deltakertema WHERE DNr = d.DNr) AS
AntInteresser FROM Deltaker d
ORDER BY AntInteresser DESC;
```

eller:

```
SELECT d.Fornavn, d.Etternavn, COUNT(t.TemaNr) AS AntTemaer
FROM deltakertema dt
NATURAL RIGHT JOIN deltaker d
NATURAL LEFT JOIN tema t
GROUP BY d.Fornavn, d.Etternavn
ORDER BY AntTemaer DESC;
```

OBS! Et lite trekk hvis de ikke har med de om ikke har noen interesser (kan f.eks skyldes feil bruk av JOIN).

	fornavn	etternavn	AntInteresser
▶	Hans	Hansen	5
	Billy	Betong	5
	Oline	Jensen	4
	Ole	Olsen	3
	Hans	Jensen	3
	Pelle	Parafin	3
	Frida	Frosk	3
	Kari	Normann	2
	Eva	Dahl	2
	Leon	Latex	2
	Jens	Jensen	1
	Benny	Ball	1
	Sandra	Salamander	1
	Ragna	Rekkverk	0
	Anders	Andersen	0
	Julie	Jensen	0
	Igor	Olsen	0
	Svetlana	Iversen	0

f) Legg inn totalt 5 måltidsbestillinger.

```
INSERT INTO måltidbestilling
 VALUES (1, 'Lunsj', 1);
 INSERT INTO måltidbestilling
 VALUES (2, 'Middag', 2);
 INSERT INTO måltidbestilling
 VALUES (1, 'Middag', 1);
 INSERT INTO måltidbestilling
```

```
VALUES (1, 'Lunsj', 2);
INSERT INTO måltidbestilling
VALUES (1, 'Middag', 2);
```

g) Svetlana Iversen har blitt lagt inn med feil e-postadresse. Skriv en SQL som retter det opp slik at hun får riktig adresse: 'svetlana@iversen.no'.

```
UPDATE deltaker SET Epost = 'svetlana@iversen.no' WHERE Epost = 'svetlana@olsen.no';
```

h) Lag en spørring som henter alle etternavn som forekommer mer enn en gang blant deltakerne, og hvor mange ganger de forekommer.

```
SELECT etternavn, COUNT(*) AS Antall FROM deltaker
 GROUP BY etternavn
 HAVING COUNT(*)>1;
```

	etternavn	Antall
▶	Jensen	4
	Olsen	2

i) Konferansen ligger et stykke unna en flyplass, og det kan være hensiktsmessig å tilby transport til og fra flyplassen. Opprett en eller flere tabeller som holder informasjon om deltakere som ønsker transport til eller fra flyplassen. Velg kolonner som du selv mener er hensiktsmessig å benytte.
Legg deretter inn to rader i tabellen(e).

```
CREATE TABLE TransportFraFlyplass(
        Id INT AUTO_INCREMENT,
   Deltaker_id INT NOT NULL,
        FlightNummer VARCHAR(20),
   Tidsønske DATETIME NOT NULL,
        PRIMARY KEY (Id),
   FOREIGN KEY (Deltaker_id) REFERENCES Deltaker(DNr)
);

CREATE TABLE TransportTilFlyplass(
        Id INT AUTO_INCREMENT,
   Deltaker_id INT NOT NULL,
        FlightNummer VARCHAR(20),
   Tidsønske DATETIME NOT NULL,
        PRIMARY KEY (Id),
   FOREIGN KEY (Deltaker_id) REFERENCES Deltaker(DNr)
);
Men her kan vi forvente litt ulike valg, f.eks kun én tabell:
CREATE TABLE flyplassTransport(
   Id INT AUTO_INCREMENT,
   DNr INT NOT NULL,
   Tidsønske DATETIME NOT NULL,
   Reiseretning ENUM('Til flyplass','Fra flyplass') NOT NULL,
   FlightNummer VARCHAR(20),
   PRIMARY KEY (Id),
   FOREIGN KEY (DNr) REFERENCES Deltaker(DNr)
);
```

Husk at de blir bedt om å legge inn to rader i tabellen(e)…

```
INSERT INTO flyplassTransport (DNr, Tidsønske, Reiseretning) VALUES
(1, '2020-12-15 08:30', 'Fra flyplass'),
(1, '2020-12-16 16:15', 'Til flyplass');
```

j) (Vanskelig) Lag et view som inneholder informasjon om alle presentasjoner med tilhørende informasjon om rom, den som presenterer og tema.
Viewets kolonner skal se slik ut: RomNr, Tidspunkt (start og slutt), tittel på presentasjon, fornavn og etternavn på den som presenterer, temanavn og antall plasser i rommet.
Viewet skal være sortert på romnummer først, deretter tidspunkt.
Tidspunktet skal vises på et leselig format med dato først, og deretter tidspunkt (start og slutt).
Eksempel: En presentasjon som starter 2020-11-24 09:45:00 og varer i 20 minutter skal vises slik: '24. november kl 09.45-10.05'.
Fornavn og etternavn skal vises samlet.
Eksempel: 'Hans Hansen'.
Kjør deretter en spørring mot viewet slik at du kan vise (skjermdump) hvordan resultatet ble. Ja, oppgaven er vanskelig, men fokuser på å få med alt innholdet. Deretter kan du jobbe videre med formatteringen. Noen forslag til hendige funksjoner som kan være nyttige: CONCAT, DATE_FORMAT og DATE_ADD.

```
CREATE OR REPLACE VIEW PresentasjonInformasjon AS(
 SELECT r.RomNr, CONCAT(DATE_FORMAT(p.starttid, '%d. %M kl %h.%i') , '-' ,
DATE_FORMAT(DATE_ADD(p.startTid, INTERVAL p.varighet MINUTE),'%h.%i')) AS Tidspunkt,
p.tittel, CONCAT(d.fornavn, ' ' , d.etternavn) AS Presenterer, t.TemaNavn, r.AntPlasser
 FROM presentasjon p JOIN rom r ON p.RomNr = r.RomNr
 JOIN deltaker d ON p.DNr = d.DNr
 JOIN tema t ON p.TemaNr = t.TemaNr
 ORDER BY r.RomNr, p.starttid);
```

Resultat (SELECT *):

RomNr	Tidspunkt	tittel	Person	TemaNavn	AntPlasser
A1	24. November kl 09.45-10.05	Feasability of Optimizations Requiring Bounded …	Hans Hansen	Performance and Optimization	100
A1	24. November kl 10.15-10.35	Evaluation of graph algorithm frameworks for m…	Kari Normann	Performance and Optimization	100
F1	24. November kl 09.45-10.05	IT students perceptions of mandatory coursework	Jens Jensen	IT didactics	50
F1	24. November kl 10.15-10.35	Introducing ePortfolios to IT students: The sup…	Eva Dahl	IT didactics	50
F1	24. November kl 10.45-11.05	Teaching AI Ethics: Observations and Challenges	Ole Olsen	IT didactics	50
F1	24. November kl 11.15-11.35	The Live Programming Lecturing Technique: A S…	Benny Ball	IT didactics	50
F2	25. November kl 09.45-10.05	INERTIA AND CHANGE IN TRANSFORMATION …	Oline Jensen	Digital transformation	40
F2	25. November kl 10.15-10.35	DIGITAL TRANSFORMATION UNDER A PANDEM…	Hans Jensen	Digital transformation	40
F2	25. November kl 10.45-11.05	Exploring the Impact of Mob Programming on th…	Sandra Salamander	Digital transformation	40
F2	25. November kl 11.15-11.35	Exploring the Hiring Process of a Norwegian Mu…	Billy Betong	Digital transformation	40

Oppgave 3 – Normalisering (30%)

En bedrift har en database som holder oversikt over bedriftens utstyr (maskiner, stoler etc.), ansatte og hvor de ansatte har arbeidsplassen sin. Informasjonen er samlet i to tabeller og **utdrag av disse** er vist nedenfor. Kolonnenavn i uthevet (bold) er primærnøkler. AnsattNr i den første tabellen er fremmednøkkel til AnsattNr i den andre tabellen (markert med understrek).

Ansattnr	Fnavn	Enavn	Tlfnr	**Utstyr**	Innkjøpspris	**Innkjøpsdato**	Type
123456	Jens	Jensen	55555555	PC	10000	2019-01-02	Lenovo
123456	Jens	Jensen	55555555	Mobil	9000	2019-01-02	IPhone
123456	Jens	Jensen	55555555	Stol	3490	2018-12-12	Zareto

234567	Kari	Normann	66666666	Mac	13900	2017-05-05	MacBook Pro
234567	Kari	Normann	66666666	Mobil	9900	2019-05-05	Samsung
234567	Kari	Normann	66666666	Stol	3900	2017-07-05	Watford
234567	Kari	Normann	66666666	Headsett	2900	2017-08-05	Boss

Ansattnr	Fnavn	Enavn	Tlfnr	Rom	Sted	Etasje	Adresse
123456	Jens	Jensen	55555555	12	Oslo	1	Smalveien 1
234567	Kari	Normann	66666666	12	Oslo	1	Smalveien 1
345678	Ole	Olsen	77777777	22	Oslo	2	Smalveien 1
445544	Lise	Olsen	88888888	Gløtt	Bergen	5	Brygga 2
554455	Per	Persen	88668866	Gløtt	Bergen	5	Brygga 2
989898	Eva	Jensen	45454545	Regn	Bergen	5	Brygga 2
323232	Nils	Nilsen	23343223	Regn	Bergen	5	Brygga 2

Normaliser tabellene til 3. normalform. Gjør rede for egne antagelser om dataene der du trenger det. Du kan velge å introdusere nye kolonner om ønskelig.

Begrunn hvorfor løsningen din oppfyller kravene til 3. normalform.

Ansatt

Ansattnr(PK), Fornavn, Etternavn, Tlfnr, KontorID(FK)

Kontor

KontorId(PK), Rom, etasje, AvdelingId(FK) --- Ny kolonne KontorId

Avdeling

AvdelingId(PK), Sted, Adresse --- Ny kolonne AvdelingId

Utstyr

AnsattNr(PK,FK), Utstyr(PK), Innkjøpsdato(PK), Innkjøpspris, Type --- Kan fint introdusere ny id-kolonne her også.

Her kan vi forvente litt forskjellige svar (der flere kan være riktige).

Oppgave 4 Diverse (10%)

Du er ansatt i en bedrift som utvikler og drifter en applikasjon som benytter en relasjonsdatabase. Dere får inntrykk av at noen typer spørringer tar lang tid å utføre. Kom med noen tips for hvilke grep som kan utføres for å undersøke hvorfor problemet oppstår, og hvilke muligheter man har for å senke tidsbruken i databasen – særlig for spørringene som opptrer ofte.

Dette er en åpen oppgave der vi kan forvente mange typer svar. Basert på pensum i emnet er det særlig to momenter som er særlig relevante:

- Indekser
- Denormalisering

Et annet tema som kan dukke er optimaliseringer av selve spørringene (men det har vi ikke vært mye inne på).

Lykke til!