

Kont-eksamen DB1102 H2020

Hjemmeeksamen

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 (30%)

En venn av deg har planer om å starte en egen bedrift der han utfører hjelp til å stelle hager i området der han bor. Men det blir mye data å holde styr på, så han lurte på om du kunne modellere en database som kan holde på disse dataene. Slik beskriver han behovet:

Jeg trenger å holde styr på de ulike kundene mine – sånn typisk navn og telefonnummer. Med disse kundene kan jeg inngå en eller flere avtaler. Jeg vil gjerne vite når en avtale ble inngått og hvilken varighet den har. Jeg tenker at prisen vil forhandles for hver enkelt avtale, så jeg må huske å ta vare på hvilken pris vi ble enige om. Når jeg inngår en avtale beskriver jeg hvilke aktiviteter jeg skal utføre på hvilke lokasjoner. Kunden kan nemlig faktisk ha flere lokasjoner. Jeg kan for eksempel ha én kunde som ønsker at jeg skal gjøre hagearbeid både i hagen hans og i hagen til foreldrene som bor i nærheten. Tipper det er lurt å ta vare på en beskrivelse av lokasjonen i tillegg til adressen, selvfølgelig.

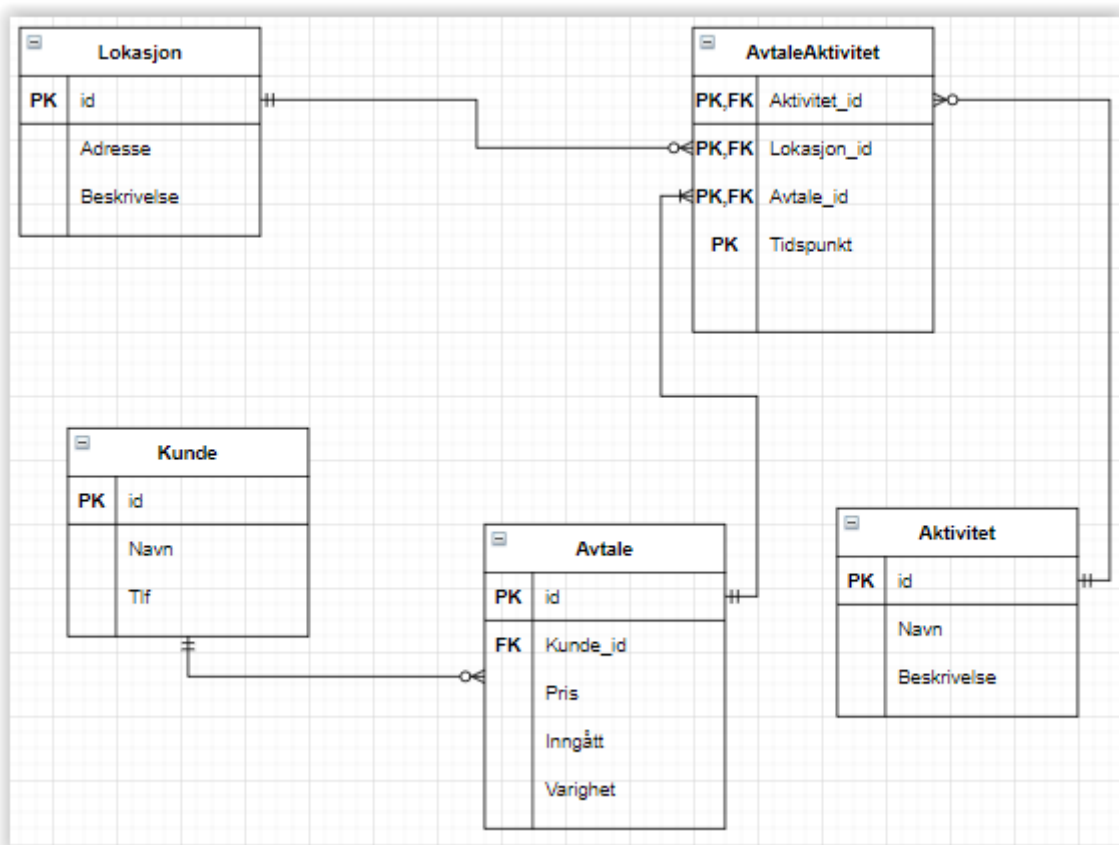
Det kan altså være ulike aktiviteter som jeg skal utføre på en bestemt lokasjon. Men jeg kommer til å ha et fast sett med aktiviteter som jeg tilbyr utført som for eksempel klippe gress, klippe hekk osv. Og jeg må huske å ta vare på hvilke tidspunkt jeg skal utføre de ulike aktivitetene. Tenk hvis jeg hadde glemt det! Jeg tror det også er lurt å ha en beskrivelse av de ulike aktivitetene jeg tilbyr, slik at det ikke blir uenigheter når arbeidet er utført.

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.

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.



Oppgave 2 – SQL (40%)

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

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

- a) Lag en SQL som henter ut all informasjon om alle stasjoner. Sorter resultatet på stasjonens navn.

SELECT * FROM stasjon
ORDER by navn;

SNr	Bemannet	Navn	Åpner	Tlf	Lukker
10	0	Alna	NULL	NULL	NULL
3	1	Asker	08:00:00	22222224	20:00:00
12	0	Fjellhamar	NULL	NULL	NULL
20	1	Grefsen	08:00:00	22222238	19:00:00
17	1	Greverud	08:00:00	67676767	16:00:00
5	1	Høvik	08:00:00	22222226	20:00:00
4	0	Hvalstad	NULL	NULL	NULL
24	1	Jaren	07:40:00	22222239	19:50:00
16	1	Kolbotn	08:00:00	22222236	20:00:00
14	1	Lillestrøm	06:00:00	22222235	22:00:00
11	1	Lørenskog	08:00:00	22222232	20:00:00
22	0	Movatn	NULL	NULL	NULL
8	1	Nationalt...	09:00:00	22222229	23:00:00
23	1	Nittedal	07:30:00	22222238	19:30:00
15	0	Nordstrand	NULL	NULL	NULL
9	1	Oslo S	07:00:00	22222230	20:00:00
2	1	Røyken	08:00:00	22222223	20:00:00
19	1	Ski	07:00:00	22222237	19:00:00
7	1	Skøyen	08:00:00	22222228	20:00:00
21	0	Snippen	NULL	NULL	NULL
1	1	Spikkestad	08:00:00	22222222	20:00:00
6	0	Stabekk	NULL	NULL	NULL
13	1	Strømmen	07:00:00	22222234	20:00:00
18	0	Vevelstad	NULL	NULL	NULL

OBS! Greverud har opprinnelig ikke bemanning...

- b) Lag en SQL som henter ut navn og åpningstider for alle stasjoner som er bemannet.

SELECT navn, Åpner, Lukker FROM Stasjon
WHERE Bemannet = true;

navn	Åpner	Lukker
Spikkestad	08:00:00	20:00:00
Røyken	08:00:00	20:00:00
Asker	08:00:00	20:00:00
Høvik	08:00:00	20:00:00
Skøyen	08:00:00	20:00:00
Nationaltheateret	09:00:00	23:00:00
Oslo S	07:00:00	20:00:00
Lørenskog	08:00:00	20:00:00
Strømmen	07:00:00	20:00:00
Lillestrøm	06:00:00	22:00:00
Kolbotn	08:00:00	20:00:00
Greverud	08:00:00	16:00:00
Ski	07:00:00	19:00:00
Grefsen	08:00:00	19:00:00
Nittedal	07:30:00	19:30:00
Jaren	07:40:00	19:50:00

OBS! Greverud er i utgangspunktet ubemannet...

- c) Vivian Eilertsen ønsker å reservere to billetter på TogRute L2 fra Oslo S til Greverud med avgang fra Oslo S kl 06.37.

Lag en SQL som legger inn reservasjonen.

```
INSERT INTO Reservasjon(Antall, KNr, AvgNr, Fra, Til) VALUES
(2,
(SELECT KNr FROM Kunde WHERE Fornavn = 'Vivian' AND Etternavn = 'Eilertsen'),
(SELECT AvgNr FROM Stopp WHERE Avreise = '06:37:00' AND SNr = 9),
9,
17);
```

Studenten kan selv velge om de vil benytte SELECT eller id direkte (slik det er en sammenblanding av ovenfor)

- d) Lag en SQL som henter ut informasjon om hvor mange **stoppesteder** det er på de ulike togrutene. Kall kolonnen med antall stoppesteder for AntStoppesteder.

```
SELECT COUNT(*) AS AntStoppesteder, RuteNr FROM Stoppested
GROUP BY RuteNr;
```

	AntStoppesteder	RuteNr
▶	14	L1
	9	L2
	6	L3

- e) Hent ut navn på alle stasjoner på stoppesteder på togruten 'L1'.

```
SELECT Navn FROM stasjon WHERE SNr IN(
SELECT SNr FROM Stoppested WHERE RuteNr = 'L1');
```

Evt

```
SELECT navn FROM stasjon NATURAL JOIN stoppested WHERE stoppested.RuteNr = 'L1';
```

	navn
▶	Spikkestad
	Røyken
	Asker
	Hvalstad
	Høvik
	Stabekk
	Skøyen
	Nationaltheateret
	Oslo S
	Alna
	Lørenskog
	Fjellhamar
	Strømmen
	Lillestrøm

- f) Gode nyheter! Greverud stasjon skal bli bemannet med åpningstider 8-16.

Lag en SQL som oppdaterer Greverud stasjon. De har telefonnummer 67676767.

```
UPDATE Stasjon SET Bemannet = true, Åpner = '08:00:00', Lukker = '16:00:00',
Tlf='67676767'
WHERE Navn = 'Greverud';
```

- g) Lag en SQL som henter ut informasjon om alle stoppesteder på rute L2. Kolonnene skal være: StoppNr, Stasjonsnavn og om stasjonen er bemannet.

Resultatet skal sorteres på StoppNr.

```
SELECT sto.StoppNr, sta.navn AS Stasjonsnavn, sta.bemannet
FROM stoppested sto JOIN stasjon sta
ON sto.SNr = sta.SNr
WHERE sto.RuteNr = 'L2'
ORDER BY StoppNr;
```

	StoppNr	Stasjonsnavn	bemannet
►	1	Stabekk	0
	2	Skøyen	1
	3	Nationaltheateret	1
	4	Oslo S	1
	5	Nordstrand	0
	6	Kolbotn	1
	7	Greverud	0
	8	Vevelstad	0
	9	Ski	1

Her er det ikke påkrevd å benytte ALIAS for stasjonsnavn da navn er likt nok. 0 eller 1 på Greverud (bemannet) vil variere etter om UPDATE i F) er kjørt eller ei.

- h) Lag en spørring som henter ut informasjon om alle unike etternavn i tabellen kunde, og hvor mange ganger de unike etternavnene forekommer. Sorter resultatet slik at det vanligste etternavnet kommer først, og alfabetisk hvis antallet er likt.

```
SELECT Etternavn, Count(*) AS Antall
FROM Kunde
GROUP BY Etternavn
ORDER BY Count(*) DESC, Etternavn;
```

Etternavn	Antall
Hansen	3
Olsen	2
Dahl	1
Eilertsen	1
Jensen	1
Larsen	1
Normann	1

- i) (Vanskelig) Lag et View (Reservasjonsinformasjon) som viser alle reservasjon med tilhørende informasjon om
Fornavn og etternavn på den som har reservert.
Navnet på stasjonene som reservasjonene gjelder (fra og til).
Om det er restaurant på avgangen.
Når toget går og ankommer.
Kolonnene skal være slik: Reservasjonsnummer, Fornavn, Etternavn, Rutenummer, Restaurant, Avgang, FraStasjon, Ankomst, TilStasjon.
I kolonnen Restaurant skal det stå 'Ja', hvis avgangen har restaurant om bord, og 'Nei' hvis den ikke har det. Hint: CASE.
For kolonnene som omhandler tidspunkt (avgang og ankomst) skal tiden vises i timer og minutter (ikke sekunder). Eksempel: 05:36. Hint: TIME_FORMAT.
CREATE OR REPLACE VIEW Reservasjonsinformasjon AS(

```

SELECT r.ResNr AS Reservasjonsnummer, k.fornavn, k.etternavn, a.RuteNr AS
Rutenummer,
CASE a.restaurant WHEN true THEN 'Ja' ELSE 'Nei' END AS Restaurant,
TIME_FORMAT(st1.avreise, '%H:%i') AS Avgang, s1.navn AS FraStasjon,
TIME_FORMAT(st2.ankomst, '%H:%i') AS Ankomst, s2.navn AS TilStasjon
FROM reservasjon r JOIN Kunde k ON k.KNr = r.KNr
JOIN Stasjon s1 ON r.fra = s1.SNr
JOIN Stasjon s2 ON r.til = s2.SNr
JOIN avgang a ON r.AvgNr = a.AvgNr
JOIN Stopp st1 ON st1.AvgNr = a.AvgNr AND st1.SNr = s1.SNr
JOIN Stopp st2 ON st2.AvgNr = a.AvgNr AND st2.SNr = s2.SNr);

```

	Reservasjonsnummer	fornavn	etternavn	Rutenummer	Restaurant	Avgang	FraStasjon	Ankomst	TilStasjon
▶	1	Hans	Hansen	L1	Nei	05:36	Asker	05:53	Høvik
	2	Kari	Normann	L1	Nei	06:41	Hvalstad	07:06	Nationaltheateret
	3	Jens	Hansen	L1	Nei	07:20	Røyken	08:18	Alna
	4	Kari	Olsen	L2	Nei	05:20	Stabekk	06:02	Vevelstad
	5	Hans	Hansen	L2	Nei	06:28	Skøyen	06:58	Greverud
	6	Oda	Hansen	L3	Ja	10:01	Oslo S	10:41	Nittedal
	7	Vivian	Eilertsen	L2	Nei	06:37	Oslo S	06:58	Greverud

Det er ikke påkrevd med ALIAS for Reservasjonsnummer og Rutenummer.

- j) Vi ønsker å legge til informasjon om avganger har toalett om bord, eller ikke.
Lag en SQL som endrer Avgang slik at tabellen også inneholder informasjon om toalett.
Hvis informasjon om toalett ikke blir registrert skal standard-verdien (default value) være
at det ikke har toalett.

```
ALTER TABLE Avgang ADD COLUMN Toalett BOOLEAN DEFAULT false;
```

Oppgave 3 – Normalisering (20%)

En bedrift har en database som holder oversikt over diverse informasjon. Informasjonen er samlet i en tabell og **utdrag av denne** er vist nedenfor:

Kundenr	Fnavn	Enavn	Tlfnr	Ordreld	Ordredato	ProduktId	PNavn	Pris	Antall
123456	Jens	Jensen	55555555	1	2019-05-05	12	X-box	5900	1
123456	Jens	Jensen	55555555	1	2019-05-05	13	IPhone	9900	1
123456	Jens	Jensen	55555555	1	2019-05-05	5	HDMI-kabel	149	2
234567	Kari	Normann	66666666	3	2019-05-06	12	X-box	5900	1
234568	Eva	Jensen	77777777	17	2019-05-06	13	IPhone	9900	1
234568	Eva	Jensen	77777777	17	2019-05-06	7	Skjerm	3900	1
123456	Jens	Jensen	55555555	4	2019-05-20	7	Skjerm	3900	1

Normaliser tabellen til 3. normalform. Gjør rede for egne antagelser om dataene der du trenger det.
Vis primærnøkler og fremmednøkler.

Begrunn hvorfor løsningen din oppfyller kravene til 3. normalform.

Kunde: Kundenr (PK), Fnavn, Enavn, Tlfnr

Ordre: Ordreld (PK), Ordredato, Kundenr (FK)

OrdreVare: ProduktId (PK, FK), Ordreld (FK), Antall

Produkt: Produkt_id (PK), PNavn, Pris

Oppgave 4 Diverse (10%)

Tabellen nedenfor inneholder noen rader med data.

Id	Fornavn	Etternavn	E-post
1	Per	Persen	per@persen.no
2	Per	Jensen	per@jensen.no
3	Per	Persen	p@persen.no
4	Eva	Jensen	eva@jensen.no

Basert på innholdet i tabellen beskriv:

- a) Supernøklerne.
- b) Kandidatnøklerne.

Super: Alle kombinasjoner av kolonner som inneholder Id eller E-post.

Kandidat: Id og E-post.

Lykke til!