

## Eksamen DB1102 H2023

Hjemmeeksamen 12.desember 2022 – kl 10-16 (6 timer)

Teller 100% av karakteren i emnet. Vurdering: Bestått / Ikke bestått.

Bestått: *Kandidaten har oppfylt kravene til læringsutbytte og viser nødvendige kunnskaper, ferdigheter og kompetanse.*

Det er tre hovedtemaer i emnet: SQL, modellering og normalisering. For å bestå eksamen må studenten vise oppnådd læringsutbytte innen alle de tre hovedtemaene.

Alle hjelpemidler tillatt. **Ikke tillatt å samarbeide med andre.**

Å presentere andres arbeid som sitt eget regnes som fusk. Husk derfor å vise til kilder som er brukt.

Studenter som forsettlig medvirker til fusk hos andre, anses også for å ha fusket.

Ved mistanke om fusk kan sensor bestemme at en student kan/skal få mulighet til å forklare seg i en samtale med sensor.

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

### Oppgave 1 – modellering (30%)

Du har en bekjent som er lidenskapelig opptatt av podkaster. Han har en drøm om å lage sin egen podkast-plattform, der han kan tilby et stort utvalg av podkaster i ulike kategorier, språk og formater. Han har allerede noen ideer om hvordan plattformen skal se ut og fungere, men han trenger din hjelp til å modellere databasen som skal lagre all informasjonen om podkastene, episodene, brukerne og deres interaksjoner. Han beskriver caset slik:

*Plattformen jeg ser for meg har mange podkaster i ulike kategorier, som kan spilles av eller lastes ned av brukere. Jeg tenker at en podkast bør ha et fengende navn, kort beskrivelse, en kategori og et språk. Og du, jeg tenker vi trenger utgiver og bilde også. Hver podkast kan selvfølgelig ha mange episoder, som har en tittel, beskrivelse, varighet, dato og lydfil. Jeg tenker det er lurt at vi også kan lagre en tekstlig representasjon av det som sies i episoden også. Vi får vel til det? Vi trenger naturlig nok brukere. Hver bruker har et navn, e-postadresse og passord. Det er vel nok? Hver bruker kan abonnere på flere podkaster, og gi vurderinger og kommentarer til episoder. Hver vurdering har en stjerne-verdi fra 1 til 5, og en tekstlig begrunnelse. Hver kommentar har en tekst og en dato. Kanskje vurdering trenger en dato også? Hmmm, nei det tenker jeg ikke er nødvendig.*

*Høres ikke det bra ut? Kan ikke du modellere dataene involvert her – please? Du kan få gratis abonnement for life!*

**Oppgave:** Tegn en ER-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 (50%)

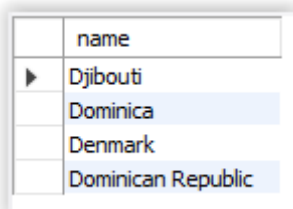
I eksamensoppgaven vil du finne et SQL-script (brettspill.sql) som opprettet en database (brettspill) og fyller tabeller med data. Kjør SQL-scriptet mot din egen MySQL-server. Deretter løser du oppgavene som beskrevet nedenfor. Svaret på hver deloppgave er todelt:

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

Men du skal altså benytte **brettspill-databasen** som er vedlagt eksamensoppgaven. Husk at det er bedre å gjøre et forsøk på å besvare en oppgave enn å svare blankt. Husk at denne oppgaven teller 50%, noe som utgjør ca 3 timer. Hvis du står veldig fast på en oppgave, så kan det være lurt å gå videre til neste.

- a) Skriv en SQL-spørring som viser navn og type på alle brettspillene i databasen.
- b) Skriv en SQL-spørring som viser navn og kallenavn på alle spillerne som har en epost-adresse som slutter på .no.
- c) Skriv en SQL-spørring som viser dato og sted for de tre siste spillkveldene som har funnet sted.
- d) Skriv en SQL-spørring som viser navn og type på alle brettspillene som har blitt spilt.
- e) Skriv en SQL-spørring som viser navn og kallenavn på alle spillerne som har vunnet minst ett brettspill, og hvor mange spill de har vunnet.
- f) Skriv en SQL-spørring som viser navn og kallenavn på den spilleren som har flest poeng i gjennomsnitt per spillrunde.
- g) Skriv en SQL som legger til en kolonne 'beskrivelse' i tabellen brettspill.
- h) Skriv en SQL som legger til et nytt brettspill i databasen. Du velger selv hvilke data som skal legges inn for brettspillet.
- i) Skriv en SQL som endrer e-postadressen til Billy Betong til 'billy.betong@ppbb.no'.
- j) Skriv en SQL som lager et view som heter 'topp\_spillere', som viser navn, kallenavn og antall vunnet spill for de tre spillerne som har vunnet flest spill. Viewet skal være oppdatert hver gang det blir spilt en ny spillrunde.

### Oppgave 3 – Normalisering (20%)

En organisasjon har en relasjonsdatabase med **én stor tabell** med 13 kolonner som inneholder informasjon om donasjoner til miljøvernsprosjekter. Et utdrag (6 rader) fra tabellen er nedenfor klippet i to (av plasshensyn), men det er en og samme tabell:

De første 6 kolonnene:

DID	OrgID	OrgNavn	Formål	Siiftet	ProsjektID
1	1	Naturverforbundet	Å bevare naturen og fremme bærekraftig utvikling	1914	1
2	1	Naturverforbundet	Å bevare naturen og fremme bærekraftig utvikling	1914	1
3	1	Naturverforbundet	Å bevare naturen og fremme bærekraftig utvikling	1914	2
4	2	WWF	Å bevare verdens natur og biologiske mangfold	1970	1
5	2	WWF	Å bevare verdens natur og biologiske mangfold	1970	2
6	1	Naturverforbundet	Å bevare naturen og fremme bærekraftig utvikling	1914	1

De siste 7 kolonnene:

PNavn	PBeskrivelse	Budsjett	DBeløp	GID	GNavn	GEpost
Redd regnskogen	Å stoppe avskoging og støtte urfolk i regnskogsområder	1000000	500	1	Ola Normann	ola@tmail.no
Redd regnskogen	Å stoppe avskoging og støtte urfolk i regnskogsområder	1000000	1000	2	Kari Normann	kari@tmail.no
Rent hav	Å redusere plastforurensing og overfiske i havet	500000	200	3	Per Olsen	per@tmail.no
Redd isbjørnen	Å bekjempe klimaendringer og beskytte isbjørnens habitat	800000	300	4	Lise Hansen	lise@tmail.no
Norsk urskog	Å bevare norsk urskog	100000	300	4	Lise Hansen	lise@tmail.no
Redd regnskogen	Å stoppe avskoging og støtte urfolk i regnskogsområder	1000000	400	1	Ola Normann	ola@tmail.no

**Altså:** Det er én stor tabell med 13 kolonner. Primærnøkkel i tabellen er **DID** (uthevet i tabellen over). De 13 kolonnene kan beskrives slik:

**DID:** Et unikt nummer som identifiserer hver donasjon til et miljøprosjekt.

**OrgID:** Organisasjons-id. Identifiserer en miljøorganisasjon.

**OrgNavn:** Navnet på miljøorganisasjonen.

Formål: Formålet til miljøorganisasjonen.

Siftet: Året miljøorganisasjonen ble stiftet.

ProsjektID: Identifikator for et miljøprosjekt. Unikt innad i en miljøorganisasjon.

PNavn: Navnet på miljøprosjektet.

PBeskrivelse: En kort beskrivelse av miljøprosjektet.

Budsjett: Det totale budsjettet for miljøprosjektet i norske kroner.

DBeløp: Beløpet som er donert til et miljøprosjekt i norske kroner.

GID: Identifikator for en giver som har donert til et eller flere miljøprosjekter.

GNavn: Navnet på giveren som har donert til et eller flere miljøprosjekter.

GEpost: Epost-adressen til giveren som har donert til et eller flere miljøprosjekter.

- a) Vis to tilfeller av redundans i tabellen.
- b) Forklar hvorfor tabellen ikke er på 3NF.
- c) Normaliser tabellen slik at vi oppnår 3NF. Husk å angi primær- og fremmednøkler.

**Lykke til!**