

Eksamen DB1102 H2022

Hjemmeeksamen 13.desember 2022 – kl 09-15 (6 timer)

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

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

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

Oppgave 1 – modellering (30%)

Noen venner av deg ønsker å ta opp kampen med Cutters og andre frisører. De vil starte en frisørsalong der kunden kan bestille time på nett. Frisørsalongen er tenkt å passe for de som ønsker å bestemme seg for frisyre før de ankommer salongen. De henvender seg til deg ettersom du har lært å modellere data. Slik beskriver de ideen sin:

Vår frisørsalong handler i stor grad om frisyre. Det finnes utrolig mange kule frisyre der ute som de aller fleste går glipp av. Vi ønsker å lage en nettside der kunden kan bla igjennom ulike frisyre å velge den de vil ha – før de kommer for å klippe seg. Derfor trenger vi å lagre informasjon om ulike frisyre vi tilbyr, hva frisyren kalles og hvor mye den koster. Og i tillegg trenger vi selvsagt en beskrivelse av frisyren, og et bilde.

Nå er det ikke slik at alle våre frisører kan utføre alle typer frisyre. Vi trenger derfor en oversikt over hvilke frisører som kan utføre hvilke frisyre. Tanken er da at når kunden har funnet drømmefrisyren sin, så kan de velge blant frisører som kan utføre den frisyren. Vi trenger også å ha informasjon om de ulike frisørene våre: Hva de heter, og hvilken bruker de har på Instagram.

Når kundene har valgt frisør og frisyre, så kan de booke en avtale. Avtalen må naturlig nok ha et tidspunkt, i tillegg til informasjon om hvilken frisør som skal utføre hvilken frisyre. Og det er jo fint å vite hvilken kunde som har gjort avtalen. Det er en fordel at kundene våre kan opprette en brukerkonto. Nei, forresten; de MÅ opprette en brukerkonto før de bestiller avtale. Vi må vite navn, adresse, e-postadresse og telefonnummer til brukeren. Kan du fikse en databasemodell for dette?

(Til info: Det finnes en datatype som kan lagre bilder.)

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 (50%)

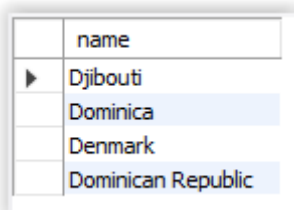
I eksamensoppgaven vil du finne et SQL-script (arcade.sql) som opprettet en database (arcade) og fyller tabeller med data. Kjør SQL-scriptet mot din egen MySQL-server. Deretter løser du oppgavene som beskrevet nedenfor. Svaret på hver deloppgave er todelt:

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 **arcade-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) Lag en spørring som henter ut informasjon om alle spillere (SpillerId, Dato og Kode). Sorter resultatet basert på Kode.
- b) Lag en spørring som henter ut all informasjon om spill lansert etter 1. januar 2020.
- c) Lag en spørring som henter ut informasjon om de ulike spill-kategoriene, og hvor mange spill som finnes i databasen innen hver kategori.
- d) Lag en spørring som henter ut informasjon (SpillerId, Dato og Kode) om den spilleren som har spilt flest ganger (har flest resultater), og hvor mange ganger spilleren har spilt. Altså: SpillerId, Dato, Kode, AntResultater.
- e) Lag en spørring som henter ut informasjon om alle spill (Tittel og Kategori) og gjennomsnittlig Score som er oppnådd i spillet.
- f) **(Vanskelig)** Lag en spørring som henter ut informasjon om alle spill (Tittel og Kategori), hvilken spiller (Kode) som har høyest Score i spillet, når rekorden (høyeste Score) ble satt, og hva rekorden er.
- g) Velg deg et spill, og lag et view for spillet som viser alle resultater for spillet i **synkende rekkefølge basert på Score**.
 - Viewet skal hete "Resultattavle_<Spillet navn>".
 - Viewet skal inneholde: Koden til spilleren, tidspunktet for resultatet og Score.
- h) Legg inn informasjon om det nye spillet: Ping Pong.
 - Spillet er i kategorien "Sports", ble lansert 8. september 2022 og trenger en fin beskrivelse som du kan finne på.
 - Legg også inn informasjon om hvem som har utviklet spillet: Jens Petrus var programmerer. Frida Frosk var designer. En ny utvikler som heter Billy Betong var manusforfatter. Billy kan kontaktes på billy.betong@boeljeband.no.

- i) Hvilke utviklere har vært med på å utvikle spill med til sammen flest resultater? Lag en spørring som henter ut:

- Navn (på spillutvikler)
- Ant_resultater (for antall resultater i spill hen har vært med å utvikle)

Resultatet skal sorteres synkende på flest resultater.

Du skal bare ha med utviklere som har flere enn 30 resultater til sammen på spillene sine.

- j) Det må gjøres en endring i spill-kategoriene. Action-kategorien er for omfattende. Det er et ønske om å erstatte Action-kategorien med 'Shooter', 'Platformer' og 'Fighter'.

- Gjør en endring slik at mulige kategorier for Spill skal være: 'Adventure', 'Shooter', 'Platformer', 'Fighter', 'RPG', 'Simulation', 'Sports', 'Puzzle', 'Annet'.
- Sørg for at allerede eksisterende Action-spill havner i rett kategori. Du kan selv bestemme om de er 'Shooter', 'Platformer' eller 'Fighter'.

Oppgave 3 – Normalisering (20%)

DVD-utleiefirmaet «Mot normalt» har en tabell med data som beskriver hvilke filmer som er lånt ut til hvilke kunder. Et utdrag av tabellen er vist nedenfor.

kundenummer	film_id	kundenavn	filmtittel
8	37	Morten Hanssen	Tatt av vinden
9	15	Lene Jenssen	Pretty woman
11	24	Hans Hanssen	Terminator 2
12	15	Andre Jenssen	Pretty woman
12	24	Andre Jenssen	Terminator 2
12	37	Andre Jenssen	Tatt av vinden

Tabellen har en sammensatt primærnøkkel (kundenummer, film_id). En kunde kan identifiseres ved sitt kundenummer, og har navn lik kundenavn. En film kan identifiseres ved sin film_id, og har tittel lik filmtittel.

- Vis to tilfeller av redundans i tabellen.
- Forklar hvorfor tabellen ikke er på 2NF.
- Normaliser tabellen slik at vi oppnår 2NF.

«Mot normalt» har også en tabell som holder informasjon om de ansatte. Et utdrag av tabellen er vist nedenfor.

ansattnummer	fornavn	etternavn	avdelingsnummer	avdelingssted
3	Per	Persson	3	Hovseter
5	Ole	Olsen	4	Drammen
8	Liv	Hanssen	3	Hovseter
9	Beate	Jensen	2	Fredrikstad

En ansatt kan identifiseres ved sitt ansattnummer som er primærnøkkel i tabellen. Tabellen holder informasjon om den ansattes fornavn og etternavn. Det registreres også hvilken avdeling den ansatte jobber i. Avdelingen har et unikt avdelingsnummer og avdelingsstedet forteller oss hvor i Norge avdelingen holder til.

- Forklar hvorfor tabellen ikke er på 3NF.
- Normaliser tabellen slik at vi oppnår 3NF.

Lykke til!