Eksamen DB1102 H2021

Hjemmeeksamen 7.desember - 8. desember 2021.

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%)

Robinsonekspedisjonen er endelig tilbake på TV-skjermen. Det begynner å bli så mange sesonger nå, at vi trenger en database som kan holde orden på alle episodene, deltakerne og noen hendelser som har skjedd. Hvis du ikke vet hva Robinsonekspedisjonen er, så er det et av de første realityprogrammene på norsk TV. Et kjapt Google-søk vil gi deg mer informasjon.

Vi må ha oversikt over alle sesongene; årstall og hvem som var programleder. Det har vært tre programledere i den norske versjonen (og det er den norske versjonen vi modellerer): Nils Ove Oftebro, Christer Falck og Silje Torp. For en sesong må vi vite hva premien var for å vinne. Vi må også kunne lagre en tekstlig, overordnet beskrive av sesongen som helhet. For programlederne ønsker vi å vite når de er født, og hva de heter.

Vi ønsker å lagre data om hver enkelt episode. En episode hører naturlig nok til en sesong. Vi ønsker å lagre premieredato for alle episodene. Vi trenger nok også å kunne lagre en beskrivelse av episoden.

La oss ikke glemme deltakerne. Vi ønsker å lagre navn, fødselsdato, bosted og yrke (da de var med i Robinson) og hvilken personlig eiendel de valgte å ha med på øya.

En episode har gjerne noen faste hendelser, blant annet:

-  En Robinsonkamp der deltakerne i kampen kan vinne immunitet i neste øyråd.

-  Øyråd: En avstemming der den med flest stemmer blir stemt ut av ekspedisjonen.

Robinsonkamp: Robinsonkampene er individuelle. Vi ønsker å lagre hvem som deltok i kampen, hvilken episode kampen foregikk og hvem som vant immunitet.

Øyråd: Vi ønsker å vite hvilke deltakere som mottok stemmer i Øyråd. Eks: Kåre Hansen 3 stemmer, Else Jensen 1 stemme og Balder Haukedal 4 stemmer. Vi har ikke mulighet til å lagre hvem som stemte på hvem, for det er som oftest ukjent. Vi må også vite i hvilken episode øyrådet ble avholdt.

En sesong avsluttes med en siste avstemming for å kåre en vinner. Noen av deltakerne stemmer på den de mener fortjener seieren, og den med flest stemmer vinner ekspedisjonen. Vi ønsker å ha oversikt over alle disse avsluttende avstemningene som har vært i Robinsonekspedisjonene; hvem som ga sin stemme til hvilken kandidat. For her kan du forutsette at vi kjenner til hvem som stemte på hvem. I siste sesong fikk en av deltakerne anledning til å gi to stemmer. Du må derfor ta høyde for at en deltaker kan gi flere stemmer i denne avstemmingen.

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.

Diagram

Description automatically generated**Løsning:**

PS:

1. Ved avstemming i tabell «Øyråd\_Stemmer» vil deltaker som oppnår flest stemmer resultere i utstemning. Vedkommende velger hvem som får immunitet i neste episode. Jeg har derfor valgt å legge til en Attributt; Utmeldt.
2. Ved sesongavslutning gjennomføres en sluttavstemning for å finne en vinner der alle deltagerne i sesongen kan avgi stemme. I tabellen «Sesong» Jeg har laget en attributt «Vinner» der deltagere og deres stemmer for deltagerkandidatene registreres. Den deltager som får mest stemmer (MAX) blir vinner.
3. «Premier» er en tekst som beskriver premien.
4. «Beskrivelse» er en tekst som beskriver henholdsvis episodene og sesongene.

Oppgave 2 – SQL (50%)

I eksamensoppgaven vil du finne et SQL-script (lotto.sql) som opprettet en database (lotto) 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%';*

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

1. a)  Lag en spørring som gir informasjon om hvilken av de registrerte kommunene som har størst innbyggertall, og hva denne kommunen heter.

Select Navn, Innbyggertall from Kommune

order by Innbyggertall desc;

A picture containing graphical user interface

Description automatically generated

1. b)  Lag en spørring som gir informasjon om hvilke registrerte spillere som bor i en kommune som heter Herøy.

SELECT spiller.Navn, Kommune.Navn from spiller

join Kommune on Spiller.KommuneID = Kommune.KommuneID

where Kommune.Navn like 'Herøy';

Graphical user interface, application

Description automatically generated

1. c)  Lag en spørring som gir navn på alle registrerte kommuner som har samme navn, men ligger i forskjellige fylker.

Select k.Navn, f.navn As Fylke from fylke f

join Kommune as k on f.FylkeID = k.Fylke

where k.navn like 'he%'

or k.navn like 'v%';

Graphical user interface, application

Description automatically generated

1. d)  Lag en spørring som viser hvor mange registrerte kommuner som har en liten 'u' i navnet sitt. Navngi kolonnen i svaret: Antall U-Kommuner.

select Navn from Kommune

where navn like '%u%';

Graphical user interface, text, application

Description automatically generated

1. e)  Lag en spørring som viser hvilket fylke som samlet sett har hatt flest premievinnere til nå. Resultatet skal vise fylket, og antall vinnere. Hvis noen har vunnet flere ganger, så skal de telles for hver gang de vinner.

SELECT \* FROM Fylke; #Navn #FylkeID

SELECT \* FROM Kommune; #Fylke #KommuneID

SELECT \* FROM Spiller; #KommuneID #SpillerNr

SELECT \* FROM Vinner; #SpillerNr #TrekkningID

select f.Navn, count(v.TrekningsID) as AntallVinner

from Vinner

Graphical user interface

Description automatically generatednatural right join Fylke f

natural left join Vinner v

group by f.navn

order by AntallVinner desc;

Kommentar: Jeg er ikke lykkes å koble tabellene siden det finnes ikke noe identifikatorer mellom fylke og spiller vinner.

1. f)  Lag en spørring som viser hvilke trekninger som ikke har hatt noen vinnere. Resultatet skal vise trekningens dato, og navnet på hvem som var trekningsansvarlig.

Select Trekning.TrekningsID, Trekning.Dato, Trekning.Utbetaling,

Trekning.AnsattID, Ansatt.Navn from Trekning

join Ansatt on Ansatt.AnsattID = Trekning.AnsattID

where Utbetaling = '0';

A screenshot of a computer screen

Description automatically generated with medium confidence

1. g)  Lag en spørring som viser navn på spillere har vunnet flere enn en gang, hvor mange ganger de har vunnet, og hvilken kommune de bor i.

select s.Navn, count(v.TrekningsID) as AntallVinner

from Vinner

natural right join Spiller s

A picture containing graphical user interface

Description automatically generatednatural left join Vinner v

group by s.navn

order by AntallVinner desc;

Kommentar: Jeg fikk ikke til attributtet « Kommune navn » det ble « error » hver gang.

1. h)  Legg inn en ny kolonne Areal i kommunetabellen. Legg inn fornuftige verdier i den nye kolonnen for de eksisterende kommunene. Velg datatype du selv mener er passende. Arealet skal oppgis i antall kvadratkilometer, med to desimaler.

ALTER TABLE `lotto`.`Kommune`

ADD COLUMN `Arial` FLOAT(8,2) NOT NULL AFTER `Navn`;

UPDATE `lotto`.`Kommune` SET `Arial` = '6970.1' WHERE (`KommuneID` = '0301');

UPDATE `lotto`.`Kommune` SET `Arial` = '147.87' WHERE (`KommuneID` = '1101');

UPDATE `lotto`.`Kommune` SET `Arial` = '1441.47' WHERE (`KommuneID` = '1103');

UPDATE `lotto`.`Kommune` SET `Arial` = '373.23' WHERE (`KommuneID` = '1106');

UPDATE `lotto`.`Kommune` SET `Arial` = '666.7' WHERE (`KommuneID` = '1507');

UPDATE `lotto`.`Kommune` SET `Arial` = '88.58' WHERE (`KommuneID` = '1515');

UPDATE `lotto`.`Kommune` SET `Arial` = '17.93' WHERE (`KommuneID` = '1818');

UPDATE `lotto`.`Kommune` SET `Arial` = '58.05' WHERE (`KommuneID` = '3018');

UPDATE `lotto`.`Kommune` SET `Arial` = '600.34' WHERE (`KommuneID` = '3020');

UPDATE `lotto`.`Kommune` SET `Arial` = '35.87' WHERE (`KommuneID` = '3419');

A screenshot of a computer

Description automatically generated with medium confidence

1. i)  Det har vært en ny trekning. Legg inn følgende informasjon i databasen: Trekningen ble avholdt 4. desember 2021. Det var nøyaktig 11 millioner i utbetaling. Det var en ny trekningsansvarlig: Jens Jensen, som bor i Oslo. Det var to vinnere som delte utbetalingen: Lars Andersen, som bor i Ålesund (Lilliveien 56) og Line Jensen som bor på Elverum (Blåklokkaleen 4).

Graphical user interface, text, application

Description automatically generated

Table

Description automatically generated

Background pattern

Description automatically generated

A picture containing table

Description automatically generated

1. j) (Vanskelig) Lag et view som viser hvilke fylker som har vunnet hvor mye penger. Viewet skal inneholde fylkets navn, og totale utbetalinger til fylkets spillere, sortert slik at fylket som har vunnet mest kommer først.

Graphical user interface, application

Description automatically generated

Kommentar: Jeg vet at jeg er på riktig spor men langt fra mål.

Oppgave 3 – Normalisering (20%)

En bedrift leier ut diverse verktøy og utstyr til privatmarkedet. De har en tabell i en database som

holder oversikt over utleien. Kolonnenavn i uthevet (bold) er primærnøkler. Et utdrag av tabellen

vises nedenfor:

Table

Description automatically generated

1. (5%) Slik tabellen fremstår, hvilke svakheter og begrensninger har denne tabellen?
2. Det er redundans i tabellen med redundant informasjon øker muligheten å for feil og vi kan få motstridene informasjon ved oppdatering (inkonsistent). 1

Dessuten bruker vi unødvendig lagring plass. 2

Tabellen kan sette på normalform og vil bruke marginalt mere tid i søkeprosessen, men det er insignifikant i forhold til de problematiske søk slik som tabellen nå er.

1. (10%) Normaliser tabellen til 3. normalform. Gjør rede for egne antagelser om dataene der du trenger det. Du kan velge å introdusere nye kolonner om ønskelig.

Table

Description automatically generated

C) (5%) Begrunn hvorfor løsningen din oppfyller kravene til 3. normalform.

Det fin ingen transitivet eller partielle avhengigheter i tabellene i figur 3 b) som vis ovenfor og tabellene måter derfor kravene til 3NF. Som vist har jeg laget 3 tabeller der kunde nummer og utstyr nummer er PK og FK. Jeg opprettet to tabeller siden det var to primærnøkler i tabellen (KundeNr og UtstyrID).

Jeg tok utgangspunkt i at det ikke finnes fler enn ett utstyr av hver UtstyrID. Det kan derfor ikke lånes ut mer enn ett UtstyrID av gangen.

PK er merket med bold skrift, FK er merket med understrek.

Kilder:

1. Database-systemer 5. utgaver Bjørn Kristoffersen side 48
2. Database-systemer 5. utgaver Bjørn Kristoffersen side 94
3. Database-systemer 5. utgaver Bjørn Kristoffersen side 242
4. <https://kristiania.instructure.com/courses/6545/files/722662?module_item_id=256268>