

Prøveeksamen IN2090 Høsten 2020

Tid: 4 timer

Tillatte hjelpemidler: Alle hjelpemidler tillatt

Prøveeksamen består av 3 deler (maksimal poengsum i parentes):

1. Modellering (40)
2. SQL (40)
3. Relasjonsmodellen og dekomponering (20)

1 Modellering (40)

Lag en ER-modell for databasen til et nettsted (kalt FileShare) som har som formål å gjøre det mulig for brukere å lagre, konvertere og dele filer.

- Du må gjerne dele modellen opp i deler (f.eks. over flere sider), men sørg for at det fremkommer tydelig hvilke deler av modellen som besvarer hvilke oppgaver.
- Du må gjerne inkludere kommentarer i modellene. Dersom det er uklarheter eller tvetydigheter i oppgavebeskrivelsen, bruk sunn fornuft og skriv en kommentar til modellene dine om hvilke antagelser og tolkninger du eventuelt gjør.
- Dersom du ser at det er mulig å modellere deler av oppgaven på flere ulike måter, diskuter kort hvordan din fremgangsmåte er i forhold til de ulike alternativene.
- Tegn og skriv tydelig.

1 Brukere (7)

Lag en ER-modell som fanger opp informasjon om brukere og brukergrupper på FileShare: En viktig funksjon for nettstedet FileShare er å kunne holde orden på informasjon om brukerne som er registrert på nettsiden. For enkelhets skyld identifiserer FileShare sine brukere internt med unike IDer (f.eks. 324). Informasjon som kreves av brukere av FileShare er e-postadresse, passord (kryptert, selvfølgelig), og tidspunktet da brukeren registrerte seg på FileShare. FileShare tillater ikke brukere å registrere mer enn én e-postadresse, og den samme adressen kan ikke benyttes av flere brukere. I tillegg kan brukere legge inn navn (bestående av for- og etternavn), stedet der brukeren bor, samt brukerens telefonnumre (potensielt fler nummere per bruker).

Brukere organiseres i brukergrupper som er tilknyttet organisasjoner. FileShare lagrer navnet på organisasjonen brukergruppe er tilknyttet, samt organisasjonens nettside. Organisasjoner har et unikt navn. Hver bruker er tilknyttet én eller fler organisasjoner, og en organisasjon kan ha mange brukere.

2 Filer og mapper (13)

Utvid modellen med informasjon om filer og mapper og hvordan de lagres på FileShare: En av kjernefunksjonene til FileShare er å lagre informasjon om filer lastet opp av brukere på FileShare. Hver bruker kan opprette én eller fler mapper, men en mappe er opprettet av nøyaktig én bruker. Hver mappe har en unik sti (f.eks. `/home/jon/xls/`). En mappe kan så inneholde potensielt mange filer.

Filer kan lastes opp av brukere, og blir lagret i filsystemet til FileShare. Filene identifiseres internt av en unik kombinasjon av mappen der filen er lagret i filsystemet og filens navn (f.eks. `lønn.xls`). FileShare vil lagre informasjon om hvilke brukere som lastet opp hvilke filer, samt når hver fil ble lastet opp til FileShare. I tillegg til navn og lokasjon må FileShare for filer også lagre informasjon om språket til filen. F.eks. `test.pdf` er et dokument skrevet på engelsk. For eksekverbare filer vil språket være programmeringsspråket filene ble skrevet i (f.eks. `run.exe` ble skrevet i C). En fil kan bli lastet opp av nøyaktig én bruker, men en bruker kan laste opp mange filer.

Brukere kan også endre filer, og FileShare ønsker å lagre når en bruker sist endret en fil (altså kun ett tidspunkt per kombinasjon av bruker og fil). Hver bruker kan endre mange filer, og hver fil kan endres av mange brukere.

3 Operasjoner (10)

Støtte for å utføre operasjoner på de lagrede filene er en annen viktig funksjon hos FileShare. En operasjon har et unikt navn (f.eks. `FlyttFil` eller `KjørFil`). Operasjoner krever rettigheter for å kunne kjøre. Men for å gjøre det enkelt har FileShare laget nøyaktig én rettighet per operasjon. Altså krever hver operasjon nøyaktig én rettigheter og en rettighet er påkrevd for nøyaktig én operasjon. Rettigheter er identifisert med en kode, og kan ha en eller fler navn, f.eks. `Administrator` eller `Eier`.

Brukere kan ha ingen eller fler rettigheter, og hver rettighet kan naturligvis være gitt til mange brukere.

Brukere kan utføre operasjoner på filer. F.eks. “Brukeren 324 utførte `KjørFil`-operasjonen på filen `run.exe` (i mappen `/home/jon/xls`)”. En bruker kan utføre mange operasjoner på samme fil; en operasjon kan utføres på mange filer av en bruker; og en operasjon kan utføres av mange brukere på samme fil. Ettersom opprettelsen av en fil også blir lagret som en operasjon vil alle filer være del av minst én operasjonsutførelse.

4 Realisering (10)

Realiser ER-modellen i figur 1 til et relasjonsdatabaseskjema. Relasjonsdatabaseskjemaet skal være korrekt (tilsvare ER-modellen), effektivt (unngå redundans og begrenset antallet tabeller), og tydelig (lett å forstå). Bruk realiseringsalgo-

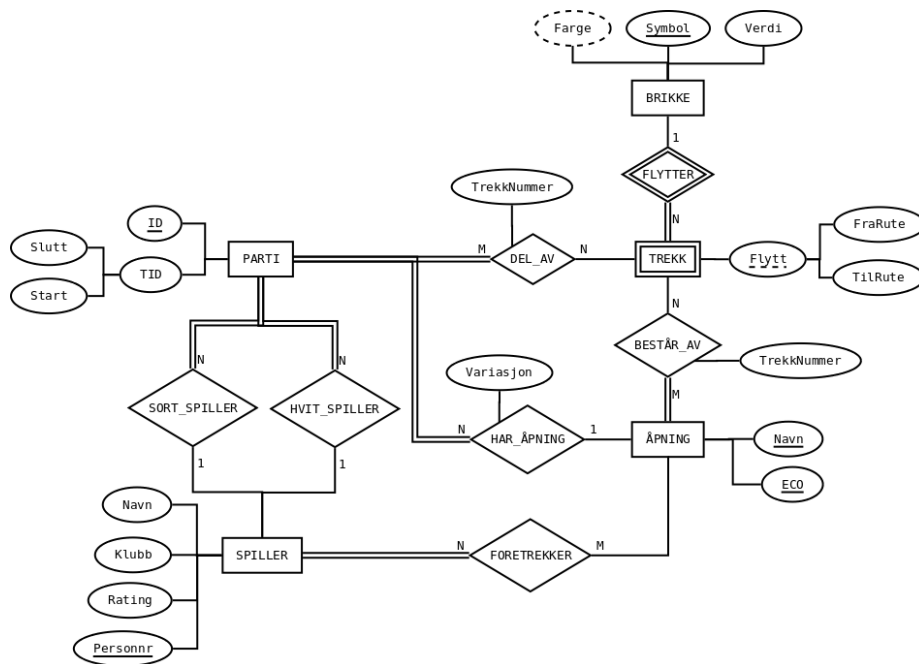


Figure 1: ER-modell

ritmen for å danne et slikt relasjonsdatabaseskjema. For hver relasjon, spesifiser relasjonens navn og navnet til hvert attributt. Du skal ikke spesifisere datatyper/domener for attributtene, og ikke benytte SQL i denne oppgaven. Marker primærnøkler med én understrek. Dersom en relasjon har flere kandidatnøkler, marker alle kandidatnøkler med én strek, og primærnøkkelen med fet skrift i tillegg. Skriv fremmednøkler enten med ord ($T(A)$ referer til $S(B)$) eller på formen $T(A) \rightarrow S(B)$ for å indikere at relasjon T sin A -attributt er en fremmednøkkel som peker på relasjon S sin B -attributt.

2 SQL (40)

Du er ansatt som ny databaseadministrator hos Julenissens Verksted, hvor det er mye som skal gjøres før jul. Databasen til Julenissen er laget av følgende SQL-script:

```
CREATE TABLE barn(
    bid int PRIMARY KEY,
    navn text NOT NULL,
    snill boolean NOT NULL
);
```

```

CREATE TABLE gave(
    gid int PRIMARY KEY,
    navn text NOT NULL,
    nyttig boolean NOT NULL
);

CREATE TABLE ønskeliste(
    barn int REFERENCES barn(bid),
    gave int REFERENCES gave(gid)
);

```

Tabellen `barn` inneholder informasjon om alle barn, hvor hvert barn har en unik ID (`bid`), et navn og en boolsk verdi som er sann om barnet har vært snill det siste året og usann hvis ikke.

Tabellen `gave` inneholder informasjon om gavene som Julenissens Verksted kan lage. Hver gave har en unik ID (`gid`), et navn samt en boolsk verdi som sier om gaven er en nyttig gave eller ikke.

Den siste tabellen, `ønskeliste`, inneholder alle barns ønsker, hvor hver rad inneholder et barns ID sammen med IDen til en gave det barnet ønsker seg. Merk: Et barn kan ønske seg mange gaver og en gave kan naturligvis bli ønsket av mange barn.

Under er et eksempel på hvordan databasen *kan* se ut.

barn

bid	navn	snill
0	Ola	t
1	Kari	t
2	Per	f
3	Nils	t
4	Mari	f
5	Kine	f
6	Jasmin	t

gave

gid	navn	nyttig
0	Lekebil	f
1	Sokker	t
2	Sykkel	t
3	Lekepistol	f
4	Dataspill	f
5	Fuglebrett	t

6		Dukke		f
7		Bok		t

ønskeliste

barn		gave
1		0
1		6
0		6
0		5
3		2
5		1
5		3
5		4
4		3
4		0
6		7

5 Snille barn (5)

Skriv en SQL-kommando som oppdaterer barnet med `bid` lik 0 sin `snill`-verdi til `false`.

6 Nyttige gaver (5)

Skriv en spørring som finner alle barn som ønsker seg nyttige gaver som har navn som starter med strengen `'Sokker'`. Spørringen skal returnere navnet på barnet og navnet på gaven.

7 Oversikts-VIEW (5)

Skriv en SQL-kommando som lager et `VIEW` som heter `oversikt` og inneholder én rad for hvert ønske med `bid` og navnet på barnet som har ønsket, `gid` og navnet på gaven som er ønsket, hvorvidt barnet har vært `snill`, og hvorvidt gaven er `nyttig`. `VIEWet` skal være sortert først på barnets navn, og så på gavens navn i alfabetisk rekkefølge.

Innholdet i `VIEWet` kan f.eks. se slik ut:

bid		barn		gid		gave		snill		nyttig
6		Jasmin		7		Bok		t		t
1		Kari		6		Dukke		t		f
1		Kari		0		Lekebil		t		f
5		Kine		4		Dataspill		f		f
5		Kine		3		Lekepistol		f		f

5		Kine		1		Sokker		f		t
4		Mari		0		Lekebil		f		f
4		Mari		3		Lekepistol		f		f
3		Nils		2		Sykkel		t		t
0		Ola		6		Dukke		t		f
0		Ola		5		Fuglebrett		t		t

8 Like ønkser (5)

Skriv en SQL-spørring som finner alle par av *ulike* barn som har ønsket seg det samme. Svaret skal kun inneholde unike rader. Du kan benytte VIEWet fra oppgave 7 om du ønsker.

9 Populære gaver (10)

Skriv en spørring som finner de tre mest populære nyttige gavene, og de tre mest populære unyttige gavene. Resultatet skal inneholde navn på gaven, antall barn som ønsker gaven, samt hvorvidt den er nyttig eller ikke. Du kan benytte view'et fra oppgave 7.

10 Gaveliste (10)

Skriv en spørring som tilordner gaver til barn i henhold til følgende regler:

- Snille barn får alt de ønsker seg
- Usnille får kun nyttige ting de ønsker seg. Om de ikke ønsker seg noen nyttige ting får de gaven med navn 'Genser'.

Spørringen skal skrive ut navnet på barnet og navnet på gaven. Merk, du kan bruke view'et fra oppgave 7.

3 Relasjonsmodellen og dekomponering (20)

11 Relasjonsalgebra (5)

I denne oppgaven skal du bruke samme databaseskjema som oppgavene i SQL.

Sriv et uttrykk i relasjonsalgebraen som finner navn på alle barn som ønsker seg hoppestokk.

12 Nøkler (5)

Gitt følgende relasjon

$R(A, B, C, D, E)$

med FDene

1. $A \rightarrow B$

2. $BC \rightarrow D$
3. $DE \rightarrow A$

Hvilke kandidatnøkler har R ? Vis hvordan du kommer frem til svaret.

13 Tapsfri dekomponering (10)

Gitt følgende relasjon

$R(A, B, C, D, E, F)$

med kandidatnøkler AB og CD , og FDer:

1. $AB \rightarrow C$
2. $AB \rightarrow D$
3. $CD \rightarrow A$
4. $CD \rightarrow B$
5. $A \rightarrow E$
6. $C \rightarrow F$

Dekomponer relasjonen tapsfritt til BCNF. Vis hvordan du kommer frem til svaret. For hver relasjon du får underveis i dekomponeringen, list opp hvilke FDer som holder, og hvilke kandidatnøkler relasjonen har.