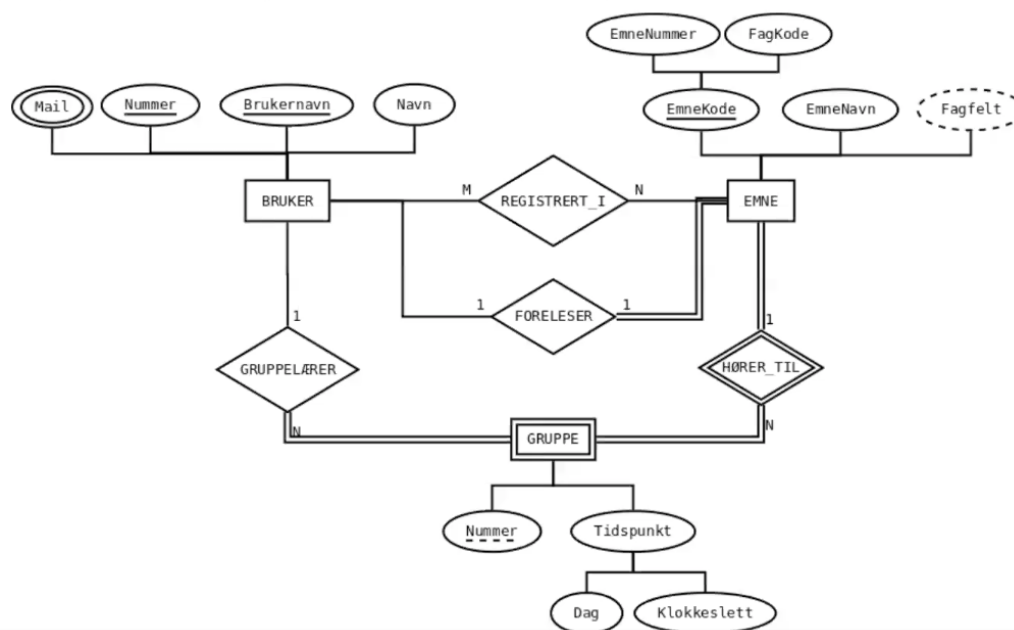


Realiser følgende ER-diagram til et databaseskjema.



Ved realisering så har vi en realiseringsalgoritme som følges når vi skal realisere et gitt ER-diagram til et databaseskjema. **I eksamen så er det utrolig viktig at man noterer seg valgene man gjør ved realisering av et gitt databaseskjema. Ellers kan man miste en del poeng så husk dette !!!**

Steg 1: Start med de enkel entitetene, noter deres attributter og eventuelle kandidat og primærnøkler.

Ved primærnøkkelen så skal man velge "1" av kandidatnøklerne så vi kan velge "Nummer". Kunne også ha valgt "Brukernavn", men kan være lurt å velge det mest fornuftige hvor i dette tilfellet så er "Nummer" mer unikt enn "Brukernavnet". Viktigste er å være "1" av kandidatnøklerne.

Bruker(Nummer, Brukernavn, Navn)

KN:{Nummer, Brukernavn}

PN: {Nummer}

Vi har at BRUKER-entiteten har en multi-verdi("Mail") men disse utsettes til senere som står i realiseringsalgoritmen siden det er egne regler for slike verdier. Vi kan realisere "EMNET"-entiteten som neste.

Før vi starter så er det viktig å notere at "EMNE" har en såkalt "Composite attributt". Dvs, en attributt som er avhengig av flere attributter hvor i dette tilfellet tilsvarer "Emnekode". Denne er da avhengig av "Emnenummer" og "Fagkode". Vi skal kun notere de attributtene som utgjør "Composite attributtet", så "Emnenummer" og "Fagkode" trengs å noteres, ikke "Emnekode". I tillegg skal ikke "Fagfelt" inkluderes ettersom den er utledbar og disse attributtene er ikke nødvendig å notere ettersom "Fagkode" hjelper oss med å identifisere et gitt fagfelt. I tillegg til at det blir en form for dobbeltlagring hvor vi lagrer flere ganger, noe som er unødvendig. Da kan vi starte å realisere denne relasjonen:

Emne(Emnenummer, Fagkode, Emnenavn)

KN: {Emnenummer, Fagkode}

PN: {Emnenummer, Fagkode}

Her utgjør "Emnenummer" og "Fagkode" kandidat- og primærnøkkelen vår siden disse to utgjorde unikheten for en gitt emne. Dette kan sees gjennom modellen over ettersom disse to attributtene utgjorde nøkkelen "Emnekode", så dermed er disse attributtene både kandidat- og primærnøkkel.

Steg 2: Realiser de svake entitetene, noter deres attributter og fremmednøkler.

Ved realisering av de svake entitetene så må vi huske at vi må inkludere de svake nøklene og nøkkelen til den gitte entiteten som gjør den svake identiteten "Identifiserbar" eller får den unike karakteristikken. Dermed må "GRUPPE"-entiteten ha primærnøkkel3n til "EMNE"-entiteten i sin relasjon. Dermed inkluderer vi primærnøkkelen til "EMNE" som tilsvarer da både {Emnenummer, Fagkode} og "GRUPPE"-entiteten inkludert.

Gruppe(Nummer, Dag, Klokkeslett, Emnenummer, Fagkode)

KN: {Emnenummer, Fagkode, Nummer}

PN: {Emnenummer, Fagkode, Nummer}

Vi må også nevne at "GRUPPE"-entiteten har fremmednøklene som er da {Emnenummer, Fagkode} så det gjør vi slikt:

Fremmednøkler:

Gruppe(Emnenummer, Fagkode) -> Emne(Emnenummer, Fagkode)

Steg 3: Realiser de ulike relasjonene som knytter to entiteter med hverandre

Først starter vi med å realisere de relasjonene som har 1:1(en til en) i kardinalitet. Deretter tar vi de som er 1:N(en til mange) og tilslutt N:M(Mange til mange) relasjonene.

Steg 3A: Realisere de relasjonene som har 1:1

Ved realisering av slike relasjoner så finnes det "3" valg når vi skal realisere de ulike relasjonene. Dermed starter vi å realisere "FORELESER"-relasjonen som knytter da BRUKER og EMNE. Bruker da valg 3.).

- 1.) Lag en ny realisering som representerer relasjonen og inkluderer kandidatnøkkelene fra begge entiteten som knyttes til den gitte relasjonen. Noterer også primær- og kandidatnøkkelene for den nye relasjonen.

Gjennom "FORELESER"-relasjonen så får vi da følgende:

Foreleser(Emnenummer, Fagkode, Nummer)

KN: {Emnenummer, Fagkode, Nummer}

PN: {Emnenummer, Fagkode, Nummer}

Det er ikke ideelt hvor ett av grunnene skyldes av at det blir vanskelig å lage spørringer.

- 2.) Inkluder kandidatnøkkelene til ett av entitetene i relasjonen.

Ta ett av relasjonene og inkluder kandidatnøkkelene for den gitte relasjonen så vi tar da BRUKER-relasjonen fra tidligere men gjør noen endringer:

Bruker(Nummer, Brukernavn, Navn,Emnenummer, Fagkode)

Vi bare inkludere kandidatnøkkelene til EMNE-relasjonen og setter den inn i BRUKER-relasjonen.

- 3.) Lag et attributt som representerer hele relasjonen og setter den inn i ett av entitetene.

I dette tilfellet, er det best å sette "Foreleser"-attributtet inn i EMNE-relasjonen siden den har "total utledning" i relasjonen så dermed er dette mest fornuftig ettersom EMNE-relasjonen må minst inkludere en "Foreleser" i relasjonen. Dermed får vi følgende:

Emne(Emnenummer, Fagkode, Emnenavn, Foreleser)

Ved å gjøre dette så må du sette fremmednøkkelen for det nye attributtet og sette det mot primærnøkkelen til det andre relasjonen. Så vi får da følgende:

Fremmednøkler

Emne(Foreleser) -> Bruker(Nummer)

Steg 3B: Realisere de relasjonene som har 1:N

Ved realisering av slike relasjoner så finnes det "2" valg når vi skal realisere de ulike relasjonene.

- 1.) Lag en ny realisering som representerer relasjonen og inkluderer kandidatnøkkelene fra begge entiteten som knyttes til den gitte relasjonen.

Dette er ikke ideelt hvor ett av grunnene skyldes av at det blir vanskelig å lage spørringer.

- 2.) Lag et attributt som representerer hele relasjonen og setter den inn i ett av entitetene.

Velger nummer 2.) og får da følgende:

Gruppe(Nummer, Dag, Klokkeslett, Emnenummer, Fagkode, Gruppelærer)

Fremmednøkler:

Gruppe(Gruppelærer) -> Bruker(Nummer)

Steg 3C: Realisere de relasjonene som har N:M

Ved realisering av slike relasjoner så finnes det "1" valg når vi skal realisere de ulike relasjonene.

- 1.) Lag en ny realisering som representerer relasjonen og inkluderer kandidatnøkkelene fra begge entiteten som knyttes til den gitte relasjonen.

Dette er greit ettersom vi kun har dette valget tilgjengelig. Ved kardinaliteter som ikke tilsvarer "N:M", men "1:M" eller "1:1", så brukes ellers ikke dette valget.

Siden vi kun har et valg, så bruker vi den og får da følgende:

Registrert_i(Nummer, Emnenummer, Fagkode)

KN: {Nummer, Emnenummer, Fagkode}

PN: {Nummer, Emnenummer, Fagkode}

Fremmednøkler:

Registrert_i(Nummer) -> Bruker(Nummer)

Registrert_i(Emnenummer, Fagkode) -> Emne(Emnenummer, Fagkode)

Steg 4: Realiser multiverdiene

Lager en ny relasjon som uttrykker multiverdiene, i dette tilfellet vil det være "Mail" og inkludere kandidatnøklerne som inkluderes for entiteten som "Mail" tilhører som er da "BRUKER". Dermed får vi følgende:

Mail(mail, nummer)

KN: {mail, nummer}

PN: {mail, nummer}

Fremmednøkler:

Mail(nummer) -> Bruker(nummer)

Slik ser hele realiseringen ut:

```
25 # Relasjoner
24
23 Bruker(nummer, brukernavn, navn)
22 -I KN: {nummer}, {brukernavn}
21 - PN: {nummer}
20 Emne(emnenummer, fagkode, emnenavn, foreleser)
19 - KN: {emnenummer, fagkode}
18 - PN: {emnenummer, fagkode}
17 Gruppe(nummer, emnenummer, fagkode, dag, klokkeslett, gruppelærer)
16 - KN: {nummer, emnenummer, fagkode}
15 - PN: {nummer, emnenummer, fagkode}
14 Registrert_i(nummer, emnenummer, fagkode)
13 - KN: {nummer, emnenummer, fagkode}
12 - PN: {nummer, emnenummer, fagkode}
11 Mail(nummer, mailadresse)
10 - KN: {nummer, mailadresse}
9 - PN: {nummer, mailadresse}
8
7 # Fremmednøkler
6
5 Gruppe(emnenummer, fagkode) -> Emne(emnenummer, fagkode)
4 Emne(foreleser) -> Bruker(nummer)
3 Gruppe(gruppelærer) -> Bruker(nummer)
2 Registrert_i(nummer) -> Bruker(nummer)
1 Registrert_i(emnenummer, fagkode) -> Emne(emnenummer, fagkode)
7 Mail(nummer) -> Bruker(nummer)
```