

Denne kolonne er
forbeholdt sensor.

^{eliminere}
① a) En database er ikke automatisk redundans, men det kan gjøres dersom du setter opp databasen "riktig", eller "godt" om du vil.

b) Fakt 2-normalform er dersom den er på 1NF (primærnøkkel) i tillegg til at det ingen attributter er som avhengig av deler ~~er~~ av en delbet- eller mer primærnøkkel.

Denne kolonne er forbeholdt sensor.

② E-R diagram

kommentar/antakelser:

* Det står ikke noe om det at leverandøren er enten kunsthåndlere eller personlige samlere skal lagres i databasen, eller noe spesifikt om kunden selv. Jeg velger derfor å samle disse, nemlig kunde og leverandør, i samme tabell ettersom det skal lagres akkurat like attributter. I tillegg kommer det tydelig frem at reklamen de sender ut er plagsom, antar jeg at de ikke er så opptatte av & hvem som får den, så dvs. om de sender den ut til både kunder og de de kjøper av.

* ~~Legger til en tabell med litt info~~


* Det er muligheter for å samle salgsforhandlings Tabellen og kjøpsforhandlings Tabellen til en tabell, men jeg velger å skille disse fra hverandre.

* Jeg har valgt å trekke kommentarer ut i en egen tabell. Det kommer frem i oppgaven at det kan være flere kommentarer på hver gjenstand.

* legger til attributter der det kreves for å oppnå kravet til ~~til~~ tabelten, f.eks. pris på hver enkelt del av en gjenstand da det var mulig å selegge enkelte deler.


* notasjon. Det finnes flere muligheter å sette opp et ER-diagram. Jeg har bestemt meg for følgende notasjon:

 1 og kun 1.

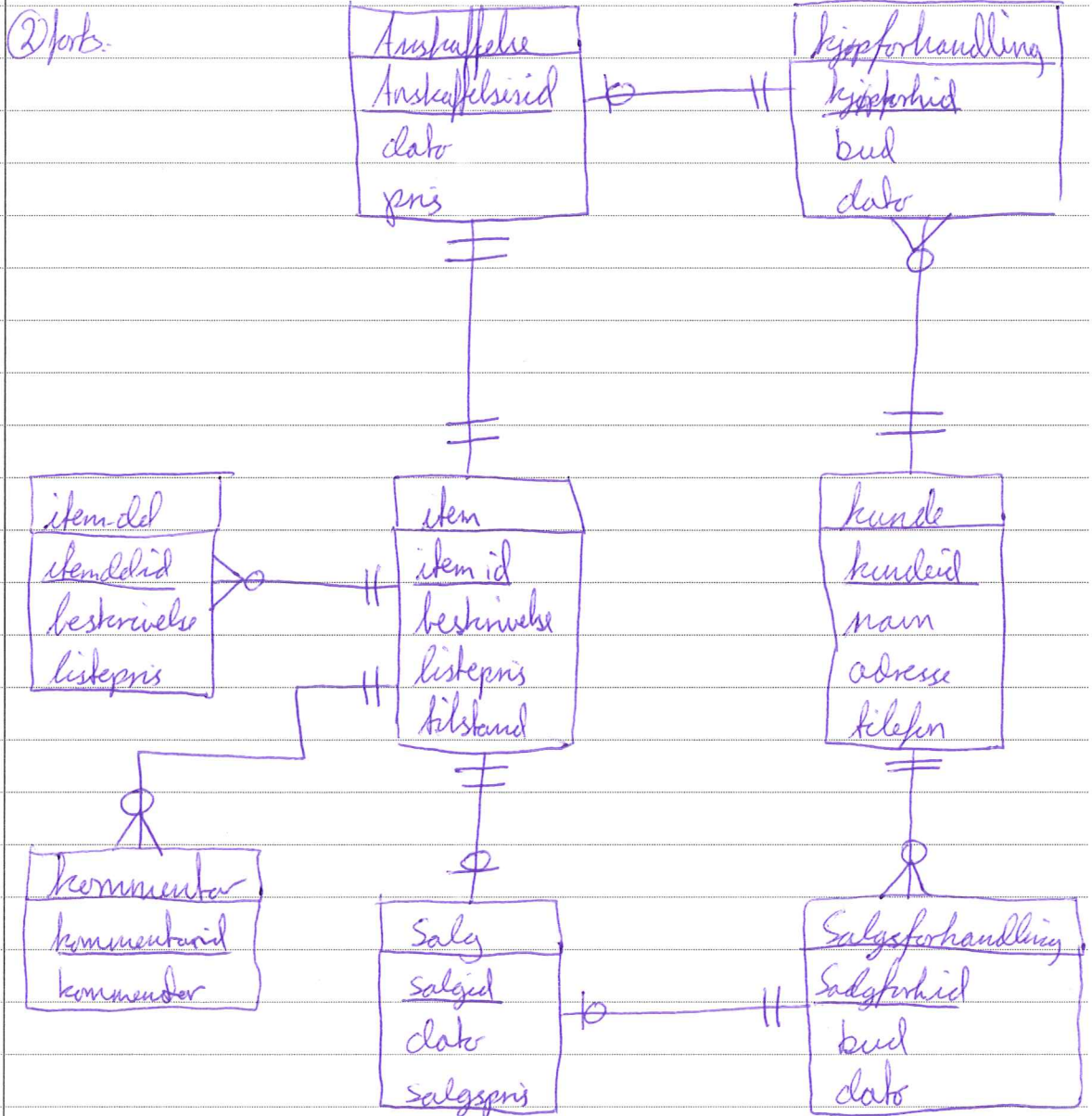
 0 eller flere

 0 eller 1.

primærnykkelen er understreket.

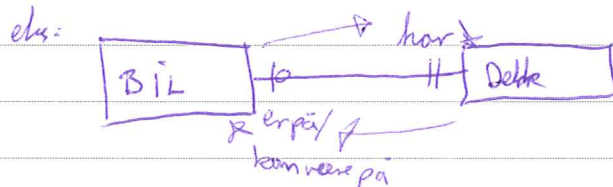
 1 eller flere

Denne kolonne er forbeholdt sensor.



MERKNAD:

En liten ekstra kommentar: det er forstjellig måte å tolke et ER-diagram på. Jeg har satt mitt diagram på følgende måte



Denne kolonne er
forbeholdt sensor.

③ Normalisering

kommentar/antakelser.

* Det er ingen optimal, unik attributt på nurse-nivået av det rapporten gir meg. Jeg regner med at hver sykepleier har et unikt ansattnummer, men jeg kan ikke garantere dette. Telefonnummer er ingen god primærnøkkel da dette kan endres ofte. Jeg har valgt fullt navn som primærnøkkel, men dette har også svakheter. Navn kan blant annet endres, og dersom to sykepleiere har likt navn har vi problemet.

* Telefonnummeret på linje med Ward og doctor-in-charge kan både være til Warden eller til doktoren, jeg har antatt at hver Ward/avdeling har et telefonnummer, og at det er dette som er listet her.

* primærnøkkel er understreket, og fremmednøkkel markeres med prikket understreik.

* jeg har ~~ikke~~ valgt å ikke dele adresse ipp, selv om det kan være en fordel til tiden.

Denne kolonne er
forbeholdt sensor.

Ø fortl

1NF:

Pasientrapport (Wardno, doctor-in-charge, wardphone,
nursename, nursephone, nurseaddress, patientno.,
lastname, firstname, home-city, visit-date, diagnosis,
referred-to)

2NF:

Ward (Wardno, doctor-in-charge, wardphone)

Nurse (nursename, phone, address, wardno)

Patient (patientno., lastname, firstname, homecity, visit-date,
diagnosis, referred-to, nursename).

3NF:

den forrige løsningen (2NF) er også på 3NF!

Denne kolonne er forbeholdt sensor.

④ a) Constraints er krav som tabellen har, Primær- og fremmednyklene er constraints, og disse må være definert for å kunne fylle inn i en tabell.

I selve koden til amazing.com er det enda ikke definert noen constraints, men de er derimot listet under koden hvor primærnyklene og fremmednyklene er listet opp.

For å ~~skape~~ de constraints som hører til, ordredetalj må 3 linjer legges til i koden. Jeg vil skrive koder for hele denne tabellen og markere de tre nødvendige linjene med *

```

Create table ordredetalj (
  ordrenr integer,
  produktnr integer,
  antall-besidd integer,
  * constraint ordredetalj_pk primary key(ordrenr, produktnr),
  * constraint ordredetalj_pkordr_fk foreign key(ordrenr) references
    ordre(ordrenr),
  * constraint ordredetalj-produkt_fk foreign key (produktnr)
    references produkt(produktnr)
);

```

Denne kolonne er forbeholdt sensor.

(4) b) Et database view er en "spesiell" spørring som ~~er~~ hvor det lagres svaret eller tabellen som du fikk, som om det var en egen tabell.

Views brukes blant annet hvor det er en komplisert spørring som vil bli brukt hyppigt. Ved å lage dette viewet kan man slippe å kjøre denne enkelte spørringen hver gang.

Jeg vil lage ett enkelt view som skal kun vise ~~alle~~ ordrenr på hver ordre samt for kundenummer på den som har bestilt ordren. Jeg velger å kalle viewet for kundeordrer.

```
CREATE VIEW kundeordrer AS  
SELECT ordrenr, kunde-navn FROM kunde INNER JOIN  
ordre ON kunde.kundnr = ordre.kundnr ;
```

Kurskode/Fag : IS-201Kandidatnr. : 2445Dato : 19.12.2008Ark nr. : 8 av 11Denne kolonne er
forbeholdt sensor.

④ c) Det finnes flere måter å løse dette på,
Jeg har valgt å gå frem på følgende måte:

1: - plukke ut alle kunder og ordrene de har (i tillegg
til de som ikke har ordre).

2: - luke ut, eller trekke fra, alle kunder med ordre.

```
SELECT kunder, kunde-navn FROM kunde LEFT JOIN  
ordre ON kunde.kundnr = ordre.kundnr;
```

MINUS

```
SELECT kunder, kunde-navn FROM kunde INNER JOIN  
ordre ON kunde.kundnr = ordre.kundnr;
```


Denne kolonne er
forbeholdt sensor.

④ d) Jeg går ut i fra at attributter kunde-nr-eier i tabellen kunde er som dersom kunden er bedrift ikke er datterselskap, og har et kundens dersom det er et datterselskap av det opplyste kundens nr kunde-nr-eier:

```
SELECT kundensr, kunde-nr-eier FROM kunde WHERE  
kunde-nr-eier != "#";
```

* Vid gi en tabell som liste opp kundensr (datterselskap) og kunde-nr-eier (moderselskapet) kun dersom kunde-nr-eier attributtet er forskj, altså at det er et datterselskap.
ikke

Denne kolonne er
forbeholdt sensor.

④ e) Her har vi en litt verre spørning. Jeg vil skrive koden så tydelig som mulig, men tanken er at jeg først lager en midlertidig tabell, og så kjøre jeg en spy spørning på denne tabellen igjen. Bruker også alias enkelte steder:

```
SELECT kundeNr, kunde_namn, SUM(antall_besitt) antall,
      SUM(ordresum) Totalsum FROM
```

```
SELECT kundeNr, kunde_namn, antall_besitt,
      (antall_besitt * enhetspris) ordresum FROM
```

```
((kunde INNER JOIN ordre ON kunde.kundeNr =
      ordre.kundeid)
```

```
INNER JOIN ordredetalj ON ordre.ordreNr =
      ordredetalj.ordreNr)
```

```
INNER JOIN produkt ON ordredetalj.produktNr =
      produkt.produktNr
```

```
GROUP BY kundeNr, kunde_namn ;
```

Denne kolonne er
 forbeholdt sensor.

4) f) Det spørres etter produkter det er bestilt mer
 enn 5 eksemplarer av. Jeg regner her med at det menes
 totalt bestilt fra alle kunder samlet, ikke per ordre, og regner
 videre med at attributtet bestillingsnivå gir meg dette tallet
 (~~hvor mange~~ hvor mange det er bestilt av den enkelte
 varen.)

Jeg henger ikke koble opp postnr-tabellen, og derfor vil jeg heller
 ikke gjøre dette. Postnr kan føies direkte fra kunde-
 tabellen.

På merere spørringen fra e) en smule, så jeg
 skriver opp hele koden og understreker det jeg legger til
 for f):

```

SELECT kundeNr, kunde_namn, SUM(antall_bestill) antall,
      SUM(ordresum) Totalsum FROM
  
```

```

(
SELECT kundeNr, kunde_namn, antall_bestill,
      (antall_bestill * enhetspris) ordresum FROM
      ((kunde INNER JOIN ordre ON kunde.kundeNr =
      ordre.kundeNr WHERE postnr >= 4600 AND
      postnr <= 4699)
  
```

```

      INNER JOIN ordredetalj ON ordre.ordreNr =
      ordredetalj.ordreNr)
  
```

```

      INNER JOIN produkt ON ordredetalj.produktNr =
      produkt.produktNr WHERE ant_bestillingsnivå > 5
  )
  
```

```

GROUP BY kundeNr, kunde_namn ;
  
```