

## Vejledende løsning for Sommerekksamen 12. juni 2003

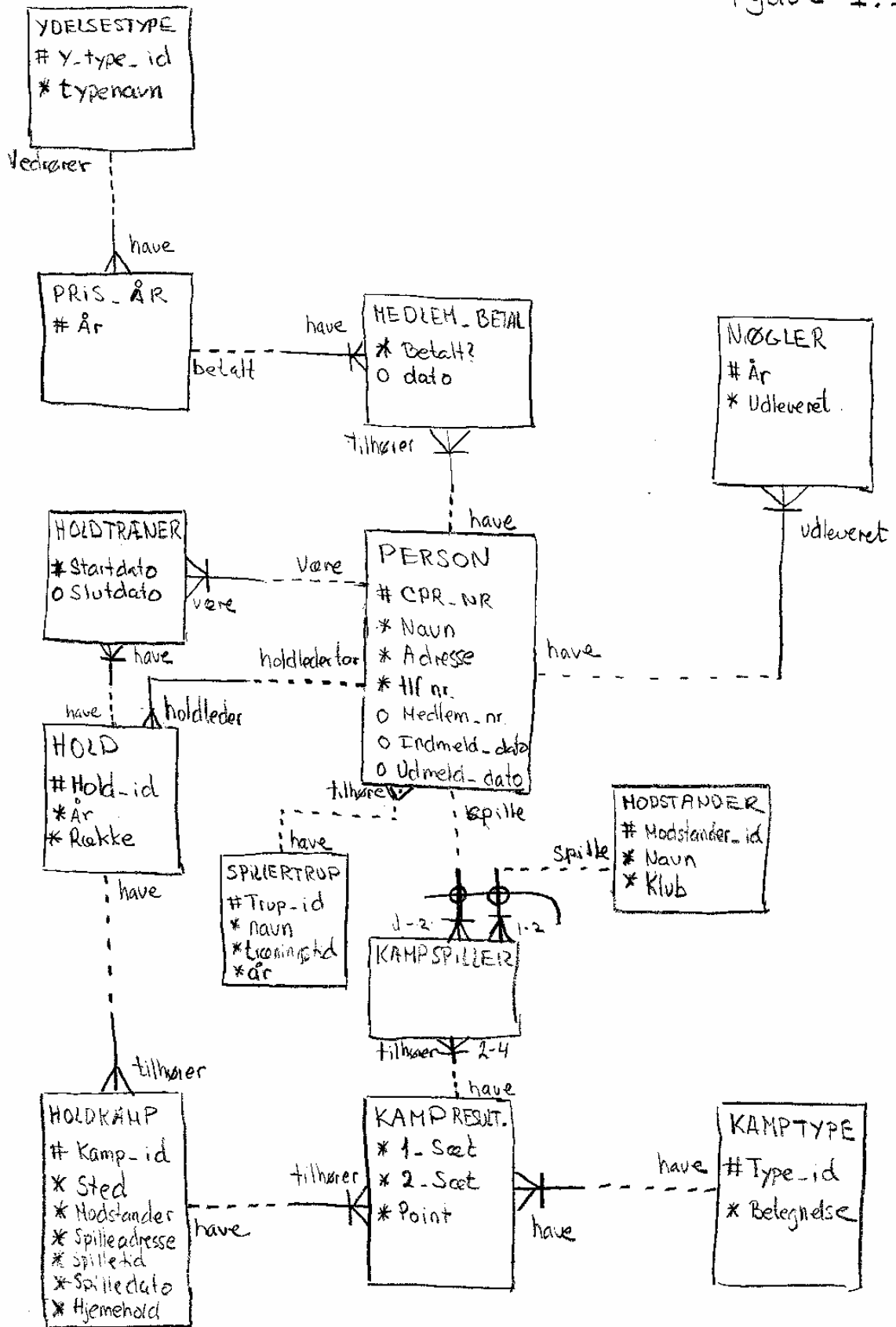
### Opgave 1.1

Der er ikke nogen entydig løsning. Løsningen afhænger af din fortolkning af casen, du bør derfor skrive din fortolkning ned i form af forudsætninger.

Min datamodel der bygger på nedenstående antagelser er vedlagt.

- Jeg forudsætter, at der er en række fælles data, der registreres omkring alle personer (medlemmer, trænere, holdledere m. f.), uanset hvilke roller de har i forhold til klubben. Jeg har derfor valgt at anvende en person entitet med disse oplysninger. Jeg kunne så have valgt at lave en sub-entitet for de data, som er specielle for de enkelte roller, men da den eneste sub-entitet jeg her har brug for er medlem og jeg går ud fra, at (næsten) alle der har en rolle i forhold til klubben også er medlemmer, har jeg alligevel valgt at tilføje medlemsdata som ”optional” attributter på personentiteten.
- Jeg går ud fra, at der udleveres nye nøgler hvert år og at nøglerne for det enkelte år er ens for alle medlemmer.
- For at kunne se, hvad det enkelte medlem skal betale, har jeg valgt at anvende en ydelsestype, som går igen fra år til år, men prisen for det enkelte år bestemmes af entiteten pris\_år. Betaling af medlemskontingent og holdtræning er forskellige ydelsestyper og det enkelte medlem kan således skulle betale flere ydelsestyper pr. år. Jeg forudsætter at det fulde beløb for en ydelsestype skal betales på en gang.
- Jeg forudsætter at man ønsker at kende spillernavnet på modstanderens spiller i de enkelte kampe. Jeg har derfor medtaget en modstander entitet. Hvis klubben kun ønsker at kende sine egne spillere i de enkelte kampe, så kan denne entitet blot tages ud.
- Jeg forudsætter, at der ikke oprettes hold, uden at holdet spiller i en række, derfor er denne attribut gjort obligatorisk.
- Jeg har medtaget en attribut kaldet spillertrup, idet det af casen fremgår, at holdspillerne formelt er opdelt i spillertruppe, som træner sammen. Da en spiller kan spille for mere end et hold, mener jeg ikke, at spillerne kan være tilknyttet noget hold direkte. Det er op til den enkelte holdleder at sikre, at spillerne til de enkelte kampe hentes fra den rigtige spillertrup, og at ingen regler vedrørende spil for flere hold brydes.

Opgave 1.1.



### Opgave 1.2

Også her kan der gøres en række forudsætninger, som vil kvalificere din løsning. Mit forslag er som følger:

- Personer fra Viben der ved noget om det administrative arbejde, eks. en kasserer eller en anden fra bestyrelsen.
- Personer fra Viben der ved meget om hold, rækker, kampe m.m. f.eks. en træner og en holdleder.
- Personer fra Sport IT, som skal udarbejde datamodellen og implementere databasen.

Det er vigtigt at repræsentanterne fra Viben og medarbejderne fra Sport IT samarbejder om udarbejdelsen af datamodellen for at sikre en hurtig og god informationsudveksling.

Ansvar for udarbejdelsen af datamodellen har medarbejderne fra Sport IT, repræsentanterne fra Viben deltager i arbejdet og godkender det endelige version af det konceptuelle design.

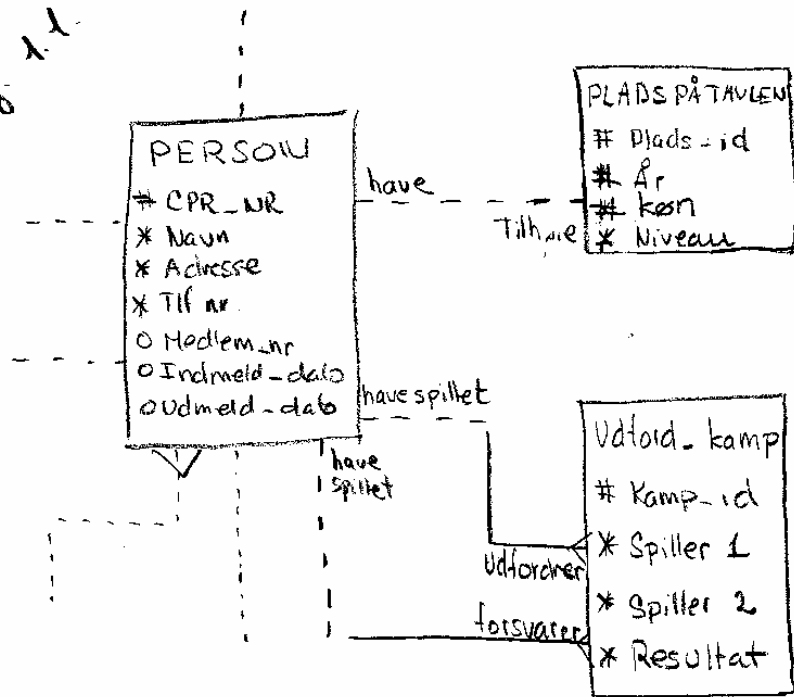
### Opgave 1.3

Min tilføjelse til datamodellen fra opgave 1.1. ses vedlagt. Den bygger på nedenstående forudsætninger.

- Jeg forudsætter, at der er en ansvarlig person, som opretter og nedlægger pladser på udfordringstavlen i forhold til hvor mange medlemmer der ønsker at være på tavlen. Hver plads på tavlen har en id og et niveau. På den måde bliver alle pladser entydige og det er samtidig muligt at afgøre, hvilke kampe, der formelt opfylder reglerne for udfordringskampe.
- Jeg forudsætter at aftaler om kampe og afviklingen af kampen foretages af de enkelte spillere uden at systemet "ved noget om det". Når kampen er afsluttet rapporteres hvem der har spillet og hvad resultatet er. Jeg forudsætter at der er en, der manuelt gennemgår de rapporterede resultater og opdaterer tavlen i forhold til de sidste nye kampe.

Opgave 1.3.

Eksisterende datamodel  
fra opg. 1.1.



## Opgave 2.1

- a)  $EF \rightarrow G$  og  $F \rightarrow H$
- b)  $EF \rightarrow G$  og  $G \rightarrow H$

## Opgave 2.2

- a) Det er ok, hvis du har svaret 3. normalform, eller du har argumenteret for Boyce/Cood normalformen.

## Opgave 2.3

- a) Det skal undersøges, hvilken funktionelle afhængigheder er for alle attributterne i de enkelte entiteter, og derefter skal der foretages normalisering så 5 normalform opfyldes.
- b) Nej, der kan være flere hensyn at tage. 3. normalform vil oftest være tilstrækkelig.

## Opgave 2.4

- a) Entiteten kampspiller har en flerværdiet afhængighed medlem  $\rightarrow$  kamp
- b) Flerværdie afhængigheder findes for at kunne opnå 4. og 5. normalform.

## Opgave 3.1

```
CREATE TABLE person (  
    Cpr_nr          CHAR (11)          NOT NULL  
    Navn           CHAR (30)          NOT NULL  
    Adresse        CHAR (30)          NOT NULL  
    Tlf.nr.        NUMBER (15)        NOT NULL  
    Medlem_nr      NUMBER (3)  
    Indmeld_dato   DATE  
    Udmeld_dato    DATE  
    Trup_id        CHAR (2)          NOT NULL
```

```
PRIMARY KEY (Cpr_nr),  
FOERIGN KEY (Trup_id) REFERENCE Spillertrup);
```

## Opgave 3.2

Jeg vælger entiteten Pladspåtavlen

```
CREATE TABLE Træner (
```

Cpr_nr	NUMBER (15)	NOT NULL
Start_dato	DATE	NOT NULL
Slut_dato	DATE	
Hold_id	Number (3)	NOT NULL

```
PRIMARY KEY (Cpr_nr, Start_dato, Hold_id, ),  
FOERIGN KEY (Cpr_nr) REFERENCE Person)  
FOERIGN KEY (Hold_id) REFERENCE hold);
```

## Opgave 3.3

- I tabellen Person har jeg valgt Cpr\_nr, jeg kunne alternativt have valgt medlem\_nr, idet det også forudsættes at være entydigt. Jeg har imidlertid valgt Cpr\_nr, idet jeg forudsætter at Vibens af hensyn til medlemstilskud m.m. fra kommunen skal kunne dokumentere entydigt, hvem der er medlem og derfor skal have oplysningen. Samtidig vurderer jeg, at et medlem har nemmere ved at huske/genfinde sit cpr.nr. end sit medlemsnr. Endelig vil det være muligt at oprette den samme person flere gange med forskellige medlemsnr., hvis det bruges som nøgle, mens et cpr.nr. absolut entydigt identificerer en person og dermed sikrer, at den enkelte kun er registreret en gang.
- En primærnøgle identificerer altid alle attributter i en tabel.

## Opgave 4.1

```
SELECT Spiller_navn, adresse, tlf.nr., fødselsdato  
FROM Medlem, Hold  
WHERE medlem.holdid = hold.holdid  
AND Holdnavn = 'puslingedrenge'  
ORDER BY spillernavn;
```

## Opgave 4.2

```
SELECT holdid, holdnavn  
FROM Hold, Holdkamp  
WHERE hold.holdid = holdkamp.holdid  
AND Modstander = 'Lyngby'  
AND sæson = 2002
```

## Opgave 4.3

```
SELECT Spiller_navn, adresse, tlf.nr, holdid
FROM Medlem
ORDER BY holdid, spiller_navn
```

## Opgave 4.4

```
SELECT Medlemsid, spiller_navn, kampid, modstander, mål_scoret, kampdato
FROM Holdkamp, Kampspiller, Medlem, Hold
WHERE Kampspiller.medlemsid = Medlem.medlemsid
AND holdkamp.kampid = kampspiller.kampid
AND Medlem.holdid = hold.holdid
AND Holdnavn = 'puslingedrenge'
ORDER BY medlemsid, kampdato
```

## Opgave 4.5

Jeg mener, at denne opgave kan tolkes på flere forskellige måder. Jeg har valgt at forstå det således, at det skal udregnes hvad det gennemsnitlige antal mål scoret af klubbens egne spillere på puslingehold er (modstanderens mål tæller ikke med).

```
SELECT kampid, AVG(mål_scoret)
FROM Kampspiller
WHERE kampid IN (SELECT holdid FROM hold, holdkamp
                 WHERE hold.holdid = holdkamp.holdid
                 AND holdnavn = 'puslingedrenge'
                 AND kampdato < 010103
                 AND kampdato >= 010102)
GROUP BY kampid
```

## Opgave 4.6

```
SELECT Medlemsid
FROM Kampspiller
WHERE mål_scoret >
      (SELECT AVG(mål_scoret) FROM Kampspiller WHERE kampid= '1234')
AND Kampid= '1234'
```

## Opgave 4.7

Denne opgave er lidt kompliceret at gennemskue og der var ingen der løste den korrekt ved sommereksamen. Det der er problemet er, at det er nødvendigt at finde den mængde af kampe, som tilhører et bestemt hold, og at afgøre hvilke hold en bestemt spiller er knyttet til. Opgaven kan kun løses ved at anvende alle 4 tabeller. Jeg har her løst opgaven ved først at finde det antal kampe, som et hold har spillet i perioden og derefter tælle det antal kampe en spiller har spillet i perioden. Da en spiller kun kan spille for et hold, tillader jeg

mig at konkludere, at hvis antallet af kampe et hold har spillet i perioden er det samme som det antal kampe en spiller har spillet i perioden, så har spilleren spillet alle holdets kampe.

```
CREATE VIEW Vholdkampe (holdid, antal) AS
SELECT holdid, COUNT(kampid)
FROM Holdkamp
WHERE kampdato>311202
AND kampdato >010603
GROUP BY holdid
```

```
CREATE VIEW Vspillerkampe (medlemsid, antal) AS
SELECT medlemsid, COUNT(kampid)
FROM Medlem, Kampspiller, Holdkamp
WHERE medlem.medlemsid = kampspiller.medlemsid
AND holdkamp.kampid = kampspiller.kampid
AND kampdato> 311202
AND kampdato<010603
GROUP BY medlemsid
```

```
SELECT medlemsid, spiller_navn
FROM Vholdkampe, Vspillerkampe, medlem
WHERE Vholdkampe.holdid = medlem.holdid
AND Vspillerkampe.medlemsid = medlem.medlemsid
AND Vholdkampe.antal = Vspillerkampe
```

#### Opgave 4.8

```
CREATE VIEW holdleder (medlemsid, spiller_navn, adresse, tdf.nr, fødselsdato, køn,
holdid) AS
SELECT medlemsid, spiller_navn, adresse, tlf.nr, fødselsdato, køn, holdid)
FROM medlem, hold
WHERE medlem.holdid = hold.holdid
AND holdleder=135769
```

#### Opgave 5.1

1. Giver ikke inkonsistens, da transaktion 1 aborteres uden at andre transaktioner har læst den værdi af (x), som den havde opdateret.
2. Vil give inkonsistens, da transaktion 3 læser den værdi af (x), som transaktion 2 har skrevet inden den aborteres.

#### Opgave 5.2

Her bør du illustrere svarene med grafer eller andet, som dokumenterer hvordan man er kommet frem til resultatet, her få I imidlertid blot svaret.

1. Kørselsafviklingen (transaktion 2, transaktion 1) giver de samme read-from og samme final w, så den er altså VSR.
2. Der er ikke nogen af de 6 mulige serielle kørselsafviklinger, som giver de samme read-from og de samme final w. Den er derfor ikke VSR

## Opgave 5.3

Her bør du også tegne konfliktgraferne, svarene er.

1. Er CSR, da der ikke er nogen cirkulærreference.
2. Er ikke CSR, da der er en cirkulærreference mellem t2 og t3.

## Opgave 5.4

T1	T2	T3	A	B	C	D
			Free	Free	Free	Free
Bot						
R lock (c)					R:1	
		Bot				
		R lock (d)				R:3
R lock (d)						R:1
		W lock (d)				3 venter
	Bot					
	R lock (a)		R:2			
	R lock (b)			R:2		
W-lock (c)					W:1	
Commit						
Un-ock (c)					Free	
Unlock (d)						W:3
		R lock (b)		R:3		
	W lock (a)		W:2			
	W-lock (b)			2 venter		
		W lock (b)		3 venter		

Transaktion 2 og 3 venter på hinanden.

## Opgave 5.5

Konfliktgrafen viser, at t2 og t3 venter gensidigt på hinanden. Der er altså en dead-lock.

## Opgave 5.6

Her tager jeg udgangspunkt i S1 som den står i opgaven.

```
<t1, start>
<t3, start>
<t3, D, old:1, new:1.15>
<t2, start>
<t1, C, old:2, new:2>          *OBS
<t1, commit>
<t2, A, old:200, new:150>
<t2, A, old:200, new:150>
<t2, B, old:500, new:550>
<t2, C, old:2, new:102>
<t2, commit>
<t3, B, old:500, new:600>    **OBS
<t3, B, old:500, new:600>
<t3, commit>
```

\*Transaktion t1 har læst værdien af D før den blev opdateret af transaktion t3, derfor bliver den nye værdi 2 og ikke 2,3, som ville have været korrekt, hvis denne opdatering blev taget i betragtning.

\*\* Også her er der læst en gammel værdi af B

## Opgave 5.7

UNDO-transaktioner: t3, t6

REDO-transaktioner: t2, t4, t5, t7