

Databasesystemer

IT Universitetet i København

8. juni 2006

Eksamenssættet består af 5 opgaver med 16 spørgsmål, fordelt på 10 sider (inklusive denne side), *samt et svarark, hvorpå visse spørgsmål skal besvares.*

Vægten af hver opgave er angivet. Du har 4 timer til at besvare alle spørgsmål. Hvis du ikke er i stand til at give et fuldt svar på et spørgsmål, så prøv at give et delvist svar. Du kan vælge at skrive på engelsk eller dansk (evt. med engelske termer).

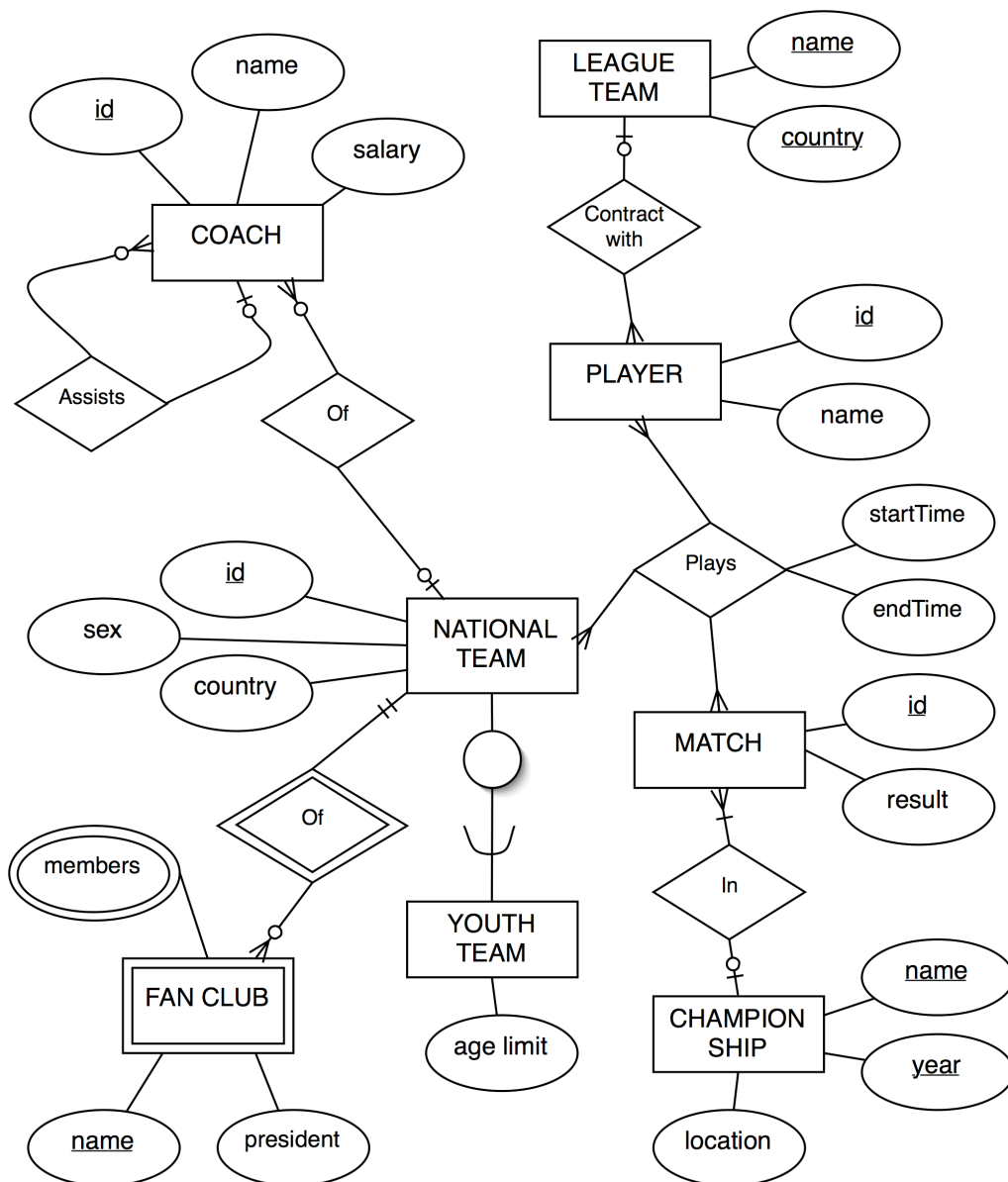
Siderne i besvarelsen skal være nummererede, og forsynet med navn, CPR nummer og kursuskode (DBS). Skriv kun på forsiden af arkene, og sortér dem inden nummereringen, så opgaverne forekommer i nummerrækkefølge.

“MDM” refererer i sættet til kursusbogen “Modern Database Management 7th edition” af Jeffery A. Hoffer, Mary B. Prescott and Fred R. McFadden.

Alle skriftlige hjælpemidler er tilladt.

1 Datamodellering (35%)

Betragt nedenstående EER diagram, der modellerer data om landsholdsfodbold: Trænere (COACH), fanklubber (FAN CLUB), kampe (MATCH), mesterskaber (CHAMPIONSHIP) og spillere (PLAYER). For trænere modelleres data om, hvem der assisterer hvem (Assists). For spillere modelleres data om, hvilket ligahold (LEAGUE TEAM) de har kontrakt med (Contract with). Visse landshold er ungdomslandshold (YOUTH TEAM). For fanklubber modelleres data om medlemmer, og hvem der er formand (president). For hver kamp modelleres data om hvilke spillere, som var med (Plays), og hvornår de blev skiftet ind og ud (startTime og endTime). Hvis hele kampen spilles, er værdierne af disse attributter henholdsvis 0 og 90.



a) Indikér for hvert af følgende udsagn hvorvidt det stemmer overens med EER diagrammet. (Bemærk at diagrammet ikke nødvendigvis er en eksakt modellering af virkeligheden.) Brug svararket til dine svar — sæt præcis ét X i hver søjle.

1. Et landshold har altid mindst 1 træner.
2. En trænerassistent kan selv have en assistent.
3. En spiller har højst kontrakt med 1 ligahold.
4. En spiller kan deltage i kampe for mere end 1 land.
5. En spiller kan blive skiftet ind flere gange i en kamp, og således have flere starttider.
6. Et ungdomshold kan spille med i et mesterskab.
7. Der kan være 20 spillere på banen for hvert hold i en kamp.
8. Der kan findes to fanklubber med samme navn.

Svar:

Spørgsmål 1.a	1	2	3	4	5	6	7	8
Stemmer med EER		X	X	X		X	X	X
Stemmer ikke med EER	X				X			

b) Konvertér EER diagrammet til relationer. I forbindelse med konvertering af 1-til-mange relationships skal du bruge den metode, der giver det mindste antal relationer. Din besvarelse skal angive skemaerne for de resulterende relationer, med attributterne i primærnøglen understreget.

Svar:

```

COACH(id, name, salary, assists, nationalteamId)
LEAGUETEAM(name, country)
PLAYER(id, name, contractwithName, contractwithCountry)
NATIONALTEAM(id, sex, country)
MATCH(id, result, championshipName, championshipYear)
FANCLUB(name, nationalteamId, president)
YOUTHTEAM(nationalteamId, agelimit)
CHAMPIONSHIP(name, year, location)
Plays(playerId, nationalteamId, matchId, startTime, endTime)
FanClubMembers(name, nationalteamId, member)

```

EER diagrammet modellerer ikke historiske data om spilleres karriere (hvilke hold, de har spillet for, i hvilke perioder, og til hvilken løn). Desuden vil en spillende træner svare

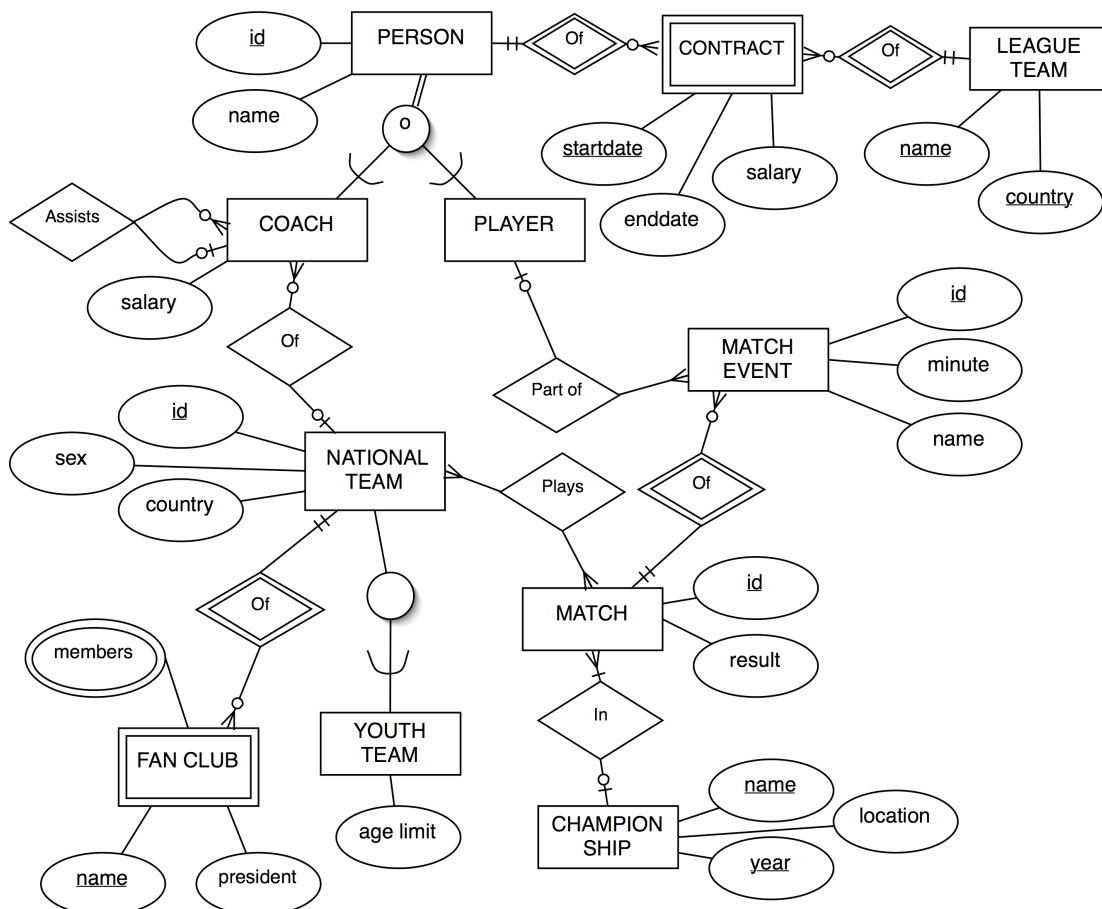
til en instans af COACH entiteten såvel som PLAYER entiteten, uden nogen information om, at der er tale om samme person. Der ønskes en ny datamodel, hvor disse restriktioner ikke gælder.

Desuden ønskes det, at datamodellen skal gøre det muligt ikke blot at registrere kampens resultat, men også de vigtigste hændelser i løbet af en kamp:

- Scoringer (hvem målscoreren er, og i hvilket minut scoringen er lavet).
- Straffespark (hvilket minut, hvem der begik straffesparket, og mod hvem).
- Advarsler og udvisninger (hvem og hvornår).
- Udskiftninger og indskiftninger – som i det nuværende EER diagram.

c) Lav en revideret EER model, der tager højde for de beskrevne ønsker. De dele af EER diagrammet, der ikke ændres, kan evt. udelades. Der lægges vægt på, at datamodellen let skal kunne tilpasses ønsker til mere detaljeret information. Supplér om nødvendigt dit diagram med forklarende tekst.

Svar:



Forklaring: Alle hændelser er abstraheret til ”match events”, inklusive ind- og udskiftninger af spillere. Et straffespark registreres som to hændelser (at det bliver begået, og at det bliver dømt). Det er valgt at lade en ”contract” være mellem en person og et ligahold, da dette muliggør at gemme information om træneres kontrakter med ligahold (og det ville ikke være en simplere løsning ikke at tillade dette).

2 Normalisering (15%)

Betragt en relation med skemaet: Varesalg(forhandler, producent, produkt, omsætning). Følgende er en gyldig instans af Varesalg:

forhandler	producent	produkt	omsætning
Silman	SoftFloor AG	Velour	101000
Bjarnes Tæpper	Bøgetæpper	Berber	207000
Top Tæpper	Bøgetæpper	Kashmir	77000
Silman	SoftFloor AG	Berber	72000
Bjarnes Tæpper	Bøgetæpper	Valnød	17000

a) Hvilke af følgende potentielle FDer gælder *ikke*, baseret på ovenstående instans?

1. omsætning → produkt
2. omsætning → produkt forhandler
3. produkt → producent
4. producent → produkt
5. forhandler produkt → omsætning

Hvis instansen ikke siger noget om en FD, markeres den som ”måske FD”. Brug svararket til dit svar — sæt præcis ét X i hver søjle..

Svar:

Spørgsmål 2.a	1	2	3	4	5
Ikke FD			X	X	
Måske FD	X	X			X

Bemærk, at instansen ovenfor kan fås ved ”natural join” af følgende relationer:

forhandler	producent	produkt	omsætning
Silman	SoftFloor AB	Velour	101000
Bjarnes Tæpper	Bøgetæpper	Berber	207000
Top Tæpper	Bøgetæpper	Kashmir	77000
Silman		Berber	72000
Bjarnes Tæpper		Valnød	17000

b) Angiv en funktionel afhængighed (FD), der sikrer at **Varesalg** kan opsplittes som i ovenstående eksempel uden tab af information. FDen skal med andre ord sikre, at SQL sætningen

```
(SELECT forhandler, producent FROM Varesalg) NATURAL JOIN
(SELECT forhandler, produkt, omsætning FROM Varesalg)
```

altid returnerer en relation, der er identisk med **Varesalg**. Forklar endvidere i ord, hvad FDen udtrykker.

Svar:

FDen **forhandler** → **producent** skal gælde. (FDen **forhandler** → **produkt omsætning** rækker også, men gælder ikke ifølge den viste instans.) FDen udtrykker, at enhver forhandler kun har produkter fra een producent.

c) Angiv en instans af **Varesalg**, hvor den viste opsplittning går galt, dvs. hvor SQL sætningen i spørgsmål b) *ikke* returnerer den samme instans.

Svar:

forhandler	producent	produkt	omsætning
Bjarnes Tæpper	Bøgetæpper	Berber	207000
Bjarnes Tæpper	SoftFloor AG	Velour	17000

3 SQL (30 %)

Betragt relationerne **fan(id,navn,cprnr,indmeldelse,favorit)** og **spiller(id,navn,land)**, og instanser med følgende data:

id	navn	cprnr	indmeldelse	favorit
1	Birger Hansen	1412861261	2000	5
2	Mads Mikkelsen	2605807413	1995	5
3	Jens Green	0909928475	2005	2
4	Hans Westergaard	1006701245	1980	1
5	Christian Lund	1102524895	1975	2
6	Jesper Andersen	1501661569	2000	3
7	Betina Jørgensen	1506751486	2005	5

id	navn	land
1	Peter Ijeh	Nigeria
2	Marcus Allbäck	Sverige
3	Martin Bernburg	Danmark
4	Jesper Christiansen	Danmark
5	Michael Gravgaard	Danmark

Relationerne indeholder data om medlemmerne i en fanklub, og deres favoritspillere.

a) Hvor mange tupler returneres der fra hver af følgende forespørgsler, hvis de køres på ovenstående instanser?

1. `SELECT * FROM fan WHERE indmeldelse = 2003;`
2. `SELECT * FROM fan WHERE indmeldelse >= 2000 AND favorit <> 5;`
3. `SELECT COUNT(*), indmeldelse FROM fan GROUP BY indmeldelse;`
4. `SELECT * FROM fan WHERE navn LIKE 'Hans%';`
5. `SELECT R1.navn, R2.navn FROM fan R1, fan R2
WHERE R1.favorit = R2.favorit and R1.id < R2.id;`
6. `SELECT navn FROM fan R1
WHERE (select count(*) FROM fan R2 WHERE R2.favorit=R1.favorit) > 1;`
7. `SELECT navn FROM fan WHERE favorit NOT IN
(SELECT id FROM spiller WHERE land='Danmark');`

Brug svararket til dine svar.

Svar:

Spørgsmål 3.a	1	2	3	4	5	6	7
Antal tupler	0	2	5	1	4	5	3

b) Skriv en SQL-kommando der, for alle tupler i relationen `fan` hvor `cprnr` er større end 3112999999 eller mindre end 0101000000, sætter `cprnr` til værdien `NULL`.

Svar:

```
UPDATE fan SET cprnr = NULL WHERE cprnr > 3112999999 OR cprnr < 0101000000;
```

c) Skriv en SQL-kommando der sletter alle tupler i `fan` hvor `cprnr` har værdien `NULL`.

Svar:

```
DELETE fan WHERE cprnr = NULL;
```

d) Skriv en SQL forespørgsel der for hver medlem i fanklubben viser medlemmets navn (`fan`) og navn på medlemmets favorit (`spiller`).

Svar:

```
SELECT fan.navn,spiller.navn FROM fan,spiller WHERE fan.favorit=spiller.id;
```

e) Skriv en SQL forespørgsel der beregner gennemsnittet af kolonnen `indmeldelse` i relationen `fan`.

Svar:

```
SELECT AVG(indmeldelse) FROM fan;
```

f) Definér i SQL et view der for hver medlem i fanklubben viser medlemmets navn (`fan`) og navn på medlemmets favorit (`spiller`). Brug dit view til at udregne hvor mange fans de forskellige spillere har (spillerens navn skal fremgå).

Svar:

```
CREATE VIEW MyView AS
SELECT fan.navn AS fannavn, spiller.navn AS spillernavn
FROM fan,spiller
WHERE fan.favorit=spiller.id;
```

(Omdøbningen er nødvendigt i Oracle for at undgå at flere attributer har samme navn. En besvarelse uden omdøbning er også korrekt.) Brug af viewet:

```
SELECT spillernavn,COUNT(*) FROM MyView GROUP BY spillernavn;
```

g) Skriv en SQL forespørgsel der returnerer en relation med *én* attribut indeholdende alle navne i `fan` og `spiller`. Du kan antage, at datatyperne for `navn` attributterne er identiske.

Svar:

```
(SELECT navn FROM spiller) UNION ALL (SELECT navn FROM fan);
```

h) Skriv en SQL forespørgsel der returnerer navnene på alle spillere, der har flere fans blandt kvindelige end blandt mandlige fans. En person i `fan` er mand hvis udtrykket `cprnr % 2 = 1` er sandt, og kvinde hvis `cprnr % 2 = 0`.

Svar:

```
SELECT navn FROM spiller s WHERE
(SELECT COUNT(*) FROM fan f WHERE f.favorit = s.id AND cprnr % 2 = 1) <
(SELECT COUNT(*) FROM fan f WHERE f.favorit = s.id AND cprnr % 2 = 0);
```

4 Transaktioner (10 %)

Betragt to databaseforbindelser, der laver opdateringer og forespørgsler på relationen `MyFan(id, navn)`:

Forbindelse 1	Forbindelse 2
<pre>INSERT INTO MyFan VALUES (3,'Bent Ølgård'); INSERT INTO MyFan VALUES (5,'Birger Hansen'); COMMIT; SELECT * FROM MyFan; (2)</pre>	<pre>INSERT INTO MyFan VALUES (7,'Birger Hansen'); SELECT * FROM MyFan; (1) DELETE FROM MyFan; ROLLBACK; SELECT * FROM MyFan; (3)</pre>

a) Antag at MyFan ikke indeholder nogen tupler, at transaktionerne kører på isoleringsniveau READ COMMITED, og at de enkelte SQL kommandoer sendes til DBMSen i den rækkefølge, der er vist ovenfor. Hvilke tupler returneres af hver af de 3 SELECT statements?

Svar:

- (1) (7,'Birger Hansen')
- (2) (3,'Bent Ølgård') og (5,'Birger Hansen')
- (3) (3,'Bent Ølgård') og (5,'Birger Hansen').

5 Constraints (10%)

Antag at relationerne fan og spiller er oprettet uden nogen form for constraint og at tablerne indeholder de data der er vist i opgave 3. Vi tilføjer nu constraints til tabellerne med følgende kommandoer:

- ALTER TABLE spiller ADD CONSTRAINT MyFirstConstraint PRIMARY KEY (id);
- ALTER TABLE fan ADD CONSTRAINT MySecondConstraint FOREIGN KEY (favorit) REFERENCES spiller(id);
- ALTER TABLE fan ADD CONSTRAINT MyThirdConstraint UNIQUE (cprnr);

a) Angiv for hver af følgende kommandoer hvilke af de tre ovenstående constraints (om nogen) der brydes, dvs. giver anledning til en fejlmeddelelse.

1. DELETE FROM spiller WHERE land='Sverige';
2. INSERT INTO spiller VALUES (6,'Michael Gravgaard','Danmark');
3. UPDATE fan SET cprnr=1214650124 where navn LIKE '%Hans%';
4. INSERT INTO fan VALUES (7,'Hans Metz',NULL,2001,7);
5. UPDATE fan set favorit=NULL where navn LIKE '%e%';

Brug svararket til dit svar. Skriv plus (+) for at markere en brudt constraint, og minus (-) for at markere en constraint, der ikke brydes. Hvis du er i tvivl om et svar, kan du skrive et spørgsmålstegn.

Svar:

Spørgsmål 5.a	1	2	3	4	5
MyFirstConstraint	-	-	-	-	-
MySecondConstraint	+	-	-	+	-
MyThirdConstraint	-	-	+	-	-

Svarark (afleveres)

Navn		Sidenummer	
CPR		Totalt sidetal	

Instruktioner. I spørgsmål 1.a og 2.a skal du sætte præcis ét X i hver søjle. I spørgsmål 3.a skal du skrive 7 heltal. I spørgsmål 5.a skal du markere svarene med + og -. Hvis du er i tvivl kan du angive et spørgsmålstegn.

Bemærk at retningen vil blive gjort på en måde, så tilfældige svar ikke betaler sig. For eksempel vil to korrekte svar og et forkert svar give det samme antal point som ét korrekt svar og to spørgsmålstegn. Det er tilstrækkeligt at give korrekte svar for at få maksimumpoint, men hvis du vælger at forklare dine svar vil dette blive taget i betragtning ved retningen.

Spørgsmål 1.a	1	2	3	4	5	6	7	8
Stemmer med EER								
Stemmer ikke med EER								
?								

Spørgsmål 2.a	1	2	3	4	5
Ikke FD					
Måske FD					
?					

Spørgsmål 3.a	1	2	3	4	5	6	7
Antal tupler							

Spørgsmål 5.a	1	2	3	4	5
MyFirstConstraint					
MySecondConstraint					
MyThirdConstraint					