Database Systems

IT University of Copenhagen

June 8, 2006

This exam consists of 5 problems with 16 questions, printed on 7 numbered pages. The weight of each problem is stated. You have 4 hours to answer all questions. If you are unable to answer a question, try to give a partial answer. You may choos to write in English or Danish.

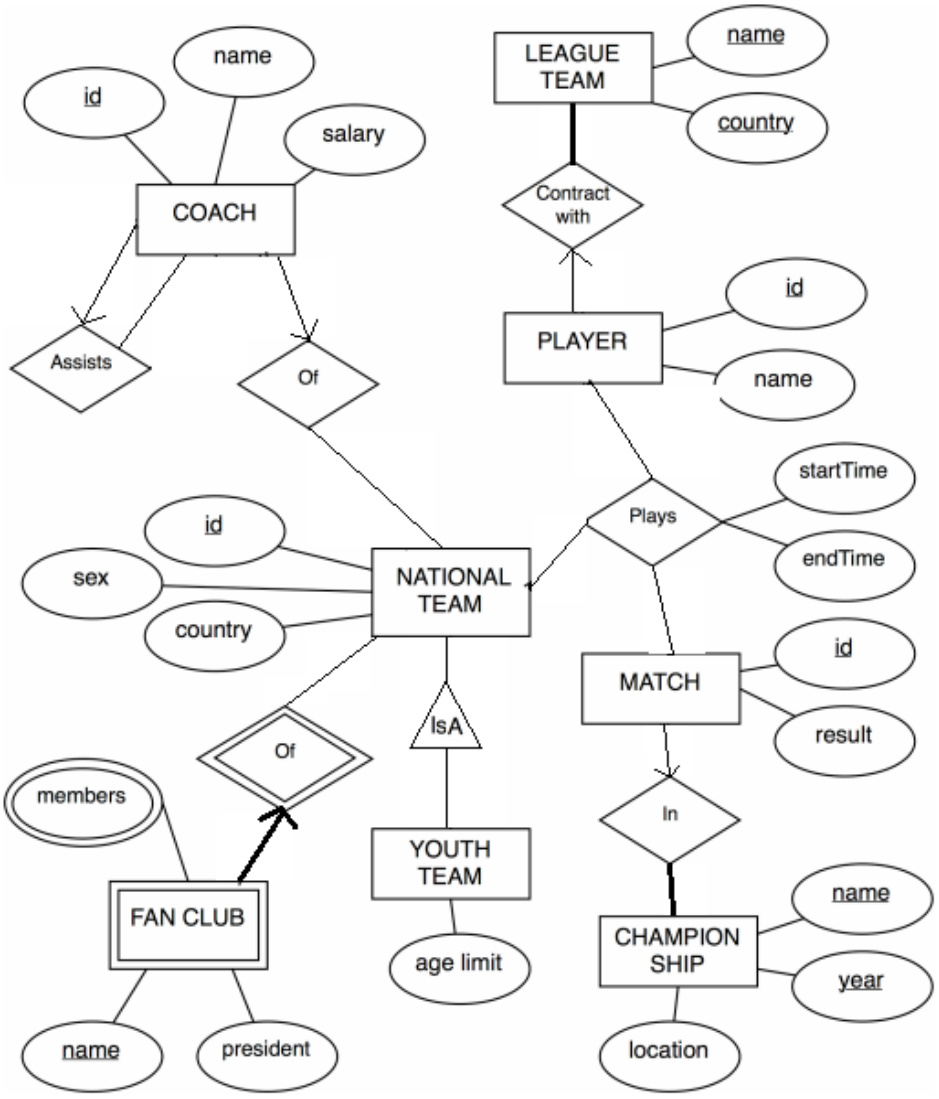
Pages in your answer should be numbered and contain name, CPR number, and course code (DBS). Write only on the front page of pages, and put the pages in order before handing in.

"MDM" refers to the course book "Modern Database Management 7th edition" by Jeffery A. Hoffer, Mary B. Prescott, and Fred R. McFadden.

All written aids are allowed.

1 Data modeling (35%)

Consider the below EER diagram (extended Chen notation), modeling data on national team soccer: Coaches, fan clubs, matches, championships, and players. For coaches, it is recorded who is assisting (Assists). For players it is modeled with national league team (LEAGUE TEAM) they have a contract with (Contract with). Some national teams are youth teams – the YOUTH TEAM and NATIONAL TEAM entity sets are connected by an "IsA" relationship. For fan clubs it models who are members, and who is president. For each game it models which players were active (Plays), and in what time period (between startTime and endTime). If the whole match is played, these numbers are 0 and 90, respectively



a) Indicate for each of the following statements if they agree with the EER diagram. (Observe that the diagram does not necessarily model reality exactly.)

1. A national team always has at least 1 coach.
2. The assistant of a coach can have an assitant herself.
3. A player has a contract with at most 1 league team.
4. A player cna take part in matches for more than 1 country.
5. A player can be substituted in and out several times in a match, and hence have several starting times.
6. A youth team can take part of a championship.
7. There can be 20 players on court for each team in a match.
8. There can be two fan clubs of the same name.

b) Convert the EER diagram to relations. When there are several choices, you should choose a method that minimizes the number of relations. Write the schemas of the resulting relations, with primary key attributes underlined.

The EER diagram does not model historic data on player careers (what teams they have played for, in what periods, and for what salary). Further, a playing coach will correspond to an instance of the COACH entity as well as a PLAYER entity, with no information that this is the same person. A new data model is sought where these restrictions do not apply.

Further, the new data model should make it possible to register not only the result of the match, but also the most important events in a match:

- Goals (who is goal scorer is, and in what minute the goal was scored).
- Penalties (what minute, who committed the penalty, and against whom).
- Red and yellow cards (who and when).
- Substitutions – as in the present ER diagram.

c) Draw a revised ER model in your chosen notation, taking the above wishes into account. You should strive to make a flexible data model, which can easily be extended with more detailed information. Write explanatory text if needed to understand your reasoning.

2 Normalization (15%)

Consider a relation with the schema: `Sales(seller,producer,product,amount)`. The following is a legal instance of `Sales`:

seller	producer	product	amount
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) Which of the following potential FDs do *not* hold, based on the instance above?

1. `amount` → `product`
2. `amount` → `product seller`
3. `product` → `producer`
4. `producer` → `product`
5. `seller product` → `amount`

The instance above can be computed as the join of these relations:

seller	producer
Silman	SoftFloor AG
Bjarnes Tæpper	Bøgetæpper
Top Tæpper	Bøgetæpper

seller	product	amount
Silman	Velour	101000
Bjarnes Tæpper	Berber	207000
Top Tæpper	Kashmir	77000
Silman	Berber	72000
Bjarnes Tæpper	Valnød	17000

b) State a functional dependency (FD) that ensure that `Sales` can be split as in the example given with no loss of information. In other words, the FD should ensure that the SQL statement

```
(SELECT seller, producer FROM Sales) NATURAL JOIN
(SELECT seller, product, amount FROM Sales)
```

always returns a relation that is identical with `Sales`. Further, give an explanation in words of what the FD expresses.

c) Give an instance of `Sales` where the chosen split does not work, i.e., where the SQL statement in question b) does *not* return the same instance.

3 SQL (30 %)

Consider the relations `fan(id,name,cprnr,memberSince,favorite)` og `player(id,name,country)`, and instance with the following data:

id	name	cprnr	memberSince	favorite
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	name	country
1	Peter Ijeh	Nigeria
2	Marcus Allbäck	Sverige
3	Martin Bernburg	Danmark
4	Jesper Christiansen	Danmark
5	Michael Gravgaard	Danmark

The relations contain data on members in a fan club, and their favorite players.

a) How many tuples are returned for each of the following queries, when run on the instances above?

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

b) Write an SQL command that, for each tuple in the relation `fan` where `cprnr` is larger than `3112999999` or less than `0101000000`, sets `cprnr` to the value `NULL`.

c) Write an SQL command that deletes all tuples in `fan` where `cprnr` has the value `NULL`.
   ```

d) Write a query that, for each member in the fan club, computes the name of the member and the name of his/her favorite player.

e) Write an SQL query that computes the average of the column `memberSince` in the relation `fan`.

f) Define an SQL view that for each member in the fan club shows the name of the member, and the name of the members' favorite player. Use your view to write a query that computes the number of fans of each player (the name of the player must be shown).

g) Write an SQL query that returns a relation with a *single* attribute, containing all names in `fan` and `player`. You can assume that the data types for the `name` attributes are identical.

h) Write an SQL query that returns the names of all players that have more female than male fans. A person in `fan` is male if and only if the expression `cprnr % 2 = 1` is true.

4 Transactions (10 %)

Consider two database connections that make updates and queries on the relation `MyFan(id, name)`:

Connection 1	**Connection 2**
`INSERT INTO MyFan VALUES (3,'Bent Ølgård');`	
	`INSERT INTO MyFan VALUES (7,'Birger Hansen');`
`INSERT INTO MyFan VALUES (5,'Birger Hansen');`	
	`SELECT * FROM MyFan;` (1)
`COMMIT;`	
	`DELETE FROM MyFan;`
`SELECT * FROM MyFan;` (2)	
	`ROLLBACK;`
	`SELECT * FROM MyFan;` (3)

a) Assume that `MyFan` does not contain any tuples, that the transactions are running at isolation level `READ COMMITED`, and that the individual SQL commands are sent to the DBMS in the sequence shown above. Which tuples are returned for each of the 3 `SELECT` statements?

5 Constraints (10%)

Assume that the relations `fan` and `player` have been created with no constraints, and that the tables contain the data shown in problem 3. We now add constraints with the following commands:

- `ALTER TABLE player ADD CONSTRAINT MyFirstConstraint PRIMARY KEY (id);`
- `ALTER TABLE fan ADD CONSTRAINT MySecondConstraint FOREIGN KEY (favorite) REFERENCES player(id);`
- `ALTER TABLE fan ADD CONSTRAINT MyThirdConstraint UNIQUE (cprnr);`

a) State for each of the following commands which of the three above constraints (if any) are violated, i.e., result in an error message.

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%';`