Introduction to Databases

IT University of Copenhagen

January 7, 2005

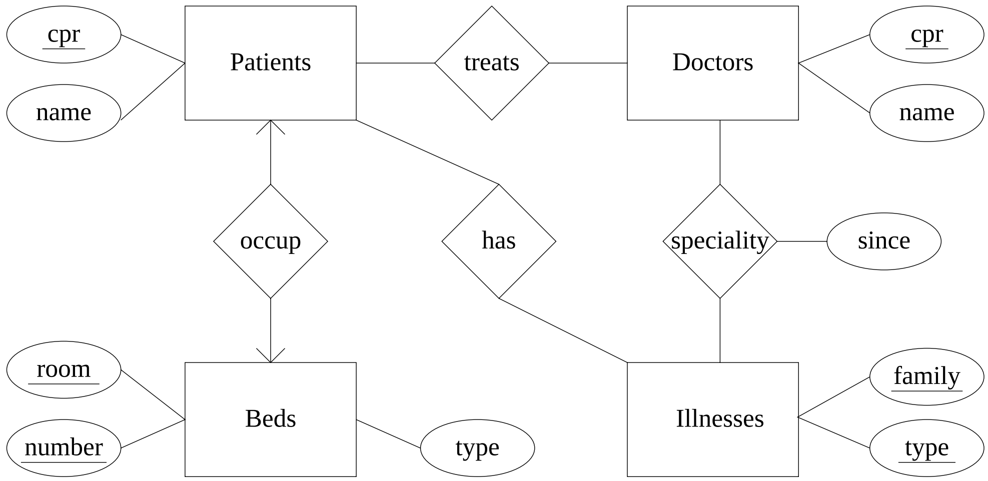
This exam consists of 6 problems with a total of 16 questions. The weight of each problem is stated. You have 4 hours to answer all 16 questions. If you cannot give a complete answer to a question, try to give a partial answer. You may choose to write your answer in Danish or English. Your answers should appear in the same order as the corresponding questions. Remember to write the page number and your CPR-number on each page of your written answer. The complete assignment consists of 5 numbered pages (including this page).

GUW refers to *Database Systems – The Complete Book* by Hector Garcia-Molina, Jeff Ullman, and Jennifer Widom, 2002.

All written aids are allowed / Alle skriftlige hjælpemidler er tilladt.

1 Database design (20%)

Consider the following E/R diagram, modeling data about patients in a hospital:



a) Perform a conversion of the E/R diagram into relation schemas, using the method described in GUW. You should eliminate relations that are not necessary (e.g., by combining relations).

Imagine you are given the task of integrating the database for the data of the E/R diagram with a database used for managing beds. It has the following relational database schema:

```
Beds(room_id,bed_number,type,buy_date)
Rooms(room_id,type,capacity)
BedBookings(room_id,bed_number,from_date,to_date,patient_cpr)
```

The `Beds` relation contains information on all beds in the hospital, including which room they are in. It corresponds to the "Beds" entity set, plus one additional attribute. The `Rooms` relation contains information about each room. Finally, the `BedBookings` relation contains information about what beds will be used in future, planned hospitalizations. It contains information on at most one hospitalization for each patient.

b) Integrate the relational database schema in the E/R diagram above, and draw the complete E/R diagram. The resulting E/R diagram should have the property that it can be converted into relations using the method described in GUW, so that the relations `Beds`, `Rooms`, and `BedBookings` have exactly the above schemas.

2 Normalization (15%)

Consider the relations `Beds`, `Rooms`, and `BedBookings` from Problem 1. After questioning the facilities management of the hospital, you have the following information about what kind of data can occur in these relations:

- The only key of `Beds` is `(room_id,bed_number)`.
- All beds bought on the same date are of the same type.
- All beds in the same room were bought on the same date.
- The only key of `Rooms` is `(room_id)`.
- Rooms of the same type can have different capacities.
- There can be rooms of different types with the same capacity.
- The only key of `BedBookings` is `(room_id,bed_number,patient_cpr)`.
- The attribute `to_date` is either `NULL` or larger than `from_date`.

a) Based on the above information, give an example of redundancy and an example of an update anomaly in the relations.

b) Identify all avoidable functional dependencies in the three relation schemas (i.e., nontrivial functional dependencies that do not have a superkey on the left hand side).

c) Based on the functional dependencies from b), perform a decomposition of the relations into BCNF.

3 Transactions (10%)

Relation `BedBookings` from Problem 1 has three types of transactions performed on it:

1. Booking a bed. This involves finding a bed in `Beds` that only occurs in `BedBookings` with `to_date` before a certain date, and using it for a new tuple in `BedBookings`.
2. Signing out a patient, i.e., setting the `to_date` attribute to a certain date.
3. Gathering statistics, i.e., computing an aggregate on the relation.

a) Suggest an appropriate SQL isolation level for each type of transaction. Argue in favour of your choices.

4 SQL and relational algebra (30%)

Consider the relations Beds, Rooms, and BedBookings from Problem 1.

a) Write an SQL query that computes the total capacity of all rooms where type is 'T'.

b) Write a query in SQL that computes the room_ids and types of rooms with at least one bed having buy_date < '1990'. (If desired, you may use the fact from Problem 2 that all beds in the same room were bought on the same date.)

Consider the following SQL expression:

```
UPDATE BedBookings
SET room_id=NULL, bed_number=NULL
WHERE (room_id,bed_number) IN (SELECT room_id,bed_number
                               FROM Beds
                               WHERE type='OLD');
```

c) Give a short and precise explanation of what changes are performed on the data when the above expression is run.

d) Write an SQL statement that sets the capacity of every room in Rooms to the number of beds that are currently in the room (as registered in the Beds relation).

e) Write relational algebra expressions corresponding to the SQL of questions a) and b).

5 OLAP (10%)

This problem concerns the construction of a relational OLAP system for data about traffic. Data for the system comes from a sensor that registers passing vehicles, and measures their speed and their type ("bike", "car", "van", or "truck"). The sensor data is combined with information about the time and day of week, and the weather ("snowy", "rainy", or "dry").

a) Identify the facts, measures and dimensions to be used in an OLAP system for the traffic data.

b) Give a star schema for the data.

6 SQL privileges (15%)

Consider the relation `BedBookings` from Problem 1. Suppose that it is created by the user `dba`, who executes the following statements:

```
GRANT SELECT ON BedBookings TO adm WITH GRANT OPTION;
GRANT UPDATE ON BedBookings TO adm WITH GRANT OPTION;
GRANT DELETE ON BedBookings TO adm;
```

Subsequently, the user `adm` executes these statements (some of which may result in error messages from the DBMS):

```
GRANT SELECT ON BedBookings TO doc;
GRANT UPDATE(from_date,to_date) ON BedBookings TO doc WITH GRANT OPTION;
GRANT DELETE ON BedBookings TO doc;
```

a) State what kinds of rights (SELECT, UPDATE, DELETE) the user `doc` has on the relation `BedBookings`.

Now assume that the user `dba` executes the following statements (some of which may result in error messages from the DBMS):

```
REVOKE SELECT ON BedBookings FROM adm CASCADE;
REVOKE UPDATE(from_date) ON BedBookings FROM adm CASCADE;
```

b) State the rights of the user `doc` after the above REVOKE statements.

The following SQL query returns all tuples in `BedBookings` concerning female patients, omitting the `patient_cpr` attribute. (It uses the fact that females have even CPR numbers.)

```
SELECT room_id,bed_number,from_date,to_date
FROM BedBookings
WHERE (patient_cpr%2=0);
```

c) Write SQL statements that, if executed by the user `dba`, allows the user `public` to retrieve the information produced by the above query, but does *not* allow `public` to access any CPR numbers, or any tuples concerning males. **Hint:** First define a view.