# Exam 2021 Comments

Frederik Emil Nyborg

April 2022

## 1 Question 1

The text under conceptual design states that a patient may or may not have been allocated a *roomNo*, this tells us that patient has partial participation, and that a patient can associated with 0 or 1 rooms. From the two schemas, we can see that *roomNo* 12 is associated with two patients, meaning that a room can have multiple patiennts. From this information we know, that the relationship between the two entities is a many-to-one. We can also see that the room with *roomNo* 13 does not have any patients, we can therefore conclude that room has partial partial participation. To summarize then we know that both of the entities have partial participation and the relationship is many-to-one. The correct answer is therefore option 2, since option 1 is not a correct depiction of a relationship in an E-R diagram and the other options does not follow the constrains mentioned above. (See slide 9-10 and 16-17)

## 2 Question 2.1

The participation for the patients is partial, so a patient is allowed to not have received any treatment. (See slide 17 in week 6)

*date* is a multivalued attribute, which is showcased by the {}, and it should therefore be a part of the primary key. The rest of the primary key of the relation will be the union of the primary keys from each of the entities in the relationship, due to it being a many-to-many relationship. The Relation Schema for the relationship received will be *received(<u>patientID</u>, <u>treatmentName</u>, <u>date</u>)*. (See slide 12, 31 and demo exercise 6.4 in week 6). A patient can't receive the same treatment on the same date, as the value of the key would then be the same for the two events and there would be no way to register that there had been two treatments on the same date for the same patient.

We found out above that the Relation Schema for the relationship received will be *received(<u>patientID</u>, <u>treatmentName</u>, <u>date</u>)*. Since *date* is a part of the primary key, then a patient can receive the same treatment on different dates.

Since *treatmentName* is a part of the primary key, then a patient can receive two different treatments on the same date.

Since *patientID* is a part of the primary key, then two different patients can receive the same treatment on the same date.

# 3 Question 2.2

The primary key of the relation will contain the union of the primary keys from each of the entities in the relationship, due to it being a many-to-many relationship. Since *date* is a multivalued attribute, which is showcased by the {} encapsulating *date*, then it should be a part of the primary key. The Relation Schema for the relationship received will be *received(<u>patientID</u>, <u>treatmentName</u>, <u>date</u>)*. (See slide 12, 31 and demo exercise 6.4 in week 6)

# 4 Question 3.1

To determine the candidate key we are gonna be using Armstrong's Rules (slide 15-16 week 7). From the stated assumed function dependencies we can do the following:

| | |
|---|---|
| 1.$AB \rightarrow AB$ | Rule 4 Self-determination |
| 2.$AB \rightarrow E$ | Rule 3 Transitivity of AB → C and given C → E |
| 3.$AB \rightarrow ABCDE$ | Rule 7 Composition AB → ABCE and B → D |
| 4.Candidate keys: | A and B combined is the only candidate key |

The first option is the correct option.

# 5 Question 3.2

If we assume that all the attribute values are a single value we can assume that the relation is at least in 1NF. The relation can not be in 2NF since the attributes D do not depend on the entire primary key, but only the attribute B. So the highest normal form the relation R can be is in 1NF (See week 7 slide 18 or 21)

# 6 Question 3.3

In exercise 3.2 we argued why the relation was only in 1NF. To make the relation in 2NF we create a new relation R1 to where we move all attributes (D) which depend on the subset of the primary key (B): R0($\underline{A}$, $\underline{B}$, C, E) and R1($\underline{B}$, D) (See week 7 slide 21). R1 is in 3NF since all attributes (D) depend directly on the primary key (B). R0 is not in 3NF since E depends transitively on the primary key via the attribute C. To normalize R0 to 3NF we create yet another relation R2 together with the attribute (E) which depends transitively on the primary key and a copy of the attribute it depends via (C) which becomes the primary key of the new relation, and we remove E from R0 and call the resulting relation R3: R1($\underline{B}$, D) and R2($\underline{C}$,E), R3($\underline{A}$, $\underline{B}$, C) (See slide 24). All three relations are in BCNF all non primary key attributes in the relations only depend on the primary key (See week 7 slide 28). Regarding 4NF, then the only non-trivially multivalued dependences $\alpha - - > \beta$ for the relations are some that are also functional dependences, and we can see that for these that $\alpha$ are a key (See week 7 slides 31, 33). Therefore we end out with the following relations being in 4NF: R1($\underline{A}$, $\underline{B}$, C), R2($\underline{B}$, D) and R3($\underline{C}$,E) which matches the first answer.

## 7 Question 4.1

Since *patientID* is the primary key, then its not allowed to be null (See week 1 slide 11)
Since *roomNo* has no "NOT NULL" constraint then its allowed to be null (see week 2 slide 20)

## 8 Question 4.2

Since patient has no foreign key constraints on *roomNo* then a delete violation occurs and no rows are deleted (See week 2 slide 17-18).

## 9 Question 4.3

If a *roomNo* should cease to exist then we might still want to keep the data for any patients which might be located inside of the room. Therefore we would simply wish to replace the *roomNo* with a *null* value in the *patient* relation instead of deleting the row. On delete set null is therefore the most natural one.

## 10 Question 4.4

The last foreign key constraint is regarding the *patient* relation. If a patient was to be removed from the database, then it would not make sense to keep a entry in the *treats* relation with a *doctorID* and a *null* value as *patientID*. Instead we would much rather just delete the row in the *treats* relation, since the doctor is no longer treating that given patient. On delete cascade is therefore the most natural one.

## 11 Question 5.1

To find the *roomNo* in *room* which is not in use in *patient* we need to compute the set difference between the *roomNo* in *room* and *roomNo* in *patient*. On slide 8 in week 10 the set difference is defiend to be $R - S$ in relationsal algebra where R and S are relations. The *roomNo* in *room* and *patient* can be found using projection as seen on slide 9 in week 10. This gives us the following expression:$\Pi_{roomNo}(room) - \Pi_{roomNo}(patient)$. This question is also very similar to 10.1.4 on slide 10 in week 10.

## 12 Question 5.2

In this exercise I would try to find a similar type of question in the slides or exercises/solutions. 10.3.4 on slide 27 in week 10 does what we want, just on other relations. Using 10.3.4 and using the attributes in our relations we get the following expression: $\{< r > | \exists c(< r, c > \in room \wedge \neg(\exists pid, n(< pid, n, r > \in patient)))\}$

## 13 Question 6.1

The *doctor* relation is sorted based on *doctorID* and the index file is sorted on *specialty*. Since the relation and the index file are sorted on two different search keys then the index is secondary (See week 11 slide 10)

## 14 Question 6.2

The index file is dense as it contains a record for every search key value in the *doctor* relation, i.e. all of the values in *specialty* in the *doctor* relation can be found in the index file (See week 11 slide 11)

## 15 Question 6.3

Since the common query search for a specific value and not a range of values then hashing should be used on the attribute which is being searched on. I.e. I would use hashing with search key *patientName* (See week 11 slide 46).