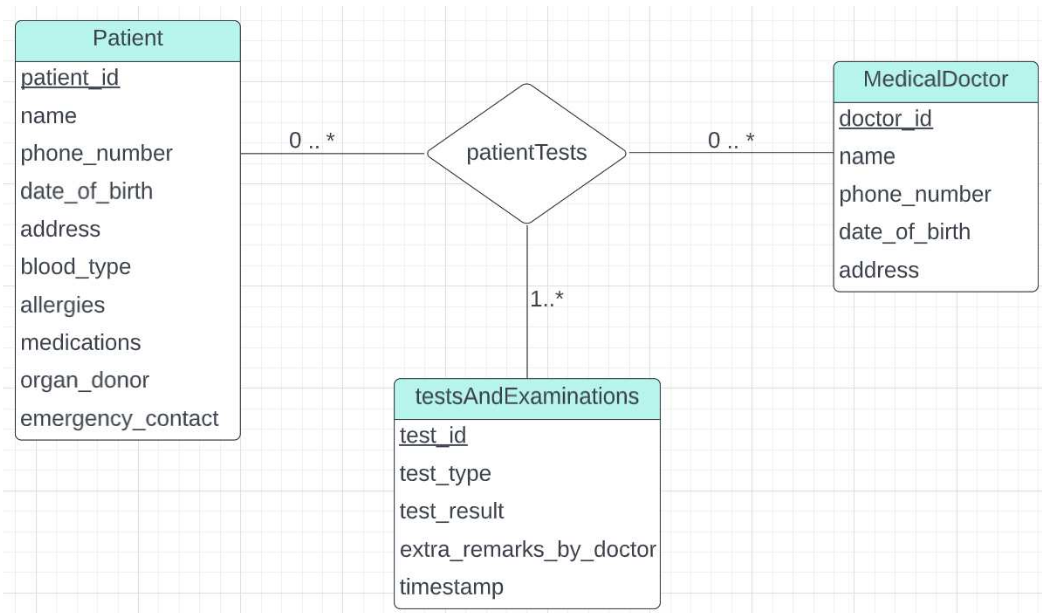
Database Design Exercises (Ex No: 3)

Practitioner: Mobin Kheibary [994421017]

Supervisor: Dr. Ehsan Shoja

Chapter 6

Solution to *Ex-15*:

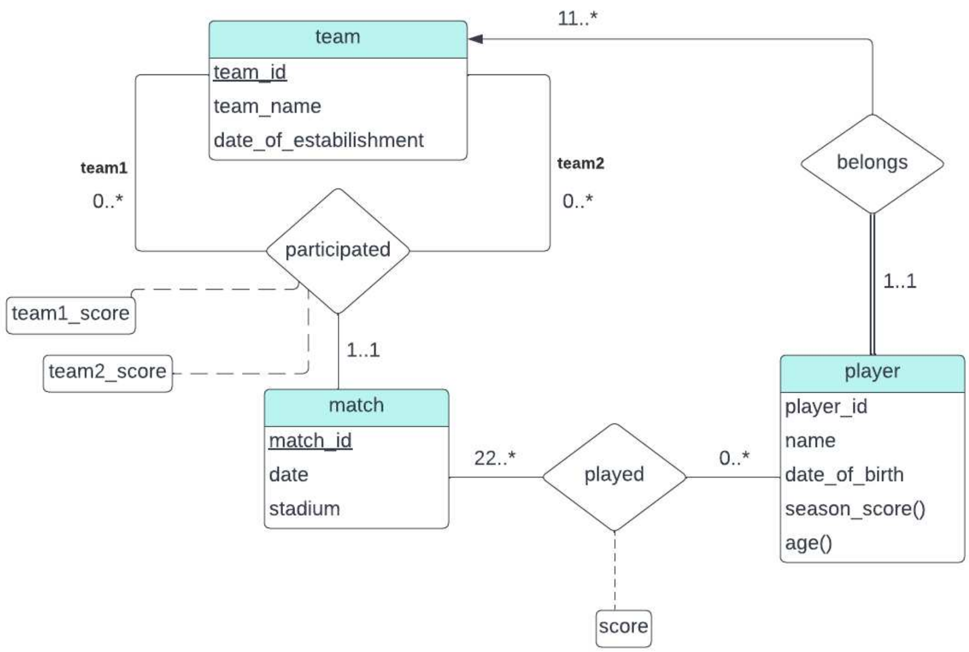


patientTests is a ternary relationship set.

Another method is, to make the testsAndExaminations entity a weak entity having identifying entity set Patient. And then adding a relationship set between the weak entity testsAndExaminations and MedicalDoctor, representing which medical doctor performed which test and examination. In fact doing that has the added benefit of constraining each entity in testsAndExaminations to a single Patient.

But using a ternary relationship as depicted in the above diagram, also has its benefits. For example, if a group of patients are tested and examined by the same type of test and have the same result, we might associate each of the patients in the group to the same entity in testsAndExaminations.

Solution to *Ex-16*:



The above design assumes that the game is soccer. That explains the mapping cardinalities given in the picture.

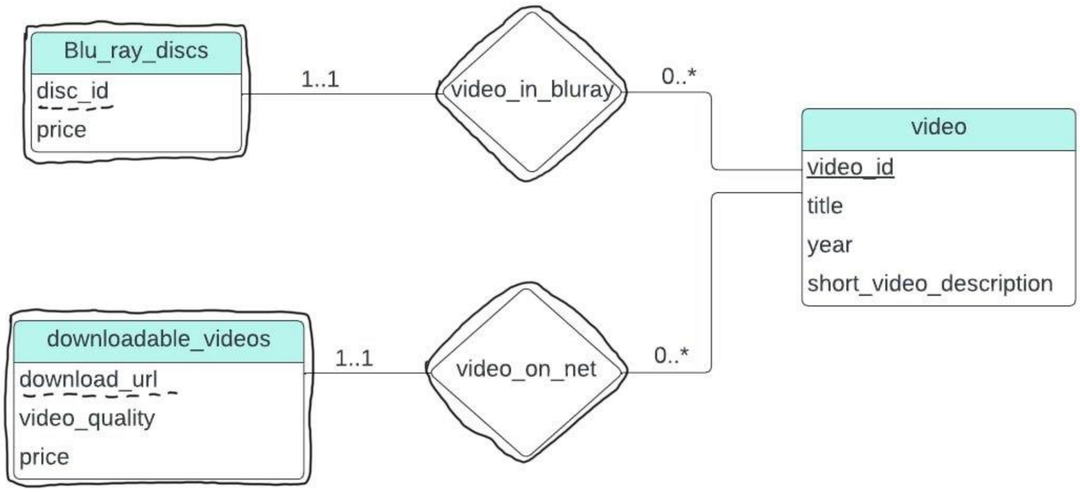
Solution to *Ex-17*:

A weak entity set is one whose existence is dependent on another entity set, called its identifying entity set; instead of associating a primary key with a weak entity, we use the primary key of the identifying entity, along with extra attributes, called discriminator attributes to uniquely identify a weak entity.

An entity set that is not a weak entity set is termed a strong entity set.

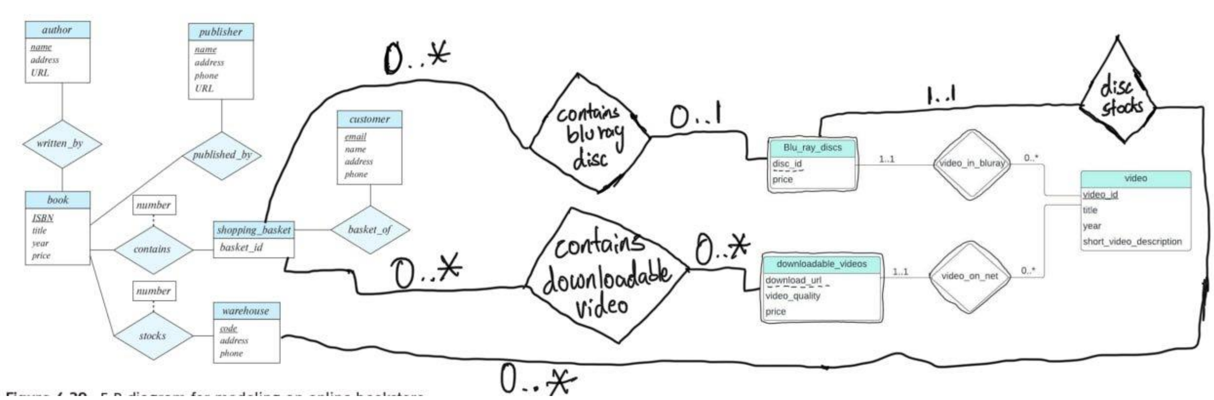
Solution to *Ex-21*:

a.



Note that Blu_ray_discs and downloadable_videos are weak entities while video_in_bluray and video_on_net are the identifying relationships sets. video is the identifying entity set and owns both of the weak entities.

b.



Chapter 7

Solution to *Ex-21*:

We will use the algorithm given on Figure 7.11 (BCNF decomposition algorithm).

One possible decomposition is:

$$(A, B, C, E), (B, D)$$

Solution to *Ex-23*:

- *repetition of information* : When inserting data into our database model, if the model requires us to insert the same information multiple times, then we say our database model has the *repetition of information* issue. Note that we may sometimes intentionally want some information to be repeated for performance reasons.

- *inability to represent information* : If the database model was not designed well or not taking into account some things in reality, then the issue of *"inability to represent information"* may arise. For example, in our university schema if we removed the *department* relation, and instead used the schema *instructor(ID, name, dept_name, salary)* to represent both the instructor and the departments in the university, then our database model would NOT be able to represent a department having no instructors.

Solution to *Ex-26*:

α	γ	β
1	6	7
2	3	5
2	4	5

Solution to *Ex-29*:

Take the following instance of $r(R)$:-

A	B	C	D	E
1	6	5	7	3
2	8	5	9	4

Then $\Pi_{A,B,C}(r)$ is:-

A	B	C
1	6	5
2	8	5

$\Pi_{C,D,E}(r)$ is:-

C	D	E
5	7	3
5	9	4

And their natural join $\Pi_{A,B,C}(r) \bowtie \Pi_{C,D,E}(r)$ is:-

A	B	C	D	E
1	6	5	7	3
1	6	5	9	4
2	8	5	7	3
2	8	5	9	4

Thus, the decomposition is a **lossy decomposition**.

Solution to *Ex-30*:

a.

$$B^+ = A, B, C, D, E$$

b.

$A \rightarrow BCD$ holds (given).

By **Decomposition rule** (I know that Decomposition rule is not one of Armstrong's axioms, but since I have proved it in Exercise 7.27 using Armstrong's axioms I think it is okay to use it here.)

$A \rightarrow BC$ holds (Decomposition rule).

$BC \rightarrow DE$ holds (given).

By **Transitivity rule** $A \rightarrow DE$ holds.

Thus, $A \rightarrow BCDE$ holds by **Union rule** (see Exercise 7.4).

By **Augmentation rule** $AG \rightarrow ABCDEG$.

This proves that AG is a superkey.

c.

Apply algorithm given in Figure 7.9.

D is extraneous in $A \rightarrow BCD$ so, remove it.

D is also extraneous in $BC \rightarrow DE$ so, remove it.

Thus the following is a canonical cover of F.

$$A \rightarrow BCBC \rightarrow EB \rightarrow DD \rightarrow A$$

d.

The following is a 3NF decomposition of the given schema based on a canonical cover given above.

$$A, B, C, B, C, E, B, D, D, A, A, G$$

e.

$$A, B, C, B, D, A, E, A, G$$

Solution to *Ex-32*:

a.

I claim that such a functional dependency does **NOT** exist. Suppose to the contrary that such a functional dependency exists. Say $\alpha \to \beta$. Thus $\alpha \to \beta$ is a nontrivial functional dependency containing no extraneous attributes and it is logically implied by F. Define $F_1 := F \cup \alpha \to \beta$. Now consider an attribute $X \in \beta$. I am going to prove that X is extraneous.

Consider the set

$$F_1' = (F_1 - \alpha \to \beta) \cup \alpha \to (\beta - X)$$

Since the functional dependency $\alpha \to X$ can be inferred from F_1' (in fact the whole $\alpha \to \beta$ can be inferred from $F \subset F_1'$) X is extraneous in β.

Thus such a functional dependency does **NOT** exist.

b.

We use the algorithm given on Figure 7.11.

By using the functional dependency $A \to BC$ we decompose the schema into

$$A, D, E, G, A, B, C$$

Note that A, B, C is in BCNF. By applying **Augmentation rule** followed by **Transitivity rule** on the functional dependencies given in F, we see that the functional dependency $AD \to E$ holds. We use that to decompose the schema A, D, E, G, into $ {A, D, G} $ and $ {A, D, E} $.

Thus,

$$A, D, G, A, D, E, A, B, C$$

form a BCNF decomposition of R.

c.

The decomposition that the algorithm given on Figure 7.11 generates is always a lossless decomposition. Thus our decomposition is a lossless decomposition.

d.

Our decomposition is **not** dependency preserving. The functional dependency $BD \to E$ is not preserved (we used the second test given in section 7.4.4)

The End.