

Schema Refinement and Normalization

Doga Devrim Celik and Nuri Tas

May 2023

1 Definitions of BCNF and 3NF

For all $X \rightarrow A \in F^+$

BCNF:

1. $A \in X$ or
2. X is a superkey.

3NF:

1. $A \in X$ or
2. X is a superkey or
3. A is part of some key for R .
4. q11 is skipped for the time being

2 Changes Made to the first ER Diagram

1. The foreign key condition in the Director table is updated to make sure the platform id in the Movies has the same platform of its director:
FOREIGN KEY (username, platform id) REFERENCES Director(username, platform id)
2. The primary key condition in the Director table is converted to UNIQUE constraint to allow directors to have no platforms:
UNIQUE (username, platform_id)
3. A constraint is added to Movie Sessions only to allow 4 different time slots per day:
CONSTRAINT CHK-TimeSlot CHECK (time slot between 1 and 4)
4. Check condition is added to Ratings to ensure only ratings between 0 and 5 are allowed:
CHECK rating-check (rating between 0 and 5)

5. A new table named *AverageRatings* is created
6. The following primary key constraint is added to the Subscribed Platforms table to make sure there are no duplicate rows in the table.
PRIMARY KEY (username, platform_id)
7. The primary key constraint on the Movie table has changed to the following form to ensure Movies satisfy 3NF and each movie's platform id is the same as its director's:
PRIMARY KEY (movie_id, username) Note, Movie table isn't in BCNF as is.
8. A primary key constraint is added to the Predecessors:
PRIMARY KEY (predecessor_id, movie_id)
9. A primary key constraint is added to the has_genre table:
PRIMARY KEY (movie_id, genre_id)
10. A primary key constraint is added to the Database Managers table as username must be unique:
PRIMARY KEY (username)

3 Relations and Their Normal Forms

Note, R represents all the attributes for a given relation. FDs include all functional dependencies for a given relation. Additionally, there have been made some assumptions about some attributes to ensure all functional dependencies of a given relation are included, although the absence of those assumptions would not change whether a given relation is in BCNF. For instance, we assumed that the movie name uniquely determines movie id, duration, etc.

3.1 Users

User(username, _password, _name, surname)
FDs:

1. username \rightarrow R

User satisfies BCNF as username is a superkey (actually the primary key).

3.2 Audience

Audience(username)
the Audience table also has only one attribute and serves as an abstract table.
FDs:

1. $\text{username} \rightarrow R$

Audience satisfies BCNF as username is a superkey.

3.3 Subscribed Platforms

Subscribed Platforms(username, platform id)

the Subscribed Platforms table has only two columns and both are together primary key FDs:

1. $(\text{username, platform id}) \rightarrow R$

Subscribed Platforms satisfies BCNF as username and platform id are together superkey.

3.4 Director

Director(username, platform id, nationality)

FDs:

1. $(\text{username, platform id}) \rightarrow R$

Director is in BCNF as username and platform id are together primary key, hence superkey

3.5 Movie

Movie(movie id, username, movie name, duration, username)

We can see Movie doesn't satisfy BCNF from the following functional relation

FDs:

1. $(\text{movie id, username}) \rightarrow R$
2. $\text{username} \rightarrow \text{platform id}$

Since username is not a superkey, yet determines another attribute. We could decompose the Movie table by dropping platform id table and creating another table with username and platform id to create two tables each in BCNF. However, this way we would lose the information regarding the platform id of each movie and couldn't track if the platform of each movie is the same as its directors. Hence, we intentionally avoided decomposition and left the Movie table in 3NF form. The table is in 3NF as username is part of the key.

3.6 Predecessors

Predecessors(predecessor id, movie id)

Predecessors is clearly in BCNF as predecessor and movie id are together primary key.

FDs:

1. $(\text{predecessor id, movie id}) \rightarrow R$

3.7 Genre List

Genre List(genre_id, genre_name)

Since, genre_name has UNIQUE constraint it is a superkey and Genre List is in BCNF

FDs:

1. $\text{genre_id} \rightarrow R$
2. $\text{genre_name} \rightarrow R$

3.8 has_genre

has_genre(movie_id, genre_id,)

The only two attributes are together primary key and hence has_genre is in BCNF

FDs:

1. $(\text{movie_id}, \text{genre_id}) \rightarrow R$

3.9 Ratings

Ratings(username, movie_id, rating)

Ratings satisfies BCNF as the only FD can be constructed only with superkeys:

FDs:

1. $(\text{username}, \text{movie_id}) \rightarrow R$

3.10 Average_Ratings

Average_Ratings(movie_id, average_rating)

Similarly, Average_Ratings satisfies BCNF as the only FD can be constructed only with superkeys:

FDs:

1. $\text{movie_id} \rightarrow R$

3.11 Theatre

Theatre(theatre_id, theatre_district, theatre_name, theatre_capacity)

Theatre is in BCNF as only the primary key can determine other attributes:

FDs:

1. $\text{theatre_id} \rightarrow R$

3.12 Movie Sessions

Movie Sessions(session_id, time slot, _date, theatre_id)

Similarly, the Movie Sessions table satisfies BCNF as only the session id (primary key) can imply other attributes

FDs:

1. $\text{session_id} \rightarrow R$

3.13 Screens_as

Screens_as(movie_id, session_id)

Screens_as is in BCNF as the only two attributes are together primary key, hence superkey

FDs:

1. $(\text{movie_id}, \text{session_id}) \rightarrow R$

3.14 Bought_Tickets

Bought_Tickets(username, session_id)

Similarly, Bought_Tickets is in BCNF as the only two attributes are together primary key, hence superkey:

FDs:

1. $(\text{username}, \text{session_id}) \rightarrow R$

3.15 Database_Managers

Database_Managers(username, _password)

Database_Managers satisfies BCNF as the only FD includes a superkey:

FDs:

1. $\text{username} \rightarrow R$