Database Normalized Form Documentation

Overview

This document presents the normalized form analysis of the CineMaster Pro movie management database. The database is designed to store information about movies, actors, directors, genres, users, and reviews with proper normalization to eliminate redundancy and ensure data integrity.

Database Schema

Tables and Their Normalized Forms

1. Movies Table (1NF, 2NF, 3NF)

Column Name	Data Type	Constraints	Description
movie_id	UUID	PRIMARY KEY, NOT NULL	Unique identifier for each movie
title	VARCHAR(255)	NOT NULL	Movie title
description	TEXT	NULL	Movie description/plot
release_date	DATE	NULL	Release date of the movie
duration_minutes	INTEGER	NULL	Duration in minutes
rating	NUMERIC(3,1)	NULL	Average rating of the movie
poster_url	TEXT	NULL	URL to the movie poster image

trailer_url	TEXT	NULL	URL to the movie trailer
created_at	TIMESTAMP	NOT NULL, DEFAULT CURRENT_TIMESTAMP	Record creation timestamp
updated_at	TIMESTAMP	NOT NULL, DEFAULT CURRENT_TIMESTAMP	Record last update timestamp

Normalization Analysis:

- **1NF**: All attributes contain atomic values
- **2NF**: No partial dependencies (all non-key attributes depend on the entire primary key)
- **3NF**: No transitive dependencies (non-key attributes depend only on the primary key)

2. Actors Table (1NF, 2NF, 3NF)

Column Name	Data Type	Constraints	Description
actor_id	UUID	PRIMARY KEY, NOT NULL	Unique identifier for each actor
first_name	VARCHAR(100)	NOT NULL	Actor's first name
last_name	VARCHAR(100)	NOT NULL	Actor's last name
created_at	TIMESTAMP	NOT NULL, DEFAULT CURRENT_TIMESTAMP	Record creation timestamp
updated_at	TIMESTAMP	NOT NULL, DEFAULT CURRENT_TIMESTAMP	Record last update timestamp

Normalization Analysis:

- **1NF**: All attributes contain atomic values
- **2NF**: No partial dependencies
- **3NF**: No transitive dependencies

3. Directors Table (1NF, 2NF, 3NF)

Column Name	Data Type	Constraints	Description
director_id	UUID	PRIMARY KEY, NOT NULL	Unique identifier for each director
first_name	VARCHAR(100)	NOT NULL	Director's first name
last_name	VARCHAR(100)	NOT NULL	Director's last name
created_at	TIMESTAMP	NOT NULL, DEFAULT CURRENT_TIMESTAMP	Record creation timestamp
updated_at	TIMESTAMP	NOT NULL, DEFAULT CURRENT_TIMESTAMP	Record last update timestamp

Normalization Analysis:

- **1NF**: All attributes contain atomic values
- **2NF**: No partial dependencies
- **3NF**: No transitive dependencies

4. Genres Table (1NF, 2NF, 3NF)

Column Name	Data Type	Constraints	Description
genre_id	UUID	PRIMARY KEY, NOT NULL	Unique identifier for each genre
genre_name	VARCHAR(50)	NOT NULL, UNIQUE	Name of the genre

Normalization Analysis:

- **1NF**: All attributes contain atomic values
- **2NF**: No partial dependencies
- **3NF**: No transitive dependencies

5. Users Table (1NF, 2NF, 3NF)

Column Name	Data Type	Constraints	Description

user_id	UUID	PRIMARY KEY, NOT NULL	Unique identifier for each user
username	VARCHAR(50)	NOT NULL, UNIQUE	User's username
email	VARCHAR(255)	NOT NULL, UNIQUE	User's email address
password_hash	VARCHAR(255)	NOT NULL	Hashed password
first_name	VARCHAR(100)	NOT NULL	User's first name
last_name	VARCHAR(100)	NOT NULL	User's last name
role	VARCHAR(20)	NOT NULL, DEFAULT 'user'	User's role (user/admin)
created_at	TIMESTAMP	NOT NULL, DEFAULT CURRENT_TIMESTAMP	Record creation timestamp
updated_at	TIMESTAMP	NOT NULL, DEFAULT CURRENT_TIMESTAMP	Record last update timestamp

Normalization Analysis:

- **1NF**: All attributes contain atomic values
- **2NF**: No partial dependencies
- **3NF**: No transitive dependencies

6. Reviews Table (1NF, 2NF, 3NF)

Column Name	Data Type	Constraints	Description
review_id	UUID	PRIMARY KEY, NOT NULL	Unique identifier for each review
user_id	UUID	NOT NULL, FOREIGN KEY (users.user_id)	Reference to the user who wrote the review
movie_id	UUID	NOT NULL, FOREIGN KEY (movies.movie_id)	Reference to the movie being reviewed

rating	INTEGER	NOT NULL, CHECK (rating >= 1 AND rating <= 10)	Rating given by the user (1-10)
comment	TEXT	NULL	Review comment
created_at	TIMESTAMP	NOT NULL, DEFAULT CURRENT_TIMESTAMP	Record creation timestamp
updated_at	TIMESTAMP	NOT NULL, DEFAULT CURRENT_TIMESTAMP	Record last update timestamp

Normalization Analysis:

- **1NF**: All attributes contain atomic values
- **2NF**: No partial dependencies
- **3NF**: No transitive dependencies

7. Junction Tables (For Many-to-Many Relationships)

Movie_Genres Table (1NF, 2NF, 3NF)

Column Name	Data Type	Constraints	Description
movie_id	UUID	FOREIGN KEY (movies.movie_id), NOT NULL	Reference to movie
genre_id	UUID	FOREIGN KEY (genres.genre_id), NOT NULL	Reference to genre

Normalization Analysis:

- **1NF**: All attributes contain atomic values
- **2NF**: Composite key eliminates partial dependencies
- **3NF**: No transitive dependencies

Movie_Actors Table (1NF, 2NF, 3NF)

Column Name	Data Type	Constraints	Description
movie_id	UUID	FOREIGN KEY (movies.movie_id), NOT NULL	Reference to movie

actor_id	UUID	FOREIGN KEY (actors.actor_id), NOT NULL	Reference to actor
character_name	VARCHAR(255)	NULL	Name of character played by actor

Normalization Analysis:

- **1NF**: All attributes contain atomic values
- **2NF**: Composite key eliminates partial dependencies
- **3NF**: No transitive dependencies

Movie_Directors Table (1NF, 2NF, 3NF)

Column Name	Data Type	Constraints	Description
movie_id	UUID	FOREIGN KEY (movies.movie_id), NOT NULL	Reference to movie
director_id	UUID	FOREIGN KEY (directors.director_id), NOT NULL	Reference to director

Normalization Analysis:

- **1NF**: All attributes contain atomic values
- **2NF**: Composite key eliminates partial dependencies
- **3NF**: No transitive dependencies

Relationships

Entity Relationship Diagram

```
MOVIES ||--o{ MOVIE_GENRES : has
GENRES ||--o{ MOVIE_GENRES : contains
MOVIES ||--o{ MOVIE_ACTORS : features
ACTORS ||--o{ MOVIE_ACTORS : plays_in
MOVIES ||--o{ MOVIE_DIRECTORS : directed_by
DIRECTORS ||--o{ MOVIE_DIRECTORS : directs
```

MOVIES ||--o{ REVIEWS : has

USERS ||--o{ REVIEWS : writes

USERS }|--|{ MOVIES : "can manage"

MOVIES

movie_id PK

title

description

release_date

duration_minutes

rating

poster_url

trailer_url

created_at

updated_at

ACTORS

actor_id PK

first_name

last_name

created_at

updated_at

DIRECTORS

director_id PK

first_name

last_name

created_at

updated_at

GENRES

genre_id PK

genre_name

USERS

user_id PK

username

email

password_hash

first_name

last_name

role

created_at
updated_at

REVIEWS
review_id PK
user_id FK
movie_id FK
rating
comment
created_at
updated_at

MOVIE_GENRES
movie_id FK
genre_id FK

MOVIE_ACTORS
movie_id FK
actor_id FK
character_name

MOVIE_DIRECTORS
movie_id FK
director_id FK

Normalization Summary

First Normal Form (1NF)

All tables in the database satisfy 1NF as all attributes contain atomic (indivisible) values and there are no repeating groups.

Second Normal Form (2NF)

All tables satisfy 2NF as they are in 1NF and all non-key attributes are fully functionally dependent on the primary key. For junction tables with composite keys, all attributes depend on the entire composite key.

Third Normal Form (3NF)

All tables satisfy 3NF as they are in 2NF and there are no transitive dependencies. Non-key attributes depend only on the primary key and not on other non-key attributes.

Boyce-Codd Normal Form (BCNF)

All tables satisfy BCNF as every determinant is a candidate key. This is achieved because:

1. All tables have a single primary key
2. Junction tables have composite primary keys consisting of foreign keys
3. All functional dependencies are based on these keys

Benefits of This Normalized Design

1. **Elimination of Redundancy**: Data is stored only once, reducing storage requirements and ensuring consistency
2. **Data Integrity**: Foreign key constraints maintain referential integrity between related tables
3. **Flexibility**: Easy to add new movies, actors, directors, genres, users, and reviews without modifying the schema
4. **Scalability**: The design can accommodate growth in data volume without performance degradation
5. **Maintainability**: Changes to data require updates in only one place
6. **Query Efficiency**: Proper indexing and normalized structure enable efficient querying

Conclusion

The CineMaster Pro database is properly normalized to at least 3NF (and actually BCNF), which ensures data integrity, eliminates redundancy, and provides a solid foundation for the movie management application. The design supports all required functionality including movie cataloging, user reviews, and relationship management between movies, actors, directors, and genres.