CMPE321 REPORT

Description of the schema refinement step

Director(name: string, password:string, surname:string, username: string, nation: string)

Notation:IPSUN

Functional Dependencies: U -> IPSUN

The only non-trivial FD is key constraint. So, this relation is in BCNF.

Audience(name: string, password:string, surname:string, username: string)

Notation:NPSU

Functional Dependencies: U -> NPSU

The only non-trivial FD is key constraint. So, this relation is in BCNF.

Platform(platform_name: string, platform_id: integer)

Notation:NI

Functional Dependencies: I -> NI, N -> I

I -> NI is key constraint and N -> I implies that N is also super key. So this relation is in BCNF.

Theatre(district: string, capacity: integer, theatre_id: integer, theatre_name: string)

Notation:DCIN

Functional Dependencies: I -> DCIN

The only non-trivial FD is key constraint. So, this relation is in BCNF.

Movie_Session(date: date, time_slot: integer, session_id: integer)

Notation:DSI

Functional Dependencies: I -> DSI

The only non-trivial FD is key constraint. So, this relation is in BCNF.

Database_Manager(password: string, manager_username: string)

Notation:PU

Functional Dependencies: U -> PU

The only non-trivial FD is key constraint. So, this relation is in BCNF.

Genre(genre_name: string, genre_id: integer)

Notation:NI

Functional Dependencies: I -> NI, N -> I

I -> NI is key constraint and N -> I implies that N is also super key. So this relation is in BCNF.

Directed_Movie(movie_name: string, duration: integer, movie_id: integer, username:string, avg_rating)

Notation:NDIUR

Functional Dependencies: I -> NDIUR

The only non-trivial FD is key constraint. So, this relation is in BCNF.

Classify(movie_id: integer, genre_id: integer)

Notation:MG

Functional Dependencies: MG -> MG

This table has only the trivial FD which is a key constraint. So, it is in BCNF.

Next_To(pre_id: integer, suc_id: integer)

Notation:PS

Functional Dependencies: PS -> PS

This table has only the trivial FD which is a key constraint. So, it is in BCNF.

Located(session_id: integer, theatre_id: integer)

Notation:ST

Functional Dependencies: ST -> ST

This table has only the trivial FD which is a key constraint. So, it is in BCNF.

Play(session_id: integer, movie_id: integer)

Notation:SM

Functional Dependencies: SM -> SM

This table has only the trivial FD which is a key constraint. So, it is in BCNF.

Rate(username: string, movie_id: integer, rating:real)

Notation:UIR

Functional Dependencies: UI -> UIR

The only non-trivial FD is key constraint. So, this relation is in BCNF.

Buy(username: string, session_id: integer)

Notation:UI

Functional Dependencies: UI -> UI

This table has only the trivial FD which is a key constraint. So, it is in BCNF.

Agreement(username: string, platform_id: integer)

Notation:UI

Functional Dependencies: UI -> UI

This table has only the trivial FD which is a key constraint. So, it is in BCNF.

Subscribe(username: string, platform_id: integer)

Notation:UI

Functional Dependencies: UI -> UI

This table has only the trivial FD which is a key constraint. So, it is in BCNF.

Updated database design

We deleted User table, because it can make our work a little bit harder in this project. The only aim of that is ensuring that users have unique name, so we checked this with other queries in the project before insert a new value for director or audience.

Director(name: string, password: string, surname: string, username: string, nation: string)

Audience(name: string, password: string, surname: string, username: string)

Platform(platform_name: string, platform_id: integer)

Theatre(district: string, capacity: integer, theatre_id: integer, theatre_name: string)

Movie_Session(date: date, time_slot: integer, session_id: integer)

Database_Manager(password: string, manager_username: string)

Genre(genre_name: string, genre_id: integer)

Directed_Movie(movie_name: string, duration: integer, movie_id: integer, username:string)

Classify(movie_id: integer, genre_id: integer)

Next_To(pre_id: integer, suc_id: integer)

Located(session_id: integer, theatre_id: integer)

Play(session_id: integer, movie_id: integer)

Rate(username: string, movie_id: integer, rating:real)

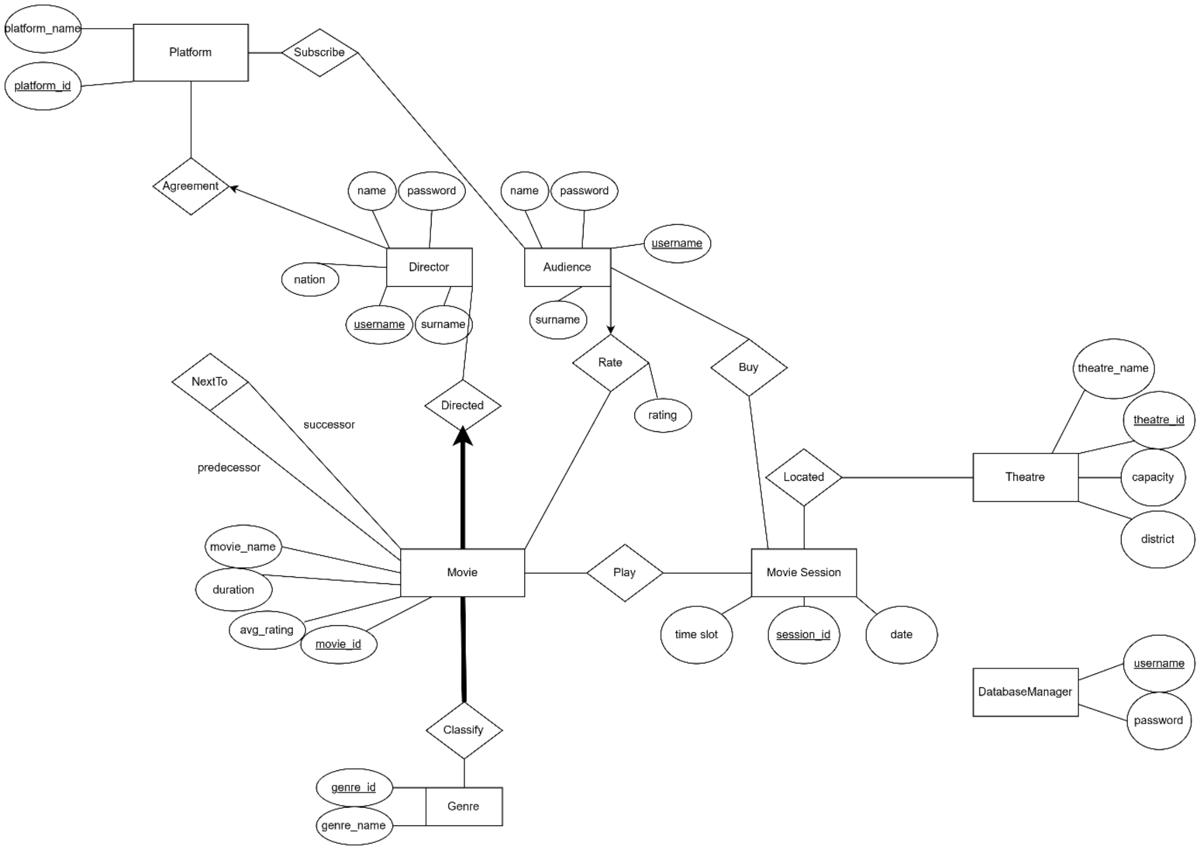
Buy(username: string, session_id: integer)

Agreement(username: string, platform_id: integer)

Subscribe(username: string, platform_id: integer)

The new version of create_table is uploaded if you want to check the relations.

Updated ER Diagram:


platform_name
platform_id
Platform
Subscribe
Agreement
name
password
name
password
username
Director
Audience
nation
username
surname
surname
NextTo
Directed
Rate
Buy
rating
theatre_name
theatre_id
successor
predecessor
Located
Theatre
capacity
district
movie_name
Movie
Play
Movie Session
duration
avg_rating
movie_id
time slot
session_id
date
username
DatabaseManager
password
Classify
genre_id
Genre
genre_name