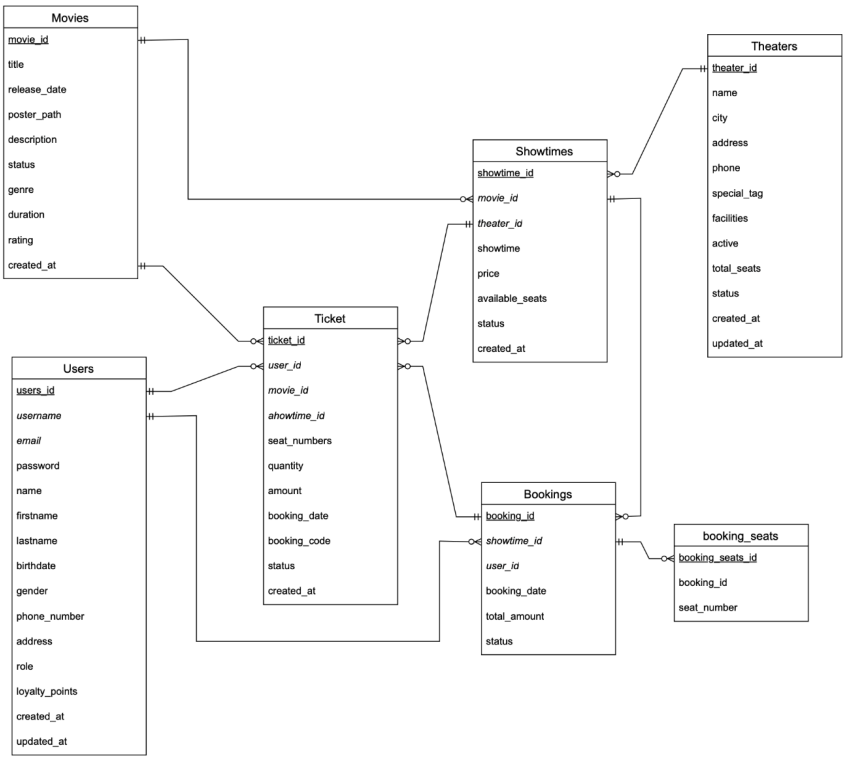
C.1. REVISED BUSINESS RULES AND ASSUMPTIONS

1. The user should be able to create an account with the correct email address, username, name, and password. On the creation of every user account, a unique identifier will be automatically generated.
2. Users must register with personal information including username, email, password, first name, last name, birthdate, gender, phone number, and address. Email and username must be unique within the system.
3. The system supports two types of user roles: regular users and administrators, determined by the "role" attribute in the Users entity.
4. Regular users can view movies, search for movies, view movie details, view showtimes, reserve tickets, make payments, receive e-tickets, view order history, and modify their profile information.
5. Administrators can manage movies (add, edit, delete), manage schedules, manage theaters, view sales reports, and manage user accounts if necessary.
6. A movie entry must include a title, release date, and basic description. Additionally, it may include genre, poster path, duration, rating, and status.
7. Each movie can have multiple showtimes across different theaters, and each showtime is for a specific movie in a specific theater.
8. Theaters have attributes such as name, city, address, phone number, special tags, facilities, and total number of seats available.
9. For each showtime, the system maintains information about available seats, price, and status.
10. When booking a ticket, the user selects a specific showtime, seats, and quantity. The system generates a unique booking code.
11. A booking can include multiple seats, and each seat is uniquely identified within a booking by its seat number.
12. The system tracks the status of bookings (e.g., pending, confirmed, canceled) and tickets (e.g., active, used, expired).
13. Users accumulate loyalty points based on their booking activities.
14. The system must record all ticket booking transactions for reporting and audit purposes, including timestamps for creation and updates.
15. The system must handle booking cancellations according to predefined policies.
16. The system must notify users of their booking and payment status.
17. All financial transactions must be recorded with appropriate details including the total amount paid.

C.2. REVISED ERD



The revised Entity Relationship Diagram for the movie ticket booking system consists of the following entities and their relationships:

- **Users** entity with primary key users_id
- **Movies** entity with primary key movie_id
- **Theaters** entity with primary key theater_id
- **Showtimes** entity with primary key showtime_id, foreign keys movie_id and theater_id
- **Bookings** entity with primary key booking_id, foreign keys user_id and showtime_id
- **Tickets** entity with primary key ticket_id, foreign keys booking_id, user_id, movie_id, and showtime_id
- **Booking_Seats** entity with primary key booking_seats_id, foreign key booking_id

Key relationships:

- Users make Bookings (one-to-many)
- Users have Tickets (one-to-many)
- Movies have Showtimes (one-to-many)
- Theaters have Showtimes (one-to-many)
- Showtimes have Bookings (one-to-many)
- Bookings have Tickets (one-to-many)
- Bookings have Booking_Seats (one-to-many)

C.3. RELATIONS (LOGICAL DESIGN / SCHEMA CONVERSION)

users (users_id, username, email*, password, firstname, lastname, birthdate, gender, phone_number, address, role, loyalty_points, created_at, updated_at)*

movies (movie_id, title, release_date, description, poster_path, genre, duration, rating, status, created_at)

theaters (theater_id, name, city, address, phone, special_tag, facilities, total_seats, status, created_at, updated_at)

showtimes (showtime_id, movie_id, theater_id*, showtime, available_seats, price, status, created_at)*
movie_id references movies
theater_id references theaters

bookings (booking_id, user_id, showtime_id*, booking_date, total_amount, status)*
user_id references users
showtime_id references showtimes

tickets (ticket_id, booking_id, user_id*, movie_id*, showtime_id*, seat_numbers, quantity, booking_code, status, created_at)*
booking_id references bookings
user_id references users
movie_id references movies
showtime_id references showtimes

booking_seats (booking_seats_id, booking_id, seat_number)*
booking_id references bookings

C.4. LIST OF FUNCTIONAL DEPENDENCIES RELATED TO EACH BUSINESS RULE

USERS

BR1: The user should be able to create an account with the correct email address, username, name, and password. On the creation of every user account, a unique identifier will be automatically generated.

- *users_id → {username*, email*, password, firstname, lastname, birthdate, gender, phone_number, address, role, loyalty_points, created_at, updated_at}*

BR2: Users must register with personal information including username, email, password, first name, last name, birthdate, gender, phone number, and address. Email and username must be unique within the system.

- *email → {users_id, username, email, password, firstname, lastname, birthdate, gender, phone_number, address, role, loyalty_points, created_at, updated_at}*
- *username → {users_id, username, email, password, firstname, lastname, birthdate, gender, phone_number, address, role, loyalty_points, created_at, updated_at}*

BR3: The system supports two types of user roles: regular users and administrators.

- *users_id → role*

BR13: Users accumulate loyalty points based on their booking activities.

- *users_id → loyalty_points*

BR14: The system must record all transactions with timestamps.

- *users_id → created_at*
- *users_id → updated_at*

MOVIES

BR6: A movie entry must include a title, release date, and basic description.

- *movie_id → {title, release_date, description, poster_path, genre, duration, rating, status, created_at}*

BR14: The system must record all transactions with timestamps.

- *movie_id → created_at*

THEATERS

BR8: Theaters have attributes such as name, city, address, phone number, special tags, facilities*, and total number of seats available.*

- *theater_id → {name, city, address, phone, special_tag, facilities, total_seats, status, created_at, updated_at}*

BR14: The system must record all transactions with timestamps.

- *theater_id → created_at*
- *theater_id → updated_at*

SHOWTIMES

BR7: Each movie can have multiple showtimes across different theaters, and each showtime is for a specific movie in a specific theater.

- *showtime_id → {movie_id*, theater_id*, showtime, available_seats, price, status, created_at}*

BR9: For each showtime, the system maintains information about available seats, price, and status.

- *showtime_id → available_seats*
- *showtime_id → price*
- *showtime_id → status*

BR14: The system must record all transactions with timestamps.

- *showtime_id → created_at*

BOOKINGS

BR10: When booking a ticket, the user selects a specific showtime. The system generates a unique booking code.

- *booking_id → {user_id*, showtime_id*, booking_date, total_amount, status}*

BR12: The system tracks the status of bookings.

- *booking_id → status*

BR14: The system must record all transactions with timestamps.

- *booking_id → booking_date*

BR17: All financial transactions must be recorded with appropriate details.

- *booking_id → total_amount*

TICKETS

BR10: When booking a ticket, the user selects a specific showtime, seats, and quantity. The system generates a unique booking code.

- *ticket_id → {booking_id*, user_id*, movie_id*, showtime_id*, seat_numbers, quantity, booking_code, status, created_at}*

BR12: The system tracks the status of tickets.

- *ticket_id → status*

BR14: The system must record all transactions with timestamps.

- *ticket_id → created_at*

BOOKING_SEATS

BR11: A booking can include multiple seats, and each seat is uniquely identified within a booking by its seat number.

- *booking_seats_id → {booking_id*, seat_number}*
- *{booking_id*, seat_number} → booking_seats_id**

C.5. NORMALIZATION (LOGICAL DESIGN)

Criteria:

- **1NF:** Has a Primary Key, no repeating attributes
- **2NF:** Every non-key attribute is functionally dependent on the PK
- **3NF:** There are no transitive functional dependencies
- **BCNF:** For all functional dependencies $\alpha \rightarrow \beta$, either $\alpha \rightarrow \beta$ is trivial ($\beta \subseteq \alpha$) or α is a superkey

USERS entity

- *users_id → {username, email, password, firstname, lastname, birthdate, gender, phone_number, address, role, loyalty_points, created_at, updated_at}*
- *email → {users_id, username, email, password, firstname, lastname, birthdate, gender, phone_number, address, role, loyalty_points, created_at, updated_at}*
- *username → {users_id, username, email, password, firstname, lastname, birthdate, gender, phone_number, address, role, loyalty_points, created_at, updated_at}*

1NF: Yes, it has a unique identifier for each row (users_id) and no repeating attributes.

2NF: Yes, every non-key attribute is functionally dependent on the PK (users_id).
3NF: Yes, there are no transitive functional dependencies.
BCNF: Yes, because:

1. Even though there are functional dependencies not from the primary key (email → users_id and username → users_id), both email and username are superkeys for the users entity as per BR2, which states they must be unique.
2. When considering the functional dependencies with determinants email and username, we can see these are trivial dependencies.

Therefore, the final relation for users is:

users (<u>users_id</u>, username, email*, password, firstname, lastname, birthdate, gender, phone_number, address, role, loyalty_points, created_at, updated_at)*

MOVIES entity

- *movie_id → {title, release_date, description, poster_path, genre, duration, rating, status, created_at}*

1NF: Yes, it has a unique identifier for each row (movie_id) and no repeating attributes.
2NF: Yes, every non-key attribute is functionally dependent on the PK (movie_id).
3NF: Yes, there are no transitive functional dependencies.
BCNF: Yes, there are no other functional dependencies not from the primary key.

Therefore, the final relation for movies is:

movies (<u>movie_id</u>, title, release_date, description, poster_path, genre, duration, rating, status, created_at)

THEATERS entity

- *theater_id → {name, city, address, phone, special_tag, facilities, total_seats, status, created_at, updated_at}*

1NF: Yes, it has a unique identifier for each row (theater_id) and no repeating attributes.
2NF: Yes, every non-key attribute is functionally dependent on the PK (theater_id).
3NF: Yes, there are no transitive functional dependencies.
BCNF: Yes, there are no other functional dependencies not from the primary key.

Therefore, the final relation for theaters is:

theaters (<u>theater_id</u>, name, city, address, phone, special_tag, facilities, total_seats, status, created_at, updated_at)

SHOWTIMES entity

- *showtime_id → {movie_id, theater_id, showtime, available_seats, price, status, created_at}*

1NF: Yes, it has a unique identifier for each row (showtime_id) and no repeating attributes.
2NF: Yes, every non-key attribute is functionally dependent on the PK (showtime_id).
3NF: Yes, there are no transitive functional dependencies.
BCNF: Yes, there are no other functional dependencies not from the primary key.

Therefore, the final relation for showtimes is:

showtimes (showtime_id, movie_id, theater_id*, showtime, available_seats, price, status, created_at)*
movie_id references movies
theater_id references theaters

BOOKINGS entity

- *booking_id → {user_id, showtime_id, booking_date, total_amount, status}*

1NF: Yes, it has a unique identifier for each row (booking_id) and no repeating attributes.
2NF: Yes, every non-key attribute is functionally dependent on the PK (booking_id).
3NF: Yes, there are no transitive functional dependencies.
BCNF: Yes, there are no other functional dependencies not from the primary key.

Therefore, the final relation for bookings is:

bookings (booking_id, user_id, showtime_id*, booking_date, total_amount, status)*
user_id references users
showtime_id references showtimes

TICKETS entity

- *ticket_id → {booking_id, user_id, movie_id, showtime_id, seat_numbers, quantity, booking_code, status, created_at}*

1NF: Yes, it has a unique identifier for each row (ticket_id) and no repeating attributes.
2NF: Yes, every non-key attribute is functionally dependent on the PK (ticket_id).
3NF: Yes, there are no transitive functional dependencies.
BCNF: Yes, there are no other functional dependencies not from the primary key.

Therefore, the final relation for tickets is:

tickets (ticket_id, booking_id, user_id*, movie_id*, showtime_id*, seat_numbers, quantity, booking_code, status, created_at)*
booking_id references bookings
user_id references users
movie_id references movies
showtime_id references showtimes

BOOKING_SEATS entity

- *booking_seats_id → {booking_id, seat_number}*
- *{booking_id, seat_number} → booking_seats_id*

1NF: Yes, it has a unique identifier for each row (booking_seats_id) and no repeating attributes.
2NF: Yes, every non-key attribute is functionally dependent on the PK (booking_seats_id).
3NF: Yes, there are no transitive functional dependencies.
BCNF: Yes, because even though there is a functional dependency {booking_id, seat_number} → booking_seats_id not from the primary key, the determinant {booking_id, seat_number} is a candidate key according to BR11, which states that each seat is uniquely identified within a booking by its seat number.

Therefore, the final relation for booking_seats is:

booking_seats (booking_seats_id, booking_id, seat_number)*
booking_id references bookings