UNOX FUNCTIONAL REQUIREMENTS AND DATABASE DESIGN

Requirements Document for UNOX Database Schema

1. Introduction The UNOX database is designed to manage an online movie ticket booking and food ordering system. It includes tables to handle screens, seats, movies, bookings, payments, food orders, and user memberships efficiently.

2. Objectives

- Provide structured storage for movie and show data.
- Enable seat selection and booking functionality.
- Support user management and membership rewards.
- Process payments and transactions securely.
- Manage food orders and delivery options.

3. Constraints and Assumptions

- A show cannot have more seats booked than its capacity.
- A user can have multiple bookings but cannot book the same seat twice for the same show.
- Payments must be linked to a valid booking.
- Food orders must be placed for an active booking.

4. Security and Performance Considerations

- Data integrity ensured with primary and foreign keys.
- Indexing on frequently queried fields (e.g., movie_id, user_id).
- Secure storage of sensitive payment details with encryption.
- Transaction logs for auditing purposes.

5. Future Enhancements

- Support for user authentication with role-based access.
- Dynamic pricing based on demand.
- Integration with third-party loyalty programs.
- Advanced seat selection with real-time availability updates.

6. Conclusion This database schema provides a comprehensive solution for managing a movie ticketing and food ordering system while ensuring data integrity, security, and efficiency.

Database Design

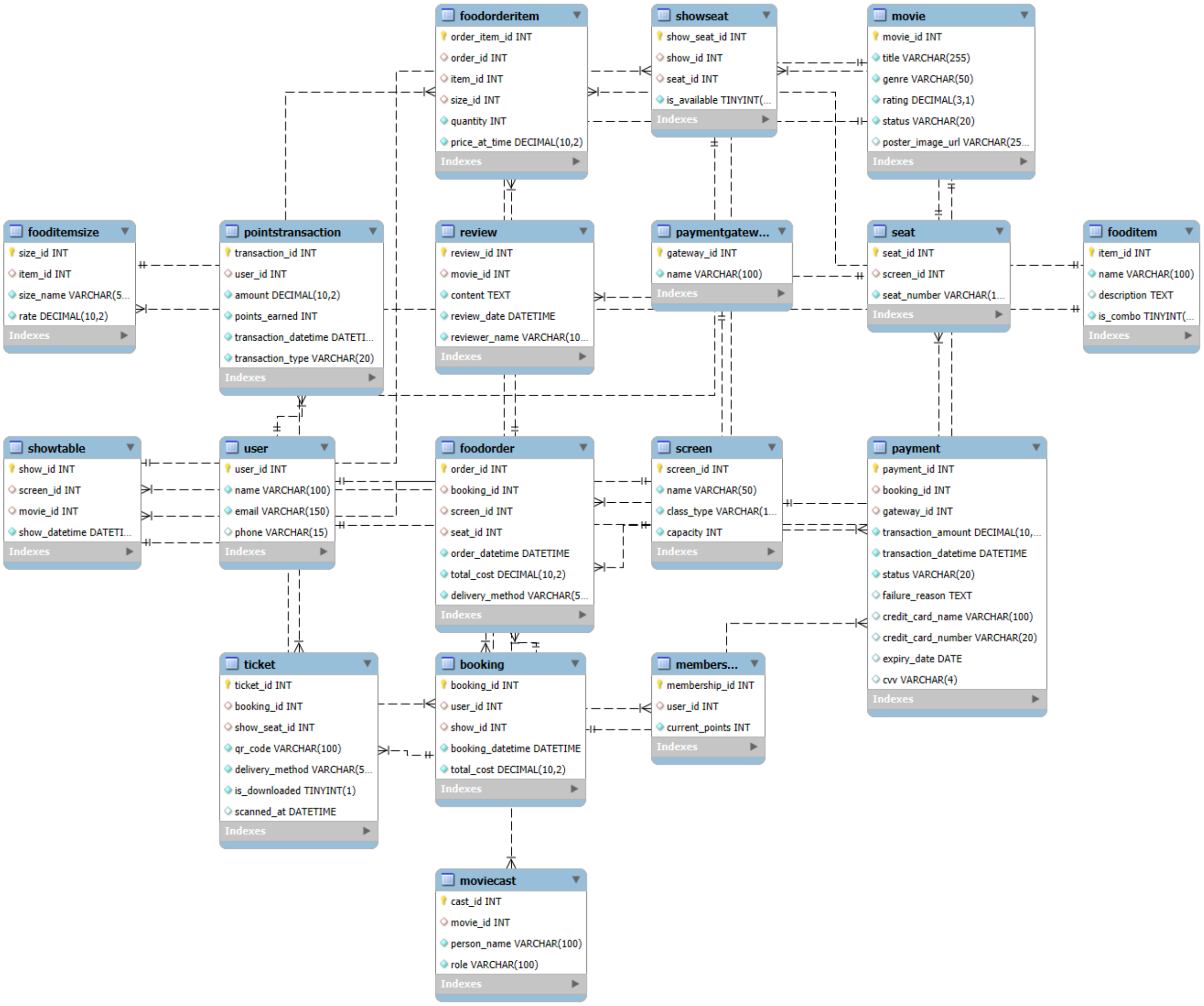

Table Structure

1. Screen

- screen_id (INT, PK, Auto Increment)
- name (VARCHAR(50), NOT NULL)
- class_type (VARCHAR(10), NOT NULL)
- capacity (INT, NOT NULL)

2. Seat

- seat_id (INT, PK, Auto Increment)
- screen_id (INT, FK → Screen)
- seat_number (VARCHAR(10), NOT NULL)

3. Movie

- movie_id (INT, PK, Auto Increment)
- title (VARCHAR(255), NOT NULL)
- genre (VARCHAR(50), NOT NULL)
- rating (DECIMAL(3,1), NOT NULL)
- status (VARCHAR(20), NOT NULL)
- poster_image_url (VARCHAR(255), NULL)

4. MovieCast

- cast_id (INT, PK, Auto Increment)
- movie_id (INT, FK → Movie)
- person_name (VARCHAR(100), NOT NULL)
- role (VARCHAR(100), NOT NULL)

5. Review

- review_id (INT, PK, Auto Increment)
- movie_id (INT, FK → Movie)
- content (TEXT, NOT NULL)
- review_date (DATETIME, NOT NULL)
- reviewer_name (VARCHAR(100), NOT NULL)

6. Show

- show_id (INT, PK, Auto Increment)
- screen_id (INT, FK → Screen)
- movie_id (INT, FK → Movie)
- show_datetime (DATETIME, NOT NULL)

7. ShowSeat

- show_seat_id (INT, PK, Auto Increment)
- show_id (INT, FK → Show)
- seat_id (INT, FK → Seat)
- is_available (BOOLEAN, NOT NULL, DEFAULT TRUE)

8. User

- user_id (INT, PK, Auto Increment)
- name (VARCHAR(100), NOT NULL)
- email (VARCHAR(150), NOT NULL)
- phone (VARCHAR(15), NULL)

9. Membership

- membership_id (INT, PK, Auto Increment)
- user_id (INT, FK → User)
- current_points (INT, NOT NULL, DEFAULT 0)

10. Booking

- booking_id (INT, PK, Auto Increment)
- user_id (INT, FK → User)
- show_id (INT, FK → Show)
- booking_datetime (DATETIME, NOT NULL)
- total_cost (DECIMAL(10,2), NOT NULL)

11. Ticket

- ticket_id (INT, PK, Auto Increment)
- booking_id (INT, FK → Booking)
- show_seat_id (INT, FK → ShowSeat)
- qr_code (VARCHAR(100), NOT NULL)
- delivery_method (VARCHAR(50), NOT NULL)
- is_downloaded (BOOLEAN, NOT NULL, DEFAULT FALSE)
- scanned_at (DATETIME, NULL)

12. PaymentGateway

- gateway_id (INT, PK, Auto Increment)
- name (VARCHAR(100), NOT NULL)

13. Payment

- payment_id (INT, PK, Auto Increment)
- booking_id (INT, FK → Booking)
- gateway_id (INT, FK → PaymentGateway)
- transaction_amount (DECIMAL(10,2), NOT NULL)
- transaction_datetime (DATETIME, NOT NULL)
- status (VARCHAR(20), NOT NULL)
- failure_reason (TEXT, NULL)
- credit_card_name (VARCHAR(100), NULL)
- credit_card_number (VARCHAR(20), NULL)
- expiry_date (DATE, NULL)
- cvv (VARCHAR(4), NULL)

14. FoodItem

- item_id (INT, PK, Auto Increment)
- name (VARCHAR(100), NOT NULL)
- description (TEXT, NULL)
- is_combo (BOOLEAN, NOT NULL, DEFAULT FALSE)

15. FoodItemSize

- size_id (INT, PK, Auto Increment)
- item_id (INT, FK → FoodItem)
- size_name (VARCHAR(50), NOT NULL)
- rate (DECIMAL(10,2), NOT NULL)

16. FoodOrder

- order_id (INT, PK, Auto Increment)
- booking_id (INT, FK → Booking)
- screen_id (INT, FK → Screen)
- seat_id (INT, FK → Seat)
- order_datetime (DATETIME, NOT NULL)
- total_cost (DECIMAL(10,2), NOT NULL)
- delivery_method (VARCHAR(50), NOT NULL)

17. FoodOrderItem

- order_item_id (INT, PK, Auto Increment)
- order_id (INT, FK → FoodOrder)
- item_id (INT, FK → FoodItem)
- size_id (INT, FK → FoodItemSize)
- quantity (INT, NOT NULL)
- price_at_time (DECIMAL(10,2), NOT NULL)

18. PointsTransaction

- transaction_id (INT, PK, Auto Increment)
- user_id (INT, FK → User)
- amount (DECIMAL(10,2), NOT NULL)
- points_earned (INT, NOT NULL)
- transaction_datetime (DATETIME, NOT NULL)
- transaction_type (VARCHAR(20), NOT NULL)