Itahari International College

Introduction to Information Systems

Database Management System Assignment

Module Leader: Binay Koirala
Submission Deadline: December 1st, 2024, 01:00 PM

Scenario

DreamForge Gaming Studios is an emerging game development company that creates mobile and PC games. The studio manages multiple game projects and has recently launched their flagship game "Mystic Realms," which has gained significant popularity. The company needs a database system to manage their game development and player interactions efficiently.
The studio employs developers, artists, and testers who work on different game projects. Each employee has specific details including their expertise, joining date, salary grade, department, and contact information. When working on games, they track project timelines, development phases, budget allocation, target platforms, and genre classifications.

Players create detailed profiles to access games, including their personal information, gaming preferences, account creation date, and gaming history. Each player maintains a wallet for in-game purchases, tracking transaction history, current balance, purchase dates, and payment methods. The studio organizes regular tournaments with specific prize pools, entry requirements, start and end dates, participant limits, and tournament rules.

To support their players, the company has implemented a help desk system where players can submit support tickets. Each ticket includes priority levels, submission timestamps, issue categories, current status, and resolution details. They also offer a premium membership system where players can subscribe monthly to receive exclusive content.

Questions

1. **Business Rules Analysis**
    - List all business rules that will govern how data is stored and managed in the database
    - Consider rules for player registration, tournament participation, employee project assignments, and support ticket handling
    - Minimum of 10 business rules required
2. **System Assumptions**
    - Document all assumptions about how the system operates
    - Include assumptions about tournament organization, employee assignments, and player subscriptions
    - Minimum of 8 assumptions required
3. **Entity Relationship Diagram** Your ERD must include at minimum the following entities with at least 5 attributes each:
    - Employee (emp_id, name, email, phone, department, joining_date, salary_grade, etc.)
    - Game (game_id, title, release_date, genre, platform, budget, status, etc.)
    - Player (player_id, username, email, registration_date, membership_status, wallet_balance, etc.)
    - Tournament (tournament_id, name, start_date, end_date, prize_pool, max_participants, etc.)
    - Support_Ticket (ticket_id, submit_date, priority, status, description, resolution, etc.)

Additional entities should be created as needed to handle relationships and additional functionality.

4. **Relational Diagram**
    - Convert your ERD into a relational diagram
    - Show all tables with their attributes and data types
    - Indicate primary and foreign keys clearly
    - Use proper crow's foot notation for relationships
    - Break down all many-to-many relationships into junction tables
5. **Table Creation** Write SQL queries to create tables with appropriate:
    - Primary Keys for each table (Both single-column and composite where necessary)
    - Foreign Keys with proper references and ON DELETE/UPDATE actions
    - Not Null constraints on required fields (e.g., employee name, player username)
    - Check constraints (e.g., salary_grade between 1 and 10, tournament prize_pool > 0)

- Default values (e.g., ticket status = 'Open', player membership_status = 'Regular')
- Unique constraints (e.g., employee email, player username)

6. **Data Manipulation Queries**

   a) Insert the following data:

   - Add 5 new games with different status and genres
   - Register 5 players with different membership types
   - Add 5 employees with different roles and departments
   - Create 5 tournaments with different prize pools
   - Record 5 support tickets with varying priority levels

   b) Update the following records:

   - Change the status of two games
   - Update three players' membership status
   - Modify two employees' departments
   - Update four support ticket status

   c) Delete the following records:

   - Remove three cancelled tournaments
   - Delete two inactive player accounts
   - Remove three resolved support tickets

   d) Write SELECT queries to:

   - List all games by release date
   - Show all employees in a specific department
   - Display all high-priority support tickets
   - Find players who participated in at least 3 tournaments
   - Calculate total prize money won by each player
   - Show all games developed by a specific employee
   - List all open support tickets older than 7 days
   - Display tournament participation statistics

Instructions

1. **Understanding Relationships:**
    - Each entity must have a primary key (e.g., emp_id for Employee table)
    - Define appropriate foreign keys (e.g., game_id in Tournament table to reference the Game)
    - Many-to-many relationships must be broken down using junction tables Example: Employee and Game relationship:

```
CREATE TABLE Employee_Game (
assignment_id INT PRIMARY KEY,
emp_id INT,
game_id INT,
assignment_date DATE,
role VARCHAR(50),
FOREIGN KEY (emp_id) REFERENCES Employee(emp_id),
FOREIGN KEY (game_id) REFERENCES Game(game_id)
);
```

2. **Table Design:**
    - Begin with core entities (Employee, Game, Player, Tournament, Support_Ticket)
    - Add necessary junction tables
    - Include at least 5 relevant attributes for each entity
    - Choose appropriate data types for all attributes
3. **Query Writing:**
    - Follow consistent naming conventions
    - Ensure proper data integrity
    - Test queries with sample data