

CS3700: Assignment 1

Group-16

Akshat Meena
(CS19B052)

Rohit Bhagat
(CS19B038)

Siddharth Singh
(CS19B072)

Hemank Dahiya
(CS19B058)

Prathmesh Kharade
(CS19B037)

Domain

Video Game Database

Broad Purpose

The purpose of this database is to store profiles of all the players with their inventory, friends, missions completed, membership, match logs and other related data. This database is designed for a game with 1 vs 1 or team match modes with missions and item reward functions.

Domain Description

In a game there is a need to store different information such as player profiles, team information, items, missions, etc. There are several players in the game. Each player has a name, list of titles, points earned based on wins and losses, level depending on the points earned by him/her. Each player also has an inventory consisting of some items they own. Every item is of a particular type. Each item has a name, and stats such as attack and defense. Each player has a list (possibly empty) of missions that (s)he has accomplished. Each mission also has name, and description. When a player completes a mission he is given a reward of an item type. There are different levels of membership available that a player can purchase to get extra rewards in the game. Each membership has a unique name, price and some rewards associated with it. Also, the players may have some friends among each other. Players can form new teams and also join existing teams. A team consists of a number of players. Each team has a name, members, and some points earned by it based on wins and losses. Matches can be held individually as well as team-wise and a winner is decided after each match.

ER Model

Entities

1. **Player** - Entity to store the profile of each player

- **ID** (Key) - Primary key for the player entity
 - Type: Simple Single-Valued
 - Domain: Integer (≥ 1)
- **Name** - "In-game" name of the player
 - Type: Simple Single-Valued
 - Domain: String (max 20 chars)
- **Level** - Level of the player derived from points
 - Type: Derived Single-Valued
 - From: Points
 - Domain: Integer (max 999, min 1)
- **Title** - Titles acquired by the player
 - Type: Simple Multi-Valued
 - Domain: String (max 20 chars)
- **Points** - number of wins and losses

- Type: Composite Single-Valued
 - Components:
 - **Wins** - number of matches won
 - Type: Simple Single-Valued
 - Domain: Integer (≥ 0)
 - **Losses** - number of matches lost
 - Type: Simple Single-Valued
 - Domain: Integer (≥ 0)
2. **ItemType** - Entity to store information of different item types
- **Type** (Key) - Primary Key for the ItemType entity
 - Type: Simple Single-Valued
 - Domain: String
3. **Item** - Entity to store information of each item type entity. A weak entity with ItemType as owner entity
- **ID** (Key) - Primary key of Item entity
 - Type: Simple Single-Valued
 - Domain: Integer (≥ 1)
 - **Name** - Item name
 - Type: Simple Single-Valued
 - Domain: String (max 20 chars)
 - **Stats** - Attributes of the item
 - Type: Composite Single-Valued
 - Components:
 - **Attack** - Attack stats of the item
 - Type: Simple Single-Valued
 - Domain: Integer (≥ 0)
 - **Defense** - Defense stats of the item
 - Type: Simple Single-Valued
 - Domain: Integer (≥ 0)
4. **Mission** - Entity to store information of each mission
- **ID** (Key) - Primary key for Mission entity
 - Type: Simple Single-Valued
 - Domain: Integer (≥ 1)
 - **Name** - Name of the mission
 - Type: Simple Single-Valued
 - Domain: String (max 20 chars)
 - **Description** - Details of the mission
 - Type: Simple Single-Valued
 - Domain: String (max 256 chars)
5. **Team** - Entity for each team formed by players
- **ID** (Key) - Primary key for Team entity

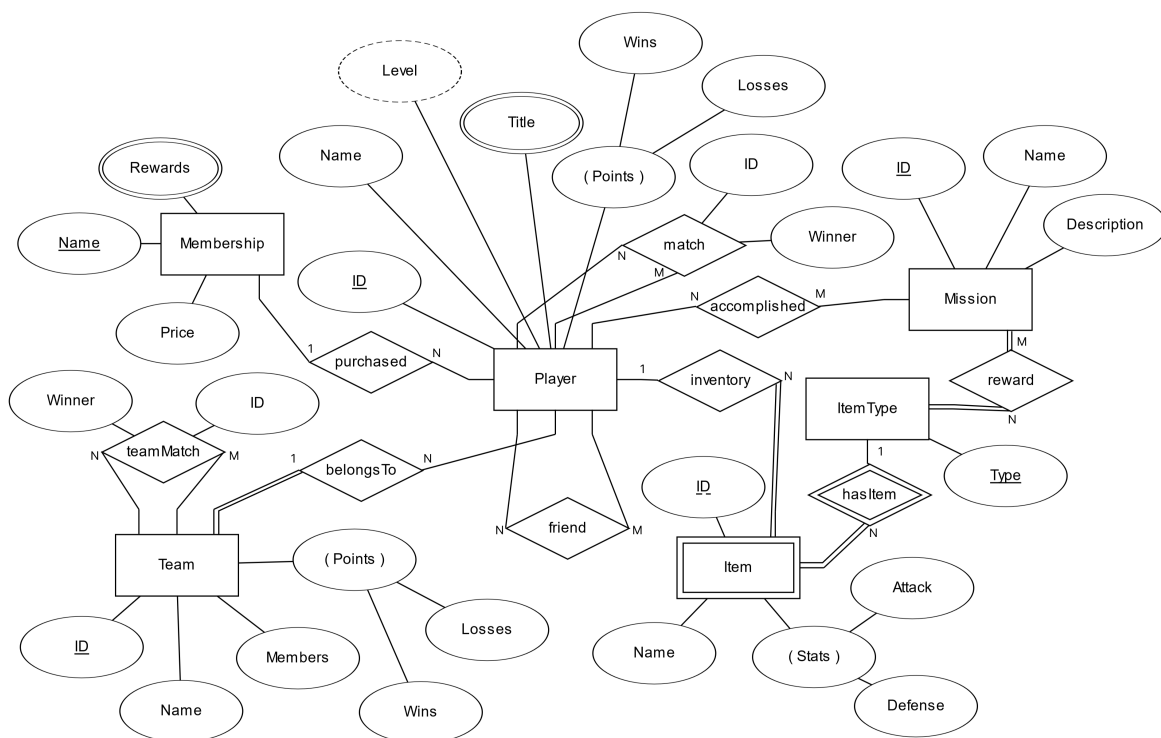
- Type: Simple Single-Valued
 - Domain: Integer (≥ 1)
 - **Name** - Team name
 - Type: Simple Single-Valued
 - Domain: String (max 20 chars)
 - **Members** - Number of players in a team
 - Type: Simple Single-Valued
 - Domain: Integer ($\geq 1, \leq 10$)
 - **Points** - number of wins and losses
 - Type: Composite Single-Valued
 - Components:
 - **Wins** - Total team matches won
 - Type: Simple Single-Valued
 - Domain: Integer (≥ 0)
 - **Losses** - Total team matches lose
 - Type: Simple Single-Valued
 - Domain: Integer (≥ 0)
6. **Membership** - Entity for the type of membership
- **Name** (Key) - Membership Type (also primary key to Membership Entity)
 - Type: Simple Single-Valued
 - Domain: String
 - **Price** - Price for purchasing membership
 - Type: Simple Single-Valued
 - Domain: Integer
 - **Rewards** - Rewards granted for purchasing the membership
 - Type: Simple Multi-Valued
 - Domain: Strings (max 20 chars)

Relationships

1. **friend** (Player - Player | M:N) - Relationship between 2 players to make friends with each other in game
2. **match** (Player - Player | M:N) - 1 vs 1 match played between 2 players
 - **ID**
 - Type: Simple Single-Valued
 - Domain: Integer (≥ 0)
 - **Winner**
 - Type: Simple Single-Valued
 - Domain: Integer (≥ 0)
3. **inventory** (Player - Item | 1:N) - Each player possesses a number of items (can be 0) in their inventory. Total participation of Item entity
4. **hasItem** (ItemType - Item | 1:N) - Each Item Type have a variety of items
5. **accomplished**(Player - Mission | M:N) - Missions completed by the player

6. **reward** (Mission - ItemType | M:N) - Rewards given for completion of a mission. Total participation of both entity
7. **purchased** (Player - Membership | N:1) - Players buy memberships for extra rewards
8. **teamMatch** (Team - Team | M:N) - Team match played between 2 teams
 - **ID**
 - Type: Simple Single-Valued
 - Domain: Integer (≥ 0)
 - **Winner** -
 - Type: Simple Single-Valued
 - Domain: Integer (≥ 0)
9. **belongsTo** (Player - Team | N:1) - Relation of players with their teams. Total participation of Team entity

ER Diagram



Relational Database Scheme

Relations

1. player

<u>id</u>	name	wins	losses	level	team	membership
-----------	------	------	--------	-------	------	------------

- Foreign keys
 - team → team.id
 - membership → membership.name

2. friend

--	--

<u>player</u>	<u>friend</u>
---------------	---------------

- Foreign keys
 - player → player.id
 - friend → player.id

3. match

<u>id</u>	player1	player2	winner
-----------	---------	---------	--------

- Foreign keys
 - player1 → player.id
 - player2 → player.id
 - winner → player.id

4. playerTitle

<u>player</u>	<u>title</u>
---------------	--------------

- Foreign keys
 - player → player.id

5. mission

<u>id</u>	name	description
-----------	------	-------------

6. accomplished

<u>player</u>	<u>mission</u>
---------------	----------------

- Foreign keys
 - player → player.id
 - mission → mission.id

7. reward

<u>mission</u>	<u>type</u>
----------------	-------------

- Foreign keys
 - type → itemType.type
 - mission → mission.id

8. itemType

<u>type</u>

9. item

<u>type</u>	<u>id</u>	name	attack	defense	owner
-------------	-----------	------	--------	---------	-------

- Foreign keys
 - type → itemType.type
 - owner → player.id

10. membership

<u>name</u>	price
-------------	-------

11. membershipReward

<u>membership</u>	<u>reward</u>
-------------------	---------------

- Foreign keys
 - membership → membership.name

12. team

<u>id</u>	name	members	wins	losses
-----------	------	---------	------	--------

13. teamMatch

<u>id</u>	team1	team2	winner
-----------	-------	-------	--------

- Foreign keys
 - team1 → team.id
 - team2 → team.id
 - winner → team.id

Diagram

