Project Summary

Apr 14, 2024

/\*

\*\* Author: Jaykumar Suthar

\*\* SQL Server Version: Microsoft Azure SQL Edge Developer (RTM) - 15.0.2000.1574 (ARM64)

\*\* Jan 25 2023 10:36:08 Copyright (C) 2019 Microsoft Corporation Linux (Ubuntu 18.04.6 LTS aarch64) <ARM64>

\*\*

\*\* Project History

\*\* Date Created Comments

\*\*

\*\* 04/24/2024 Create Database

\*\* 04/24/2024 Create Tables

\*\* 04/24/2024 Populate Tables

\*\* 04/24/2024 View Populated Tables

\*\* 04/24/2024 Create View

\*\* 04/24/2024 Check the View

\*\* 04/24/2024 Create The Audit Table

\*\* 04/24/2024 Create Trigger

\*\* 04/24/2024 Check the GameAudit table with Insert-Update-Delete Queries

\*\* 04/24/2024 Create Stored Procedures

\*\* 04/24/2024 Check Stored Procedures

\*\* 04/24/2024 Create User Define Function

\*\* 04/24/2024 Check User Define Function

\*\* 04/24/2024 Create Cursor

\*\*

\*/

**Project Summary**

**1. Project Details:**

**Project Name:** Gaming Database Management System

**Description:** The Gaming Database is designed to manage information related to gaming activities, including players, games, achievements, and reviews. It aims to provide a comprehensive platform for tracking gaming data, facilitating game development, and enhancing the gaming experience for users.

* ***Task: Most of the tables would be dimension (lookup) tables. You must have at least one or two fact tables (transactional tables).***

In a database context, "Transactional Tables" typically refer to tables that record individual transactions or events, often associated with specific actions or operations. In the context of a gaming database, the Gameplay and PlayerInteractions tables can be considered transactional because they record specific interactions or transactions within the gaming environment.

For example, the Gameplay table tracks details of each gameplay session, such as when it started and ended, the duration, and the player's score. These are discrete events or transactions within the game environment.

Similarly, the PlayerInteractions table could track interactions between players, such as adding friends, sending messages, or other in-game interactions. Each row in this table represents a specific interaction between players, which can be seen as a transaction within the game's social or interactive aspects.

**2. Table Details:**

**Players Table:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Size** | **Null/Not Null** | **Constraint** | **PK** | **Purpose** |
| player\_id | INT |  | Not Null | Primary Key | Yes | Unique identifier for each player |
| name | VARCHAR | 255 | Not Null |  |  | Name of the player |
| email | VARCHAR | 255 | Not Null | Unique |  | Email of the player (used for login and communication) |
| date\_of\_birth | DATE |  | Not Null |  |  | Date of birth of the player |

**Players Table:** Essential for managing player accounts and personal information.

**Reason:** Stores player information such as player ID, name, email, and date of birth.

**Games Table:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Size** | **Null/Not Null** | **Constraint** | **PK** | **Purpose** |
| game\_id | INT |  | Not Null | Primary Key | Yes | Unique identifier for each game |
| title | VARCHAR | 255 | Not Null |  |  | Title of the game |
| genre | VARCHAR | 255 |  |  |  | Genre of the game |
| release\_date | DATE |  |  |  |  | Release date of the game |
| developer | VARCHAR | 255 |  |  |  | Name of the developer or game studio |

**Games Table:** Stores details about each game, enabling easy access and reference.

**Reason:** Contains details about the games, including game ID, title, genre, release date, and developer information.

**Achievements Table:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Size** | **Null/Not Null** | **Constraint** | **PK** | **Purpose** |
| achievement\_id | INT |  | Not Null | Primary Key | Yes | Unique identifier for each achievement |
| player\_id | INT |  | Not Null | Foreign Key | Yes | Identifier of the player who achieved the achievement |
| game\_id | INT |  | Not Null | Foreign Key | Yes | Identifier of the game for which the achievement is achieved |
| description | TEXT |  | Not Null |  |  | Description of the achievement |
| date\_achieved | DATE |  | Not Null |  |  | Date when the achievement was achieved |

**Achievements Table:** Tracks player achievements, enhancing engagement and motivation.

**Reason:** Tracks achievements within games, linking them to specific players and games, with fields for achievement ID, player ID, game ID, description, and date achieved.

**Reviews Table:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Size** | **Null/Not Null** | **Constraint** | **PK** | **Purpose** |
| review\_id | INT |  | Not Null | Primary Key | Yes | Unique identifier for each review |
| player\_id | INT |  | Not Null | Foreign Key | Yes | Identifier of the player who wrote the review |
| game\_id | INT |  | Not Null | Foreign Key | Yes | Identifier of the game for which the review is written |
| rating | DECIMAL | 2,1 | Not Null |  |  | Rating given by the player for the game (e.g., 4.5 out of 5) |
| review\_content | TEXT |  | Not Null |  |  | Content of the review |

**Reviews Table:** Provides player feedback, aiding in game improvement and decision-making.

**Reason:** Stores game reviews submitted by players, with fields for review ID, player ID, game ID, rating, and review content.

**Inventory Table:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Size** | **Null/Not Null** | **Constraint** | **PK** | **Purpose** |
| inventory\_id | INT |  | Not Null | Primary Key | Yes | Unique identifier for each inventory item |
| player\_id | INT |  | Not Null | Foreign Key | Yes | Identifier of the player who owns the item |
| game\_id | INT |  | Not Null | Foreign Key | Yes | Identifier of the game for which the item belongs |
| item\_name | VARCHAR | 255 | Not Null |  |  | Name of the item |
| quantity | INT |  | Not Null |  |  | Quantity of the item |
| date\_acquired | DATE |  | Not Null |  |  | Date when the item was acquired |

**Inventory Table:** Manages player inventories, crucial for in-game progression and item management.

**Reason:** Manages player inventories, linking them to specific players and games, with fields for inventory ID, player ID, game ID, item name, quantity, and date acquired.

**Friends Table:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Size** | **Null/Not Null** | **Constraint** | **PK** | **Purpose** |
| friendship\_id | INT |  | Not Null | Primary Key | Yes | Unique identifier for each friendship |
| player\_id1 | INT |  | Not Null | Foreign Key | Yes | Identifier of the first player in the friendship |
| player\_id2 | INT |  | Not Null | Foreign Key | Yes | Identifier of the second player in the friendship |
| date\_added | DATE |  | Not Null |  |  | Date when the friendship was added |

**Friends Table:** Facilitates social interactions between players, enhancing the gaming community.

**Reason:** Keeps track of player friendships, with fields for friendship ID, player ID 1, player ID 2, and date added.

**Transactions Table:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Size** | **Null/Not Null** | **Constraint** | **PK** | **Purpose** |
| transaction\_id | INT |  | Not Null | Primary Key | Yes | Unique identifier for each transaction |
| player\_id | INT |  | Not Null | Foreign Key | Yes | Identifier of the player who made the transaction |
| game\_id | INT |  | Not Null | Foreign Key | Yes | Identifier of the game for which the transaction is made |
| transaction\_type | VARCHAR | 255 | Not Null |  |  | Type of transaction (e.g., Purchase, Sale, Gift) |
| amount | DECIMAL | 10,2 | Not Null |  |  | Amount of the transaction |
| date | DATE |  | Not Null |  |  | Date when the transaction occurred |

**Transactions Table:** Tracks player transactions, ensuring transparency and accountability.

**Reason:** Records transactions within the gaming system, including purchases and in-game transactions, with fields for transaction ID, player ID, game ID, transaction type, amount, and date.

**Events Table:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Size** | **Null/Not Null** | **Constraint** | **PK** | **Purpose** |
| event\_id | INT |  | Not Null | Primary Key | Yes | Unique identifier for each event |
| event\_name | VARCHAR | 255 | Not Null |  |  | Name of the event |
| description | TEXT |  | Not Null |  |  | Description of the event |
| date | DATE |  | Not Null |  |  | Date of the event |
| game\_id | INT |  | Not Null | Foreign Key | Yes | Identifier of the game associated with the event |

**Events Table:** Manages gaming events, fostering community engagement and competition.

**Reason:** Manages gaming events, including tournaments and special events, with fields for event ID, event name, description, date, and game ID.

**Gameplay Table:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Size** | **Null/Not Null** | **Constraint** | **PK** | **Purpose** |
| gameplay\_id | INT |  | Not Null | Primary Key | Yes | Unique identifier for each gameplay |
| player\_id | INT |  | Not Null | Foreign Key | Yes | Identifier of the player who played the game |
| game\_id | INT |  | Not Null | Foreign Key | Yes | Identifier of the game that was played |
| start\_time | DATETIME |  | Not Null |  |  | Start time of the gameplay |
| end\_time | DATETIME |  | Not Null |  |  | End time of the gameplay |
| duration\_seconds | INT |  | Not Null |  |  | Duration of the gameplay in seconds |
| score | INT |  | Not Null |  |  | Score achieved in the gameplay |

**Gameplay Table:** Tracks game sessions, including start and end times, duration, and player scores.

**Reason:** This table stores data about each gameplay session, helping analyze player behavior and game performance.

**PlayerInteractions Table:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Size** | **Null/Not Null** | **Constraint** | **PK** | **Purpose** |
| interaction\_id | INT |  | Not Null | Primary Key | Yes | Unique identifier for each interaction |
| player\_id1 | INT |  | Not Null | Foreign Key | Yes | Identifier of the first player in the interaction |
| player\_id2 | INT |  | Not Null | Foreign Key | Yes | Identifier of the second player in the interaction |
| interaction\_type | VARCHAR | 255 | Not Null |  |  | Type of interaction (e.g., Friend Request, Message, Game Invitation) |
| interaction\_date | DATE |  | Not Null |  |  | Date of the interaction |

**PlayerInteractions Table:** Manages interactions between players, including friend requests, messages, and game invitations.

**Reason:** The PlayerInteractions table tracks various interactions between players, providing insights into player relationships and engagement within the gaming community.

Overall, this database system aims to enhance the gaming experience by providing a centralized platform for managing gaming-related information, fostering community interaction, and improving game development through player feedback and engagement tracking.

*A screenshot of a computer

Description automatically generatedER Diagram*