Name: Osiame Moloro (pt.2022.y4c8x2)

Project 1: MMORPG

Date Modified: 07 February 2023

Normalization

Normalization is essential when designing a large database since adherence to a set of steps is required and it is then less common for human error to occur, it is also useful in understanding business rules and finding attributes and entities. It is guaranteed that is the normalization process id fully understood and followed, the data will be laid out efficiently.

We will now start with the Zero Normal Form (0NF) where we collect all the raw data and finding the primary key, then we will continue to the First Normal Form (1NF) where we will remove the repeating groups, after that we will continue to the Second Normal Form (2NF) where we will remove the part-key dependencies then finally we will do the Third Normal Form (3NF) where we will remove inter-data dependencies.

Zero Normal Form (0NF)

0NF
(PK)accountName
accountPassword
accountTime
accountStatus
playerName
playerTeam
playerSkillLevel
personEmail
personName
personCountry
itemNum
itemName
itemDesription
errorNum
errorType
errorDescription
quantity

First Normal Form (1NF)

1NF

(PK)accountName	PK	playerName(FK)
accountPassword		itemNum
accountTime		itemName
accountStatus		itemDescription
personEmail		quantity
personName		
personCountry		
errorNum		
errorType		
errorDescription		

Second Normal Form (2NF)

2NF

(PK)accountName	PK	playerName(FK)	(PK)itemNum
accountPassword		itemNum(FK)	itemName
accountTime		quantity	itemDescription
accountStatus			
personEmail			
personName			
personCountry			
errorNum			
errorType			
errorDescription			

Third Normal Form (3NF)

2NF

(PK)accountName
accountPassword
accountTime
accountStatus
personEmail(FK)

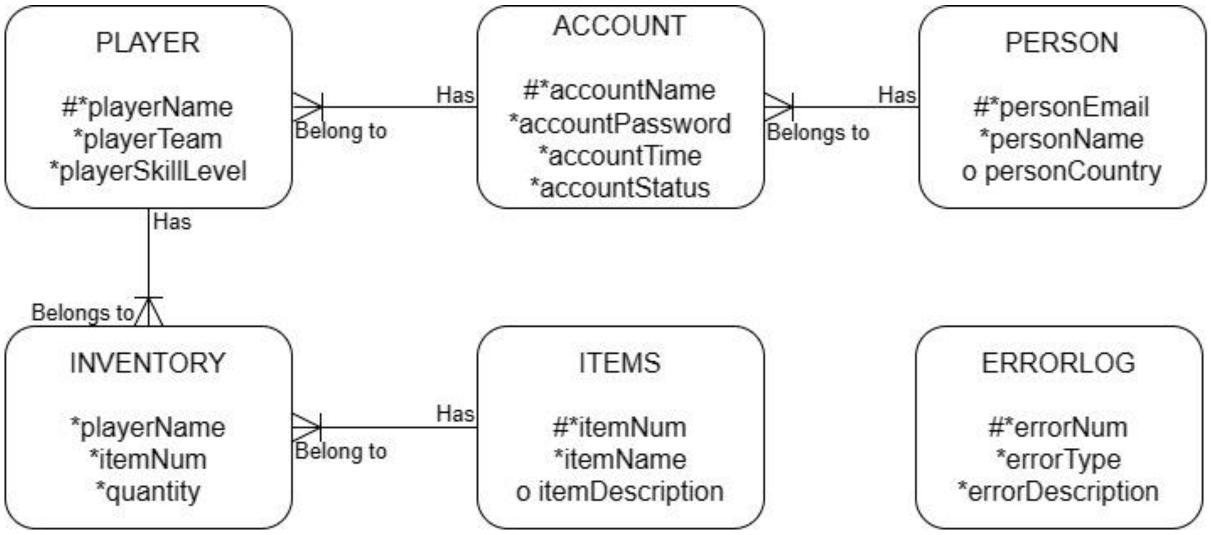
$\text{PK}\begin{cases}\text{playerName}\\ \text{itemNum}\end{cases}$
quantity

(PK)itemNum
itemName
itemDescription

(PK)personEmail
personName
personCountry

(PK)errorNum
errorType
errorDescription

The ER Diagram



Views

Views can be thought of as virtual tables or stored SELECT statements. The only difference between a view and a select statement is that a view does not actually contain any values. The values that appear to be in a view are pointers to the real values in a table or tables.

The first view that I had created on the database was to view all the blocked accounts from the database, blocked accounts are the accounts that users did not buy days for until they reached 0 days.

The second view that I created on the database was to view the top skilled players on the game and their account details, players gain skills as they play the game and the skill level gets incremented.

The third view that I created on the database was to view the 20 most stacked items and also include the players whose inventory are stacked in, Items that are the same can stack on the same slot inside the inventory.

The forth view that I created on the database was to view the five most popular items that are contained inside the player's inventory also including the number of characters that have the item, players can have more than one item inside their inventory but they have space or only up to 8 items.

Procedures

Stored procedures are a set of SQL statements typically grouped together to perform a specific routine. They usually provide the best performance because they are pre-compiled, i.e., they do not have to be compiled every time they are used. There is no need to write stored procedures repeatedly; hence they increase reusability of code.

The first procedure that I created (spRegister) was to register a new account in the database, the procedure checks whether the account does not already exist and if it does it will give us an error and then store the error details in the Error Log table it also checks whether the account time is a positive number e.g. (1, 6, 17).

The second procedure that I created (spAddTime) was to add time to the records in the Account table when users buy more time, it also checks whether the user-specified account exists. Users need to buy time to play the game or else their accounts will be blocked, when they buy time again their accounts will be unblocked.

The third procedure that I created (spAddItem) was to add items in the player's inventory, as I have mentioned, characters can have more than one item in their inventory, but they may have up to only 8 items, it also checks whether the player exists in the database and if there is enough space inside the inventory.

The fourth procedure that I had created (spAddChar) was to add a player to a specified account, it checks whether the specified account exists, an account can have one or more player to it.

The fifth procedure that I had created (spSendLetter) was to send a letter via email to give the user a notice of how much they have left on their account.

Triggers

A trigger is a special type of stored procedure that is automatically based on occurrence of a database event. Unlike stored procedures, triggers are only executed when a user or application attempts to modify data using Data Manipulation Language (DML).

The first trigger (tr_account) that I created on the database displays a message to the user, informing the user that a new record in the Account table has been added, this trigger goes together with the first procedure that I created (spRegister).

The second trigger (tr_accountUpdate) that I created displays a message to the user, informing the user that a record in the Account table has been updated, this trigger goes hand in hand with the second procedure that I created (spAddTime).

The third trigger (tr_player) that I created displays a message to the user, informing the user that a record in the Player table has been added, this trigger goes together with the fourth procedure that I created (spAddChar).

The fourth trigger (tr_playerUpdate) that I created displays a message to the user, informing the user that a record in the Player table has been updated, this trigger will be used when the player skill level is added or when a player changes a team.

The fifth trigger (tr_inventory) that I created displays a message to the user, informing the user that a record has been added in the Inventory table, this trigger works together with the third procedure that I created (spAddItem).

The sixth trigger (tr_errorLog) that I created displays a message to the user, informing the user that a record has been added in the ErrorLog, this trigger will be used when ever there is an error while the database checks and validates user input.

Indexes

An index is a data structure that improves data retrieval speed in a database table. Indexes can be categorised into two primary types: clustered and non-clustered. <u>Note that in my project I only created non-clustered indexes.</u>

The first index that I created (idx_account) was on the accountStatus column in the Account table, I believe that When the accountStatus column is indexed it will improve data retrieval in the Account table.

The second index that I created (idx_person) was on the personName column in the Person table, I believe that when the personName column is indexed it will improve data retrieval in the Person table.

The third index that I created (idx_player) was on the playerTeam column in the Player table, I believe that when the playerTeam column is indexed it will increase data retrieval speeds in the Player table.

The forth index that I created (idx_inventory) was on the quantity column in the Inventory table, I believe that when the quantity column is indexed it will increase the data retrieval speeds in the Inventory table .

The fifth index that I created (idx_item) was on the itemName column in the Item table, I believe that when the itemName column is indexed it will increase the data retrieval speeds in the Item table.

The sixth index that I created (idx_errorLog) was on the errorType column in the ErrorLog table, I believe that when the errorType column is index it will improve the data retrieval in the ErrorLog table.