

Author: Jonathan Otshudi

Date: 2024-03-08

Description:

This is documentation on how to set up and use the mmorpg database and its associated views and stored procedures.

Database Management (MySQL) Project

Documentation of the mmorpg Database

Set up

Unzip the folder titled And retrieve the required sql script files. Execute the script files in the following order:

1. Open and execute the create mmorpg script file to create the database, its tables and the indexes.
2. Open and execute the create populate mmorpg script file which will insert all the sample data for the mmorpg database tables.
3. Open and execute the create views script file, this will create all the required views for the database. Instruction on how to use the views created a provided latter in this document.
4. Open and execute the create stored procedures script, file this wilc create all the required stored procedures for the database. Instructions on how to use the stored procedures will be provided later in this document.

Using Views

The mmorpg database offers four viewpoints; the following is a guide to their utilization:

1. vwBlockedAccounts: Creates a view that shows all the accounts that are currently blocked. To view the output of the vwBlockedAccounts view insert the following SQL command at the bottom of the create views script file: `SELECT * FROM mmorpg.vwblockedaccounts;` . Or if you prefer to use the MySQL workbench GUI navigate to the schemas panel on the left hand side of the screen and click on the mmorpg database(note: you ma have to refresh the schemas panel for the mmorpg database object to be visible) then click on views dropdown menu and when hovering on the vwblockedaccounts view click on the right most popup icon appearing above the vwblockedaccounts view. This should execute the statement provided earlier automatically.

2. vwTopSkill: The top 20 characters with the highest skill levels and their account details. To view the output of the vwTopSkill view insert the following SQL command at the bottom of the create views script file: `SELECT * FROM mmorpg.vwTopSkill;` . Or if you prefer to use the MySQL workbench GUI navigate to the schemas panel on the left hand side of the screen and click on the mmorpg database(note: you may have to refresh the schemas panel for the mmorpg database object to be visible) then click on views dropdown menu and when hovering on the vwTopSkill view click on the right most popup icon appearing above the vwTopSkill view. This should execute the statement provided earlier automatically.
3. vwTopStackedItems: A view of the 20 items that are stacked most frequently, including the characters whose inventory they are stacked in. To view the output of the vwTopStackedItems view insert the following SQL command at the bottom of the create views script file: `SELECT * FROM mmorpg.vwTopStackedItems;` . Or if you prefer to use the MySQL workbench GUI navigate to the schemas panel on the left hand side of the screen and click on the mmorpg database(note: you may have to refresh the schemas panel for the mmorpg database object to be visible) then click on views dropdown menu and when hovering on the vwTopStackedItems view click on the right most popup icon appearing above the vwTopStackedItems view. This should execute the statement provided earlier automatically.
4. vwPopItems: The five most popular items that are contained inside the character's inventory, including the number of characters that have the item. To view the output of the vwPopItems view insert the following SQL command at the bottom of the create views script file: `SELECT * FROM mmorpg.vwPopItems;` . Or if you prefer to use the MySQL workbench GUI navigate to the schemas panel on the left hand side of the screen and click on the mmorpg database(note: you may have to refresh the schemas panel for the mmorpg database object to be visible) then click on views dropdown menu and when hovering on the vwPopItems view click on the right most popup icon appearing above the vwPopItems view. This should execute the statement provided earlier automatically.

Using Stored procedures.

The mmorpg database consists of five stored procedures; the following is a guide to their utilization:

1. spRegister: Creates a stored procedure that registers a new account. To execute the stored procedure insert the following statement: `CALL spRegister('lukus');`. The spRegister stored procedure takes one argument (account_N) which is the name of the new account to be inserted into the accounts table(Note: if an already existing account name is provided then an error message stating that the account already exists will be displayed). Once the statement is executed the new record will be shown in a results

tab. A select statement could be used and the new account will be seen as the last entry in the accounts table.

2. `spAddTime`: Adds the specified time to the specified account. To execute the stored procedure insert the following statement: `CALL spAddTime('lukus');` This will add one month worth of time to the `nextPaymentDate` and the `prePaymentDate` indicating that the account has received one month's worth of playtime. The procedure takes one argument which is the account name (`account_N`) and raises an error message stating 'Account not found' if an incorrect or non-existing account is provided. To view the result check the dates of the account you wish to add time to before and after executing the procedure. In this example the account 'lukus' has a `priceID` of 13
3. `spAddItem`: Creates a stored procedure that adds an item to a character's inventory. To execute the stored procedure insert the following statement: `CALL spAddItem('olivia', 7, 32);` The procedure takes three arguments (`character_N`) the character's name to which the item will be added to, (`item_id`) the ID of the item you wish to add and (`item_q`) which is how much of the particular item you wish to allocate. To view the result of this procedure run the following command before executing the procedure: `SELECT * FROM mmorpg.inventoryslot;` This will bring up a table of all the items that are currently in a character's inventory. For this example we are using 'olivia' which is a character with the character ID of 4. Before the procedure is executed it is seen in the inventory slot table that there are only 2 entries with the `characterID` value of 4. After executing the procedure a new entry will be added with the `itemID` and quantity provided in the procedure parameters.
4. `spAddChar`: Creates a stored procedure that adds a specified character to a specified account. To execute the stored procedure insert the following statement: `CALL spAddChar('verymade', 'flora', 'the_boys');` The procedure takes three arguments (`account-N`) the name of the account you wish to add a new character to, (`character_N`) the name of the new character and (`team_N`) the name of the team the new character will be assigned to (this could be a new or existing team). To view the result of the procedure run the following statement before executing the procedure. `SELECT * FROM mmorpg.`character`;` For this example we add a character named 'flora' to the account named 'verymade' (with account ID 1) and assign the character to a team called 'the_boys'. Before the procedure is executed it can be seen that the account 'verymade' has not made a character yet as there is no entry in the characters table of a character with the account ID of 1. After the procedure is executed it can be seen that a new character named 'flora' is added to the characters table with the account ID of 1. (Note the character skill level is assigned the value of zero because it is a new character)
5. `spSendLetter`: Creates a stored procedure that prints out a letter for each account, indicating how much time they have left. To execute the stored procedure insert the following statement: `CALL spSendLetter('Weekend xp boost sale on the 9th -10th of march!!');` The procedure requires a single argument (`News`) representing the news that

will be distributed to all accounts within the game. On execution a results tab will popup with all the messages sent to the account that have not been blocked because they missed the previous payment date.

And that's about it for the instructions on the mmorpg database. Hopefully, this was helpful, and best of luck!