SWE 4602

SOFTWARE DESIGN AND ARCHITECTURE

Database Design

1 Task

Suppose you are given the task of automating the operations of a large-scale multiplayer RPG game via a single platform. There are multiple regions (e.g., Noxus, Ionia, Demacia, Piltover), and each region has thousands of players. Each player can create multiple characters, and each character embarks on quests and collects magical items.
Each character can:

- Take part in multiple quests (e.g., “Rescue the Oracle”, “The Titan’s Vault”), and each quest may involve many characters.
- Collect multiple magical items like weapons or potions, and each item may be owned by many characters.
- Quests also reward items, and some items may be rewarded in multiple quests.

Players can see:

- Their characters and inventories
- The quests they’ve completed
- The items they have earned

To enable advanced in-game analytics and personalized content:

- Each player can mark multiple favorite item types (like "Potion", "Weapon", "Ring")
- Players can give up-vote/down-vote to each quest after completing it

You can assume necessary attributes if necessary.

2 User Story

As a player, I want to rate quests I’ve completed with a 5-star rating system instead of simple up/-down votes.

3 EDD Tasks:

1. The quest_participation table tracks quest completion with a boolean is-up-vote column (true=upvote, false=downvote, null=unrated) and a simple completed_at timestamp.
2. Modify quest_participation table:
   (a) Replace is-up-vote with a rating column (1-5 stars)
   (b) Keep rating_timestamp to track when ratings are given
3. Add Average Quest Rating:
   (a) Add average_rating column to quest table
   (b) Create stored procedure recalculate_quest_ratings() to update averages
4. New Stored Procedures:
   (a) add_quest_rating(character_id, quest_id, rating_value)
   (b) get_quest_average_rating(quest_id)

Initial Tasks
The tasks are not in any specific order. You need to figure out in which order you should do the tasks.

- Execute the seed script
- Write a migration script to complete the database end of the user story
- Execute the initialization script
- You have to maintain **change_log** for evolutionary practice. To do that, you should complete the following-
    1. Create a change_log schema with automatic id along with applied_at, created_by, script_name, and script_details.
    2. Script name should have a prefix for serial number—for example, ***000_init_schema.sql***. The change_log file should be the first script.
    3. For every previous task, you have a separate SQL file. An insertion to the

Notes: Write one migration script for each task.

4 Reporting Database Tasks:

Create the star schema that covers the below queries. When creating the star schema, use prefixes to distinguish between the fact table and dimension tables. For example, you may use fact_tableName as a table name. You have to identify the fact and dimension tables for your star schema.

4.1 Tasks

1. Add two more tables:
   `time (timeID, date, day, month, quarter, year, weekday)`
   `ratings(ratingID, playerID, questID, rating)`
2. Identify the fact table and dimension table.
3. Create a script to copy data from the operational DB to the reporting DB. As you introduced denormalization in the operational database, select only the relevant columns that would be needed to fulfill the queries.
4. Write the following queries using the star schema.
    (a) Region-Wise Most Popular Quests Based on Votes
    (b) Top 5 Most Rewarded Players by Region
    (c) Quest Ratings and Rewards Across Months
    (d) Player Activity Summary (Quests Completed, Total Time, Reward Points)
    (e) Monthly Region-Based Number of Player Engagement
    (f) Most Frequently Played Quests with Average Reward Points Higher than 100 & Average Duration More than 30 minutes