SWE 4602

SOFTWARE DESIGN AND ARCHITECTURE

Database Design

Task

In **SpeedVerse**, players participate in competitive racing events across multiple cities such as *NeoTokyo*, *Skyline Bay*, *Mecha Hills*, and *Solar Drift*. Each player owns multiple cars, tunes their vehicles with parts, and participates in races for rewards and rankings.
Each car:

- Can enter multiple **races** (e.g., "Turbo Rally", "Skyline Showdown"), and each race involves many cars.
- Is equipped with **multiple parts** (engines, tires, turbos), and each part can be used by multiple cars.
- **Races may reward parts**, and the same part may be a reward in multiple races.

Players can:

- View their cars and equipped parts
- See the races they've participated in
- Track the parts they've won

For game analytics and personalization:

- Players can select **preferred part types** (like "Turbo", "Suspension", "Tire")
- Players rate each race for challenge and enjoyment after completing it

Assume necessary attributes such as timestamps, ratings, and statuses as needed.

1 User Story

As a player, I want to rate races I've participated in using a 1–5 star system, so that I can give better feedback on race quality and difficulty.

2 EDD Tasks:

1. The race_participation table tracks race completion with a boolean is_up_vote column (true = upvote, false = downvote, null = unrated) and a simple completed_at timestamp.
2. Modify race_participation table:
   (a) Replace is_up_vote with a rating column (1–5 stars)
   (b) Add rating_timestamp to track when ratings are given
3. Add Average Race Rating:
   (a) Add average_rating column to the race table
   (b) Create stored procedure `recalculate_race_ratings()` to update average ratings for all races
4. New Stored Procedures:
   (a) `add_race_rating(car_id, race_id, rating_value)`
   (b) `get_race_average_rating(race_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.

Notes: Write one migration script for each task.

3 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.

3.1 Tasks

1. Add two more tables:
   `time (timeID, date, day, month, quarter, year, weekday)`
   `ratings (ratingID, playerID, raceID, rating)`
2. Identify the fact table and dimension tables.
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 procedures using the star schema.
    (a) City-Wise Most Popular Races Based on Ratings
    (b) Top 5 Most Rewarded Players by City
    (c) Race Ratings and Rewards Across Months
    (d) Player Activity Summary (Races Completed, Total Time, Reward Points)
    (e) Monthly City-Based Number of Player Engagement
    (f) Most Frequently Played Races with Average Reward Points Higher than 100 & Average Duration More than 30 minutes