CS306 - Post 2000 F1 Database Phase 3 - Report

Ardıl Yüce 32375 Kadir Yağız Ebil 32327 Zeynep Zilan Turunç 32066

This phase focuses on enhancing a Formula 1-themed database system we wrote through the use of MySQL triggers and stored procedures. Triggers were implemented to automate real-time updates to key statistics (such as wins, podiums and poles) after a race is inserted. Stored procedures were designed to provide summarized statistics and manage complex queries efficiently, offering reusable and maintainable solutions for interacting with the database.

```
INSERT INTO Race (race_id, circuit_id, car_id, driver_id, team_id, race_date, winning_team_id, winning_driver_id, pole_position_driver_id, fastest_lap_driver_id, grid_position, finishing_position)
VALUES (11, 1, 3, 4, 3, '2024-09-08', 3, 4, 4, 4, 1, 1);
```

Trigger Implementations

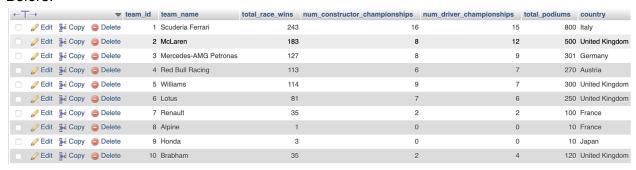
UpdateTeamStats: This trigger is executed after a new race is inserted. It performs two key updates.

- If the race has a winning_team_id, the respective team's total_race_wins is incremented
- If the finishing position of the race is less than or equal to 3, indicating a podium finish, the corresponding team's *total_podiums* is also incremented.

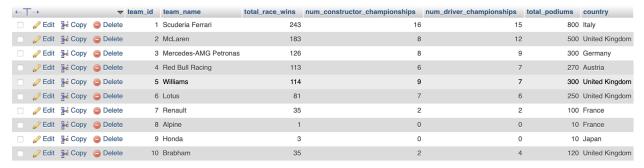
This trigger helps maintain accurate team performance statistics without manual intervention.

```
DELIMITER //
CREATE TRIGGER UpdateTeamStats
AFTER INSERT ON Race
FOR EACH ROW
 IF NEW.winning_team_id IS NOT NULL THEN
   UPDATE Team
    SET total_race_wins = total_race_wins + 1
   WHERE team_id = NEW.winning_team_id;
 END IF;
  -- Increment podium count
  IF NEW.finishing_position <= 3 THEN</pre>
   UPDATE Team
   SET total_podiums = total_podiums + 1
   WHERE team_id = NEW.team_id;
 END IF;
END;
DELIMITER :
```

Before:



After:



UpdateDriverStats: This trigger updates individual driver statistics on race insertion. Specifically:

- **num_wins** is incremented if the driver won the race.
- **num_poles** is incremented if the driver secured the pole position.
- **num_podiums** is incremented if the driver finished the race in podium position.

This ensures real-time maintenance of core driver performance metrics.

UpdateCarStats: The car statistics are updated via this trigger, which handles:

- Incrementing wins if the car's driver won the race.
- Incrementing the *poles* if the car was used in a pole-winning performance.
- Incrementing podiums if the car was used to finish first in the race.

This trigger ensures the cars' performance records stay consistent with race outcomes.

Stored Procedure Implementation:

GetDriverCareerSummary: This stored procedure retrieves the career summary of a specific driver using their *driver_id* as input. It returns key statistics including:

- Driver name and number
- Date of birth
- Total number of wins, podiums, pole positions, and championships.

It provides a clean and direct way to analyze a driver's historical performance in the system.

```
-- Procedure 1: Get Driver's Career Summary
DELIMITER //
CREATE PROCEDURE GetDriverCareerSummary (IN driverId INT)
BEGIN
    SELECT
        driver_name,
        driver_number,
        date_of_birth,
        num_wins,
        num_podiums,
        num_championships,
        num_poles
    FROM Driver
   WHERE driver_id = driverId;
END;
DELIMITER;
```

GetTeamWonRaces:This procedure lists all the races that were won by a specific team. It joins the *Race*, *Driver*, and *Circuit* tables to retrieve contextual data about each race, including:

- race_id and race_date
- circuit_name
- Winning driver's *driver_name*

It is especially useful for analyzing a team's competitive history across seasons.

GetTeamStandings: This procedure displays the standings of all teams based on their total race wins. It returns:

- team_name
- num_contructor_championships and num_driver_championship
- total_race_wins

The results are ordered from most successful to least, providing a leaderboard-style output ideal for reporting or dashboard integration.