Project 1: Let's Formulathon!

CMPE 321, Introduction to Database Systems, Spring 2021

Due: **April 30, Friday, 23:55**

Please ask the project related questions in the Moodle forum!

1 Introduction

High speed, adrenaline, teamwork, fear, victory, defeat, joy, frustration, despair... Those words describe the event which aims to break the records every time it happens. In this event, each year, we see 20 drivers, who are selected from various countries; 10 teams, which are contesting to rule the domain of money, power and fame; 20+ tracks to feel the rush of adrenaline in your veins; and one simple yet powerful motivation: ***To be Victorious****.*

We all know what it is, and what will come along with it. I hereby announce the start of the season of Formula 1 Grands Prix 2021. Let the fastest win!

Formula One (F1)[1] is a racing event that is governed by **Fédération Internationale de l'Automobile (FIA)**. Every year, 10 teams participate in F1 with two identical cars, each of which is driven by a talented driver. The drivers race in many circuits all around the world to simply become the fastest driver alive. At each race (or Grand Prix as a fancy word), the first ten drivers collect points that accumulate towards the season ranking and at the end of the season only one becomes the world champion! Enough introduction, let's talk some code!

Özgür is a junior computer science undergraduate student and a huge fan of F1. One day, he bumps into an ad that says "Let's Formulathon!" on the Internet. He immediately reads the details and learns that this is a

[1] You can check out this link for further information on F1.

hackathon organized by FIA, with a prize of 6 Formula 1 Grand Prix tickets. A hackathon by FIA? A prize of F1 tickets? He starts to hear his heartbeats!

He reads that Formulathon is an online data retrieval and querying contest in which at most 200 teams of 2 or 3 people can participate. He immediately clicks "participate" and sees the error: "Team limit is reached. Thanks for your interest!". What a buzzkill...

Of course, this does not stop Özgür. He could not miss a lifetime opportunity. He starts scanning every team one by one and finds several teams of 2 people. He contacts each one of them to be a member and one of them (your team) accepts! Hurraaah!

Not much time to celebrate though, the contest is already on! Özgür is experienced in data retrieval and you assign him the task of aggregating official data on drivers, teams, Grand Prixs, tracks, and power unit manufacturers. He collects the data in no time and the jury approves the integrity of the dataset. This means only one thing: start querying!

Özgür has high hopes in you. He is confident that you can nail the second part and win this once and for all!

MAY THE BEST QUERIES WIN!

2 Data

Özgür has provided the dataset as an SQLite database file (f1.db) and also documented each table as below. PK stands for primary key and FK stands for foreign key.

- **Table 1: Drivers**
  - (PK) driver_id: *INTEGER*
  - first_name: *TEXT*
  - last_name: *TEXT*
  - year_of_birth: *INTEGER*
  - country_of_birth: *TEXT*

Figure 1: Starting line-up of in an F1 race. All the green lights are on, meaning START! (Source: Drivetribe)

  - country_of_driving_license: *TEXT*
  - (FK on Teams:team_id) team: *INTEGER*
  - driver_wc_count: *INTEGER* (wc stands for world championship)
  - total_points: *INTEGER*
  - total_podiums: *INTEGER*
  - driver_hrf: *INTEGER* (hrf stands for highest race finish, i.e., the best rank a driver obtained throughout his/her career.)
  - driver_hrf_count: *INTEGER*

- **Table 2: Teams**
  - (PK) team_id: *INTEGER*.
  - short_name: *TEXT*
  - long_name: *TEXT*
  - base_location: *TEXT*
  - team_chief: *TEXT*

  - technical_chief: *TEXT*
  - chassis: *TEXT*
  - (FK on PUMs:pum_id)pum: *INTEGER* (pum stands for power_unit_manufacturer)
  - team_wc_count: *INTEGER*
  - team_hrf: *INTEGER*
  - team_hrf_count: *INTEGER*
  - (FK on Drivers:driver_id) driver_one: *INTEGER*
  - (FK on Drivers:driver_id) driver_two: *INTEGER*

- **Table 3: GrandPrixs**
  - (PK) grand_prix_id: *INTEGER*
  - grand_prix_name: *TEXT*
  - grand_prix_country: *TEXT*
  - (FK on Tracks:track_id) circuit: *INTEGER*
  - starting_date: *TEXT* (formatted as ISO8601)
  - ending_date: *TEXT* (formatted as ISO8601)

- **Table 4: Tracks**
  - (PK) track_id: *INTEGER*
  - circuit_name: *TEXT*
  - circuit_country: *TEXT*
  - circuit_type: *TEXT*
  - clim: *INTEGER* (clim stands for circuit length in meters)
  - total_laps: *INTEGER*
  - lrims: *INTEGER* (lrims stands for lap record in milliseconds)
  - (FK on Drivers:driver_id) lap_recorder: *INTEGER* (can be NULL)

- **Table 5: PUMs**
  - (PK) pum_id: *INTEGER*
  - pum_name: *TEXT*

- (FK on Teams:team_id) buyer_one: *INTEGER* (can be NULL)
- (FK on Teams:team_id) buyer_two:*INTEGER* (can be NULL)
- (FK on Teams:team_id) buyer_three: *INTEGER* (can be NULL)
- (FK on Teams:team_id) buyer_four: *INTEGER* (can be NULL)

3 Queries

The organizers described 15 scenarios and ask you to write SQLite queries for each of them. The queries are as follows:

1. Return the total number of Grand Prixs that starts and ends in April.

2. Print the sum of the total points of the drivers by birth year. Include only the drivers born between 1990 and 1999, inclusive. Display birth year and sum in descending order with respect to sum.

3. Write the following relational algebra query in SQL. List the results in descending order of team world championship count:
   $\Pi_{\text{full\_name, team\_chief, technical\_chief, team\_wc\_count}}($
   $\quad \text{Teams} \bowtie_{short\_name} \rho(P(1 \to short\_name), \Pi_{pum\_name}\text{PUMs})$
)

4. Return the sum of the total points of the drivers who received their driving licenses from the country they were born in and whose teams are based in England and do not buy power units from Mercedes.

5. List the first name, last name, base locations of their teams, total podiums, and highest race finishes (hrf) of the drivers whose teams' base location end with "LAND" and sort them with respect to the base location in descending order, then total podiums in ascending order, and then drivers highest race finish (hrf) in ascending order.

6. List the short names, full names, world championship counts, and team total podiums of the teams that have at least 10 team total podiums and that contain at least one driver who was NOT born in the country where he/she received his/her license. Sort the results in descending order of short names.

7. Display the total length of the circuits per circuit type in descending order with respect to total length. Consider only the circuits in the countries with an "A" in their names except for Italy. (**Hint:** You can compute the length of a track in meters by multiplying its clim and number of laps.)

8. List the team id, team short name, team full name, and team driven total in KMs in ascending order with respect to driven total. Here, compute the driven total of a team by considering only the drivers of the team who hold a lap record in any track and who are not born in England. Display only the teams that have driven at least 500 KMs in total.

9. Return the absolute difference of total points received by the old and young drivers? Here we define old drivers as the ones born between 1980 and 1989 and young drivers as the ones born between 1990 and 1999, both ranges being inclusive.

10. For the Red Bull Racing drivers who hold a lap record, list driver id, first name, last name, name of the circuit which they hold a lap record, and the lap record in milliseconds (lrims). Sort the results in ascending order of lrims.

11. List the short names and base locations of the teams that contain at least one driver who holds a lap record only in France or Monaco track. List the lap recorder driver names, their countries of birth and the countries they hold a lap record in, as well. Sort the results in ascending order with respect to team's short name.

12. List the name of the PUMs whose all customers had become world champion in the past.

13. Find the most successful teams by circuit type (street, race) where the most successful team is the one with the highest number of lap records. If there are multiple winners for a circuit type, return all of them. Display circuit type, short name of the team, and the number of lap records they hold in ascending order of circuit type and team name.

14. Find the teams whose highest race finish (hrf) count is less than the sum of their drivers'. List the short names, hrf counts and sums of their drivers' hrf counts of such teams in descending order of short name.

15. Find the teams whose driver one's podium count is at least five times of driver two's podium count. Display the short names, driver one podium counts and driver two podium counts of such teams in ascending order of short name.

Submission Details

Write each query to a file named `q<query_index>.sql` (e.g. `q1.sql`, `q10.sql`) and include a comment in each file that explains the reasoning that led you to the query in detail. **The `.sql` files that do not contain explanatory comments will receive 0.**

In turn, place all `.sql` files into a folder named with the student IDs of the team members separated by an underscore (e.g. `2017400200_2018400122`). Zip the folder for submission and name the `.zip` file with the same name. Submit the `.zip` file through Moodle until the deadline. **Any other submission method and late submissions are not allowed.**

Compliance with the file naming is essential as the submissions will be graded automatically. This will constitute 10 points of your grade and 90 points will be distributed equally to each query.

Additional Remarks

- This project can be implemented either individually or as a team of two people. You are free to change teams in the upcoming projects.
- Do not inject your observations into your queries to avoid steps needed to return the result. For instance, while writing a query on "Red Bull Racing" do not get the team id manually from the database and inject it in your query. Instead, get the team ID from name through SQL if you need it. **Violating this condition will be treated as a dishonest behaviour and the cheating procedures in the syllabus will be applied.**