Problem 3308: Find Top Performing Driver

Problem Information

Difficulty: Medium
Acceptance Rate: 48.89%
Paid Only: Yes
Tags: Database

Problem Description

Table: `Drivers`

+--------------+---------+ | Column Name | Type | +--------------+---------+ | driver_id | int | | name | varchar | | age | int | | experience | int | | accidents | int | +--------------+---------+ (driver_id) is the unique key for this table. Each row includes a driver's ID, their name, age, years of driving experience, and the number of accidents they've had.

Table: `Vehicles`

+--------------+---------+ | vehicle_id | int | | driver_id | int | | model | varchar | | fuel_type | varchar | | mileage | int | +--------------+---------+ (vehicle_id, driver_id, fuel_type) is the unique key for this table. Each row includes the vehicle's ID, the driver who operates it, the model, fuel type, and mileage.

Table: `Trips`

+--------------+---------+ | trip_id | int | | vehicle_id | int | | distance | int | | duration | int | | rating | int | +--------------+---------+ (trip_id) is the unique key for this table. Each row includes a trip's ID, the vehicle used, the distance covered (in miles), the trip duration (in minutes), and the passenger's rating (1-5).

Uber is analyzing drivers based on their trips. Write a solution to find the **top-performing driver** for **each fuel type** based on the following criteria:

1. A driver's performance is calculated as the **average rating** across all their trips. Average rating should be rounded to `2` decimal places. 2. If two drivers have the same average

rating, the driver with the **longer total distance** traveled should be ranked higher. 3. If there is **still a tie** , choose the driver with the **fewest accidents**.

Return _the result table ordered by_ `fuel_type` _in_ **ascending** _order._

The result format is in the following example.

Example:

Input:

`Drivers` table:

+----------+----------+-----+------------+----------+ | driver_id | name | age | experience | accidents | +----------+----------+-----+------------+----------+ | 1 | Alice | 34 | 10 | 1 | | 2 | Bob | 45 | 20 | 3 | | 3 | Charlie | 28 | 5 | 0 | +----------+----------+-----+------------+----------+

`Vehicles` table:

+------------+-----------+---------+-----------+---------+ | vehicle_id | driver_id | model | fuel_type | mileage | +------------+-----------+---------+-----------+---------+ | 100 | 1 | Sedan | Gasoline | 20000 | | 101 | 2 | SUV | Electric | 30000 | | 102 | 3 | Coupe | Gasoline | 15000 | +------------+-----------+---------+-----------+---------+

`Trips` table:

+---------+------------+----------+----------+--------+ | trip_id | vehicle_id | distance | duration | rating | +---------+------------+----------+----------+--------+ | 201 | 100 | 50 | 30 | 5 | | 202 | 100 | 30 | 20 | 4 | | 203 | 101 | 100 | 60 | 4 | | 204 | 101 | 80 | 50 | 5 | | 205 | 102 | 40 | 30 | 5 | | 206 | 102 | 60 | 40 | 5 | +---------+------------+----------+----------+--------+

Output:

+-----------+-----------+--------+----------+ | fuel_type | driver_id | rating | distance | +-----------+-----------+--------+----------+ | Electric | 2 | 4.50 | 180 | | Gasoline | 3 | 5.00 | 100 | +-----------+-----------+--------+----------+

Explanation:

* For fuel type `Gasoline`, both Alice (Driver 1) and Charlie (Driver 3) have trips. Charlie has an average rating of 5.0, while Alice has 4.5. Therefore, Charlie is selected. * For fuel type `Electric`, Bob (Driver 2) is the only driver with an average rating of 4.5, so he is selected.

The output table is ordered by `fuel_type` in ascending order.

Code Snippets

MySQL:

```
# Write your MySQL query statement below
```

MS SQL Server:

```
/* Write your T-SQL query statement below */
```

PostgreSQL:

```
-- Write your PostgreSQL query statement below
```