Mystics Data Analyst Position – Technical Exercise Part 1: SQL Assessment

Vitor Freitas

-- 1. Write a query that returns the number of players whose first name is Robert

SELECT COUNT(\*) AS robert\_count

FROM players

WHERE first\_name = 'Robert';

This query is pretty self-explanatory. It selects players with the first name Robert using the WHERE clause and counts the number of columns returned.

-- 2. Write a query that selects “player\_id “and average “min” for each player who played in more than one all star game

SELECT player\_id, AVG(min) AS avg\_minutes

FROM season\_totals\_allstar

GROUP BY player\_id

HAVING COUNT(\*) > 1;

This query calculates the average minutes played with the AVG(min) function. The HAVING COUNT(\*) > 1 clause ensures that only players who have played in more than one all-star game are included in the results.

-- 3. Write a query that returns the full name of all players who did not make an all-star game ordered by their career minutes played

SELECT p.full\_name

FROM players p

LEFT JOIN career\_totals\_allstar a ON p.id = a.player\_id

INNER JOIN career\_totals\_regular\_season r ON p.id = r.player\_id

WHERE a.player\_id IS NULL

ORDER BY r.min DESC;

In this query, the LEFT JOIN with the career\_total\_allstar table ensures that all players are included, even those who never made an all-star game. Then we check for NULL values in the player\_id column of the career\_total\_allstar table using the WHERE clause and order the players by minutes played.

-- 4. Select data that will be returned by the SQL query. Correct answers are:

f. Mary,2

h. Brenda,1

This query retrieves the names of female users and the count of their friend relationships. The LEFT JOIN ensures that all rows from the users table are included, even if they have no matching rows in the friends table, so that is why Brenda has a count of 1 even thought there are no matching columns on friends. The ON users.id = friends.user1 OR users.id = friends.user2 condition checks if the user is in either column of the friends table. Here is the SQL code I used to recreate the tables and verify my answer:

-- Create the 'users' table

CREATE TABLE users (

id INT PRIMARY KEY,

name VARCHAR(50),

sex CHAR(1)

);

INSERT INTO users (id, name, sex) VALUES

(1, 'Ann', NULL),

(2, 'Steve', 'm'),

(3, 'Mary', 'f'),

(4, 'Brenda', 'f');

-- Create the 'friends' table

CREATE TABLE friends (

user1 INT,

user2 INT,

FOREIGN KEY (user1) REFERENCES users(id),

FOREIGN KEY (user2) REFERENCES users(id)

);

INSERT INTO friends (user1, user2) VALUES

(1, 2),

(1, 3),

(2, 3);