Team Name: SQL Warriors

4/4/2024
GitHub
Sean Dixon:
Manny Walia:
Daniella Raj:
Jacob Gates:

Prompt: Pretend you are the owner/operator of a tennis (or football, soccer - your choice) club needing to build a relational database. You hired some students from the MIST 4610 class at the University of Georgia to create the database for you. They need to know more about your organization to identify which entities, attributes, and relationships are important for you. Start by describing your business as a real client.

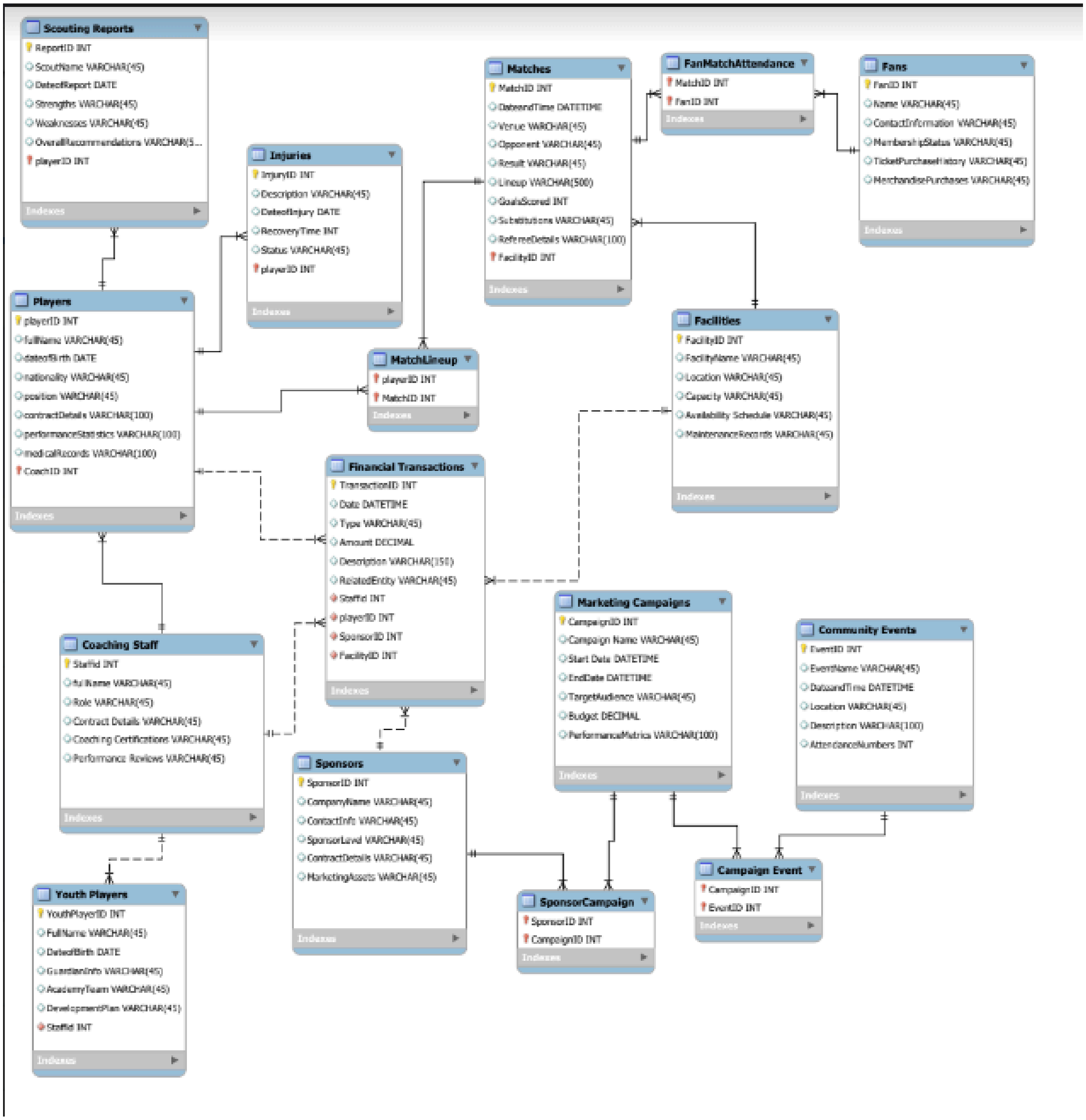
Problem Description

I am the owner/operator of a tennis club located in a bustling suburban area. Our club is known for its state-of-the-art facilities and a vibrant community of tennis enthusiasts ranging from beginners to seasoned professionals. Our primary goal is to provide a top-notch tennis experience for our members, offering various services such as coaching, court rentals, tournaments, and social events.

We have a diverse membership base consisting of individuals of all ages, skill levels, and backgrounds. Our club hosts regular tennis leagues and tournaments, both for recreational play and competitive matches. Additionally, we offer coaching programs led by certified instructors to help our members improve their skills and reach their full potential on the court.

In terms of operations, we manage court reservations, membership registrations, coaching schedules, tournament logistics, and facility maintenance. We rely on efficient organization and communication to ensure smooth functioning and deliver a positive experience for our members.

Overall, our tennis club prides itself on providing a comprehensive tennis experience that caters to the needs and preferences of our diverse membership base, fostering a welcoming and inclusive environment for all tennis enthusiasts.



How these all connect:
Players and Matches:
Each match involves a lineup of players.
Connect via a many-to-many relationship table called "MatchLineup":
MatchID (Foreign Key referencing Matches)
PlayerID (Foreign Key referencing Players)

Players and Coaching Staff:
Players are coached by members of the coaching staff.
Connect via a foreign key in the Players table:
CoachID (Foreign Key referencing Coaching Staff)

Matches and Facilities:
Each match is played at a specific facility.
Connect via a foreign key in the Matches table:
FacilityID (Foreign Key referencing Facilities)

Financial Transactions and Related Entities:
Financial transactions involve various entities like players, staff, facilities, and sponsors.
Connect via foreign keys depending on the related entity:
PlayerID (Foreign Key referencing Players)
StaffID (Foreign Key referencing Coaching Staff)
FacilityID (Foreign Key referencing Facilities)
SponsorID (Foreign Key referencing Sponsors)

Sponsors and Marketing Campaigns:
Sponsors may be associated with specific marketing campaigns.
Connect via a many-to-many relationship table:
SponsorCampaign:
SponsorID (Foreign Key referencing Sponsors)
CampaignID (Foreign Key referencing Marketing Campaigns)

Fans and Matches:
Fans attend matches.
Connect via a many-to-many relationship table:
FanMatchAttendance:
FanID (Foreign Key referencing Fans)
MatchID (Foreign Key referencing Matches)

Youth Players and Coaching Staff:
Youth players are coached by members of the coaching staff.
Connect via a foreign key in the Youth Players table:
CoachID (Foreign Key referencing Coaching Staff)

Marketing Campaigns and Community Events:
Marketing campaigns may promote or be associated with community events.
Connect via a many-to-many relationship table:
CampaignEvent:
CampaignID (Foreign Key referencing Marketing Campaigns)
EventID (Foreign Key referencing Community Events)

Injuries and Players:
Each injury is associated with a specific player.
Connect via a foreign key in the Injuries table:
PlayerID (Foreign Key referencing Players)

Scouting Reports and Players:
Each scouting report pertains to a particular player.
Connect via a foreign key in the Scouting Reports table:
PlayerID (Foreign Key referencing Players)

Data Model Requirements:
Given the diverse operations and stakeholders involved, the database should effectively manage information related to players, coaching staff, matches, finances, facilities, sponsors, and fan engagement activities. Key functionalities include:

Player Management: Recording player profiles, contracts, performance statistics, medical records, and scouting reports.

Match Management: Tracking match schedules, results, lineup selections, performance analysis, and referee assignments.

Financial Management: Managing budgets, revenue streams, sponsorships, payroll, ticket sales, and expenditure across different departments.

Facility Management: Maintaining records of training facilities, stadium bookings, maintenance schedules, and equipment inventory.

Fan Engagement: Capturing data on fan demographics, ticket purchases, merchandise sales, loyalty programs, and feedback mechanisms.

Youth Development: Monitoring the progress of academy players, tracking their training sessions, educational requirements, and scouting evaluations.

Marketing and Sponsorship: Storing information on sponsorship agreements, marketing campaigns, advertising assets, and fan outreach initiatives.

Reporting and Analytics: Generating reports, dashboards, and analytical insights to aid decision-making across various departments.

Entities
Players:
PlayerID (Primary Key)
Full Name
Date of Birth
Nationality
Position

Contract Details (Start Date, End Date, Salary)

Performance Statistics (Goals scored, Assists, Yellow/Red Cards)
Medical Records

Coaching Staff:
StaffID (Primary Key)
Full Name
Role (Head Coach, Assistant Coach, Fitness Trainer, etc.)
Contract Details
Coaching Certifications
Performance Reviews

Matches:
MatchID (Primary Key)
Date and Time
Venue
Opponent
Result
Lineup
Goals Scored
Substitutions
Referee Details

Financial Transactions:
TransactionID (Primary Key)
Date
Type (Income, Expense)
Amount
Description
Related Entity (Player, Staff, Facility, Sponsor, etc.)

Facilities:
FacilityID (Primary Key)
Facility Name
Location
Capacity
Availability Schedule
Maintenance Records

Sponsors:
SponsorID (Primary Key)
Company Name
Contact Information

Sponsorship Level
Contract Details
Marketing Assets Provided

Fans:
FanID (Primary Key)
Name
Contact Information
Membership Status
Ticket Purchase History
Merchandise Purchases

Youth Players:
YouthPlayerID (Primary Key)
Full Name
Date of Birth
Guardian Information
Academy Team
Development Plan

Marketing Campaigns:
CampaignID (Primary Key)
Campaign Name
Start Date
End Date
Target Audience
Budget
Performance Metrics

Community Events:
EventID (Primary Key)
Event Name
Date and Time
Location
Description
Attendance Numbers

Injuries:
InjuryID (Primary Key)
PlayerID (Foreign Key referencing Players)
Description
Date of Injury
Expected Recovery Time
Status (Active, Recovered, Out for Season, etc.)

Scouting Reports:
ReportID (Primary Key)
PlayerID (Foreign Key referencing Players)
Scout Name
Date of Report
Strengths
Weaknesses
Overall Recommendation

```
-- Players table
CREATE TABLE Players (
    PlayerID INT PRIMARY KEY,
    FullName VARCHAR(100),
    DateOfBirth DATE,
    Nationality VARCHAR(50),
    Position VARCHAR(50),
    ContractStart DATE,
    ContractEnd DATE,
    Salary DECIMAL(10, 2)
);

-- Performance Statistics table
CREATE TABLE PerformanceStatistics (
    PlayerID INT,
    GoalsScored INT,
    Assists INT,
    YellowCards INT,
    RedCards INT,
    FOREIGN KEY (PlayerID) REFERENCES Players(PlayerID)
);

-- Medical Records table
CREATE TABLE MedicalRecords (
    PlayerID INT,
    RecordID INT PRIMARY KEY,
    Description TEXT,
    DateOfRecord DATE,
    FOREIGN KEY (PlayerID) REFERENCES Players(PlayerID)
);

-- Coaching Staff table
```

```
CREATE TABLE CoachingStaff (
   StaffID INT PRIMARY KEY,
   FullName VARCHAR(100),
   Role VARCHAR(50),
   ContractStart DATE,
   ContractEnd DATE,
   Certifications VARCHAR(100)
);

-- Performance Reviews table
CREATE TABLE PerformanceReviews (
   StaffID INT,
   ReviewID INT PRIMARY KEY,
   DateOfReview DATE,
   Rating INT,
   Comments TEXT,
   FOREIGN KEY (StaffID) REFERENCES CoachingStaff(StaffID)
);

-- Matches table
CREATE TABLE Matches (
   MatchID INT PRIMARY KEY,
   DateAndTime DATETIME,
   Venue VARCHAR(100),
   Opponent VARCHAR(100),
   Result VARCHAR(10),
   Referee VARCHAR(100)
);

-- Financial Transactions table
CREATE TABLE FinancialTransactions (
   TransactionID INT PRIMARY KEY,
   Date DATE,
   Type VARCHAR(10),
   Amount DECIMAL(10, 2),
   Description TEXT,
   RelatedEntity VARCHAR(100)
);

-- Facilities table
CREATE TABLE Facilities (
   FacilityID INT PRIMARY KEY,
   FacilityName VARCHAR(100),
   Location VARCHAR(100),
   Capacity INT,
   AvailabilitySchedule TEXT,
   MaintenanceRecords TEXT
);

-- Sponsors table
CREATE TABLE Sponsors (
   SponsorID INT PRIMARY KEY,
   CompanyName VARCHAR(100),
   ContactInformation VARCHAR(100),
   SponsorshipLevel VARCHAR(50),
   ContractDetails TEXT,
   MarketingAssetsProvided TEXT
);

-- Fans table
CREATE TABLE Fans (
   FanID INT PRIMARY KEY,
   Name VARCHAR(100),
   ContactInformation VARCHAR(100),
   MembershipStatus VARCHAR(20),
   TicketPurchaseHistory TEXT,
   MerchandisePurchases TEXT
);

-- Youth Players table
CREATE TABLE YouthPlayers (
   YouthPlayerID INT PRIMARY KEY,
   FullName VARCHAR(100),
   DateOfBirth DATE,
   GuardianInformation TEXT,
   AcademyTeam VARCHAR(100),
   DevelopmentPlan TEXT
);

-- Marketing Campaigns table
CREATE TABLE MarketingCampaigns (
   CampaignID INT PRIMARY KEY,
   CampaignName VARCHAR(100),
   StartDate DATE,
   EndDate DATE,
   TargetAudience VARCHAR(100),
   Budget DECIMAL(10, 2),
   PerformanceMetrics TEXT
```

```
);

-- Community Events table
CREATE TABLE CommunityEvents (
   EventID INT PRIMARY KEY,
   EventName VARCHAR(100),
   DateAndTime DATETIME,
   Location VARCHAR(100),
   Description TEXT,
   AttendanceNumbers INT
);

-- Injuries table
CREATE TABLE Injuries (
   InjuryID INT PRIMARY KEY,
   PlayerID INT,
   Description TEXT,
   DateOfInjury DATE,
   ExpectedRecoveryTime INT,
   Status VARCHAR(20),
   FOREIGN KEY (PlayerID) REFERENCES Players(PlayerID)
);



-- Scouting Reports table
CREATE TABLE ScoutingReports (
   ReportID INT PRIMARY KEY,
   PlayerID INT,
   ScoutName VARCHAR(100),
   DateOfReport DATE,
   Strengths TEXT,
   Weaknesses TEXT,
   OverallRecommendation TEXT,
   FOREIGN KEY (PlayerID) REFERENCES Players(PlayerID)
);
```

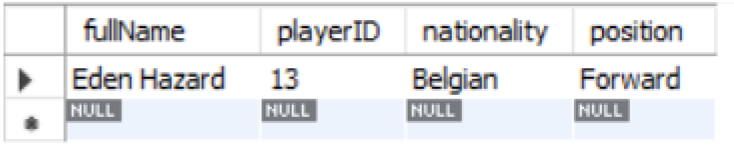
DATA DICTIONARY
https://docs.google.com/spreadsheets/d/1M424csQrZS3ED60HFmVYdFGILIPOhZZsH4MI0gDdIO4/edit?usp=sharing

Formulate 10 queries (4 simple, 6 complex)

Simple

1) Write a query to list the full names and playerID of all forwards who are from Belgium

```
SELECT fullName, playerID, nationality, position
FROM Players
Where nationality REGEXP 'Belgian' AND position REGEXP 'Forward';
```

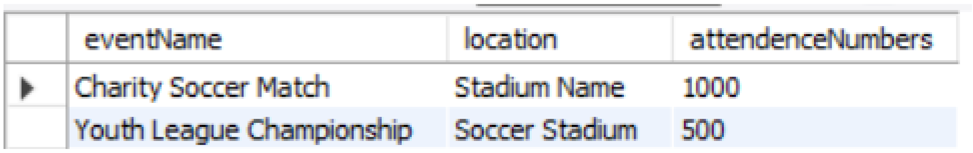


fullName	playerID	nationality	position
Eden Hazard	13	Belgian	Forward
NULL	NULL	NULL	NULL

This query would be useful to identify players who play a certain position from a certain country, in this case, Belgian players who play Forward.

2) Write a query to list all community events that are expected to have at least 400 attendees

```
SELECT eventName, location, attendenceNumbers
FROM CommunityEvents
WHERE attendenceNumbers >= 400;
```

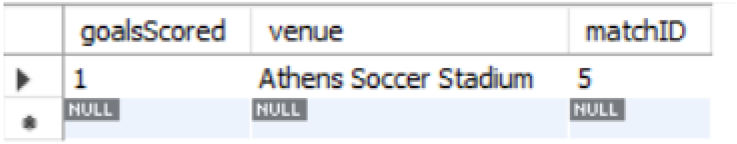


eventName	location	attendenceNumbers
Charity Soccer Match	Stadium Name	1000
Youth League Championship	Soccer Stadium	500

This query would be helpful for planning out food vendors, entertainment, and other additions the venues during games. Larger events may require more options to account for all of the fans.

3) Write a query to list every match that had a total of one goal scored by both teams and also took place in Athens Soccer stadium

```
SELECT goalsScored, venue, matchID
FROM Matches
WHERE goalsScored=1;
```

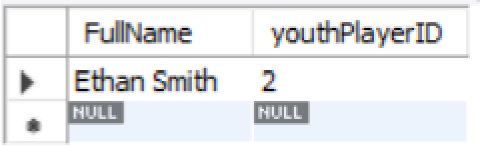


goalsScored	venue	matchID
1	Athens Soccer Stadium	5
NULL	NULL	NULL

Lots of matches take place in each stadium and some are more entertaining than others, and goals being scored directly impacts how entertaining the games are. This query identifies the games that were considered less exciting.

4) Write a query to list the names of all Youth players from the U 14 academy team and have birthdays in first half of the year

```
SELECT FullName, youthPlayerID
FROM YouthPlayer
WHERE AcademyTeam = 'Under-14s' AND MONTH(DateOfBirth) <= 6;
```



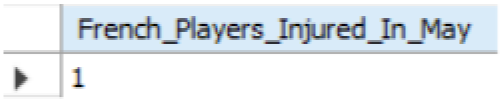
FullName	youthPlayerID
Ethan Smith	2
NULL	NULL

Youth teams are more likely to have get-togethers and parties for birthdays and the season ends around the middle of the year. This query lists the players who will have a birthday during the season so that plans to celebrate can be made.

Complex

1) Write a query to list the amount of French players who were injured in the months of May, June, or July

```
SELECT COUNT(DISTINCT p.PlayerID) AS French_Players_Injured_In_May
FROM Players p
JOIN Injuries i ON p.PlayerID = i.PlayerID
WHERE p.Nationality = 'French' AND MONTH(i.DateOfInjury) REGEXP '5|6|7';
```

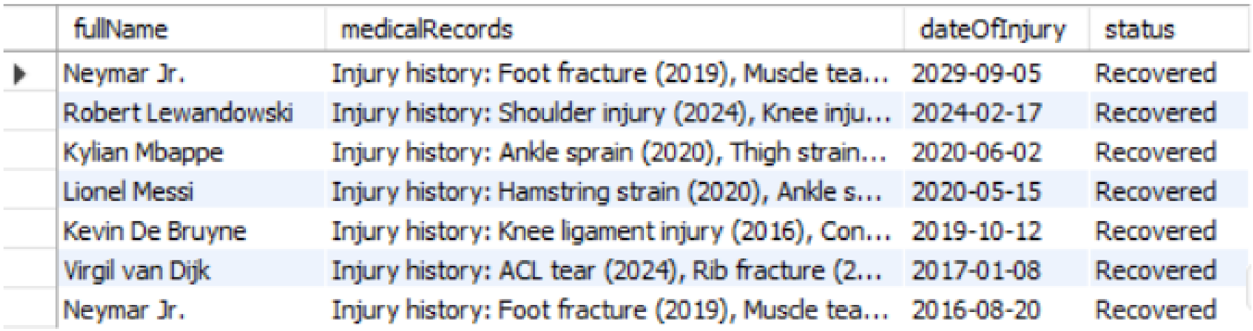


French_Players_Injured_In_May
1

If the French player was struggling during the months of May, June, and July, this query allows us to look deeper into why that may be. If they were missing key components on the pitch, it could be the reason why the team was underperforming.

2) Write a query to list the names of players who have recovered from injuries, with their dates in descending order.

```
SELECT fullName, medicalRecords, dateOfInjury, status
FROM Players p
JOIN Injuries i ON p.playerID=i.playerID
WHERE status REGEXP 'Recovered'
ORDER BY dateOfInjury DESC;
```

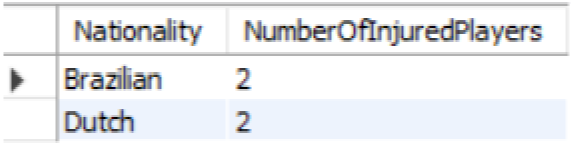


fullName	medicalRecords	dateOfInjury	status
Neymar Jr.	Injury history: Foot fracture (2019), Muscle tea...	2029-09-05	Recovered
Robert Lewandowski	Injury history: Shoulder injury (2024), Knee inju...	2024-02-17	Recovered
Kylian Mbappe	Injury history: Ankle sprain (2020), Thigh strain...	2020-06-02	Recovered
Lionel Messi	Injury history: Hamstring strain (2020), Ankle s...	2020-05-15	Recovered
Kevin De Bruyne	Injury history: Knee ligament injury (2016), Con...	2019-10-12	Recovered
Virgil van Dijk	Injury history: ACL tear (2024), Rib fracture (2...	2017-01-08	Recovered
Neymar Jr.	Injury history: Foot fracture (2019), Muscle tea...	2016-08-20	Recovered

Managers of the team may need to schedule follow up appointments with doctors or physical therapists for players after they have been injured. This query would identify the recency of injuries so that necessary appointments can be scheduled.

3) Write an SQL query to retrieve the nationality of players along with the count of injured players for each nationality, but only include nationalities where the count of injured players is greater than one

```
SELECT p.Nationality, COUNT(*) AS NumberOfInjuredPlayers
FROM Players p
JOIN Injuries i ON p.PlayerID = i.PlayerID
GROUP BY p.Nationality
HAVING COUNT(*)>1;
```



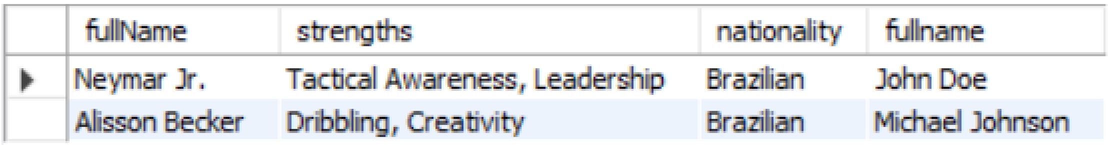
Nationality	NumberOfInjuredPlayers
Brazilian	2
Dutch	2

Certain teams may require more attention if they have more players on the IR (injury report). This query identifies the players that are most in need of aid.

4) Write a query retrieve the name of all brazilian players scouted, their strengths, and the full name of the coach involved in scouting

```
SELECT p.fullName, strengths, nationality, c.fullname
```

```
FROM ScoutingReports sr
JOIN Players p ON sr.playerID=p.playerID
JOIN CoachingStaff c ON p.coachID=c.staffID
WHERE nationality = 'Brazilian';
```

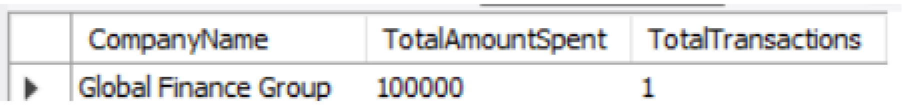


	fullName	strengths	nationality	fullname
▶	Neymar Jr.	Tactical Awareness, Leadership	Brazilian	John Doe
	Alisson Becker	Dribbling, Creativity	Brazilian	Michael Johnson

This query singles out a nationality, Brazilian in this case, and views their strengths. It allows coaches and scouts to recruit new talent and also shows which coach took notes on the player because certain coaches may value different attributes on the pitch.

5) Write a query should select the company name of each sponsor and the total number of transactions they've conducted, only if their transaction was greater than or equal to 100,000

```
SELECT s.CompanyName, SUM(ft.Amount) AS TotalAmountSpent, COUNT(ft.TransactionID)
AS TotalTransactions
FROM Sponsors s
JOIN FinancialTransactions ft ON s.CompanyName = ft.RelatedEntity
GROUP BY s.SponsorID, s.CompanyName
HAVING TotalAmountSpent >= 100000;
```

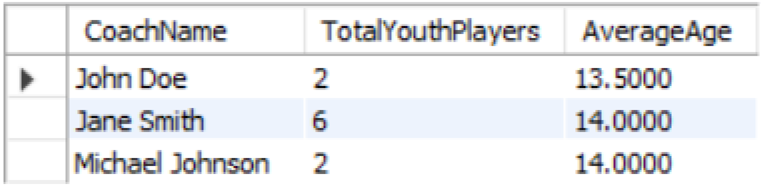


	CompanyName	TotalAmountSpent	TotalTransactions
▶	Global Finance Group	100000	1

This query is useful for finding the most valuable sponsors. It selects the name, how often they have sponsored events, and the amount too, lots of relevant information for choosing future sponsors.

6) Write a code to list the names of coaches, the amount of players they oversee, and the average age of those players.

```
SELECT c.FullName AS CoachName, COUNT(yp.YouthPlayerID) AS TotalYouthPlayers,
AVG(YEAR(CURRENT_DATE) - YEAR(yp.DateOfBirth)) AS AverageAge
FROM CoachingStaff c
JOIN YouthPlayer yp ON c.staffID = yp.staffID
GROUP BY c.StaffID, c.FullName;
```



CoachName	TotalYouthPlayers	AverageAge
John Doe	2	13.5000
Jane Smith	6	14.0000
Michael Johnson	2	14.0000

This query is good for analyzing coaches and potentially adding/removing players from youth rosters. If there are new players, this data would be used to find where to place them based on what teams need players. Also, fitting a new player in with other players of the same age.