MIST 4610 Group Project

Due: March 27

Turning in work as a GitHub Repository

GitHub Link: <https://github.com/hannahganeriwal/MIST4610-Project-1--Group-6.git>

What we need to do:

1. Construct a data model,
2. Build a database corresponding to that data model,
3. Populate the database with data and
4. Formulate 10 SQL queries (6 complex and 4 simple). These queries must be relevant from a managerial perspective (i.e why would a manager care about these queries).
5. Every team member needs to construct/maintain a github repository with details of the team project described below.

* Problem Description
* Data Model Explanation
* Query Descriptions
* Query Justifications

Prompt Used:

Pretend you are the owner/operator of a college football team 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

**Scenario from ChatGPT:**

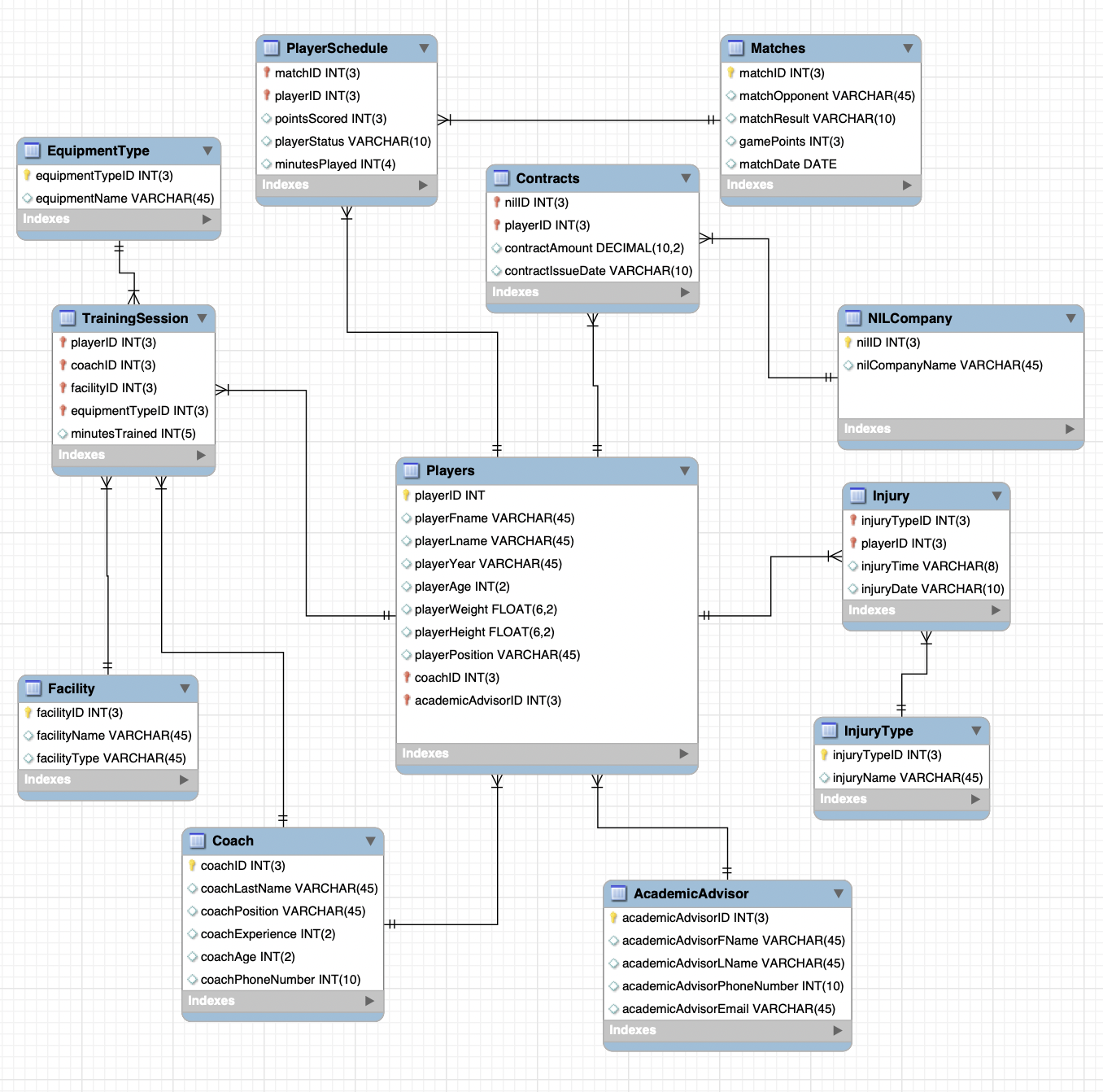
You are the owner/operator of a college football team, the Bulldogs, at a prestigious university. With a rich history of success on the field and a commitment to academic excellence, your football program is integral to the university's identity. As the owner, you understand the importance of building a comprehensive relational database to streamline operations and support the development of your student-athletes.

You've hired a team of students from the MIST 4610 class at the University of Georgia to create the database for you. To help them understand your organization, you provide insights into key aspects of your football program:

**Entities:**

1. Players
2. Coaches
3. Injury Type
4. Injury
5. Academic Advisors
6. Facility
7. Training Session
8. Equipment
9. Player Schedule
10. Position
11. NIL
12. NIL Contract

As the owner/operator of the Bulldogs football team, you're spearheading the implementation of a comprehensive relational database system to optimize team performance and support the development of your student-athletes. This database will efficiently manage essential aspects of the football program, including player management with detailed player information such as name, position, eligibility status, and academic standing, as well as coach details encompassing names, positions, and experience. Additionally, training schedules, workout routines, and medical records will be meticulously organized to optimize player performance and minimize injury risk during training sessions. Game planning and analysis will benefit from data on upcoming matchups and opponent details. Efficient team operations and logistics will be ensured through the management of practice facilities, equipment inventory, and usage. Academic support services for student-athletes will be facilitated by tracking academic advisor information, while injury prevention and management will be supported through the tracking of player injuries, treatments, and severity levels. By centralizing critical data and streamlining operations, this relational database will contribute to the continued success and excellence of the Bulldogs' football program.



10 Queries (4 simple and 6 complex)

1. List out the first name, last name, and position of all players who have an NIL contract that is worth more than the average contract amount. (simple)

Select playerFname, playerLname, playerPosition

from Players

join Contracts using(playerID)

where contractAmount > (select avg(contractAmount) from Contracts);

1. List out the first name and last name of all Juniors who are training for more than 200 minutes and whose academic advisor is Hannah Rossi. (complex)

select playerFname, playerLname

from Players

join TrainingSession using(playerID)

join AcademicAdvisor using(academicAdvisorID)

where playerYear = "Junior" and academicAdvisorLName = "Rossi"

group by 1,2

having sum(minutesTrained) > 200;

1. List out the last name of every player who has scored a point or points and what team they scored those points against. (This would use a GROUP BY for the player name because some will have scored points against multiple teams) (simple)

select playerLName, matchOpponent

from Players

join PlayerSchedule using (playerID)

join Matches using (matchID)

group by 1,2;

1. List out the last name and position of the coaches that have trained players greater than the average amount of time coaches train their players. (This should use SUM) (complex)

with

x as (select sum(minutesTrained) as mins

from Coach

join Players using (coachID)

join TrainingSession using (playerID)

group by Coach.coachID)

select coachLastName, coachPosition

from Coach

join Players using (coachID)

join TrainingSession using (playerID)

group by 1,2

having sum(minutesTrained) > (select avg(mins) from x);

1. List out the last name of every player who was injured in November of 2023 (2023-11) and the name of that injury. (simple)

select playerLName, injuryName

from Players

join Injury using (playerID)

join InjuryType using (injuryTypeID)

where injuryDate regexp("^2023-11");

1. List the first and last name of every player that has not scored a point and how much total NIL money they have earned from contracts. (complex)

select playerFName, playerLName, sum(contractAmount)

from Players

join PlayerSchedule using (playerID)

join Contracts using (playerID)

where pointsScored = 0  
group by playerFName, playerLName;

1. List out the player first name, player last name, player position, coach last name, and academic advisor last name of players whose coaches and academic advisors have the same area code in their phone numbers. Order the list by the players’ names. (complex)

select playerFName, playerLName, playerPosition, coachLastName, academicAdvisorLName

from Players

join Coach using (coachID)

join AcademicAdvisor using (academicAdvisorID)

where coachPhoneNumber regexp ("^404") and academicAdvisorPhoneNumber regexp("^404")

order by 1,2;

1. List out the NIL Companies that have not issued a contract to players, and do not contain the word “Twitter”. Order the list by company name in alphabetical order. (complex)

select nilCompanyName

from NILCompany

where not exists (select \* from Contracts where Contracts.nilID = NILCompany.nilID) and nilCompanyName not regexp("Twitter")

order by nilCompanyName;

1. List out the different types of facilities, the number of times training sessions have been held in those facilities, and the average weight of players that have participated in training sessions at those facilities. (complex)

select facilityType, count(facilityID), avg(playerWeight)

from Facility

join TrainingSession using (facilityID)

join Players using (playerID)

group by facilityType;

1. List out the average height and weight of players from each position. Order the results by height and weight in descending order. (simple)

select playerPosition, avg(playerHeight), avg(playerWeight), sum(contractAmount)

from Players

join Contracts using (playerID)

group by playerPosition

order by 4 desc;

**Query Matrix**

| Features | Query 1 | Query 2 | Query 3 | Query 4 | Query 5 | Query 6 | Query 7 | Query 8 | Query 9 | Query 10 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Multiple table join |  | x | x | x | x | x | x |  | x |  |
| subquery | x |  |  | x |  |  |  | x |  |  |
| GROUP BY |  |  | x | x |  | x |  |  | x | x |
| GROUP BY with HAVING |  | x |  | x |  |  |  |  |  |  |
| Multi condition WHERE |  | x |  |  |  |  | x |  |  |  |
| Built-in functions | x | x |  | x |  | x |  |  | x | x |
| REGEXP |  |  |  |  | x |  | x | x |  |  |
| NOT EXISTS |  |  |  |  |  |  |  | x |  |  |

**Table: Matches**

| Column Name | Description | Data Type | Size | Format | Key? |
| --- | --- | --- | --- | --- | --- |
| matchID | Single non-repeating identifier for each match | INT | 3 |  | PK |
| matchOpponent | Name of the opponent played in each match | VARCHAR | 45 |  |  |
| matchResult | Text recording whether the match was a win, loss, or a draw | VARCHAR | 10 |  |  |
| gamePoints | The total number of points scored in the game | INT | 3 |  |  |
| matchDate | The date the match was played | DATE |  | YYYY-MM-DD |  |

**Table: PlayerSchedule**

| Column Name | Description | Data Type | Size | Format | Key? |
| --- | --- | --- | --- | --- | --- |
| matchID | Single non-repeating identifier for each match | INT | 3 |  | FK |
| playerID | Single non-repeating identifier for each player | INT | 3 |  | FK |
| pointsScored | Number of points the player has scored in a specific match | INT | 3 |  |  |
| playerStatus | Text recording whether the player starts, is a sub, is injured, or is suspended during a match | VARCHAR | 10 |  |  |
| minutesPlayed | Number of minutes the player played in a match | INT | 4 |  |  |

**Table: Players**

| Column Name | Description | Data Type | Size | Format | Key? |
| --- | --- | --- | --- | --- | --- |
| playerID | Single non-repeating identifier for each player | INT | 3 |  | PK |
| playerFname | First name of every player | VARCHAR | 45 |  |  |
| playerLname | Last name of every player | VARCHAR | 45 |  |  |
| playerYear | Text recording each player’s year in college (Freshman - Senior, then 5th year etc.) | VARCHAR | 20 |  |  |
| playerAge | Age of every player | INT | 2 |  |  |
| playerWeight | Weight of every player (lbs) | FLOAT | 6,2 |  |  |
| playerHeight | Height of every player (cm) | FLOAT | 6,2 |  |  |
| playerPosition | Text recording the player’s position | VARCHAR | 45 |  |  |
| coachID | Single non-repeating identifier for each coach | INT | 3 |  | FK |
| academicAdvisorID | Single non-repeating identifier for each academic advisor | INT | 3 |  | FK |

**Table: Contracts**

| Column Name | Description | Data Type | Size | Format | Key? |
| --- | --- | --- | --- | --- | --- |
| nilID | Single non-repeating identifier for each NIL Company | INT | 3 |  | FK |
| playerID | Single non-repeating identifier for each player | INT | 3 |  | FK |
| contractAmount | Amount of money detailed in the contract | DECIMAL | 10,2 |  |  |
| contractIssueDate | Date the contract was issued | VARCHAR | 10 |  |  |

**Table: NILCompany**

| Column Name | Description | Data Type | Size | Format | Key? |
| --- | --- | --- | --- | --- | --- |
| nilID | Single non-repeating identifier for each NIL Company | INT | 3 |  | PK |
| nilCompanyName | Name of each NIL Company | VARCHAR | 45 |  |  |

**Table: Injury**

| Column Name | Description | Data Type | Size | Format | Key? |
| --- | --- | --- | --- | --- | --- |
| injuryType | Single non-repeating identifier for each injury | INT | 3 |  | FK |
| playerID | Single non-repeating identifier for each player | INT | 3 |  | FK |
| injuryTime | Approximate time the injury was sustained | VARCHAR | 8 | HH:MM:SS |  |
| injuryDate | Date the injury was sustained | VARCHAR | 10 |  |  |

**Table: InjuryType**

| Column Name | Description | Data Type | Size | Format | Key? |
| --- | --- | --- | --- | --- | --- |
| injuryTypeID | Single non-repeating identifier for each injury | INT | 3 |  | PK |
| injuryName | Name of the type of injury sustained | VARCHAR | 45 |  |  |

**Table: EquipmentType**

| Column Name | Description | Data Type | Size | Format | Key? |
| --- | --- | --- | --- | --- | --- |
| equipmentTypeID | Single non-repeating identifier for each equipment type | INT | 3 |  | PK |
| equipmentName | Name of each type of gear/training equipment | VARCHAR | 45 |  |  |

**Table: TrainingSession**

| Column Name | Description | Data Type | Size | Format | Key? |
| --- | --- | --- | --- | --- | --- |
| playerID | Single non-repeating identifier for each player | INT | 3 |  | FK |
| coachID | Single non-repeating identifier for each coach | INT | 3 |  | FK |
| facilityID | Single non-repeating identifier for each facility | INT | 3 |  | FK |
| equipmentTypeID | Single non-repeating identifier for each equipment type | INT | 3 |  | FK |
| minutesTrained | Number of minutes trained during the training session | INT | 5 |  |  |

**Table: Facility**

| Column Name | Description | Data Type | Size | Format | Key? |
| --- | --- | --- | --- | --- | --- |
| facilityID | Single non-repeating identifier for each facility | INT | 3 |  | PK |
| facilityName | Name of each facility | VARCHAR | 45 |  |  |
| facilityType | Type of facility (gym/track etc.) | VARCHAR | 45 |  |  |

**Table: Coach**

| Column Name | Description | Data Type | Size | Format | Key? |
| --- | --- | --- | --- | --- | --- |
| coachID | Single non-repeating identifier for each coach | INT | 3 |  | PK |
| coachLastName | Last name of each coach | VARCHAR | 45 |  |  |
| coachPosition | Position each coach is responsible for | VARCHAR | 45 |  |  |
| coachExperience | Total number of years each coach has been a coach | INT | 2 |  |  |
| coachAge | Age of each coach | INT | 2 |  |  |
| coachPhoneNumber | Phone number for each coach | INT | 10 | 9999999999 |  |

**Table: AcademicAdvisor**

| Column Name | Description | Data Type | Size | Format | Key? |
| --- | --- | --- | --- | --- | --- |
| academicAdvisorID | Single non-repeating identifier for each academic advisor | INT |  |  | PK |
| academicAdvisorFName | First name of each academic advisor | VARCHAR | 45 |  |  |
| academicAdvisorLName | Last name of each academic advisor | VARCHAR | 45 |  |  |
| academicAdvisorPhoneNumber | Phone number of each academic advisor | INT | 10 | 9999999999 |  |
| academicAdvisorEmail | UGA email of each academic advisor | VARCHAR | 45 |  |  |