

Daniel Regan
CSC6302 Database Principles
Professor Amanda Menier
Final Project Planning Phase
May 5, 2025

1. What is the theme of your application?

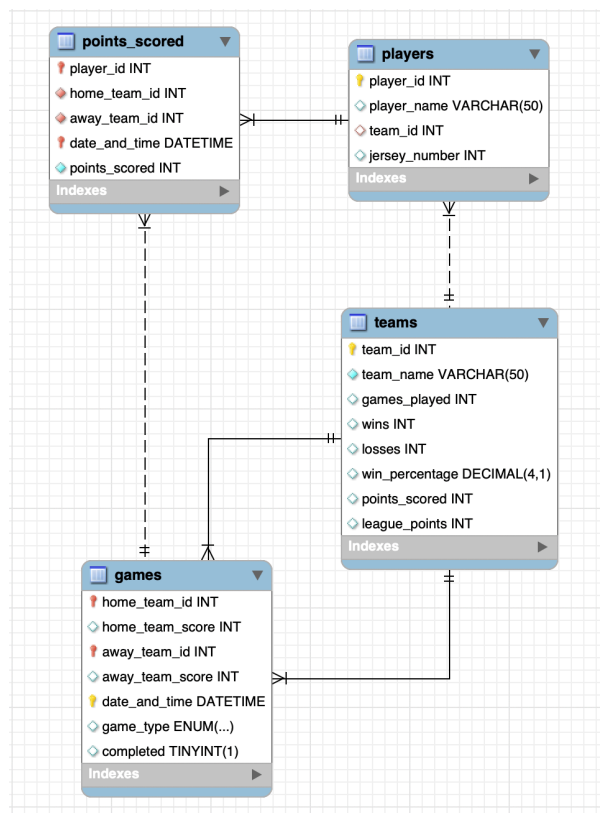
The theme of this project will be sports. I'd like to create an application that allows users to connect with a database that manages a pickup sports league. Users will be able to create teams, add players to those teams, input game information and results, and track basic player statistics.

2. What language(s) will you use? What frameworks will you use for your front-end and back-end?

I'll use Python to create a user interface and manage the app's logic, and MySQL will be used for the database. The libraries I'll be using are Tkinter and tabulate on the front-end and mysql-connector-python on the backend. Tkinter will help in creating the GUI, tabulate will help format tables within the GUI, and mysql-connector-python will manage the connection between Python and the database.

3. Create an ER diagram that shows your tables, all column names and datatypes, and all primary and foreign key designations and relationships.

- a. You should use a tool like MySQL workbench or Luna Modeler to create this diagram. You may *not* use AI to generate the table structure or diagram.



4. For your table that uses a composite key, list all superkeys. Identify the candidate keys, and describe why you chose the primary key.

Since both my points table and games table have composite keys, I'll answer this question by referencing the points table.

Superkeys:

{player_id, date_and_time}
{player_id, home_team_id, date_and_time}
{player_id, home_team_id, date_and_time, points}
{player_id, home_team_id, away_team_id, date_and_time}
{player_id, date_and_time, points}
{player_id, away_team_id, date_and_time, points}
{player_id, away_team_id, date_and_time}
{player_id, home_team_id, away_team_id, date_and_time, points}

Candidate keys:

{player_id, date_and_time}

Primary key:

{player_id, date_and_time}

I chose this primary key because no single player can be a part of more than one game at a time—you can't be in two places at once. Therefore, the player_id and date_and_time is sufficient enough to uniquely identify a tuple in this table.

5. Describe how you know your tables are in third normal form. What dependencies did you decompose?

My tables:

- 1) Don't use row order to convey information
- 2) Don't mix data types within the same column
- 3) Don't have tables without a primary key
- 4) Don't have repeating data groups
- 5) Don't have columns with the same name
- 6) Have non-key attributes that all depend on the primary key
- 7) Don't have transitive dependencies

I decomposed this database into four tables to create a manageable and precise framework to house the data. I separated the players from the teams to support changes to team rosters. I separated the points from the players to track player performance in specific games. I separated the games from the teams to track schedules independently of

changes to teams. And I separated the teams from the games and players to make changes to rosters without affecting the schedules and visa versa.

6. Describe the views you will include in your database. What is their purpose and what value do they add to the user?

I will create four views:

- 1) league_standings will show the user how teams are performing relative to one another.
- 2) team_roster will show the user which players are a member of the team.
- 3) schedule will show the user past game results and upcoming games.
- 4) top_scorers will show the user the 10 players with the most points scored.

All of these tables allow the user to gain a holistic understanding of the league, the teams' standings, and the players' standings. These are important views that will help the user track their league/teams/players over time and provide important information to members of the league.

7. Describe each stored procedure: Which table does it reference and what does it do?

- 1) add_team will reference the teams table and insert a new team into the table.
- 2) delete_team will reference the teams table and remove an existing team from the table.
- 3) add_player will reference the players table and insert a new player into the table.
- 4) add_game will reference the games table and insert a new game into the table.
- 5) get_team_roster will reference the players table and will get all the players of a certain team.
- 6) get_top_scorers will reference the points table and will get the top 10 best scorers in the league.
- 7) get_league_standings will organize and display the teams in order of how good they are, according to their league_points.
- 8) get_schedule will reference the games table and display the games of a particular team.

8. What advanced feature are you implementing in your application? What value does it add to the user?

The advanced feature will allow a user to download the schedule of a particular team as a .pdf file. This is beneficial to any user who follows a particular team. They can download their schedule and have it saved for easy reference for when their team is playing next.

9. What will your testing process be to ensure the sql and application run the first time with no issue?

I plan to test my application through a series of steps.

- 1) I will test each stored procedure with valid and valid inputs.
- 2) I'll ensure that views return accurate and complete data and that they're updated as information is added and deleted.
- 3) I'll use the app thoroughly, thinking of all ways a user might interact with my application.
- 4) I'll ensure that all buttons, views, and forms work correctly when interacted with both properly and incorrectly.
- 5) I'll ensure that valid and invalid inputs are handled correctly. I'll then test edge cases, such as when a player is added without being assigned a team.
- 6) I'll simulate normal user interaction to ensure the app flows well under normal use both when starting with no data and with test data.

10. Include a timeline of when each stage of your project, including testing and submission will be completed.

My database will be complete by Wednesday, May 7th. All tables, sample data, views, and stored procedures will be finalized.

I'll finish the GUI by Friday, May 9th. I'll have built and connected the user interface components to the database.

I'll test it and submit it on Saturday, May 10th. After functional and edge-case testing, I'll fix errors in the program that I find and submit the project before the deadline.

11. Which part of the project do you expect to be most difficult for you? How do you plan to mitigate that difficulty?

I anticipate that creating the GUI and connecting the components to the database will be the most challenging aspect of this project. Ensuring that user interface components look good and are intuitive, while also working the way they're intended to was a challenge in our last GUI project.

To mitigate this, I plan on dividing the app into smaller, more manageable sections, building and testing each section frequently, and saving my progress along the way. That way, I can troubleshoot more efficiently and prevent issues from becoming too overwhelming.

12. What questions do you have about the assignment?

Firstly, I'd like to know whether the scope of my project is reasonable. Am I too ambitious or should I look to add more to the app's functionality?

Secondly, for part two question eight, do I need to hard code the calls to each function/stored procedure/view, or will they be called manually through your review of the code?

Thank you!