A Recreational Sports League

This project will have you create tables to hold information about an imaginary sports league.

> The league will consist of teams, players, and games in some sport or other competition. A general description of the required tables is given, but the actual design of the table's details is up to you. Feel free to customize the data for your favorite sport or game.

Requirements

- The project should be in a github project named `sports_league`.
- Each of the following numbered steps should be in a file named "`step##.sql`". Each file will be sourced from a mysql> interactive prompt.

`step01.sql`	Create a table to hold data about teams. Each team has a name and a unique id.
`step02.sql`	Create a table to keep track of players. Each player has a unique id, a first name, and a last name. A player can <u>optionally</u> be assigned to a team. If assigned to a team, a player also has a player number.
`step03.sql`	Create a separate table to store personal info about players: age, height, weight, etc.
`step04.sql`	Create a table to keep track of games played. Each game has a unique id, date and start time, home team id, and vistors' team id.
`step05.sql`	Create a table to store each team's score for each game. There should be two records in this table for each game, (one for the home team, one for the visitor's team). This table should have columns to contain the following data: game id, team id, points scored, win or lose (this column can contain "Win" or "Lose", "Y" or "N", etc.
`step06.sql`	Make up some data and insert records into each table. There should be a minimum of 4 teams and a minimum of 16 players within the league.
`step07.sql`	What is average height of all players?
`step08.sql`	Print a list of players in the following format, alphabetized by last name. The first column of the output should be of the form "`last name, first name`" with a field header of "`name`". The other columns should display the age height and weight of each player.
`step09.sql`	Given a team id, list the players' names and numbers, ordered by player number.
`step10.sql`	What is the highest score for any team in a single game?
`step11.sql`	Which team has the highest total of points scored in all games played by that team?
`step12.sql`	What is the highest number of points scored by both teams in a single game?
`step13.sql`	List the players that are not assigned to a team.
`step14.sql`	How many teams are there?
`step15.sql`	How many players are there?
`step16.sql`	How many games were played on a given date?
`step17.sql`	Who is the tallest player?

`step18.sql`	Print a report showing all games and game dates. The report should include the home and visitors' team names.
`step19.sql`	Which team has the most wins?
`step20.sql`	Which team has the highest average score per game?
`step21.sql`	Print a report that shows each team's win-loss record. Sort the listing by number of wins.
`step22.sql`	Print a report that shows the final score of each game.
`step23.sql`	Create an SQL script that will drop any and all tables created in your project if they exist.
`step24.sql`	Create a single SQL script with statements to print the entire contents of each of the tables in the project.

Challenges

- Create a Python script named **`step25.py`** that when run prompts the user for a table name. The Python script then creates a SQL script named **`step25.sql`** that is a select statement that selects all the records for the given table name. (This requires no database access from within the Python script. You are simply creating a string representing the select query and writing it out to a text file.)
- There is the ability in MYSQL to use the "`SELECT ... INTO`" Syntax to output the contents of the select to a file. Write a SQL script named **`step26.sql`** that will output comma separated tables with one record per line in the output for each of the tables within your project. The filenames should be "`tablename.txt`" and be written to the same directory as all of SQL scripts for this project.

Rubric Weights	
Correct	30%
Readable	30%
Architecture	30%
WriteUp	10%