Team AG_LOL

Ameya S Rathod 2022111021 Annamalai Senthil 2022101032
Anish R Joishy 2022111014 Ashwin Kumar 2022101091
Kushal Shah 2022101059

This report consists of the list of all commands that the given code is able to execute. Here we go !

S.No	Command Name	Description
1.	Show all teams in a league	The command takes as input the league id and the corresponding query is : Select * from (Team join Plays_in on Team.Team_ID=Plays_in.Team_ID) Where LeagueID ="+n. It joins the tables Team and Plays_in based on the foreign key Team_id. This command can be used by the organiser and fans if they need to know about all the teams in the league.
2.	Show Player with victory greater than ‘n’	It takes the number of victories by a player as the input and the corresponding sql query is : "Select * from (Players join Player_Stats on Players.Player_ID=Player_Stats.Player_ID) Where No_of_Matches_won_singles + No_of_Matches_won_doubles >"+n This query initially joins the tables Players and Players_stat tables on the foreign key Player_id and then displays those players who have their victory count greater than n, the total count is calculated by adding the matches won in singles and doubles format. It can be used by the organisers and fans in order to filter out the players based on the number of victories.

3.	Delete Team	It takes the team ID as the input and the sql query is : "delete from Team where id="+id This is a simple query and simply deletes the team whose id matches with the input id given. This command is in particular useful for the organiser in case a team is banned or disqualified from the league.
4.	Delete Player	The input is the player ID and query is : "delete from Player where id="+id. Similar to the previous query, it deletes the player whose id matches with the input id given. It can be used by the organiser if a player decides to leave the league.
5.	Show all stadiums and their City	This command does not require any input from the user. The corresponding SQL query is : "Select Name,City from Stadium" This query simply selects all the Stadium names and their cities from the table and displays them. It can be used by the organisers and fans in order to have the details of the venues in the league.
6.	Insert Team	It takes multiple inputs : Team_id, Team_name, Format and Head coach. The sql query is : "INSERT INTO Team (Team_ID, Team_Name, Format, No_of_Matches_Played, No_of_Players, Head_Coach, No_of_Wins) VALUES ({Team_id}, {Team_name}, {format}, {no_of_Matches_Played}, {no_of_Players}, {head_coach}, {no_of_Wins});" It uses the ‘insert into’ command and using the input values, stores them into their corresponding volume in the table. This is useful for the league organiser.
7.	Insert Player	The inputs here are player_id, first name, middle name, last name, DOB, gender, country, handed. The sql query is : "INSERT INTO Team (Team_ID, Team_Name, Format, No_of_Matches_Played, No_of_Players,

		Head_Coach, No_of_Wins) VALUES ({Team_id}, {Team_name}, {format}, {no_of_Matches_Played}, {no_of_Players}, {head_coach}, {no_of_Wins});" Exactly like the ‘insert team’ command, but with different input parameters. Used by the organisers to add a new player.
8.	Get id of Players in a Team	It takes as input the team_id and the sql command is : select Player_ID from Belongs_To where Team_ID="+id We simply iterate through the rows we get in after running the above command, and then using for loops simply print the Player_id from a given team. It can be used by the organisers and fans in order to get the id of the players in a given team.
9.	Add Player in a Team	It takes the playd_id as well as team_id as input. The sql commands used are : Insert into Belongs_To values ({player_id},{team_id}); Update Team SET No_of_Players=No_of_Players+1; We just add the player with the input id into the team which we want to. It can be used by the organisers when a new player comes to the league and his data needs to be added.
10.	Insert League	This takes the league name, date of beginning, date of end, format and country as input. The corresponding SQL query is quite large to write in the table, and can be seen in the code. It can be used by the organisers to create a new league and store its details in the database.
11.	Add Match	Its inputs are the id’s of both the teams, corresponding league id, also the date and format of match. Insert into Matches (Team_1_ID,Team_2_ID,Format,Date,League_ID,Team_1_Score,Team_2_Score) values ('{team_id1}','{team_id2}','{format}','{date}','{league_id}',0,0) Using the above command we add the details of that particular match into the Matches table and substitute the

		attribute values using the given inputs. It can be used by the organisers to schedule a new match between two teams.
12.	Update Match Score	Its inputs are match_id, refree_id, the scores of both the teams. The data for all the players involved, team's matches details and referee details all get updated accordingly based on the scoreline of the match. As it's a complex command, its SQL query cannot be written in this table, it can be found easily under the 'scoreline' function in the given code. This can be used by the organisers and can save a lot of extra work for them by taking care of all the required relational changes throughout the database.
13.	Delete a Player from a Team	It takes the player_id and his team_id as input. The sql query goes like : DELETE FROM Belongs_To where Player_ID = {Player_ID} and Team_ID = {Team_ID}; Using the delete command we delete the player whose id matches with the given player and team id. Used by the organisers to delete the details of a player in case he/she left the team.
14.	Delete a Team from a League	It takes the team_id and league_id as the input. The sql query is : DELETE FROM Plays_in where Team_ID = {Team_ID} and League_ID = {League_ID}; Using the delete command the team is deleted whose league id and team id matches with the input values. Used bv the organisers to deregister a team from the league due to any reason.
15.	Add Team in a League	The inputs are team_id and league_id. The sql query is : INSERT INTO Plays_in (Team_ID,League_ID)VALUES ({Team_ID},{League_ID} It essentially is the exact opposite of the previous command and adds a team to a league. By the organisers if a new team registers for the league.