Assignment 2
COL362/632

February 17, 2018

General Instructions

1. Do this assignment in groups of 2 people.
2. **The assignment is due on 23rd February,11:59 PM. For each late day, you will lose 25% of points**.
3. Use Postgres for this assignment. These tutorials will help you get started
   - `http://www.w3resource.com/PostgreSQL/tutorial.php`
   - `http://www.tutorialspoint.com/postgresql/index.htm`
4. You will submit 1 .sql file: **assignment2_< Entry number 1>_<Entry number 2>.sql**. ONLY ONE MEMBER OF THE GROUP SHOULD SUBMIT ON MOODLE. Please make sure you follow this naming convention and mention your entry numbers in the file name (of the form 2014CSXXXXX), and **not the Kerberos IDs (like csXXXXXXX)**.
5. The .sql files are run automatically, so please ensure that there are no syntax errors in the file. *If we are unable to run your file, you get an automatic reduction to 0 marks.*
6. The format of the file should be as follows. All queries should be in a *single* line. Leave a blank line after each query. Write the question number in comments before each query.An example solution file can be found here (`www.cse.iitd.ac.in/~prajna/col362_2018/Assignment_2/example.sql`).
7. Many of the queries below require an 'ORDER BY' clause. Using these clauses would by default return results in *increasing* order. If you omitted this clause, your answer will be evaluated as incorrect and zero marks will be awarded.
8. Errors with respect to equality and inequality conditions will also be evaluated as incorrect and zero marks will be awarded.
9. **No changes are allowed in the i) data, ii) attribute names, iii) table names**
10. **You are NOT allowed to use views.**

1 Dataset

1.1 Instructions

1. This assignment deals with the Indian Premier League dataset. It contains information about 568 cricket matches played in the duration from 2008-2016. The datasets and their schemas are described below. The primary keys are underlined.

1. `www.cse.iitd.ac.in/~prajna/col362_2018/Assignment_2/player.csv`
   This file lists the information of all the players in IPL. The schema is described as follows:

Column Name	Data Type
player_id	integer
player_name	text
dob	date
batting_hand	text
bowling_skill	text
country_name	text

2. `www.cse.iitd.ac.in/~prajna/col362_2018/Assignment_2/team.csv`
   This file describes the teams participating in IPL. Following is the schema:

Column Name	Data Type
team_id	integer
name	text

3. `www.cse.iitd.ac.in/~prajna/col362_2018/Assignment_2/match.csv`
   It contains the information about all matches.

Column Name	Data Type
match_id	integer
team_1	integer
team_2	integer
match_date	date
season_id	integer
venue	text
toss_winner	integer
toss_decision	text
win_type	text
win_margin	integer
outcome_type	text
match_winner	integer
man_of_the_match	integer

   Note : season_id can take values from 1-9

4. `www.cse.iitd.ac.in/~prajna/col362_2018/Assignment_2/player_match.csv`
   Describes the role and the team represented by the player in a match.

Column Name	Data Type
match_id	integer
player_id	integer
role	text
team_id	integer

5. `www.cse.iitd.ac.in/~prajna/col362_2018/Assignment_2/ball_by_ball.csv`
   Describes the ball by ball information of any match.

Column Name	Data Type
match_id	integer
over_id	integer
ball_id	integer
innings_no	integer
team_batting	integer
team_bowling	integer
striker_batting_position	integer
striker	integer
non_striker	integer
bowler	integer

Note : over_id can take values from 1-20, ball_id can take values from 1-9 and innings_no can take values from 1-4

6. `www.cse.iitd.ac.in/~prajna/col362_2018/Assignment_2/batsman_scored.csv`
Contains information about run scored in every ball of match.

Column Name	Data Type
match_id	integer
over_id	integer
ball_id	integer
runs_scored	integer
innings_no	integer

7. `www.cse.iitd.ac.in/~prajna/col362_2018/Assignment_2/wicket_taken.csv`
Contains information about the batsman who got out in a ball of the match.

Column Name	Data Type
match_id	integer
over_id	integer
ball_id	integer
player_out	integer
kind_out	text
innings_no	integer

8. `www.cse.iitd.ac.in/~prajna/col362_2018/Assignment_2/extra_runs.csv`
Extra runs given in a particular ball of a match.

Column Name	Data Type
match_id	integer
over_id	integer
ball_id	integer
extra_type	text
extra_runs	integer
innings_no	integer

1.2 Queries (the column ordering of outputs are mentioned in braces after each query. Your output should come exactly in that order. Please do not print the angular braces in the output.) Total: 53 marks

1. List the names of all left-handed batsmen from England. Order the results alphabetically. (<player_name>) [2 marks]

2. List the names and age (in years, should be integer) as on 2018-12-02 (12th Feb, 2018) of all bowlers with skill "Legbreak googly" who are 28 or more in age. Order the result in decreasing order of their ages. Resolve ties alphabetically. (<player_name, player_age>) [2 marks]

3. List the match_ids and toss winning team IDs where the toss winner of a match decided to bat first. Order result in increasing order of match_ids. (<match_id, toss_winner>) [2 marks]

4. In the match with match_id 335987, list the over_ids and runs scored where at most 7 runs were scored. Order the over_ids in decreasing order of runs scored. Resolve ties by listing the over_ids in increasing order. (<over_id, runs_scored>)[2 marks]

5. List the names of those batsmen who were bowled at least once in alphabetical order of their names. (<player_name>) [2 marks]

6. List all the match_ids along with the names of teams participating (team 1, team 2), name of the wining team, and win margin where the win margin is at least 60 runs, in increasing order of win margin. Resolve ties by listing the match ids in increasing order. (<match_id, team_1, team_2, winning_team_name, win_margin>) [2 marks]

7. List the names of all left handed batsmen below 30 years of age as on 2018-12-02 (12th Feb, 2018) alphabetically. (<player_name>) [2 marks]

8. List the match wise total for the entire series. The output should be match_id, total runs. Return the results in increasing order of match ids. (<match_id, total_runs>) [2 marks]

9. For each match_id, list the maximum runs scored in any over and the bowler bowling in that over. If there is more than one over having maximum runs, return all of them and order them in increasing order of over_id. Order results in increasing order of match ids. (<match_id, maximum_runs, player_name>) [3 marks]

10. List the names of batsmen and the number of times they have been "run out" in decreasing order of being "run out". Resolve ties alphabetically. (<player_name, number>) [2 marks]

11. List the number of times any batsman has got out for any out type. Return results in decreasing order of the numbers. Resolve ties alphabetically (on the out type name). (<out_type, number>) [2 marks]

12. List the team name and the number of times any player from the team has received man of the match award. Order results alphabetically on the name of the team. (<name, number>) [2 marks]

13. Find the venue where the maximum number of wides have been given. In case of ties, return the one that comes before in alphabetical ordering. Output should contain only 1 row. (<venue>) [2 marks]

14. Find the venue(s) where the team bowling first has won the match. If there are more than 1 venues, list all of them in order of the number of wins (by the bowling team). Resolve ties alphabetically. (<venue>) [3 marks]

15. Find the bowler who has the best average overall. Bowling average is calculated using the following formula:

$$bowling\_average = \frac{Number\_of\_runs\_given}{Number\_of\_wickets\_taken} \tag{1}$$

Calculate the average upto 3 decimal places and return the bowler with the *lowest* average runs per wicket. In case of tie, return the results in alphabetical order. (<player_name>) [4 marks]

16. List the players and the corresponding teams where the player played as "CaptainKeeper" and won the match. Order results alphabetically on the player's name. (<player_name, name>) [3 marks]

17. List the names of all players and their runs scored (who have scored at least 50 runs in any match). Order result in decreasing order of runs scored. Resolve ties alphabetically. (<player_name, runs_scored>) [2 marks]

18. List the player names who scored a century but their teams lost the match. Order results alphabetically. (<player_name>) [3 marks]

19. List match ids and venues where KKR has lost the game. Order result in increasing order of match ids. (<match_id, venue>) [3 marks]

20. List the names of top 10 players who have the best batting average in season 5. Batting average can be calculated according to the following formula:

$$batting\_average(player) = \frac{Number\_of\_runs\_scored\_by\_player}{Number\_of\_matches\_player\_has\_batted\_in} \tag{2}$$

The output should contain exactly 10 rows. Report results upto 3 decimal places. Resolve ties alphabetically. (<player_name>) [4 marks]

21. Using the formula provided in the previous query, find out the batting average of all players across *all* seasons. The batting average of a country X is:

$$country\_batting\_average = \sum_{n \in player\_of\_country\_X} \frac{batting\_average(n)}{|player\_of\_country\_X|} \tag{3}$$

List the top 5 countries with the highest *country_batting_average*. Report results upto 3 decimal places. Output may contain 5 or more rows. Resolve ties alphabetically. (<country_name>) [4 marks]