SQL

Complete the problems below. One Jupyter Notebook cell, one problem. Submission for this assignment will require 2 parts: your your Jupyter Notebook code (.ipynb), and a PDF of your notebook.

WORLD CUP BASIC QUERIES

1) What player on a team with "ia" in the team name played less than 200 minutes and made more than 100 passes? Return the player surname. Note: To check if attribute A contains string S use "A like '%S%'".

2) Find all players who made more than 100 shots. Return all information about the players in descending order of shots made.

3) Find the goalkeepers of teams that played more than four games. List the surname of the goalkeeper, the team, and the number of minutes the goalkeeper played.

4) How many players who play on a team with ranking <10 played more than 350 minutes? Return one number in a column named 'superstar'.

5) What is the average number of passes made by forwards? By midfielders? Write one query that gives both values.

6) Which team has the highest average number of passes per minute played? Return the team and average passes per minute. Hints: (1) You can compute a team's average number of passes per minute played by dividing the total number of passes by the total number of minutes. To force floating point division, multiply one operand by 1.0. (2) Consider using Limit.

WORLD CUP ADVANCED QUERIES

1) Find all pairs of teams who have the same number of goalsFor as each other and the same number of goalsAgainst as each other. Return the teams and numbers of goalsFor and goalsAgainst. Make sure to return each pair only once.

2) Find all teams with ranking <30 where no player made more than 150 passes. Return the team and ranking.

3) Which team has the highest ratio of goalsFor to goalsAgainst?

4) Find all team-position pairs where the average number of passes made by players in that position on that team is greater than 150. Return the team-position pairs.

5) Find all teams whose defenders averaged more than 150 passes. Return the team and

average number of passes by defenders, in descending order of average passes.

TITANIC BASIC QUERIES

1) How many married women over age 50 embarked in Cherbourg?

Note: To check if attribute A begins with string S use "A like 'S%"'

2) List the average fare paid by passengers in each of the embarkation cities (along with the city), in descending order of average fare.

3) What's the most common last name among passengers?

4) Write three queries: the total number of passengers; the number of passengers under 30; the number of passengers 30 or older. Why do the second and third numbers not add up to the first?

5) Blanks in SQL tables are given a special value called 'null', and conditions 'A is null' and 'A is not null' can be use in Where clauses to check whether attribute A has the 'null' value. How many passengers don't have a value for age? Now do your numbers add up?

6) For male survivors, female survivors, male non-survivors, and female non-survivors, how many passengers were in each of those four categories and what was their average fare?

TITANIC ADVANCED QUERIES

1) Are there any pairs of passengers with the same last name where one is in first class and the other is in third class? If so, return the last name and the two first names. Label the first-name columns 'first' for the passenger in first class and 'third' for the passenger in third class.

2) Which embarkation cities have more than 40 passengers whose age is missing? Reminder: Blanks in SQL tables are given a special value called 'null', and conditions 'A is null' and 'A is not null' can be used in Where clauses to check whether attribute A has the 'null' value.

3) Find all classes where the average fare paid by passengers in that class was more than twice the overall average or less than half the overall average.

4) EXTRA CREDIT CHALLENGE: List each class and its survival rate, i.e., the fraction of passengers in that class who survived.

TITANIC DATA MODIFICATION

1) Subtract 5 from the fare paid by any passenger under the age of 10. Then compute the new average fare. NOTE: You can put two SQL statements in one cell separated by a semicolon.

2) Create a new table called Survivors, containing the last and first names of all passengers who survived. Then count the number of tuples in the new table.

3) In the Titanic table delete all but the highest-paying passengers.
4) In what's left of the table after (3), insert a new tuple for yourself. You can decide your class, fare, where you embarked, and whether you survived. Then show the whole table.

TITANIC DATA MODIFICATION

1) Subtract 5 from the fare paid by any passenger under the age of 10. Then compute the new average fare. NOTE: You can put two SQL statements in one cell separated by a semicolon.

2) Create a new table called Survivors, containing the last and first names of all passengers who survived. Then count the number of tuples in the new table.

3) In the Titanic table delete all but the highest-paying passengers.

4) In what's left of the table after (3), insert a new tuple for yourself. You can decide your class, fare, where you embarked, and whether you survived. Then show the whole table.