Q1

YARN was introduced to address limitations in MapReduce. What does YARN offer that MapReduce couldn’t address? (Answer in less than three sentences.)

YARN is responsible for cluster resource management and job scheduling means which job will be executed by which system get decide by YARN, while MapReduce is just a programming framework to work on a particular job. YARN took over the responsibility of managing resources from MapReduce and started to give Hadoop the ability to run non-MapReduce jobs within the Hadoop framework.

Q2

Provide an example of HDFS command that will copy a file (File 1) from local file system to your folder in HDFS:

```
$ hdfs dfs -copyFromLocal/global/project/file1/user/barry
```

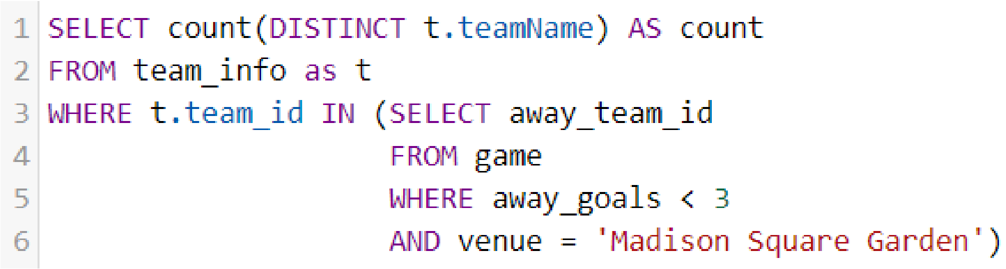
Q3

How would you display contents of a file stored in HDFS?

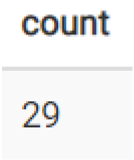
```
$ hdfs dfs -cat Lecture 1/game-info.csv
```

Q4: Practice subquery

Use subquery and count unique team names of teams that played as **away team** at Madison Square Garden and scored less than 3.



```sql
SELECT count(DISTINCT t.teamName) AS count
FROM team_info as t
WHERE t.team_id IN (SELECT away_team_id
                    FROM game
                    WHERE away_goals < 3
                    AND venue = 'Madison Square Garden')
```

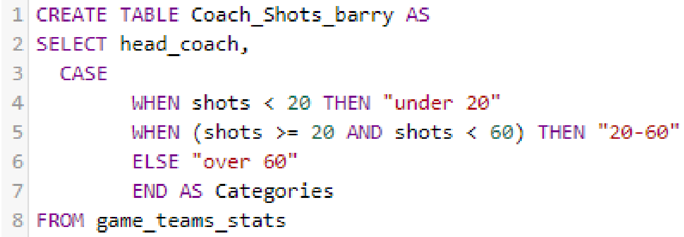


count
29

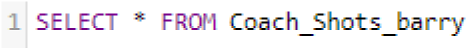
Q5: Practice CASE function

- Create a new table **'Coach_Shots' that** contain two columns:
- First column shows head coachs' names

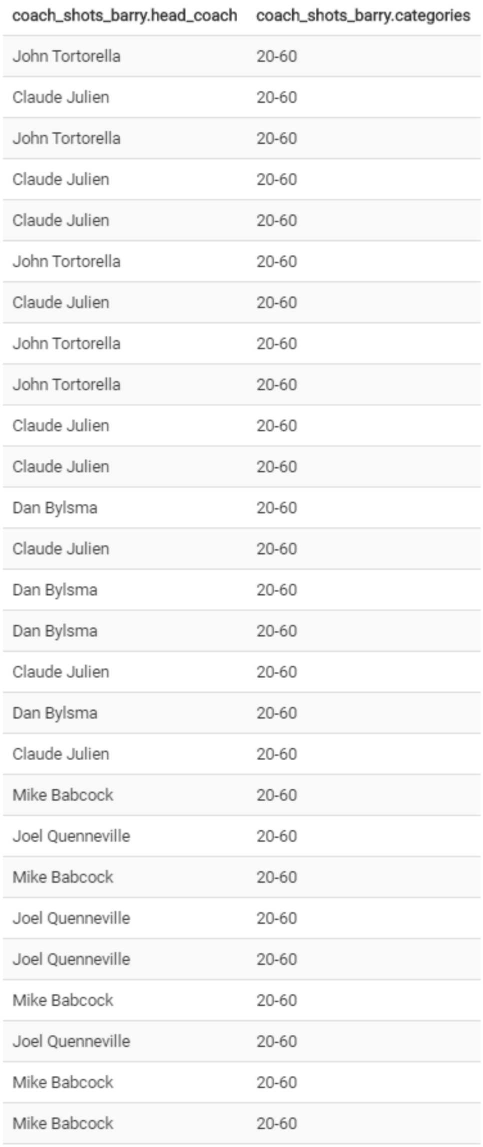
Second column displays information about shots, number of shots is presented using three categories: under 20, 20-60, and over 60.



```
CREATE TABLE Coach_Shots_barry AS
SELECT head_coach,
  CASE
        WHEN shots < 20 THEN "under 20"
        WHEN (shots >= 20 AND shots < 60) THEN "20-60"
        ELSE "over 60"
        END AS Categories
FROM game_teams_stats
```



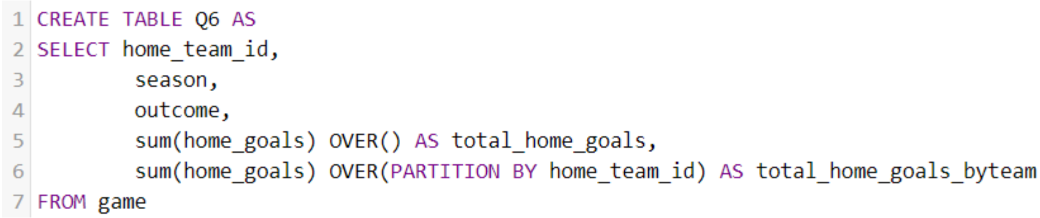
```
SELECT * FROM Coach_Shots_barry
```



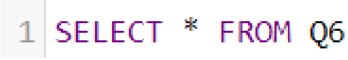
coach_shots_barry.head_coach	coach_shots_barry.categories
John Tortorella	20-60
Claude Julien	20-60
John Tortorella	20-60
Claude Julien	20-60
Claude Julien	20-60
John Tortorella	20-60
Claude Julien	20-60
John Tortorella	20-60
John Tortorella	20-60
Claude Julien	20-60
Claude Julien	20-60
Dan Bylsma	20-60
Claude Julien	20-60
Dan Bylsma	20-60
Dan Bylsma	20-60
Claude Julien	20-60
Dan Bylsma	20-60
Claude Julien	20-60
Mike Babcock	20-60
Joel Quenneville	20-60
Mike Babcock	20-60
Joel Quenneville	20-60
Joel Quenneville	20-60
Mike Babcock	20-60
Joel Quenneville	20-60
Mike Babcock	20-60
Mike Babcock	20-60

Q6: Practice OVER Function

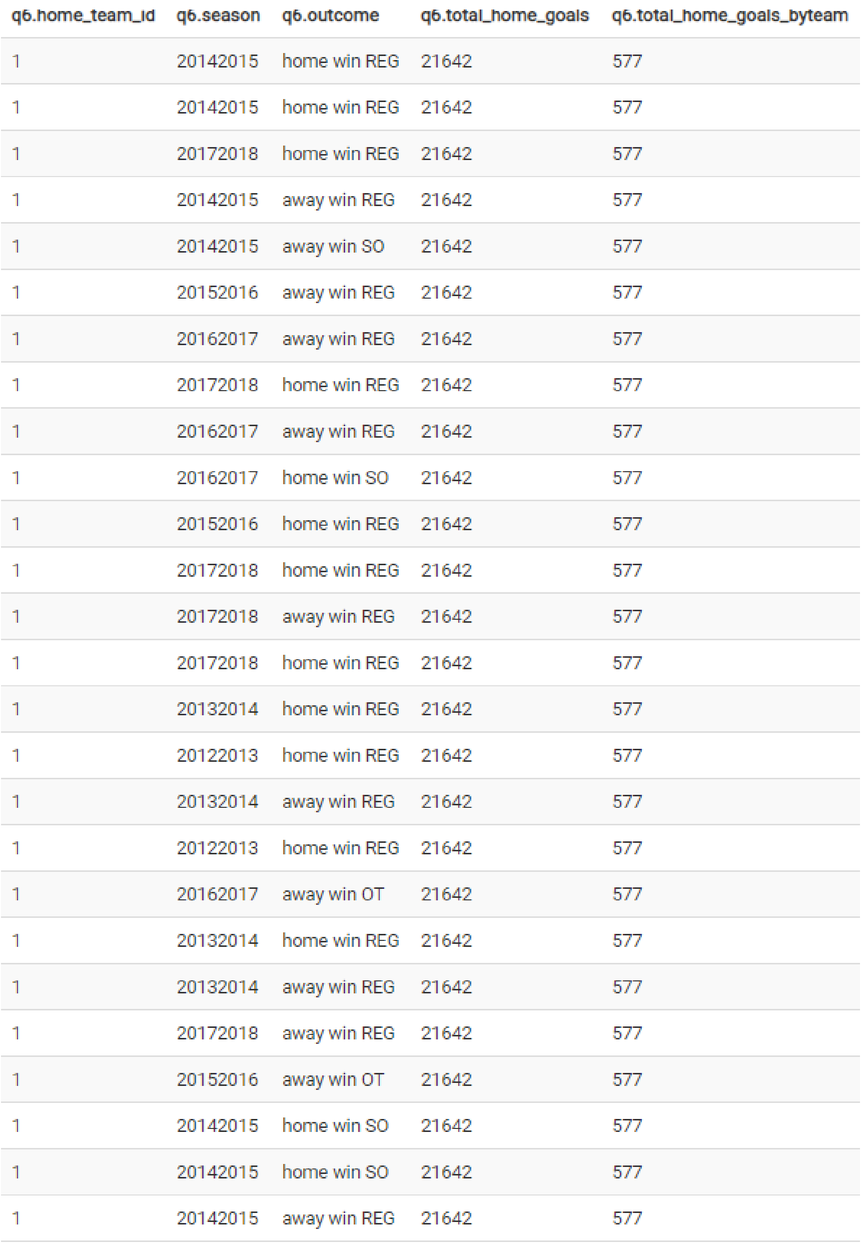
Use OVER function to create a table with five columns: home team id, season, outcome, and total home goals for all teams in the history, total home goals of this team in the history.



```
CREATE TABLE Q6 AS
SELECT home_team_id,
        season,
        outcome,
        sum(home_goals) OVER() AS total_home_goals,
        sum(home_goals) OVER(PARTITION BY home_team_id) AS total_home_goals_byteam
FROM game
```



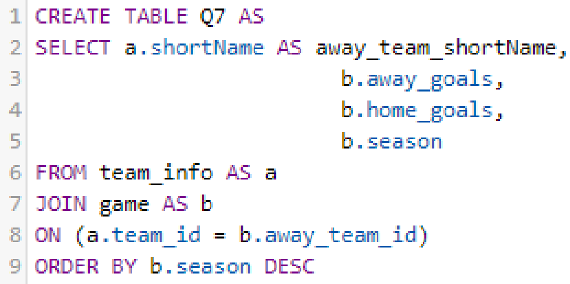
```
SELECT * FROM Q6
```



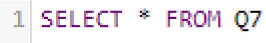
q6.home_team_id	q6.season	q6.outcome	q6.total_home_goals	q6.total_home_goals_byteam
1	20142015	home win REG	21642	577
1	20142015	home win REG	21642	577
1	20172018	home win REG	21642	577
1	20142015	away win REG	21642	577
1	20142015	away win SO	21642	577
1	20152016	away win REG	21642	577
1	20162017	away win REG	21642	577
1	20172018	home win REG	21642	577
1	20162017	away win REG	21642	577
1	20162017	home win SO	21642	577
1	20152016	home win REG	21642	577
1	20172018	home win REG	21642	577
1	20172018	away win REG	21642	577
1	20172018	home win REG	21642	577
1	20132014	home win REG	21642	577
1	20122013	home win REG	21642	577
1	20132014	away win REG	21642	577
1	20122013	home win REG	21642	577
1	20162017	away win OT	21642	577
1	20132014	home win REG	21642	577
1	20132014	away win REG	21642	577
1	20172018	away win REG	21642	577
1	20152016	away win OT	21642	577
1	20142015	home win SO	21642	577
1	20142015	home win SO	21642	577
1	20142015	away win REG	21642	577

Q7: Practice JOIN

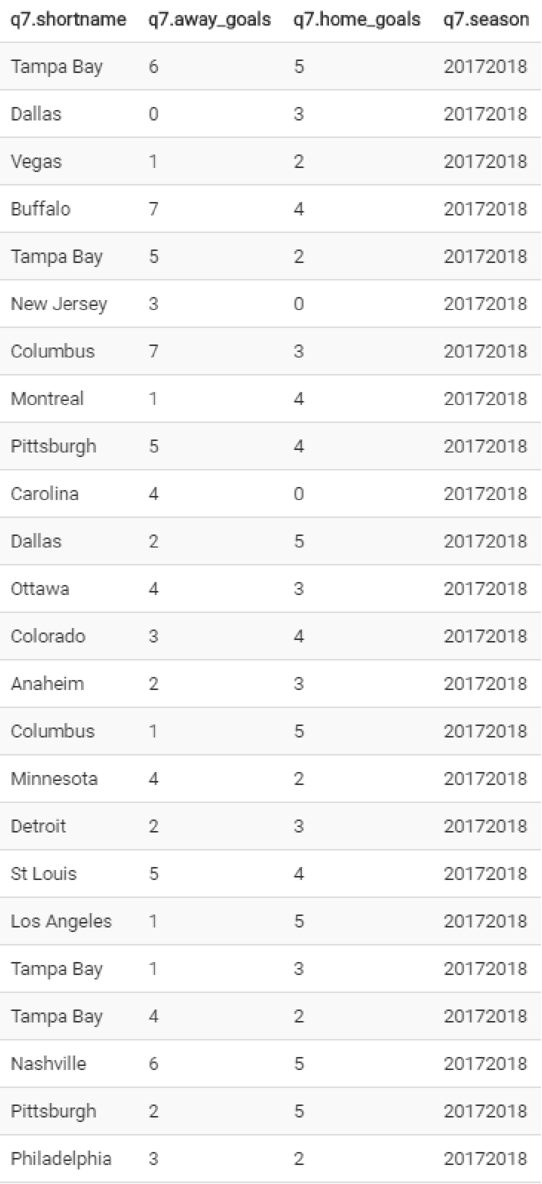
Created a table that has four columns: away team’s short name, away goals, home goals and season. Order records by season starting with most recent season.



```
CREATE TABLE Q7 AS
SELECT a.shortName AS away_team_shortName,
                        b.away_goals,
                        b.home_goals,
                        b.season
FROM team_info AS a
JOIN game AS b
ON (a.team_id = b.away_team_id)
ORDER BY b.season DESC
```



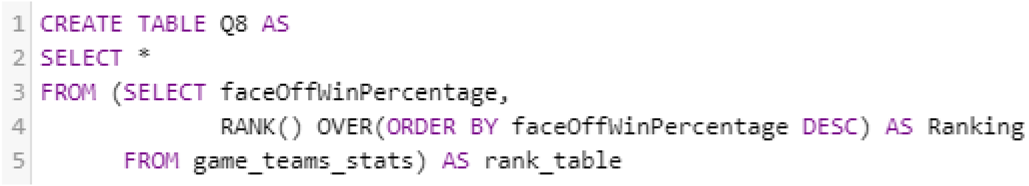
```
SELECT * FROM Q7
```



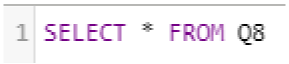
q7.shortname	q7.away_goals	q7.home_goals	q7.season
Tampa Bay	6	5	20172018
Dallas	0	3	20172018
Vegas	1	2	20172018
Buffalo	7	4	20172018
Tampa Bay	5	2	20172018
New Jersey	3	0	20172018
Columbus	7	3	20172018
Montreal	1	4	20172018
Pittsburgh	5	4	20172018
Carolina	4	0	20172018
Dallas	2	5	20172018
Ottawa	4	3	20172018
Colorado	3	4	20172018
Anaheim	2	3	20172018
Columbus	1	5	20172018
Minnesota	4	2	20172018
Detroit	2	3	20172018
St Louis	5	4	20172018
Los Angeles	1	5	20172018
Tampa Bay	1	3	20172018
Tampa Bay	4	2	20172018
Nashville	6	5	20172018
Pittsburgh	2	5	20172018
Philadelphia	3	2	20172018

Q8: Practice sub queries

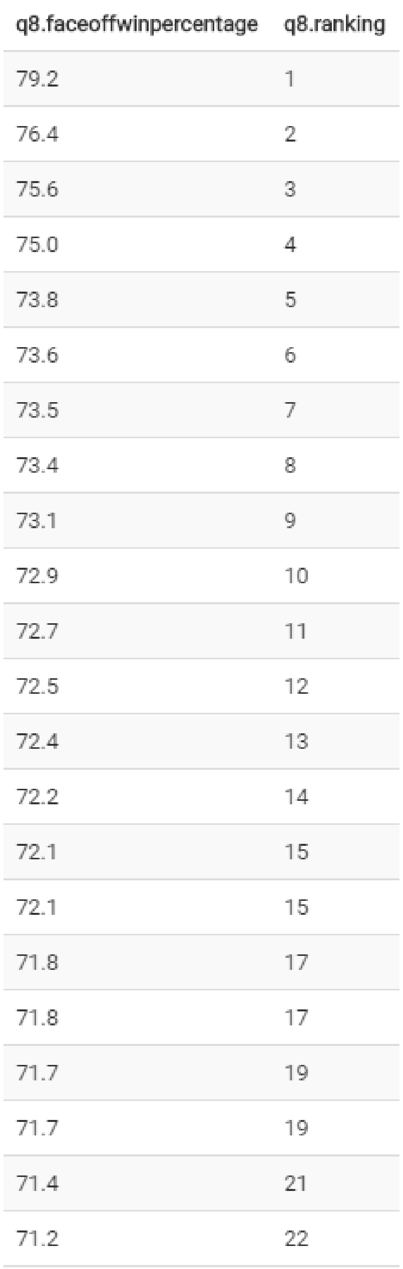
Created a table that has two columns: face Off Win Percentage, rank(their ranking). Sort by their ranking.



```
CREATE TABLE Q8 AS
SELECT *
FROM (SELECT faceOffWinPercentage,
             RANK() OVER(ORDER BY faceOffWinPercentage DESC) AS Ranking
      FROM game_teams_stats) AS rank_table
```

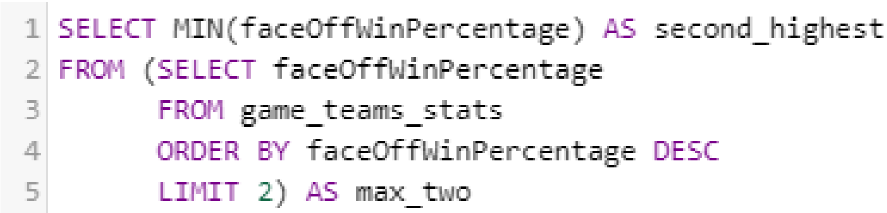


```
SELECT * FROM Q8
```

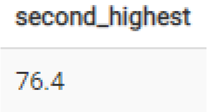


q8.faceoffwinpercentage	q8.ranking
79.2	1
76.4	2
75.6	3
75.0	4
73.8	5
73.6	6
73.5	7
73.4	8
73.1	9
72.9	10
72.7	11
72.5	12
72.4	13
72.2	14
72.1	15
72.1	15
71.8	17
71.8	17
71.7	19
71.7	19
71.4	21
71.2	22

Q9: Select the second highest face Off Win Percentage, a one column table with one row



```sql
SELECT MIN(faceOffWinPercentage) AS second_highest
FROM (SELECT faceOffWinPercentage
      FROM game_teams_stats
      ORDER BY faceOffWinPercentage DESC
      LIMIT 2) AS max_two
```



second_highest
76.4