

CS3700 Introduction to Database Systems

Assignment 3: SQL on Group DB

CS18B050 Aniswar Srivatsa Krishnan

1)

Description:

Obtain the leaderboard for the tournament, i.e, get the number of wins for each team sorted in the descending order of wins.

Query:

```
select team_name, T.c as wins
  from
    (select result_winning_team_id, count(*) as c
     from Matches
     group by result_winning_team_id) as T, Team
 where T.result_winning_team_id = team_id
 order by wins desc;
```

Result:

| team_name | wins |
|-----------------------------|------|
| Mumbai Indians | 80 |
| Chennai Super Kings | 79 |
| Royal Challengers Bangalore | 69 |
| Kolkata Knight Riders | 68 |
| Kings XI Punjab | 61 |
| Rajasthan Royals | 61 |
| Delhi Daredevils | 56 |
| Sunrisers Hyderabad | 33 |
| Deccan Chargers | 29 |
| Pune Warriors | 12 |
| Gujarat Lions | 9 |
| Kochi Tuskers Kerala | 6 |
| Rising Pune Supergiants | 5 |

2)

Description:

Obtain the list of names of orange cap holders (highest run-scorer in a particular year of the tournament) for each year along with the runs that they have scored.

Query:

```
create view scores_per_year as
  select b1.striker, year(m1.match_date) as year, sum(b1.runs_scored) as score
  from Ball_stats b1, Matches m1
  where m1.match_id = b1.match_id
  group by b1.striker, year;

select p.player_name, sc.year, sc.score
  from Player p, scores_per_year sc,
  (select year, max(score) as mscore
   from
    (select b.striker, year(m.match_date) as year, sum(b.runs_scored) as score
     from Ball_stats b, Matches m
     where m.match_id = b.match_id group by b.striker, year) as T
   group by year) as S
  where S.mscore=sc.score and p.player_id = sc.striker and sc.year = S.year;
```

Result:

| player_name | year | score |
|--------------|------|-------|
| SE Marsh | 2008 | 616 |
| ML Hayden | 2009 | 572 |
| SR Tendulkar | 2010 | 617 |
| CH Gayle | 2011 | 604 |
| CH Gayle | 2012 | 733 |
| MEK Hussey | 2013 | 733 |
| RV Uthappa | 2014 | 660 |
| DA Warner | 2015 | 562 |
| V Kohli | 2016 | 969 |

3)

Description:

Obtain the head-to-head wins in the tournament across the years (i.e, for each pair of teams, what is the number of times each team has won when the match was played between the pair of teams).

Query:

```
create view head_to_head1 as
  select T1.team_name as team_1, T2.team_name as team_2,
    Sum(case when m.result_winning_team_id = T1.team_id then 1 else 0 end) as
team_1_wins,
    Sum(case when m.result_winning_team_id = T2.team_id then 1 else 0 end) as
team_2_wins
  from Team T1, Team T2, Matches m
  where m.played_by_team_id1 = T1.team_id and m.played_by_team_id2 = T2.team_id
 and T1.team_id < T2.team_id
  group by team_1, team_2;
```

```
create view head_to_head2 as
  select T1.team_name as team_1, T2.team_name as team_2,
    Sum(case when m.result_winning_team_id = T1.team_id then 1 else 0 end) as
team_1_wins,
    Sum(case when m.result_winning_team_id = T2.team_id then 1 else 0 end) as
team_2_wins
  from Team T1, Team T2, Matches m
  where m.played_by_team_id1 = T1.team_id and m.played_by_team_id2 = T2.team_id
 and T1.team_id > T2.team_id
  group by team_1, team_2;
```

```
select h1.team_1, h1.team_2, (h1.team_1_wins + h2.team_2_wins) as team_1_wins,
(h2.team_1_wins + h1.team_2_wins) as team_2_wins
  from head_to_head1 h1, head_to_head2 h2
  where h1.team_1 = h2.team_2 and h1.team_2 = h2.team_1
  order by rand();
```

Result:

| team_1 | team_2 | team_1_wins | team_2_wins |
|-----------------------------|-------------------------|-------------|-------------|
| Kolkata Knight Riders | Pune Warriors | 4 | 1 |
| Rajasthan Royals | Pune Warriors | 4 | 1 |
| Rajasthan Royals | Kochi Tuskers Kerala | 1 | 1 |
| Chennai Super Kings | Pune Warriors | 4 | 2 |
| Royal Challengers Bangalore | Delhi Daredevils | 9 | 6 |
| Rajasthan Royals | Sunrisers Hyderabad | 4 | 3 |
| Delhi Daredevils | Mumbai Indians | 9 | 9 |
| Kolkata Knight Riders | Kings XI Punjab | 13 | 6 |
| Kolkata Knight Riders | Sunrisers Hyderabad | 6 | 3 |
| Royal Challengers Bangalore | Deccan Chargers | 5 | 6 |
| Pune Warriors | Sunrisers Hyderabad | 0 | 2 |
| Kolkata Knight Riders | Gujarat Lions | 0 | 2 |
| Chennai Super Kings | Delhi Daredevils | 11 | 5 |
| Royal Challengers Bangalore | Kochi Tuskers Kerala | 2 | 0 |
| Delhi Daredevils | Deccan Chargers | 7 | 4 |
| Rajasthan Royals | Delhi Daredevils | 10 | 6 |
| Kings XI Punjab | Sunrisers Hyderabad | 2 | 6 |
| Sunrisers Hyderabad | Gujarat Lions | 3 | 0 |
| Royal Challengers Bangalore | Mumbai Indians | 8 | 11 |
| Kolkata Knight Riders | Delhi Daredevils | 10 | 7 |
| Rajasthan Royals | Mumbai Indians | 6 | 10 |
| Mumbai Indians | Deccan Chargers | 6 | 4 |
| Kolkata Knight Riders | Mumbai Indians | 5 | 13 |
| Mumbai Indians | Rising Pune Supergiants | 1 | 1 |

4)

Description:

Obtain the names of the top 3 bowlers according to the number of maiden overs (i.e., overs with 0 runs conceded) that they have bowled and the number of maiden overs they bowled.

Query:

```
select p.player_name, S.number_of_maidens
  from Player p,
       (select T.bowler, sum(case when T.runs_conceded=0 then 1 else 0 end) as
number_of_maidens
  from
       (select bowler, over_id, match_id, sum(runs_scored) as runs_conceded
  from Ball_stats
  group by bowler, over_id, match_id) as T
  group by T.bowler
  order by number_of_maidens desc limit 3) as S
 where p.player_id = S.bowler;
```

Result:

| player_name | number_of_maidens |
|-------------|-------------------|
| P Kumar | 17 |
| SL Malinga | 12 |
| IK Pathan | 11 |

5)

Description:

Obtain the name of a player such that he is the only player from his country.

Query:

```
select p2.player_name, T.nationality
  from
    (select p1.nationality, count(p1.player_name) as number_of_people
     from Player p1
    group by p1.nationality) as T, Player p2
 where T.number_of_people = 1 and T.nationality = p2.nationality;
```

Result:

| player_name | nationality |
|------------------|-------------|
| RN ten Doeschate | Netherlands |

6)

Description:

Obtain the names of players who have scored atleast 2000 runs and taken atleast 75 wickets along with the runs scored and the number of wickets taken by them.

Query:

```
create view runs_scored as
  select striker, sum(runs_scored) as score
  from Ball_stats group by striker;
```

```
create view wickets_taken as
  select bowler, count(*) as wickets
  from Ball_stats
```

```
where wicket_type is NOT NULL group by bowler;  
  
select p.player_name, r.score, w.wickets  
      from Player p, runs_scored r, wickets_taken w  
      where r.striker = w.bowler and r.score >= 2000 and w.wickets >= 75 and p.player_id =  
r.striker;
```

Result:

| player_name | score | wickets |
|-------------|-------|---------|
| SR Watson | 2469 | 94 |

Note: There are outputs that contain more than 50 rows. Only the top 10 rows are shown here for the sake of brevity.