Dear Andrew,

It's my great pleasure to help you and your students in tennis. Actually, I sometimes play tennis in my spare time. I am also curious about the questions you mention. I use the data from ATP players from 2018 to 2022 and obtain the following results.

## Basic Information

**1. Will age be a big concern to the tennis player? Please analyze whether the older player will be likely to lose the game as well as the age distribution of the winner and the loser.**
I first use the query to find the frequency of the match that the older wins the younger

```
1  select avg(winner_age > loser_age) as avg_old_win from matches;
```

```
Database changed
mysql> select avg(winner_age > loser_age) as avg_old_win from matches;
+-------------+
| avg_old_win |
+-------------+
|      0.4827 |
+-------------+
1 row in set (0.06 sec)
```

We can find that the frequency of the match where the older wins is roughly closer to 0.48. So, it is hard to say the age will significantly influence the results.
Plus, I also aggregate and get the following distribution

```
1  select w.age, win_cnt, lose_cnt
2  from (select round(winner_age) as age, count(1) as win_cnt from matches group by round(winner_age)) w
3  join (select round(loser_age) as age, count(1) as lose_cnt from matches group by round(loser_age)) l
4  on w.age = l.age order by w.age;
```

```
mysql> select w.age, win_cnt, lose_cnt
    -> from (select round(winner_age) as age, count(1) as win_cnt from matches group by round(winner_age)) w
    -> join (select round(loser_age) as age, count(1) as lose_cnt from matches group by round(loser_age)) l
    -> on w.age = l.age order by w.age;
+------+---------+----------+
| age  | win_cnt | lose_cnt |
+------+---------+----------+
|   17 |       1 |       14 |
|   18 |      83 |       93 |
|   19 |     227 |      192 |
|   20 |     456 |      394 |
|   21 |     503 |      445 |
|   22 |     857 |      766 |
|   23 |     837 |      732 |
|   24 |     934 |      858 |
|   25 |     720 |      662 |
|   26 |     946 |      961 |
|   27 |     640 |      686 |
|   28 |     751 |      905 |
|   29 |     771 |      828 |
|   30 |     780 |      929 |
|   31 |     671 |      725 |
|   32 |     670 |      659 |
|   33 |     501 |      468 |
|   34 |     499 |      448 |
|   35 |     312 |      330 |
|   36 |     217 |      245 |
|   37 |     105 |      115 |
|   38 |      94 |       81 |
|   39 |      17 |       31 |
|   40 |      27 |       42 |
|   41 |       2 |        9 |
|   42 |       2 |        5 |
+------+---------+----------+
26 rows in set (0.08 sec)
```

So, we can find the distribution is almost similar.

**2. How long will each game take on average for different tourney levels and surfaces (i.e., clay, grass, and hard)?**

I use the following query

```sql
select description, surface, avg_time from (
    select surface_id, tourney_level, avg(minutes / (w_game + l_game)) as avg_time from matches m
    join tourney t on m.tourney_id = t.id group by surface_id, tourney_level
) a
join surface s on a.surface_id = s.id
join level l on a.tourney_level = l.id
order by description, surface;
```

```
mysql> select description, surface, avg_time from (
    -> select surface_id, tourney_level, avg(minutes / (w_game + l_game)) as avg_time from matches m
    -> join tourney t on m.tourney_id = t.id group by surface_id, tourney_level
    -> ) a
    -> join surface s on a.surface_id = s.id
    -> join level l on a.tourney_level = l.id
    -> order by description, surface;
+-----------------------------------------+---------+------------+
| description                             | surface | avg_time   |
+-----------------------------------------+---------+------------+
| Davis Cup                               | Clay    | 4.49689684 |
| Davis Cup                               | Grass   | 3.81004444 |
| Davis Cup                               | Hard    | 4.26601541 |
| Grand Slams                             | Clay    | 4.42424901 |
| Grand Slams                             | Grass   | 4.05189425 |
| Grand Slams                             | Hard    | 4.38747697 |
| Masters 1000s                           | Clay    | 4.65641004 |
| Masters 1000s                           | Hard    | 4.41574291 |
| other tour-level events                 | Clay    | 4.60966189 |
| other tour-level events                 | Grass   | 4.09965473 |
| other tour-level events                 | Hard    | 4.33250426 |
| Tour finals and other season-ending events | Hard | 4.18192967 |
+-----------------------------------------+---------+------------+
12 rows in set (0.08 sec)
```

So, it is interesting to find that the average time per game is around 4.3 minutes. The average time in clay is the largest, while that in the grass is the shortest. Perhaps, it is because the bouncing speed of ball in the grass is fast, while the ball speed in the clay is slow. However, it is not clear that the tourney levels will affect the average time.

**3. Analyze the average aces and double faults in each kind of surface**

I use the query and obtain the following

```
1  select  surface, avg(d1.ace + d2.ace) as avg_ace, avg(d1.df + d2.df) as avg_df from matches m
2  join (select match_id, player_id, ace, df from match_details) d1 on m.id = d1.match_id and m.winner_id =
   d1.player_id
3  join (select match_id, player_id, ace, df from match_details) d2 on m.id = d2.match_id and m.loser_id = d2.player_id
4  join tourney t on m.tourney_id = t.id
5  join level l on t.tourney_level = l.id
6  join surface s on t.surface_id = s.id
7  group by surface;
```

```
mysql> select  surface, avg(d1.ace + d2.ace) as avg_ace, avg(d1.df + d2.df) as avg_df from matches m
    -> join (select match_id, player_id, ace, df from match_details) d1 on m.id = d1.match_id and m.winner_id = d1.player_id
    -> join (select match_id, player_id, ace, df from match_details) d2 on m.id = d2.match_id and m.loser_id = d2.player_id
    -> join tourney t on m.tourney_id = t.id
    -> join level l on t.tourney_level = l.id
    -> join surface s on t.surface_id = s.id
    -> group by surface;
+---------+---------+---------+
| surface | avg_ace | avg_df  |
+---------+---------+---------+
| Clay    |  8.3022 | 5.5532  |
| Hard    | 14.2570 | 6.1837  |
| Grass   | 17.2358 | 7.2160  |
+---------+---------+---------+
3 rows in set (0.23 sec)
```

We can clearly find that there are less aces and double faults in clay. So, I may recommend the player good at serving not to play on the clay.

## Tennis Skills

**4. Would you like to find the top 5 tennis players who finish the match faster with a winning rate over 0.8 in best-three-out-of-five matches?**
I use the query and obtain the following

```
select player_id, min(name_first) as firstname, min(name_last) as lastname, avg(minutes) as avg_min,
   avg(IF(m.winner_id = d.player_id, 1, 0)) as avg_win from match_details d
join  matches m on d.match_id = m.id
join players p on d.player_id = p.id
where best_of = 5
group by player_id
having avg_win > 0.8
order by avg_min limit 5;
```

```
mysql> select player_id, min(name_first) as firstname, min(name_last) as lastname, avg(minutes) as avg_min,
    ->     avg(IF(m.winner_id = d.player_id, 1, 0)) as avg_win from match_details d
    -> join  matches m on d.match_id = m.id
    -> join players p on d.player_id = p.id
    -> where best_of = 5
    -> group by player_id
    -> having avg_win > 0.8
    -> order by avg_min limit 5;
+-----------+-----------+----------+----------+---------+
| player_id | firstname | lastname | avg_min  | avg_win |
+-----------+-----------+----------+----------+---------+
|      3819 | Roger     | Federer  | 136.4038 |  0.8462 |
|      4920 | Novak     | Djokovic | 149.3800 |  0.9300 |
|      4742 | Rafael    | Nadal    | 159.4615 |  0.9011 |
|     27285 | Jurabek   | Karimov  | 220.0000 |  1.0000 |
|     45471 | Zsombor   | Piros    | 260.0000 |  1.0000 |
+-----------+-----------+----------+----------+---------+
5 rows in set (0.08 sec)
```

I think the results meet the expectation. We find many famous players like Roger Federer, Novak Djokovic, and Rafael Nadal. Hope this results will help your students.

**5. Would you like to find the top 10 players who save the most break points they have faced and rank top 50 in 2022-09-12? Please order them by the descending order of the ratio of the break points saved to the break points faced, and then by the ascending order of average break points faced and player id.**
I use the query and find that

```
select player_id, min(name_first) as firstname, min(name_last) as lastname,
      avg(bpFaced) as avg_bpFaced, avg(bpSaved/bpFaced) as avg_saveRate from (
   select player_id, bpFaced, bpSaved from match_details d
   join (select id, winner_id from matches) m on d.match_id = m.id
   where player_id in (select player_id from rankings where ranking_date = "2022-09-12" and ranking <= 50)
) as dmr
join players p on dmr.player_id = p.id
group by player_id
order by avg_saveRate desc, avg_bpFaced, player_id
limit 10;
```

```
mysql> select player_id, min(name_first) as firstname, min(name_last) as lastname,
    ->        avg(bpFaced) as avg_bpFaced, avg(bpSaved/bpFaced) as avg_saveRate from (
    -> select player_id, bpFaced, bpSaved from match_details d
    -> join (select id, winner_id from matches) m on d.match_id = m.id
    -> where player_id in (select player_id from rankings where ranking_date = "2022-09-12" and ranking <= 50)
    -> ) as dmr
    -> join players p on dmr.player_id = p.id
    -> group by player_id
    -> order by avg_saveRate desc, avg_bpFaced, player_id
    -> limit 10;
+-----------+-----------+--------------+------------+-------------+
| player_id | firstname | lastname     | avg_bpFaced | avg_saveRate |
+-----------+-----------+--------------+------------+-------------+
|      4544 | John      | Isner        |     3.4688 |   0.69301037 |
|      6387 | Nick      | Kyrgios      |     4.3451 |   0.68227107 |
|     26595 | Matteo    | Berrettini   |     5.2452 |   0.66124439 |
|     24172 | Reilly    | Opelka       |     4.0987 |   0.64280534 |
|     52768 | Jack      | Draper       |     6.5172 |   0.64169630 |
|      4742 | Rafael    | Nadal        |     5.1132 |   0.63827979 |
|      4920 | Novak     | Djokovic     |     4.9264 |   0.63462756 |
|      5793 | Pablo     | Carreno Busta |     6.6737 |   0.63353122 |
|     28019 | Hubert    | Hurkacz      |     5.7398 |   0.63229222 |
|     34853 | Tallon    | Griekspoor   |     7.2941 |   0.62726939 |
+-----------+-----------+--------------+------------+-------------+
10 rows in set, 598 warnings (0.24 sec)
```

**6. Could you find the top 10 players who have the average ace but the lowest double faults in each match? And I hope these players have ranked in the top 100 once after June 2022.**

I use the query and obtain that

```sql
select player_id, min(name_first) as firstname, min(name_last) as lastname,
      avg(ace) as avg_ace, avg(df) as avg_df from match_details d
join (select id, tourney_id from matches) m on m.id = d.match_id
join tourney t on t.id = m.tourney_id
join players p on d.player_id = p.id
where player_id in (
   select distinct player_id from rankings
   where year(ranking_date) = 2022 and month(ranking_date) >= 6 and ranking <= 100
)
group by player_id
order by avg_ace desc, avg_df
limit 10;
```

```
mysql> select player_id, min(name_first) as firstname, min(name_last) as lastname,
    ->        avg(ace) as avg_ace, avg(df) as avg_df from match_details d
    -> join (select id, tourney_id from matches) m on m.id = d.match_id
    -> join tourney t on t.id = m.tourney_id
    -> join players p on d.player_id = p.id
    -> where player_id in (
    -> select distinct player_id from rankings
    -> where year(ranking_date) = 2022 and month(ranking_date) >= 6 and ranking <= 100
    -> )
    -> group by player_id
    -> order by avg_ace desc, avg_df
    -> limit 10;
+-----------+-----------+------------+---------+--------+
| player_id | firstname | lastname   | avg_ace | avg_df |
+-----------+-----------+------------+---------+--------+
|      4544 | John      | Isner      | 21.8698 | 2.3333 |
|     24172 | Reilly    | Opelka     | 18.5263 | 2.6842 |
|      6387 | Nick      | Kyrgios    | 17.4155 | 3.6127 |
|      5016 | Sam       | Querrey    | 15.2500 | 4.1970 |
|     47230 | Maxime    | Cressy     | 14.5789 | 8.4912 |
|     11445 | Quentin   | Halys      | 13.4286 | 7.4762 |
|     22315 | Alexander | Bublik     | 12.9176 | 6.7471 |
|      5220 | Marin     | Cilic      | 10.4978 | 3.7265 |
|      6409 | Thanasi   | Kokkinakis | 10.3000 | 2.9400 |
|     44735 | Lloyd     | Harris     | 10.2250 | 2.2333 |
+-----------+-----------+------------+---------+--------+
10 rows in set (0.24 sec)
```

Interestingly, I find some names that have appeared in the previous question like John Isner, Reilly Opelka, and Nick Kyrgios. There may be some correlation between these statistics.

## Tennis Player

**7. Are there any changes of Novak Djokovic in recent years like the aces, double faults, or the duration of the match? Please separately analyze the game he won and didn’ t win and the type of match (with 3 or 5 sets at maximum).**
I use the following query and find that

```
select year(m.tourney_date) as year, best_of, m.winner_id = md.player_id as is_win,
       count(*) as cnt, avg(ace) as avg_ace, avg(df) as avg_df, avg(minutes) as avg_min from matches m
join (
   select * from match_details
   where player_id in (select id from players where name_first = "Novak" and name_last = "Djokovic")
) md on m.id = md.match_id
group by year, best_of, is_win
order by year, best_of, is_win;
```

```
mysql> select year(m.tourney_date) as year, best_of, m.winner_id = md.player_id as is_win,
    ->        count(*) as cnt, avg(ace) as avg_ace, avg(df) as avg_df, avg(minutes) as avg_min from matches m
    -> join (
    ->     select * from match_details
    ->     where player_id in (select id from players where name_first = "Novak" and name_last = "Djokovic")
    -> ) md on m.id = md.match_id
    -> group by year, best_of, is_win
    -> order by year, best_of, is_win;
+------+---------+--------+-----+---------+--------+----------+
| year | best_of | is_win | cnt | avg_ace | avg_df | avg_min  |
+------+---------+--------+-----+---------+--------+----------+
| 2018 |       3 |      0 |  11 |  5.2727 | 2.4545 | 120.8182 |
| 2018 |       3 |      1 |  35 |  4.8857 | 1.8857 |  97.1429 |
| 2018 |       5 |      0 |   2 |  3.5000 | 5.5000 | 203.5000 |
| 2018 |       5 |      1 |  21 |  6.1905 | 2.7143 | 153.8095 |
| 2019 |       3 |      0 |   8 |  5.3750 | 2.5000 | 124.5000 |
| 2019 |       3 |      1 |  32 |  5.3438 | 1.8438 |  88.1563 |
| 2019 |       5 |      0 |   2 |  4.0000 | 4.5000 | 179.5000 |
| 2019 |       5 |      1 |  22 |  6.6364 | 3.2273 | 128.5909 |
| 2020 |       3 |      0 |   3 |  3.6667 | 2.6667 | 107.6667 |
| 2020 |       3 |      1 |  25 |  5.2000 | 3.0800 | 108.6400 |
| 2020 |       5 |      0 |   2 |  5.0000 | 2.0000 | 110.5000 |
| 2020 |       5 |      1 |  16 |  7.9375 | 3.0000 | 142.3750 |
| 2021 |       3 |      0 |   4 |  6.2500 | 3.7500 | 162.0000 |
| 2021 |       3 |      1 |  24 |  4.8333 | 1.7917 |  99.2500 |
| 2021 |       5 |      0 |   1 |  6.0000 | 3.0000 | 136.0000 |
| 2021 |       5 |      1 |  27 | 11.1111 | 3.2222 | 162.5556 |
| 2022 |       3 |      0 |   4 |  6.2500 | 4.0000 | 163.2500 |
| 2022 |       3 |      1 |  12 |  3.7500 | 1.7500 | 107.5000 |
| 2022 |       5 |      1 |   7 |  9.2857 | 3.4286 | 155.5714 |
+------+---------+--------+-----+---------+--------+----------+
19 rows in set (0.16 sec)
```

**8. Can you find all the matches between Roger Federer and Rafael Nadal between 2018 and 2022?**
I use the following query and obtain that

```
select tourney_date, best_of, (m.winner_id = d1.player_id) as RF_win,
      (m.winner_id = d2.player_id) as RN_win,
      IF(m.winner_id = d1.player_id, w_game, l_game) as RF_game,
      IF(m.winner_id = d2.player_id, w_game, l_game) as RN_game
      from matches m
join (
   select * from match_details
   where player_id in (select id from players where name_first = "Roger" and name_last = "Federer")
```

```
) d1 on d1.match_id = m.id
join (
    select * from match_details
    where player_id in (select id from players where name_first = "Rafael" and name_last = "Nadal")
) d2 on d2.match_id = m.id
order by tourney_date;
```

```
mysql> select tourney_date, best_of, (m.winner_id = d1.player_id) as RF_win,
    ->        (m.winner_id = d2.player_id) as RN_win,
    ->        IF(m.winner_id = d1.player_id, w_game, l_game) as RF_game,
    ->        IF(m.winner_id = d2.player_id, w_game, l_game) as RN_game
    ->        from matches m
    -> join (
    -> select * from match_details
    ->     where player_id in (select id from players where name_first = "Roger" and name_last = "Federer")
    -> ) d1 on d1.match_id = m.id
    -> join (
    -> select * from match_details
    ->     where player_id in (select id from players where name_first = "Rafael" and name_last = "Nadal")
    -> ) d2 on d2.match_id = m.id
    -> order by tourney_date;
+--------------+---------+--------+--------+---------+---------+
| tourney_date | best_of | RF_win | RN_win | RF_game | RN_game |
+--------------+---------+--------+--------+---------+---------+
| 2019-05-27   |       5 |      0 |      1 |       9 |      18 |
| 2019-07-01   |       5 |      1 |      0 |      20 |      19 |
+--------------+---------+--------+--------+---------+---------+
2 rows in set (0.19 sec)
```

There are only two recorded matches in the dataset. Hope this is helpful to you.

Thank you for letting me do this project! I have learned a lot about tennis and SQL from this project. Thank you again for trusting me! Hope those results are useful for you and your students.

Thank you,
Hyfrankl