DDL concerns:

* All dates are formatted YYYYMMDD. Do we import as INTS and then convert to datetime within the database? No need to change anything, the queries work using the current schema.
* Should we use player ids as a proper foreign key in Tourneys table? How do we get this done? Added two more columns, champ\_id and runner\_id, to the csv

**Queries -** 10 SQL queries. At least 4 must be complex ones (draw from multiple tables to find interesting insights about your data).

**Description** **-** A 1-2 sentence description of what each query is supposed to do.

**Credentials - Connect through datagrip using MySQL datasource**

**Host: cis450-tennisproject.crfueay0nikm.us-east-1.rds.amazonaws.com**

**User: cis450Project**

**Password: projectDB1**

**Port: 3306**

A **complex query** uses a combination of elements such as but not limited to: **multiple joins, subqueries or views, aggregations, universal/existential checks**. One or two aggregations or a simple JOIN between two tables wouldn’t suffice. A tip on designing a complex query is that you should start with a complex idea to implement; if you try to over complicate a simple idea just to meet this requirement, you’ll struggle and most importantly, we’ll notice. **We also expect complex queries to have non-trivial runtime (>20s before optimization, >1s after optimization).** Check out this doc to get a sense of what a complex query should look like. Query 1 would be on the border of complex and trivial, but Queries 2, 3, and 4 meet our expectations.

(The year and limit bounds can be substituted by user provided values)

Complex Query #1

From the top 5 largest tournaments, calculate the proportion of champions that are left handed vs. right handed and also the average height of each champion pool (from the five). This offers demographic insight into who typically wins each of the largest tournaments.

WITH top\_five\_tourneys (tourney\_name) AS

(SELECT tourney\_name

FROM Matches

GROUP BY tourney\_name

ORDER BY COUNT(\*) DESC

LIMIT 5

),

t5\_data (tourney\_name, winner\_id) AS

(SELECT M.tourney\_name as tourney\_name, winner\_id

FROM Matches M

INNER JOIN top\_five\_tourneys t5 ON M.tourney\_name = t5.tourney\_name

WHERE round = "F"

),

t5\_players (player\_id, height, hand) AS

(SELECT player\_id, height, hand

FROM Players P

INNER JOIN t5\_data ON P.player\_id = t5\_data.winner\_id

)

SELECT SUM(CASE hand WHEN 'L' THEN 1 ELSE 0 end) \* 1.0 / COUNT(\*) AS l\_prop,

SUM(CASE hand WHEN 'R' THEN 1 ELSE 0 end) \* 1.0 / COUNT(\*) AS r\_prop,

AVG(height) AS avg\_height

FROM t5\_players;

Complex Query #2

Return the top 10 player names, ages, and origin countries for those who have won a tournament match between 2000 and 2020 sorted by their entire career point totals. This finds the best players over the past two decades.

WITH players\_age (name\_first, name\_last, age, ioc, player\_id) AS

(SELECT name\_first, name\_last, YEAR(CURDATE()) - YEAR(dob) AS age, ioc, player\_id

FROM Players)

SELECT DISTINCT name\_first, name\_last, age, ioc

FROM players\_age P

JOIN (

SELECT points, player AS player\_id

FROM Rankings R

JOIN (SELECT YEAR(tourney\_date) AS year, winner\_id

FROM Matches M

WHERE YEAR(tourney\_date) >= 2000 AND YEAR(tourney\_date) <= 2022) W ON R.player = W.winner\_id) RP ON P.player\_id = RP.player\_id

ORDER BY points DESC

LIMIT 10;

Complex Query #3

Return all past grand slam champions and the earnings they have received from each of the four majors. Shows that top earning players are not necessarily the highest rated.

WITH player\_earnings (tourney\_name, name\_first, name\_last, player\_id, earnings) AS

(

SELECT tourney\_name AS t\_name, name\_first, name\_last, player\_id, earnings

FROM Players P

INNER JOIN (SELECT tourney\_name, SUM(first\_prize) AS earnings, champion\_f\_name, champion\_l\_name

FROM Tourneys

GROUP BY champion, tourney\_name)T ON P.name\_first = T.champion\_f\_name AND P.name\_last = T.champion\_l\_name

ORDER BY tourney\_name, earnings DESC

),

temp (player\_rank, tourney\_name, f\_name, l\_name) AS

(

SELECT ranking AS player\_rank,

tourney\_name,

name\_first,

name\_last

FROM player\_earnings pe

JOIN Rankings r

ON pe.player\_id = r.player

GROUP BY name\_first, name\_last

ORDER BY ranking

)

SELECT t.player\_rank, pe.tourney\_name, t.f\_name, t.l\_name, pe.earnings

FROM temp t JOIN player\_earnings pe ON t.f\_name = pe.name\_first AND t.l\_name = pe.name\_last

ORDER BY earnings;

Complex Query #4

Rank the different surfaces by how many points have been scored on them in total.

SELECT surface,

SUM(points) AS total\_points

FROM Players p

JOIN Matches m

ON p.player\_id = m.winner\_id

JOIN Rankings r

ON r.player = p.player\_id

GROUP BY surface

ORDER BY total\_points DESC;

Simple Query #1

Order match surface types by how frequent they are played on (match counts) in descending order.

SELECT surface, COUNT(\*) AS frequency

FROM matches

GROUP BY surface

ORDER BY COUNT(\*) DESC;

Simple Query #2

Rank tournament champions based on how much money they have won over their careers.

SELECT DISTINCT name\_first, name\_last, earnings

FROM players P

INNER JOIN (SELECT tourney\_name, SUM(first\_prize) AS earnings, champion\_f\_name, champion\_l\_name

FROM tourneys

GROUP BY tourney\_name) T ON P.name\_first = T.champion\_f\_name AND P.name\_last = T.champion\_l\_name

ORDER BY earnings DESC

Simple Query #3

Rank countries by the number of tournament champions they have produced.

SELECT ioc AS country, COUNT(\*) AS num\_champs

FROM matches M

JOIN (SELECT player\_id, ioc

FROM players) P ON M.winner\_id = P.player\_id

GROUP BY country

ORDER BY COUNT(\*) DESC

Simple Query #4

Return players record in tournament finals

WITH winners (wins, name\_first, name\_last) AS

(

SELECT COUNT(\*) AS wins, p.name\_first, p.name\_last

FROM Matches m JOIN Players p ON m.winner\_id = p.player\_id

WHERE m.round = "F"

GROUP BY p.name\_first, p.name\_last

),

loser (losses, name\_first, name\_last) AS

(

SELECT COUNT(\*) AS losses, p.name\_first, p.name\_last

FROM Matches m JOIN Players p ON m.loser\_id = p.player\_id

WHERE m.round = "F"

GROUP BY p.name\_first, p.name\_last

)

SELECT w.wins, l.losses, w.name\_first, w.name\_last

FROM winners w JOIN loser l ON w.name\_first = l.name\_first AND w.name\_last = l.name\_last

ORDER BY wins DESC

Simple Query #5

Return player who won the most championships for each grand slam tournament.

WITH max\_wins (wins, champion, tourney\_name) AS

(

SELECT COUNT(champion) AS wins, champion, tourney\_name

FROM Tourneys

GROUP BY champion, tourney\_name

),

t (tourney\_name) AS

(

SELECT DISTINCT tourney\_name FROM Tourneys

)

SELECT \*

FROM (

SELECT t.tourney\_name, mw.champion, Max(mw.wins) as max\_wins

FROM t JOIN max\_wins mw ON t.tourney\_name = mw.tourney\_name

GROUP BY tourney\_name, champion

ORDER BY tourney\_name, Max(wins) DESC

) AS temp

GROUP BY temp.tourney\_name;

Simple Query #6

Return player’s record on clay.

WITH winners (wins, name\_first, name\_last) AS

(

SELECT COUNT(\*) AS wins, p.name\_first, p.name\_last

FROM Matches m JOIN Players p ON m.winner\_id = p.player\_id

WHERE m.surface = "Grass"

GROUP BY p.name\_first, p.name\_last, m.surface

),

loser (losses, name\_first, name\_last) AS

(

SELECT COUNT(\*) AS losses, p.name\_first, p.name\_last

FROM Matches m JOIN Players p ON m.loser\_id = p.player\_id

WHERE m.surface = "Grass"

GROUP BY p.name\_first, p.name\_last, m.surface

)

SELECT w.wins, l.losses, w.name\_first, w.name\_last

FROM winners w JOIN loser l ON w.name\_first = l.name\_first AND w.name\_last = l.name\_last

ORDER BY wins DESC;