1. **List all competitions along with their category nam**e

SELECT c.competition\_id, c.competition\_name, cat.category\_name

FROM competitions c JOIN categories cat ON c.category\_id = cat.category\_id;

1. **Count the number of competitions in each category**

SELECT cat.category\_name, COUNT(c.competition\_id) AS competition\_count

FROM competitions c JOIN categories cat ON c.category\_id = cat.category\_id

GROUP BY cat.category\_name

1. **Find all competitions of type 'doubles'**

SELECT \* FROM competitions WHERE type = ‘doubles’

1. **Get competitions that belong to a specific category (e.g., ITF Men)**

SELECT competition\_id,competition\_name,category\_name FROM competitions com

JOIN categories cat ON cat.category\_id = com.category\_id

WHERE category\_name = 'ITF Men'

1. **Identify parent competitions and their sub-competitions**

SELECT parent.competition\_name AS parent\_competition, child.competition\_name AS sub\_competition

FROM competitions child JOIN competitions parent ON child.parent\_id = parent.competition\_id

ORDER BY parent.competition\_name, child.competition\_name;

1. **Analyze the distribution of competition types by category**

SELECT c.category\_id, cat.category\_name, c.type AS competition\_type, COUNT(\*) AS competition\_count

FROM competitions c JOIN categories cat ON c.category\_id = cat.category\_id

GROUP BY c.category\_id, cat.category\_name, c.type

ORDER BY cat.category\_name, competition\_count DESC**;**

1. **List all competitions with no parent (top-level competitions)**

SELECT \* FROM competitions WHERE parent\_id IS NULL

1. **List all venues along with their associated complex name**

SELECT ven.\*, com.complex\_name

FROM venues ven

JOIN complexes com ON ven.complex\_id = com.complex\_id;

1. **Count the number of venues in each complex**

SELECT complex\_name, COUNT(\*) AS venues\_count

FROM venues ven JOIN complexes com ON com.complex\_id = ven.complex\_id

GROUP BY complex\_name

ORDER BY venues\_count;

1. **Get details of venues in a specific country (e.g., Chile)**

SELECT \* FROM venues WHERE country\_name = 'Chile';

1. **Identify all venues and their time zones**

SELECT DISTINCT venue\_name, timezone FROM venues;

1. **Find complexes that have more than one venue**

SELECT complex\_name, COUNT(\*) AS venues\_count

FROM venues ven JOIN complexes com ON ven.complex\_id = com.complex\_id

GROUP BY complex\_name

HAVING COUNT(\*) > 1;

1. **List venues grouped by country**

SELECT country\_name, STRING\_AGG(venue\_name, ', ') AS venues\_list FROM venues

GROUP BY country\_name;

1. **Find all venues for a specific complex (e.g., Nacional)**

SELECT ven.\* FROM venues ven

JOIN complexes com ON com.complex\_id = ven.complex\_id

WHERE complex\_name = 'Nacional';

1. **Get all competitors with their rank and points.**

SELECT name, rank, points

FROM competitors com

JOIN competitor\_rankings cr ON com.competitor\_id = cr.competitor\_id;

1. **Find competitors ranked in the top 5**

SELECT com.name, cr.rank

FROM competitors com JOIN competitor\_rankings cr ON com.competitor\_id = cr.competitor\_id

ORDER BY rank

LIMIT 5;

1. **List competitors with no rank movement (stable rank)**

SELECT com.name, cr.rank, cr.movement

FROM competitors com

JOIN competitor\_rankings cr ON com.competitor\_id = cr.competitor\_id

WHERE cr.movement = 0;

1. **Get the total points of competitors from a specific country (e.g., Croatia)**

SELECT country, SUM(points) AS total\_points

FROM competitors com

JOIN competitor\_rankings cr ON com.competitor\_id = cr.competitor\_id

WHERE country = 'Croatia' GROUP BY country;

1. **Count the number of competitors per country**

SELECT country, COUNT(competitor\_id) AS no\_of\_competitors

FROM competitors

GROUP BY country;

1. **Find competitors with the highest points in the current week**