# SQL Queries for SportRadar Project

## 1. Competitions & Categories Table Queries

Q1. List all competitions along with their category name

SELECT c.competition\_id, c.competition\_name, cat.category\_name  
FROM Competitions c  
JOIN Categories cat ON c.category\_id = cat.category\_id;

Q2. 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;

Q3. Find all competitions of type 'doubles'

SELECT competition\_id, competition\_name, type  
FROM Competitions  
WHERE type = 'doubles';

Q4. Get competitions that belong to a specific category (e.g., ITF Men)

SELECT c.competition\_id, c.competition\_name  
FROM Competitions c  
JOIN Categories cat ON c.category\_id = cat.category\_id  
WHERE cat.category\_name = 'ITF Men';

Q5. 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;

Q6. Analyze the distribution of competition types by category

SELECT cat.category\_name, c.type, 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, c.type  
ORDER BY cat.category\_name, competition\_count DESC;

Q7. List all competitions with no parent (top-level competitions)

SELECT competition\_id, competition\_name  
FROM Competitions  
WHERE parent\_id IS NULL;

## 2. Complexes & Venues Table Queries

Q1. List all venues along with their associated complex name

SELECT v.venue\_id, v.venue\_name, c.complex\_name  
FROM Venues v  
JOIN Complexes c ON v.complex\_id = c.complex\_id;

Q2. Count the number of venues in each complex

SELECT c.complex\_name, COUNT(v.venue\_id) AS venue\_count  
FROM Venues v  
JOIN Complexes c ON v.complex\_id = c.complex\_id  
GROUP BY c.complex\_name;

Q3. Get details of venues in a specific country (e.g., Chile)

SELECT venue\_id, venue\_name, city\_name, timezone, complex\_id  
FROM Venues  
WHERE country\_name = 'Chile';

Q4. Identify all venues and their timezones

SELECT venue\_id, venue\_name, timezone  
FROM Venues;

Q5. Find complexes that have more than one venue

SELECT c.complex\_name, COUNT(v.venue\_id) AS venue\_count  
FROM Venues v  
JOIN Complexes c ON v.complex\_id = c.complex\_id  
GROUP BY c.complex\_name  
HAVING COUNT(v.venue\_id) > 1;

Q6. List venues grouped by country

SELECT country\_name, city\_name, venue\_name  
FROM Venues  
ORDER BY country\_name, city\_name;

Q7. Find all venues for a specific complex (e.g., Nacional)

SELECT v.venue\_id, v.venue\_name, v.city\_name, v.country\_name, v.timezone  
FROM Venues v  
JOIN Complexes c ON v.complex\_id = c.complex\_id  
WHERE c.complex\_name = 'Nacional';

## 3. Competitors & Competitor\_Rankings Table Queries

Q1. Get all competitors with their rank and points

SELECT c.name, r.rank, r.points  
FROM Competitors c  
JOIN Competitor\_Rankings r ON c.competitor\_id = r.competitor\_id  
ORDER BY r.rank;

Q2. Find competitors ranked in the top 5

SELECT c.name, r.rank, r.points  
FROM Competitors c  
JOIN Competitor\_Rankings r ON c.competitor\_id = r.competitor\_id  
WHERE r.rank <= 5  
ORDER BY r.rank;

Q3. List competitors with no rank movement (stable rank)

SELECT c.name, r.rank, r.points  
FROM Competitors c  
JOIN Competitor\_Rankings r ON c.competitor\_id = r.competitor\_id  
WHERE r.movement = 0;

Q4. Get the total points of competitors from a specific country (e.g., Croatia)

SELECT SUM(r.points) AS total\_points  
FROM Competitors c  
JOIN Competitor\_Rankings r ON c.competitor\_id = r.competitor\_id  
WHERE c.country = 'Croatia';

Q5. Count the number of competitors per country

SELECT c.country, COUNT(c.competitor\_id) AS competitor\_count  
FROM Competitors c  
GROUP BY c.country  
ORDER BY competitor\_count DESC;

Q6. Find competitors with the highest points in the current week

SELECT c.name, r.points  
FROM Competitors c  
JOIN Competitor\_Rankings r ON c.competitor\_id = r.competitor\_id  
ORDER BY r.points DESC  
LIMIT 1;