**Categories and Competitions Table:**

**1.List all competitions along with their category name**

mycursor.execute("""SELECT

c.competition\_id,

c.competition\_name,

cat.category\_name

FROM

competitions c

JOIN

categories cat ON c.category\_id = cat.category\_id

""")

out=mycursor.fetchall()

print(tabulate(out, headers=[i[0] for i in mycursor.description], tablefmt='grid'))**2.Count the number of competitions in each category**

mycursor.execute("""SELECT

c.category\_id,

cat.category\_name,

COUNT(\*) AS competition\_count

FROM

competitions c

JOIN

categories cat ON c.category\_id = cat.category\_id

GROUP BY

cat.category\_name

ORDER BY

competition\_count DESC

""")

out=mycursor.fetchall()

print(tabulate(out, headers=[i[0] for i in mycursor.description], tablefmt='grid'))

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

mycursor.execute("""

SELECT

competition\_id,

competition\_name,

type

FROM

competitions

WHERE

type = 'doubles'

""")

out = mycursor.fetchall()

print(tabulate(out, headers=[i[0] for i in mycursor.description], tablefmt='grid'))

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

mycursor.execute("""

SELECT

c.competition\_id,

c.competition\_name,

cat.category\_name

FROM

competitions c

JOIN

categories cat ON c.category\_id = cat.category\_id

WHERE

cat.category\_name = 'ITF Men'

""")

out = mycursor.fetchall()

print(tabulate(out, headers=[i[0] for i in mycursor.description], tablefmt='grid'))

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

mycursor.execute("""

SELECT

p.competition\_id AS parent\_id,

p.competition\_name AS parent\_name,

c.competition\_id AS sub\_id,

c.competition\_name AS sub\_name,

c.type,

c.gender,

c.category\_id

FROM competitions AS p

JOIN competitions AS c

ON p.competition\_id = c.parent\_id

ORDER BY p.competition\_name, c.competition\_name;

""")

out = mycursor.fetchall()

print(tabulate(out, headers=[i[0] for i in mycursor.description], tablefmt='grid'))

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

mycursor.execute("""

SELECT

cat.category\_name,

c.type,

COUNT(\*) AS competition\_count

FROM

categories cat

JOIN

competitions c ON c.category\_id = cat.category\_id

GROUP BY

c.type, cat.category\_name

ORDER BY

cat.category\_name, competition\_count DESC

""")

out = mycursor.fetchall()

print(tabulate(out, headers=[i[0] for i in mycursor.description], tablefmt='grid'))

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

mycursor.execute("""

SELECT

competition\_id,

competition\_name,

type,

gender,

category\_id

FROM

competitions

WHERE

parent\_id is NULL

""")

out = mycursor.fetchall()

print(tabulate(out, headers=[i[0] for i in mycursor.description], tablefmt='grid'))

**Complexes and venues tables:**

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

mycursor.execute("""

SELECT

c.complex\_name,

v.venue\_name

FROM

complexes c

JOIN

venues v ON c.complex\_id = v.complex\_id

""")

out = mycursor.fetchall()

print(tabulate(out, headers=[i[0] for i in mycursor.description], tablefmt='grid'))

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

mycursor.execute("""

SELECT

c.complex\_id,

c.complex\_name,

COUNT(v.venue\_id) AS venue\_count

FROM

Complexes c

JOIN

Venues v ON c.complex\_id = v.complex\_id

GROUP BY

c.complex\_name; # c.complex\_id,

""")

out = mycursor.fetchall()

print(tabulate(out, headers=[i[0] for i in mycursor.description], tablefmt='grid'))

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

mycursor.execute("SELECT \* FROM venues where country\_name = 'chile'")

out = mycursor.fetchall()

print(tabulate(out, headers=[i[0] for i in mycursor.description], tablefmt='grid'))

**4.Identify all venues and their timezones**

mycursor.execute("SELECT venue\_id, venue\_name, timezone FROM Venues")

out = mycursor.fetchall()

print(tabulate(out, headers=[i[0] for i in mycursor.description], tablefmt='grid'))

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

mycursor.execute("""

SELECT

c.complex\_id,

c.complex\_name,

COUNT(v.venue\_id) AS venue\_count

FROM

Complexes c

JOIN

Venues v ON c.complex\_id = v.complex\_id

GROUP BY

c.complex\_name

HAVING

venue\_count > 1;

""")

print(tabulate(out, headers=[i[0] for i in mycursor.description], tablefmt='grid'))

**6.List venues grouped by country**

mycursor.execute("SELECT venue\_id, venue\_name,country\_name FROM Venues Group by country\_name")

out = mycursor.fetchall()

print(tabulate(out, headers=[i[0] for i in mycursor.description], tablefmt='grid'))

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

mycursor.execute("""

SELECT

c.complex\_name,

v.venue\_name

FROM

Complexes c

JOIN

Venues v ON c.complex\_id = v.complex\_id

WHERE

c.complex\_name = 'Nacional'

""")

out = mycursor.fetchall()

print(tabulate(out, headers=[i[0] for i in mycursor.description], tablefmt='grid'))

**Competitor\_Rankings and Competitors Table:**

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

mycursor.execute("SELECT rank,points,competitor\_id FROM competitor\_rankings")

out = mycursor.fetchall()

print(tabulate(out, headers=[i[0] for i in mycursor.description], tablefmt='grid'))

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

mycursor.execute("""

SELECT

cr.rank,

cr.competitor\_id,

cr.points

FROM

competitor\_rankings cr

JOIN

competitors c ON cr.competitor\_id = c.competitor\_id

WHERE

cr.rank<=5

order by cr.rank

""")

out = mycursor.fetchall()

print(tabulate(out, headers=[i[0] for i in mycursor.description], tablefmt='grid'))

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

mycursor.execute("""

SELECT

cr.rank,

c.name,

cr.competitor\_id,

cr.movement,

cr.points

FROM

competitor\_rankings cr

JOIN

competitors c ON cr.competitor\_id = c.competitor\_id

WHERE

cr.movement = 0

""")

out = mycursor.fetchall()

print(tabulate(out, headers=[i[0] for i in mycursor.description], tablefmt='grid'))

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

mycursor.execute("""

SELECT

c.country,

SUM(r.points) AS total\_points

FROM

competitors c

JOIN

competitor\_rankings r ON c.competitor\_id = r.competitor\_id

WHERE

c.country = 'Croatia'

""")

out = mycursor.fetchall()

print(tabulate(out, headers=[i[0] for i in mycursor.description], tablefmt='grid'))

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

mycursor.execute("""

SELECT

country,

COUNT(\*) AS number\_of\_competitors

FROM

competitors

GROUP BY

country

""")

out = mycursor.fetchall()

print(tabulate(out, headers=[i[0] for i in mycursor.description], tablefmt='grid'))

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

mycursor.execute("""

SELECT

cr.rank,

cr.points,

c.name,

cr.competitor\_id,

c.country

FROM

competitor\_rankings cr

JOIN

competitors c ON cr.competitor\_id = c.competitor\_id

WHERE

cr.points = (SELECT MAX(points) FROM competitor\_rankings)

""")

out = mycursor.fetchall()

print(tabulate(out, headers=[i[0] for i in mycursor.description], tablefmt='grid'))