**1st table**

import requests

import json

url="https://api.sportradar.com/tennis/trial/v3/en/competitions.json?api\_key=ka3pataQsk83k2MwnyEAOm12jzJFnSSm8bhHKsp9"

headers={"accept":"application/json"}

response=requests.get(url,headers=headers)

print(response.text)

data=json.loads(response.text)

competitions=data["competitions"]

import pandas as pd

competitions\_list=[]

for comp in competitions:

competitions\_list.append({

"id":comp["id"],

"name":comp["name"],

"parent\_id": comp.get("parent\_id"),

"type":comp["type"],

"gender":comp["gender"],

"category\_id":comp["category"]["id"]

})

category\_list=[]

for comp in competitions:

category\_list.append({

"category\_id":comp["category"]["id"],

"category\_name":comp["category"]["name"]

})

category\_df = pd.DataFrame(category\_list)

print(category\_df)

category\_df=category\_df.drop\_duplicates()

Category\_df

competition\_df = pd.DataFrame(competitions\_list)

competition\_df

mycursor.execute("CREATE TABLE Competitions (competition\_id VARCHAR(50) PRIMARY KEY, competition\_name VARCHAR(100) NOT NULL,parent\_id VARCHAR(50),type VARCHAR(20) NOT NULL,gender VARCHAR(10) NOT NULL,category\_id VARCHAR(50) NOT NULL,FOREIGN KEY (category\_id)REFERENCES Categories(category\_id))")

data\_list=category\_df.values.tolist()

query="""

INSERT INTO categories(category\_id,category\_name) VALUES (%s,%s);

"""

mycursor.executemany(query,data\_list)

mydb.commit()

print("data inserted")

data1\_list=competition\_df.values.tolist()

query="""

INSERT INTO competitions(competition\_id, competition\_name,parent\_id,type,gender,category\_id) VALUES (%s,%s,%s,%s,%s,%s);

"""

mycursor.executemany(query,data1\_list)

mydb.commit()

print("data inserted")

**QUERIES**

1. mycursor.execute("""SELECT competitions.competition\_name, categories.category\_name from categories

INNER JOIN competitions ON categories.category\_id=competitions.category\_id ;""")

out=mycursor.fetchall()

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

1. mycursor.execute("""SELECT categories.category\_name, COUNT(competitions.competition\_id) from categories INNER JOIN competitions ON categories.category\_id=competitions.category\_id GROUP BY categories.category\_name ;""")

out=mycursor.fetchall()

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

1. mycursor.execute("""SELECT competition\_name, type, gender FROM competitions WHERE type = 'doubles';""")

out=mycursor.fetchall()

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

1. mycursor.execute("""SELECT competitions.competition\_name, categories.category\_name FROM competitions

INNER JOIN categories ON competitions.category\_id = categories.category\_id

WHERE categories.category\_name = 'ITF Men';""")

out=mycursor.fetchall()

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

1. mycursor.execute("""SELECT p.competition\_name AS parent\_competition,

c.competition\_name AS sub\_competition

FROM competitions AS p

JOIN competitions AS c ON p.competition\_id = c.parent\_id;""")

out=mycursor.fetchall()

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

1. mycursor.execute("""SELECT

categories.category\_name,

competitions.type,

COUNT(\*) AS Count

FROM competitions

JOIN categories ON competitions.category\_id = categories.category\_id

GROUP BY categories.category\_name, competitions.type

ORDER BY categories.category\_name, count DESC;""")

out=mycursor.fetchall()

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

1. mycursor.execute("""SELECT \*

FROM competitions

WHERE parent\_id IS NULL;""")

out=mycursor.fetchall()

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

**2nd table**

data=json.loads(response.text)

complexes=data["complexes"]

import pandas as pd

complexes\_table = []

for complex in data.get("complexes", []):

complexes\_table.append({

"complex\_id": complex.get("id"),

"complex\_name": complex.get("name")

})

complex\_df = pd.DataFrame(complexes\_table)

print(complex\_df)

venues\_table = []

for comp in complexes:

complex\_id = comp.get("id")

complex\_name = comp.get("name")

for venue\_info in comp.get("venues", []):

venues\_table.append({

"venue\_id": venue\_info.get("id"),

"venue\_name": venue\_info.get("name"),

"city\_name": venue\_info.get("city\_name"),

"country\_name": venue\_info.get("country\_name"),

"country\_code": venue\_info.get("country\_code"),

"time\_zone": venue\_info.get("timezone"),

"complex\_id": complex\_id

})

venue\_df = pd.DataFrame(venues\_table)

print(venue\_df)

mycursor.execute("USE game\_analytics")

mycursor.execute("CREATE TABLE Complexes (complex\_id VARCHAR(50) PRIMARY KEY, complex\_name VARCHAR(100) NOT NULL)")

mycursor.execute("CREATE TABLE Venues (venue\_id VARCHAR(50) PRIMARY KEY, venue\_name VARCHAR(100) NOT NULL, city\_name VARCHAR(100) NOT NULL, country\_name VARCHAR(100) NOT NULL, country\_code CHAR(3) NOT NULL,time\_zone VARCHAR(100) NOT NULL, complex\_id VARCHAR(50) NOT NULL,FOREIGN KEY (complex\_id)REFERENCES Complexes(complex\_id))")

data\_list=complex\_df.values.tolist()

query="""

INSERT INTO complexes(complex\_id,complex\_name) VALUES (%s,%s);

"""

mycursor.executemany(query,data\_list)

mydb.commit()

print("data inserted")

data1\_list=venue\_df.values.tolist()

query="""

INSERT INTO Venues(venue\_id,venue\_name,city\_name,country\_name,country\_code,time\_zone,complex\_id) VALUES (%s,%s,%s,%s,%s,%s,%s);

"""

mycursor.executemany(query,data1\_list)

mydb.commit()

print("data inserted")

from tabulate import tabulate

**QUERIES**

1. mycursor.execute("""SELECT venues.venue\_name,complexes.complex\_name FROM venues INNER JOIN complexes ON complexes.complex\_id=venues.complex\_id ;""")

out=mycursor.fetchall()

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

1. mycursor.execute("""SELECT complexes.complex\_name , COUNT(venues.venue\_id) FROM venues INNER JOIN complexes ON complexes.complex\_id=venues.complex\_id GROUP BY complexes.complex\_id;""")

out=mycursor.fetchall()

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

1. mycursor.execute("""SELECT \* from venues WHERE country\_name= 'chile';""")

out=mycursor.fetchall()

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

1. mycursor.execute("""SELECT venue\_name,time\_zone from venues;""")

out=mycursor.fetchall()

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

1. mycursor.execute(""" SELECT complex\_name, COUNT(venue\_id)

FROM venues

JOIN complexes ON venues.complex\_id = complexes.complex\_id

GROUP BY complexes.complex\_id

HAVING COUNT(venue\_id) > 1;""")

out=mycursor.fetchall()

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

1. mycursor.execute("""

SELECT

country\_name,

COUNT(\*) AS venue\_count

FROM venues

GROUP BY country\_name

ORDER BY venue\_count DESC;

""")

out = mycursor.fetchall()

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

1. mycursor.execute("""SELECT venues.venue\_name FROM venues JOIN complexes ON complexes.complex\_id=venues.complex\_id WHERE complexes.complex\_name='nacional';""")

out=mycursor.fetchall()

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

**3rd table**

import requests

url = "https://api.sportradar.com/tennis/trial/v3/en/double\_competitors\_rankings.json?api\_key=ka3pataQsk83k2MwnyEAOm12jzJFnSSm8bhHKsp9"

headers = {"accept": "application/json"}

response = requests.get(url, headers=headers)

print(response.text)

data=json.loads(response.text)

rankings = data["rankings"]

import pandas as pd

data = response.json()

rankings = data["rankings"]

all\_ranks = []

for group in rankings:

for r in group["competitor\_rankings"]:

all\_ranks.append({

"rank": r["rank"],

"movement": r["movement"],

"points": r["points"],

"competitions\_played": r["competitions\_played"],

"competitor\_id": r["competitor"]["id"]

})

for idx, row in enumerate(all\_ranks):

row["rank\_id"] = idx + 1

competitor\_rankings\_df = pd.DataFrame(all\_ranks)[["rank\_id", "rank", "movement", "points", "competitions\_played", "competitor\_id"]]

print(competitor\_rankings\_df)

competitors = []

for item in rankings:

for c in item["competitor\_rankings"]:

comp = c["competitor"]

competitors.append({

"competitor\_id": comp.get("id"),

"name": comp.get("name"),

"country": comp.get("country"),

"country\_code": comp.get("country\_code"),

"abbreviation": comp.get("abbreviation")

})

competitors\_df = pd.DataFrame(competitors)

print(competitors\_df.head())

mycursor.execute("USE game\_analytics")

mycursor.execute("CREATE TABLE Competitors(competitor\_id VARCHAR(50) PRIMARY KEY, name VARCHAR(10) NOT NULL, country VARCHAR(10) not null, country\_code CHAR(3) not null, abbreviations VARCHAR(10) not null)")

mycursor.execute("CREATE TABLE Competitor\_Rankings(rank\_id INT PRIMARY KEY, rank INT NOT NULL, movement INT not null, points INT not null, competitions\_played INT not null, competitor\_id VARCHAR(50), FOREIGN KEY (competitor\_id) References competitors(competitor\_id))")

data\_list1=competitors\_df.values.tolist()

query="""

INSERT INTO Competitors(competitor\_id,name,country,country\_code,abbreviations) VALUES (%s,%s,%s,%s,%s);

"""

mycursor.executemany(query,data\_list1)

mydb.commit()

print("data inserted")

data\_list=competitor\_rankings\_df.values.tolist()

query="""

INSERT INTO competitor\_rankings(rank\_id,rank,movement,points,competitions\_played,competitor\_id) VALUES (%s,%s,%s,%s,%s,%s);

"""

mycursor.executemany(query,data\_list)

mydb.commit()

print("data inserted")

from tabulate import tabulate

**QUERIES**

1. mycursor.execute("SELECT competitors.name, competitor\_rankings.rank, competitor\_rankings.points FROM competitors JOIN competitor\_rankings WHERE competitors.competitor\_id=competitor\_rankings.competitor\_id;")

out=mycursor.fetchall()

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

1. mycursor.execute("SELECT competitors.name, competitor\_rankings.rank FROM competitors JOIN competitor\_rankings ON competitors.competitor\_id=competitor\_rankings.competitor\_id WHERE competitor\_rankings.rank BETWEEN 1 AND 5 ORDER BY competitor\_rankings.rank ASC ;")

out=mycursor.fetchall()

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

1. mycursor.execute("SELECT competitors.name, competitor\_rankings.movement FROM competitors JOIN competitor\_rankings ON competitors.competitor\_id=competitor\_rankings.competitor\_id WHERE competitor\_rankings.movement=0 ;")

out=mycursor.fetchall()

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

1. mycursor.execute("SELECT competitors.country, SUM(competitor\_rankings.points) AS total\_points FROM competitors JOIN competitor\_rankings ON competitors.competitor\_id=competitor\_rankings.competitor\_id WHERE competitors.country='Croatia' GROUP BY competitors.country;")

out=mycursor.fetchall()

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

1. mycursor.execute("""

SELECT competitors.country, COUNT(competitors.competitor\_id) FROM competitors

GROUP BY competitors.country

;

""")

out = mycursor.fetchall()

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

1. mycursor.execute("SELECT competitors.name, competitor\_rankings.points FROM competitors JOIN competitor\_rankings ON competitors.competitor\_id=competitor\_rankings.competitor\_id WHERE competitor\_rankings.points = (SELECT MAX(points) FROM competitor\_rankings);")

out=mycursor.fetchall()

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

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***