İleri Cypher Sorgular ve Python-Neo4j Entegrasyonu

Open Flight Veri Tabanı İleri Cypher Sorguları

1. Havalimanının sadece giden mi, gelen mi, yoksa her iki yönde de uçuşu olduğunu sınıflandır.

```
MATCH (a:Airport)
OPTIONAL MATCH (:Flight)-[:DESTINATION]→(a)
WITH a, COUNT(*) AS InCount
OPTIONAL MATCH (:Flight)-[:SOURCE]→(a)
WITH a.name AS Airport, InCount, COUNT(*) AS OutCount
RETURN Airport,
  CASE
    WHEN OutCount > 0 AND InCount = 0 THEN "Only Departures"
    WHEN InCount > 0 AND OutCount = 0 THEN "Only Arrivals"
    WHEN OutCount > 0 AND InCount > 0 THEN "Both"
    ELSE "Unused"
  END AS UsageType
```

	Airport	UsageType
1	"Goroka Airport"	"Both"
2	"Madang Airport"	"Both"
3	"Mount Hagen Kagamuga Airport"	"Both"
4	"Nadzab Airport"	"Both"
5	"Port Moresby Jacksons International Airport"	"Both"

2. Her ülkenin kalkış yaptığı en yoğun havalimanını bul.

```
MATCH (a:Airport)-[:LOCATED_IN]→(c:Country)
CALL {
  WITH a
  MATCH (:Flight)-[:SOURCE]→(a)
  RETURN a.name AS AirportName, COUNT(*) AS OutFlights
}
WITH c.name AS Country, AirportName, OutFlights
ORDER BY Country, OutFlights DESC
WITH Country, COLLECT({airport: AirportName, count: OutFlights})[0] AS TopAirport
RETURN Country, TopAirport.airport AS BusiestAirport, TopAirport.count AS Flights
ORDER BY Flights DESC
LIMIT 10
```

	Country	BusiestAirport	Flights
1	"United States"	"Hartsfield Jackson Atlanta International Airport"	3660
2	*null*	"Hartsfield Jackson Atlanta International Airport"	3660
3	"China"	"Beijing Capital International Airport"	2124
4	"United Kingdom"	"London Heathrow Airport"	2100
5	"France"	"Charles de Gaulle International Airport"	2096

3. Her havayolunun iç hat mı dış hat mı uçtuğunu belirle.

```
MATCH (al:Airline)-[:OPERATES]→(f:Flight)
MATCH (f)-[:SOURCE]→(src:Airport)-[:LOCATED_IN]→(c1:Country)
MATCH (f)-[:DESTINATION]→(dst:Airport)-[:LOCATED_IN]→(c2:Country)
WITH al.name AS Airline,
     COLLECT(DISTINCT CASE WHEN c1.name = c2.name THEN "Domestic" ELSE "International" END) AS Types
RETURN Airline,
  CASE
    WHEN "Domestic" IN Types AND "International" IN Types THEN "Both"
    WHEN "Domestic" IN Types THEN "Domestic Only"
    WHEN "International" IN Types THEN "International Only"
    ELSE "Unknown"
  END AS FlightType
```

	Airline	FlightType
1	"40-Mile Air"	"Both"
2	"ABSA - Aerolinhas Brasileiras"	"Both"
3	"ALAK"	"Both"
4	"Abaet"	"Both"
5	"Abu Dhabi Amiri Flight"	"Both"
6	"Adria Airways"	"International Only"

4. Her havalimanı kaç farklı ülkeye uçuş gerçekleştiriyor?

```
MATCH (f:Flight)-[:SOURCE]→(a:Airport)
MATCH (f)-[:DESTINATION]→(dest:Airport)-[:LOCATED_IN]→(destCountry:Country)
RETURN a.name AS Airport, COUNT(DISTINCT destCountry.name) AS CountryCount
ORDER BY CountryCount DESC
LIMIT 10
```

	Airport	CountryCount
1	"Charles de Gaulle International Airport"	102
2	"Frankfurt am Main Airport"	92
3	"Atatürk International Airport"	89
4	"Dubai International Airport"	84
5	"Amsterdam Airport Schiphol"	78

5. Türk Havayolun'un en sık kullandığı 3 güzergahı bul.

```
MATCH (al:Airline {name: "Turkish Airlines"})-[:OPERATES]→(f:Flight)
MATCH (f)-[:SOURCE]→(src:Airport)
MATCH (f)-[:DESTINATION]→(dst:Airport)
WITH src.name AS From, dst.name AS To, COUNT(*) AS Frequency
RETURN From, To, Frequency
ORDER BY Frequency DESC
LIMIT 3
```

	From	To	Frequency
1	"Aalborg Airport"	"Atatürk International Airport"	64
2	"Port Bouet Airport"	"Atatürk International Airport"	64
3	"Aalborg Airport"	"Billund Airport"	64

6. Seçilen başkentlerin havalimanlarından kalkış var mı?

```
UNWIND ["Ankara", "Berlin", "Paris", "London", "Madrid"] AS CapitalCity
OPTIONAL MATCH (a:Airport {city: CapitalCity})<-[:SOURCE]-(f:Flight)
RETURN CapitalCity, COUNT(f) AS OutboundFlights,
       CASE WHEN COUNT(f) > 0 THEN "Has Flights" ELSE "No Flights" END AS Status
```

	CapitalCity	OutboundFlights	Status
1	"Ankara"	0	"No Flights"
2	"Berlin"	0	"No Flights"
3	"Paris"	0	"No Flights"
4	"London"	0	"No Flights"
5	"Madrid"	0	"No Flights"

Python ile Neo4j Veritabanı Entegrasyonu

1. PyCharm ile çalışma dosyası oluşturulmuştur.

2. .env dosyasına gerekli bilgiler eklenmiştir.

```
.env
OPENAI_API_KEY=keyBuraya
NEO4J_URI=bolt://localhost:7687
NEO4J_USERNAME=neo4j
NEO4J_PASSWORD=sifreBuraya
```

3. Neo4j sürücüsü yüklenmiştir.

```
(.venv) PS D:\python_projeleri\Neo4j_setup> pip install neo4j
Collecting neo4j
  Downloading neo4j-5.28.1-py3-none-any.whl.metadata (5.9 kB)
Collecting pytz (from neo4j)
  Downloading pytz-2025.2-py2.py3-none-any.whl.metadata (22 kB)
Downloading neo4j-5.28.1-py3-none-any.whl (312 kB)
Downloading pytz-2025.2-py2.py3-none-any.whl (509 kB)
Installing collected packages: pytz, neo4j
Successfully installed neo4j-5.28.1 pytz-2025.2
```

4. Bağlantı testi yapılmıştır.

```
.env    test.py
from dotenv import load_dotenv
import os
from neo4j import GraphDatabase

load_dotenv()
uri = os.getenv("NEO4J_URI")
username = os.getenv("NEO4J_USERNAME")
password = os.getenv("NEO4J_PASSWORD")

driver = GraphDatabase.driver(
    uri,
    auth=(username, password))
1 usage
def test_connection():
    records, summary, keys = driver.execute_query(
        "MATCH (n:Airline) RETURN n LIMIT 25"
    )
    print(records[0])
test_connection()

Run    test
D:\python_projeleri\demo\.venv\Scripts\python.exe D:\python_projeleri\demo\test.py
<Record n=<Node element_id='4:35b0f201-c0c1-4c6c-8e47-fa1636f273c9:15655' labels=frozenset({'Airline'}) properties={'country': '\\N', 'IATA': '-', 'name': 'Unknown', 'callsign'
```