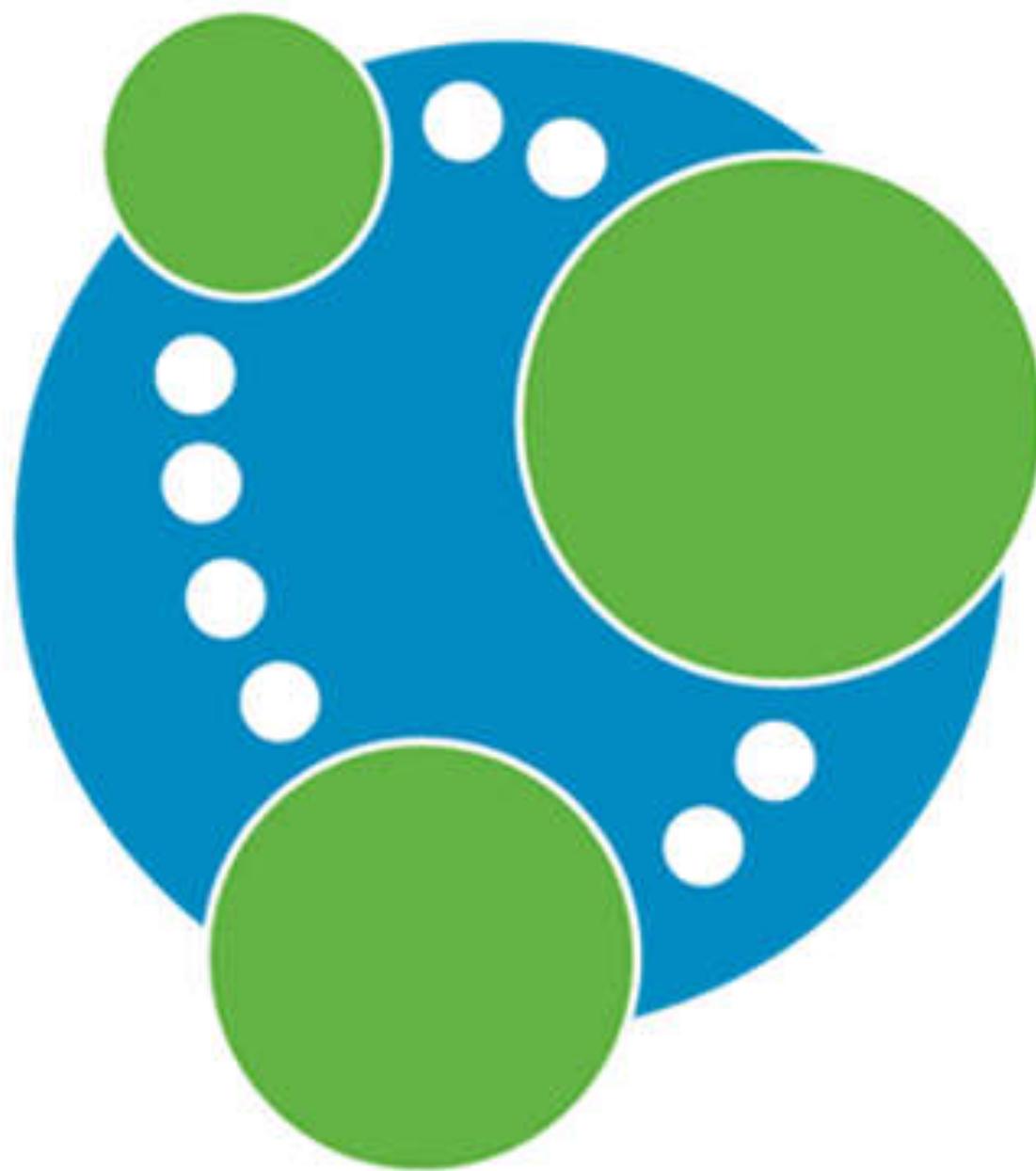


**NEO4J**

*neo4j*

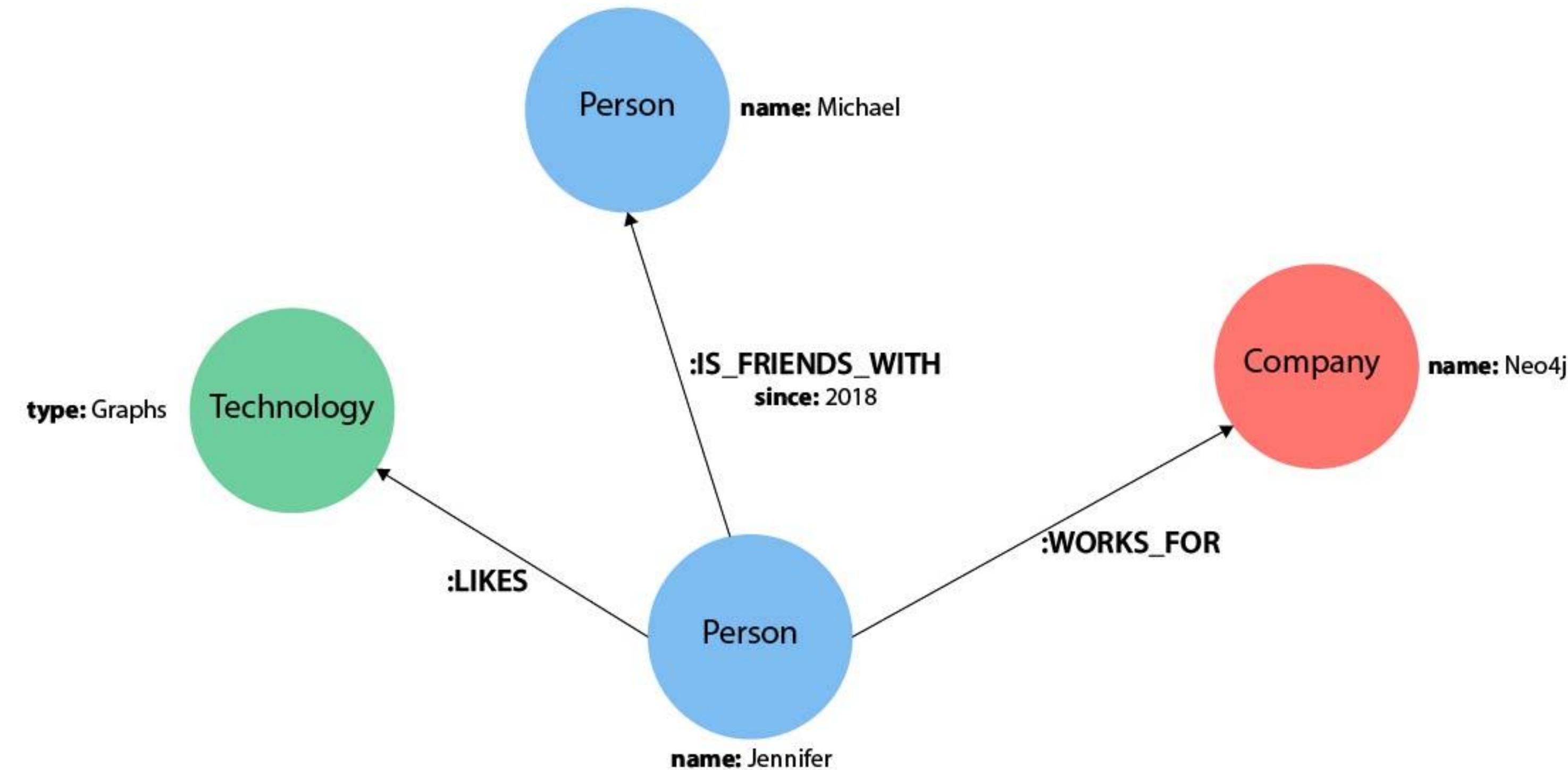


# Graf Tabanlı Veritabanları

Graf tabanlı veritabanlarında veriler düğümler (node),  
ilişkiler (edge) ve özellikler (properties) şeklinde tutulurlar.

Diğer veritabanı türlerinden farklı olarak veriler arasındaki  
ilişkiler de saklanabilir.

# Verilerin Düğüm Üzerinde Tutulması



# ilişkisel Veritabanı Kavramlarına Karşılık Graf Tabanlı Veritabanı Kavramları



# NEO4J NEDİR



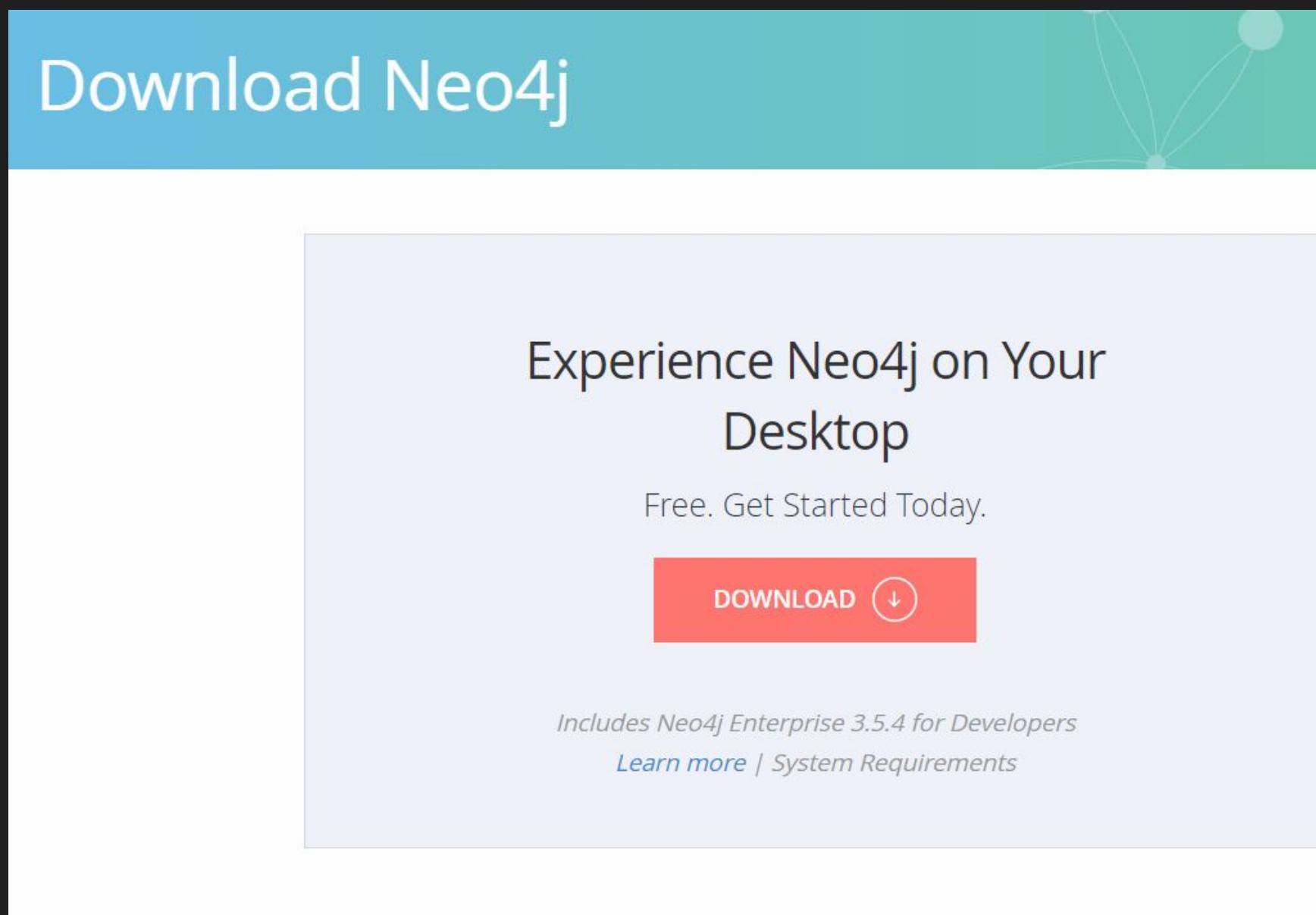
- ❖ Neo4j adının sonundaki ‘j’ harfinden çıkarım yapabileceğimiz üzere Java ile yazılmıştır.
- ❖ Neo4j’deki veri yapısı temel iki kavram üzerine oturtulmuştur; düğümler ve bağlantılar.
- ❖ Düğümler verileri, bağlantılar ise düğümler arasındaki ilişkileri gösterir.
- ❖ Herhangi bir düğümün veya bağlantının birden fazla özelliği olabilir.

# NEO4J KURULUMU

- ❖ Kurulum Linki : <https://neo4j.com/download/>
- ❖ Neo4j Community : <https://community.neo4j.com/>

# NEO4J KURULUMU (DEVAM)

## ADIM #1



## ADIM #2

The screenshot shows the 'Get Started Now' form. It starts with the heading 'Get Started Now' and a sub-instruction 'Please fill out this form to begin your download'. The form contains several input fields marked with red asterisks: 'First Name', 'Last Name', 'Business Email', 'Company Name', 'Phone Number', and a dropdown menu for 'Country' set to 'Turkey'. There's also a checkbox for opting in to receive news updates. At the bottom, there's a note about agreeing to the license agreement, a green 'Download Desktop' button, and a small privacy policy disclaimer.

Get Started Now  
Please fill out this form to begin your download

\* First Name  
\* Last Name  
\* Business Email  
\* Company Name  
\* Phone Number  
\* Turkey

Yes! Please keep me updated on the latest news about graph database products, services & events. I can unsubscribe at any time.

By downloading you agree to the [Neo4j License Agreement for Neo4j Desktop Software](#).

[Download Desktop](#)

The information you provide will be used in accordance with the terms of our [privacy policy](#).

# NEO4J TEMEL SORGULAR

.01

**CREATE(n)**

Tek düğüm oluşturma

.02

**CREATE(n),(m)**

Birden fazla düğüm oluşturma

.03

**CREATE(n:insan{ad:“Yakup”})**

İçerisinde Yakup adını bulunduran n adında bir insan düğümü oluşturuldu

.05

**MATCH(n) Return n**

Tüm düğüm ve ilişkileri yazdırma

.06

**MATCH(n) DETACH**

**DELETE n**

Tüm düğüm ve ilişkileri silme

.04

**MATCH(n:insan{name:Yakup}) SET n.name='Mustafa'**

**RETURN n**

İsmi ‘Yakup’ olan düğümleri getirdi ve bu düğümlerin isimlerini ‘Mustafa’ olarak güncelledi.

# iLişki KURMA

Yonetmen grafında Quentin Tarantino adlı düğüm  
ile Filmler grafındaki Pulp Fiction adlı düğüm ile  
ilişki kurma

```
MATCH (n : Yonetmen { name : " Quentin Tarantino "}), (b : Filmler { name : "Pulp Fiction" })  
MERGE (n)-[:YONETMEN]->(b)
```

```
MATCH (n : Yonetmen { name : " Quentin Tarantino "}), (b : Filmler { name : "Pulp Fiction" })  
CREATE (n)-[:YONETMEN]->(b)
```

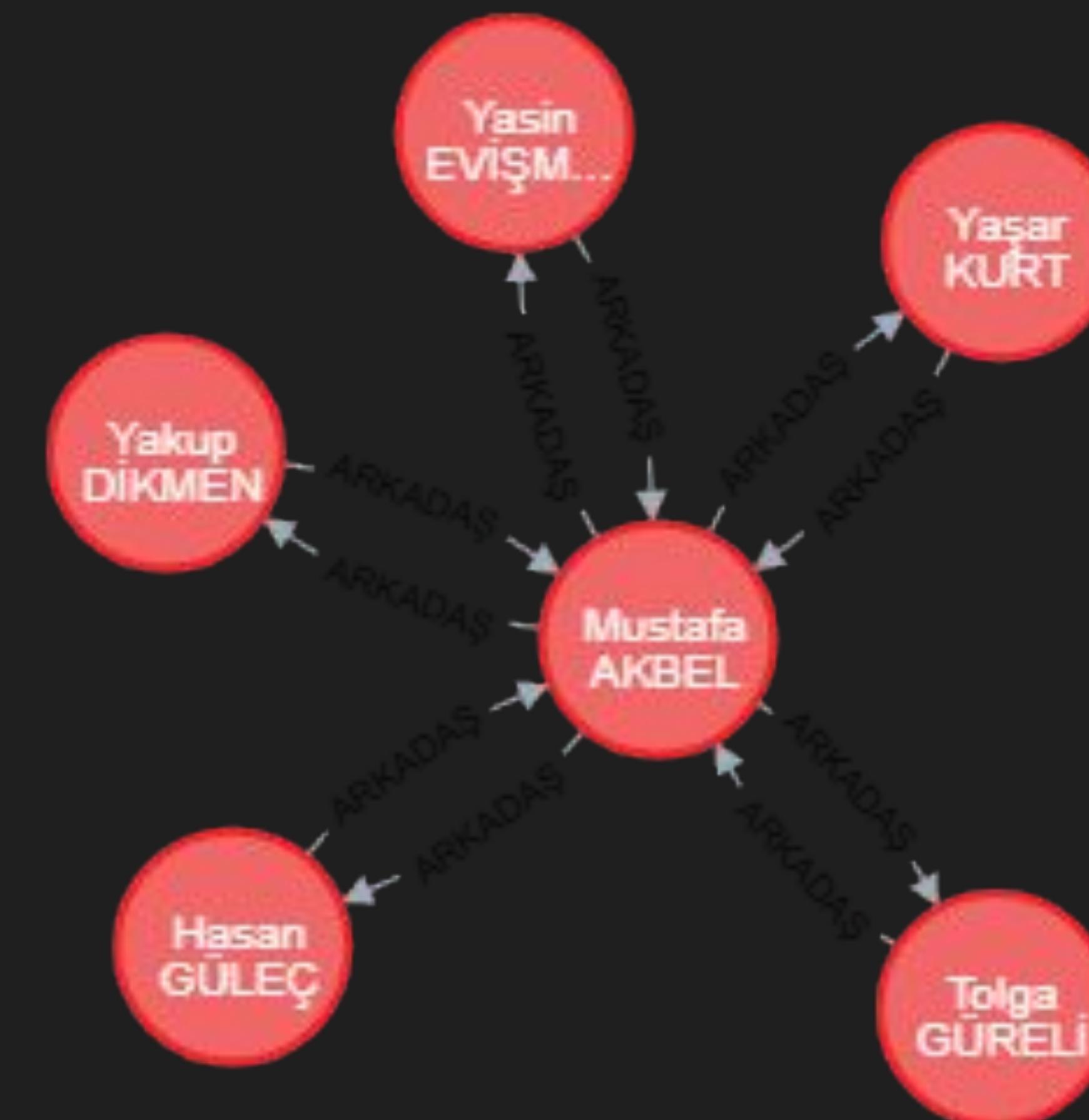


# iLişki SORGUS U

İsmi Mustafa AKBEL olan düğümün  
ARKADAŞ ilişkisini gösteren sorgu

```
MATCH (n : Kisiler {name : "Mustafa AKBEL"})-[r : ARKADAŞ]->(c : Kisiler) RETURN n ,
```

c



# WHERE KOMUT U

Filmler grafındaki tüm düğümlerdeki yıl  
özellikine bakılarak 2000 yılından önce  
yapılan düğümler gösterilmiştir.

```
MATCH (n : Filmler) WHERE n.yıl<2000 RETURN n
```

```
{
  "name": "The Shawshank Redemption",
  "yıl": 1994,
  "imdb": "9,2"
}
```

```
{
  "name": "The Godfather",
  "yıl": 1972,
  "imdb": "9,2"
}
```

# ORDER BY KOMUT

Filmler grafındaki tüm düğümlerin yıl  
özellikine bakarak küçükten büyüğe  
sıralanması yapılır.

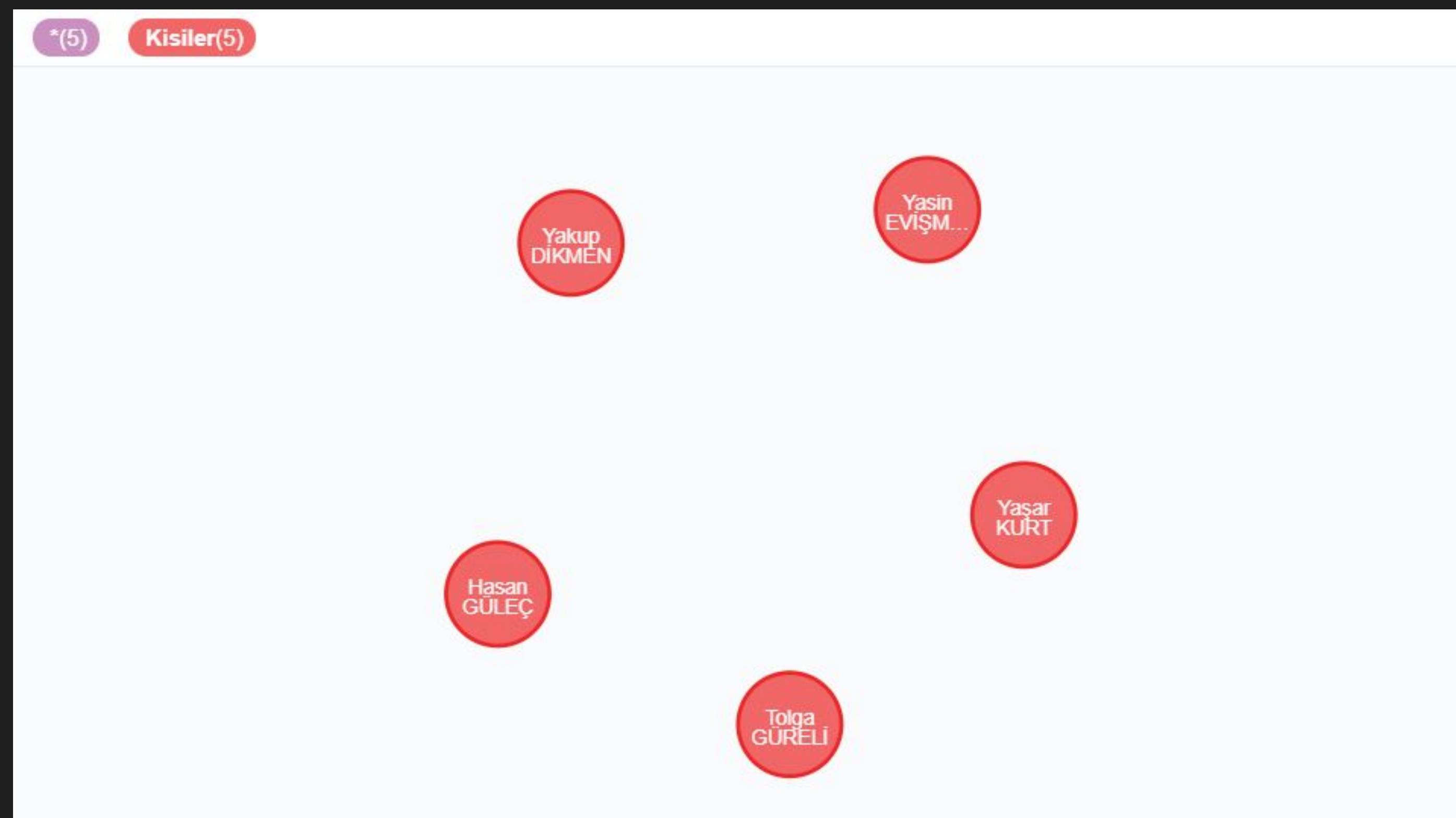
MATCH (n : Filmler) RETURN n.name, n.yil ORDER BY n.yil

n.name	n.yil
"12 Angry Men "	1957
"The Godfather"	1972
"The Godfather: Part II"	1974
"Schindler's List"	1993
"The Shawshank Redemption"	1994
"Pulp Fiction"	1994
"Forrest Gump"	1994
"Fight Club "	1999
"The Lord of the Rings: The Return of the King"	2003
"The Dark Knight"	2008

# COUNT KOMUT U

Mustafa AKBEL'in arkadaş sayısını bulan sorgu.

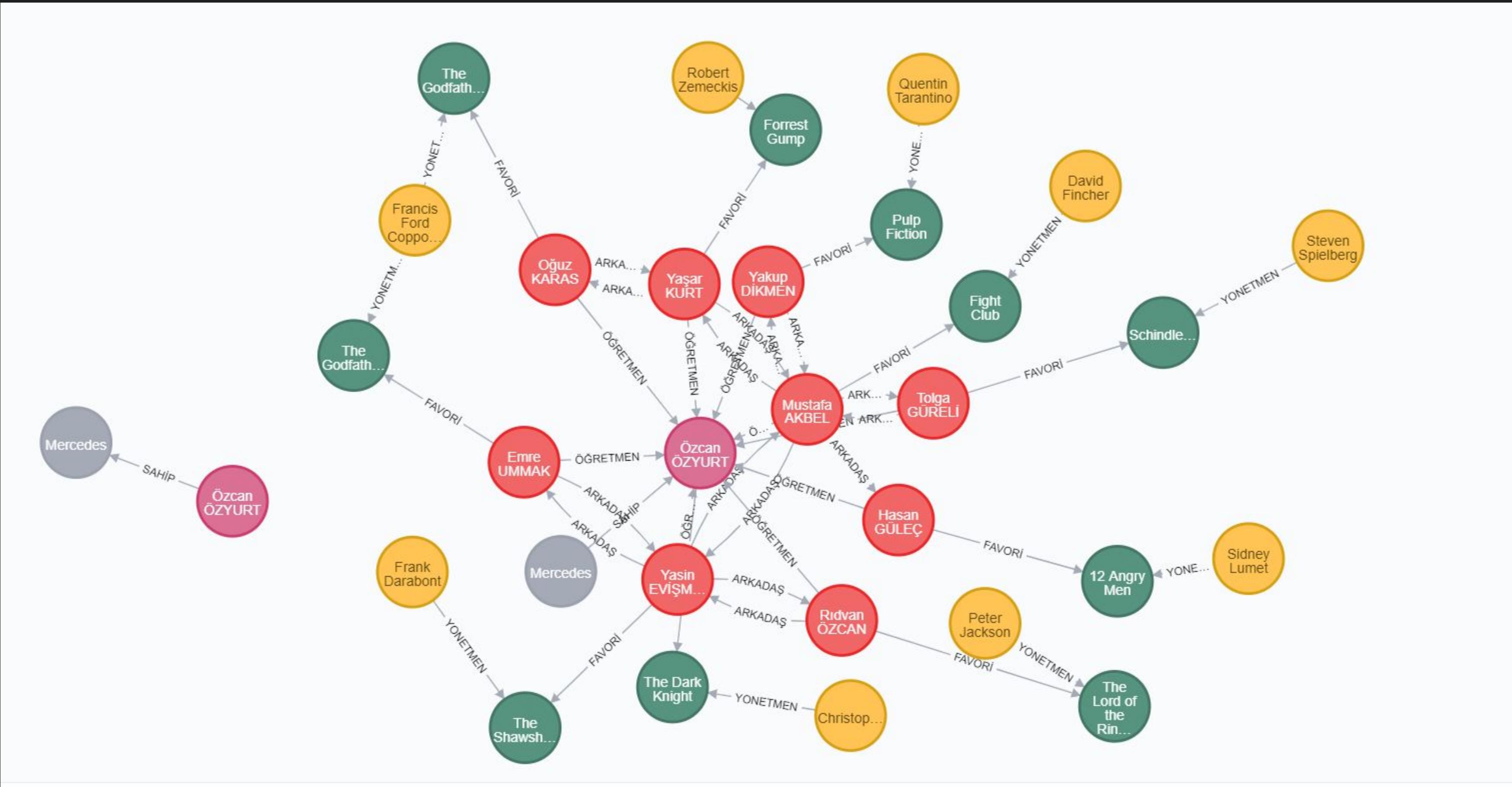
```
MATCH (n:Kisiler{name:"Mustafa AKBEL"})-[:ARKADAŞ]->(c:Kisiler) RETURN  
c,COUNT(*) as ARKADASSAYISI
```



# ÇEŞİTLİ SORGU ÖRNEKLERİ

Tüm Database ve ilişkiler

MATCH (n) WHERE n

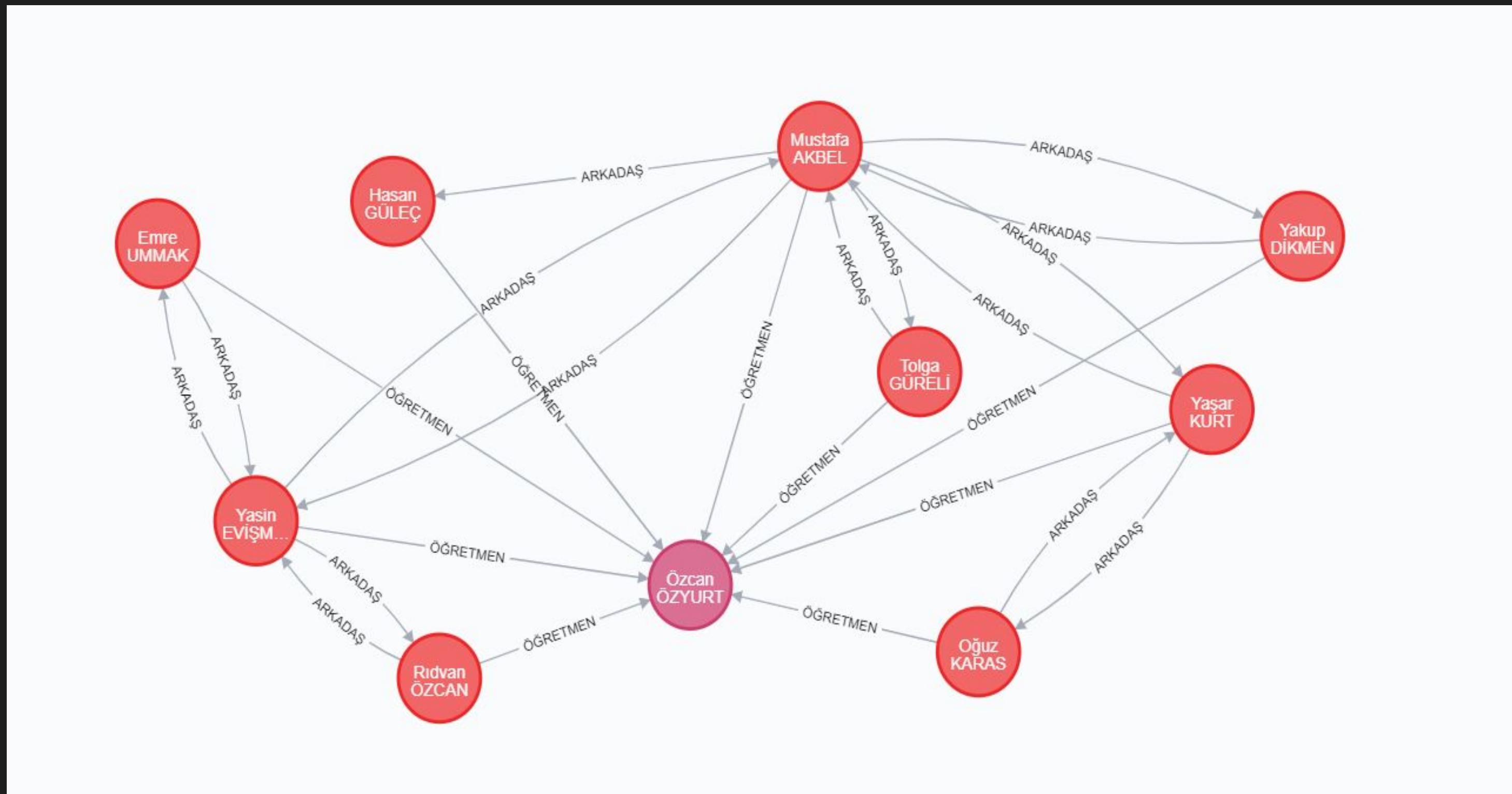


# ÇEŞİTLİ SORGU ÖRNEKLERİ

Kişilerin Özcan ÖZYURT ile olan  
öğretmen ilişkisini sorgulama.

```
MATCH (c:Kişiler) [r:ÖĞRETMEN] -> (n:Öğretmenler{name:"Özcan ÖZYURT"}) RETURN
```

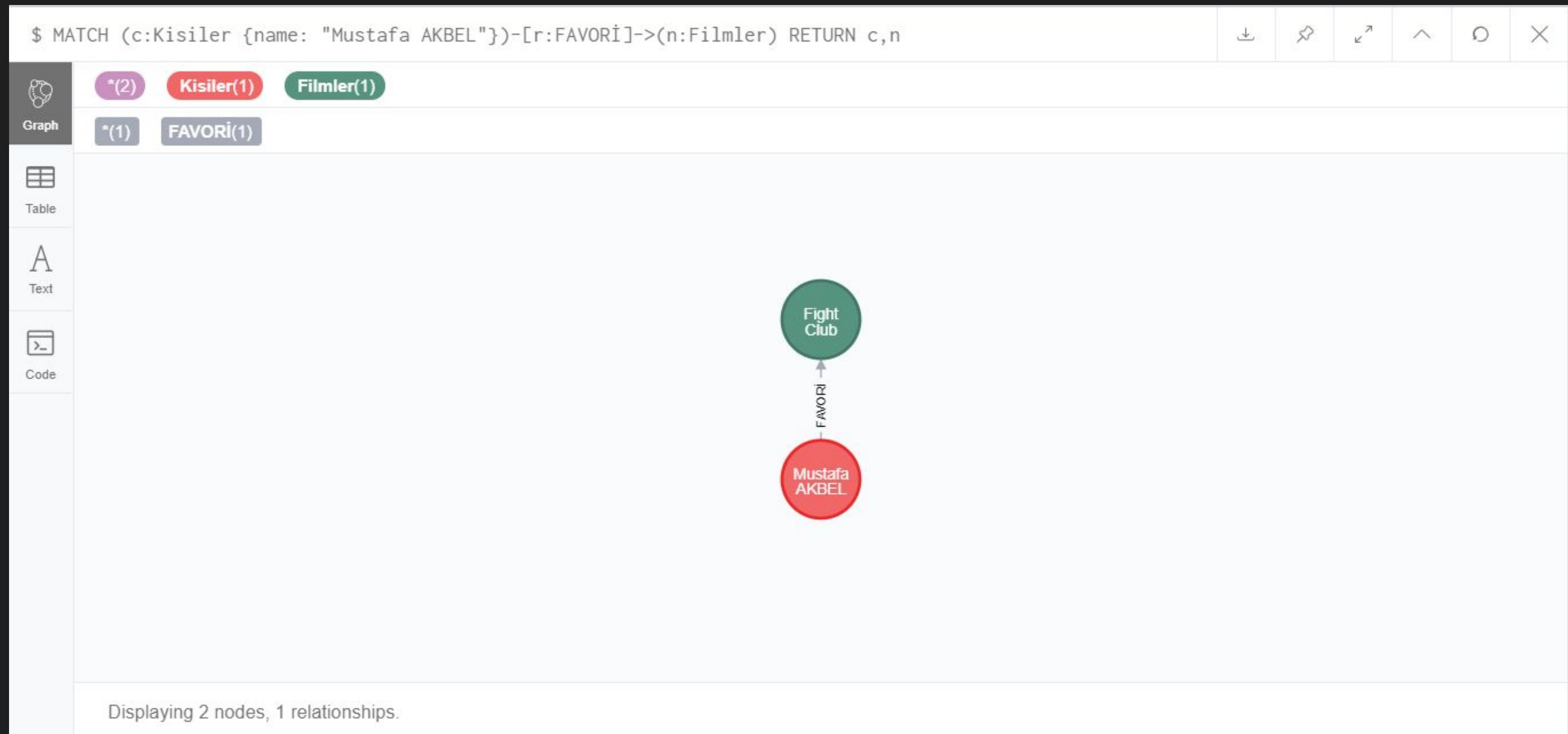
c,n



# ÇEŞİTLİ SORGU ÖRNEKLERİ

Mustafa AKBEL kişisinin favori  
filmlerini getirir.

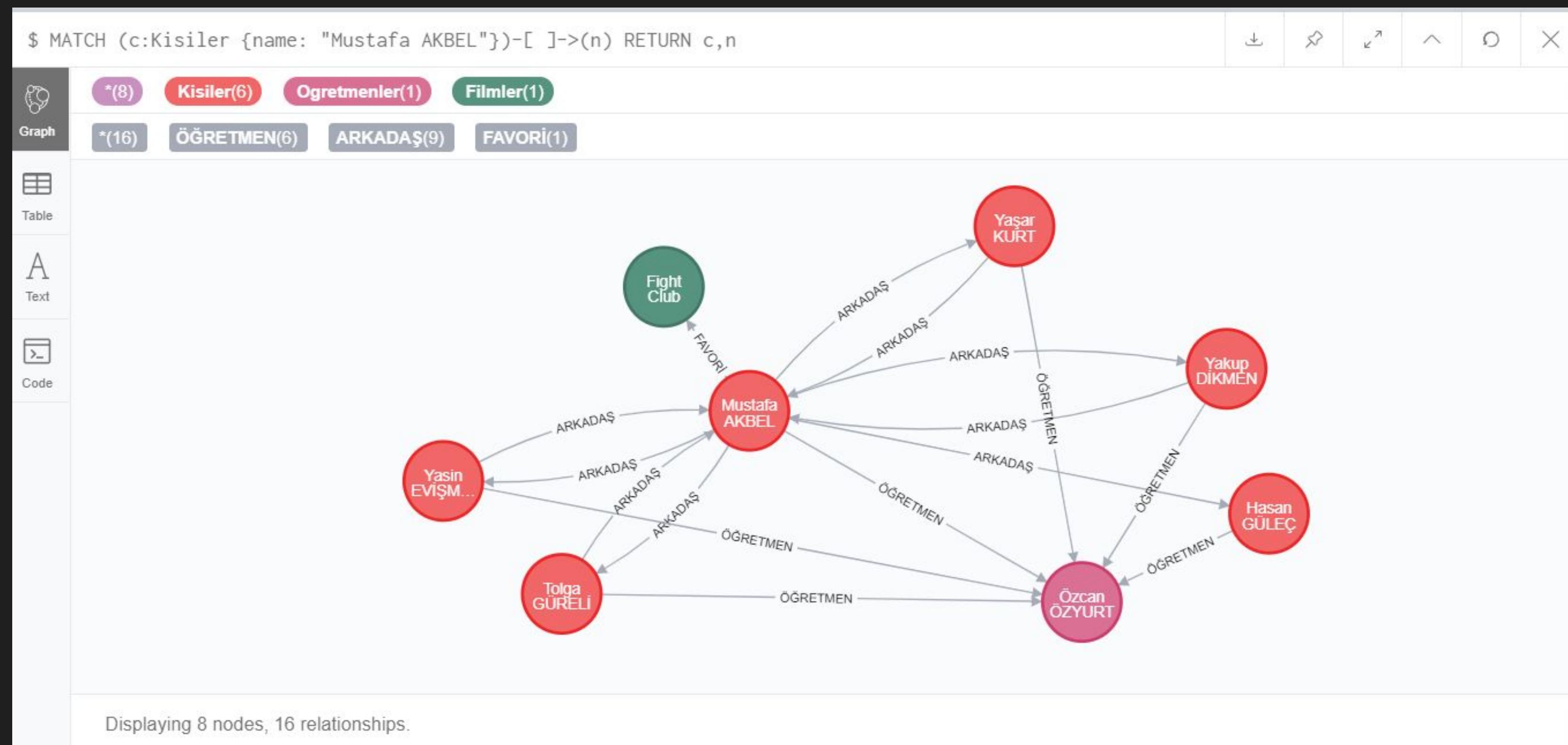
MATCH (c:Kisiler {name: "Mustafa AKBEL"})-[r:FAVORİ]->(n:Filmler) RETURN c,n



# ÇEŞİTLİ SORGU ÖRNEKLERİ

Mustafa AKBEL kişisinin tüm ilişkileri

\$ MATCH (c:Kisiler {name: "Mustafa AKBEL"})-[ ]->(n) RETURN c,n

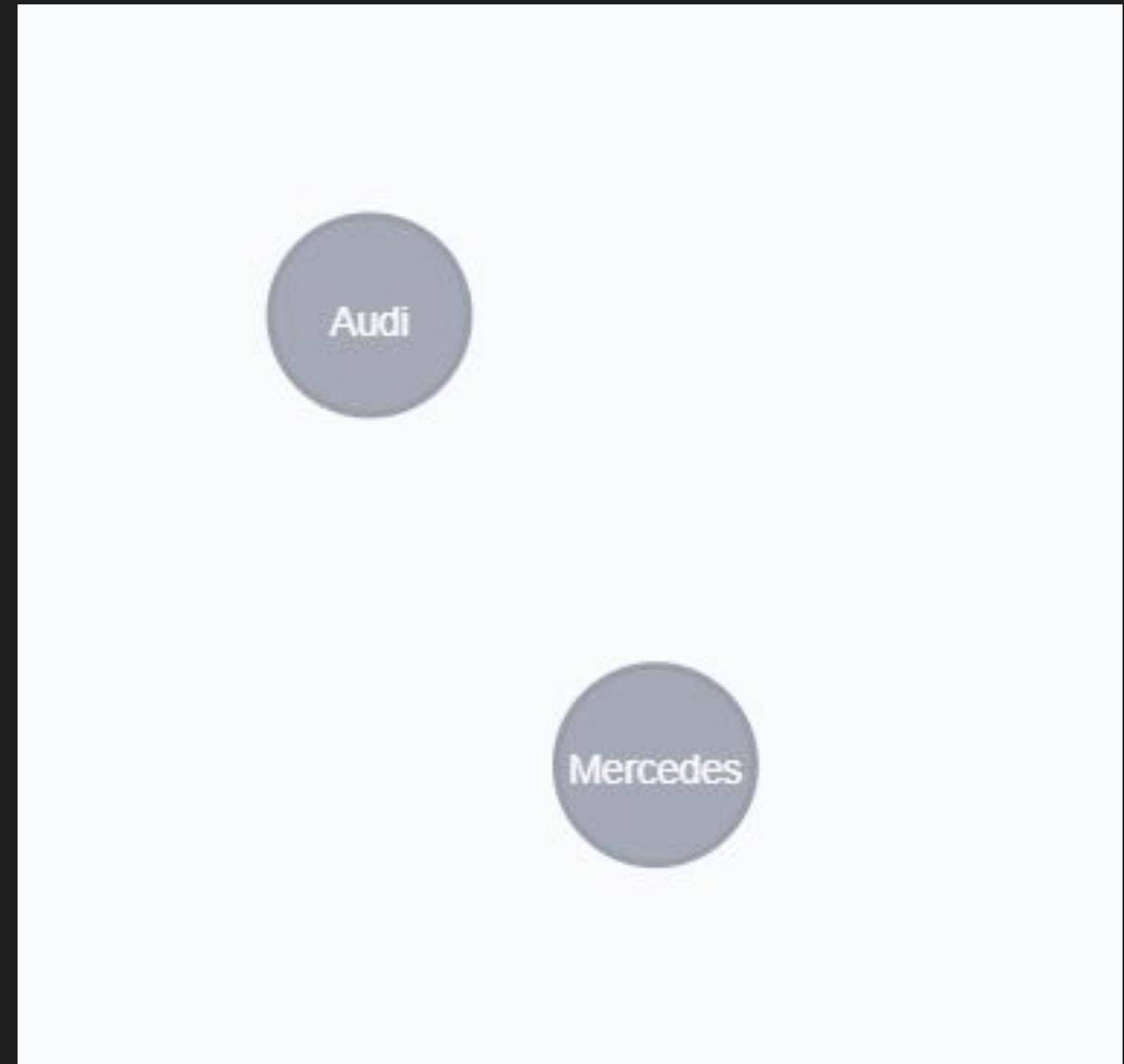


# ÇEŞİTLİ SORGU ÖRNEKLERİ

Araba grafına audi düğümünün eklenmesi

CREATE (n:Araba{name:"Audi"})

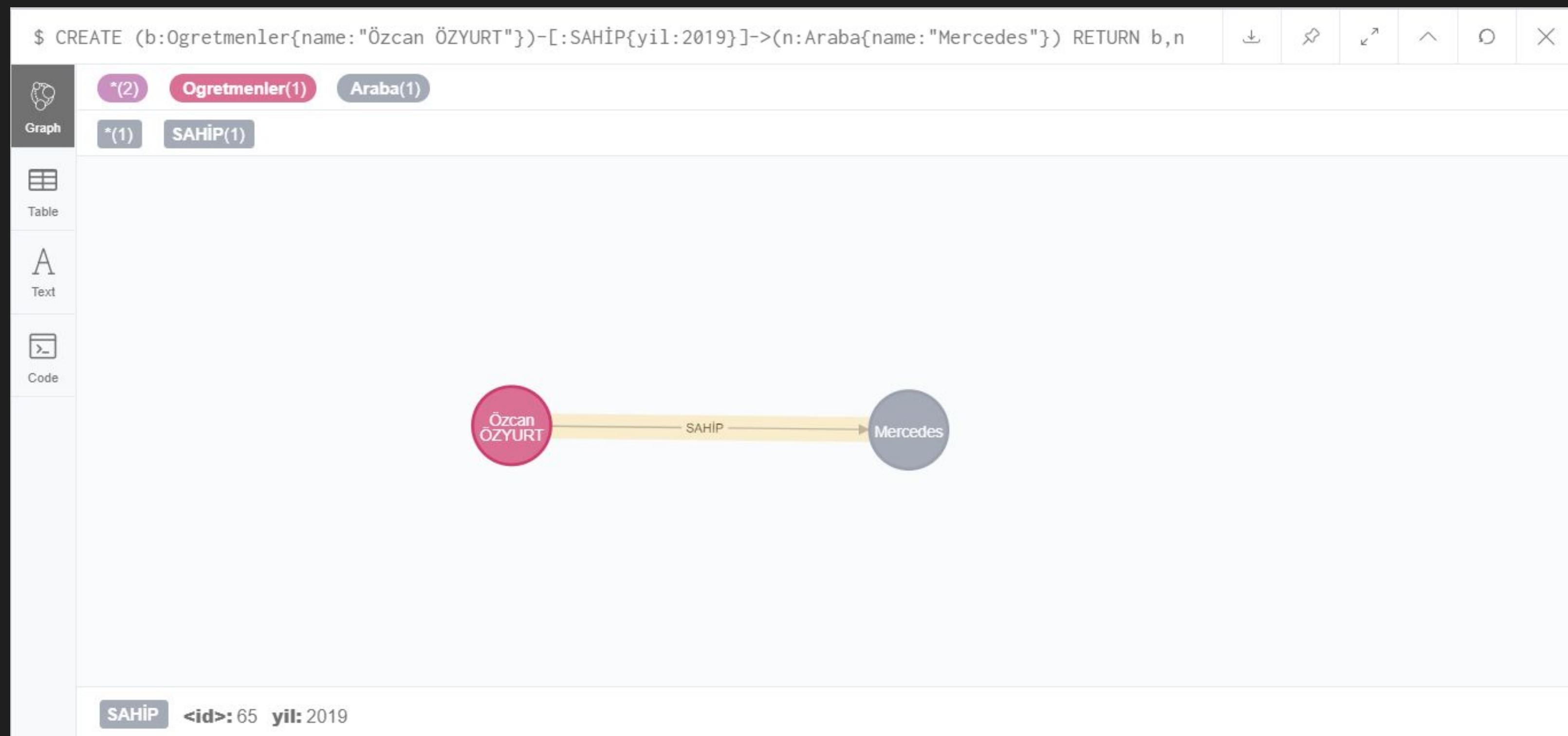
RETURN n



# ÇEŞİTLİ SORGU ÖRNEKLERİ

Özcan ÖZYURT düğümü ile Araba grafındaki  
Mercedes düğümü arasında detaylı ilişki  
kurma

CREATE (:Ogretmenler{name:'Özcan ÖZYURT"})-[:SAHİP{yıl:2019}]->(n:Araba{name:'Mercedes'}) RETURN b,n



# ÇEŞİTLİ SORGU ÖRNEKLERİ

Mustafa AKBEL ile Yaşar KURT

arasındaki arkadaşlık ilişkisinin silinmesi

```
$ MATCH (n:Kisiler{name:'Mustafa AKBEL'})-[r:ARKADAŞ]->(b:Kisiler{name:"Yaşar KURT"})
```

```
KURT")
```

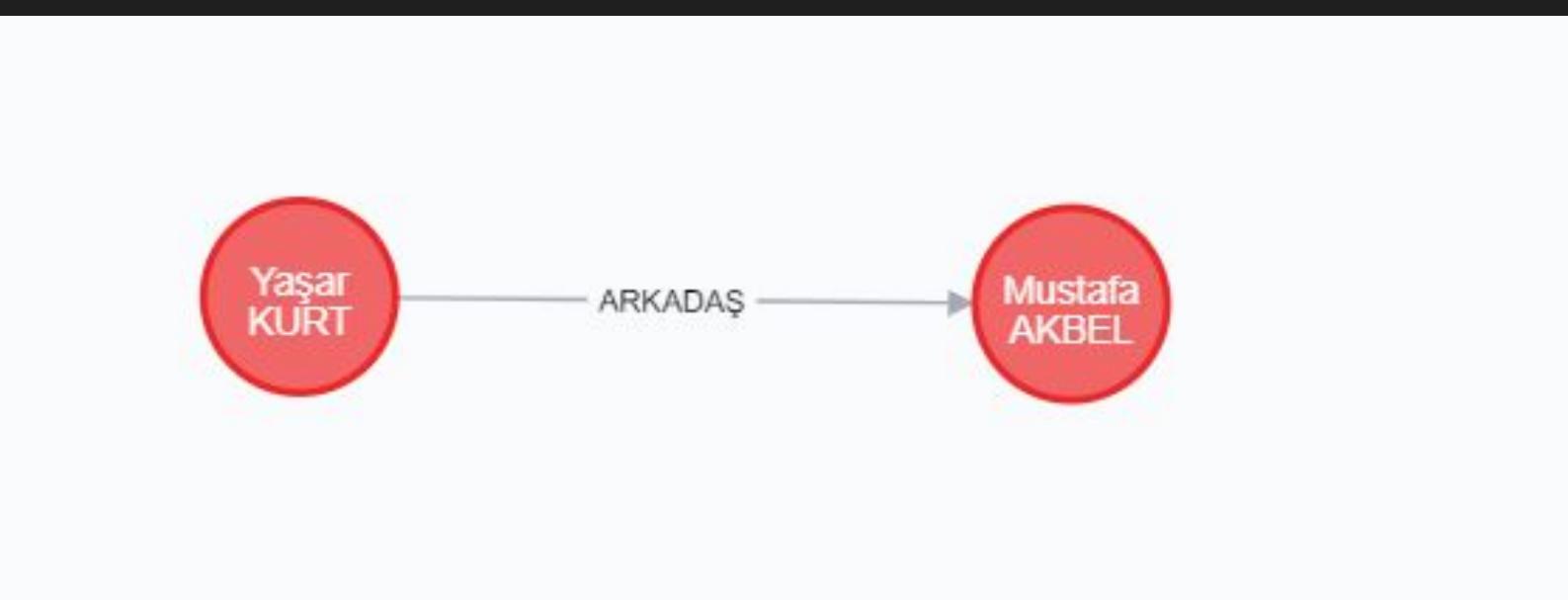
```
DELETE r
```

```
$ MATCH (n:Kisiler{name:"Mustafa AKBEL"})-[r:ARKADAŞ]->(b:Kisiler{name:"Yaşar KURT"}) DELETE r
```

Deleted 1 relationship, completed after 9 ms.

Table

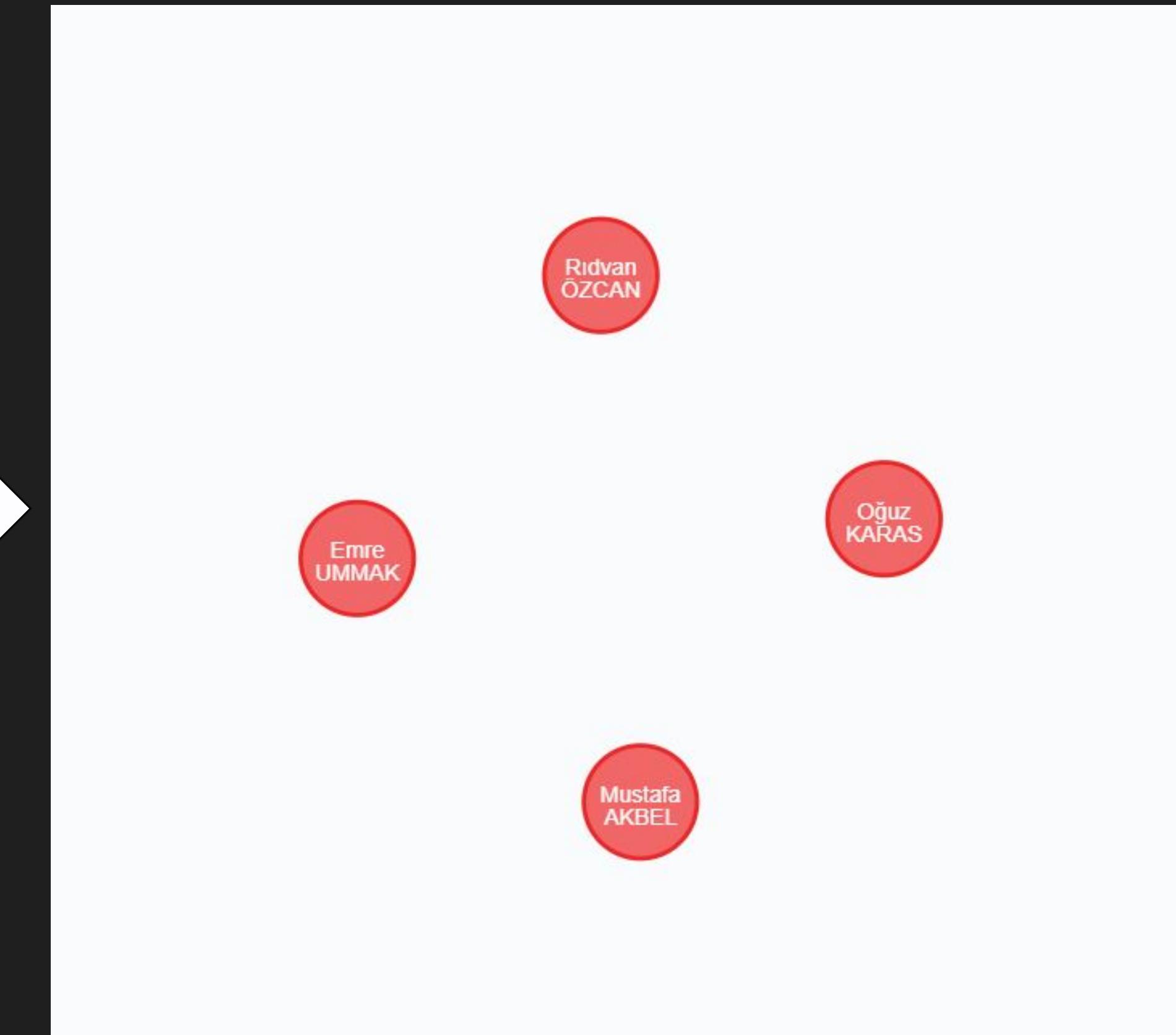
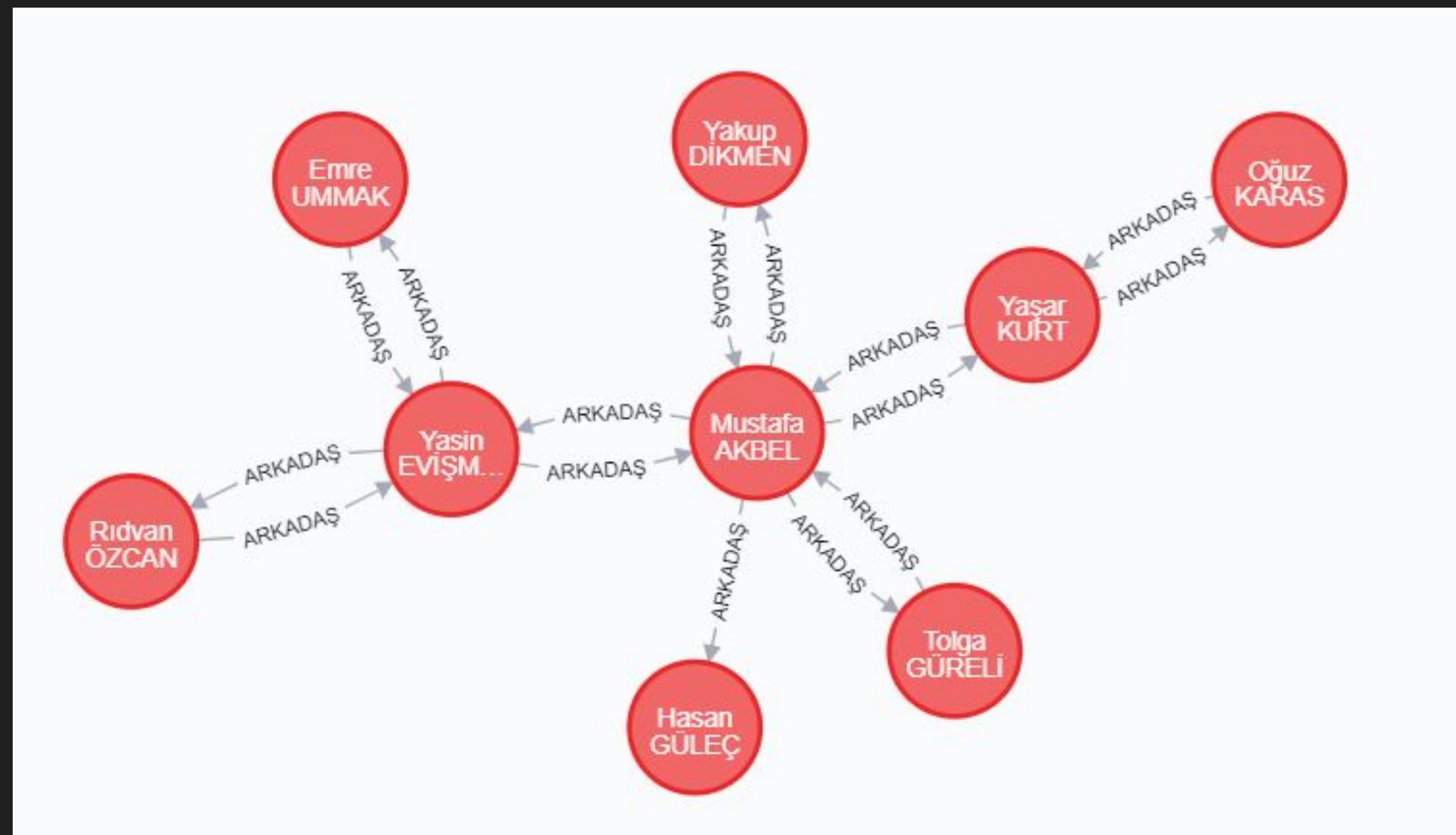
Code



# ÇEŞİTLİ SORGU ÖRNEKLERİ

Mustafa AKBEL'in arkadaşlarının  
arkadaşlarını bulan sorgu.

AKBEL"})-[:ARKADAŞ]->(c:Kisiler)-[:ARKADAŞ]->(f:Kisiler) RETURN f



C:\Users\yakup\_dikmen\Desktop\Redis\Redis-x64-3.2.100\redis-server.exe

[6548] 16 Apr 20:56:47.264 # Warning: no config file specified, using the default config. In order to specify a config file use C:\Users\yakup\_dikmen\Desktop\Redis\Redis-x64-3.2.100\redis-server.exe /path/to/redis.conf

Redis 3.2.100 (00000000/0) 64 bit

Running in standalone mode

Port: 6379

PID: 6548

http://redis.io

[6548] 16 Apr 20:56:47.264 # Server started, Redis version 3.2.100

[6548] 16 Apr 20:56:47.280 \* DB loaded from disk: 0.002 seconds

[6548] 16 Apr 20:56:47.280 \* The server is now ready to accept connections on port 6379

dis

C:\Users\yakup\_dikmen\Desktop\Redis\Redis-x64-3.2.100\redis-cli.exe

127.0.0.1:6379>

RE-

# REDIS NEDİR?

Redis key-value şeklinde tasarlanmış bir NOSQL veritabanıdır.

---

Verileri HDD'ye yazmadan RAM üzerinde tutarak işlem yapar.

---

Açık kaynak kodlu.

---

C dilinde yazılmıştır.

# NEDEN REDis?

Çalıştığı tüm operasyonları bellekte  
çalıştırdığı için oldukça hızlıdır.

# BELLEKTE ÇOK YER TUTMAZMI ?

bellekte 1 mb yer kaplamaktadır.

- Redis, hiç bir veri barındırmazken bellekte 1 mb yer kaplamaktadır.
- <String, String> işleminin 1 milyon adedi bellekte sadece 100 mb yer kaplar.
- <Hash, Obj> şeklinde ki 1 milyon adedi ise bellekte sadece 200 mb yer kaplar

# KEY TANIMLAM

**A**  
*Anahtar ve değer atama*

---

*Syntax:* SET KEY\_NAME VALUE

*Anahtarın değerini doldürme*

---

*Syntax:* GET KEY\_NAME

```
C:\Users\yakup_dikmen\Desktop\Redis\Redis-x64-3.2.100\redis-cli.exe
```

```
127.0.0.1:6379> set sehir istanbul
```

```
OK
```

```
127.0.0.1:6379> get sehir
```

```
"istanbul"
```

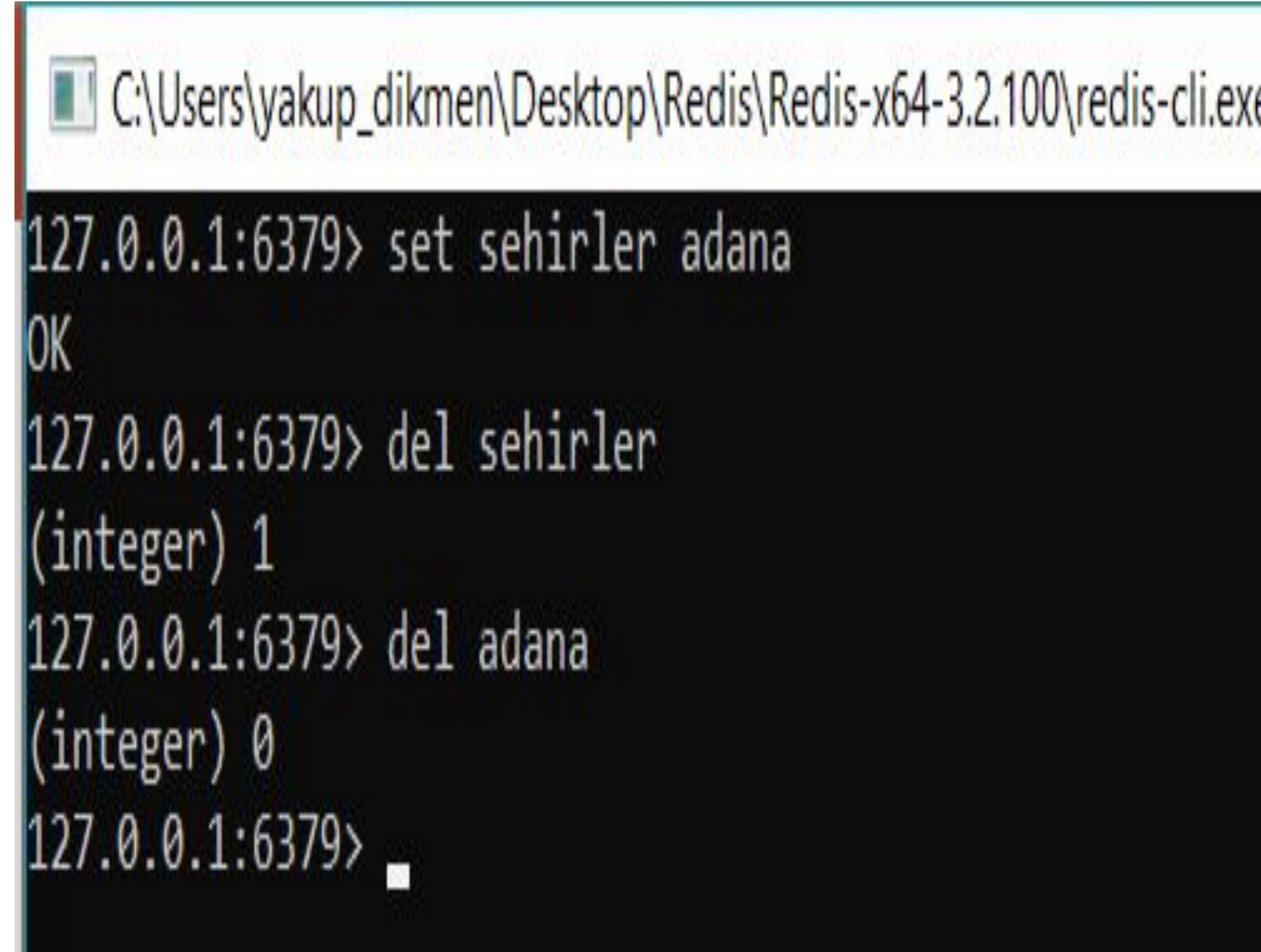
```
127.0.0.1:6379>
```

# KEY SILME

*Syntax:* DEL KEY\_NAME

*Geri dönüş değeri:*

- 0 ise anahtar silinemedi ya da yok.
- 1 ise anahtar silindi.



The screenshot shows a terminal window titled 'C:\Users\yakup\_dikmen\Desktop\Redis\Redis-x64-3.2.100\redis-cli.exe'. It displays the following Redis command history:

```
127.0.0.1:6379> set sehirler adana
OK
127.0.0.1:6379> del sehirler
(integer) 1
127.0.0.1:6379> del adana
(integer) 0
127.0.0.1:6379> ■
```

# KEY

Syntax: EXISTS KEY NAME

*Geri dönüş değeri:*

- 0 ise anahtar var
- 1 ise anahtar yok.

```
C:\Users\yakup_dikmen\Desktop\Redis\Redis-x64-3.2.100\redis-cli.exe
127.0.0.1:6379> set sehirler adana
OK
127.0.0.1:6379> exists sehirler
(integer) 1
127.0.0.1:6379> exists adana
(integer) 0
127.0.0.1:6379>
```

# KEYE ZAMAN VERME

*Syntax:* EXPIRE KEY\_NAME  
TIME\_IN\_SECONDS

*Geri dönüş değeri:*

- 0 ise anahtar için zaman ayarlandı.
- 1 ise ayarlanamadı.

```
127.0.0.1:6379> set sehirler adana
OK
127.0.0.1:6379> get sehirler
"adana"
127.0.0.1:6379> expire sehirler 10
(integer) 1
127.0.0.1:6379> get sehirler
"adana"
127.0.0.1:6379> get sehirler
"adana"
127.0.0.1:6379> get sehirler
(nil)
127.0.0.1:6379>
```

# KEYE

# ZAMAN

*Syntax:* PEXPIRE KEY NAME  
TIME\_IN\_MILLISECONDS

**Geri dönüş değeri:**

- 0 ise anahtar için zaman ayarlandı.
- 1 ise ayarlanamadı.

```
C:\Users\yakup_dikmen\Desktop\Redis\Redis-x64-3.2.100\redis-cli.exe
127.0.0.1:6379> set sehirler adana
OK
127.0.0.1:6379> pexpire sehirler 10000
(integer) 1
```

# KEYİN KALAN ZAMANI?

*Syntax:* TTL KEY\_NAME  
*Geri dönüşүş değeri:*

- Saniye türünden integer değer.
- Zamanı yoksa -2 döner.

```
C:\Users\yakup_dikmen\Desktop\Redis\Redis-x64-3.2.100\redis-cli.exe
127.0.0.1:6379> set sehirler adana
OK
127.0.0.1:6379> pexpire sehirler 10000
(integer) 1
127.0.0.1:6379> ttl sehirler
(integer) -2
127.0.0.1:6379> set sehirler adana
OK
127.0.0.1:6379> expire sehirler 60
(integer) 1
127.0.0.1:6379> ttl sehirler
(integer) 55
127.0.0.1:6379> ttl sehirler
(integer) 48
127.0.0.1:6379>
```

# KEYİN KOTROLÜ

**Syntax:** KEY S\* (s ile başlayan anahtar listesi)

**Syntax:** KEY \* (tüm anahtar listesi)

**Geri dönüş değeri:**

- Liste

```
C:\Users\yakup_dikmen\Desktop\Redis\Redis-x64-3.2.100\redis-cli.exe
127.0.0.1:6379> set sehirler adana
OK
127.0.0.1:6379> set sehir istanbul
OK
127.0.0.1:6379> set seh trabzon
OK
127.0.0.1:6379> keys s*
1) "seh"
2) "sehir5"
3) "sehirler"
4) "sehir"
5) "sayi"
127.0.0.1:6379> del sayi
(integer) 1
127.0.0.1:6379> keys s*
1) "seh"
2) "sehir5"
3) "sehirler"
4) "sehir"
127.0.0.1:6379> del sehir5
(integer) 1
127.0.0.1:6379> keys s*
1) "seh"
2) "sehirler"
3) "sehir"
```

# RANDOM KEY

**Syntax:** RANDOMKEY

**Geri dönüş değeri:**

- Varolan random bir key.

```
C:\Users\yakup_dikmen\Desktop\Redis\Redis-x64-3.2.100\redis-cli.exe
127.0.0.1:6379> keys *
1) "seh"
2) "listem"
3) "sehirler"
4) "name"
5) "key"
6) "sehir"
7) "yenisehir4"
127.0.0.1:6379> randomkey
"sehirler"
127.0.0.1:6379> randomkey
"sehir"
127.0.0.1:6379> randomkey
"seh"
127.0.0.1:6379>
```

# KEYİ YENİDEN ~~ADLANDIR~~

*Syntax:* RENAME KEY\_NAME

NEWKEY\_NAME

~~MA~~  
*Geri dönüş değeri:*

- Eski anahtar veya yeni anahtar adları eşitse veya anahtar yoksa hata döndürür.

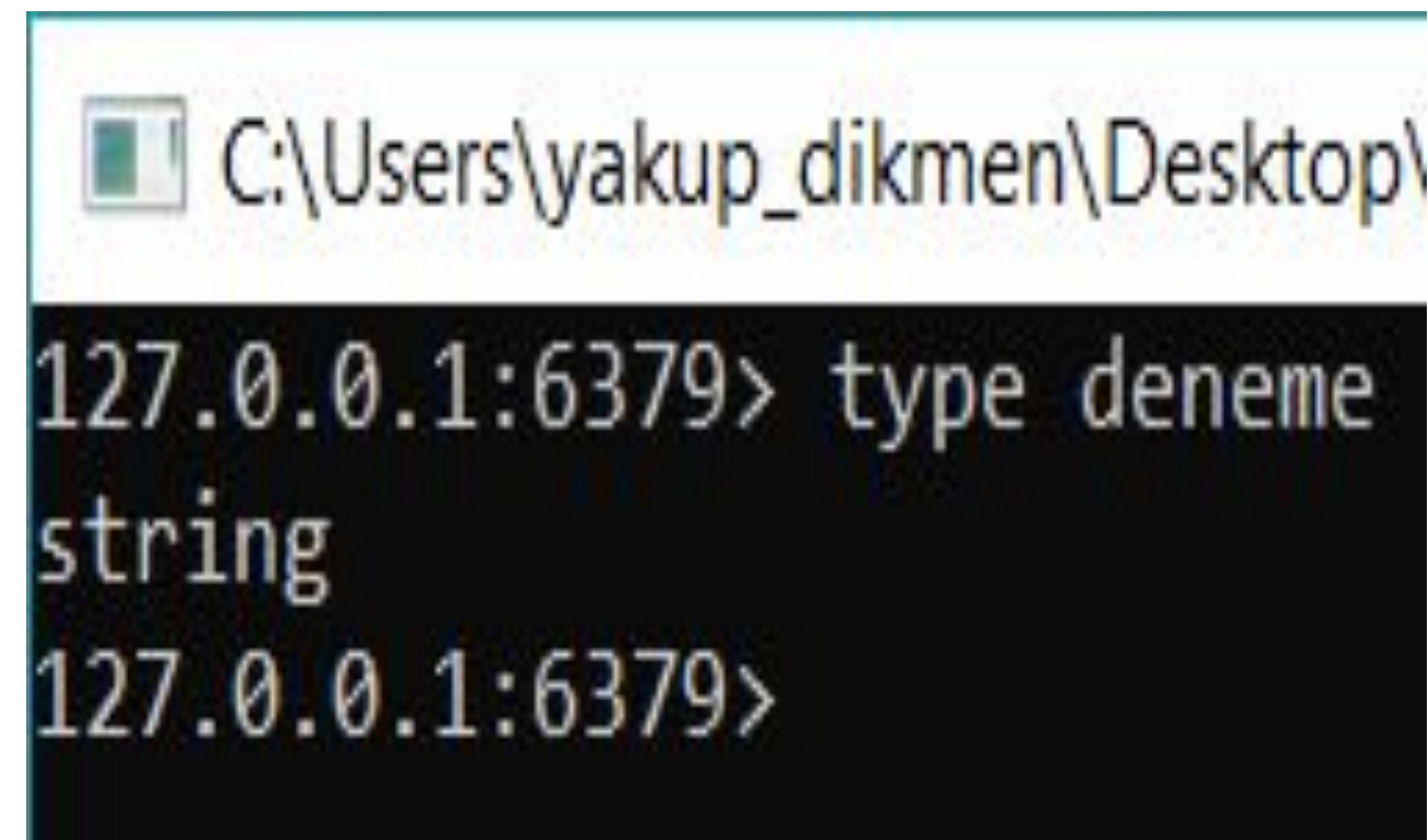
```
C:\Users\yakup_dikmen\Desktop\Redis\R
127.0.0.1:6379> keys *
1) "seh"
2) "listem"
3) "iller"
4) "name"
5) "key"
6) "sehir"
7) "yenisehir4"
127.0.0.1:6379> rename key deneme
OK
127.0.0.1:6379> keys *
1) "seh"
2) "listem"
3) "iller"
4) "deneme"
5) "name"
6) "sehir"
7) "yenisehir4"
127.0.0.1:6379>
```

# VERİ TİPi

**Syntax:** TYPE KEY\_NAME

**Geri dönüş değeri:**

- Tutulan verinin tipi.



The screenshot shows a terminal window with a light blue header bar containing a small green square icon and the path 'C:\Users\yakup\_dikmen\Desktop'. The main area of the terminal is black and displays the following text:  
127.0.0.1:6379> type deneme  
string  
127.0.0.1:6379>

# BİRDEN FAZLA

**KEY**

*Syntax:* MSET KEY\_NAME1

VALUE1 KEY\_NAME2 VALUE2

.....

*Syntax:* MGET KEY\_NAME1

VALUE1 KEY\_NAME2 VALUE2

.....

```
C:\Users\yakup_dikmen\Desktop\Redis\Redis-x64-3.2.100\redis-cli.exe
127.0.0.1:6379> mset sehir1 adana sehir2 mersin
OK
127.0.0.1:6379> keys *
1) "seh"
2) "listem"
3) "int"
4) "iller"
5) "deneme"
6) "sehir1"
7) "name"
8) "sehir2"
9) "sehir"
10) "yenisehir4"
127.0.0.1:6379> mget sehir1 sehir2 int
1) "adana"
2) "mersin"
3) "5"
127.0.0.1:6379> ■
```

# KEYDE Kİ VERİNİN İSTENİLEN

*Syntax:* GETRANGE KEY\_NAME

KİSMI  
START END

```
C:\Users\yakup_dikmen\Desktop\Redis\Redis-x64-3.2.100\redis-cli.exe
127.0.0.1:6379> get sehir1
"adana"
127.0.0.1:6379> getrange sehir1 0 3
"adan"
127.0.0.1:6379> getrange sehir1 1 4
"dana"
127.0.0.1:6379>
```

# KEYDE Kİ VERİNİN iSTENİLEN **KİSMINA** START VALUE **EKLEMЕ**

*Syntax:* SETRANGE KEY\_NAME

START VALUE

```
127.0.0.1:6379> set name istanbul
OK
127.0.0.1:6379> setrange name 8 adana
(integer) 13
127.0.0.1:6379> get name
"istanbuladana"
127.0.0.1:6379>
```

```
127.0.0.1:6379> set deneme yakup
OK
127.0.0.1:6379> setrange deneme 3 dikmen
(integer) 9
127.0.0.1:6379> get deneme
"yakdikmen"
127.0.0.1:6379>
```

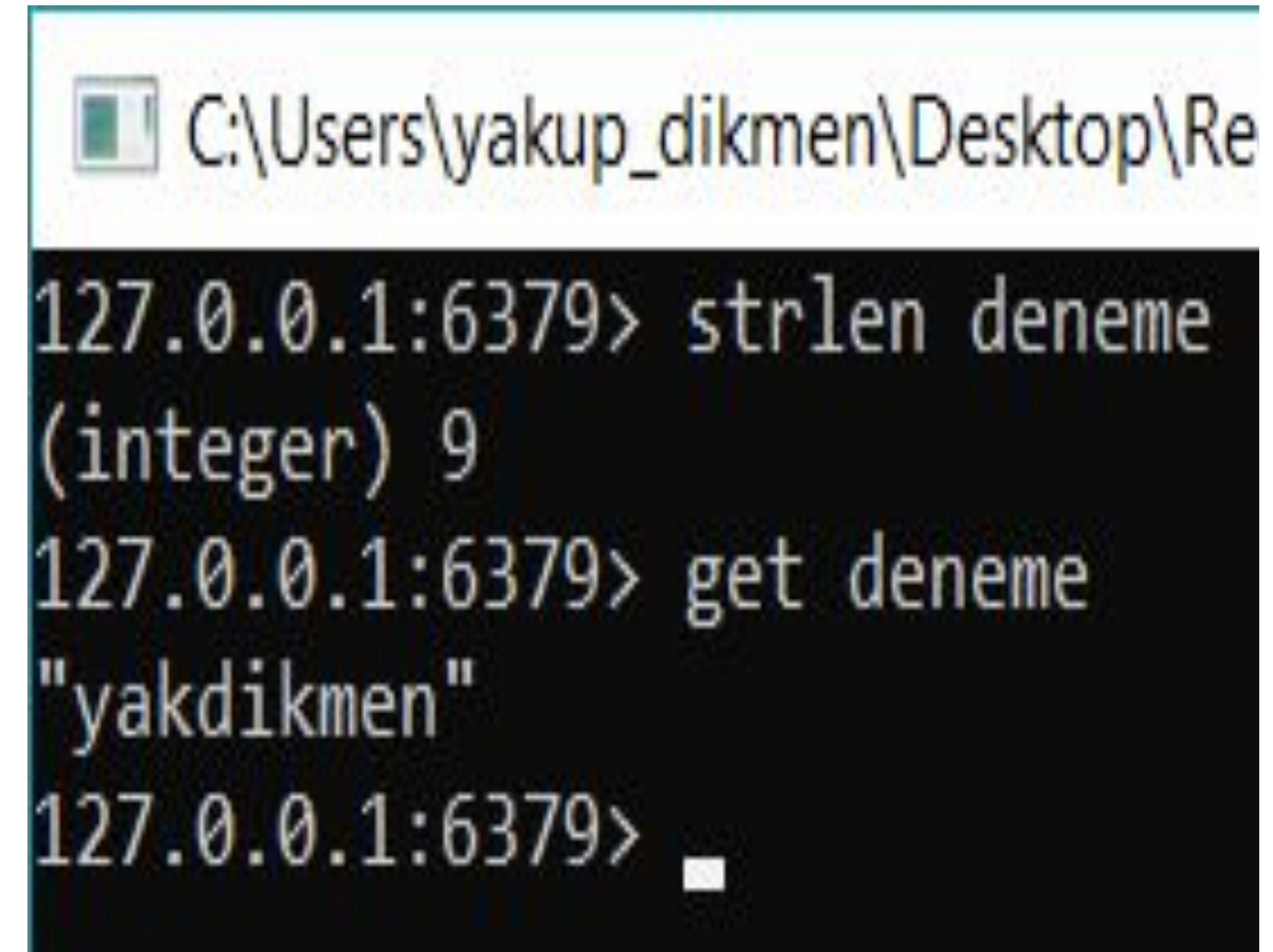
```
127.0.0.1:6379> set deneme yakup
OK
127.0.0.1:6379> setrange deneme 8 dikmen
(integer) 14
127.0.0.1:6379> get deneme
"yakup\x00\x00\x00dikmen"
127.0.0.1:6379>
```

# KEYDE Ki VERİNİN UZUNLUĞU

Syntax: STRLEN KEY\_NAME

*Geri dönüş değeri:*

- Integer değer.



The screenshot shows a terminal window with a black background and white text. At the top, there is a path: C:\Users\yakup\_dikmen\Desktop\Re... followed by a file icon. Below this, the Redis prompt 127.0.0.1:6379> is shown, followed by the command strlen deneme. The response is (integer) 9. Another prompt 127.0.0.1:6379> is shown, followed by get deneme, and the response "yakdikmen". A final prompt 127.0.0.1:6379> is shown with a small square icon next to it.

```
C:\Users\yakup_dikmen\Desktop\Re...
127.0.0.1:6379> strlen deneme
(integer) 9
127.0.0.1:6379> get deneme
"yakdikmen"
127.0.0.1:6379> ■
```

# KEYDE Ki VERİNİN UZUNLUĞU

Syntax: STRLEN KEY\_NAME

*Geri dönüş değeri:*

- Integer değer.

```
C:\Users\yakup_dikmen\Desktop\Redis\Redis-x64-3.2.100\redis-cli.exe
127.0.0.1:6379> strlen sehir1
(integer) 5
127.0.0.1:6379> strlen sehir2
(integer) 6
127.0.0.1:6379> mget sehir1 sehir2
1) "adana"
2) "mersin"
127.0.0.1:6379>
```

# HASHES VE TİPi

*Syntax:* HSET KEY\_NAME INDEX

VALUE

```
C:\Users\yakup_dikmen\Desktop\Redis\Redis-x64-3.2.100\redis-cli.exe
127.0.0.1:6379> hset hash 0 deneme0
(integer) 1
127.0.0.1:6379> hset hash 1 deneme1
(integer) 1
127.0.0.1:6379> hset hash 2 deneme2
(integer) 1
127.0.0.1:6379> hget hash 2
"deneme2"
127.0.0.1:6379> hgetall hash
1) "0"
2) "deneme0"
3) "1"
4) "deneme1"
5) "2"
6) "deneme2"
127.0.0.1:6379> .
```

# SETS VERİ TİPi

**Syntax:** SADD KEY\_NAME

VALUE

```
127.0.0.1:6379> sadd veritabani redis
(integer) 1
127.0.0.1:6379> sadd veritabani mongodb
(integer) 1
127.0.0.1:6379> sadd veritabani neo4j
(integer) 1
127.0.0.1:6379> smembers veritabani
1) "neo4j"
2) "mongodb"
3) "redis"
127.0.0.1:6379> ■
```

# SORTED SETS VERİ TIPI

*Syntax:* ZADD KEY\_NAME INDEX

VALUE

```
C:\Users\yakup_dikmen\Desktop\Redis\Redis-  
127.0.0.1:6379> zadd yeni 0 redis  
(integer) 1  
127.0.0.1:6379> zadd yeni 1 mongodb  
(integer) 1  
127.0.0.1:6379> zadd yeni 2 neo4j  
(integer) 1  
127.0.0.1:6379> zrange yeni 0 2  
1) "redis"  
2) "mongodb"  
3) "neo4j"  
127.0.0.1:6379> zrange yeni 1 2  
1) "mongodb"  
2) "neo4j"
```

# LISTE VERİ

Tipi

Syntax: LPUSH KEY\_NAME

VALUE

LRANGE KEY\_NAME START

AND

```
127.0.0.1:6379> llen list
(integer) 1
127.0.0.1:6379> llen list
(integer) 1
127.0.0.1:6379> lpush list mongodb
(integer) 2
127.0.0.1:6379> llen list
(integer) 2
127.0.0.1:6379>
```

```
C:\Users\yakup_dikmen\Desktop\Redis\Re
127.0.0.1:6379> lpush list redis
(integer) 1
127.0.0.1:6379> lpush list mongodb
(integer) 2
127.0.0.1:6379> lpush list neo4j
(integer) 3
127.0.0.1:6379> lrangle list 0 2
1) "neo4j"
2) "mongodb"
3) "redis"
127.0.0.1:6379> lpop list
"neo4j"
127.0.0.1:6379> lpop list
"mongodb"
127.0.0.1:6379> lrangle list 0 2
1) "redis"
```