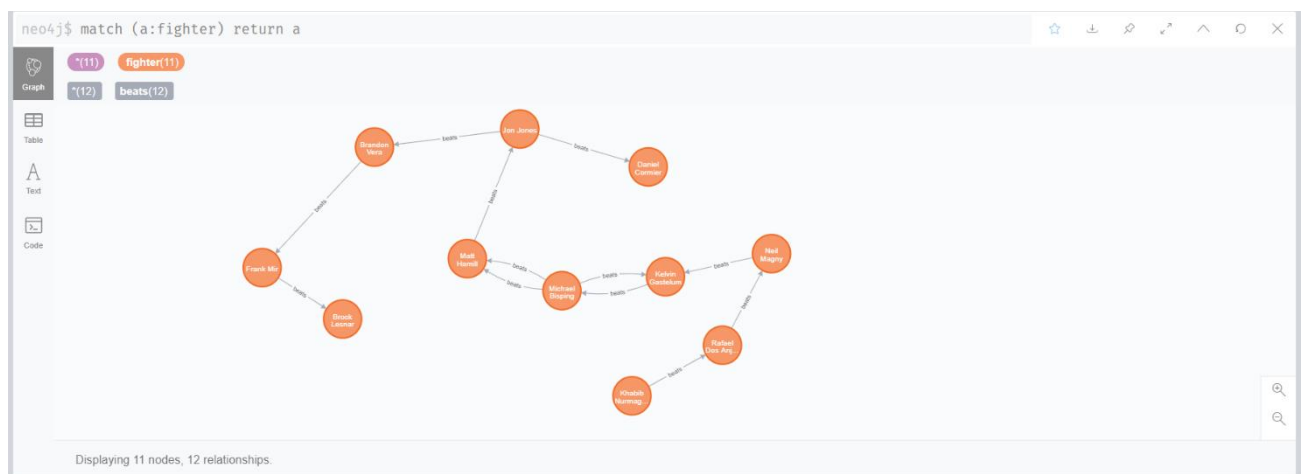


1)

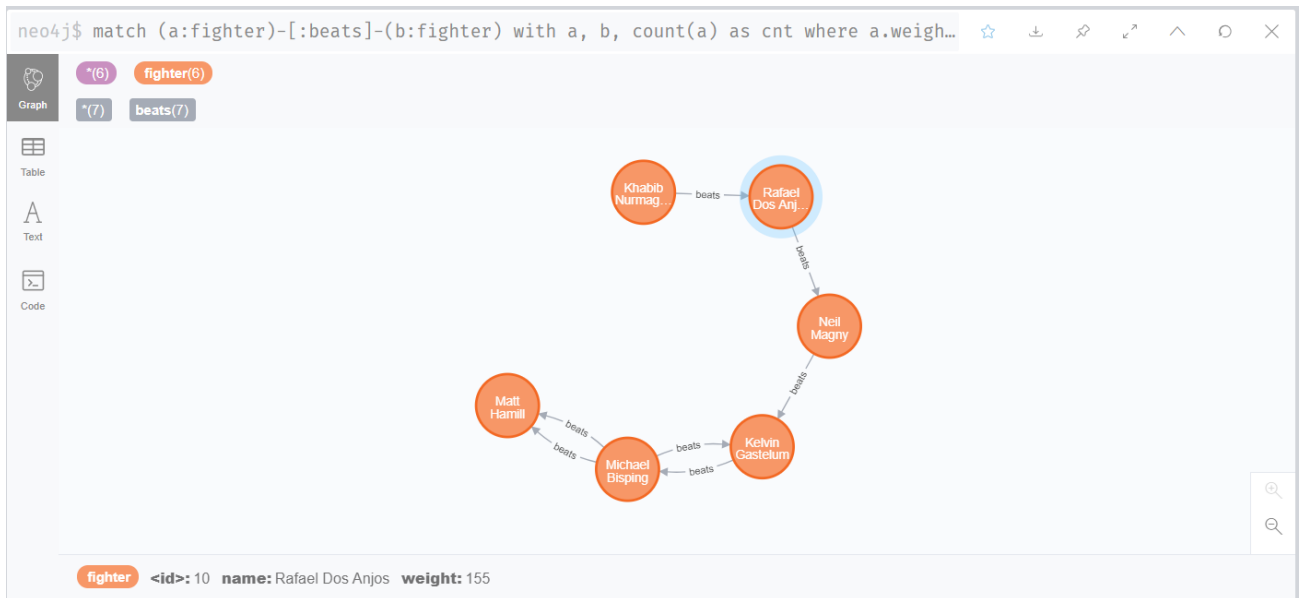
```
create (p1:fighter{name:"Brandon Vera", weight: "205"}),  
(p2:fighter{name:"Brock Lesnar", weight: "230"}),  
(p3:fighter{name:"Daniel Cormier", weight: "205"}),  
(p4:fighter{name:"Frank Mir", weight: "230"}),  
(p5:fighter {name:"Jon Jones", weight: "205"}),  
(p6:fighter {name:"Kelvin Gastelum", weight: "185"}),  
(p7:fighter {name:"Khabib Nurmagomedov", weight: "155"}),  
(p8:fighter {name:"Matt Hamill", weight: "185"}),  
(p9:fighter {name:"Michael Bisping", weight: "185"}),  
(p10:fighter {name:"Neil Magny", weight: "170"}),  
(p11:fighter {name:"Rafael Dos Anjos", weight: "155"}),
```

```
(p7)-[:beats]->(p11),(p11)-[:beats]->(p10),  
(p5)-[:beats]->(p3), (p9)-[:beats]->(p8),  
(p5)-[:beats]->(p1), (p1)-[:beats]->(p4),  
(p4)-[:beats]->(p2), (p10)-[:beats]->(p6),  
(p6)-[:beats]->(p9), (p9)-[:beats]->(p8),  
(p9)-[:beats]->(p6), (p8)-[:beats]->(p5);
```



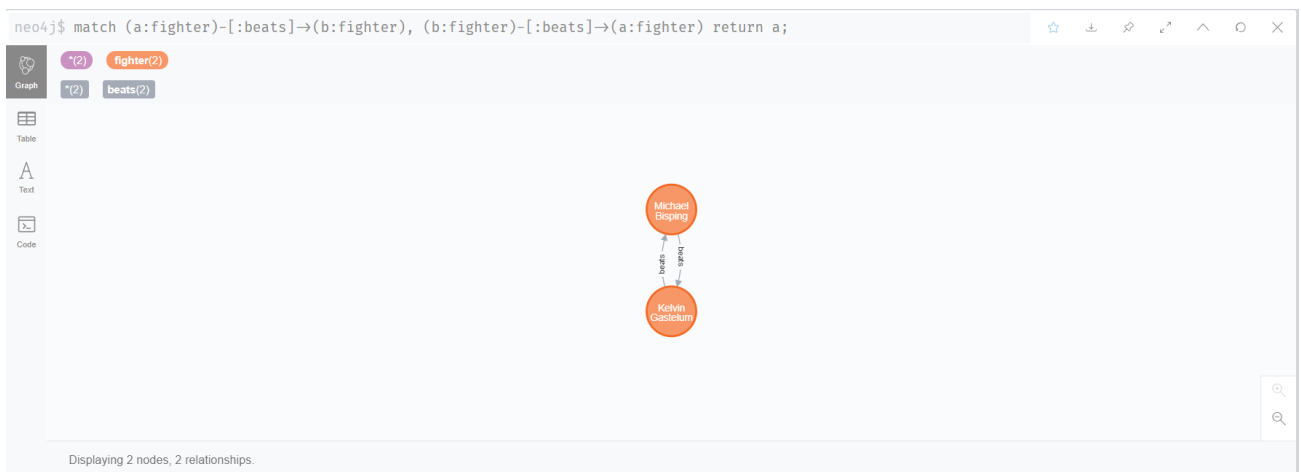
2) Return all **middle/Walter/lightweight** fighters (155,170,185) who at least have one win.

match (a:fighter)-[:beats]-(b:fighter) with a, b, count(a) as cnt where a.weight in ["155","185","170"] AND cnt>0 return a;



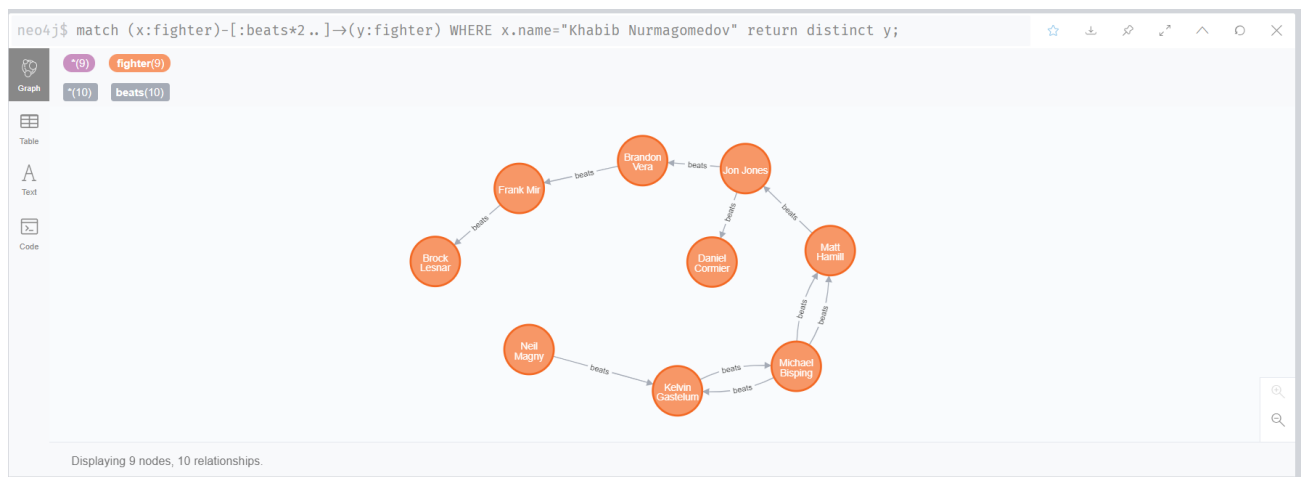
3) Return fighters who had **1-1 record** with each other.

match (a:fighter)-[:beats]->(b:fighter), (b:fighter)-[:beats]->(a:fighter) return a;



4) Return all fighters that **"Khabib Nurmagomedov"** can beat and he didn't have a fight with them yet.

match (a:fighter)-[:beats*2..]->(b:fighter) WHERE a.name="Khabib Nurmagomedov" return distinct b;



5) Return **undefeated Fighters (0 loss)**, defeated fighter (0 wins).

match (a:fighter) where not (a)-[:beats]->(a) return a



match (a:fighter) where not (a)-[:beats]->(a) return a



6) Return all fighter's MMA records and create query to enter the record as a property for a fighter {name, weight, record}.

```
match (a:fighter{name: "Matt Hamill"}) set a.record = 0  
match (a:fighter) return a.name, a.record
```

The screenshot displays the Neo4j Desktop interface. The top panel shows a Cypher query: `neo4j$ match (a:fighter) return a.name, a.record`. The results are shown in a table view with two columns: `"a.name"` and `"a.record"`. The table lists 12 fighters, with `"Matt Hamill"` having a record of 0 and the others having null records. The bottom panel shows a second query: `neo4j$ match (a:fighter{name: "Matt Hamill"}) SET a.record = 0`, with a status message: "Set 1 property, completed after 15 ms."

"a.name"	"a.record"
"Brandon Vera"	null
"Brock Lesnar"	null
"Daniel Cormier"	null
"Frank Mir"	null
"Jon Jones"	null
"Kelvin Gastelum"	null
"Khabib Nurmagomedov"	null
"Matt Hamill"	0
"Michael Bisping"	null
"Neil Magny"	null
"Rafael Dos Anjos"	null

MAX COLUMN WIDTH:

neo4j\$ match (a:fighter{name: "Matt Hamill"}) SET a.record = 0

Set 1 property, completed after 15 ms.