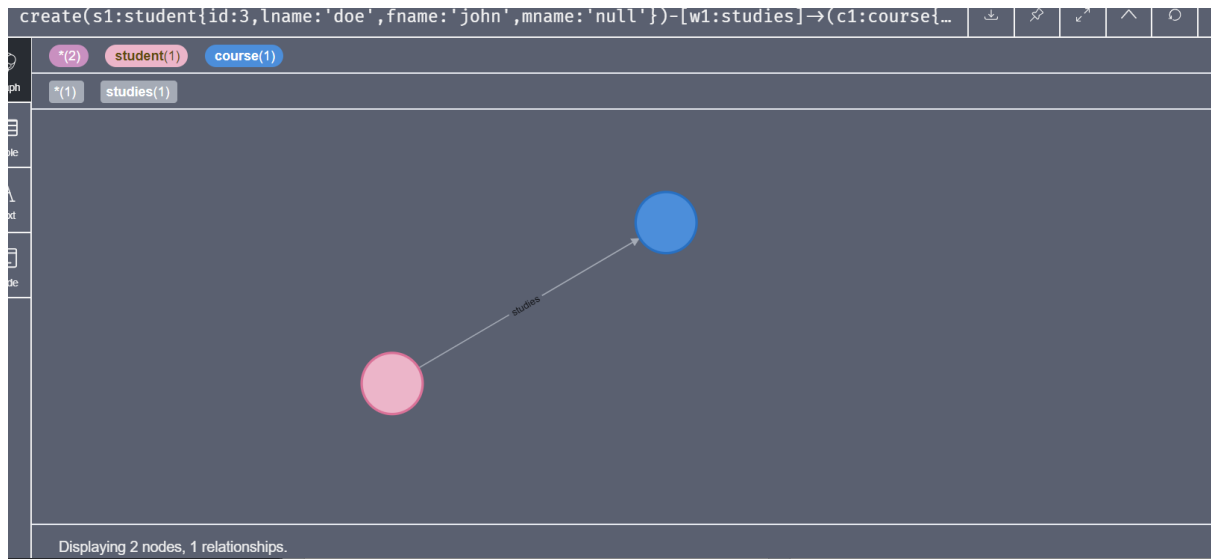
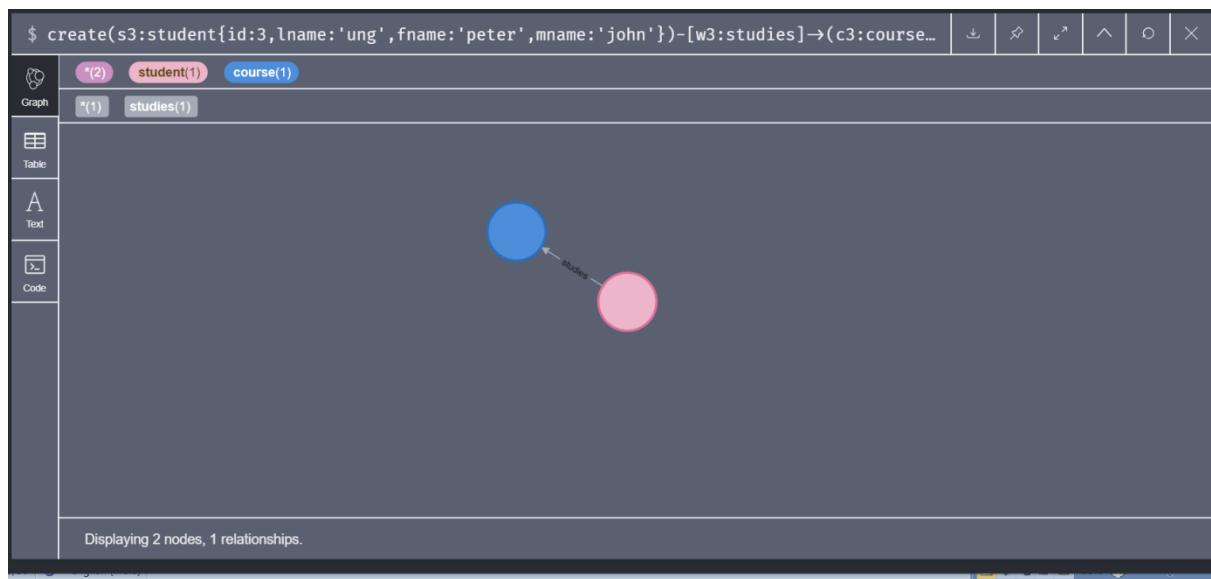


```
create(s1:student{id:3,lname:'doe',fname:'john',mname:'null'})-[w1:studies]->(c1:course{num:'c2',name:'programming'}) return s1,w1,c1
```



```
create(s3:student{id:3,lname:'ung',fname:'peter',mname:'john'})-[w3:studies]->(c3:course{num:'c2',name:'programming'}) return s3,w3,c3
```



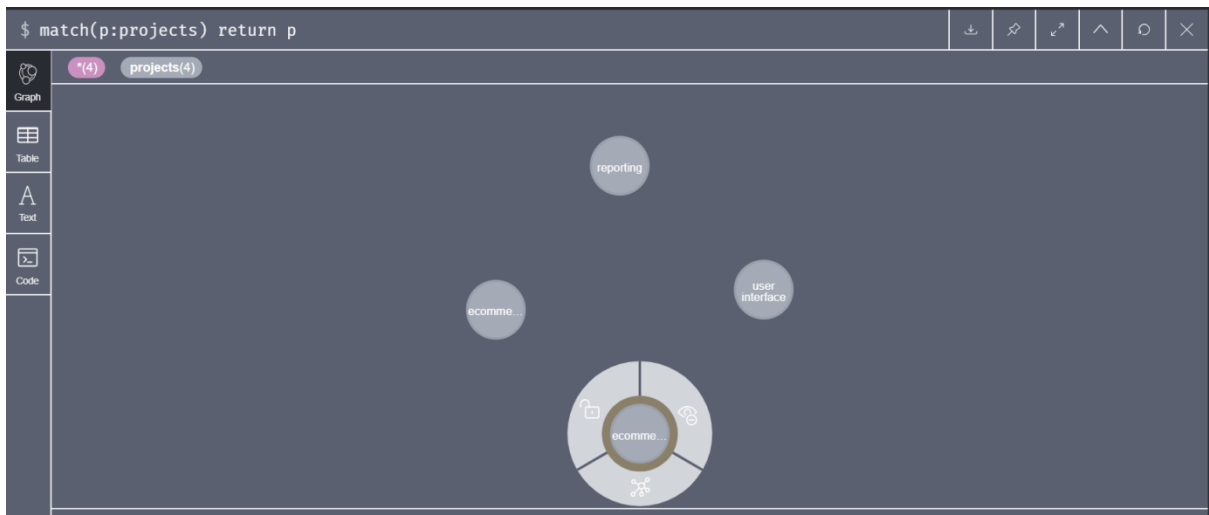
```
create(p1:projects{no:34,name:'ecommerce database'}) return p1
```

```
create(p2:projects{no:24,name:'ecommerce website'}) return p2
```

```
create(p3:projects{no:13,name:'user interface'}) return p3
```

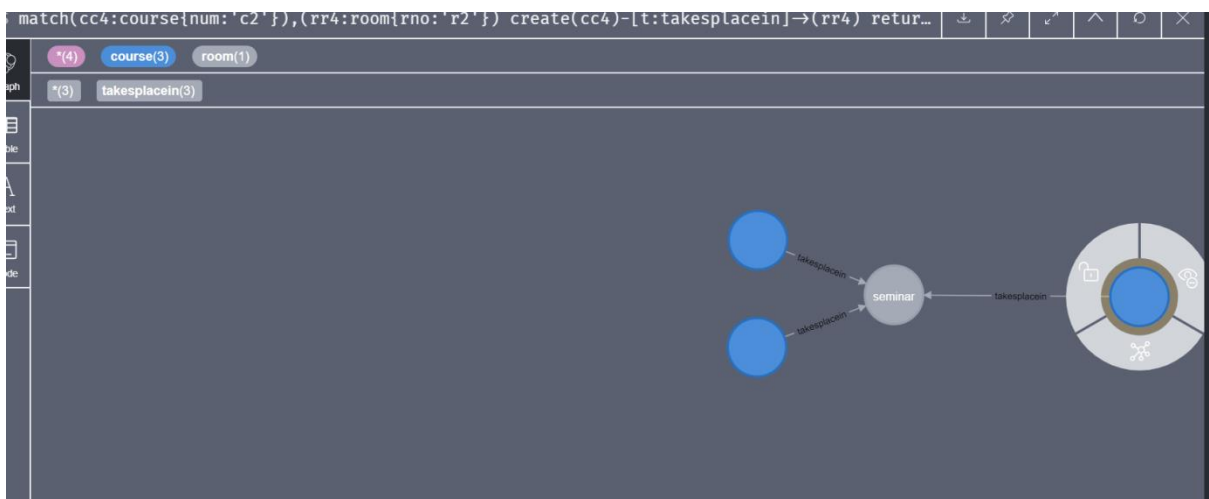
```
create(p4:projects{no:26,name:'reporting'}) return p4
```

```
match(p:projects) return p
```



```
match(st1:student{id:1}),(pr1:projects{no:'p1'}) create(st1)-[w:workson]->(pr1) set w.hrs=1
return st1,w,pr1
```

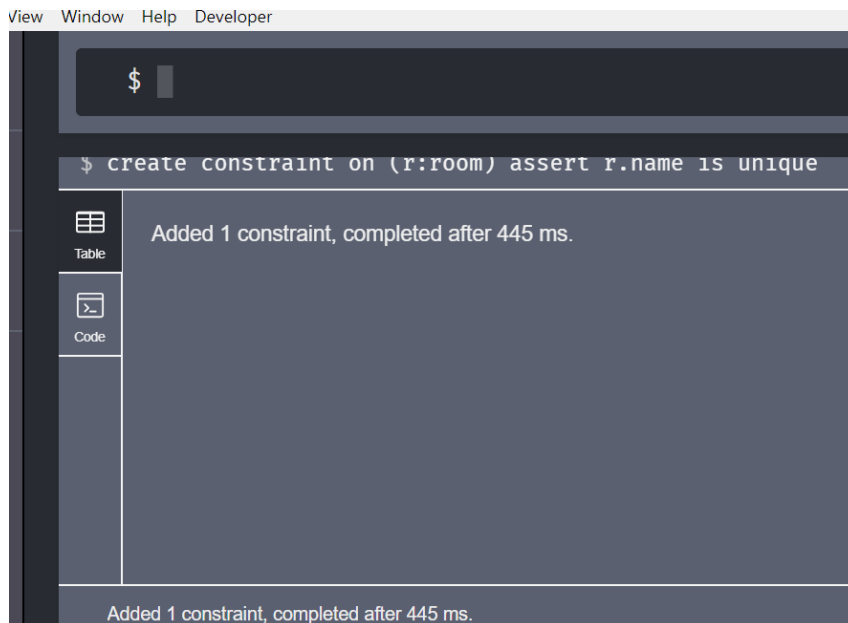
```
match(cc4:course{num:'c2'}),(rr4:room{rno:'r2'}) create(cc4)-[t:takesplacein]->(rr4) return
cc4,t,rr4
```



```
match(s:student{id:2}) set s.mname='null' return s
```

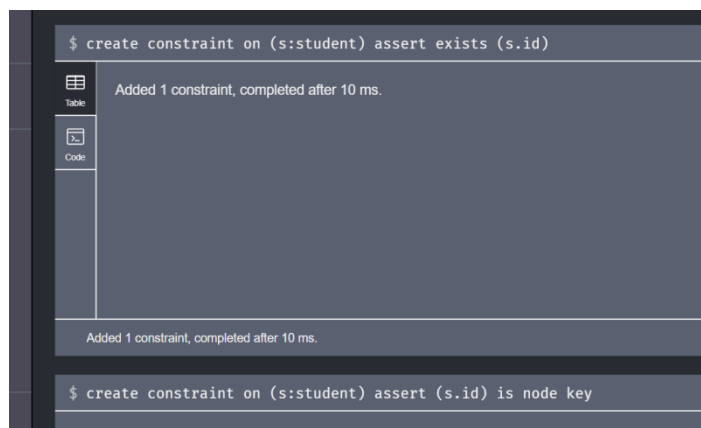


create constraint on (r:room) assert r.name is unique

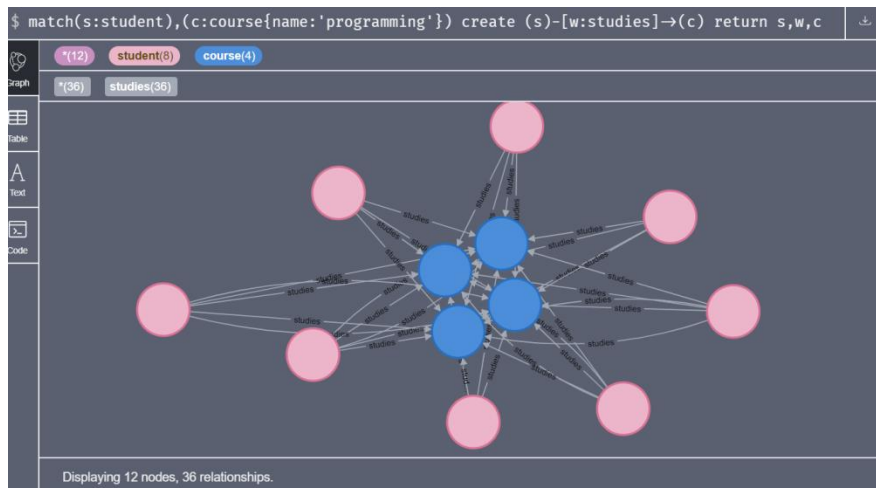


create constraint on (s:student) assert (s.id) is node key

create constraint on (s:student) assert exists (s.id)



```
match(s:student),(c:course{name:'programming'}) create (s)-[w:studies]->(c) return s,w,c
```



```
match(c:course{name:'programming'}) return c.num
```

| | |
|--|--------------|
| \$ match(c:course{name:'programming'}) return c.num | |
| Table | c.num |
| Text | "c2" |
| Code | "c2" |
| | "c2" |
| | "c2" |
| | "c2" |
| Started streaming 4 records after 28 ms and completed after 28 ms. | |

```
match(s:student{id:6})-[w3:studies]->(c:course{num:'c1'}) delete w3
```

| | |
|--|--|
| \$ match(s:student{id:6})-[w3:studies]->(c:course{num:'c1'}) delete w3 | |
| Table | |

```
match (c:course)-[t:takesplacein]->(r:room) return count(*)
```

| | |
|--|-----------------|
| \$ match (c:course)-[t:takesplacein]->(r:room) return count(*) | |
| Table | count(*) |
| Text | 9 |
| Code | |

```
match (c:course{num:'c1'})-[t:takesplacein]->(r:room) return count(*)
```

| | |
|---|----------|
| \$ match (c:course{num:'c1'})-[t:takesplacein]→(r:room) return count(*) | |
| Table | count(*) |
| Text | 6 |
| Code | |

match(s:student)-[w3:studies]→(c:course) return count(*),s.id

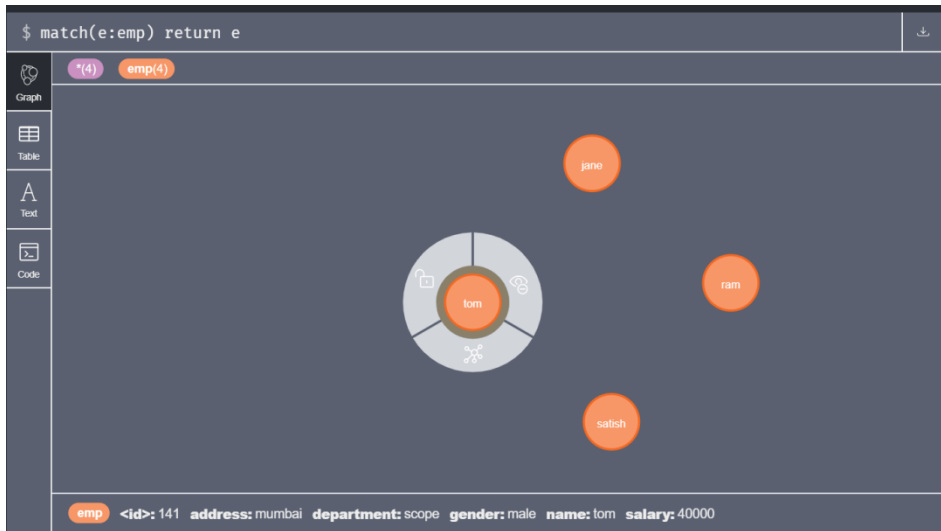
| | | |
|--|----------|------|
| \$ match(s:student)-[w3:studies]→(c:course) return count(*),s.id | | ↓ |
| Table | count(*) | s.id |
| Text | 13 | 1 |
| Code | 39 | 3 |
| | 12 | 5 |
| | 13 | 4 |
| | 13 | 2 |
| Started streaming 5 records after 84 ms and completed after 84 ms. | | |

match(s:student)-[w3:studies]→(c:course) return count(*),c.num

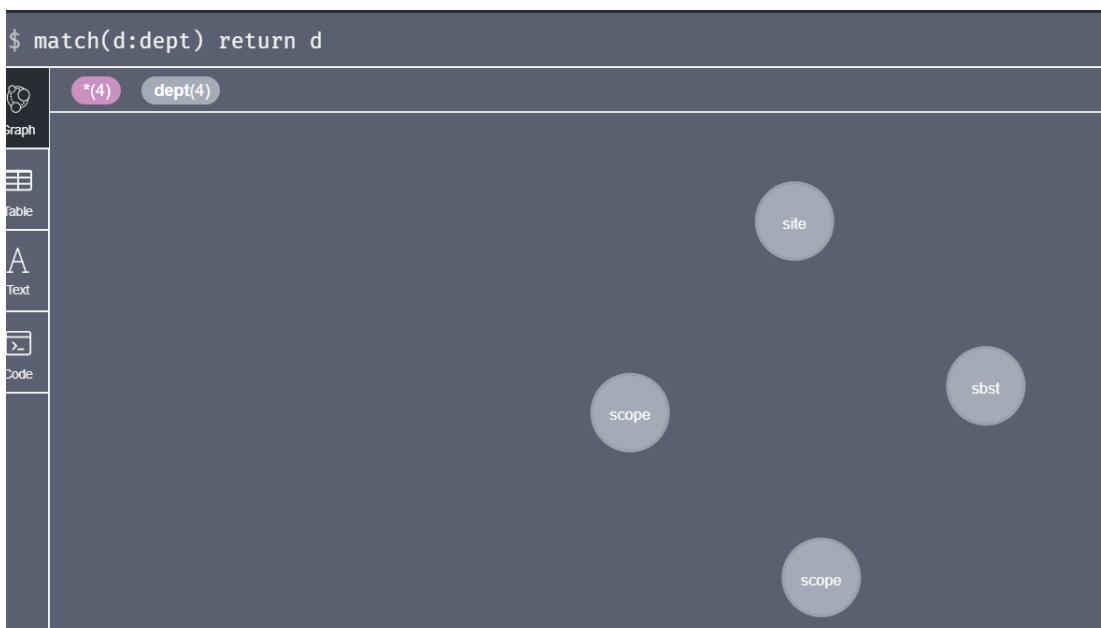
| | | | |
|---|----------|-------|---|
| \$ match(s:student)-[w3:studies]→(c:course) return count(*),c.num | | ↓ | ↗ |
| Table | count(*) | c.num | |
| Text | 88 | "c2" | |
| Code | 2 | "c1" | |
| | | | |

match(s:student)-[w3:studies]→(c:course{num:'c2'}) where s.fname starts with 'd' return s

match(e:emp) return e




match(d:dept) return d



match(p:project) return p

```
match(p:project) return p
```

*(1)
project(1)



project
<id>: 44
pname: dst

MATCH (n) RETURN distinct labels(n)

```
$ MATCH (n) RETURN distinct labels(n)
```

Table
Text
Code

labels(n)

["emp"]

["dept"]

["employee"]

["project"]

MATCH (n) RETURN distinct labels(n), count(*)

```
$ MATCH (n) RETURN distinct labels(n), count(*)
```

| | labels(n) | count(*) |
|-------|--------------|----------|
| Table | | |
| Text | ["emp"] | 4 |
| Code | ["dept"] | 4 |
| | ["employee"] | 1 |
| | ["project"] | 1 |
| | ["profect"] | 1 |

MATCH (n:dept) RETURN labels(n), count(*)

```
$ MATCH (n:dept) RETURN labels(n), count(*)
```

| | labels(n) | count(*) |
|-------|-----------|----------|
| Table | | |
| Text | ["dept"] | 4 |
| Code | | |
| | | |

Started streaming 1 records after 2 ms and completed after 2 ms.

match (e:emp{address:'chennai'}) return e

```
$ match (e:emp{address:'chennai'}) return e
```

| | | |
|-------|-------|---------|
| Graph | * (2) | emp (2) |
| Table | | |
| Text | | |
| Code | | |

The graph visualization shows two orange circular nodes. The top node is labeled 'jane' and the bottom node is labeled 'ram'. They are positioned vertically on a dark grey background.

MATCH (n)

WHERE size([label IN labels(n) WHERE label IN ['dept', 'emp'] | 1]) > 0

RETURN count(n)

| | |
|--|----------|
| \$ MATCH (n) WHERE size([label IN labels(n) WHERE label IN ['dept', 'emp'] 1]) > 0 RETURN... | |
| Table | count(n) |
| Text | 8 |

match (e:emp{gender:'male'}) return count(*)

| | |
|---|----------|
| \$ match (e:emp{gender:'male'}) return count(*) | |
| Table | count(*) |
| Text | 3 |

match(e:emp) where e.address in ['vellore','chennai'] return e

| | |
|---|-------------|
| \$ match(e:emp) where e.address in ['vellore','chennai'] return e | |
| Graph | *(3) emp(3) |
| Table | |
| Text | |
| Code | |



match(e:emp) where e.address in ['vellore','chennai'] return e.salary

```
$ match(e:emp) where e.address in ['vellore','chennai'] return e.salary
```

| | |
|-------|-----------------|
| Table | e.salary |
| Text | 20000 |
| Code | 60000 |
| | 30000 |

match(e:emp) where e.department in ['scope'] return e

```
$ match(e:emp) where e.department in ['scope'] return e
```

| | |
|-------|-------------|
| Graph | *(2) emp(2) |
| Table | |
| Text | |
| Code | |

The graph visualization shows two orange circular nodes. The top node is labeled 'tom' and the bottom node is labeled 'satish'. There are no edges visible between them.

match(e:emp) where e.department in ['sbst'] return e.gender,e.department

```
$ match(e:emp) where e.department in ['sbst'] return e.gender,e.department
```

| | | |
|-------|-----------------|---------------------|
| Table | e.gender | e.department |
| Text | "male" | "sbst" |
| Code | | |

Started streaming 1 records after 1 ms and completed after 3 ms.

match(e:emp) where e.salary > 25000 and e.department in ['scope'] return e

The image shows the Neo4j Cypher query interface. At the top, a query is entered: `$ match(e:emp) where e.salary > 25000 and e.department in ['scope'] return e`. Below the query, the results are displayed in a graph view. A single orange node labeled 'tom' is visible. On the left side, there is a sidebar with icons for Graph, Table, Text, and Code. The 'Graph' icon is selected.

CASUAL CLUSTERS

```
ar/lib/neo4j/certificates
ar/lib/neo4j/run

4.124+0000 INFO  ===== Neo4j 3.3.0-rc1 =====
4.233+0000 INFO  Starting...
5.157+0000 INFO  Bolt enabled on 0.0.0.0:7687.
5.188+0000 INFO  Initiating metrics...
5.071+0000 INFO  Started.
5.425+0000 INFO  Mounted REST API at: /db/manage
5.127+0000 INFO  Remote interface available at http://localhost:7474/
15.937+0000 INFO  Neo4j Server shutdown initiated by request
5.995+0000 INFO  Stopping...
5.150+0000 INFO  Stopped.
```

The image shows the Neo4j web interface. The browser address bar displays `172.18.0.2:7474/browser/`. The main content area features a sidebar on the left with icons for Apps, JDK, fritz, webshop, Databale, neo, and Zoom. The main area has a header with the Neo4j logo and a search bar. Below the header, there are three main sections: 'Learn about Neo4j', 'Jump into code', and 'Monitor the system'. Each section has a brief description and a 'Start Learning', 'Write Code', or 'Monitor' button respectively. The footer of the page reads 'Copyright © Neo Technology 2002-2017'.

☆

☰

\$:sysinfo

| Store Sizes | |
|--------------------|------------|
| Array Store | 8.00 KiB |
| Logical Log | 90 B |
| Node Store | 0 B |
| Property Store | 0 B |
| Relationship Store | 0 B |
| String Store | 8.00 KiB |
| Total Store Size | 136.31 KiB |

| ID Allocation | |
|----------------------|---|
| Node ID | 0 |
| Property ID | 0 |
| Relationship ID | 0 |
| Relationship Type ID | 0 |

| Page Cache | |
|---------------------|--------|
| Faults | 62 |
| Evictions | 0 |
| File Mappings | 93 |
| Bytes Read | 327650 |
| Flushes | 20 |
| Eviction Exceptions | 0 |
| File Unmappings | 76 |
| Bytes Written | 196598 |

| Transactions | |
|--------------|----|
| Last Tx Id | 2 |
| Current | 2 |
| Peak | 2 |
| Opened | 15 |
| Committed | 10 |

| Causal Cluster Members | | |
|------------------------|--|----------------------|
| Roles | Addresses | Actions |
| LEADER | bolt://core1:7687, http://core1:7474, https://core1:7473 | Open |
| FOLLOWER | bolt://core2:7687, http://core2:7474, https://core2:7473 | Open |
| FOLLOWER | bolt://core3:7687, http://core3:7474, https://core3:7473 | Open |
| READ_REPLICA | bolt://r1:7687, http://r1:7474, https://r1:7473 | Open |

\$:play start