Getting More Out of Neo4j

v 1.0

Understanding Query Execution

Analyzing Cypher queries - EXPLAIN - 1

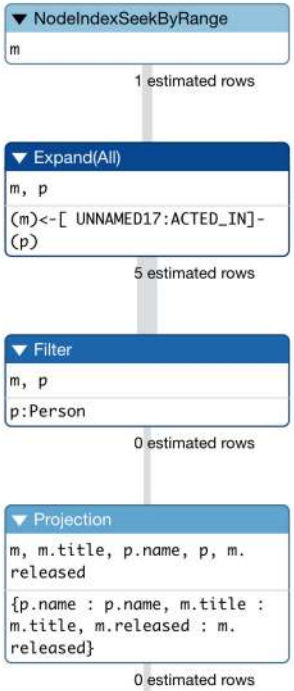
- Provides information about the query plan.
- Does <u>not</u> execute the Cypher statement.

Here is an example where we have set the *$actorName* and *$year* parameters for our session and we execute this Cypher statement to produce the query plan:

```
EXPLAIN MATCH (p:Person)-[:ACTED_IN]->(m:Movie)
WHERE p.name = $actorName AND
      m.released <  $year
RETURN p.name, m.title, m.released
```

Analyzing Cypher queries - EXPLAIN - 2

```
$ EXPLAIN MATCH (p:Person)-[:ACTED_IN]->(m:Movie) WHERE p.name = $actorName AND m.released < $year RE…
```



Analyzing Cypher queries - PROFILE - 1

- Provides information about the query plan.
- Executes the Cypher statement.
- Provides information about db hits.

```
PROFILE MATCH (p:Person)-[:ACTED_IN]->(m:Movie)
WHERE p.name = $actorName AND
      m.released <  $year
RETURN p.name, m.title, m.released
```

Analyzing Cypher queries - PROFILE - 2

Profile query where node labels are specified:

Profile query where node labels <u>not</u> are specified:

Monitoring queries

There are two reasons why a Cypher query may take a long time:

1. The query returns a lot of data. The query completes execution in the graph engine, but it takes a long time to create the result stream to return to the client.

```
MATCH (a)--(b)--(c)--(d)--(e)--(f) RETURN a
```

You should avoid these type of queries! You cannot monitor them.

1. The query takes a long time to execute in the graph engine.

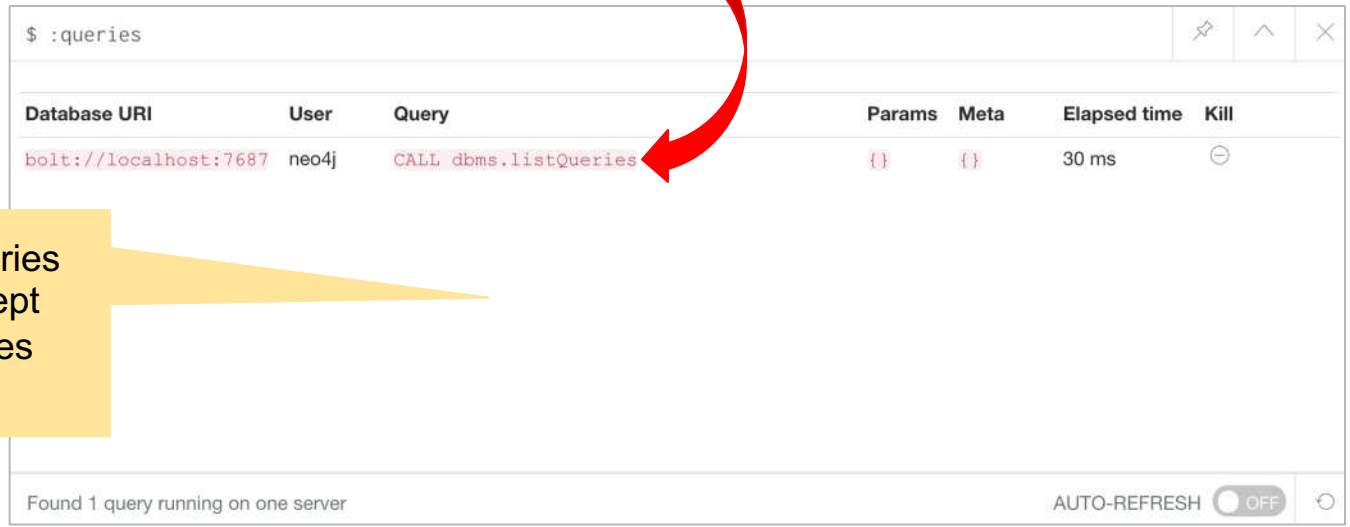
```
MATCH (a), (b), (c), (d), (e) RETURN count(id(a))
```

You can monitor and kill these types of queries.

Viewing running queries

If your query is taking a long time to execute your first have to determine if it is running in the graph engine:

1. Open a new Neo4j Browser session.
2. Execute the **:queries** command.



No other queries running, except for the :queries command.

Handling “rogue” queries

If your query is taking a long time to execute and you cannot monitor it, your options are to:

1. Close the Neo4j Browser session that is stuck and start a new Neo4j Browser session.
2. If that doesn’t work:
   a. On Neo4j Desktop, restart the database.
   b. In Neo4j Sandbox, shut down the sandbox (ouch!). You need to re-create the Sandbox.

Viewing long-running queries

$:queries

Database URI	User	Query	Params	Meta	Elapsed time	Kill
bolt://localhost:7687	neo4j	CALL dbms.listQueries	{}	{}	0 ms	
bolt://localhost:7687	neo4j	match (a), (b), (c), (d), (e) return count (id(a))	{}	{}	55526 ms	

Long-running query

Found 2 queries running on one server

AUTO-REFRESH ON

Killing long-running queries

$:queries

Database URI	User	Query	Params	Meta	Elapsed time	Kill
bolt://localhost:7687	neo4j	match (a), (b), (c), (d), (e) return count (id(a))	{}	{}	135525 ms	
bolt://localhost:7687	neo4j	CALL dbms.listQueries	{}	{}	0 ms	

Found 2 queries running on one server

AUTO-REFRESH ON

Monitoring session

$ match (a), (b), (c), (d), (e) return count (id(a))

Error

ERROR

Neo.TransientError.Transaction.Terminated

Neo.TransientError.Transaction.Terminated: The transaction has been terminated. Retry your operation in a new transaction, and you should see a successful result. Explicitly terminated by the user.

Neo.TransientError.Transaction.Terminated: The transaction has been terminated. Retry your operation in a new transa...

Query session

Cypher Parameters

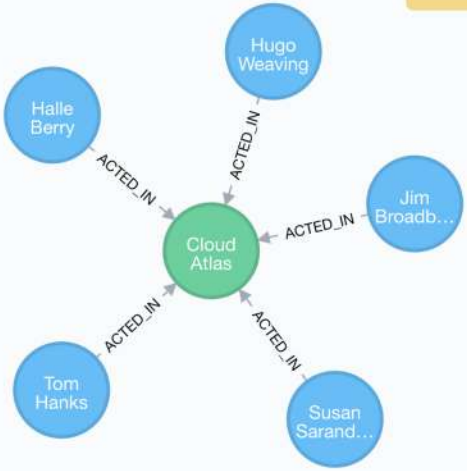
Cypher parameters

```
$ MATCH (p:Person)-[:ACTED_IN]->(m:Movie) WHERE m.title='Cloud Atlas' RETURN p, m
```

Graph | Table | Text | Code

*(6) Person(5) Movie(1)

We do not want this value to be hard-coded in the query.



Displaying 6 nodes, 5 relationships.

Using Cypher parameters - 1

1. Set values for parameters in your Neo4j Browser session before you run the query.

```
$ :param actorName => 'Tom Hanks'
```

```
{
  "actorName": "Tom Hanks"
}
```

See `:help param` for usage of the `:param` command.

Successfully set your parameters.

2. Specify parameters using '$' in your Cypher query.

```
MATCH (p:Person)-[:ACTED_IN]->(m:Movie)
WHERE p.name = $actorName
RETURN m.released, m.title ORDER BY m.released DESC
```

Using Cypher parameters - 2

When this query runs, *$actorName* has a value *Tom Hanks*:

```
$ MATCH (p:Person)-[:ACTED_IN]->(m:Movie) WHERE p.name = $actorName RETURN m.released, m.title O…
```

Table | Text | Code

m.released	m.title
2012	"Cloud Atlas"
2007	"Charlie Wilson's War"
2006	"The Da Vinci Code"
2004	"The Polar Express"
2000	"Cast Away"
1999	"The Green Mile"
1998	"You've Got Mail"
1996	"That Thing You Do"
1995	"Apollo 13"
1994	"Forrest Gump"
1993	"Sleepless in Seattle"
1992	"A League of Their Own"
1990	"Joe Versus the Volcano"

Using Cypher parameters - 3

Change the value of the parameter, *$actorName* to *Tom Cruise*:

```
:param actorName => 'Tom Cruise'
```

Re-run the same query:

```
$ MATCH (p:Person)-[:ACTED_IN]->(m:Movie) WHERE p.name = $actorName RETURN m.released, m.title O…
```

Table

m.released	m.title
2000	"Jerry Maguire"
1992	"A Few Good Men"
1986	"Top Gun"

Text