

RedisGraph quick start tutorial

For this quick start tutorial, you need:

- Either:
 - A Redis Cloud database [with the RedisGraph module](#)
You can [set up a free Redis Cloud database](#) to see the module in action.
 - A Redis Enterprise Software database [with the RedisGraph module](#)
- redis-cli with connectivity to a redis database

Give it a try

After you load RedisGraph, you can interact with it using redis-cli.

Here we'll quickly create a small graph representing a subset of motorcycle riders and teams taking part in the MotoGP league. Once created, we'll start querying our data.

With redis-cli

Connect to redis.

```
$ redis-cli -p 12543
127.0.0.1:12543>
```

Run these commands:

```
127.0.0.1:12543> GRAPH.QUERY MotoGP "CREATE (:Rider {name:'Valentino Rossi'})-[:rides]->
(:Team {name:'Yamaha'}), (:Rider {name:'Dani Pedrosa'})-[:rides]->(:Team {name:'Honda'}),
(:Rider {name:'Andrea Dovizioso'})-[:rides]->(:Team {name:'Ducati'})"
1) (empty list or set)
2) 1) Labels added: 2
   2) Nodes created: 6
   3) Properties set: 6
   4) Relationships created: 3
   5) "Query internal execution time: 0.399000 milliseconds"
```

Now that our MotoGP graph is created, we can start asking questions. For example: Who's riding for team Yamaha?

```
127.0.0.1:12543> GRAPH.QUERY MotoGP "MATCH (r:Rider)-[:rides]->(t:Team) WHERE t.name =
'Yamaha' RETURN r,t"
1) 1) 1) "r.name"
   2) "t.name"
   2) 1) "Valentino Rossi"
   2) "Yamaha"
2) 1) "Query internal execution time: 0.122000 milliseconds"
```

How many riders represent team Ducati?

```
127.0.0.1:12543> GRAPH.QUERY MotoGP "MATCH (r:Rider)-[:rides]->(t:Team {name:'Ducati'})
RETURN count(r)"
1) 1) 1) "count(r)"
   2) 1) "1.000000"
2) 1) "Query internal execution time: 0.129000 milliseconds"
```

Develop with RedisGraph

Before using RedisGraph, you should familiarize yourself with its commands and syntax as detailed in the [commands reference](#).

After you load RedisGraph, you can interact with it using `redis-cli`.

Here we'll quickly create a small graph representing a subset of motorcycle riders and teams taking part in the MotoGP league. Once created, we'll start querying our data.

Use `redis-cli`

```
127.0.0.1:12543> GRAPH.QUERY MotoGP "CREATE (:Rider {name:'Valentino Rossi'})-[:rides]->
(:Team {name:'Yamaha'}), (:Rider {name:'Dani Pedrosa'})-[:rides]->(:Team {name:'Honda'}),
(:Rider {name:'Andrea Dovizioso'})-[:rides]->(:Team {name:'Ducati'})"
1) (empty list or set)
2) 1) Labels added: 2
   2) Nodes created: 6
   3) Properties set: 6
   4) Relationships created: 3
   5) "Query internal execution time: 0.399000 milliseconds"
```

Now that our MotoGP graph is created, we can start asking questions. For example: Who's riding for team Yamaha?

```
127.0.0.1:12543> GRAPH.QUERY MotoGP "MATCH (r:Rider)-[:rides]->(t:Team) WHERE t.name =
'Yamaha' RETURN r,t"
1) 1) 1) "r.name"
   2) "t.name"
   2) 1) "Valentino Rossi"
   2) "Yamaha"
2) 1) "Query internal execution time: 0.122000 milliseconds"
```

How many riders represent team Ducati?

```
127.0.0.1:12543> GRAPH.QUERY MotoGP "MATCH (r:Rider)-[:rides]->(t:Team {name:'Ducati'})
RETURN count(r)"
1) 1) 1) "count(r)"
   2) 1) "1.000000"
2) 1) "Query internal execution time: 0.129000 milliseconds"
```

Use other client

You can interact with RedisGraph using your client's ability to send raw Redis commands. The exact method for doing that depends on your client of choice.

Python example

This code snippet shows how to use RedisGraph with raw Redis commands from Python using [redis-py](#):

```
import redis
```

```
r = redis.StrictRedis()  
reply = r.execute_command('GRAPH.QUERY', 'social', "CREATE (:person {name:'roi', age:33,  
gender:'male', status:'married'})")
```

Client libraries

Some languages have client libraries that provide support for RedisGraph's commands:

Project	Language	License	Author	URL
redisgraph-py	Python	BSD	Redis	GitHub
JRedisGraph	Java	BSD	Redis	GitHub
redisgraph-rb	Ruby	BSD	Redis	GitHub
redisgraph-go	Go	BSD	Redis	GitHub
redisgraph.js	JavaScript	BSD	Redis	GitHub
php-redis-graph	PHP	MIT	KJDev	GitHub