

001-generic-relationship-1

chapter01\$

```
MATCH (n {id:"0eLgWXRdFvQWb5Ch6Hm33Z"})-[:HAS_TRACK]->(t:Track)
-[:ARTIST]->(a:Artist) //this n matches an album
RETURN t.name AS trackName, a.name AS artistName
```

```
1 MATCH (n {id:"0eLgWXRdFvQWb5Ch6Hm33Z"})-[:HAS_TRACK]->(t:Track)
2   -[:ARTIST]->(a:Artist) //this n matches an album
3 RETURN t.name AS trackName, a.name AS artistName
```

	trackName	artistName
1	"Always"	"Bon Jovi"

002-generic-relationship-2

chapter01\$

```
MATCH (t:Track {id: "0kish3Tobj6Wq0we74343q"})<-[:HAS_TRACK]-(n)
RETURN n.name, labels(n)
```

Instance: neo4j://localhost:7687 Database: chapter01 CYPHER 5 User: neo4j

Database information

Nodes (127,932)

* Album Artist Playlist Track User

Relationships (233,263)

* ARTIST HAS_TRACK OWNS SIMILAR

Property keys

id name notSamePosition position samePosition uri

chapter01\$

```
1 MATCH (t:Track {id: "0kish3Tobj6Wq0we74343q"})<-[:HAS_TRACK]-(n)
2 RETURN n.name, labels(n)
```

	n.name	labels(n)
1	"Cross Road"	["Album"]
2	"Road Trip Punt a Fuego"	["Playlist"]

Started streaming 2 rec

003-generic-relationship-3

chapter01\$

```
MATCH (t:Track {id:"15vzANxN8G9wWfwAJLLMCg"})<-[:HAS_TRACK]-(a:Album)
RETURN t,a
```

Instance: neo4j://localhost:7687 Database: chapter01 CYPHER 5 User: neo4j

Database information

Nodes (127,932)

Album Artist Playlist Track User

Relationships (233,263)

ARTIST HAS_TRACK OWNS SIMILAR

Property keys

id name notSamePosition position samePosition uri

chapter01\$

```
1 MATCH (t:Track {id: "0kish3Tobj6Wq0we74343q"})<-[:HAS_TRACK]-(n)
2 RETURN n.name, labels(n)
```

Table RAW

	n.name	labels(n)
1	"Cross Road"	["Album"]
2	"Road Trip Punt a Fuego"	["Playlist"]

Started s

004-find-similar

chapter01\$

```
MATCH (a:Playlist)-[:SIMILAR]-(b:Playlist)
RETURN a,b
```

Instance: neo4j://localhost:7687 Database: chapter01 CYPHER 5 User: neo4j

Database information

Nodes (127,932)

Album Artist Playlist Track User

Relationships (233,263)

ARTIST HAS_TRACK OWNS SIMILAR

Property keys

id name notSamePosition position samePosition uri

chapter01\$

```
1 MATCH (a:Playlist)-[:SIMILAR]-(b:Playlist)
2 RETURN a,b
```

Graph Table RAW

Results overview

Nodes (82)

* (82) Playlist (82)

Started streaming 616 records after 2 ms at

005-double-traversal

chapter01\$

```
MATCH path=(track1:Track {id:"0BB9eUBBaaX6GALSYNcEp7"})-[*3]-(track2:Track {id:"2KmEgiY8fQs0G6WNxtzQKr"})
RETURN path
```

- `[*]` means "any number of relationships/hops"
- `[*3]` means "exactly 3 relationships/hops"
- So the pattern will match paths where there are exactly 3 relationships between `track1` and `track2`

What does this query do:

- Finds all paths that connect these two specific tracks
- Each path must have exactly 3 relationships (edges) between them
- The path could go through other nodes (likely other tracks, artists, albums, etc.)

Other examples of variable-length relationships:

- `[*1..3]` = 1 to 3 hops
- `[*..5]` = up to 5 hops
- `[*3..]` = 3 or more hops
- `[*]` = any number of hops

In your music graph context, this might find connections like:

- Track1 → Artist → Album → Track2
- Track1 → Genre → Playlist → Track2
- Or any other 3-step path through the graph structure

The screenshot displays the Neo4j Cypher query interface. On the left, the 'Database information' panel shows 127,932 nodes and 233,263 relationships, with property keys like 'id', 'name', 'notSamePosition', 'position', 'samePosition', and 'uri'. The main query editor shows the following query:

```
chapter01$
1 MATCH path=(track1:Track {id:"0BB9eUBBaaX6GALSYNcEp7"})-[*3]-(track2:Track {id:"2KmEgiY8fQs0G6WNxtzQKr"})
2 RETURN path
```

The query is executed, and the results are shown in the 'Graph' view. The graph illustrates a path of 3 relationships between two tracks. The nodes involved are 'Rockin' die/Id', 'Big Bear', 'Hotline Miami...', 'You Spin Me R...', 'Mac Electro. Fu...', and 'Baba O'Riley'. The relationships are labeled 'HAS_TRACK', 'SIMILAR', and 'HAS_TRACK'. The 'Results overview' panel on the right shows 7 nodes and 15 relationships, with a breakdown of 7 playlists and 2 tracks.

Started streaming 10 records after 1 ms and complet

006-drop-similar

chapter01\$

```
MATCH ()-[r:SIMILAR]-()
DELETE r
```

Instance: neo4j://localhost:7687 Database: chapter01 CYPHER 5 User: neo4j Claude

Database information

Nodes (127,932)

- Album
- Artist
- Playlist
- Track
- User

Relationships (232,955)

chapter01\$

```
chapter01$ MATCH ()-[r:SIMILAR]-() DELETE r
```

Deleted 308 relationships

007-single-similar-rel

chapter01\$

```
MATCH path=(p:Playlist)-[r1:HAS_TRACK]->(track)-[r2:HAS_TRACK]-(other:Playlist)
WITH p AS playlistLeft, other AS playlistRight, collect({track: track, positionLeft: r1.position, positionRight: r2.position}) AS commonTracks
WHERE size(commonTracks) > 5
WITH playlistLeft, playlistRight, size([track in commonTracks WHERE track.positionLeft = track.positionRight]) AS tracksWithSamePosition,
size([track in commonTracks WHERE NOT track.positionLeft = track.positionRight]) AS tracksAtDifferentPosition
MERGE (playlistLeft)-[r:SIMILAR]-(playlistRight)
SET r.samePosition = tracksWithSamePosition, r.notSamePosition = tracksAtDifferentPosition
```

Instance: neo4j://localhost:7687 Database: chapter01 CYPHER 5 User: neo4j Go b2

Database information

Nodes (127,932)

- Album
- Artist
- Playlist
- Track
- User

Relationships (233,109)

- ARTIST
- HAS_TRACK
- OWNS
- SIMILAR

Property keys

- id
- name
- notSamePosition
- position
- samePosition
- uri

chapter01\$

```
1 MATCH path=(p:Playlist)-[r1:HAS_TRACK]->(track)-[r2:HAS_TRACK]-(other:Playlist)
2 WITH p AS playlistLeft, other AS playlistRight,
3   collect({track: track, positionLeft: r1.position, positionRight: r2.position}) AS commonTracks
4   WHERE size(commonTracks) > 5
5 WITH playlistLeft, playlistRight,
6   size([track in commonTracks WHERE track.positionLeft = track.positionRight]) AS
7   tracksWithSamePosition,
8   size([track in commonTracks WHERE NOT track.positionLeft = track.positionRight]) AS
9   tracksAtDifferentPosition
10 MERGE (playlistLeft)-[r:SIMILAR]-(playlistRight)
11 SET r.samePosition = tracksWithSamePosition, r.notSamePosition = tracksAtDifferentPosition
```

Created 154 relationships, set 616 properties

Comp