Neo4j Cypher Query Guide

A comprehensive cheat sheet for querying Neo4j graph databases using Cypher.

📌 Table of Contents

---

1. Basic Queries

Create Nodes

```cypher
CREATE (:Person {name: "Alice", age: 30})
CREATE (:Movie {title: "Inception", year: 2010})
```

Create Relationships

```cypher
MATCH (a:Person {name: "Alice"}), (m:Movie {title: "Inception"})
CREATE (a)-[:WATCHED {rating: 5}]->(m)
```

Query All Nodes

```cypher
MATCH (n) RETURN n
```

Query Specific Nodes

```cypher

```

```cypher
MATCH (p:Person) RETURN p.name
```

---

2. Filtering & Sorting

WHERE Clauses

cypher Copy

```cypher
MATCH (p:Person)
WHERE p.age > 25
RETURN p.name
```

Regex & String Matching

cypher Copy

```cypher
MATCH (p:Person)
WHERE p.name =~ 'A.*'  // Starts with 'A'
RETURN p.name
```

Sorting (ORDER BY)

cypher Copy

```cypher
MATCH (p:Person)
RETURN p.name, p.age
ORDER BY p.age DESC
```

LIMIT & SKIP

cypher Copy

```cypher
MATCH (p:Person)
RETURN p.name
LIMIT 5
```

---

3. Relationships & Paths

Find Relationships

cypher Copy

```cypher

```

```cypher
MATCH (p:Person)-[r:WATCHED]->(m:Movie)
RETURN p.name, r.rating, m.title
```

Variable-Length Paths

cypher Copy

```cypher
MATCH (p:Person)-[:FRIENDS_WITH*1..3]->(friend)
RETURN p.name, friend.name
```

Shortest Path

cypher Copy

```cypher
MATCH (a:Person {name: "Alice"}), (b:Person {name: "Bob"})
MATCH path = shortestPath((a)-[*]-(b))
RETURN path
```

---

4. Aggregations & Grouping

COUNT, SUM, AVG

cypher Copy

```cypher
MATCH (p:Person)
RETURN COUNT(p) AS total_people, AVG(p.age) AS avg_age
```

GROUP BY

cypher Copy

```cypher
MATCH (p:Person)-[:WATCHED]->(m:Movie)
RETURN m.title, COUNT(p) AS viewers
```

COLLECT (Group into List)

cypher Copy

```cypher
MATCH (p:Person)-[:WATCHED]->(m:Movie)
RETURN p.name, COLLECT(m.title) AS movies_watched
```

---

5. Advanced Queries

OPTIONAL MATCH (Left Join)

```cypher
MATCH (p:Person)
OPTIONAL MATCH (p)-[:WATCHED]->(m:Movie)
RETURN p.name, m.title
```

UNION (Combine Results)

```cypher
MATCH (p:Person) RETURN p.name AS name
UNION
MATCH (m:Movie) RETURN m.title AS name
```

FOREACH (Batch Updates)

```cypher
MATCH (p:Person)
FOREACH (ignore IN CASE WHEN p.age > 30 THEN [1] ELSE [] END |
  SET p.senior = true
)
```

---

6. Tips & Best Practices

✅ Indexing for Performance

```cypher
CREATE INDEX FOR (p:Person) ON (p.name)
DROP INDEX FOR (p:Person) ON (p.name)
```

✅ **Parameterized Queries** (Prevent injection)

```cypher
:param name => "Alice"
MATCH (p:Person {name: $name}) RETURN p
```

✅ Profile Query Performance

```cypher
```

```
EXPLAIN MATCH (p:Person) RETURN p  // Estimated execution plan
PROFILE MATCH (p:Person) RETURN p // Actual runtime metrics
```

🚀 **Pro Tip**: Use **APOC** library for advanced functions:

cypher Copy

```
CALL apoc.help('dijkstra')  // Find shortest weighted path
```

📚 Resources

- Official Cypher Manual
- Neo4j Sandbox (Free practice environment)
- APOC Library Guide