

Neo4j Cypher Relationship Types and Symbols

1 ■■■ Core Syntax Elements

| Concept | Cypher Symbol | Description | Example |
|------------------------------|-------------------------|--|-----------------------------|
| Node | (n) | Represents an entity (circle in graph) | (a:Person) |
| Relationship | -[r]-> | Directed connection (arrow in graph) | (a)-[:KNOWS]->(b) |
| Undirected Relationship | -[r]- | Matches in either direction | (a)-[:FRIEND]-(:Person) |
| Incoming Relationship | <-[r]- | Points toward the current node | (a)<[:-LIKES]-(b) |
| Relationship Type | :TYPE | Label defining edge meaning | [:ACTED_IN] |
| Multiple Relationship Types | [:TYPE1 TYPE2] | Matches any listed type | (a)-[:LIKES HATES]->(b) |
| Variable for Relationship | -[r:TYPE]-> | Assigns variable name | (a)-[r:LIKES]->(b) |
| Relationship with Properties | -[r:TYPE {key:value}]-> | Adds attributes | (a)-[:RATED {stars:5}]->(b) |
| Anonymous Relationship | --> | Type and variable omitted | (a)-->(b) |
| Any Relationship | -[*]- | Matches any relationship | (a)-[*]->(b) |
| Variable-Length Path | -[*1..3]-> | Path between 1 and 3 relationships | (a)-[:KNOWS*1..3]->(b) |

2 ■■■ Relationship Type Rules

- Relationship names (types) are always uppercase by convention.
- Each relationship has exactly one type.
- Direction matters unless omitted (use -[:TYPE]- for both directions).
- Nodes can have multiple labels, but relationships cannot.

3 ■■■ Common Relationship Categories

| Category | Example Type | Meaning | Example Pattern |
|---------------|--------------|--------------------------|--|
| Social | FRIENDS_WITH | People connecting | (a:User)-[:FRIENDS_WITH]->(b:User) |
| Professional | WORKS_FOR | Employment structure | (a:Person)-[:WORKS_FOR]->(c:Company) |
| Creative | ACTED_IN | Creative roles | (a:Actor)-[:ACTED_IN]->(m:Movie) |
| Ownership | OWNS | Hierarchy or containment | (a:Company)-[:OWNS]->(b:Subsidiary) |
| Transactional | PURCHASED | Transactions | (a:Customer)-[:PURCHASED]->(p:Product) |

4 ■■■ Relationship Functions

| Function | Purpose | Example |
|--------------|-------------------------------------|---------------------|
| type(r) | Returns relationship type as string | RETURN type(r) |
| startNode(r) | Returns the starting node | RETURN startNode(r) |
| endNode(r) | Returns the ending node | RETURN endNode(r) |
| id(r) | Returns internal relationship ID | RETURN id(r) |

5 ■■■ Visual Symbols Summary

| Symbol | Meaning | Example |
|---------|------------------------------------|-------------------------------------|
| () | Node (circle) | (p:Person) |
| -[]- | Relationship edge | -[r:KNOWS]- |
| -> / <- | Direction of relationship | (a)-[:LIKES]->(b) |
| {} | Properties on node or relationship | (a)-[:RATED {stars:5}]->(b) |
| : | Label or Type indicator | (a:Person)-[:WORKS_AT]->(c:Company) |
| | OR operator for multiple types | (a)-[:A B]->(b) |
| * | Variable-length path | (a)-[:KNOWS*1..3]->(b) |