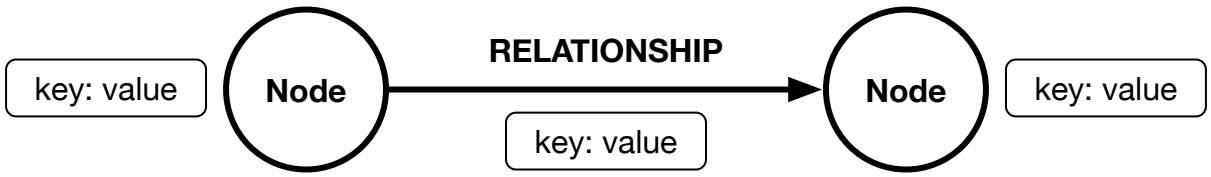
Friday, 3 October 2025

Cypher - Graph Query Language

Characteristics

- Declarative - what data you wanna have, not how
- (node)-[:RELATIONSHIP]->(node)
- (node {**key: value**})-[:RELATIONSHIP]->(node)
- (node {**key: value**})-[:RELATIONSHIP*..2]->(node) -> 2 relations deep …
- Focus on domain, not the database
- Idiomatic drivers for different languages
- Cypher is row-based (like SQL).
- Every **clause** (MATCH, UNWIND, WITH, etc.) produces a stream of rows
- Every **following clause** (CREATE, RETURN, SET …) is executed once per row. This is very fundamental. It is the Neo4J pipeline model. If a clause produces zero rows, a following clause like CREATE is executed zero times



Samples I

UNWIND [

'Amsterdam','Rotterdam','Utrecht','Den Haag','Alkmaar',

'Goes','Veere','Leiden','Delft','Hengelo',

'Enschede','Assen','Maastricht','Zwolle','Eindhoven'

] **AS** city

UNWIND is a clause that produces 15 rows (15 cities). In python we would say: oh it returns an array of cities.

CREATE(:City {name: city});

CREATE is a following clause, that is executed for as many rows the clause produced. In this case, 15 times. :City is a node label (kind). Name is a property of city. City could have a second or third property. E.g. country. By convention, node labels start with a capital, therefore :City and not :city

Samples II

So in the sample below, we only have following clauses. Therefore Rotterdam and Utrecht are created (or recreated if they already existed). This approach leads to duplicate nodes.

```
CREATE (a:City {name:'Rotterdam'})
CREATE (b:City {name:'Utrecht'})
CREATE (a)-[:ROAD {distance:65, maxSpeed:120}]->(b);
```

So here, first a matching clause is executed. If there is a binding for a and a binding for b, there will be a result row for the CREATE clause. Only then, the ROAD relationship will be created.

> The convention is that node labels (:City) are written with a starting capital and relationship labels (:ROAD) are written in capitals. Just eye-candy.

```
MATCH (a:City {name:'Amsterdam'}), (b:City {name:'Alkmaar'})
CREATE (a)-[:ROAD {distance:40}]->(b),
       (b)-[:ROAD {distance:40}]->(a);
```