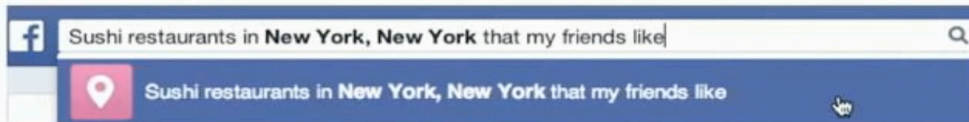


# Cypher

- [cypher relationships](#)

A pattern matching query language made for graphs

- [Declarative](#)
- [Expressive](#)
- [Pattern Matching](#)



```
MATCH (person:Person)-[:IS_FRIEND_OF]->(friend),
        (friend)-[:LIKES]->(restaurant),
        (restaurant)-[:LOCATED_IN]->(loc:Location),
        (restaurant)-[:SERVES]->(type:Cuisine)
WHERE person.name = 'Philip'
AND loc.location='New York'
AND type.cuisine='Sushi'
RETURN restaurant.name
```

```
MATCH (b:Brand) WHERE NOT(b) - [] -> () RETURN b
```

## CREATE

```
CREATE (j:Person {name: 'Jennifer'}) RETURN j
```

## UPDATE

```
MATCH (p:Person {name: 'Jennifer'})
SET p.birthdate = date('1980-01-01')
```

## RELATION

```
CREATE (j:Person {name: 'Jennifer'})-[rel:IS_FRIENDS_WITH]->(m:Person {name: 'Mark'})
```

## MATCH

```
MATCH (b:Brand {name:"Chanel"}) RETURN b
```

```
MATCH (p:Person {name: 'Jennifer'}) RETURN p
```

## INDEXING

```
CREATE INDEX ON :Product(productName);  
CREATE INDEX ON :Product(unitPrice);
```

## FILTER

```
MATCH (p:Product)  
WHERE p.productName IN ['Chocolade','Chai']  
RETURN p.productName, p.unitPrice;
```

## CONSTRAINT

```
CREATE CONSTRAINT ON (n:<LabelName>) ASSERT n.<propertyKey> IS UNIQUE
```

multiple column

```
CREATE CONSTRAINT ON (n:Person) ASSERT (n.firstname, n.surname) IS NODE KEY
```

```
SHOW CONSTRAINTS
```

neo4j\$ SHOW CONSTRAINTS

	id	name	type	entityType	labelsOrTypes	properties	ownedIndex
1	4	"constraint_2c80c3bc"	"NODE_KEY"	"NODE"	["aPerson"]	["a"]	3
2	6	"constraint_9e5764ed"	"UNIQUENESS"	"NODE"	["Brand"]	["name"]	5

## DELETE

```
MATCH (n:Person {name: 'UNKNOWN'})
DELETE n
```

## Delete all nodes and relationships

```
MATCH (n)
DETACH DELETE n
```

## Delete a node with all its relationships

```
MATCH (n {name: 'Andy'})
DETACH DELETE n
```

## Docs

- [Cypher Query](#)
- [Extend Cypher](#)
- [Graph Model](#)

## Plugin Backlinks:

From:  
<https://jace.link/> - **Various Ways**

Permanent link:  
<https://jace.link/open/cypher>

Last update: **2021/12/23 07:40**



