

Graph Database Queries – Cypher Quick Reference

LANGUAGES: CYPHER VS GQL VS GREMLIN

SQL is designed to query tables in relational databases.

Cypher is designed to adapt SQL-like capabilities to a graph database. It is developed and promoted by Neo4j, currently the most widely used graph database.

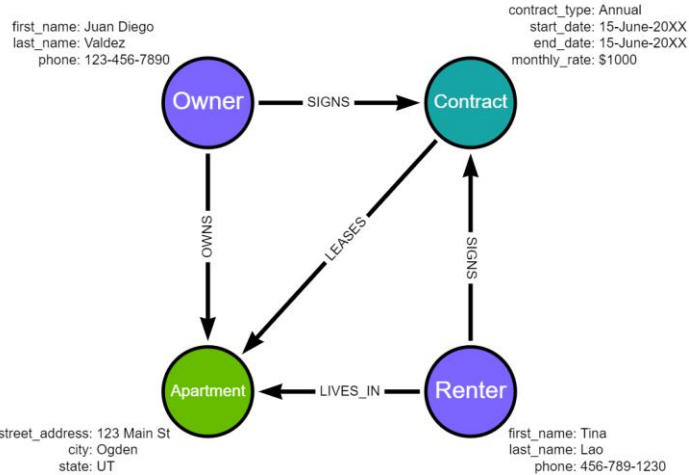
Gremlin is a generic graph query language, developed as an Apache project. It supports several vendors, including Microsoft Cosmos DB and JanusGraph.

GQL is intended to be a standard, unified graph query language, but it is a *work in progress*. Not yet released.

GraphQL is an API query language based in JavaScript.

Note: **other languages** include AQL, SPARQL, PGQL, and GSQL.

SAMPLE GRAPH



PATTERN SEARCH

Match Node Type	<code>MATCH (r:Renter)</code>
Match Pattern	<code>MATCH (o:Owner)-[:OWNS]->(a:Apartment)</code>
Optional Match	<code>MATCH (a:Apartment) OPTIONAL MATCH (c:Contract)-[:LEASES]->(a)</code>

Note: the letter before each node type label is an **alias**, used to reference it later.

TRANSFORMING

Convert Nodes to a List	<code>MATCH (r:Renter) COLLECT (r) as renter_list</code>
Combine and Clean Properties	<code>MATCH (o:Owner) WITH o.first_name+' '+o.last_name AS full_name, REPLACE(o.phone,'-',',')</code>

FILTERING

In-Line Filter	<code>MATCH (r:Renter {last_name:'Lao'})</code>
Where Filter	<code>MATCH (r:Renter) WHERE r.last_name = 'Lao'</code>
Filter on List	<code>WHERE r.city IN ['Ogden','Detroit','Chicago']</code>
Missing Property	<code>WHERE r.phone IS NULL</code>
Wildcard Filter	<code>WHERE r.phone CONTAINS '234'</code>

RETURNING A GRAPH

Return Specific Node Types	<code>MATCH (o:Owner)-[:OWNS]->(a:Apartment) WHERE a.city = 'Detroit' RETURN o;</code>
Return the Full Path	<code>MATCH path=(o:Owner)-[:SIGNS]->(c:Contract) WHERE c.monthly_rate > 900 RETURN path;</code>

RETURNING ROWS AND COLUMNS

Return Specific Properties	<code>MATCH p=(o:Owner)-[:SIGNS]->(c:Contract)-[:LEASES]->(a:Apartment) WHERE c.monthly_rate > 900 RETURN a.city, c.start_date;</code>
Sort and Limit Results	<code>MATCH (c:Contract) RETURN c.start_date, c.monthly_rate ORDER BY c.start_date DESC LIMIT 10;</code>

AGGREGATING

Basic Aggregates	<code>MATCH (c:Contract)-[:LEASES]->(a:Apartment) RETURN a.city, SUM(c.monthly_rate) AS total, AVG(c.monthly_rate) AS avg_rate</code>
Filter Based on Aggregate	<code>MATCH (c:Contract)-[:LEASES]->(a:Apartment) WITH a.city, AVG(c.monthly_rate) AS avg_rate WHERE avg_rate > 1000 RETURN a.city, avg_rate;</code>

Note: the **GROUP BY** command is not required in Cypher.