



Graph Powered NLP using Google PaLM 2, Neo4j and Python

Siddhant Agarwal

Hi, I'm Sid 🖐️

Developer Relations, APAC at Neo4j

Hi, I'm Sid 🙌

Developer Relations, APAC at Neo4j

Started my career with IBM as Java Technology Engineer

Worked with Google, Beahead, Open in the past

Passion for Innovation, Design Thinking, Startup Ecosystem

10+ years in Developer Relations & Community Building

GDG Cloud Bengaluru Organizer, GFSA Mentor, ACM Distinguished Speaker



Scan me

Agenda

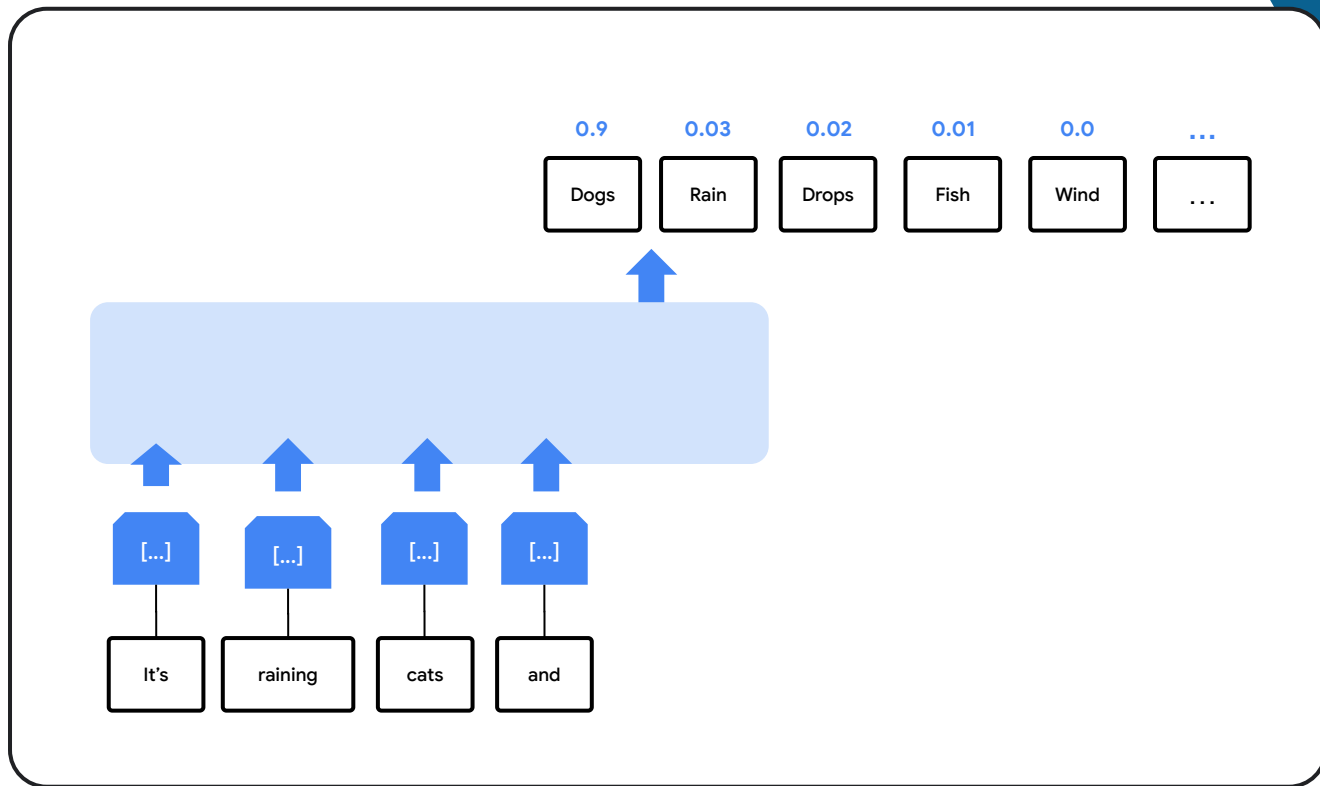
1. What is an LLM?
2. What, Why and How of a Graph Database?
3. Intro to Google MakerSuite and PaLM 2 API
4. Building things together



LLMs Explained



What is an LLM?

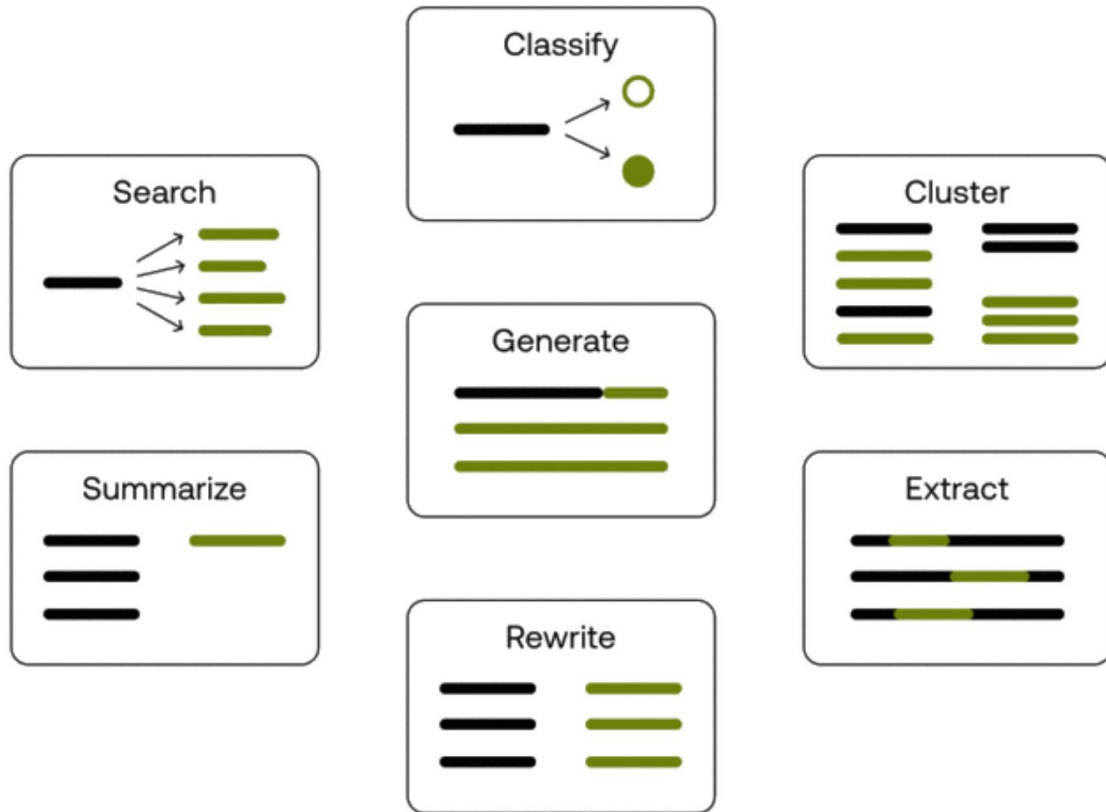


Large Language Models or LLMs are advanced machine learning models designed to understand and generate human-like text.

These models are built upon deep learning techniques and neural networks that process vast amounts of text data to learn grammar, syntax, context, and semantics.

What sets them apart is their massive scale, with millions (or even billions) of parameters that enable them to grasp intricate linguistic patterns and nuances.

LLMs Explained



Roses are red,

Roses are red,
Violets are blue,
Sugar is sweet,

```
for(var i = 0
```

```
for(var i = 0; i < 10; i++) {
```

User:

Explain this joke: “I used to be addicted to soap, but I'm clean now.”

Model:

Model: The joke is a pun on the word "clean". It has two meanings:

To be free of dirt or other unwanted substances.

To be free of addiction.

The joke is funny because it is unexpected. The listener expects the speaker to be talking about the first meaning of "clean", but they are actually talking about the second meaning. This creates a surprise and a laugh.

User:

Give me five neat ideas for science fair projects.

Model:

1. **Build a volcano.**

This is a classic science fair project that is both fun and educational. You can learn about the science of volcanoes, earthquakes, and plate tectonics.

2. **Make a solar oven.**

This is a great project to learn about renewable energy and how to harness the power of the sun. You can use your solar oven to cook food, dry fruit, or even melt chocolate.

3. **Create a 3D printer.**

This is a more challenging project, but it is also very rewarding. You can learn about 3D printing technology and design your own custom objects.

4. **Build a robot.**

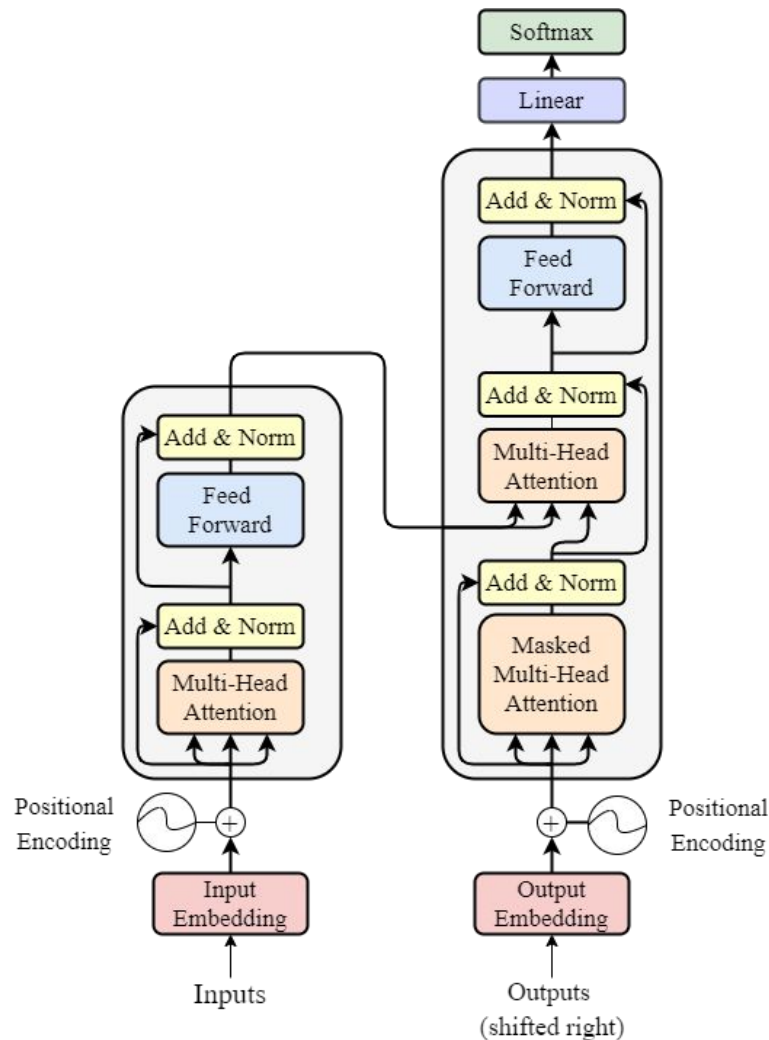
This is a great project for those who are interested in engineering and robotics. You can learn about basic electronics, mechanics, and programming.

5. **Design a water filtration system.**

This is a project that can help you learn about the importance of clean water and how to filter it. You can use your water filtration system to clean water from a stream or river.

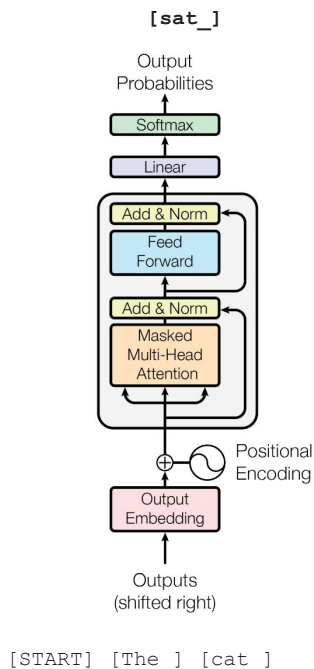
LLMs let us
prototype *fast*.

Transformers first outlined in a 2017 paper published by Google called “Attention Is All You Need”, utilize a self-attention mechanism to solve various sequence-to-sequence tasks like language translation and text generation.



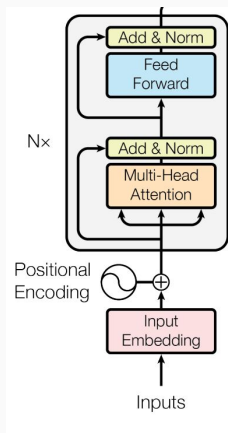
Attention Is All You Need - Summary and results

Decoder-only GPT



Encoder-only BERT

`[*] [*] [sat_] [*] [the_] [*]`

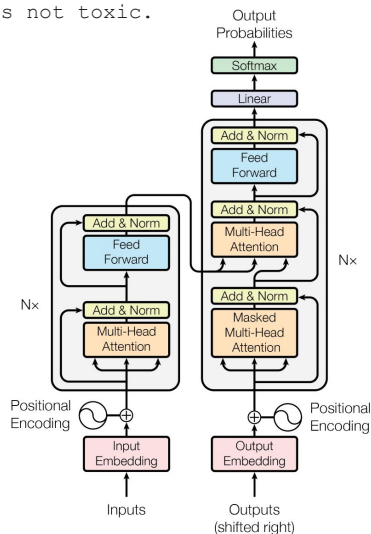


Enc-Dec T5

Das ist gut.

A storm in Attala caused 6 victims.

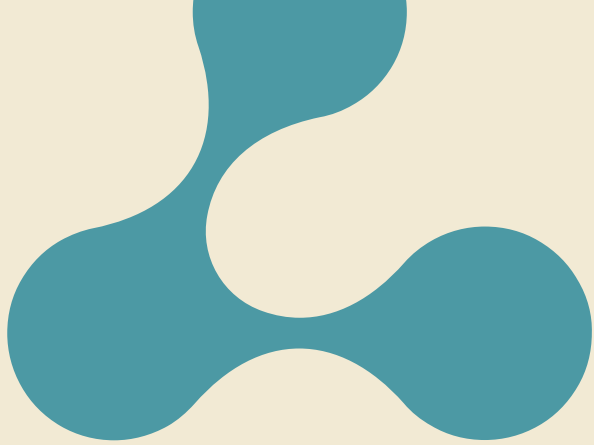
This is not toxic.



Translate EN-DE: This is good.

Summarize: state authorities dispatched...

Is this toxic: You look beautiful today!

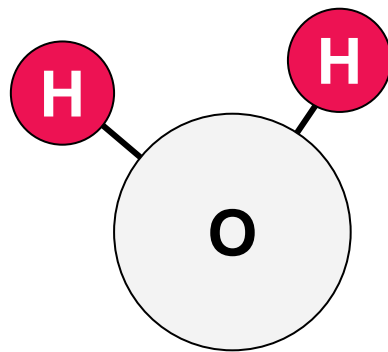
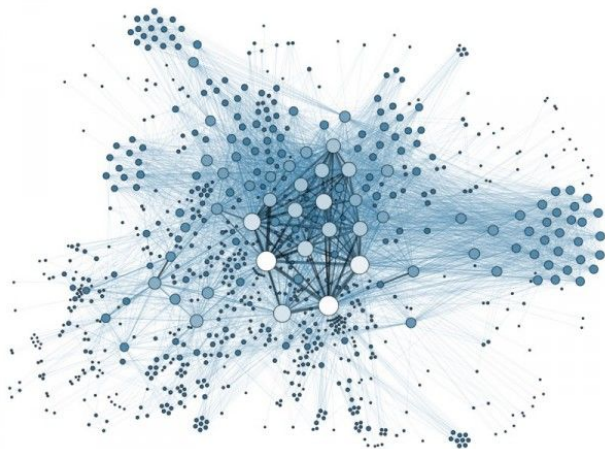


Graphs Explained

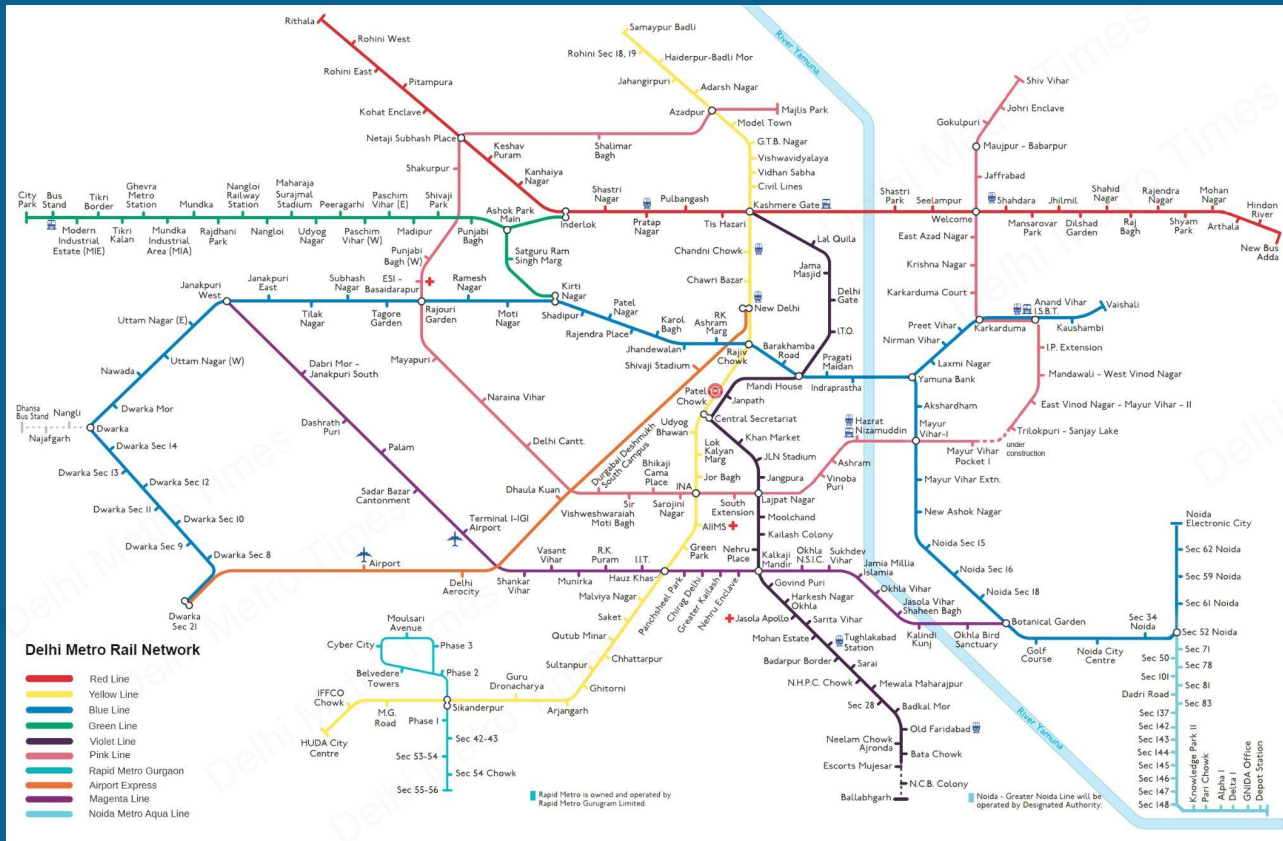


What is a Graph?

A Graph is a set of discrete entities, each of which has some set of relationships with the other entities



Graphs Explained



Why Graphs?

2013+

Graph Database

Flexible, scale-out store/retrieve + analytics

Optimized for data connections, real-time transactions and analytics

Natively store & query relationships

Queries run 1000x faster at scale, up to 1000+ hops

First database language standard in 40 years.

2005+

NoSQL Database

Scale-out, simple model store/retrieve

Optimized for application-driven, volume throughput

App
Based

1980+

Relational Database

Structured, schema-based store/retrieve

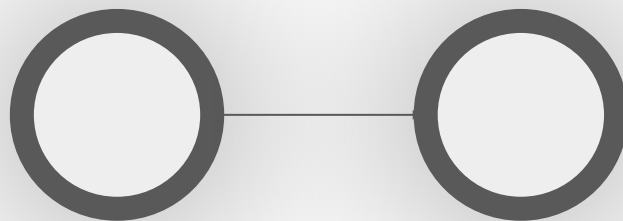
Optimized for real-time transactional processing/reporting

SQL

Core of Graph Theory



Core of Graph Theory: Nodes

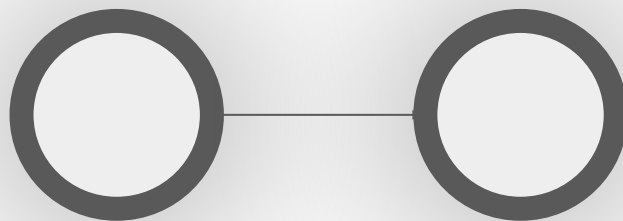


Nodes / Vertices

Core features of Neo4j: Nodes

Nodes become useful because:

- They hold data in the form of **properties**
- They have **labels** to define an optional schema

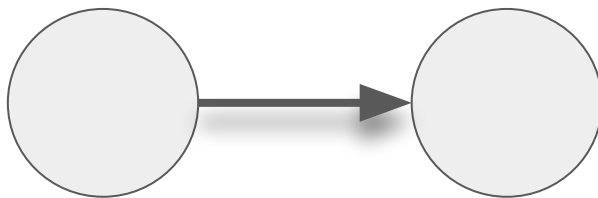


Core features of Neo4j: Nodes

Nodes are Nouns

- Which actors play in this movie?
- How many movies did this director direct?
- Which users rated movies with more 4.0/5?

Core of Graph Theory: Relationships

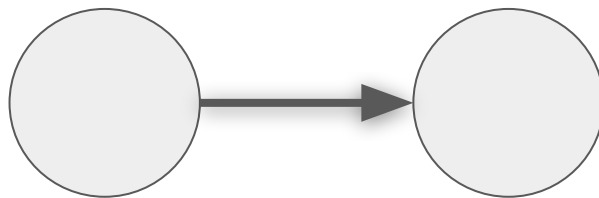


Relationship / Edge

Core features of Neo4j: Relationships

Relationships become useful because:

- They hold data in the form of **properties**
- They have **types** to define an optional schema
- They **connect** nodes explicitly
- Allow for a **direction**



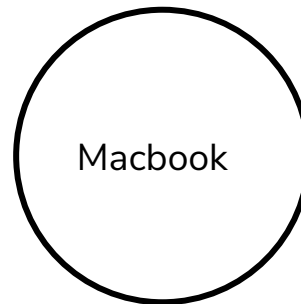
Core features of Neo4j: Relationships

Relationships are verbs

- Which actors play in this movie?
- How many movies did this director direct?
- Which users rated movies with more 4.0/5?

Graph Database Fundamentals

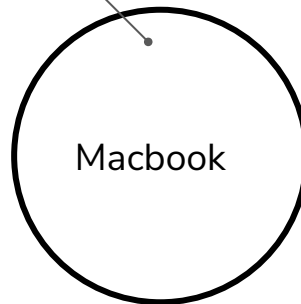
Nodes



Graph Database Fundamentals

Nodes

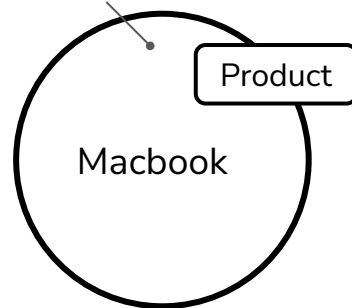
Nodes represent *things*



Graph Database Fundamentals

Nodes

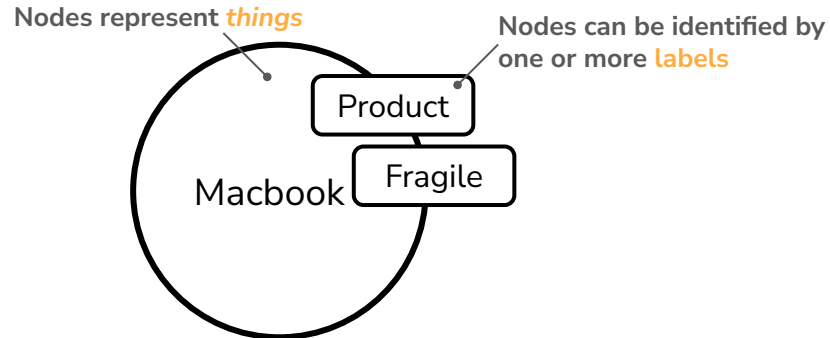
Nodes represent *things*



Nodes can be identified by one or more *labels*

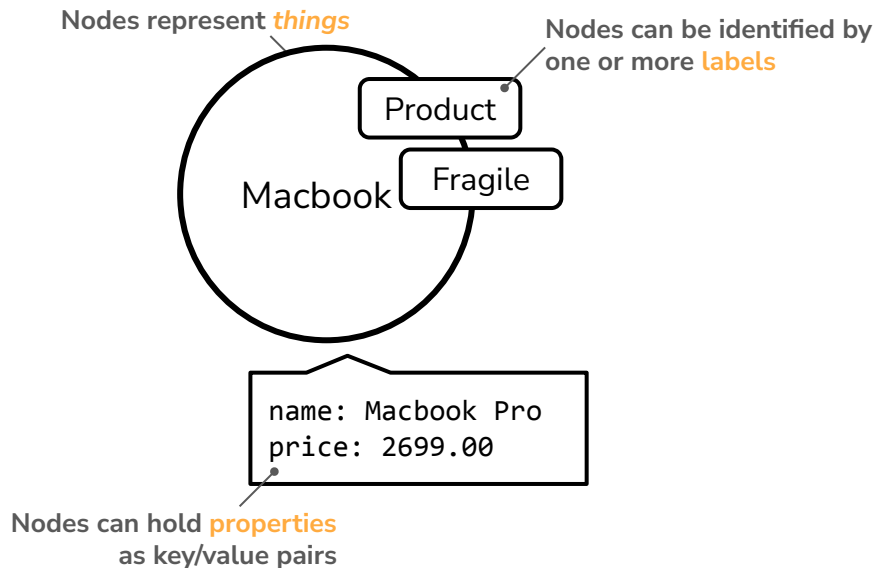
Graph Database Fundamentals

Nodes



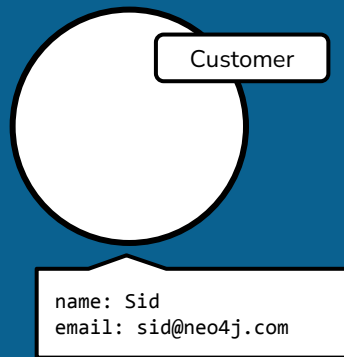
Graph Database Fundamentals

Nodes



Graph Database Fundamentals

Relationships



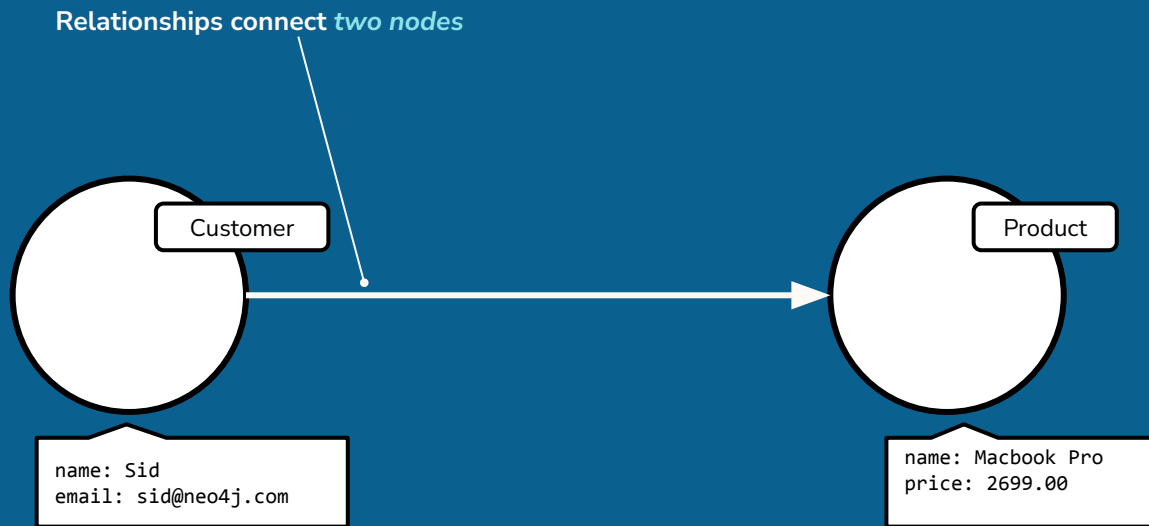
Graph Database Fundamentals

Relationships



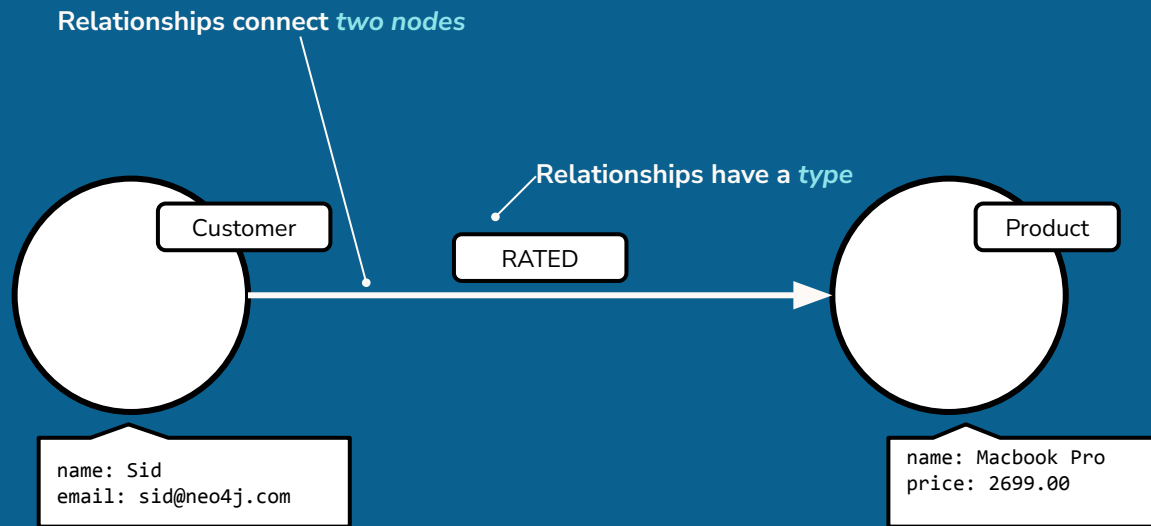
Graph Database Fundamentals

Relationships



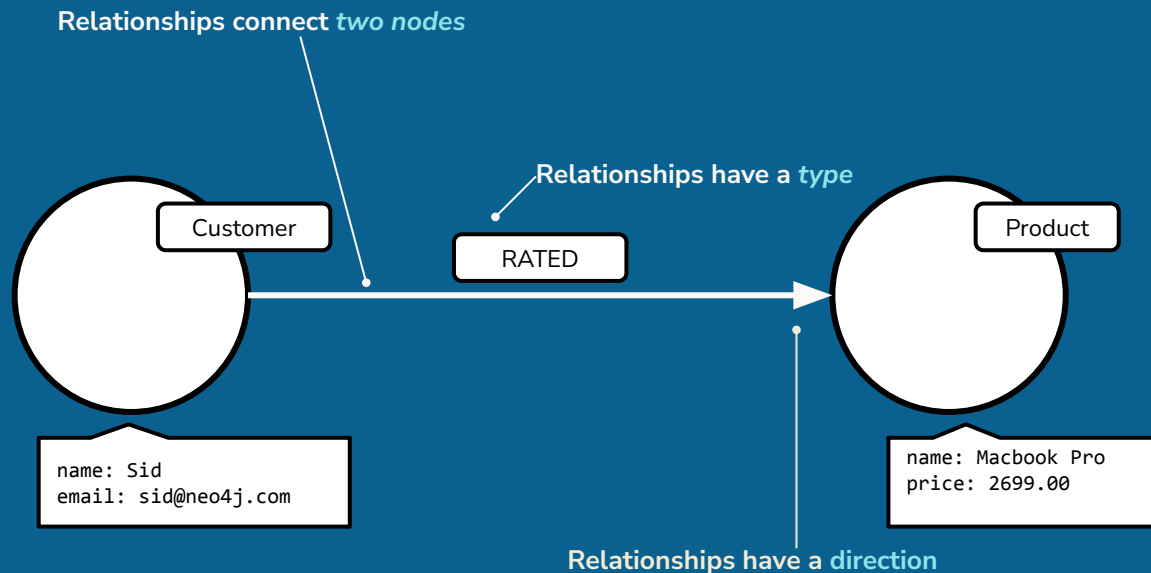
Graph Database Fundamentals

Relationships



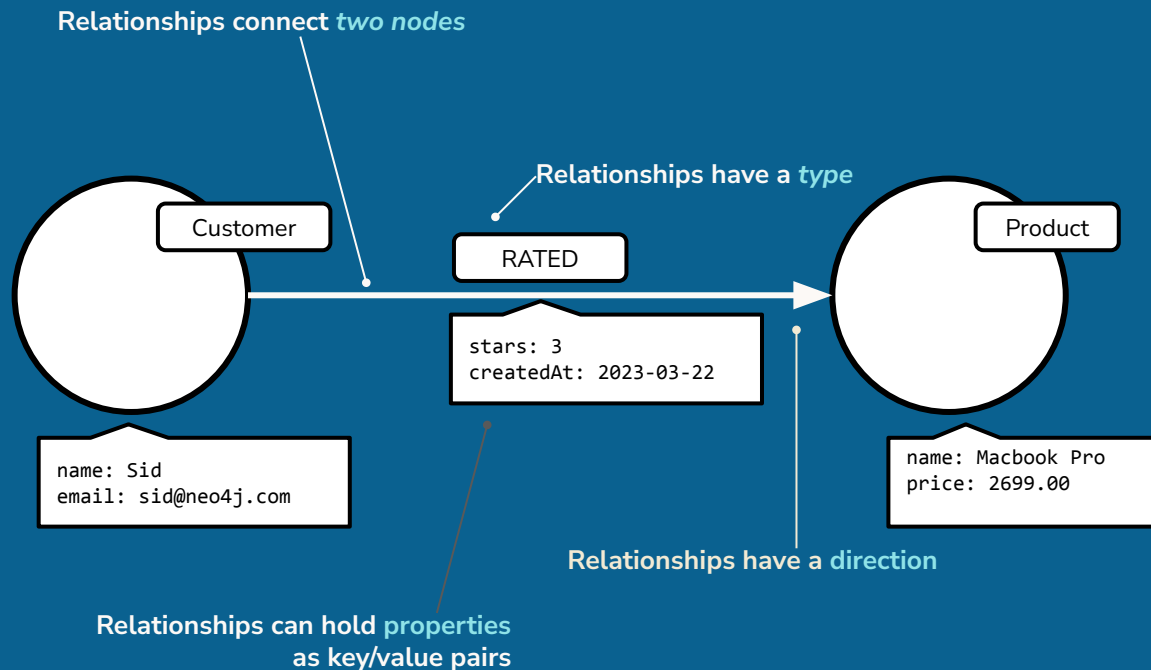
Graph Database Fundamentals

Relationships

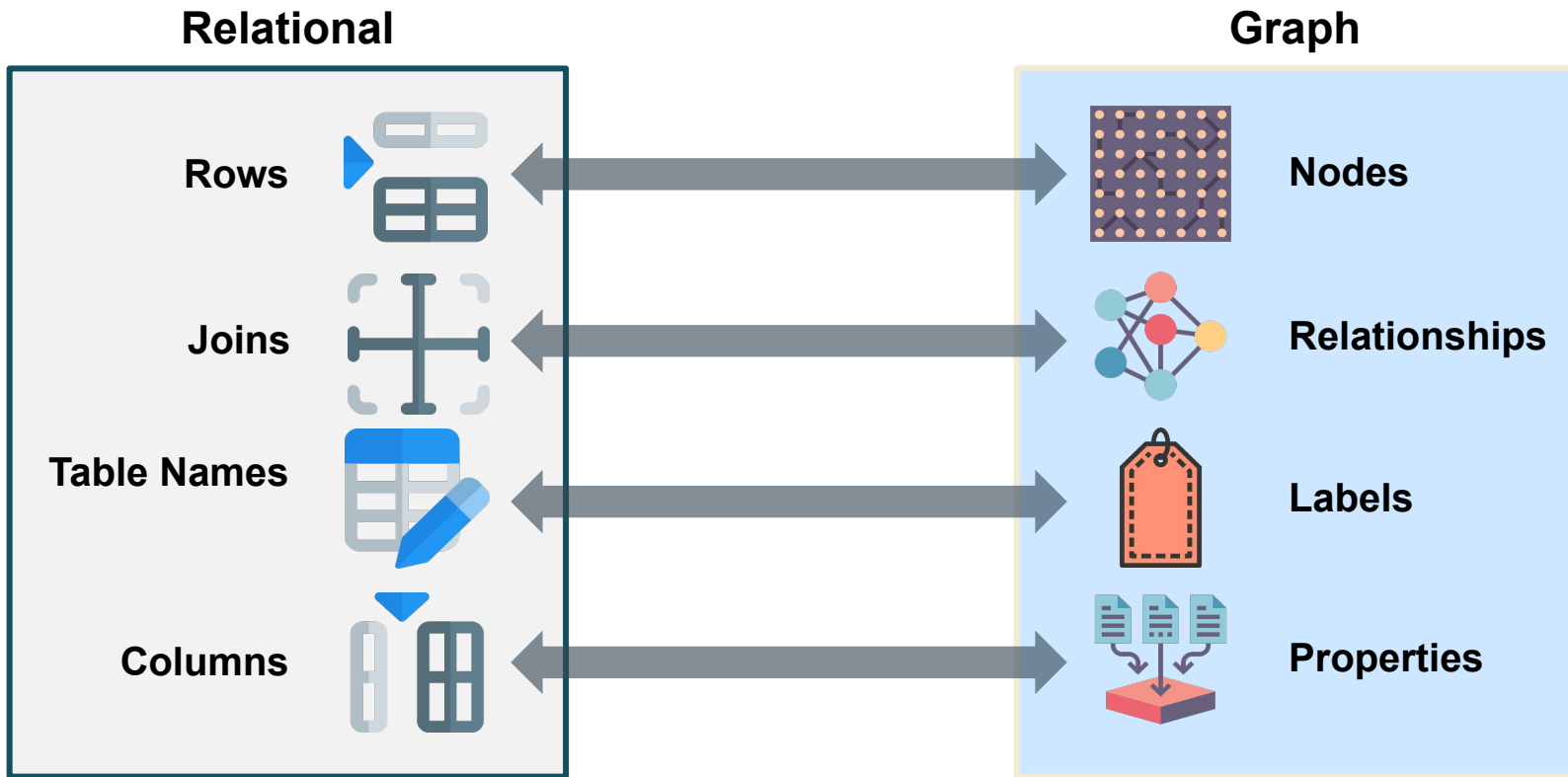


Graph Database Fundamentals

Relationships



Conceptual Mapping Relational → Graph

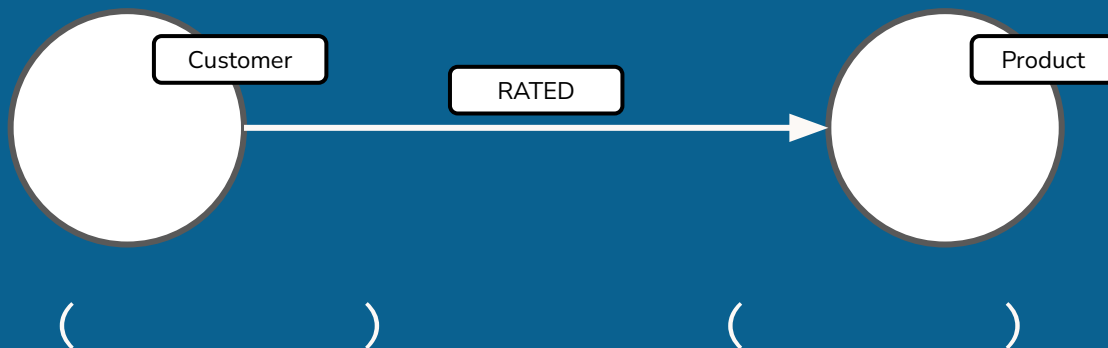


Graph Database Fundamentals

Cypher

Introduction to Cypher

Cypher is a **declarative** language that allows you to **identify patterns** in your data using an **ASCII-art style syntax** consisting of **brackets**, **dashes** and **arrows**.

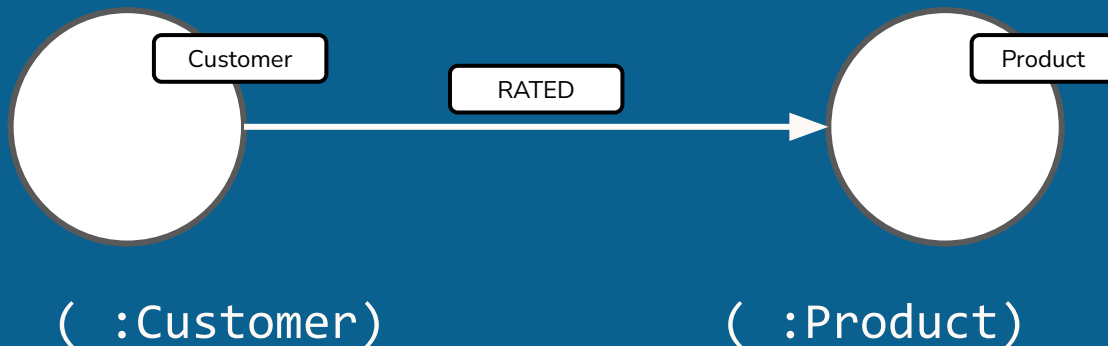


Graph Database Fundamentals

Cypher

Introduction to Cypher

Cypher is a **declarative** language that allows you to **identify patterns** in your data using an **ASCII-art style syntax** consisting of **brackets**, **dashes** and **arrows**.

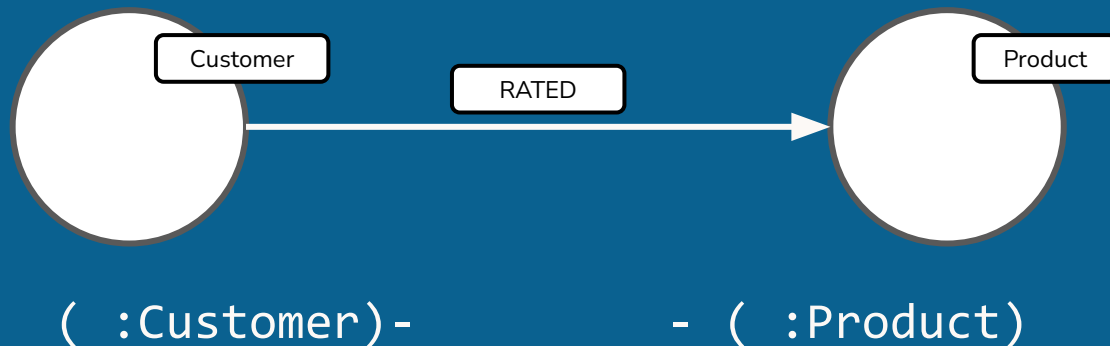


Graph Database Fundamentals

Cypher

Introduction to Cypher

Cypher is a **declarative** language that allows you to **identify patterns** in your data using an **ASCII-art style syntax** consisting of **brackets**, **dashes** and **arrows**.



Graph Database Fundamentals

Cypher

Introduction to Cypher

Cypher is a **declarative** language that allows you to **identify patterns** in your data using an **ASCII-art style syntax** consisting of **brackets**, **dashes** and **arrows**.



(:Customer) -

-> (:Product)

Graph Database Fundamentals

Cypher

Introduction to Cypher

Cypher is a **declarative** language that allows you to **identify patterns** in your data using an **ASCII-art style syntax** consisting of **brackets**, **dashes** and **arrows**.



`(:Customer) - [] -> (:Product)`

Graph Database Fundamentals

Cypher

Introduction to Cypher

Cypher is a **declarative** language that allows you to **identify patterns** in your data using an **ASCII-art style syntax** consisting of **brackets**, **dashes** and **arrows**.



```
( :Customer ) - [ :RATED ] -> ( :Product )
```

Graph Database Fundamentals

Cypher

Introduction to Cypher

Cypher is a **declarative** language that allows you to **identify patterns** in your data using an **ASCII-art style syntax** consisting of **brackets**, **dashes** and **arrows**.



```
(c:Customer) - [r:RATED] -> (p:Product)
```


Graph Database Fundamentals

The **MATCH** clause



```
MATCH (c:Customer)-[r:RATED]->(p:Product)
RETURN c.name AS customer, p.name AS product, r.stars
AS rating
```

Platform

Neo4j Aura

Neo4j as a Service. The fully-managed cloud service platform from Neo4j

Product Family

Developers

Data Scientists

Neo4j AuraDB

Neo4j Aura Graph Database for Applications

Neo4j AuraDS

Neo4j Aura for Graph Data Science

Product Tiers

AuraDB Free

AuraDB Professional

AuraDB Enterprise

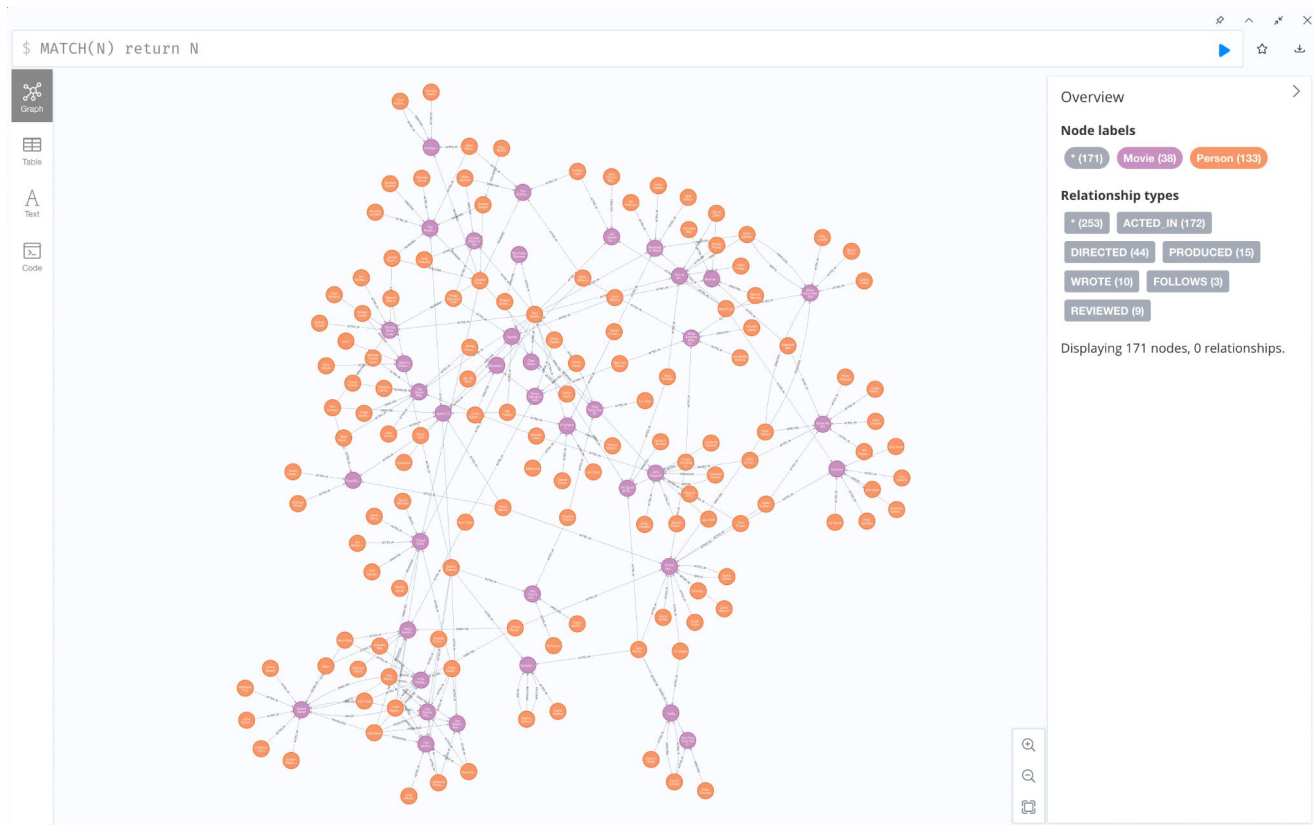
AuraDS Self Start

AuraDS Enterprise

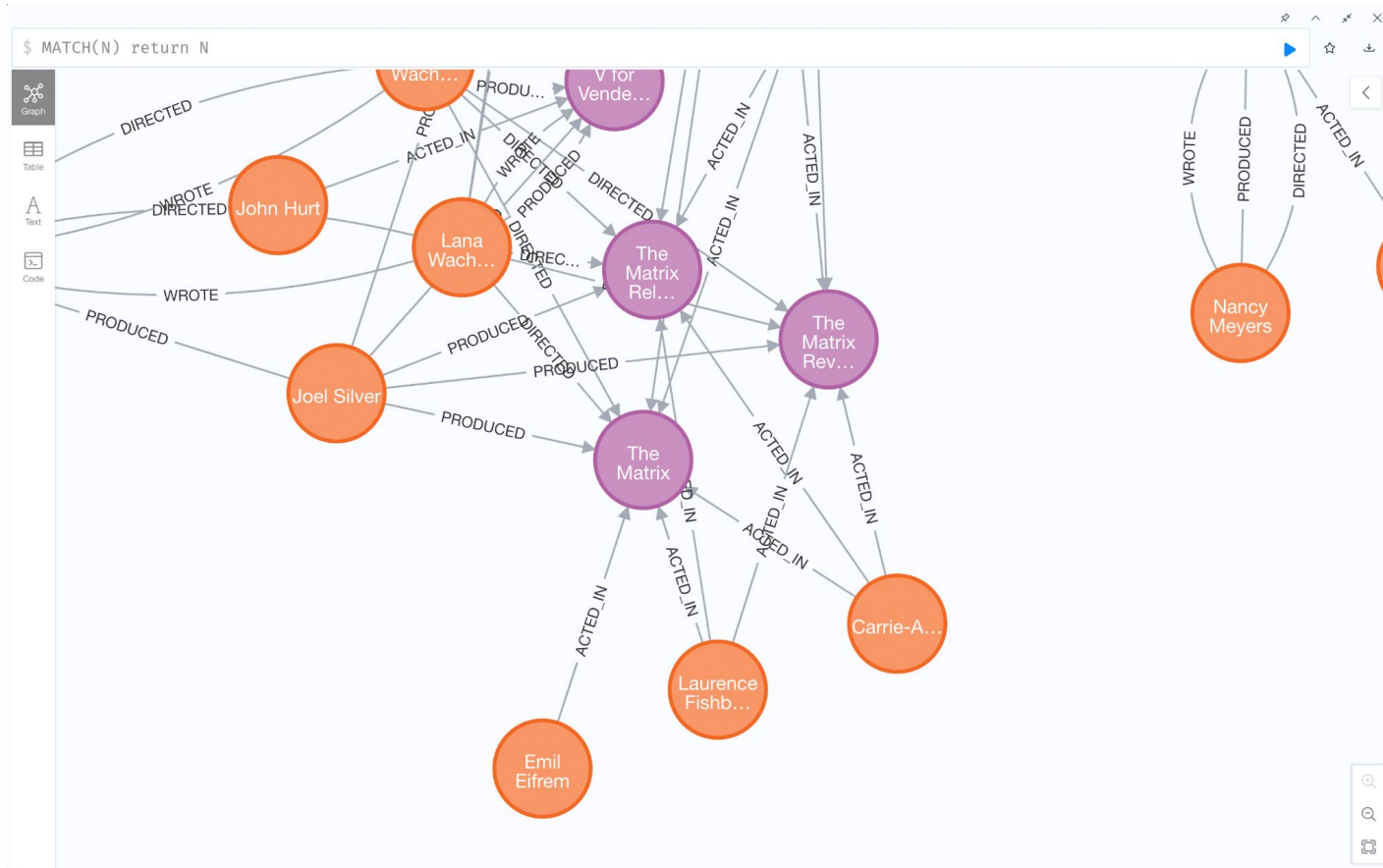
Getting started with Neo4j AuraDB

www.neo4j.com/aura

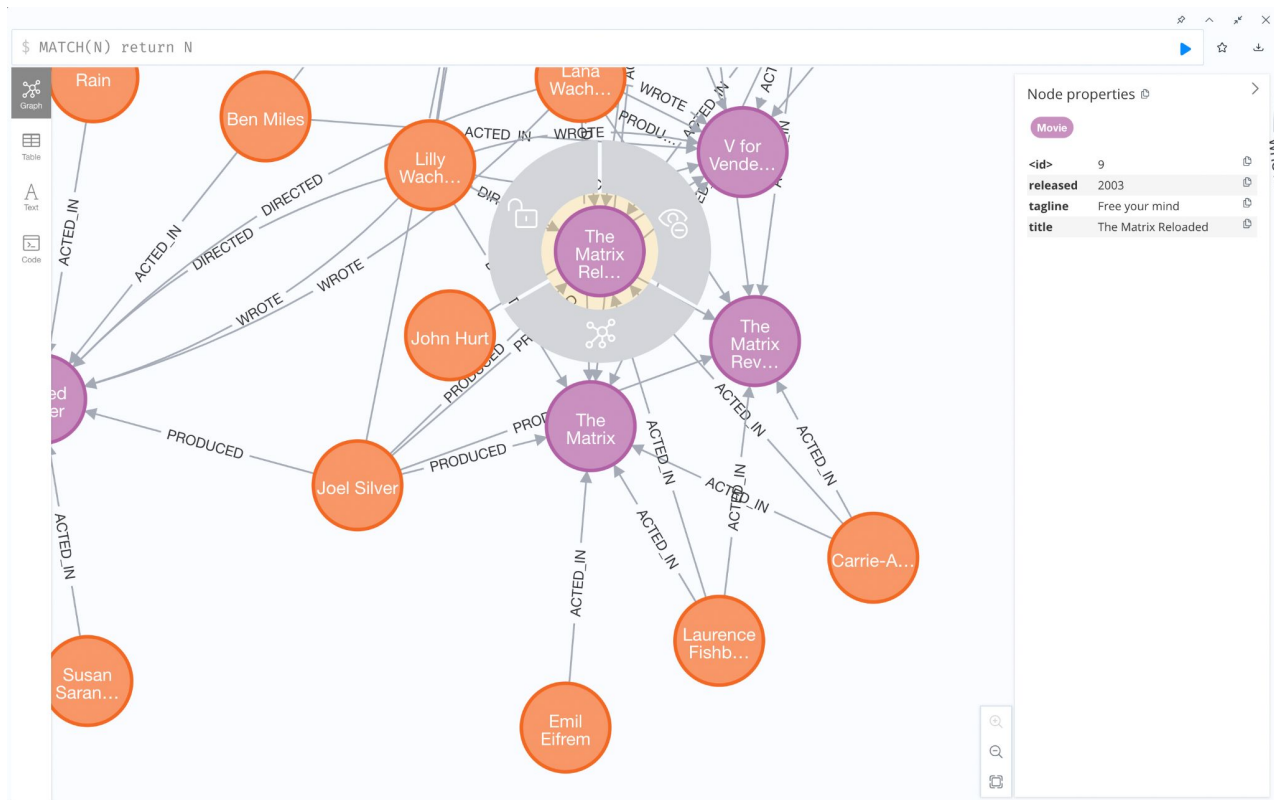
Graphs Explained



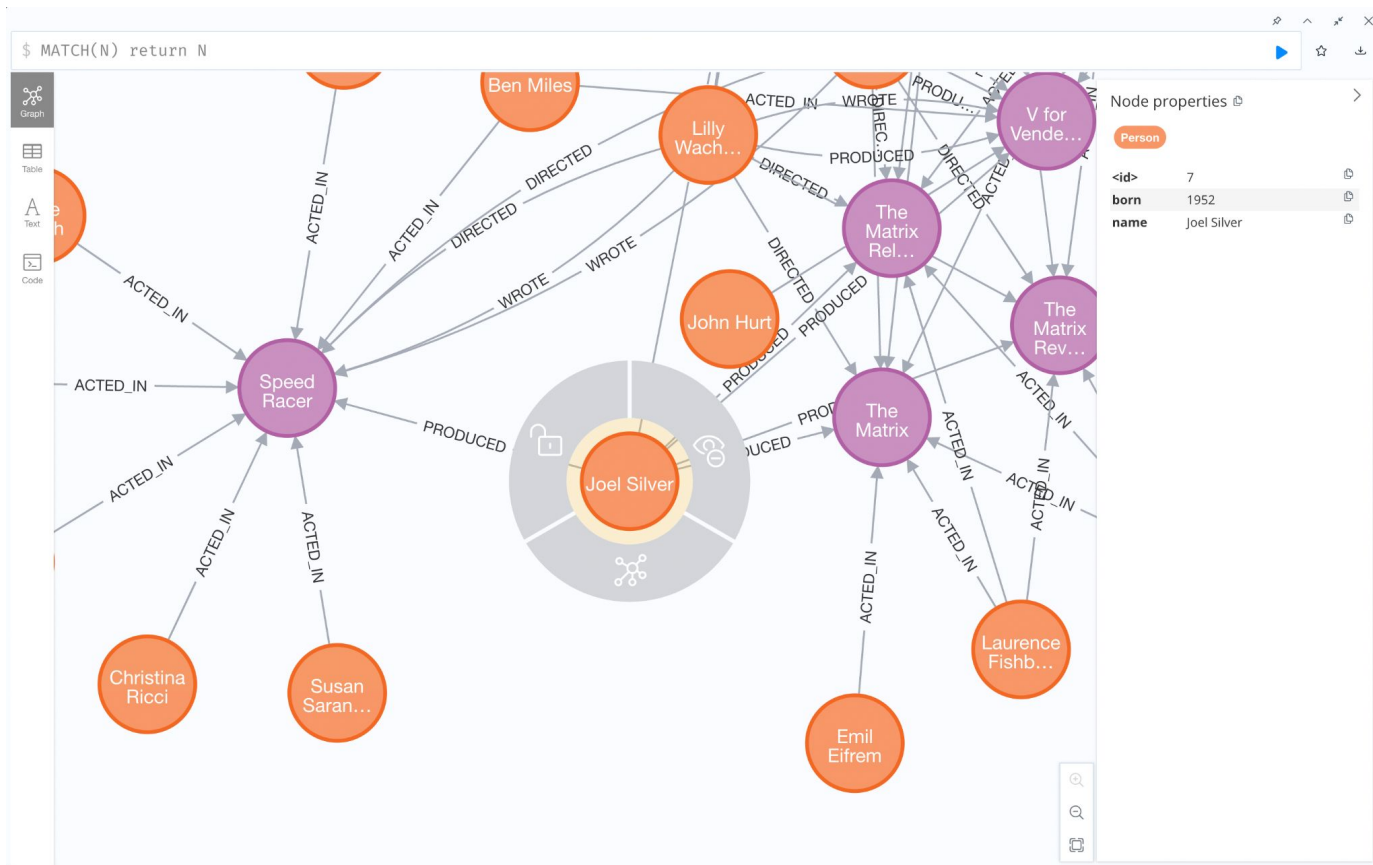
Graphs Explained



Graphs Explained

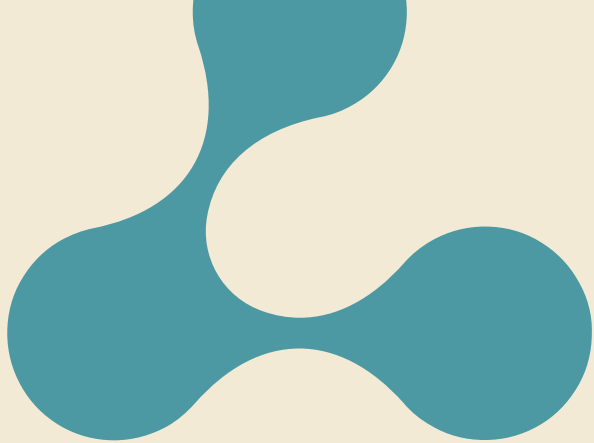


Graphs Explained




Populating data in Neo4j AuraDB






<https://github.com/sidagarwal04/graph-powered-nlp-workshop>



MakerSuite Explained





Welcome to MakerSuite

Prototype with generative AI


Pick a prompt type to get started working with the PaLM API. [Learn more](#)



Text prompt

A freeform way to experiment with language models


[Create](#)



Data prompt

A table that uses rows and columns to organize prompts

[Create](#)




Chat prompt

A template for back-and-forth chatbot conversations

[Create](#)


Explore the full capabilities of the PaLM API.



Model tuning

Improve the model's responses by using more examples than a standard prompt.

[Create a tuned model](#)



Start developing

Call the PaLM API directly from your code.

[Create an API key](#)

[Learn more about Google's generative AI capabilities.](#)

Getting started with MakerSuite

<https://makersuite.google.com/>



You are an expert in converting English questions to Neo4j Cypher Graph code! The Graph has following Node Labels - Movie, Person! the Movie Node has the following properties released, tagline, title, The Person node has the properties such as name and born. The Neo4j Graph has the following Relationship types ACTED_IN, DIRECTED, FOLLOWS, PRODUCED, REVIEWED, WROTE!

All relationships ACTED_IN, DIRECTED, FOLLOWS, PRODUCED, REVIEWED, WROTE start from Person to Movie and not the other way around.

○ ○ ○

For example,

Example 1 - List down 5 movies that released after the year 2000, the Cypher command will be something like this

```
``` MATCH (m:Movie)
WHERE m.released > 2000
RETURN m LIMIT 5
```
```


Example 2 - Get all the people who acted in a movie that was released after 2010.

```
```
MATCH (p:Person)-[r:ACTED_IN]->(m:Movie)
WHERE m.released > 2010
RETURN p,r,m
```
```

Example 3 - Name the Director of the movie Apollo 13?

```
```
MATCH (m:Movie)<-[:DIRECTED]-(p:Person)
WHERE m.title = 'Apollo 13'
RETURN p.name
```
```

Dont include ``` and \n in the output


Untitled prompt
Save
Get code

+

📁

🔍

⋮

🔗

👤

Write your prompt

Movie Node has the following properties released, tagline, title, The Person node has the properties such as name and born. The Neo4j Graph has the following Relationship types ACTED_IN, DIRECTED, FOLLOWS, PRODUCED, REVIEWED, WROTE!

All relationships ACTED_IN, DIRECTED, FOLLOWS, PRODUCED, REVIEWED, WROTE start from Person to Movie and not the other way around.

For example,

Example 1 - List down 5 movies that released after the year 2000, the Cypher command will be something like this

```
''' MATCH (m:Movie)
WHERE m.released > 2000
RETURN m LIMIT 5
'''
```

Example 2 - Get all the people who acted in a movie that was released after 2010.

```
'''
MATCH (p:Person)-[r:ACTED_IN]->(m:Movie)
WHERE m.released > 2010
RETURN p,r,m
'''
```

Example 3 - Name the Director of the movie The Matrix Reloaded?

```
'''
MATCH (m:Movie)<-[:DIRECTED]-(p:Person)
WHERE m.title = 'Apollo 13'
RETURN p.name
'''
```

Dont include ''' and \n in the output

Name the director of the movie The Matrix reloaded?

Test your prompt

| INPUT | input | OUTPUT |
|-------|---|-------------------|
| 1 | Name the director of the movie The Matrix reloaded? | Run to get output |

Add test example

Run
Text Bison 0.7
Text preview 338 / 8196

▼ Test your prompt ⓘ

| INPUT input | OUTPUT |
|---|---|
| 1 Name the director of the movie The Matrix reloaded? | MATCH (m:Movie)<-[:DIRECTED]-(p:Person) WHERE m.title = 'The Matrix Reloaded' RETURN p.name |
| ⊕ Add test example | |

Setting up Google MakerSuite

<https://github.com/sidagarwal04/graph-powered-nlp-workshop>

Untitled prompt ✎

Save <> Get code

Sample prompts Summarize a paragraph Casual ponderings Poem writer Go to gallery

Write your prompt ⓘ

Get code

Create your API key before using the code in your project

You can call this prompt from the PaLM API by copying the following code into your project

cURL JavaScript JSON Python Open in Colab Copy

```

1 """
2 At the command line, only need to run once to install the package via pip:
3
4 $ pip install google-generativeai
5 """
6
7 import google.generativeai as palm
8 palm.configure(api_key="YOUR API KEY")
9
10 defaults = {
11     'model': 'models/text-bison-001',
12     'temperature': 0.7,
13     'candidate_count': 1,
14     'top_k': 40,
15     'max_output_tokens': 1024
16 }

```

Test your prompt ⓘ

| INPUT | OUTPUT |
|---|---|
| 1 Name the director of the movie The Matrix reloaded? | MATCH (m:Movie)<-[DIRECTED]-<(p:Person) WHERE m.title = 'The Matrix Reloaded' RETURN p.name |

Add test example

Run Text Bison 0.7 Text preview 338 / 8196



Building Things Together

Building things together with Google MakerSuite and Neo4j AuraDB using Python

<https://colab.research.google.com/>

Python

○ ○ ○

```
def get_answer(input):
    defaults = {
        'model': 'models/text-bison-001',
        'temperature': 0.7,
        'candidate_count': 1,
        'top_k': 40,
        'top_p': 0.95,
        'max_output_tokens': 1024,
        'stop_sequences': []}

    prompt = f"""You are an expert in converting English questions to Neo4j Cypher Graph code!
        The Graph has following Node Labels - Movie, Person! the Movie Node has the
        following properties released, tagline, title, The Person node has the
        properties such as name and born. The Neo4j Graph has the following
        Relationship types ACTED_IN, DIRECTED, FOLLOWS, PRODUCED, REVIEWED, WROTE!

        {input}"""

    response = palm.generate_text(**defaults, prompt=prompt)
    return response.result
```

Python

○ ○ ○

```
def run_cypher_on_neo4j(inp_query, inp_key):  
    out_list = []  
    with driver.session() as session:  
        result = session.run(inp_query)  
        for record in result:  
            out_list.append(record[inp_key])  
    driver.close()  
    if len(out_list) > 1:  
        return format_names_with_ampersand(out_list)  
    else:  
        return out_list[0]
```

Python

○ ○ ○

```
URI = "URL"
```

```
AUTH = ("username", "Password")
```

```
with GraphDatabase.driver(URI, auth=AUTH) as driver:  
    driver.verify_connectivity()
```

Python

○ ○ ○

```
gr.Interface(fn = chatbot,  
            inputs = ["text", 'state'],  
            outputs = ["chatbot", 'state']).launch(debug = True)
```

Python

input

Who directed the movie The Matrix?

Clear Submit

output 0

Who directed the movie The Matrix?

Lana Wachowski & Lilly Wachowski

Flag

Python

output 0

Who directed the movie The Matrix?

Lana Wachowski & Lilly Wachowski

Flag

Python

output 0

Who directed the movie The Matrix?

Lana Wachowski & Lilly Wachowski

Name the actors of The Matrix?

Emil Eifrem, Hugo Weaving, Laurence Fishburne, Carrie-Anne Moss & Keanu Reeves

Flag

Python

output 0

Name the actors of The Matrix?

Emil Eifrem, Hugo Weaving, Laurence Fishburne, Carrie-Anne Moss & Keanu Reeves

Name the producer of When Harry Met Sally?

Nora Ephron & Rob Reiner

Flag

Python

Building a Q&A Chatbot

<https://github.com/sidagarwal04/graph-powered-nlp-workshop>

neo4j

(:neo4j)-[:LOVES]->(:community)

Join Graph Database User Groups

What are User Groups?

Developers interested in learning about and working with graph databases for social, spatial, hierarchical or other highly connected data sets. We host hands-on lab sessions, technology reviews and topical lectures. Curious about graphs? Join us!

What's more?

Apply knowledge and connections to build great products and advance your skills, career, and network. Help your community learn too.

Interested? For more information visit:
<https://www.meetup.com/pro/neo4j/>



Want to speak at our meetups?

Submit your talk



dev.neo4j.com/submit-your-talk

Continue your graph journey with Graph Academy

Free, Self-Paced, Hands-on Online Training to help you learn how to build, optimize and launch your Neo4j project, all from the Neo4j experts.

What's more?

2 free certifications designed to test you on your overall knowledge of Neo4j.



Interested? For more information visit:
www.graphacademy.neo4j.com



Online Developer Conference October 26, 2023

- 3 tracks:
 - Building Applications & APIs
 - ML & AI with Graphs
 - Powerful Visualizations
- 24 hours
- 100 talks

<https://www.youtube.com/@neo4j>

Thank you!

Siddhant Agarwal ([@sidagarwal04](https://twitter.com/sidagarwal04))