

neo4j

Graph Powered NLP using Google PaLM 2, Neo4j and Python

Hi, I'm Sid 

Developer Relations Lead, APAC at Neo4j

Hi, I'm Sid



Developer Relations Lead, APAC at Neo4j

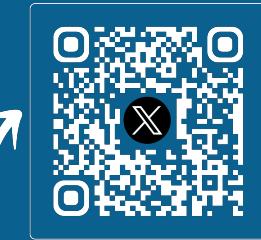
Started my career with IBM as Java Technology Engineer

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

Gen-AI & LLMs Explained

Show of hands 🙌

1. Used ChatGPT ?
2. Used GenAI APIs ?
3. Heard of Neo4j or Knowledge Graphs?

Generative AI is Predicted to Unlock

**\$6.6 Trillion
In Economic Value**

Up to 3.3% productivity improvement annually

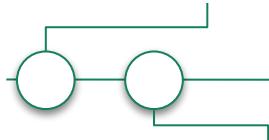
Key Terminologies

- **Generative AI (GenAI):** A type of artificial intelligence (AI) system that generates text, images, or other media in response to prompts.
- **Language Model (LM):** An ML approach that models the probability distribution over a sequence of words. Predicts the probabilities of next word/character in a sequence. Applications in GenAI as well as embedding, classification, and other ML tasks.
- **Large Language Model (LLM):** LMs consisting of large neural networks (billions of parameters) trained on large quantities of data often using self-supervised/semi-supervised approaches. Trained for General tasks and currently seen as the “**GenAI for language/text**”.
- **Generative Pre-Trained Transformer (GPT):** A type of GenAI model that combines two forms of training to produce foundation models. Specifically:
 - **Pre-Training:** Training general purpose generative capabilities on vast quantities data
 - **Fine-Tuning:** Training finite number of supervised ML tasks on a small amount of hand picked data

Imp. Innovations

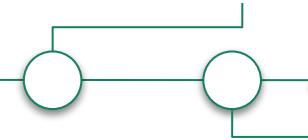
Dec, 2013

Variational Autoencoders (VAE)
Light New Spark for Generative AI



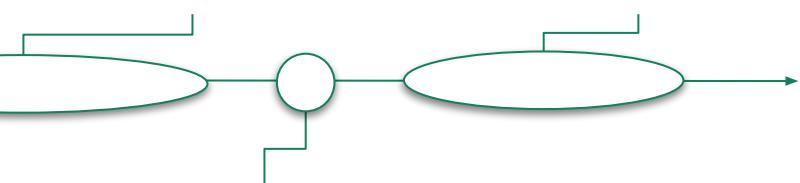
June, 2017

Google: Attention is All you Need
(Transformer Architecture)



2018-2022

OpenAI: GPT scale-up,
Few-Shot Learning &
more

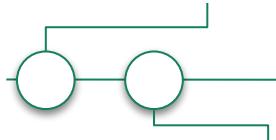


2023

GenAI arms race begins. ChatGPT-4.0 released. Microsoft, Google, Amazon, and others offering GenAI services and integrations

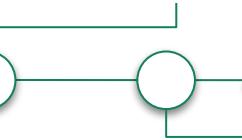
June, 2014

First Generative Adversarial Networks (GANs) - AI generated images



June, 2018

OpenAI: GPT-1 Improve Language Understanding by Generative Pre-Training



Nov, 2022

OpenAI: ChatGPT Released to public GPT 3.5

Scaling ML & AI with Foundation Models

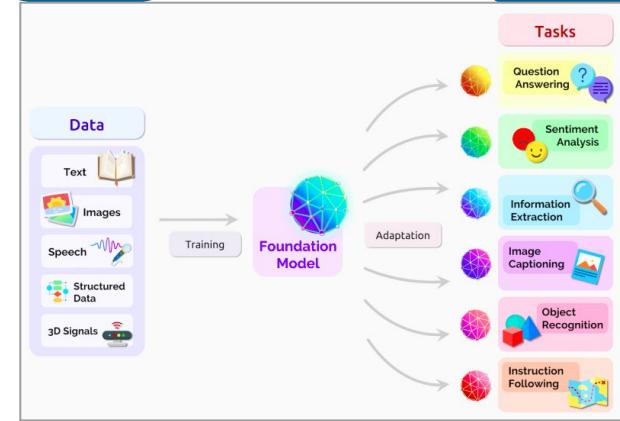
Traditional Machine Learning & AI

- Models trained for narrow specific tasks
- Poor transferability
- Difficult to get enough training data
- Challenging to scale
- Require specialized expertise

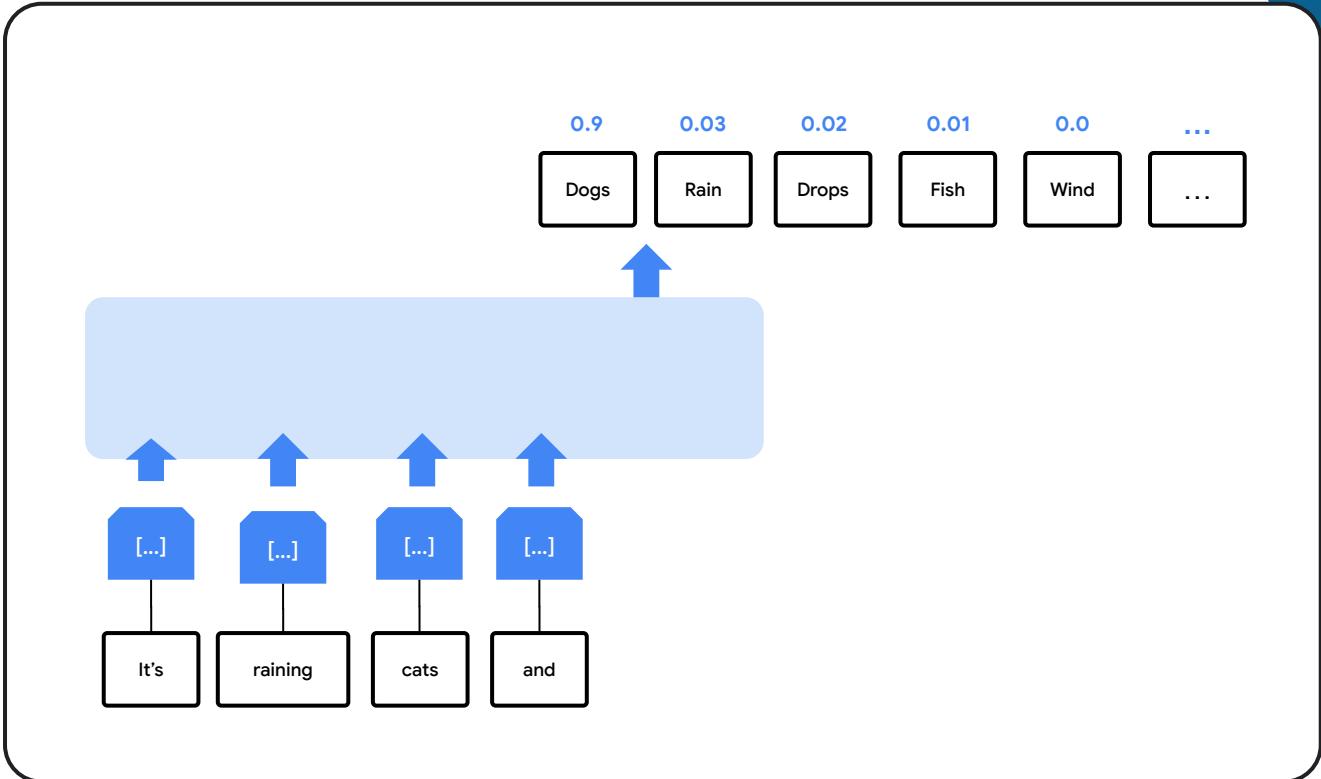


Foundation Model AI Paradigm

- Train a large “Foundation Model”
- Adapt to many different use cases through fine tuning or other forms of augmentation
- Scalable & transferable
- Accessible to developers and other non-experts



What is an LLM?



LLMs Give Us an Amazing Opportunity to:

1

Automate data retrieval tasks

2

Improve customer service experiences

3

Expedite reading, understanding, & summarizing

4

Generate content & code

Save time and money Improve growth and retention:

1

Customer Operations

2

Marketing & Sales

3

Software Engineering

4

R&D

75% of GenAI value will come from four areas

But There Are Challenges...

1

Knowledge cut-off

2

Reasonable answers,
not always accurate

3

Can inherit bias through
training data

4

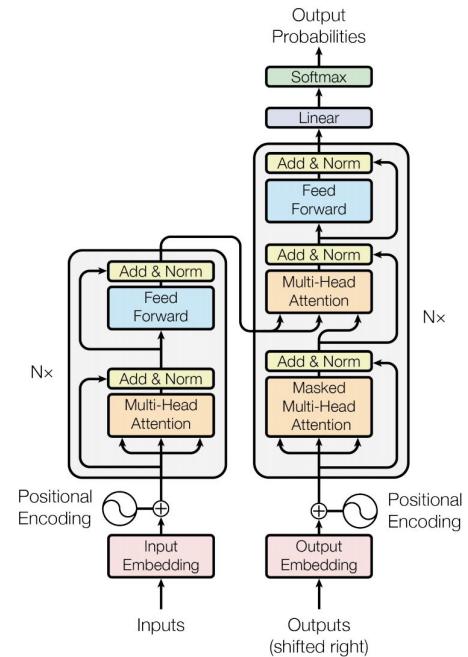
Lack of enterprise
domain knowledge

5

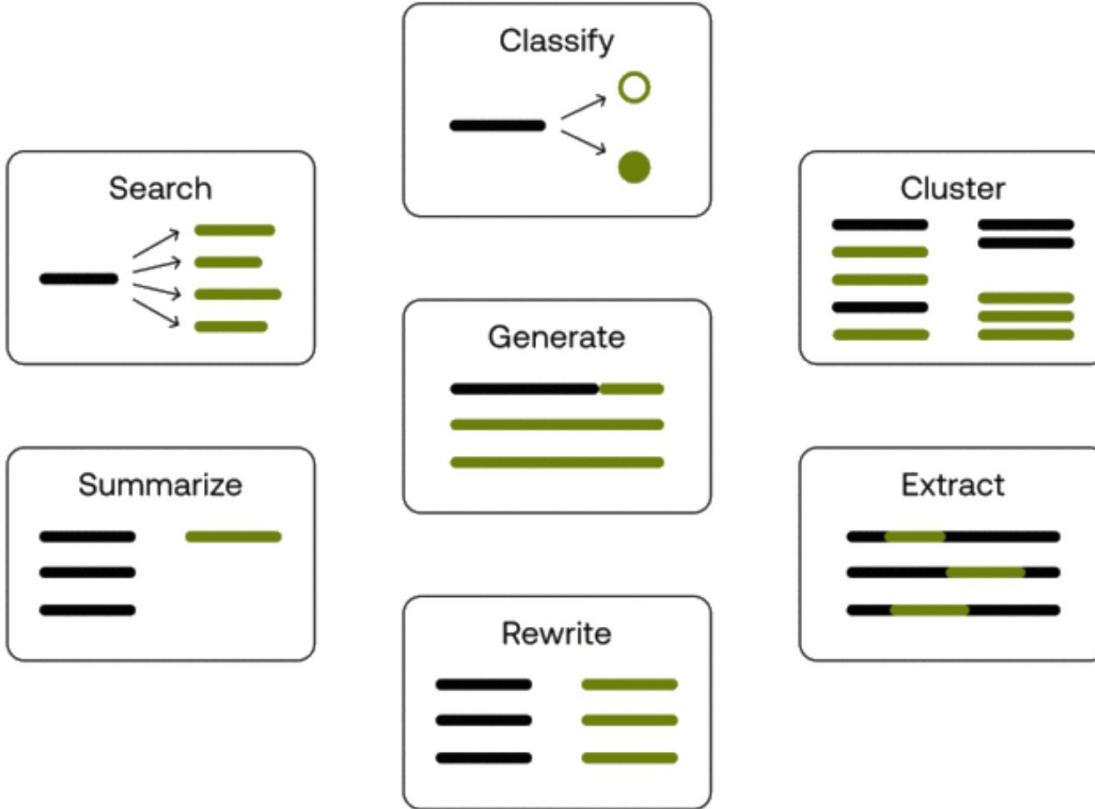
Inability to verify or
attribute sources

AI models to learn a language and generate information

- **Large Machine Learning Models** are trained on Petabytes of data (trillions of tokens, with 7-220bn parameters).
- They are based on the **Transformer Architecture** (Google) that probabilistically predicts the next token in a sequence.
- Emergent behavior, with higher than expected capabilities, appears from a certain model size. These models can generate, analyze, correct, explain different modalities - text, code, images, audio.
- GenAI utilizes existing context from previous interactions, additional context and can be fine-tuned for specific purposes



LLMs Explained



Roses are red,

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

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

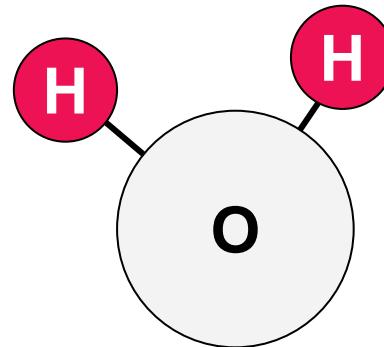
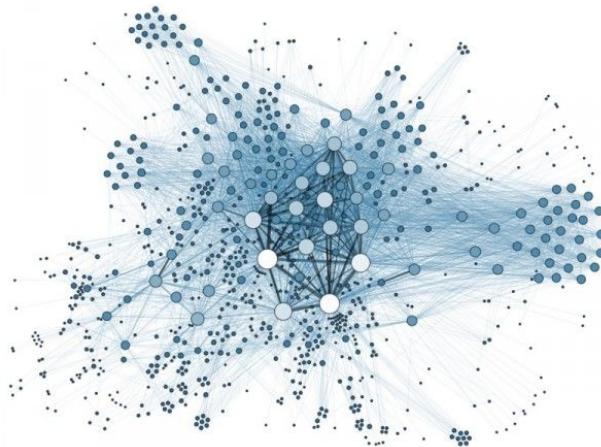
```
for(var i = 0; i <10; i++) {
```

LLMs let us
prototype *fast*.

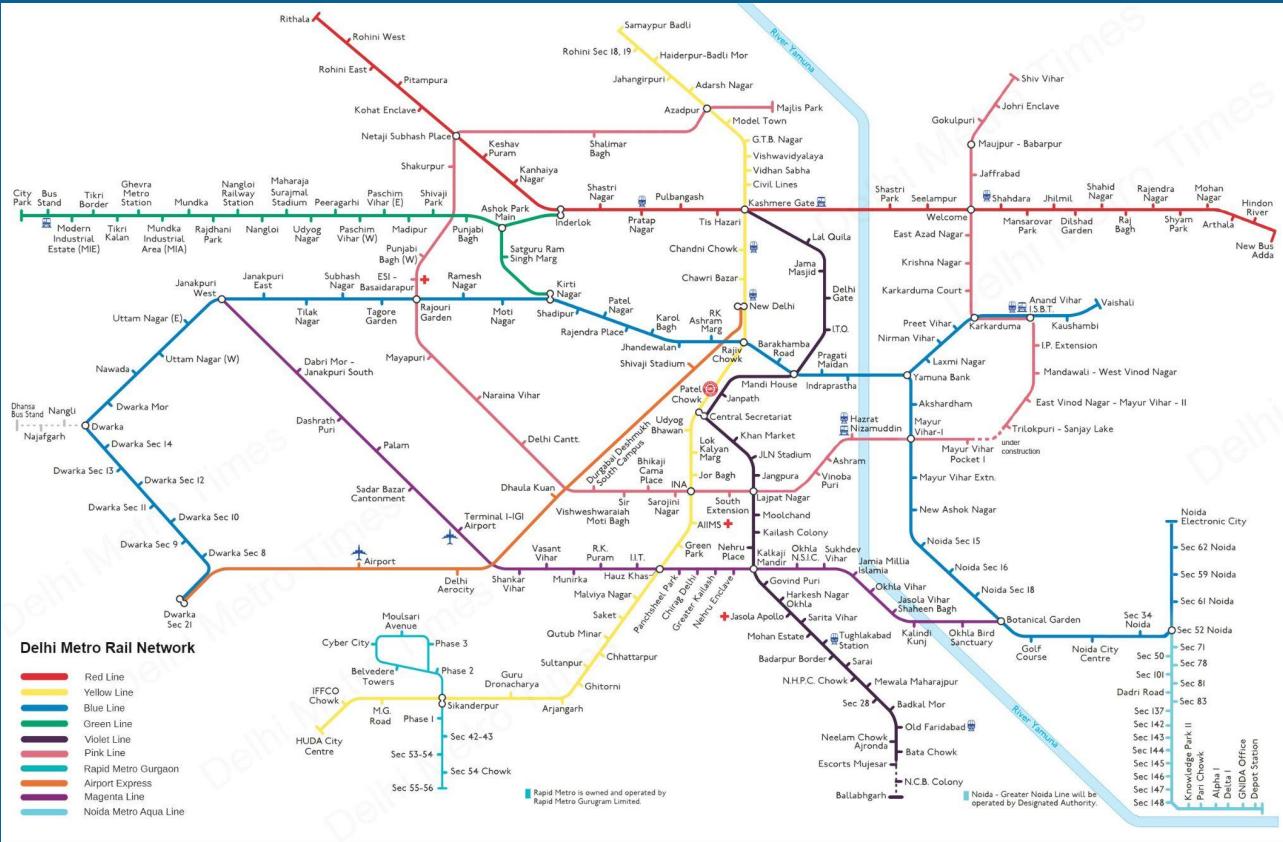
Graphs Explained

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?

1980+

Relational Database

Structured, schema-based store/retrieve

Optimized for real-time transactional processing/reporting

SQL

2005+

NoSQL Database

Scale-out, simple model store/retrieve

Optimized for application-driven, volume throughput

App
Based

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

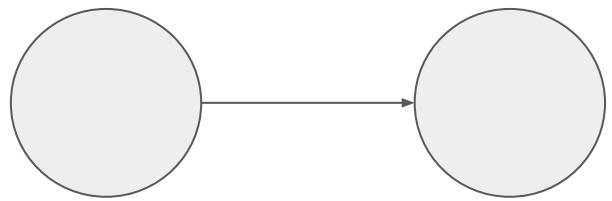
GQL

*First database language
standard in 40 years.*

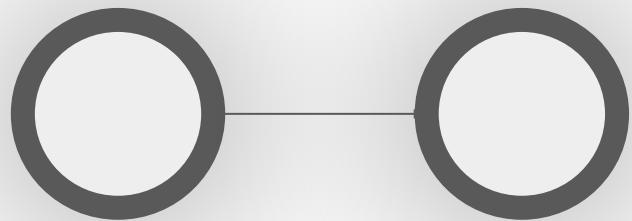
 neo4j

Pioneered the
graph database
in 2002

Core of Graph Theory



Core of Graph Theory: Nodes

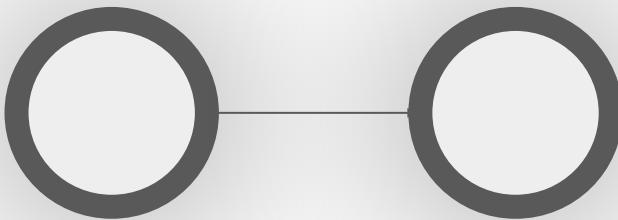


Nodes / Vertices

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

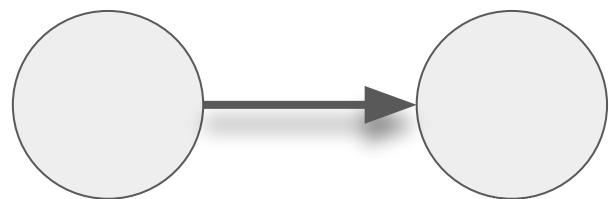


Core features of Neo4j: Nodes

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

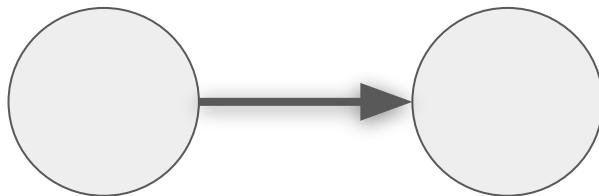


Relationship / Edge

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



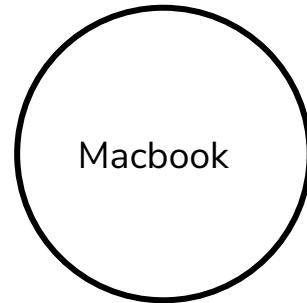
Core features of Neo4j: Relationships

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

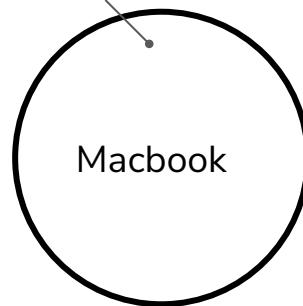
Nodes



Graph Database Fundamentals

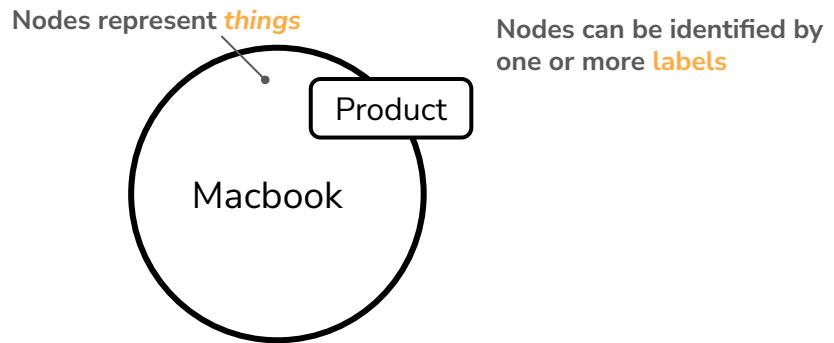
Nodes

Nodes represent *things*



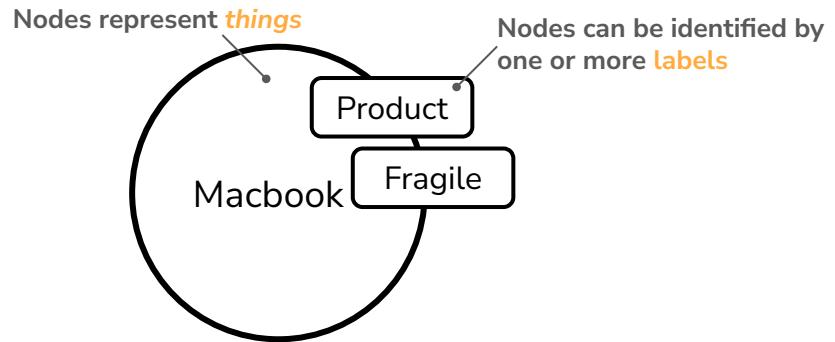
Graph Database Fundamentals

Nodes



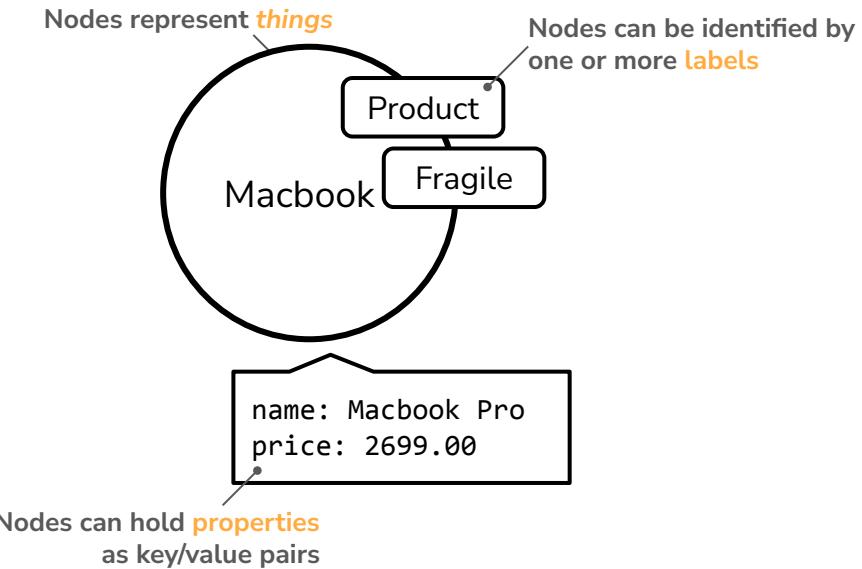
Graph Database Fundamentals

Nodes



Graph Database Fundamentals

Nodes



Graph Database Fundamentals

Relationships



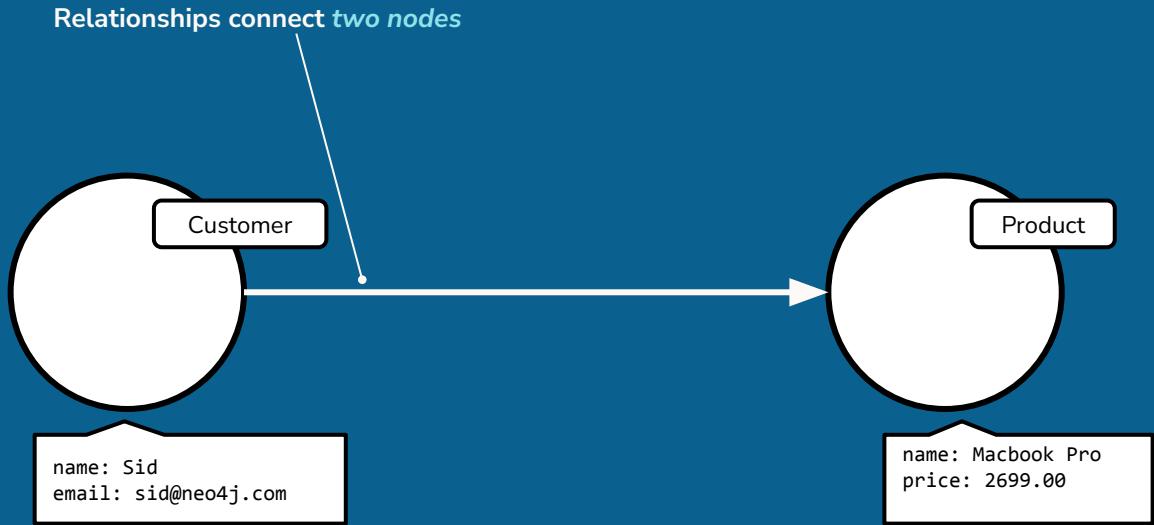
Graph Database Fundamentals

Relationships



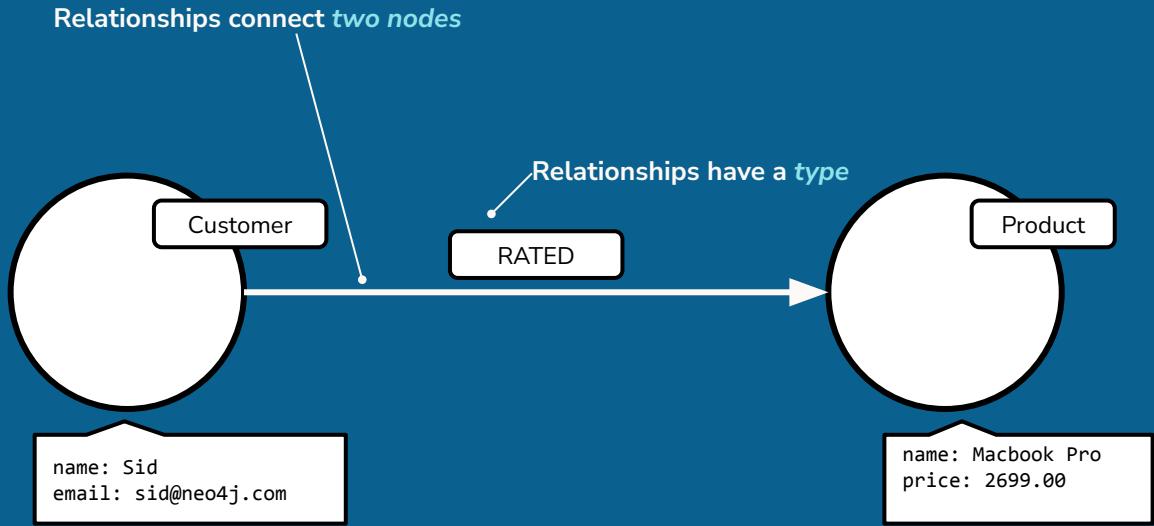
Graph Database Fundamentals

Relationships



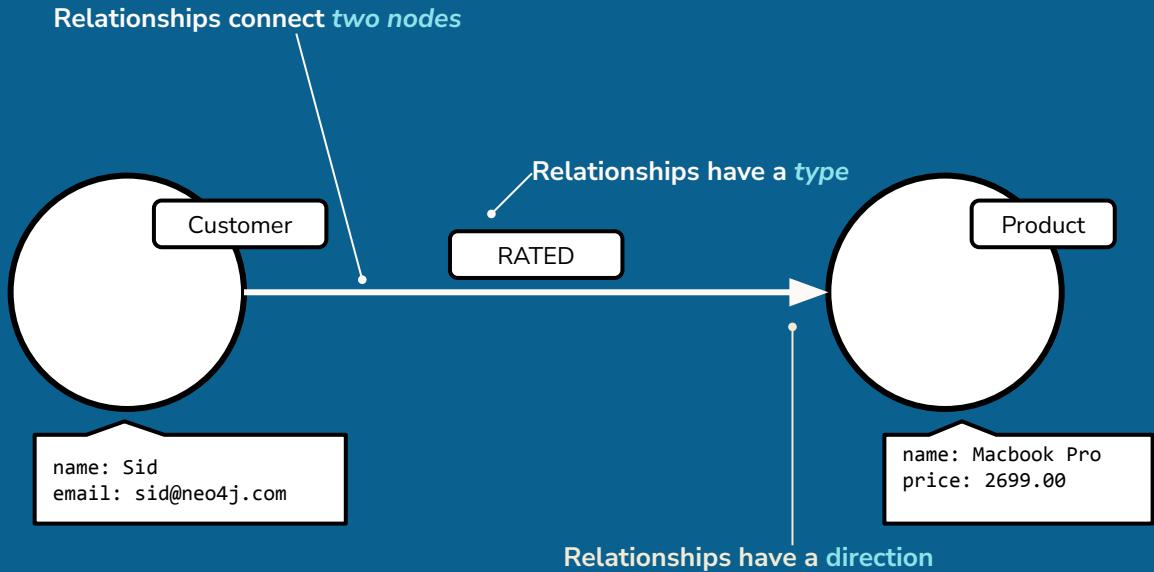
Graph Database Fundamentals

Relationships



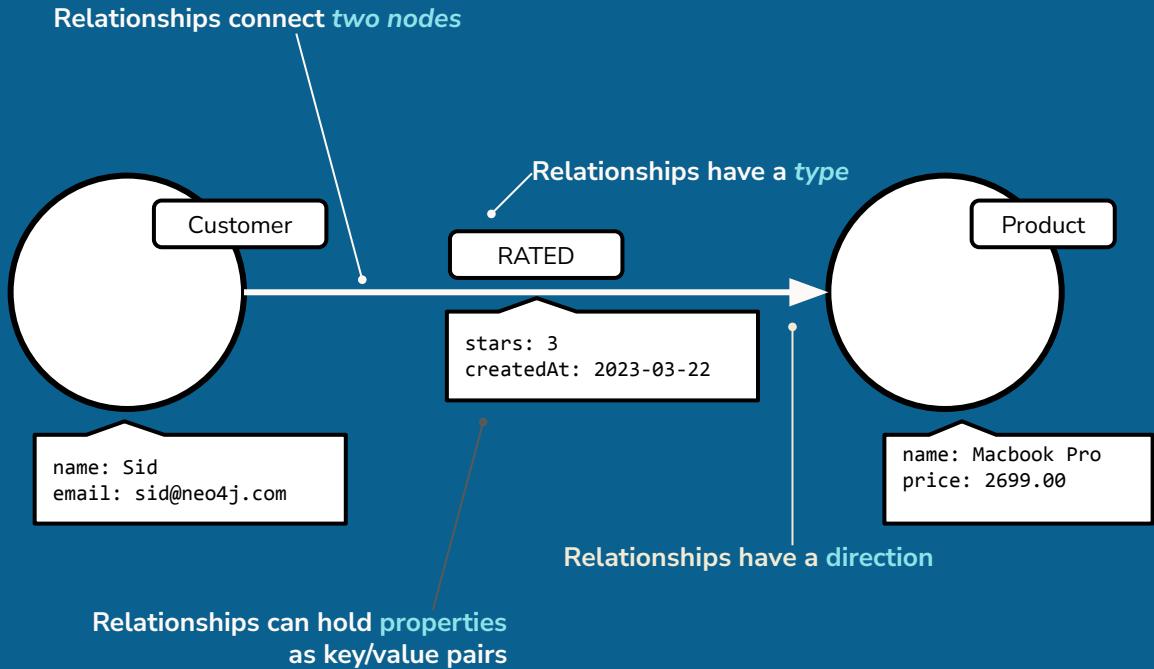
Graph Database Fundamentals

Relationships

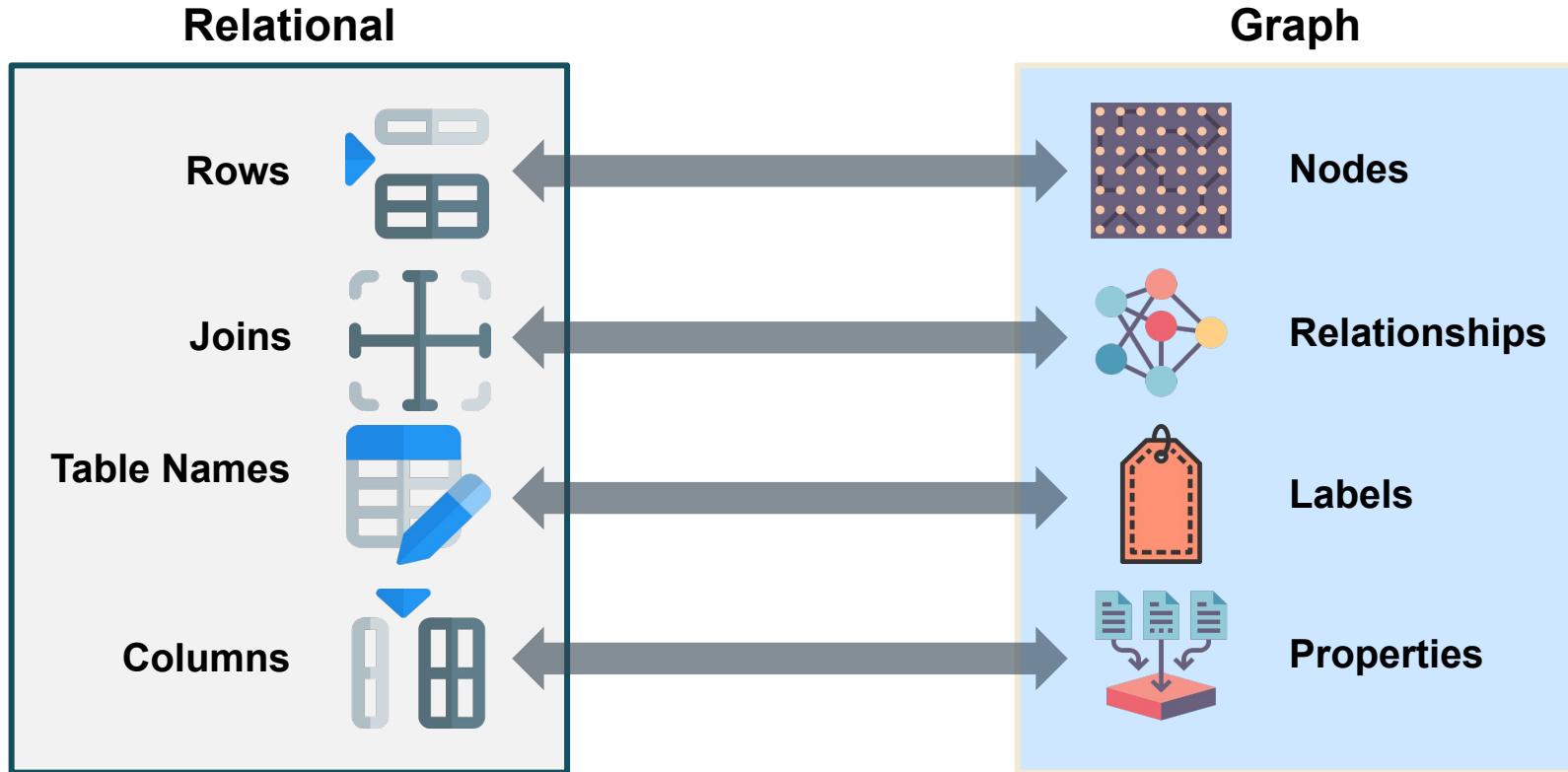


Graph Database Fundamentals

Relationships



Conceptual Mapping Relational → Graph

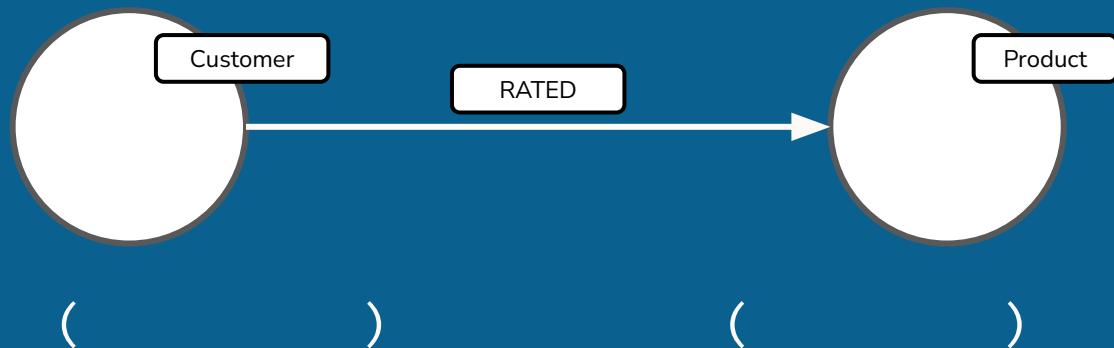


Graph Database Fundamentals

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**.

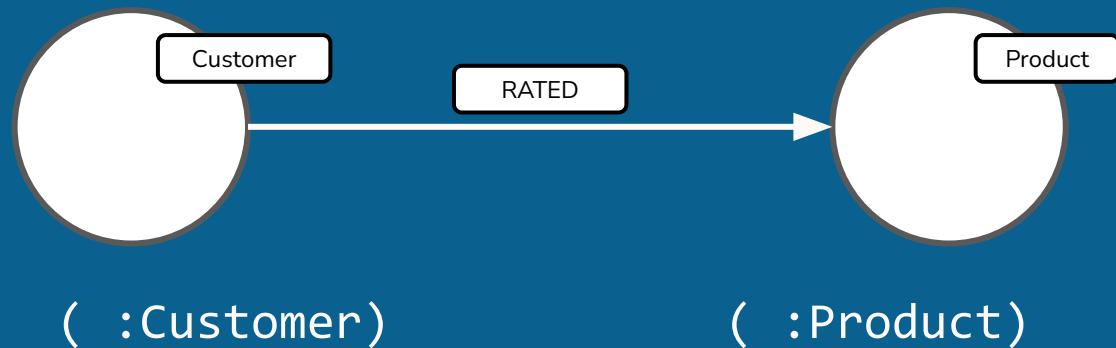


Graph Database Fundamentals

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**.



(: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**.



(: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

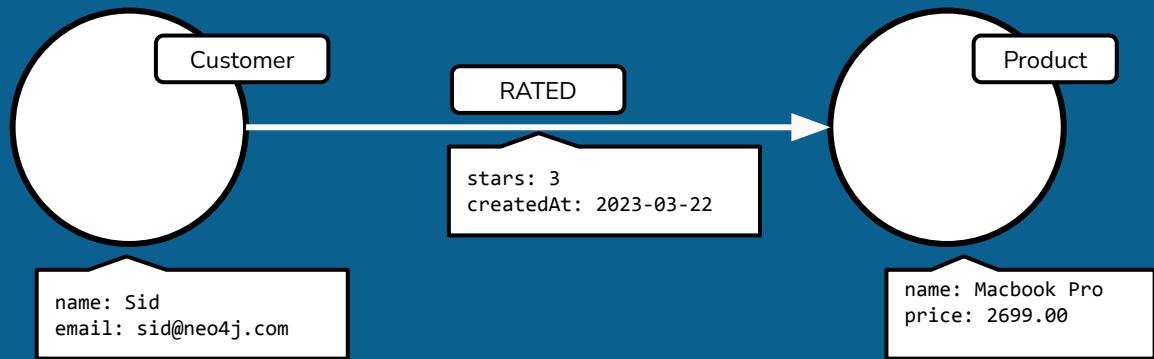
Cypher is a **declarative** language that allows you to **identify patterns** in your data using an **ASCII-art style syntax** consisting of **brackets, dashes and arrows**.



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

Graph Database Fundamentals

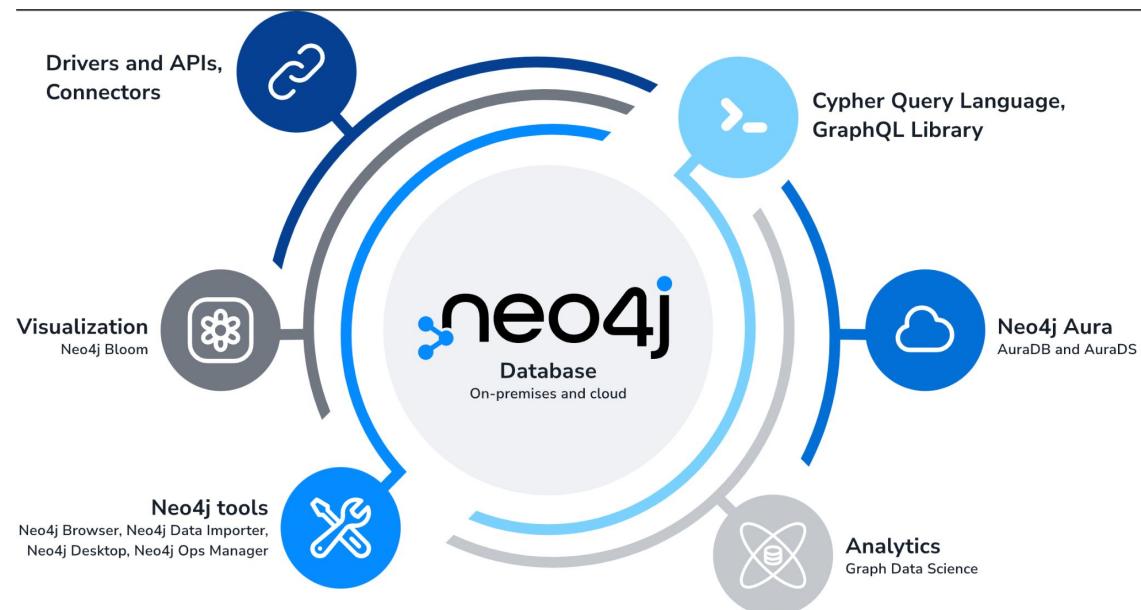
The **MATCH** clause



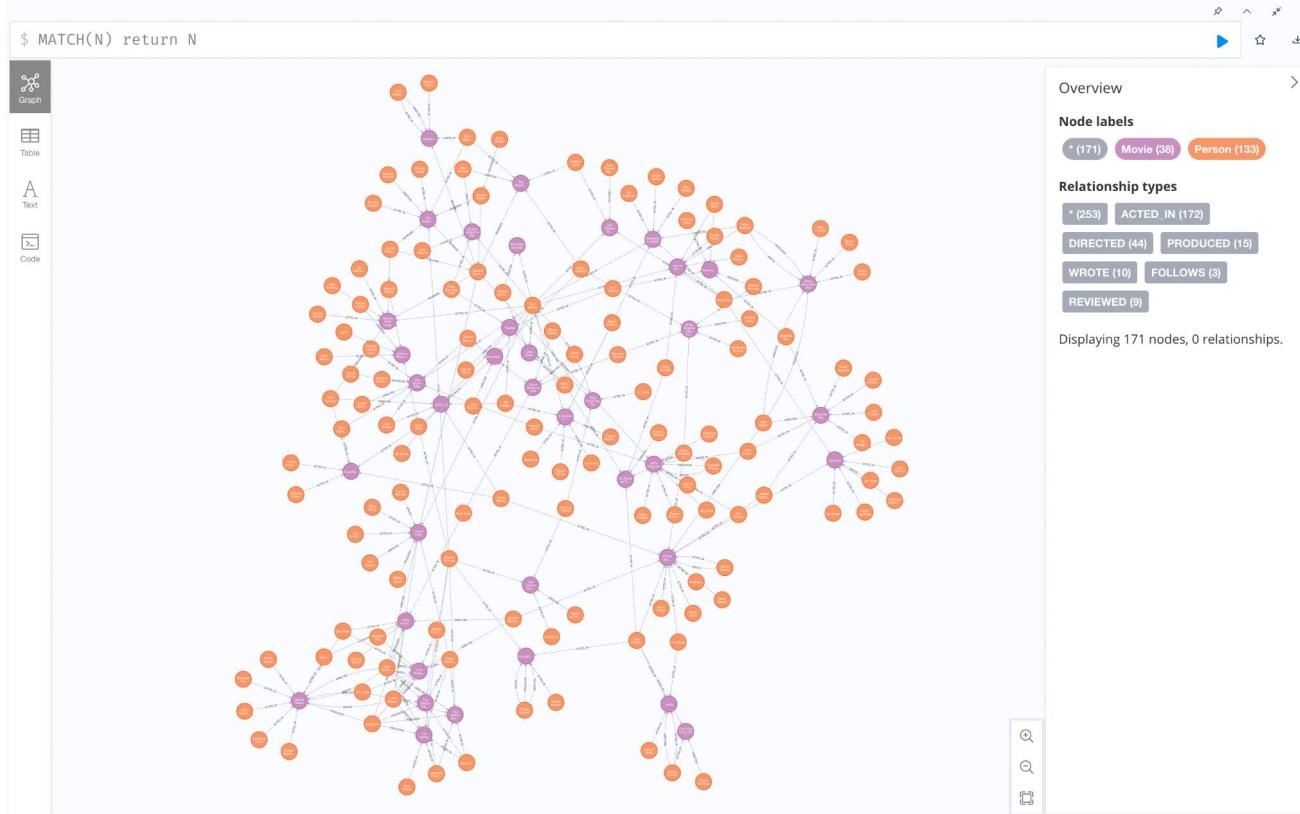
```
MATCH (c:Customer)-[r:RATED]->(p:Product)  
RETURN c.name AS customer, p.name AS product, r.stars  
AS rating
```

Getting started with Neo4j AuraDB

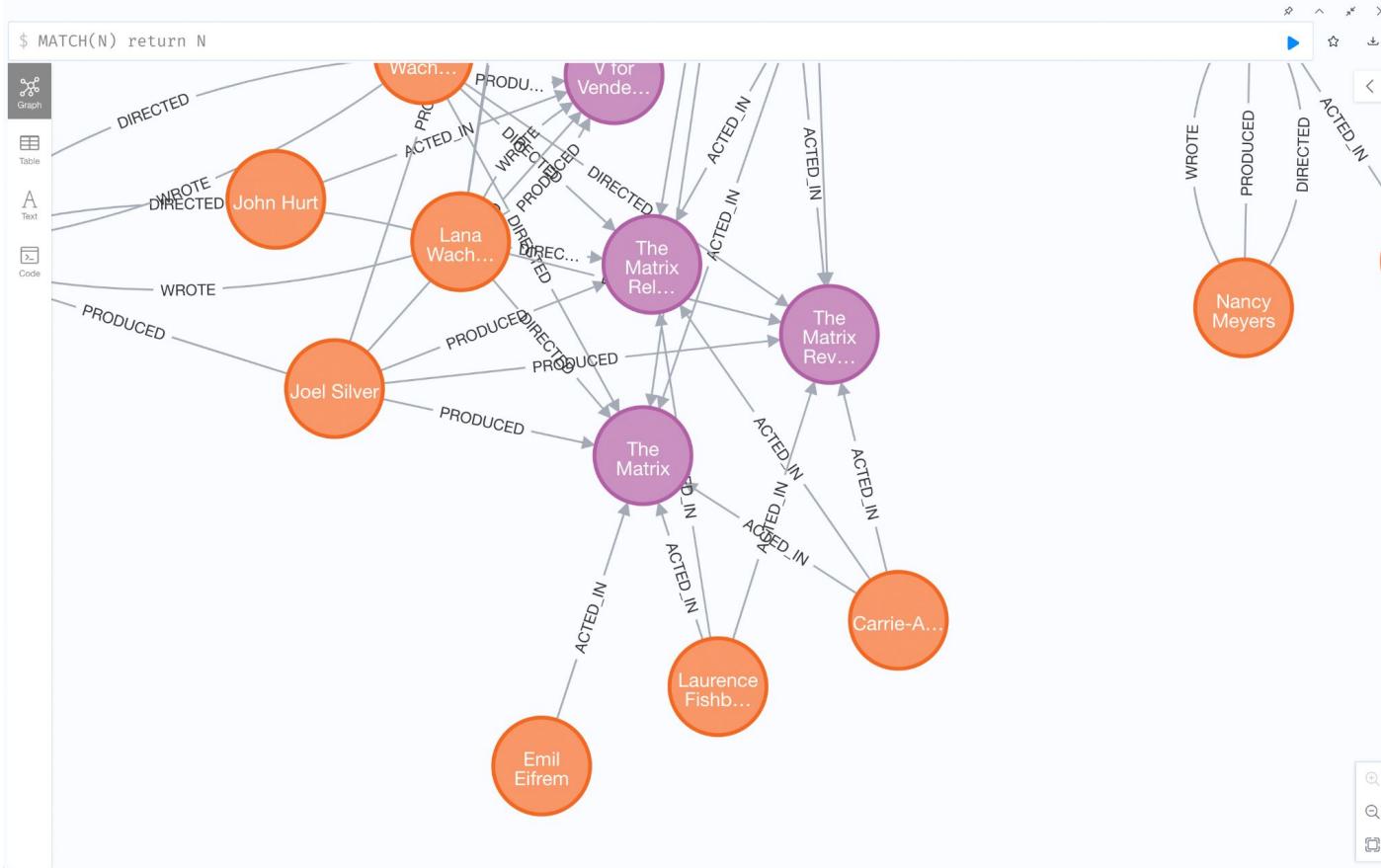
www.neo4j.com/aura



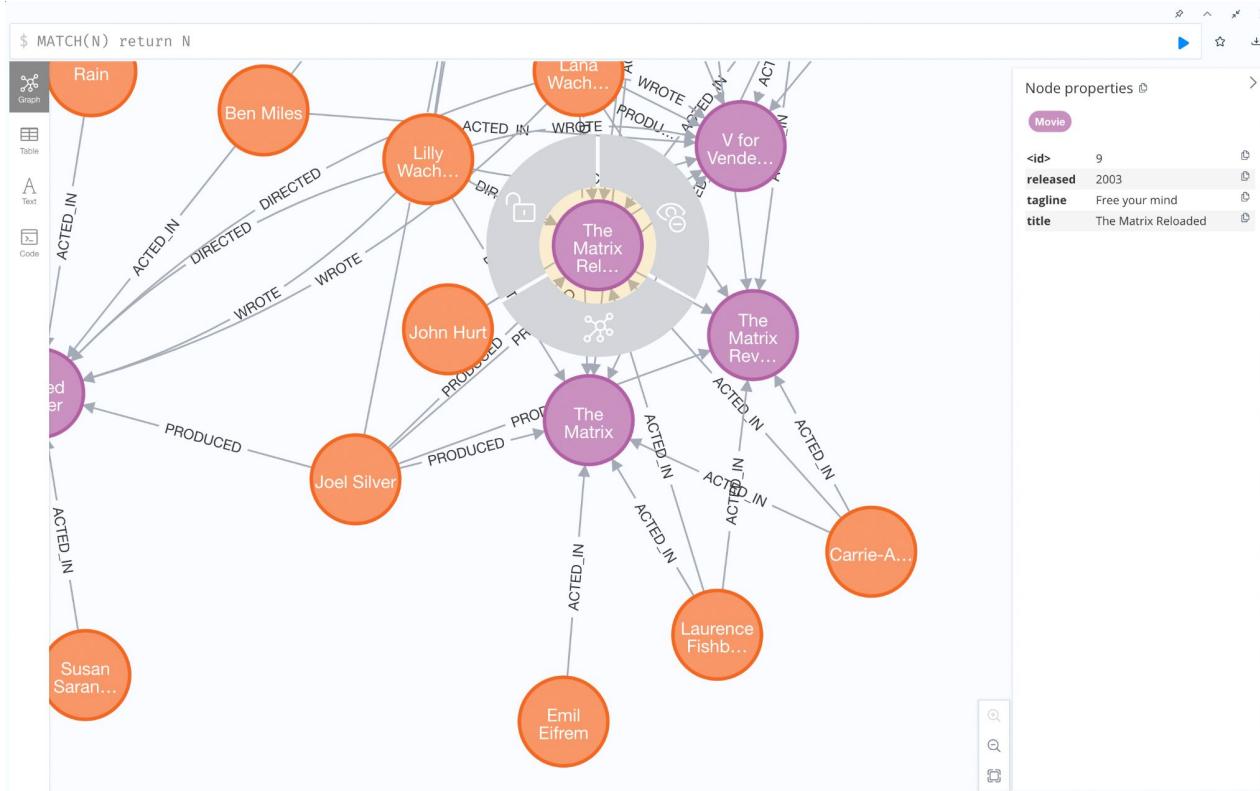
Graphs Explained



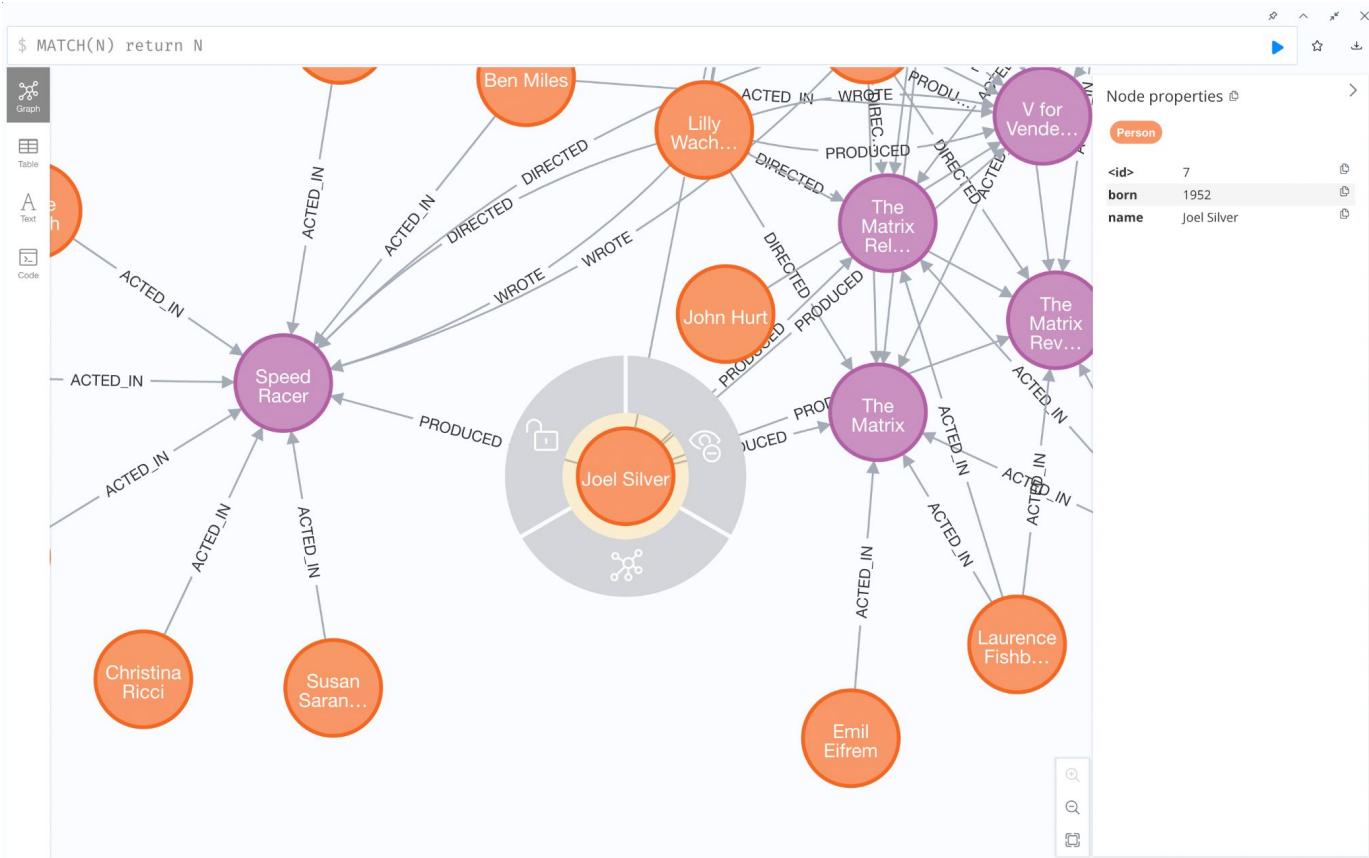
Graphs Explained



Graphs Explained



Graphs Explained



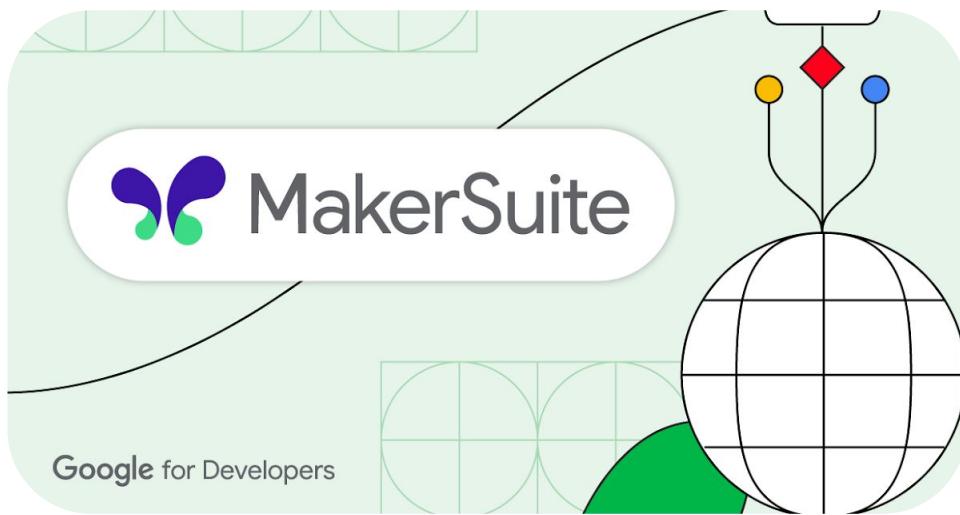
Populating data in Neo4j AuraDB

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

MakerSuite Explained

Getting started with MakerSuite

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



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



 Data prompt

A table that uses rows and columns to organize prompts



 Chat prompt

A template for back-and-forth chatbot conversations



Explore the full capabilities of the PaLM API.

 Model tuning

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



 Start developing

Call the PaLM API directly from your code.



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

○ ○ ○

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 

 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 reload?

 Test your prompt 

INPUT	OUTPUT
1 Name the director of the movie The Matrix reload?	Run to get output

 Add test example

 Run  Text Bison  0.7  Text preview 338 / 8196

▼ Test your prompt ⓘ

INPUT input

1 Name the director of the movie The Matrix reloaded?

OUTPUT

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

⊕ Add test example

MakerSuite

Untitled prompt 

 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     'top_p': 0.95
16 }
```

Test your prompt 

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

 Run  Text Bison  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

Building a Q&A Chatbot

<https://dev.neo4j.com/graph-nlp-workshop>



Scan me

: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))



Scan me