Knowledge Graphs for RAG

Knowledge Graphs Fundamentals

Nodes are data records


```
(Person)
```

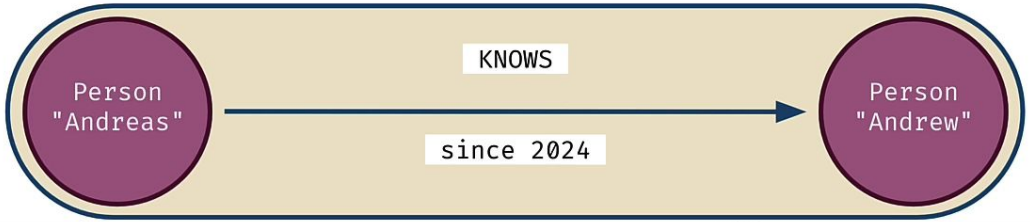
Add another person


```
(Person) , (Person)
```

Relationships are also data records


```
(Person)-[KNOWS]→(Person)
```

Relationships *contain* two nodes



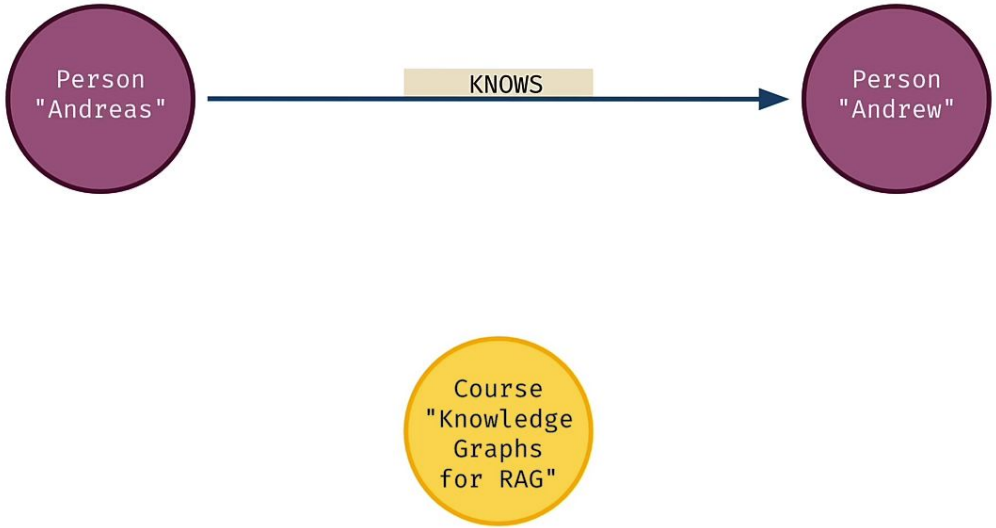
The nodes are "in" a relationship, which itself has properties.

Unlike an OO-perspective or even relational model, where the entity records would "know" about each other with direct references or foreign keys.

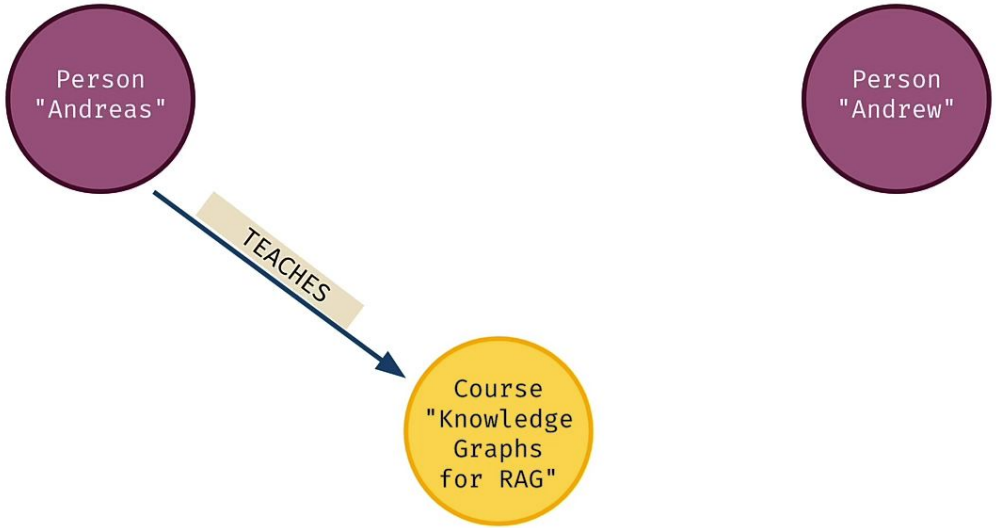
This has amazing implications for modeling and schema.

Graph types are composable!

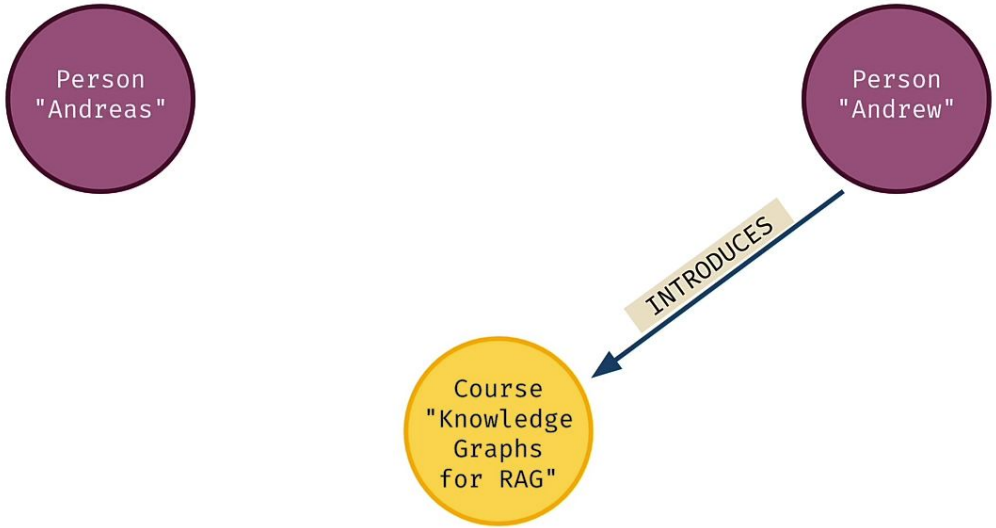
```
(Person)-[KNOWS]→(Person)
```


Person
"Andreas"
KNOWS
Person
"Andrew"
Course
"Knowledge
Graphs
for RAG"

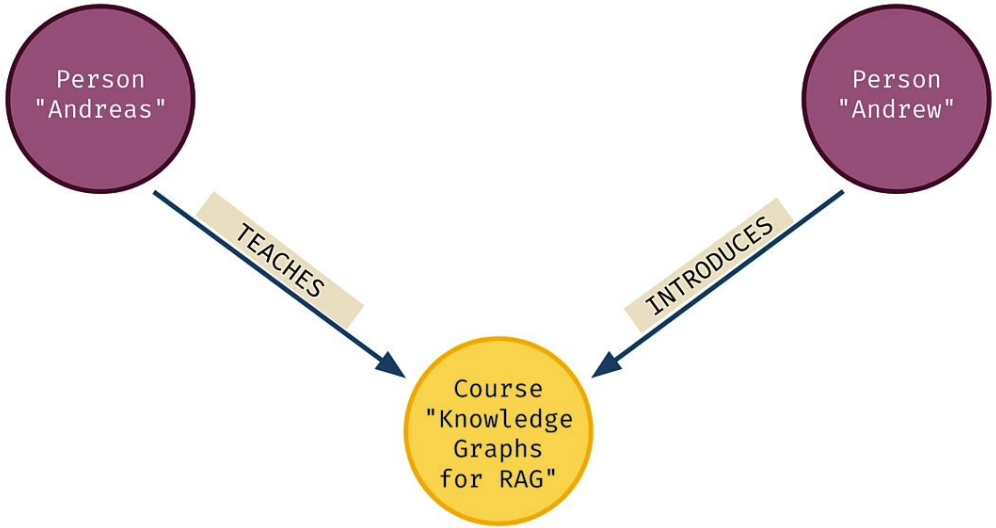
(Course)



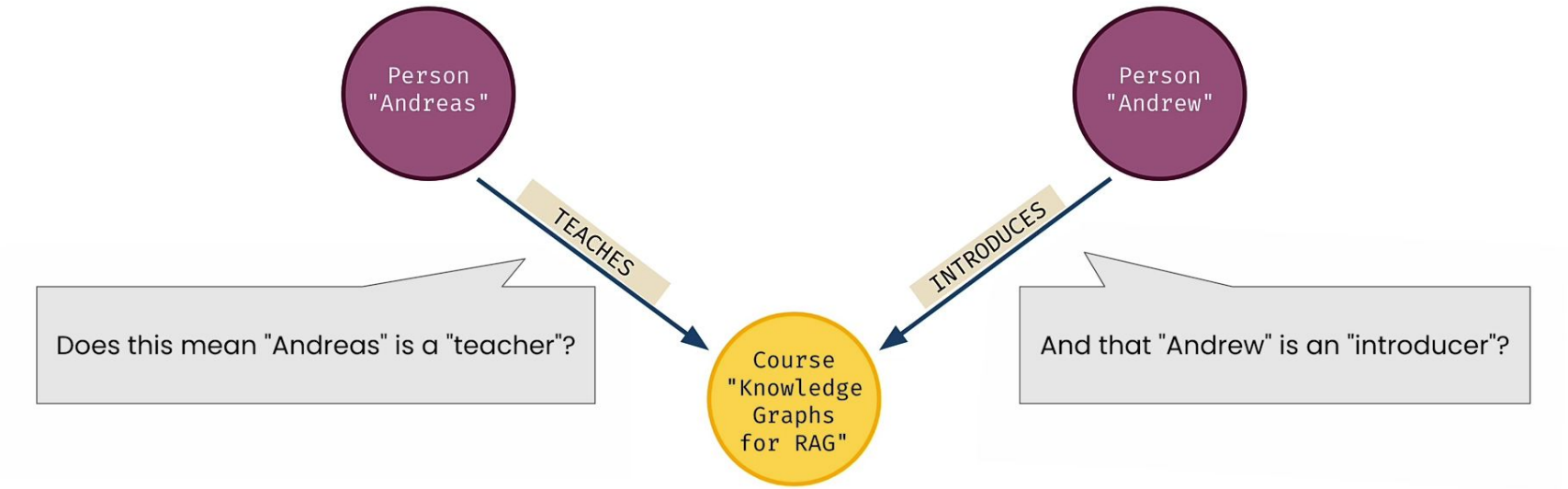

```
(Person)-[TEACHES]→(Course)
```



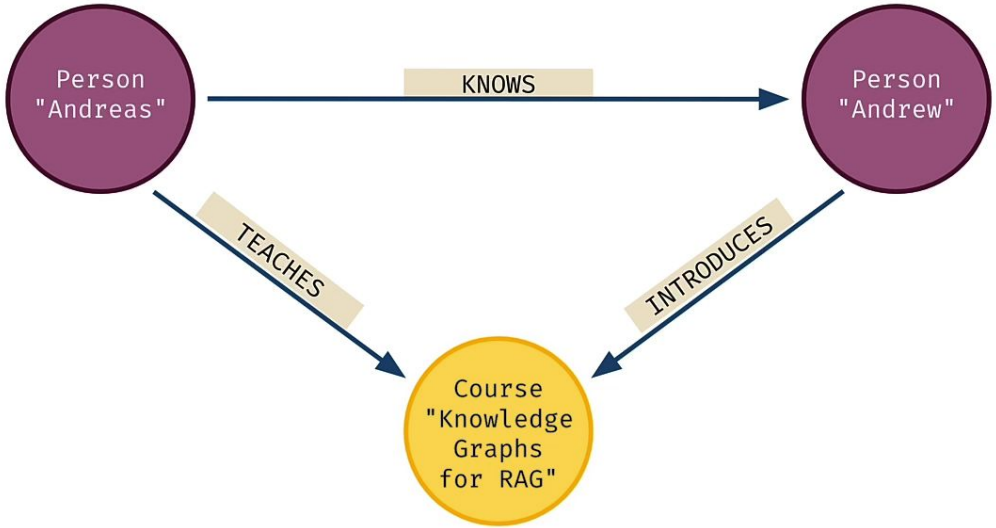
(Person)-[INTRODUCES]→(Course)




```
(Person)-[TEACHES]→(Course)←[INTRODUCES]-(Person)
```

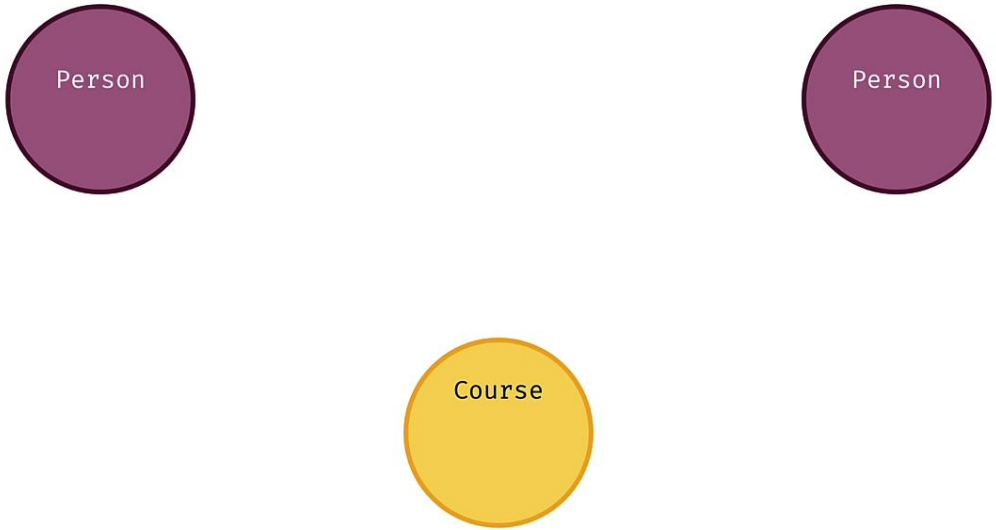



```
(Person)-[TEACHES]→(Course)←[INTRODUCES]-(Person)
```

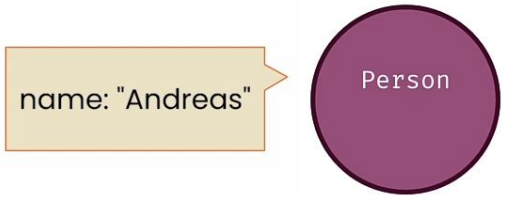


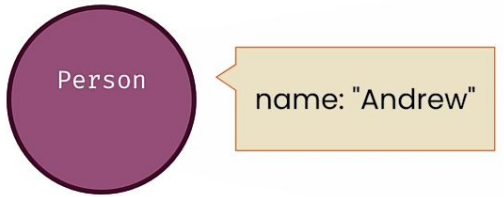

```
(Person)-[TEACHES]→(Course)←[INTRODUCES]-(Person)
```

Formally, nodes have labels

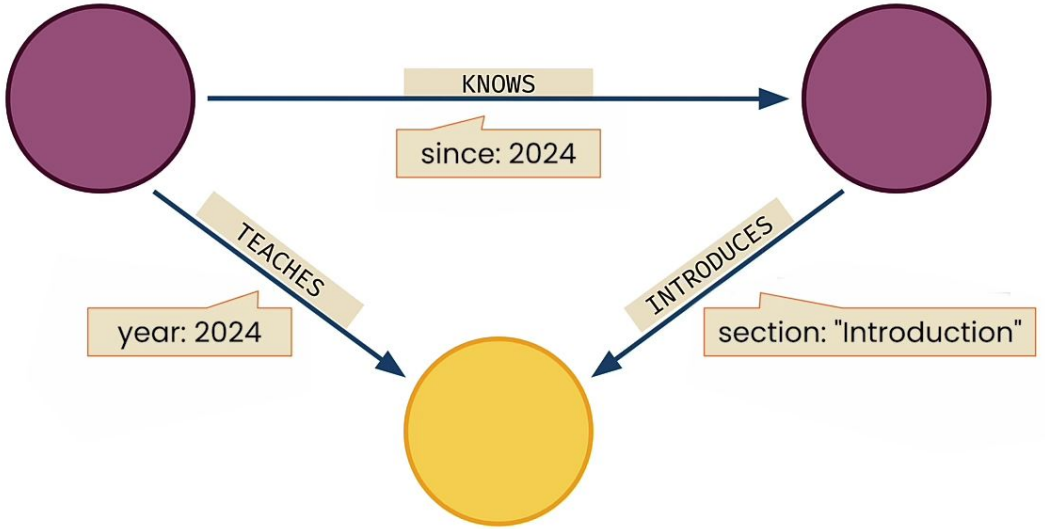


Formally, nodes are data record with key/value properties





Relationships are also data records.
They have a direction, a type, and properties



What is a Knowledge Graph?

What is a Knowledge Graph?

A knowledge graph is a database that stores information in **nodes** and **relationships**.

What is a Knowledge Graph?

A knowledge graph is a database that stores information in **nodes** and **relationships**.

Both nodes and relationships can have **properties**.

What is a Knowledge Graph?

A knowledge graph is a database that stores information in **nodes** and **relationships**.

Both nodes and relationships can have **properties**.

Nodes can be given **labels** to group them together.

What is a Knowledge Graph?

A knowledge graph is a database that stores information in **nodes** and **relationships**.

Both nodes and relationships can have **properties**.

Nodes can be given **labels** to group them together.

Relationships always have a **type** and a **direction**.

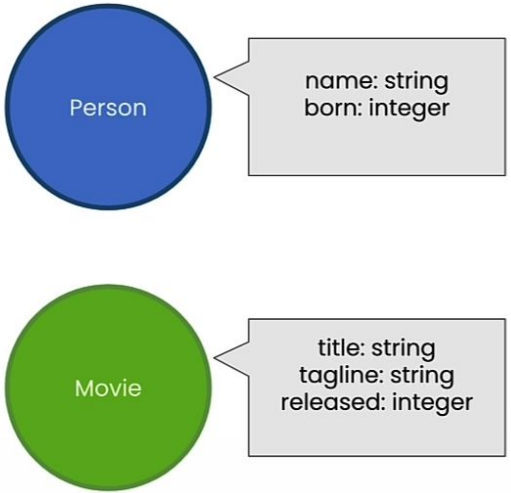
Knowledge Graphs for RAG

Lesson 2: Querying Knowledge Graphs

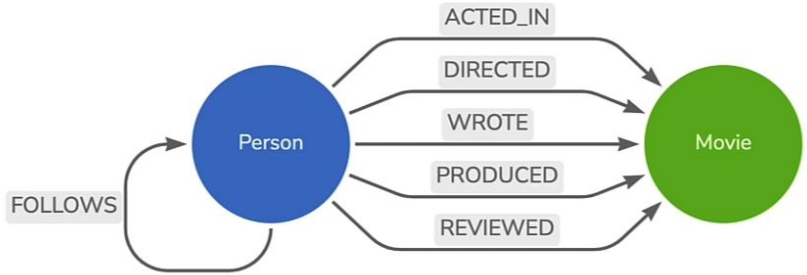
Movie Knowledge Graph - Person acted in movie


```
(Person)-[ACTED_IN]→(Movie)
```

Node properties



All relationships between a Person and a Movie



Knowledge Graphs for RAG

Lesson 3: Preparing Text Data for RAG

Knowledge Graphs for RAG

Lesson 4: Constructing a Knowledge Graph from Text Documents

SEC filing data

SEC filing data

- Companies are required to file many financial reports with the SEC each year

SEC filing data

- Companies are required to file many financial reports with the SEC each year
- An important form is the **10-K**, which is an annual report of the companies activities

SEC filing data

- Companies are required to file many financial reports with the SEC each year
- An important form is the **10-K**, which is an annual report of the companies activities
- These forms are public records, and can be accessed through the SEC's **EDGAR** database

Data cleaning

Completed 10-K forms are available to download as text files that contain XML components

Data cleaning

Completed 10-K forms are available to download as text files that contain XML components

In order to work with them, the following cleaning steps were applied

Data cleaning

Completed 10-K forms are available to download as text files that contain XML components

In order to work with them, the following cleaning steps were applied

- Cleaned up the files using regex

Data cleaning

Completed 10-K forms are available to download as text files that contain XML components

In order to work with them, the following cleaning steps were applied

- Cleaned up the files using regex
- Parsed XML into python data structures using Beautiful Soup

Data cleaning

Completed 10-K forms are available to download as text files that contain XML components

In order to work with them, the following cleaning steps were applied

- Cleaned up the files using regex
- Parsed XML into python data structures using Beautiful Soup

Data cleaning

Completed 10-K forms are available to download as text files that contain XML components

In order to work with them, the following cleaning steps were applied

- Cleaned up the files using regex
- Parsed XML into python data structures using Beautiful Soup
- Extracted CIK (Central Index Key) ID which is a company identifier used by the SEC

Data cleaning

Completed 10-K forms are available to download as text files that contain XML components

In order to work with them, the following cleaning steps were applied

- Cleaned up the files using regex
- Parsed XML into python data structures using Beautiful Soup
- Extracted CIK (Central Index Key) ID which is a company identifier used by the SEC
- Extracted specific sections of the form (Items 1, 1a, 7, and 7a)

Data cleaning

Completed 10-K forms are available to download as text files that contain XML components

In order to work with them, the following cleaning steps were applied

- Cleaned up the files using regex
- Parsed XML into python data structures using Beautiful Soup
- Extracted CIK (Central Index Key) ID which is a company identifier used by the SEC
- Extracted specific sections of the form (Items 1, 1a, 7, and 7a)

You can look in the data directory in the notebook if you'd like to examine the cleaned data for yourself.

Plan of attack

Plan of attack

1. Split form sections into chunks using a Langchain textsplitter

Plan of attack

1. Split form sections into chunks using a Langchain textsplitter

2. Create a graph where each chunk is a node, adding chunk metadata as properties

Plan of attack

1. Split form sections into chunks using a Langchain textsplitter
2. Create a graph where each chunk is a node, adding chunk metadata as properties
3. Create a vector index

Plan of attack

1. Split form sections into chunks using a Langchain textsplitter
2. Create a graph where each chunk is a node, adding chunk metadata as properties
3. Create a vector index
4. Calculate the text embedding vector for each chunk and populate the index

Plan of attack

1. Split form sections into chunks using a Langchain textsplitter
2. Create a graph where each chunk is a node, adding chunk metadata as properties
3. Create a vector index
4. Calculate the text embedding vector for each chunk and populate the index
5. Use similarity search to find relevant chunks!

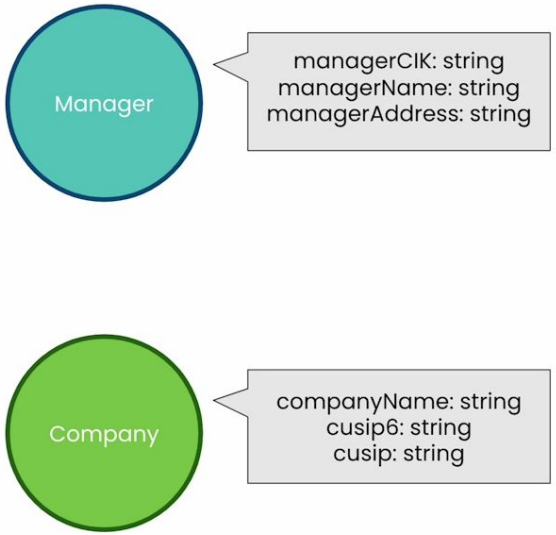
Knowledge Graphs for RAG

Lesson 5: Adding Relationships to the SEC Knowledge Graph

Knowledge Graphs for RAG

Lesson 6: Expanding the SEC Knowledge Graph

New Nodes from Form 13



Knowledge Graphs for RAG

Lesson 7: Chatting with the Knowledge Graph

How did you create a knowledge graph?

How did you create a knowledge graph?

Start with a Minimum Viable Graph (MVG), then extract, enhance, expand, and repeat to grow the graph

How did you create a knowledge graph?

Start with a Minimum Viable Graph (MVG), then extract, enhance, expand, and repeat to grow the graph

Extract: identify interesting information

How did you create a knowledge graph?

Start with a Minimum Viable Graph (MVG), then extract, enhance, expand, and repeat to grow the graph

Extract: identify interesting information

Enhance: supercharge the data

How did you create a knowledge graph?

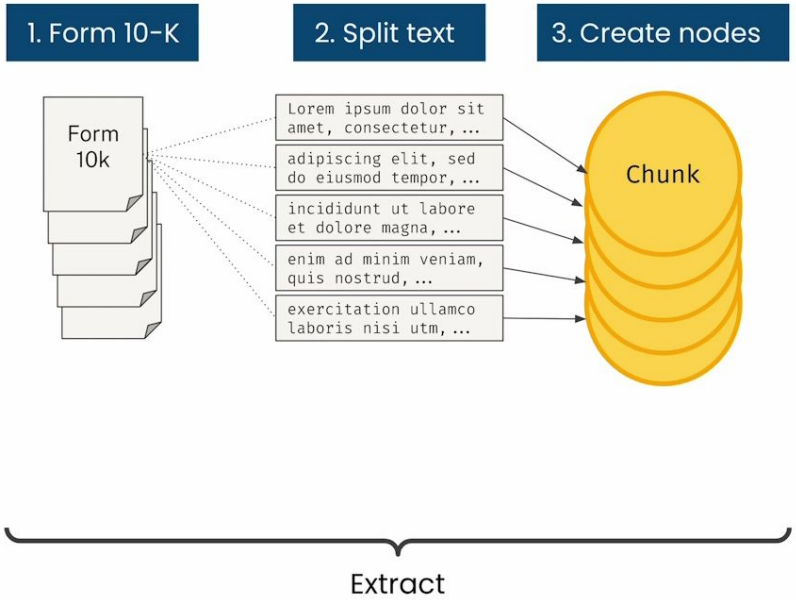
Start with a Minimum Viable Graph (MVG), then extract, enhance, expand, and repeat to grow the graph

Extract: identify interesting information

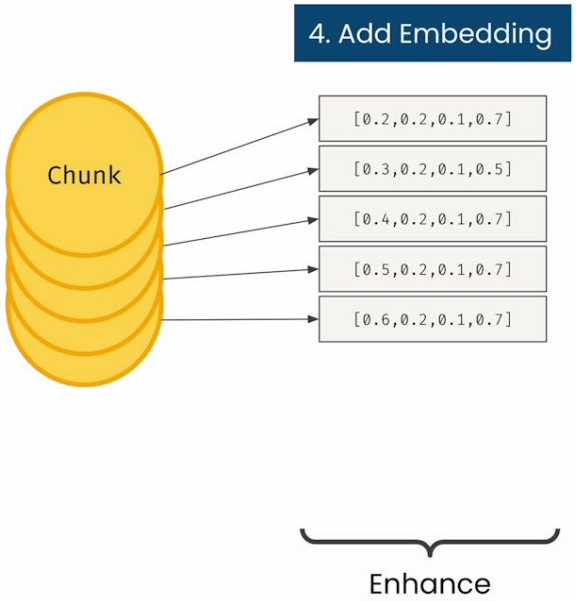
Enhance: supercharge the data

Expand: connect information to expand context

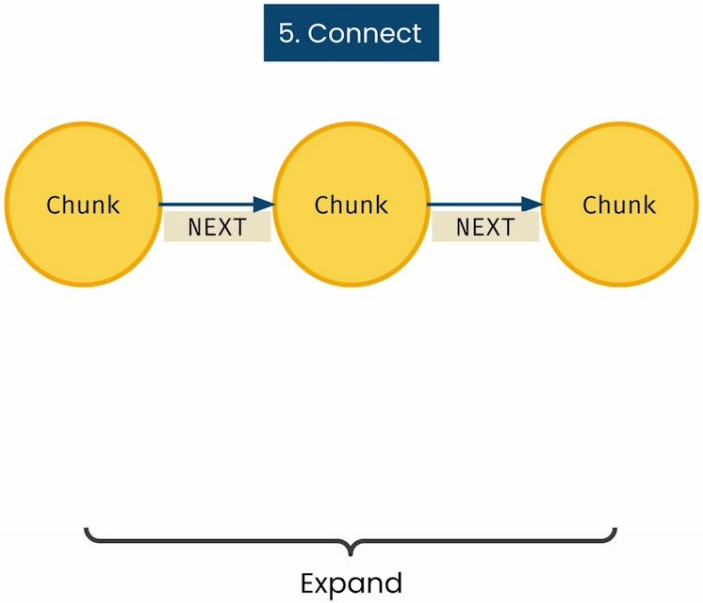
From text source to MVG



From text source to MVG



From text source to MVG



Extract, Enhance, Expand with SEC Forms

	Chunked Text Nodes	Form 10K Nodes	Companies	Management Firms
Source	Form 10K json	(:Chunk)	Form 13 CSV	Form 13 CSV
1. Extract	`(:Chunk)`	`(:Form)`	`(:Company)`	`(:Manager)`
2. Enhance	Vector embedding of text	Unique chunk IDs	Unique company cusip6	Full-text index of names
3. **Expand**	`(Chunk) -[NEXT]→ (Chunk)`	`(Chunk) -[PART_OF]→ (Form)`	`(Company) -[FILED]→ (Form)`	`(Manager) -[OWNS_STOCK_IN]→ (Company)`

Extract, Enhance, Expand with SEC Forms

	Chunked Text Nodes	Form 10K Nodes	Companies	Management Firms
Source	Form 10K json	(:Chunk)	Form 13 CSV	Form 13 CSV
1. Extract	`(:Chunk)`	`(:Form)`	`(:Company)`	`(:Manager)`
2. Enhance	Vector embedding of text	Unique chunk IDs	Unique company cusip6	Full-text index of names
3. **Expand**	`(Chunk) -[NEXT]→ (Chunk)`	`(Chunk) -[PART_OF]→ (Form)`	`(Company) -[FILED]→ (Form)`	`(Manager) -[OWNS_STOCK_IN]→ (Company)`

You can continue to grow the knowledge graph...

Extract, Enhance, Expand with SEC Forms

	Chunked Text Nodes	Form 10K Nodes	Companies	Management Firms
Source	Form 10K json	(:Chunk)	Form 13 CSV	Form 13 CSV
1. Extract	`(:Chunk)`	`(:Form)`	`(:Company)`	`(:Manager)`
2. Enhance	Vector embedding of text	Unique chunk IDs	Unique company cusip6	Full-text index of names
3. **Expand**	`(Chunk) -[NEXT]→ (Chunk)`	`(Chunk) -[PART_OF]→ (Form)`	`(Company) -[FILED]→ (Form)`	`(Manager) -[OWNS_STOCK_IN]→ (Company)`

You can continue to grow the knowledge graph...

- cross-link Companies that mention each other

Extract, Enhance, Expand with SEC Forms

	Chunked Text Nodes	Form 10K Nodes	Companies	Management Firms
Source	Form 10K json	(:Chunk)	Form 13 CSV	Form 13 CSV
1. Extract	`(:Chunk)`	`(:Form)`	`(:Company)`	`(:Manager)`
2. Enhance	Vector embedding of text	Unique chunk IDs	Unique company cusip6	Full-text index of names
3. **Expand**	`(Chunk) -[NEXT]→ (Chunk)`	`(Chunk) -[PART_OF]→ (Form)`	`(Company) -[FILED]→ (Form)`	`(Manager) -[OWNS_STOCK_IN]→ (Company)`

You can continue to grow the knowledge graph...

- cross-link Companies that mention each other
- add People, Places, Topics extracted from text

Extract, Enhance, Expand with SEC Forms

	Chunked Text Nodes	Form 10K Nodes	Companies	Management Firms
Source	Form 10K json	(:Chunk)	Form 13 CSV	Form 13 CSV
1. Extract	`(:Chunk)`	`(:Form)`	`(:Company)`	`(:Manager)`
2. Enhance	Vector embedding of text	Unique chunk IDs	Unique company cusip6	Full-text index of names
3. **Expand**	`(Chunk) -[NEXT]→ (Chunk)`	`(Chunk) -[PART_OF]→ (Form)`	`(Company) -[FILED]→ (Form)`	`(Manager) -[OWNS_STOCK_IN]→ (Company)`

You can continue to grow the knowledge graph...

- cross-link Companies that mention each other
- add People, Places, Topics extracted from text
- add more Form data, or other related sources

Extract, Enhance, Expand with SEC Forms

	Chunked Text Nodes	Form 10K Nodes	Companies	Management Firms
Source	Form 10K json	(:Chunk)	Form 13 CSV	Form 13 CSV
1. Extract	`(:Chunk)`	`(:Form)`	`(:Company)`	`(:Manager)`
2. Enhance	Vector embedding of text	Unique chunk IDs	Unique company cusip6	Full-text index of names
3. **Expand**	`(Chunk) -[NEXT]→ (Chunk)`	`(Chunk) -[PART_OF]→ (Form)`	`(Company) -[FILED]→ (Form)`	`(Manager) -[OWNS_STOCK_IN]→ (Company)`

You can continue to grow the knowledge graph...

- cross-link Companies that mention each other
- add People, Places, Topics extracted from text
- add more Form data, or other related sources
- add Users to refine relevance and enable feedback