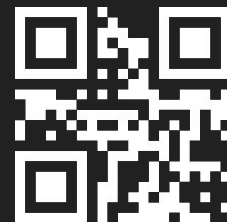# Knowledge Graphs

Article + LLM = Cypher Query

Groff Dev

Schema: Understanding the Purpose of your Graph

- The first step in designing a schema is understanding the core questions and insights you need from your data.

- Think about the entities that are central to your analysis. In our case, we are analyzing companies and their funding, so **Company** and **Investor** naturally become the core entities (nodes).

Groff Dev

Schema: Identifying Key Relationships

- Once you have your core entities, consider how they are connected. Relationships (edges) between entities represent the interactions or events you are interested in tracking.

- For funding data, the key relationship is **FUNDED_BY**: a company is funded by an investor. This relationship is the link that ties these entities together and is crucial for understanding investment dynamics.

Schema: Deciding What Becomes a Node

- Nodes should represent entities that you want to query independently or group by specific properties. Entities that have distinct properties and that you would like to analyze should be nodes.

- In our schema, **Company** and **Investor** are nodes because we want to analyze companies independently of each other and understand their investors. Each has its own set of properties such as company name, valuation, and sector for companies, and investor name for investors.

Groff Dev

Schema: Determining Your Relationships (Edges)

- Edges are the actions, connections, or events that link your nodes together. They help explain the interactions between entities.

- The **FUNDED_BY** relationship describes how the company receives funding from the investor. It also includes relevant properties like the funding **type** (e.g., Series B), **amount**, and **date** of funding.

- Relationships often carry properties of their own that give context to the interaction (e.g., the amount of investment).

Groff Dev

Schema: Choosing the Right Level of Granularity

- It's important to strike a balance between simplicity and the level of detail your queries require.

- We could have chosen to make **Series B** a node, but keeping it as a property on the **FUNDED_BY** relationship simplifies the graph without losing key information.

Groff Dev

Schema: Our Decision For This Use Case

- **Nodes:** We chose **Company** and **Investor** as our primary nodes because they are the key entities we care about.

- **Edges:** We made **FUNDED_BY** the connecting relationship, with properties for **type** (Series B), **amount**, and **date**. This structure allows us to track not just that a company is funded, but the specifics of the funding round.

Groff Dev

Using an LLM to create the Cypher Query

Prompt:

"I have the following article about a company and its investors. Could you generate a Cypher query that creates a **Company** node for the company, **Investor** nodes for each investor, and **FUNDED_BY** relationships between the company and its investors? Please make sure the funding type, amount, and date are stored as properties on the **FUNDED_BY** relationship.

Here is the article:

[Insert the article here]

Please generate the Cypher query based on the information in the article."

An example funding article

- Elon Musk's xAI, a 1-year-old foundation model company based in Burlingame, California, raised a $6 billion Series B funding at a $24 billion value. Participants in the round include Valor Equity Partners, Vy Capital, Andreessen Horowitz, Sequoia Capital, Fidelity, Prince Alwaleed Bin Talal and Kingdom Holding Co. among others.

- Company: xAI
- $6 Billion Series B
- Investors:
  - Valor Equity Partners
  - Vy Capital
  - Andreessen Horowitz
  - Sequoia Capital
  - Fidelity
  - Prince Alwaleed Bin Talal
  - Kingdom Holding Co

Using an LLM to create the Cypher Query

```cypher
CREATE (company:Company {
    name: "xAI",
    headquarters: "Burlingame, California",
    valuation: 24000000000,
    founded: 2023,
    sector: "AI"
})

WITH company

CREATE (investor1:Investor {name: "Valor Equity Partners"}),
       (investor2:Investor {name: "Vy Capital"}),
       (investor3:Investor {name: "Andreessen Horowitz"}),
       (investor4:Investor {name: "Sequoia Capital"}),
       (investor5:Investor {name: "Fidelity"}),
       (investor6:Investor {name: "Prince Alwaleed Bin Talal"}),
       (investor7:Investor {name: "Kingdom Holding Co."})
```

```cypher
WITH company, investor1, investor2, investor3, investor4, investor5, investor6, investor7

CREATE (company)-[:FUNDED_BY {amount: 6000000000, date: "2024-05", type: "Series B"}]->(investor1),
       (company)-[:FUNDED_BY {amount: 6000000000, date: "2024-05", type: "Series B"}]->(investor2),
       (company)-[:FUNDED_BY {amount: 6000000000, date: "2024-05", type: "Series B"}]->(investor3),
       (company)-[:FUNDED_BY {amount: 6000000000, date: "2024-05", type: "Series B"}]->(investor4),
       (company)-[:FUNDED_BY {amount: 6000000000, date: "2024-05", type: "Series B"}]->(investor5),
       (company)-[:FUNDED_BY {amount: 6000000000, date: "2024-05", type: "Series B"}]->(investor6),
       (company)-[:FUNDED_BY {amount: 6000000000, date: "2024-05", type: "Series B"}]->(investor7);
```
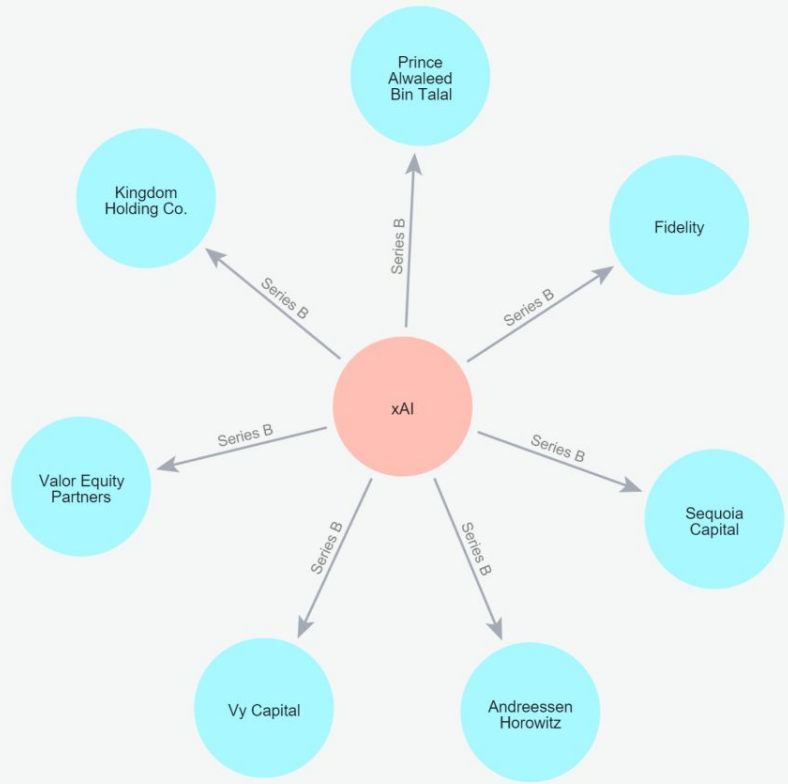
Graph | Table | RAW

Results overview

Nodes (8)

* (8)   Company (1)   Investor (7)

Relationships (7)

* (7)   FUNDED_BY (7)

Nodes in graph:
- Prince Alwaleed Bin Talal
- Kingdom Holding Co.
- Fidelity
- Valor Equity Partners
- xAI
- Sequoia Capital
- Vy Capital
- Andreessen Horowitz

Relationships labeled "Series B" connect xAI to each investor.
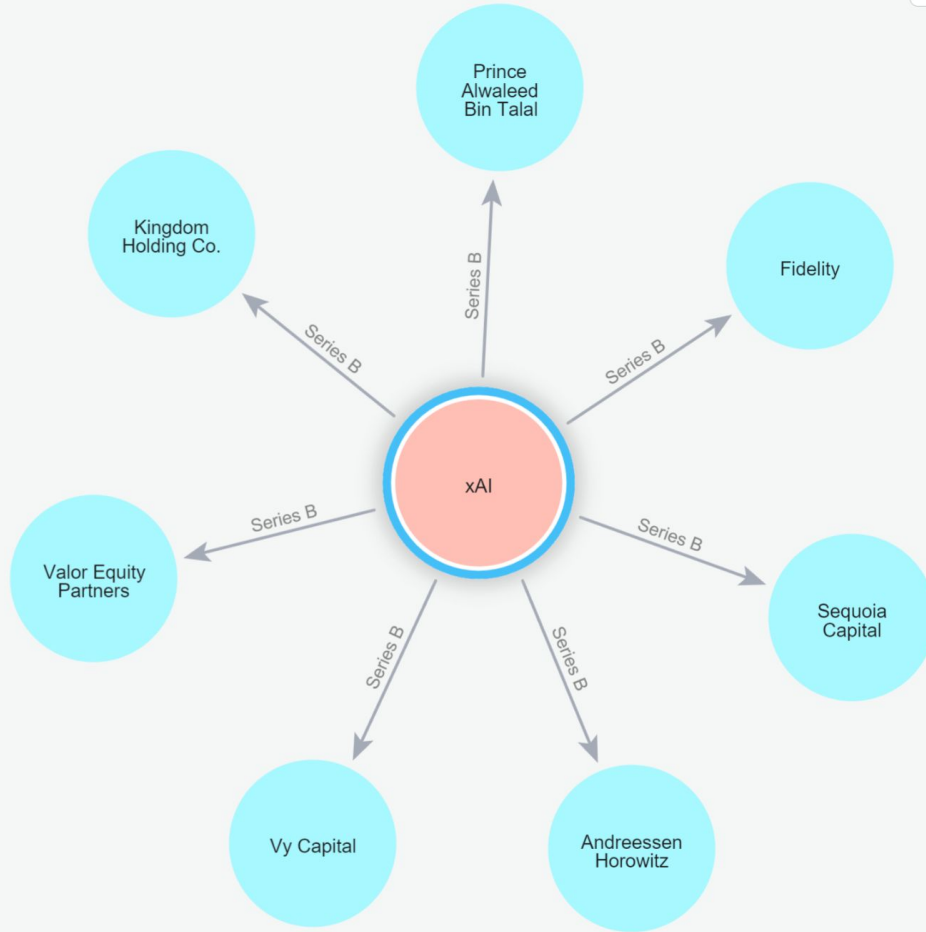
Node

Company

Node details

Company

Key	Value
<id>	8
headquarters	"Burlingame, California"
valuation	24000000000
name	"xAI"
founded	2023
sector	"AI"

Node

Investor
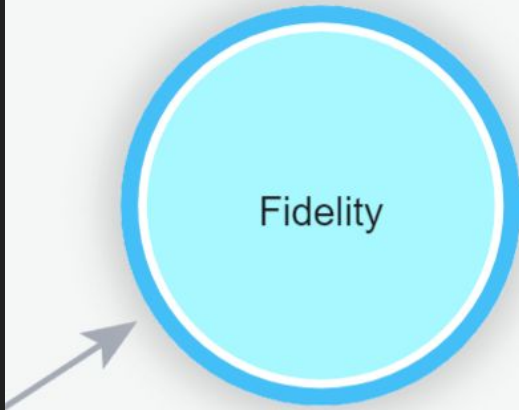
Node details

Investor

Key	Value	
<id>	4	
name	"Fidelity"	

Edge

FUNDED_BY
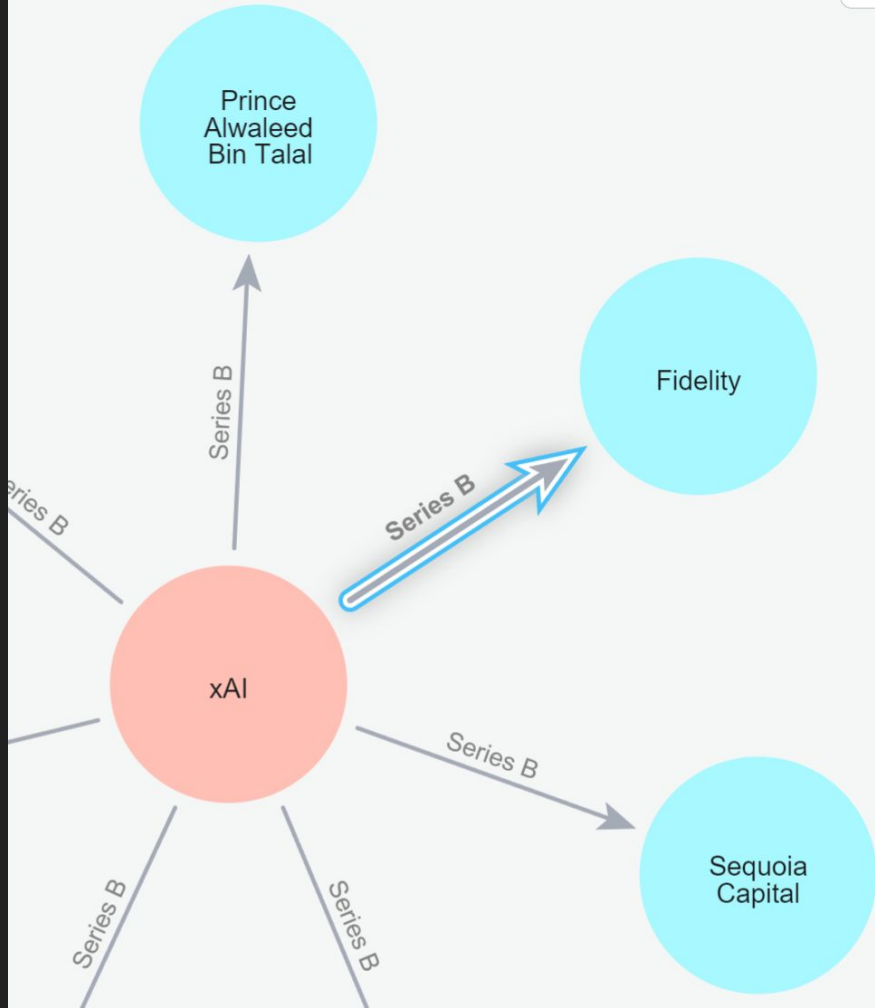
Relationship details

FUNDED_BY

Key	Value
<id>	1161929803373215752
date	"2024-05"
amount	6000000000
type	"Series B"

What kinds of questions could this Knowledge Graph answer?

- **Question**: What are all the funding rounds that a specific company has raised?

  - **Example Query**: "What are all the funding rounds raised by **xAI**?"

  - **Cypher**:

```cypher
MATCH (company:Company {name: "xAI"})-[f:FUNDED_BY]->(investor:Investor)
RETURN company.name, f.type, f.amount, f.date, investor.name;
```

- **Question**: What is the total funding raised by a company across all rounds?

  - **Example Query**: "What is the total amount raised by **xAI** across all funding rounds?"

  - **Cypher**:

```cypher
MATCH (company:Company {name: "xAI"})-[f:FUNDED_BY]->(investor:Investor)
RETURN company.name, SUM(f.amount) AS total_raised;
```

- **Question**: Which companies has a specific investor funded, and how much did they invest?

  - **Example Query**: "Which companies have **Sequoia Capital** invested in, and how much have they invested in each?"

  - **Cypher**:

```cypher
MATCH (investor:Investor {name: "Sequoia Capital"})<-[f:FUNDED_BY]-(company:Company)
RETURN company.name, f.type, f.amount, f.date;
```

- **Question**: Which companies are in a specific sector, and how much funding have they raised?

  - **Example Query**: "Which AI companies have raised funding, and what are their funding details?"

  - **Cypher**:

```cypher
MATCH (company:Company {sector: "AI"})-[f:FUNDED_BY]->(investor:Investor)
RETURN company.name, f.type, f.amount, f.date, investor.name;
```