kùzu

Towards Agentic Graph RAG

Combining the Power of Structured Outputs from LLMs with Graph Databases

Prashanth Rao
AI Engineer, Kùzu Inc. 🇨🇦

kuzudb.com

The AI Conference
San Francisco | 17 Sep 2025

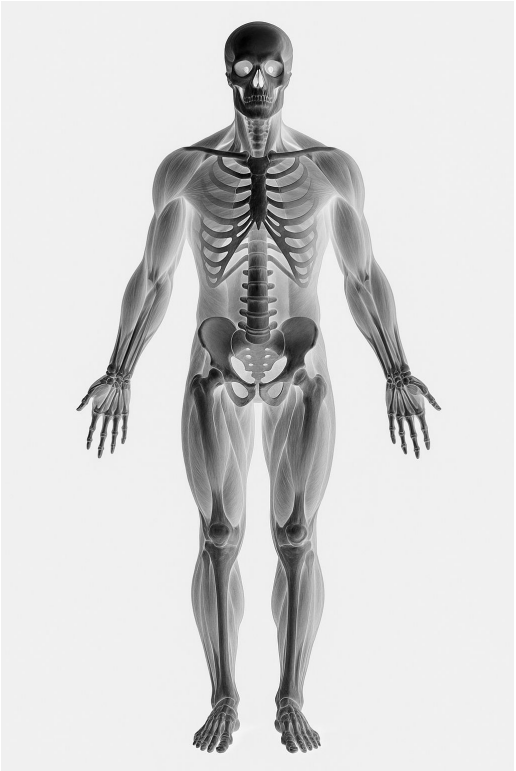
RAG bridges knowledge gaps

kùzu

- Intertwined with the use of agents
- Most agent workflows are multi-step (observe, reflect, act)
- RAG is:
  - The foundation of memory (supports the agent's memory system)
  - An extension of the model's knowledge with external facts
  - An important source of *context* at each stage
- There are *many* flavours of RAG (not just chat bots or Q/A)

For autonomous retrieval systems, context is king

kùzu



Knowledge is the ***skeleton***

Agents are the ***muscles*** that decide how to move.

Knowledge graphs can be the bones (backbone?) of such systems.

How to make messy, unstructured data *queryable*?

kùzu

"How many patients in Massachusetts are allergic to shellfish?"

Ms. Marisol Rodríguez, who was born on December 1, 1995, and resides at her home at 547 Effertz Extension Unit 28, East Longmeadow, Massachusetts, 00000, is never married. She identifies Spanish as her primary language. She can be reached at her home phone number, which is 555-808-5474.

In the past, Marisol visited the ENCOMPASS HEALTH REHAB HOSPITAL OF WESTERN MASS on December 13, 2013, from 9:15:24 until 9:30:24 in Central European Standard Time for a postnatal visit. During this visit, Dr. Cletus Paucek, a primary performer, was the practitioner managing her care. You may reach out to Dr. Paucek at his work email, which is Cletus494.Paucek755@example.com.

Of note, Marisol has a confirmed low criticality allergy towards shellfish, first noted on June 2, 1997, at 22:15:24 in Central Eastern Standard Time. Reactions to an exposure to shellfish in her case have varied. In some instances, she experiences a moderate eruption of skin. In severe cases, she has experienced anaphylaxis. More often, her symptoms are mild and include itching and coughing. Another noted allergy intolerance is currently active, yet the clinical details are not specified here.

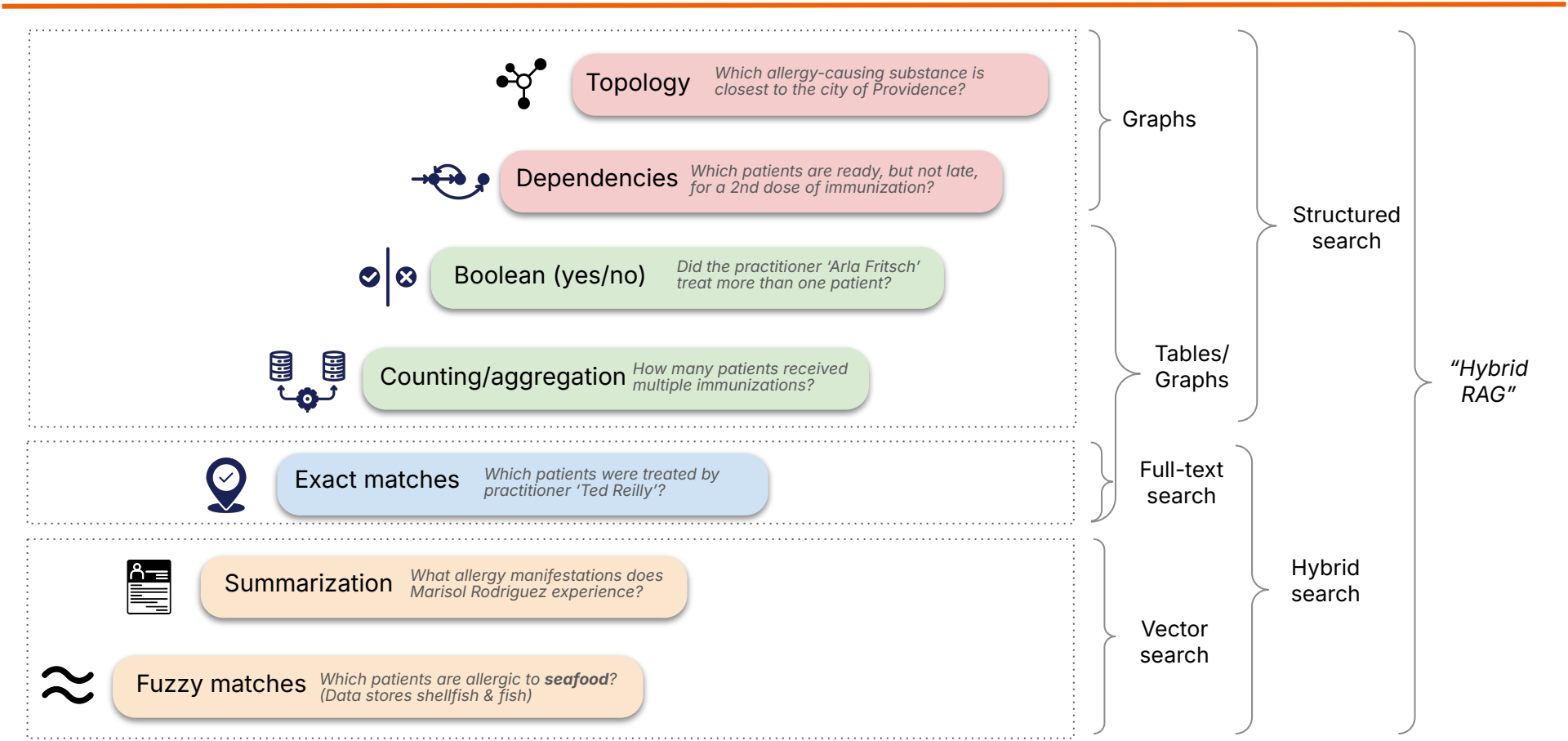

```
{
  "name": {
    "family": "Rodríguez",
    "given": [
      "Marisol"
    ],
    "prefix": "Ms."
  },
  "age": null,
  "gender": null,
  "birthDate": "1995-12-01",
  "address": {
    "line": "547 Effertz Extension Unit 28",
    "city": "East Longmeadow",
    "state": "Massachusetts",
    "postalCode": "00000",
    "country": "US"
  },
  "phone": "555-808-5474",
  "email": null,
  "maritalStatus": "NeverMarried",
  "primaryLanguage": "Spanish"
}
```

```
{
    "allergy": {
      "substance": [
        {
          "category": "food",
          "name": "shellfish",
          "manifestation": [
            "moderate eruption of skin",
            "anaphylaxis",
            "itching",
            "coughing"
          ]
        }
      ]
    }
}
```

```
{
    "practitioner": {
      "name": {
        "family": "Paucek",
        "given": [
          "Cletus"
        ],
        "prefix": "Dr."
      },
      "address": null,
      "phone": null,
      "email": "Cletus494.Paucek755@example.com"
    }
}
```

The best retrieval tool depends on the question

kùzu



Taking knowledge graphs to production

kùzu

Part 1: Graph construction 🚜

- **Schema definition**
- **Structured outputs**: Extract nodes, relationships and properties
- **Entity deduplication**: Identify and handle duplicate entities

Part 2: Retrieval and RAG 🧠

- **Graph RAG**: Use the knowledge graph as the retrieval layer (structured facts)
- **Vector-based RAG**: Use semantic similarity between the search query and the data to retrieve chunks
- **Hybrid RAG**: Combining various retrieval results via reranking

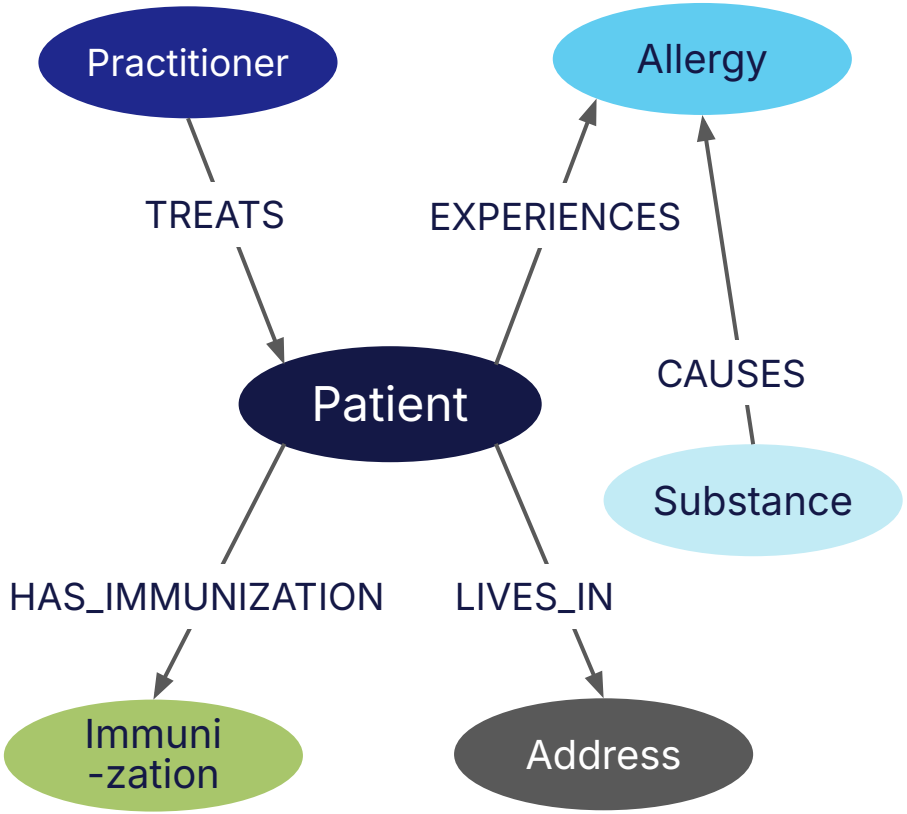
Part 1: Information Extraction

Information extraction pipeline

kùzu

Use case: Patient questions on unstructured patient notes

- Does using Graph RAG improve our ability to answer questions?
- The data lacks structure to start with
- Knowledge graphs help impose some degree of structure, baking in ***domain knowledge***
- *Paths* are a first-class citizen



From unstructured → structured data

kùzu

Frameworks like BAML, DSPy (and many others) allow us to generate ***high-quality*** structured outputs from LLMs

```python
from pydantic import BaseModel

class PersonNameAndTitle(BaseModel):
    family: str | None = Field(default=None, description="surname")
    given: list[str] | None = Field(
        default=None,
        description="Given names (first and middle names)",
    )
    prefix: str | None = Field(
        default=None, description="title of the person: Dr.
    )

class Address(BaseModel):
    line: str | None = Field(default=None, alias="street")
    city: str | None
    state: str | None
    postalCode: str | None
    country: Literal["US"] | None

class Patient(BaseModel):
    record_id: int | None = Field(default=None)
    name: PersonNameAndTitle | None
    age: int | None
    gender: Literal["male", "female"] | None
    birthDate: str | None = Field(default=None, description="ISO-8601 format for date")
    phone: str | None = Field(default=None, description="Phone number of the patient")
    email: str | None = Field(default=None, description="Email address of the patient")
    maritalStatus: Literal["Married", "Divorced", "Widowed", "NeverMarried"] | None
    address: Address | None = Field(
        default=None, description="Residence address of the patient"
    )
```

```python
import dspy

class PatientInfo(dspy.Signature):
    """
    Extract patient information from the given note.
    - Do not infer any information not explicitly mentioned.
    - If you are unsure about any field, leave it as None.
    """

    note: PatientNote = dspy.InputField()
    patient: Patient = dspy.OutputField()
```

```json
{
   "name": {
   "family": "Rodríguez",
   "given": [
     "Marisol"
    ],
    "prefix": "Ms."
  },
  "age": null,
  "gender": null,
  "birthDate": "1995-12-01",
  "address": {
    "line": "547 Effertz Extension Unit 28",
    "city": "East Longmeadow",
    "state": "Massachusetts",
    "postalCode": "00000",
    "country": "US"
  },
  "phone": "555-808-5474",
  "email": null,
  "maritalStatus": "NeverMarried",
  "primaryLanguage": "Spanish"
}
```

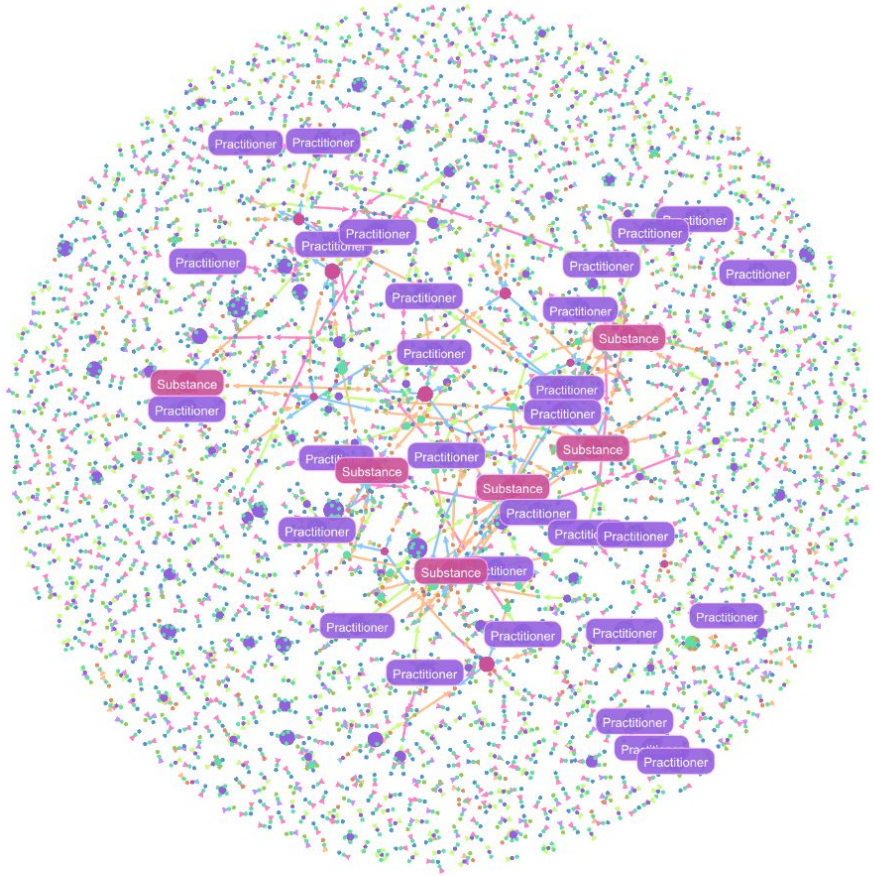
Nested JSON is a graph in disguise!

kùzu


```
{
  "name": {
    "family": "Rodríguez",
    "given": [
      "Marisol"
    ],
    "prefix": "Ms."
  },
  "age": null,
  "gender": null,
  "birthDate": "1995-12-01",
  "address": {
    "line": "547 Effertz Extension Unit 28",
    "city": "East Longmeadow",
    "state": "Massachusetts",
    "postalCode": "00000",
    "country": "US"
  },
  "phone": "555-808-5474",
  "email": null,
  "maritalStatus": "NeverMarried",
  "primaryLanguage": "Spanish"
}
```

```
{
    "allergy": {
      "substance": [
        {
          "category": "food",
          "name": "shellfish",
          "manifestation": [
            "moderate eruption of skin",
            "anaphylaxis",
            "itching",
            "coughing"
          ]
        }
      ]
    }
}
```

```
{
    "practitioner": {
      "name": {
        "family": "Paucek",
        "given": [
          "Cletus"
        ],
        "prefix": "Dr."
      },
      "address": null,
      "phone": null,
      "email": "Cletus494.Paucek755@example.com"
    }
}
```

How accurate is LM-based extraction in practice?

kùzu

Hugging Face

kishanbodybrain/test-fhir

- Synthetic dataset available on Hugging Face Datasets
- 2,726 human-annotated records in FHIR format
- *Easy to evaluate!*

Exact match accuracy

Model	BAML	DSPy
google/gemma-3-27b-it	91%	94%
openai/gpt-4.1-mini	95%	94%
google/gemini-2.5-flash	95%	96%
openai/gpt-4.1	⬆	⬆
google/gemini-2.5-pro	⬆	⬆

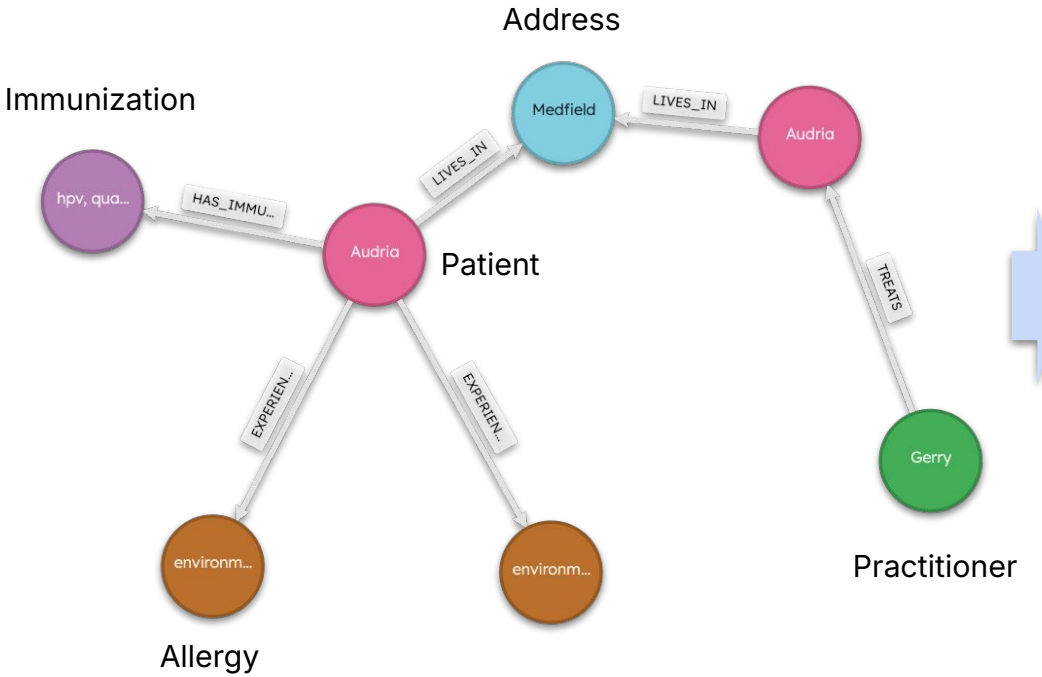
General-purpose models continue to improve (as does constrained generation)

github.com/prrao87/structured-outputs

Handling duplicates upon extraction

kùzu

"Audria Schinner, lives in Medfield, Massachusetts, treated by Gerry Block"




```python
import dspy

class DisambiguateEntities(dspy.Signature):
    """
    Return the reference `patient_id` that's most likely the
    same patient as the sample record.
    - The result must contain ONLY ONE reference record `id`
    - Also return the confidence level of the mapping based on your judgment.
    """

    sample: Patient = dspy.InputField(desc="A patient record")
    reference_records: list[Patient] = dspy.InputField(
        desc="A list of reference records"
    )
    output: int = dspy.OutputField(
      desc="Most similar reference record to the sample record"
    )
    confidence: Literal["high", "low"] = dspy.OutputField(
        desc="Confidence level of mapping"
    )
```

blog.kuzudb.com: A gentle introduction to DSPy for graph data enrichment

Part 2: Agentic RAG

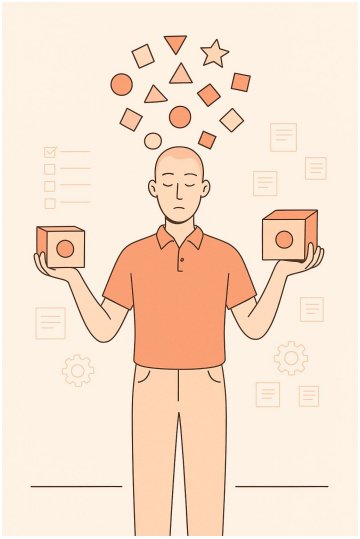
Graph RAG is inherently agentic

kùzu

Graph-based retrievers eventually become a suite of tools/techniques

Route the query to the appropriate tool

E.g. text2cypher or templated Cypher with parameters



Error handling and recovery

E.g. incorrect Cypher syntax or query terms generate by the LLM

Locate semantically relevant entry points

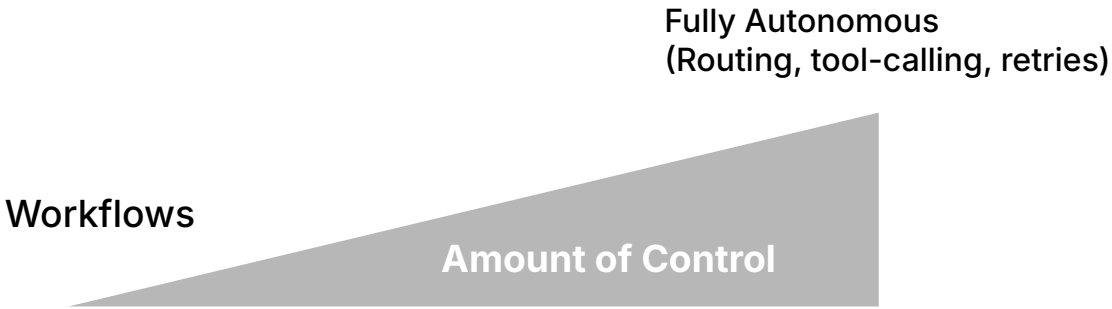
Fuzzy searches (based on embedding) can help narrow down on the right entry points to begin traversing paths

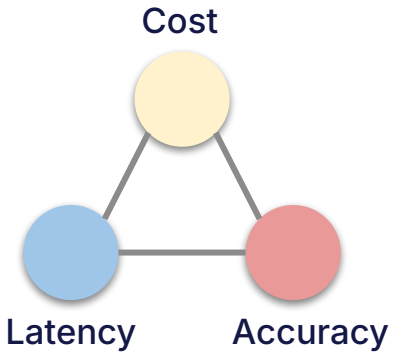
Reranking results using graphs

E.g. prioritized results based on centrality

Should everything be fully agentic?

kùzu

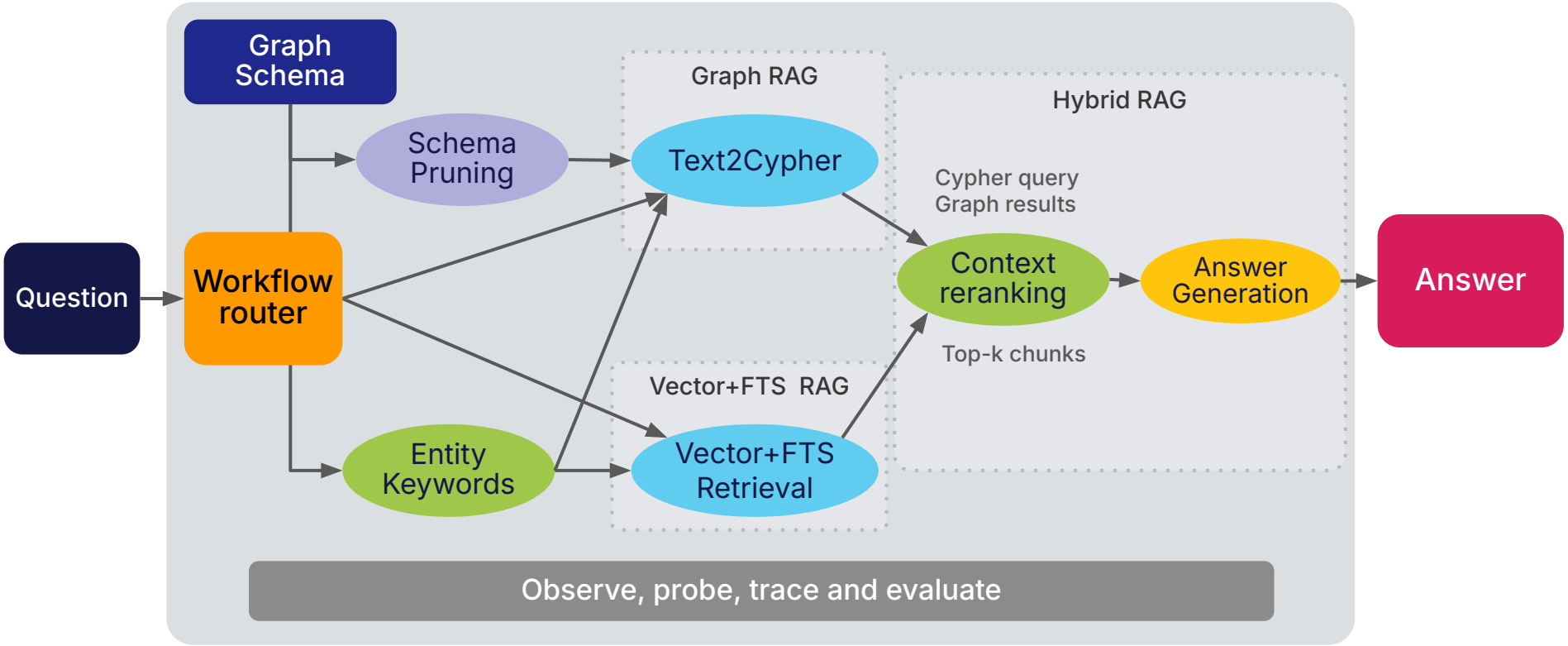




Sliding scale of autonomy:

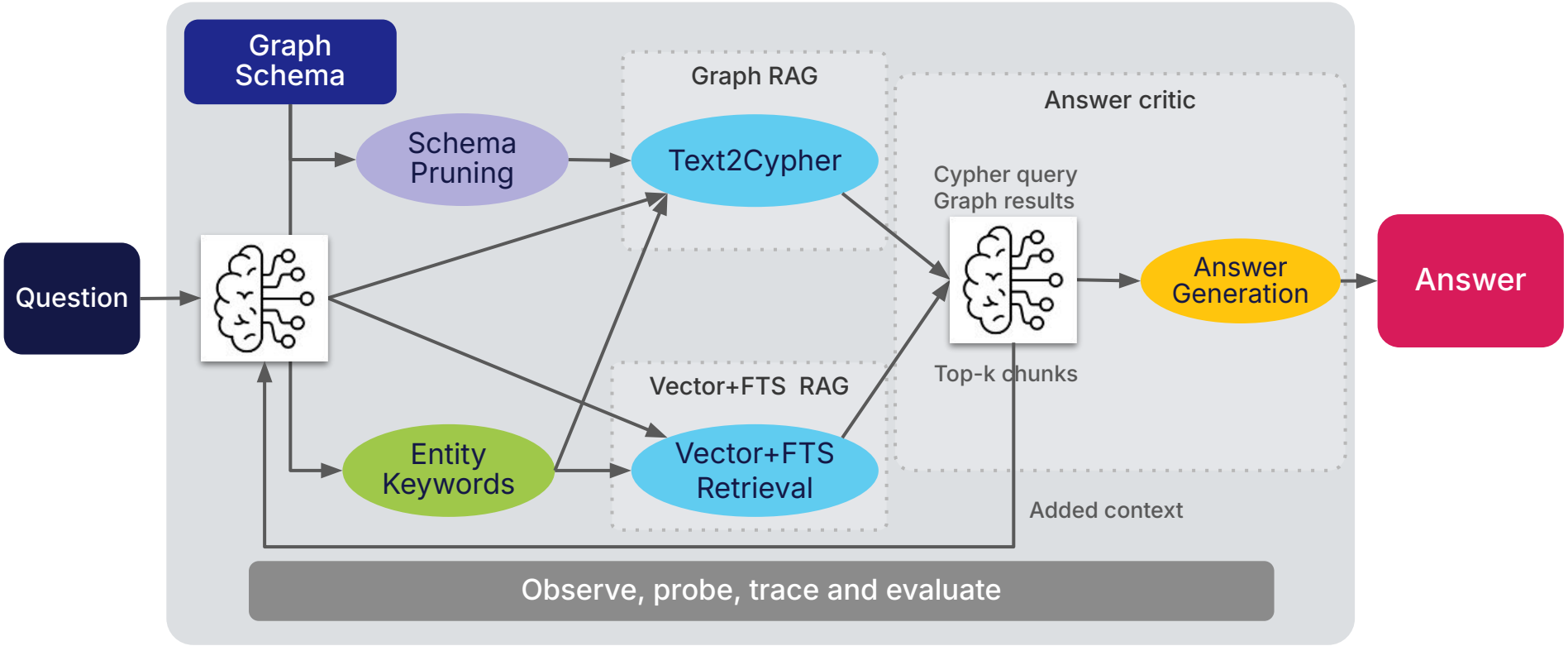
- Start with a workflow, increase the level of autonomy gradually
- Add orchestration, adaptive logic and more elaborate routing logic down the line
- Evals, evals, more evals (In DSPy, they help build self-improving systems)

Combining retrievers for RAG (1)



Combining retrievers for RAG (2)

kùzu



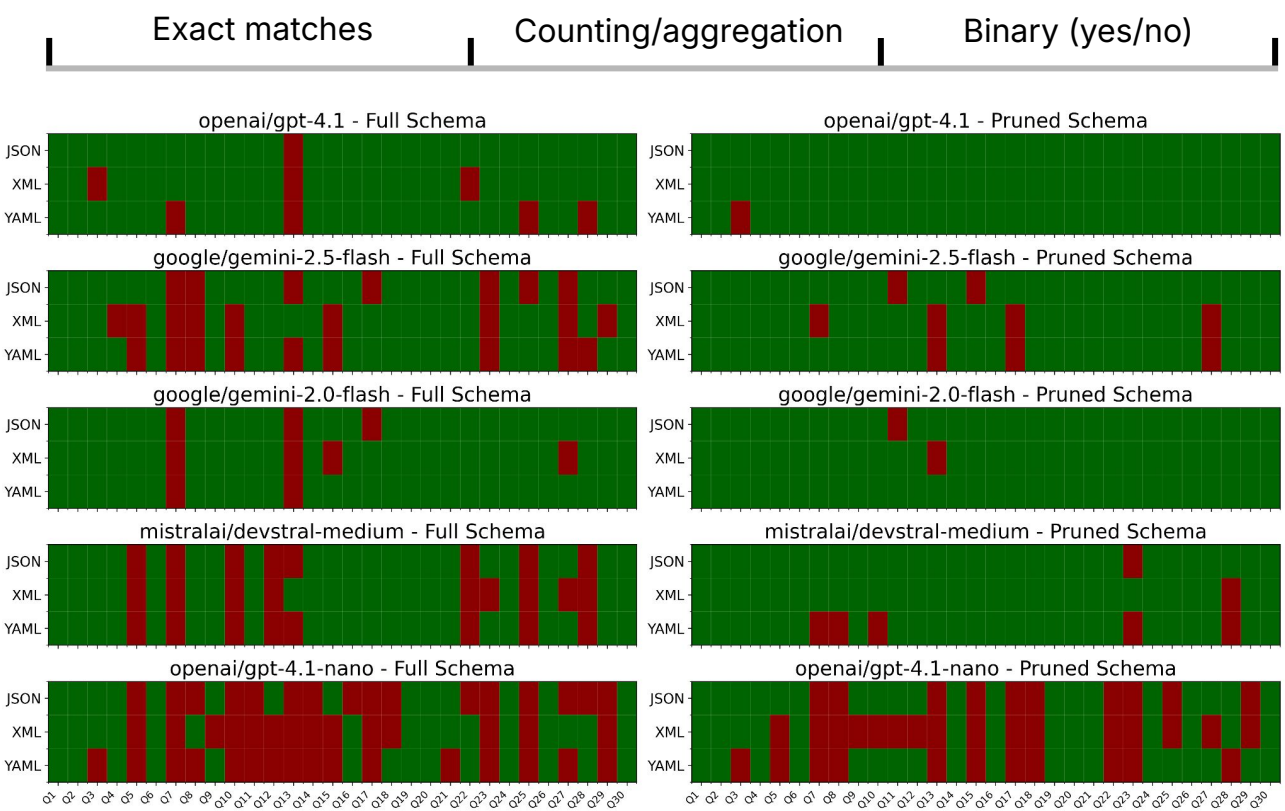
How graphs can help with retrieval: Example

kùzu

Yes/No + Follow-up	Is the patient ID 45 allergic to the substance 'shellfish'? If so, what city and state do they live in, and what is the full name of the practitioner who treated them?	*Yes, patient ID 45 is allergic to shellfish. They live in East Longmeadow, Massachusetts, and were treated by a practitioner named Cletus Paucek.*
Counting + Exact match	How many substances cause allergies in the category 'food'?	*There are 13 distinct substances that cause allergies in the 'food' category.*
Counting + Follow-up	What are the unique categories of substances patients are allergic to?	*The unique categories of substances patients are allergic to are food, other, medication, and environment.*
Exact match + Yes/No	Did the practitioner 'Arla Fritsch' treat more than one patient?	*Yes, the practitioner 'Arla Fritsch' treated more than one patient. The results show that they treated 4 patients.*
Exact Match	What are the full names of the patients treated by the practitioner named Josef Klein?	*The full names of the patients treated by the practitioner named Josef Klein are: Lili Abbie Brekke, Marinda Lindsay Veum, Gary Everette Abshire, Gabrielle Claudie Medhurst, Shaunte Adeline Batz, Jovan Rolanda Ondricka, Jerrell Weston Graham, and Della Legros.*
Counting	How many patients were born in between the years 1990 and 2000?	*There are 184 patients born between 1990 and 2000.*
Counting	How many patients have been immunized after January 1, 2022?	*The number of patients who have been immunized after January 1, 2022 is 65.*
Counting	Which practitioner treated the most patients? Return their full name and how many patients they treated.	*The practitioner who treated the most patients is Ted Reilly, with a total of 19 patients.*
Counting	How many patients are immunized for influenza?	*Based on the context, 204 patients are immunized for influenza.*
Counting	How many patients with the last name 'Rosenbaum' received multiple immunizations?	*There is one patient named Rodrigo Ernie Rosenbaum who received multiple immunizations. On June 12, 2015, he received an injectable, preservative-free seasonal Influenza vaccine and a meningococcal MCV4P vaccine.*

How good are LLMs at Text2Cypher in practice?

kùzu



github.com/kuzudb/text2cypher

blog.kuzudb.com: Improving Text2Cypher for Graph RAG via schema pruning

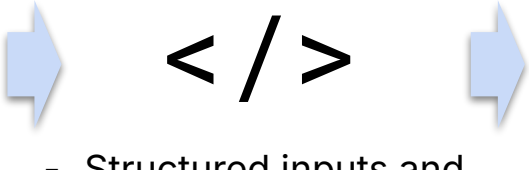
Summary

BAML

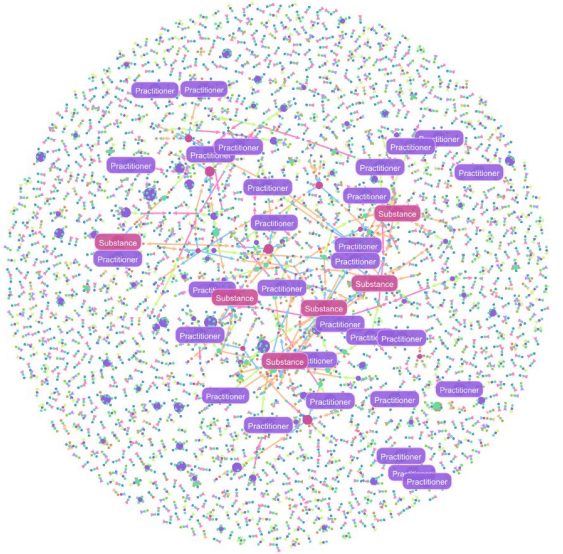
DSPy

Outlines
by dottxt.ai

langextract



- Structured inputs and outputs (code)
- Types (schema)



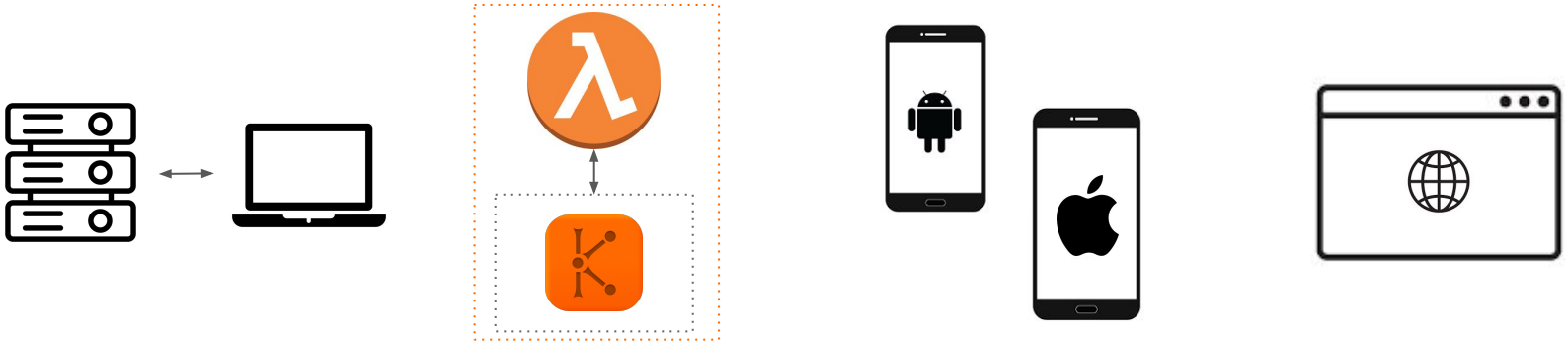
We showed how to go from messy, unstructured data → “insights”

- We want to be able to answer *difficult questions* and identify *patterns*
- We want to be able to *interpret* and *explain* the results

Why Kuzu?

kùzu

- Embedded graph database that supports the property graph model
  - Runs everywhere, without servers, or with a server on top
  - Brings a fast, portable graph query engine to wherever your data sits
- Seamlessly switch between ephemeral and persistent modes
- Don’t sweat over performance & scalability



Thank you! Learn more at kuzudb.com

kùzu

Go above and beyond your traditional RAG architectures with Kuzu.

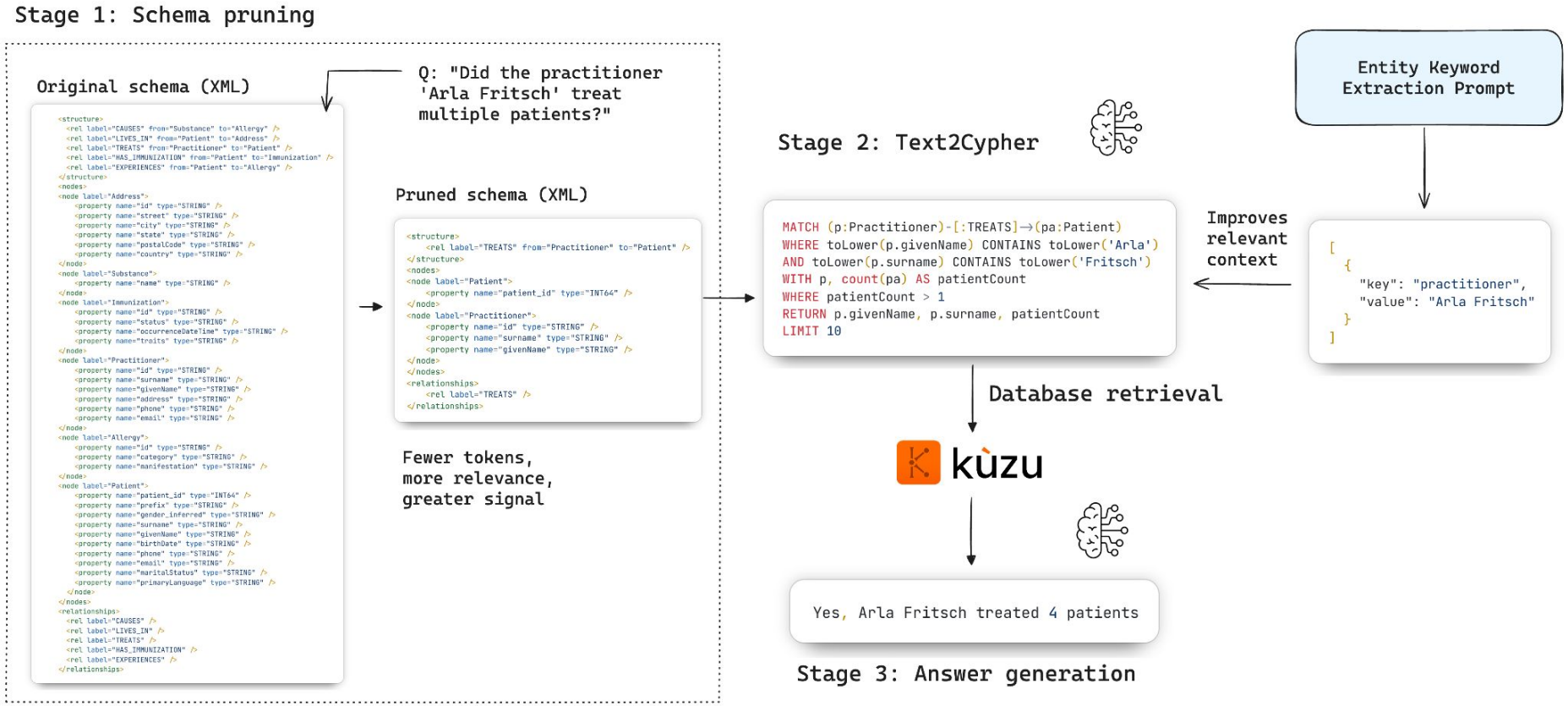
Give Kuzu a ⭐ on GitHub – and try it for yourself!

github.com/kuzudb/kuzu

@kuzudb

Read more about our work at blog.kuzudb.com

Appendix 1: Schema pruning to improve Text2Cypher



Cypher vs. SQL: Differences in verbosity

kùzu

"Are there any comments created by person that reply to a post also created by the same person?"

Cypher

```
MATCH (c:Comment)-[r1:commentHasCreator]→(p:Person)
      ←[r2:postHasCreator]-(post:Post)
      ←[r3:replyOfPost]-(c)
RETURN * LIMIT 10;
```

SQL

```
SELECT
  c.*,
  r1.comment_id AS r1_comment_id, r1.person_id AS r1_person_id,
  p.*,
  r2.post_id    AS r2_post_id,    r2.person_id AS r2_person_id,
  post.*,
  r3.comment_id AS r3_comment_id, r3.post_id   AS r3_post_id
FROM Comment AS c
JOIN commentHasCreator AS r1
  ON r1.comment_id = c.id
JOIN Person AS p
  ON p.id = r1.person_id
JOIN postHasCreator AS r2
  ON r2.person_id = p.id
JOIN Post AS post
  ON post.id = r2.post_id
JOIN replyOfPost AS r3
  ON r3.post_id = post.id
 AND r3.comment_id = c.id
LIMIT 10;
```