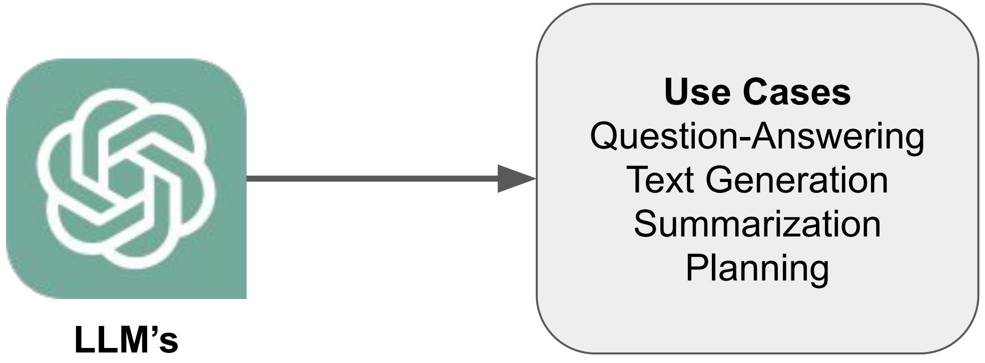
LlamaIndex

A Central Interface between LLM’s + your external data

https://github.com/jerryjliu/llama_index

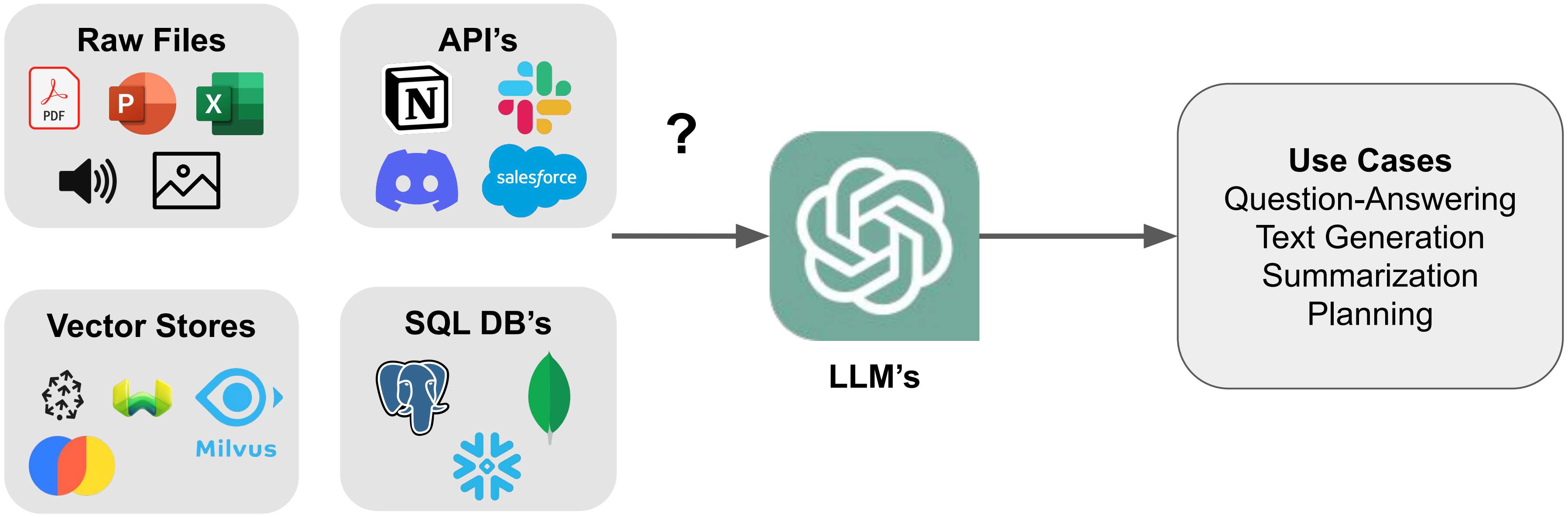
Context

- LLMs are a phenomenal piece of technology for knowledge generation and reasoning. They are pre-trained on large amounts of **publicly available data**.



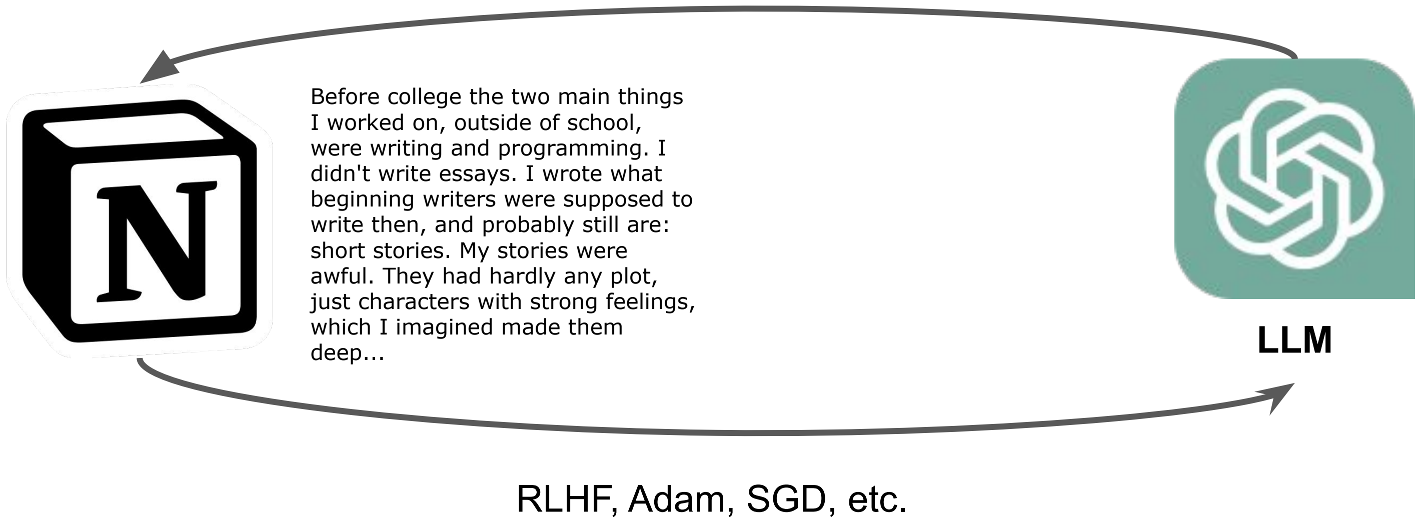
Context

- How do we best augment LLMs with our own **private data**?



Paradigms for inserting knowledge

Fine-tuning - baking knowledge into the weights of the network


Before college the two main things I worked on, outside of school, were writing and programming. I didn't write essays. I wrote what beginning writers were supposed to write then, and probably still are: short stories. My stories were awful. They had hardly any plot, just characters with strong feelings, which I imagined made them deep...

Paradigms for inserting knowledge

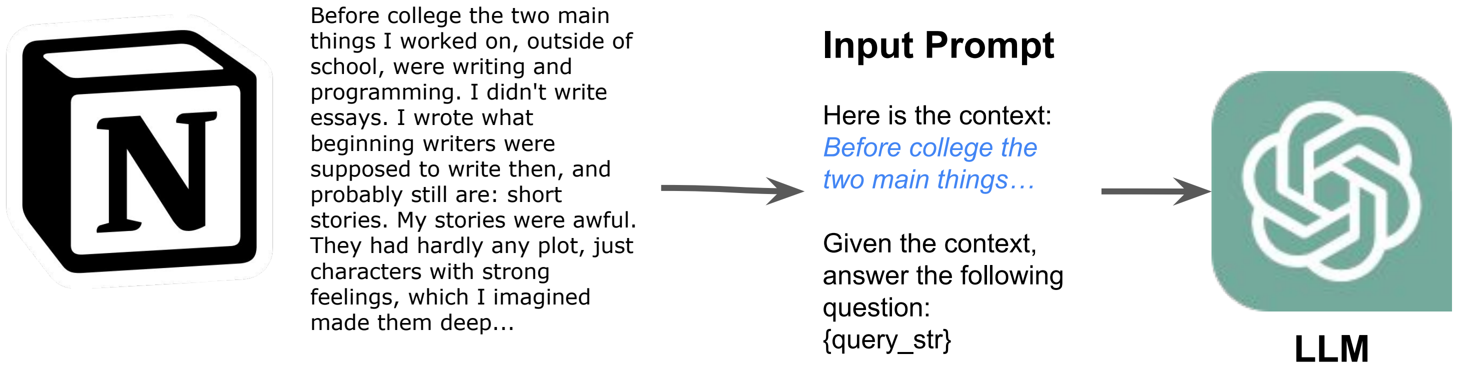
Fine-tuning - baking knowledge into the weights of the network

Downsides:

- Data preparation effort
- Lack of transparency
- Doesn’t work well
- High upfront cost

Paradigms for inserting knowledge

In-context learning - putting context into the prompt



Before college the two main things I worked on, outside of school, were writing and programming. I didn't write essays. I wrote what beginning writers were supposed to write then, and probably still are: short stories. My stories were awful. They had hardly any plot, just characters with strong feelings, which I imagined made them deep...

Here is the context:
Before college the two main things…

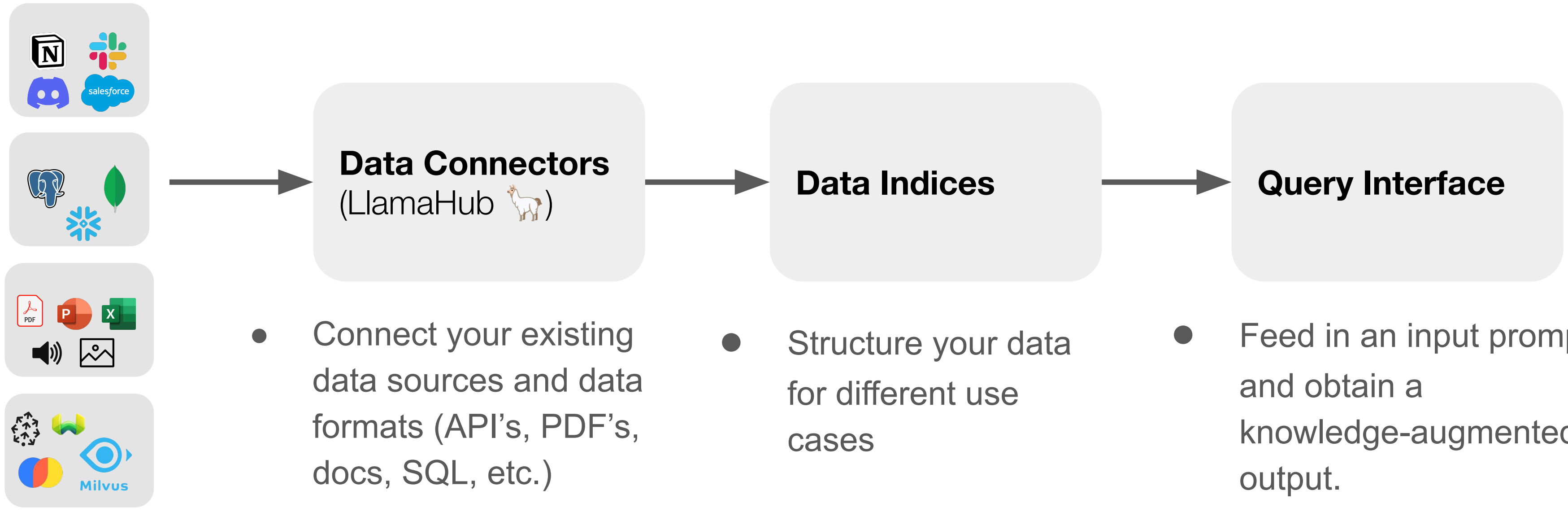
Given the context, answer the following question:
{query_str}

Key challenges of in-context learning

- How to retrieve the right context for the prompt?
- How to deal with long context?
- How to deal with source data that is potentially very large? (GB’s, TB’s)
- How to tradeoff between:
    - Performance
    - Latency
    - Cost

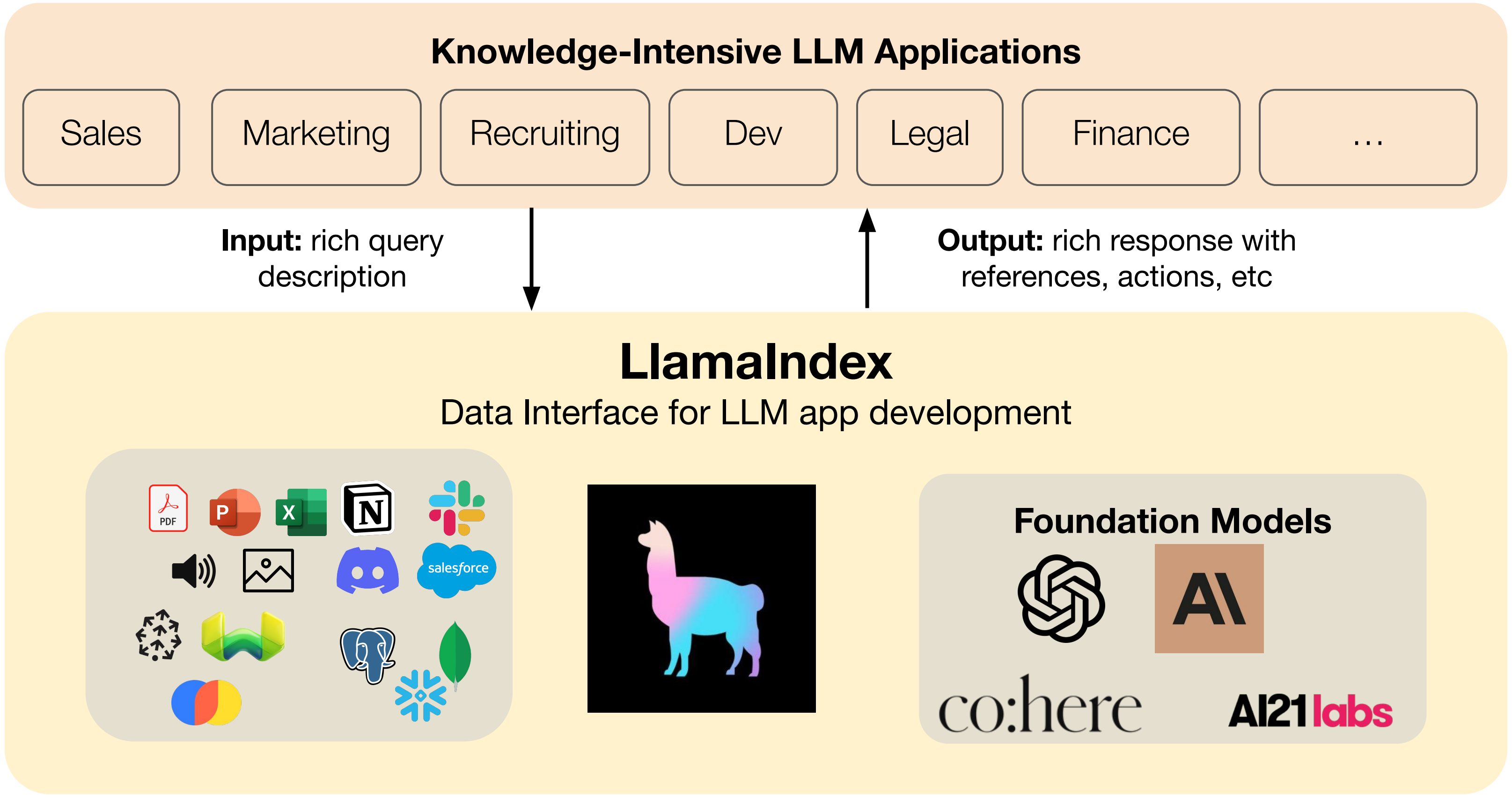
LlamaIndex: A interface between your data and LLMs

- Our goal is to make this interface *fast, cheap, efficient, and performant*



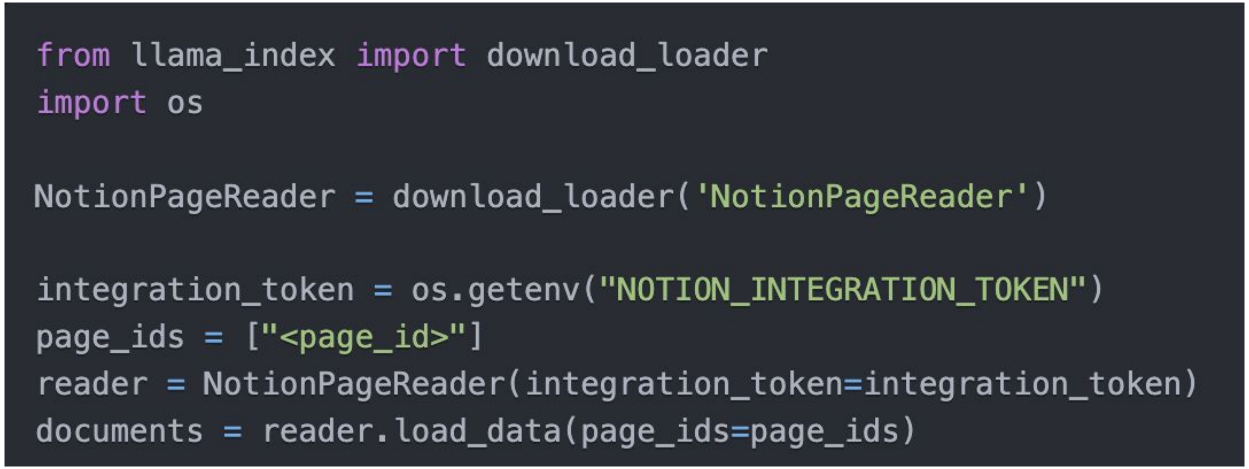
- Connect your existing data sources and data formats (API's, PDF's, docs, SQL, etc.)
- Structure your data for different use cases
- Feed in an input prompt and obtain a knowledge-augmented output.

LlamaIndex


Knowledge-Intensive LLM Applications
Sales
Marketing
Recruiting
Dev
Legal
Finance
…
Input: rich query description
Output: rich response with references, actions, etc
LlamaIndex
Data Interface for LLM app development
Foundation Models

Data Connectors: powered by LlamaHub 🦙

- Easily ingest any kind of data, from anywhere
  - into *unified* document containers
- Powered by community-driven hub
  - rapidly growing (61 loaders and counting!)
- Growing support for multimodal documents (e.g. with inline images)

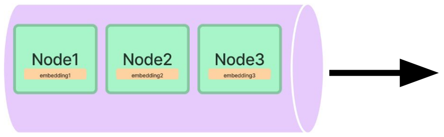


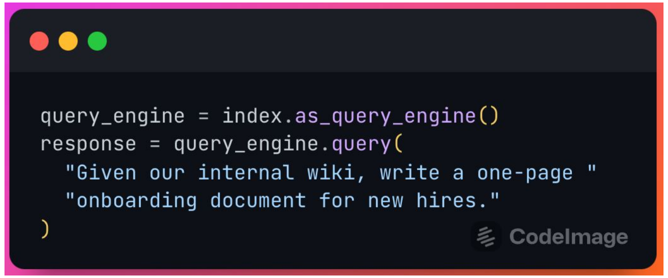
```python
from llama_index import download_loader
import os

NotionPageReader = download_loader('NotionPageReader')

integration_token = os.getenv("NOTION_INTEGRATION_TOKEN")
page_ids = ["<page_id>"]
reader = NotionPageReader(integration_token=integration_token)
documents = reader.load_data(page_ids=page_ids)
```

<10 lines of code to ingest from Notion

Data Indices + Query Interface

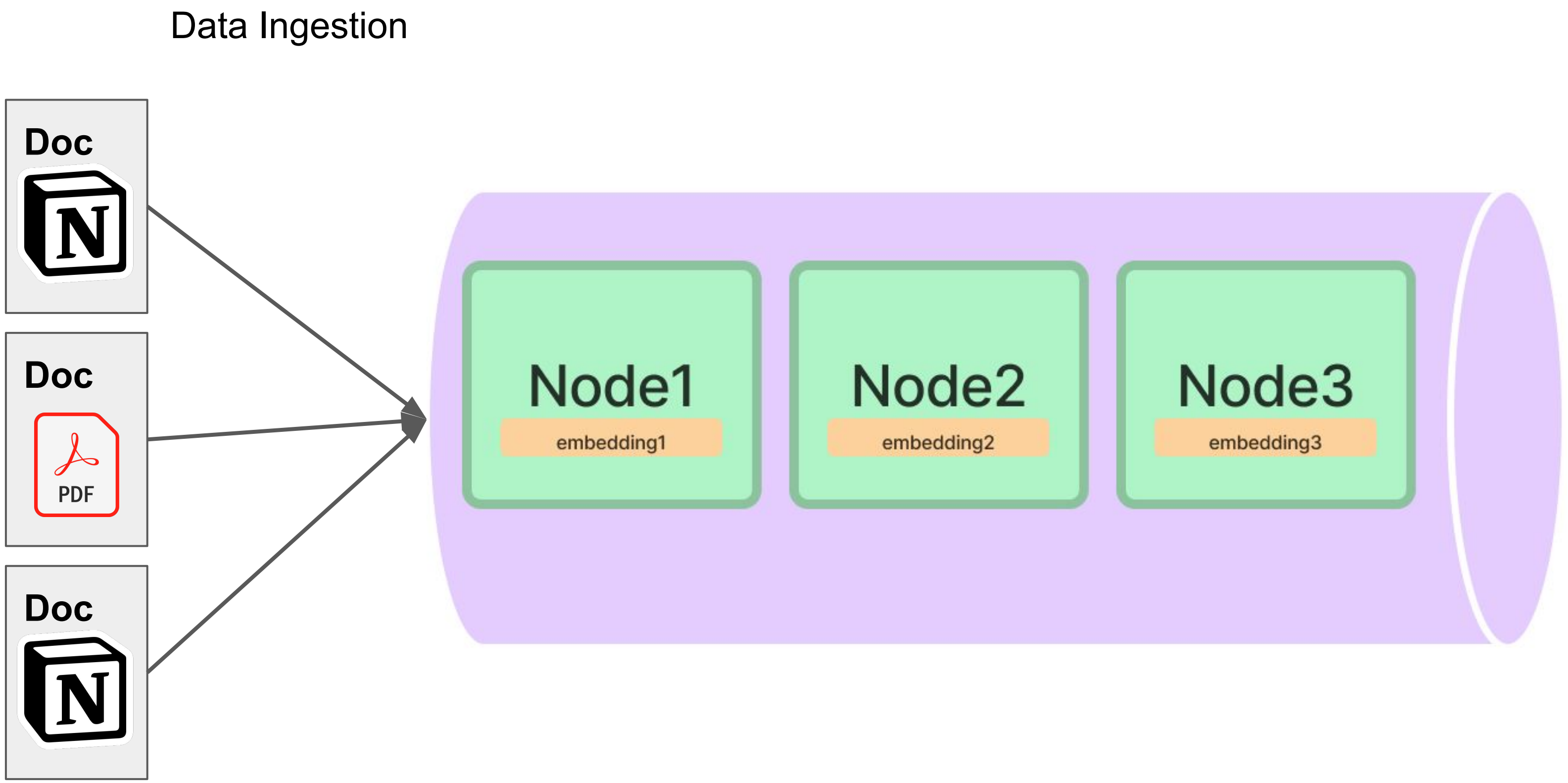
- Our **data indices** help to abstract away common boilerplate/pain points for in-context learning.
  - Storing context in an easy-to-access format for prompt insertion.
  - Dealing with prompt limitations (e.g. 4096 tokens for Davinci) when context is too big.
  - Dealing with text splitting.
- A **query interface** on top of these indices simultaneously retrieves/synthesizes information.
- Let’s walk through a few examples!



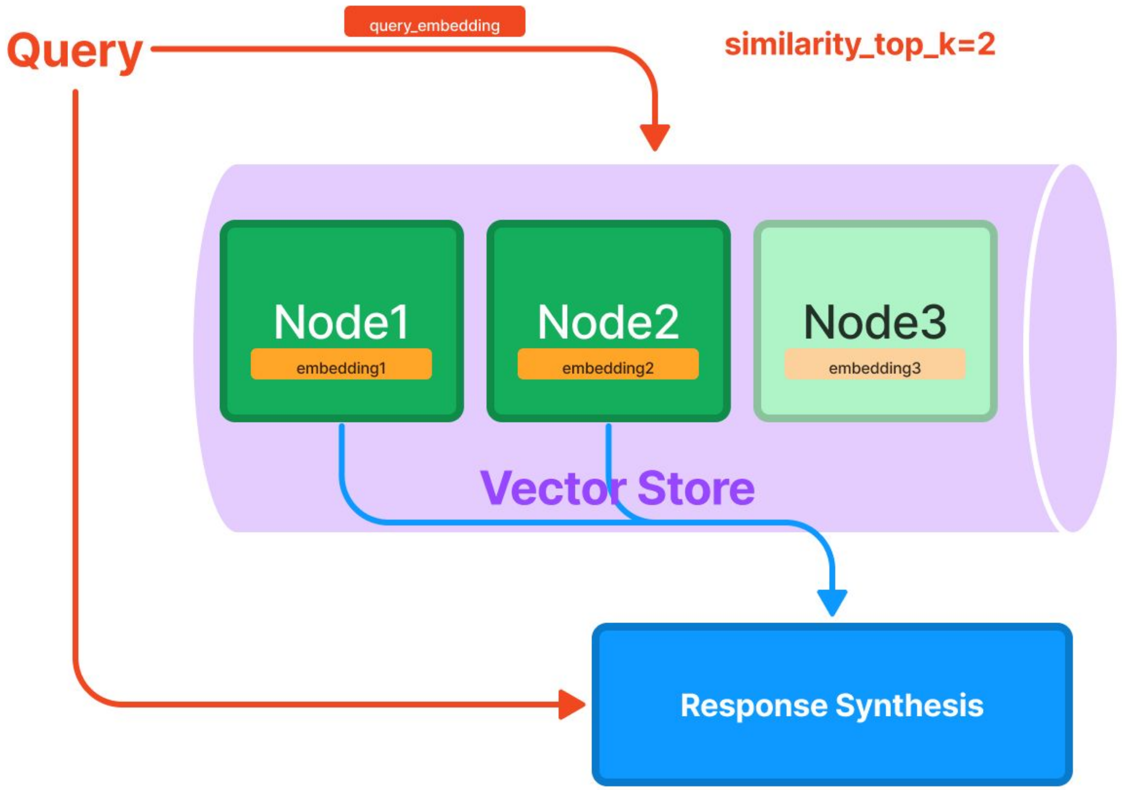


```
query_engine = index.as_query_engine()
response = query_engine.query(
  "Given our internal wiki, write a one-page "
  "onboarding document for new hires."
)
```

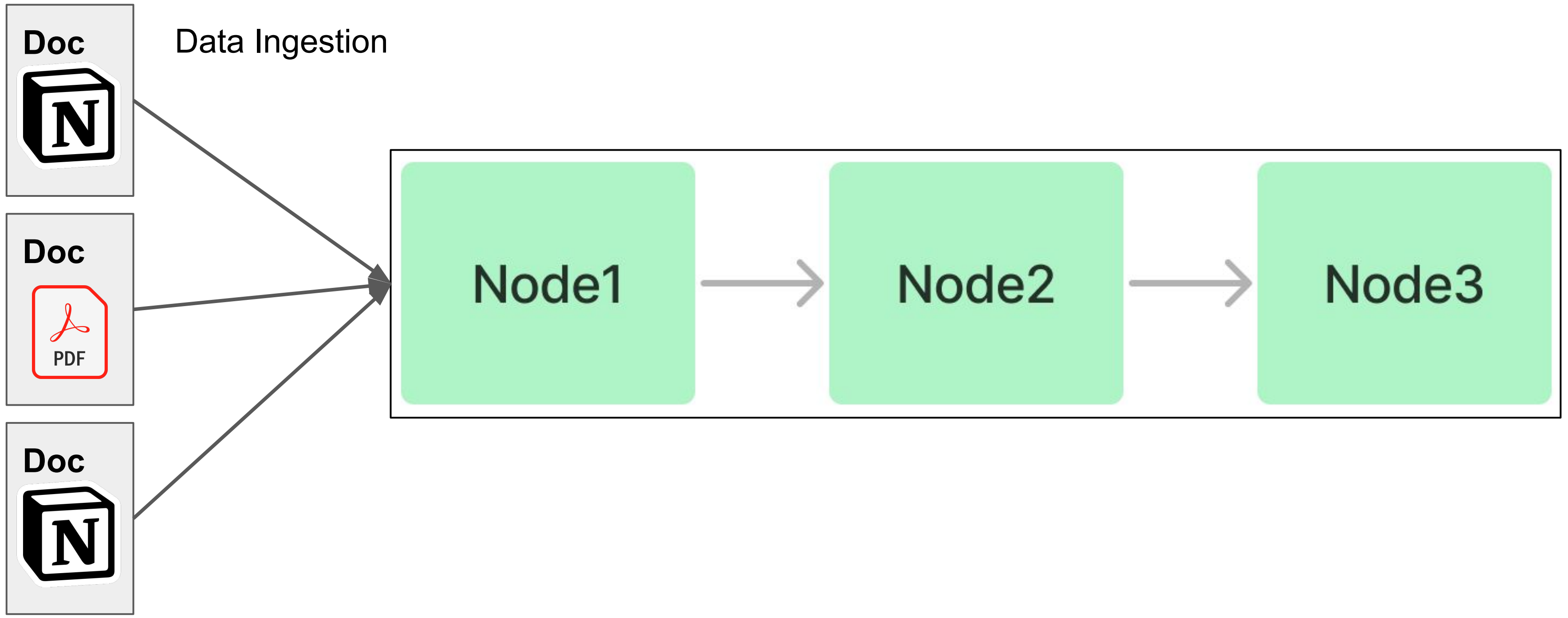
Vector Store Index



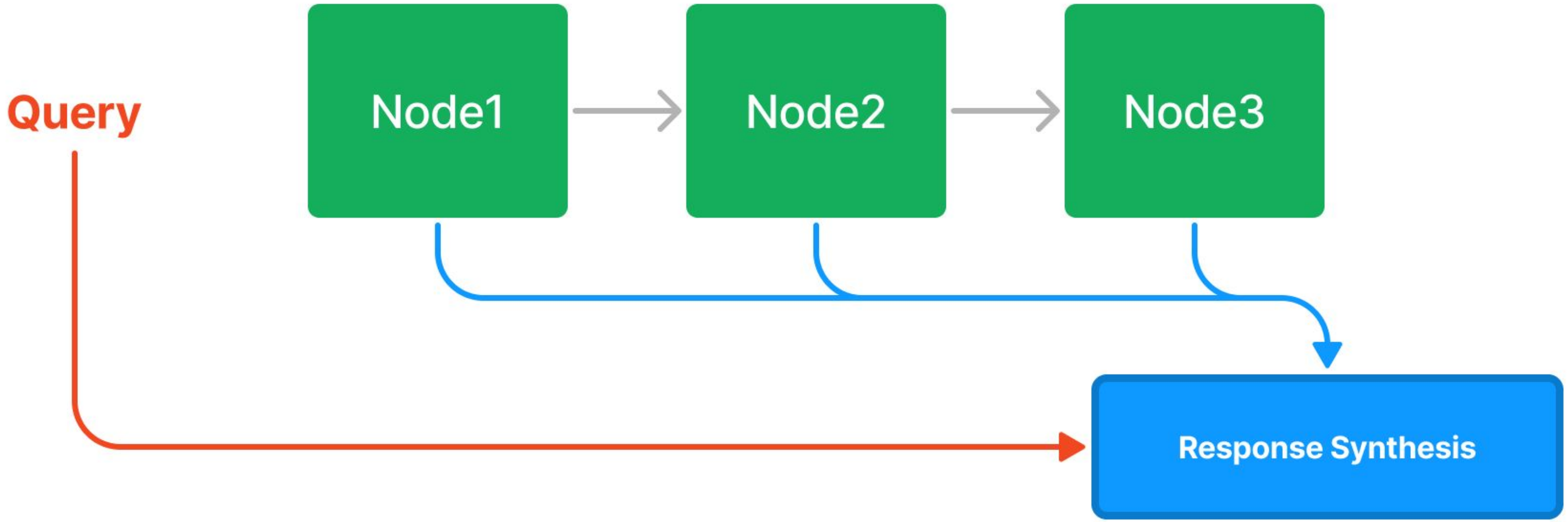
Vector Store Index



List Index

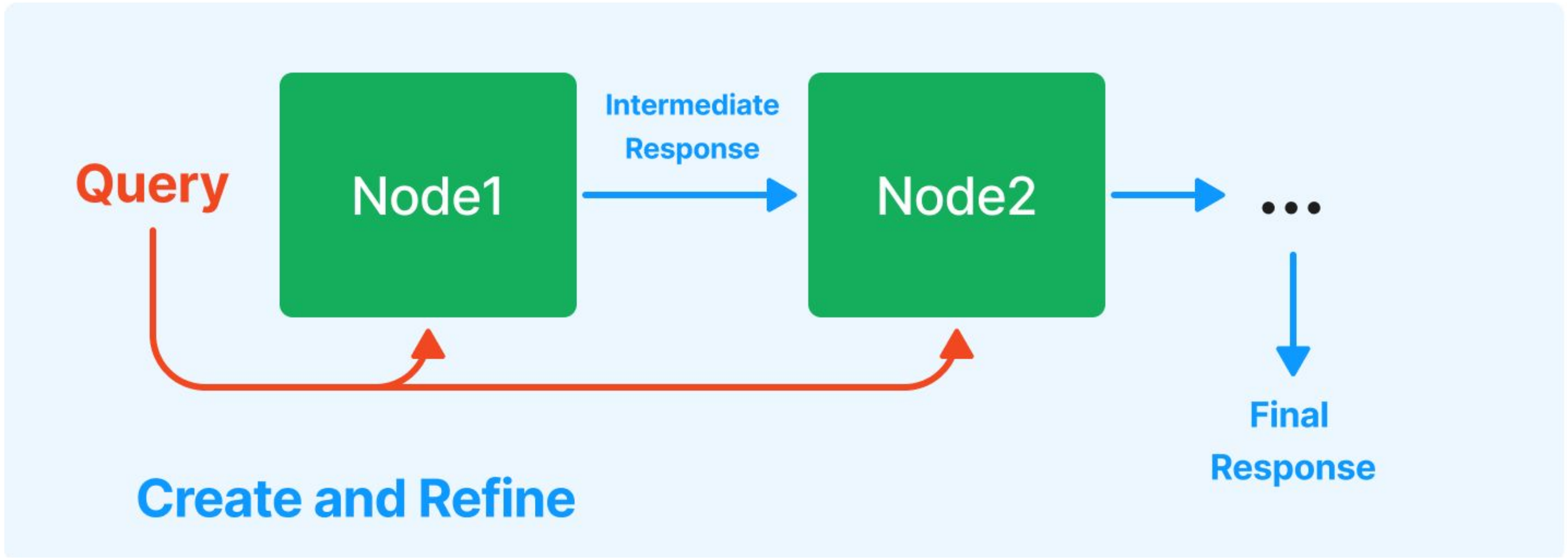


List Index



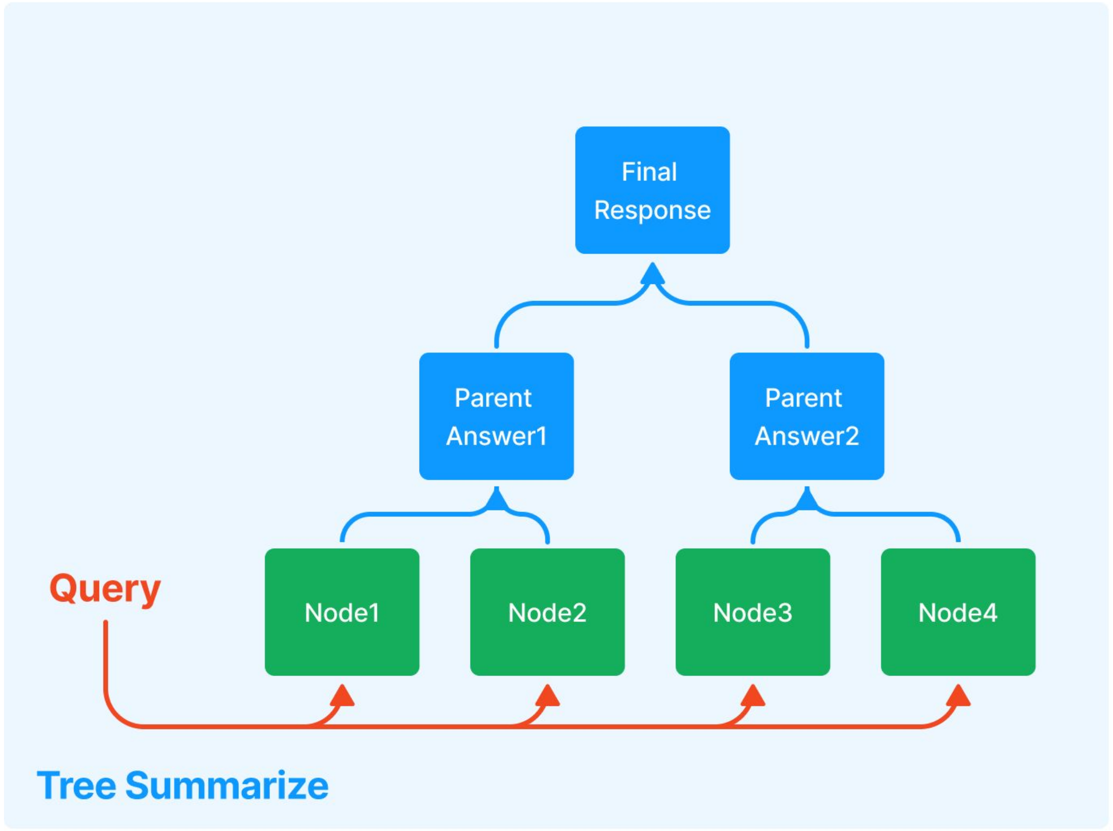
Response Synthesis

Create and refine

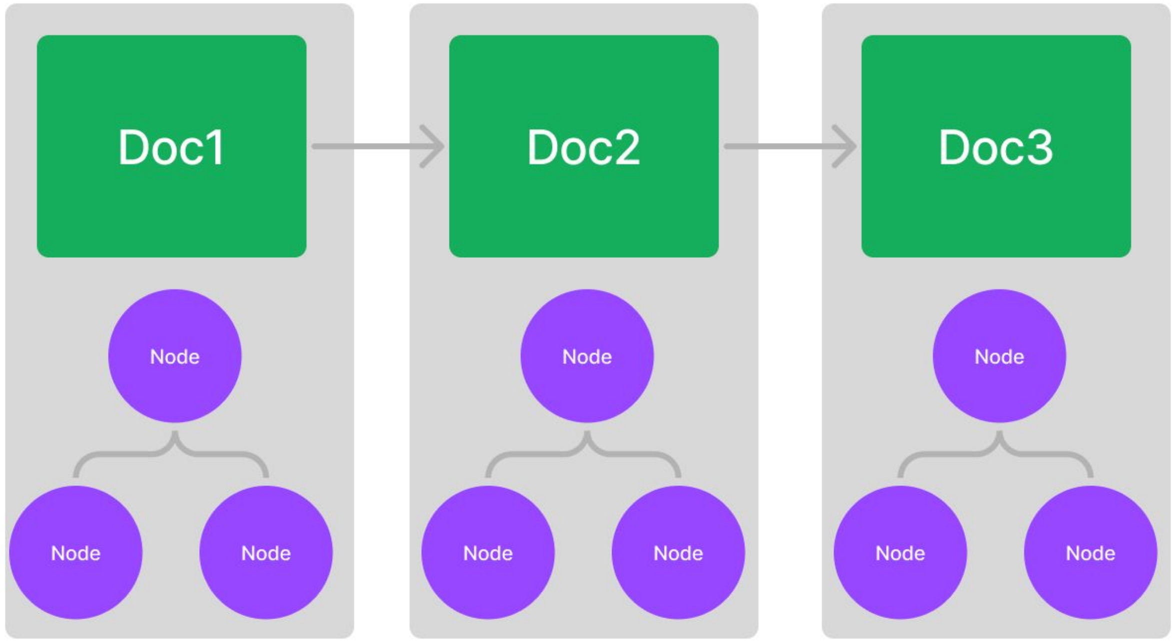


Response Synthesis

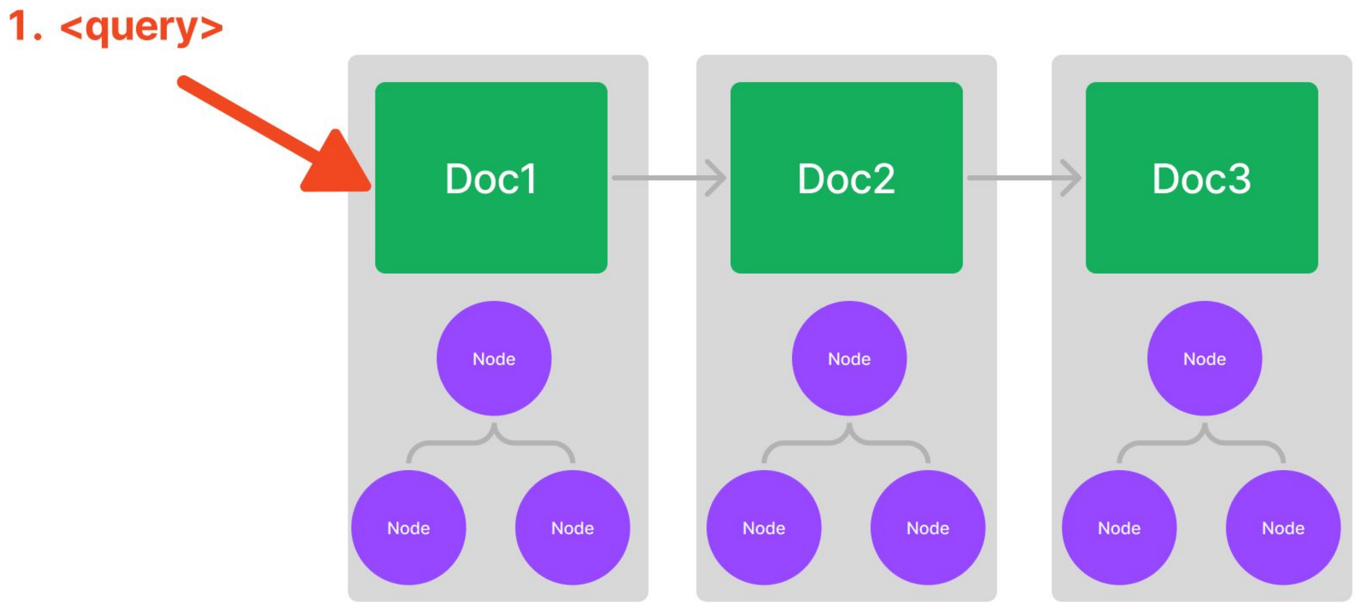
Tree Summarize



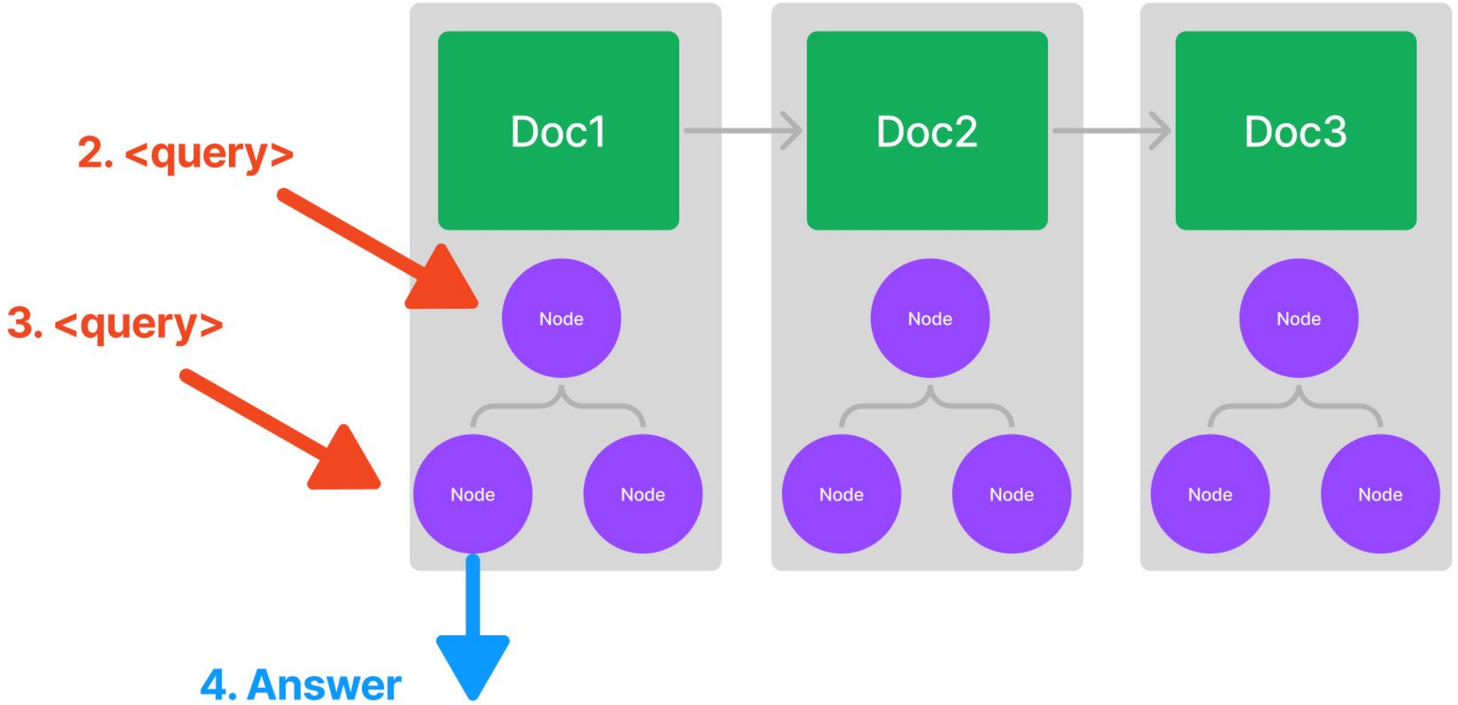
[More advanced] Composing a graph



[More advanced] Composing a graph



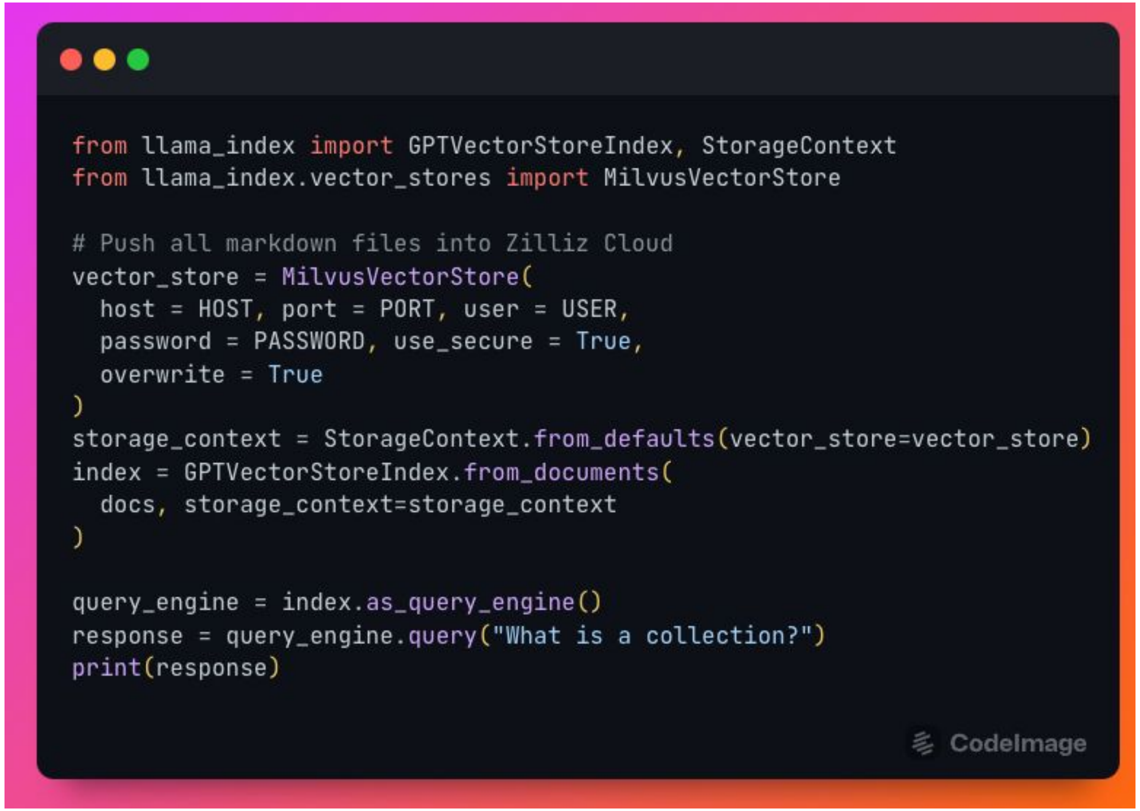
[More advanced] Composing a graph



Milvus Integration

Use Milvus as the backend vector store for your texts and embeddings!

https://zilliz.com/doc/integrate_with_llama



```python
from llama_index import GPTVectorStoreIndex, StorageContext
from llama_index.vector_stores import MilvusVectorStore

# Push all markdown files into Zilliz Cloud
vector_store = MilvusVectorStore(
  host = HOST, port = PORT, user = USER,
  password = PASSWORD, use_secure = True,
  overwrite = True
)
storage_context = StorageContext.from_defaults(vector_store=vector_store)
index = GPTVectorStoreIndex.from_documents(
  docs, storage_context=storage_context
)

query_engine = index.as_query_engine()
response = query_engine.query("What is a collection?")
print(response)
```

Demo Walkthrough

Let’s play around with LlamaHub + index + query!

Easily ingest data

https://colab.research.google.com/drive/12cdBWMpOfCxpiAS1zSqZRY66o84qMiTo?usp=sharing

Use Case: Semantic Search

```python
from llama_index import GPTVectorStoreIndex, SimpleDirectoryReader
documents = SimpleDirectoryReader('data').load_data()
index = GPTVectorStoreIndex.from_documents(documents)

query_engine = index.as_query_engine(response_mode="tree_summarize")
response = query_engine.query(
     "What did the author do growing up?"
)
```

Answer

The author grew up writing short stories, programming on an IBM 1401, and working on microcomputers. He wrote simple games, a program to predict how high his model rockets would fly, and a word processor. He studied philosophy in college, but switched to AI. He reverse-engineered SHRDLU for his undergraduate thesis and wrote a book about Lisp hacking. He visited the Carnegie Institute and realized he could make art that would last.

Use Case: Summarization

```
from llama_index import GPTListIndex, SimpleDirectoryReader
documents = SimpleDirectoryReader('data').load_data()
index = GPTListIndex.from_documents(documents)

query_engine = index.as_query_engine(response_mode="tree_summarize")

response = query_engine.query("Could you give a summary of this article in
newline separated bullet points?")
```

Answer

- The author began writing and programming before college, and studied philosophy in college before switching to AI.
- He realized that AI, as practiced at the time, was a hoax and decided to focus on Lisp hacking instead.
- He wrote a book about Lisp hacking and graduated with a PhD in computer science.
- ….

Use Case: Text-to-SQL (Structured Data)

```python
from llama_index import GPTSQLStructStoreIndex, SQLDatabase

sql_database = SQLDatabase(engine, include_tables=["city_stats"])
# NOTE: the table_name specified here is the table that you
# want to extract into from unstructured documents.
index = GPTSQLStructStoreIndex.from_documents(
    wiki_docs,
    sql_database=sql_database,
    table_name="city_stats",
  )


# set Logging to DEBUG for more detailed outputs
query_engine = index.as_query_engine(mode="default")
response = query_engine.query("Which city has the highest population?")
print(response)
```

SQL Guide
https://gpt-index.readthedocs.io/en/latest/guides/tutorials/sql_guide.html

Generated SQL

```sql
SELECT city_name, population FROM city_stats ORDER BY population DESC LIMIT 1
```

Use Case: Synthesis over Heterogeneous Data

```python
from llama_index import GPTVectorStoreIndex, GPTListIndex

from llama_index.indices.composability import ComposableGraph

index1 = GPTVectorStoreIndex.from_documents(notion_docs)

index2 = GPTVectorStoreIndex.from_documents(slack_docs)

graph = ComposableGraph.from_indices(GPTListIndex, [index1, index2], index_summaries=["summary1", "summary2"])

response = graph.as_query_engine().query("Give me a summary of these two articles")
```

In this example, we **compose** an index over other indexes (a list index over vector indexes)

The query will be routed to both simple vector indexes!

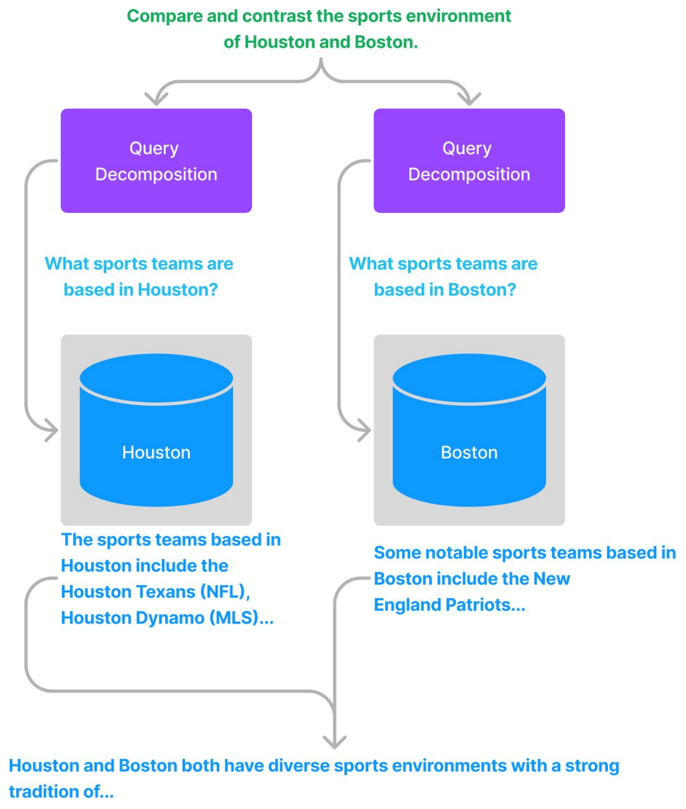
Use Case: Compare/Contrast Queries

A special case of synthesis over heterogeneous data.

Here, a query transform can help!

Notebook:

https://github.com/jerryjliu/llama_index/blob/main/examples/composable_indices/city_analysis/City_Analysis-Decompose.ipynb


Compare and contrast the sports environment of Houston and Boston.
Query Decomposition
Query Decomposition
What sports teams are based in Houston?
What sports teams are based in Boston?
Houston
Boston
The sports teams based in Houston include the Houston Texans (NFL), Houston Dynamo (MLS)...
Some notable sports teams based in Boston include the New England Patriots...
Houston and Boston both have diverse sports environments with a strong tradition of...

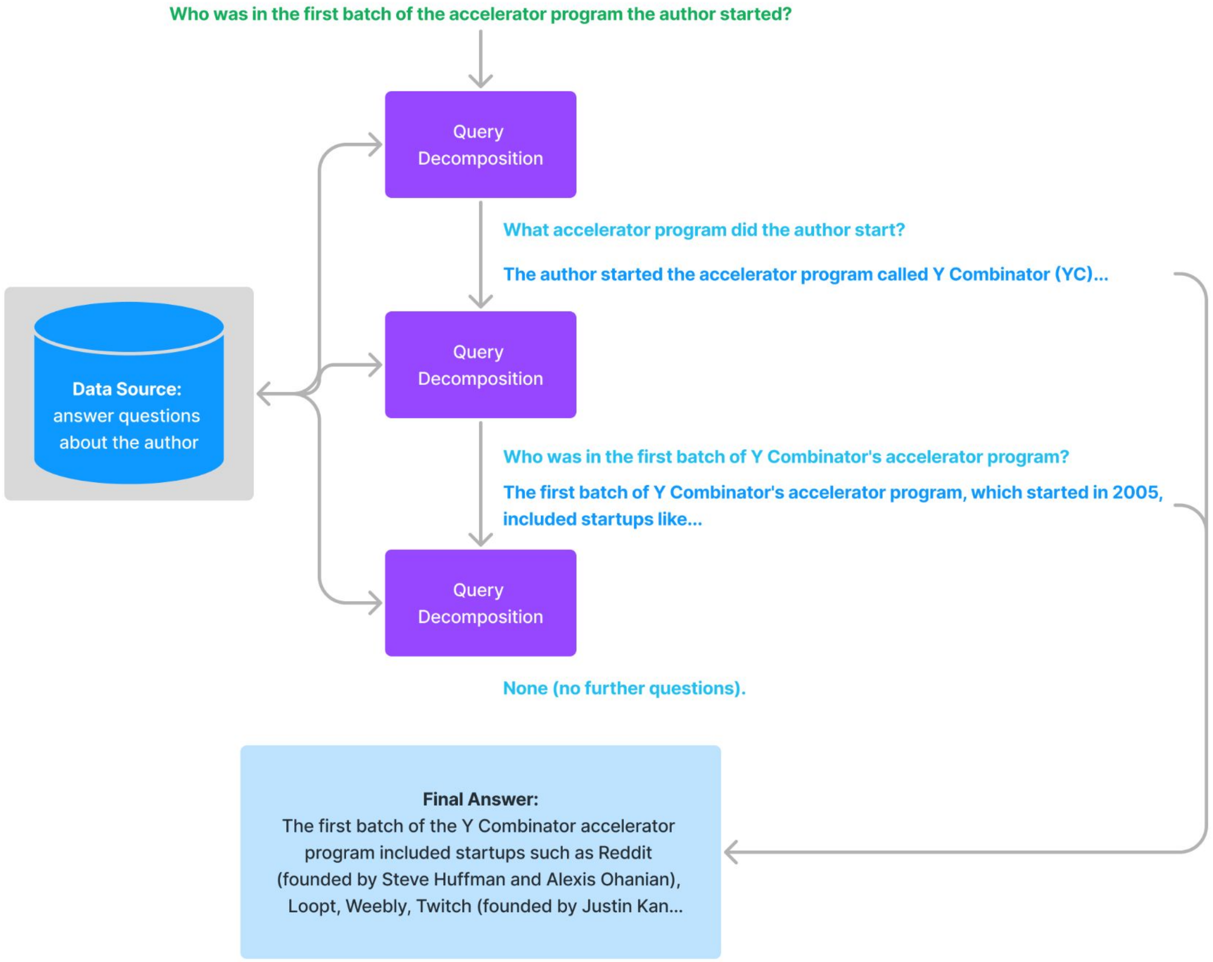
Use Case: Multi-Step Queries

Break a complex query into multiple simpler ones!

Chain-of-thought prompting over an existing data source.

Notebook:

https://github.com/jerryjliu/llama_index/blob/main/examples/vector_indices/SimpleIndexDemo-multistep.ipynb

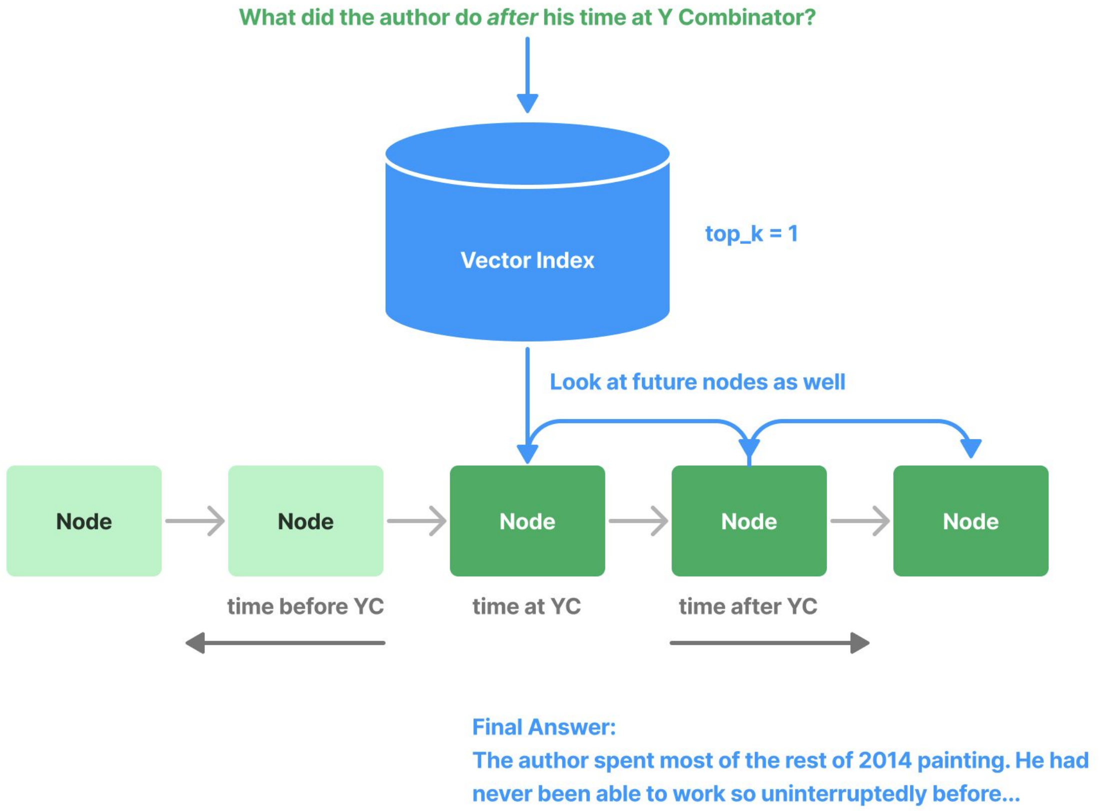

Who was in the first batch of the accelerator program the author started?
Query Decomposition
What accelerator program did the author start?
The author started the accelerator program called Y Combinator (YC)...
Query Decomposition
Who was in the first batch of Y Combinator's accelerator program?
The first batch of Y Combinator's accelerator program, which started in 2005, included startups like...
Query Decomposition
None (no further questions).
Data Source: answer questions about the author
Final Answer: The first batch of the Y Combinator accelerator program included startups such as Reddit (founded by Steve Huffman and Alexis Ohanian), Loopt, Weebly, Twitch (founded by Justin Kan...

Use Case: Exploiting Temporal Relationships

Given a question, what if we would like to retrieve additional context in the past or the future?

Example question: "What did the author do *after* his time at Y Combinator?"

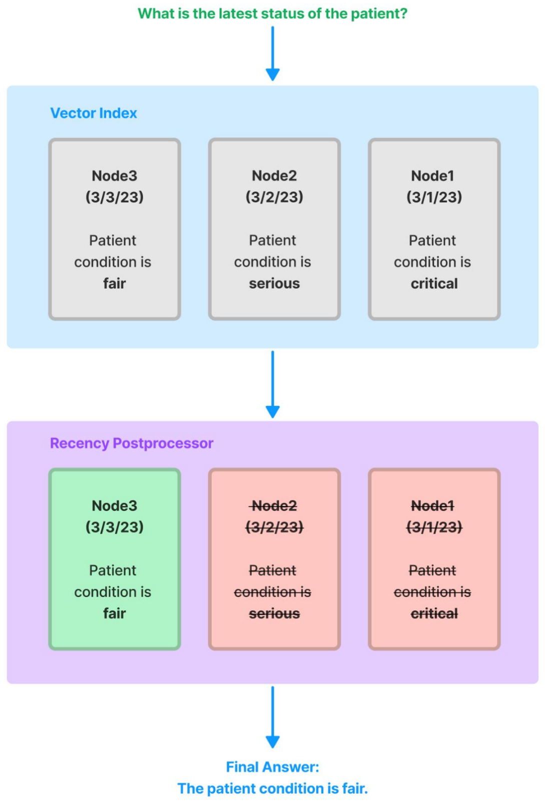
Requires looking at context in the future!


What did the author do *after* his time at Y Combinator?
Vector Index
top_k = 1
Look at future nodes as well
Node
Node
Node
Node
Node
time before YC
time at YC
time after YC
Final Answer:
The author spent most of the rest of 2014 painting. He had never been able to work so uninterruptedly before...

Use Case: Recency Filtering / Outdated nodes

Imagine you have three timestamped versions of the same data.

If you ask a question over this data, you want to make sure it's over the latest document.


What is the latest status of the patient?
Vector Index
Node3 (3/3/23) Patient condition is fair
Node2 (3/2/23) Patient condition is serious
Node1 (3/1/23) Patient condition is critical
Recency Postprocessor
Node3 (3/3/23) Patient condition is fair
~~Node2 (3/2/23) Patient condition is serious~~
~~Node1 (3/1/23) Patient condition is critical~~
Final Answer:
The patient condition is fair.

Integration into Downstream Apps

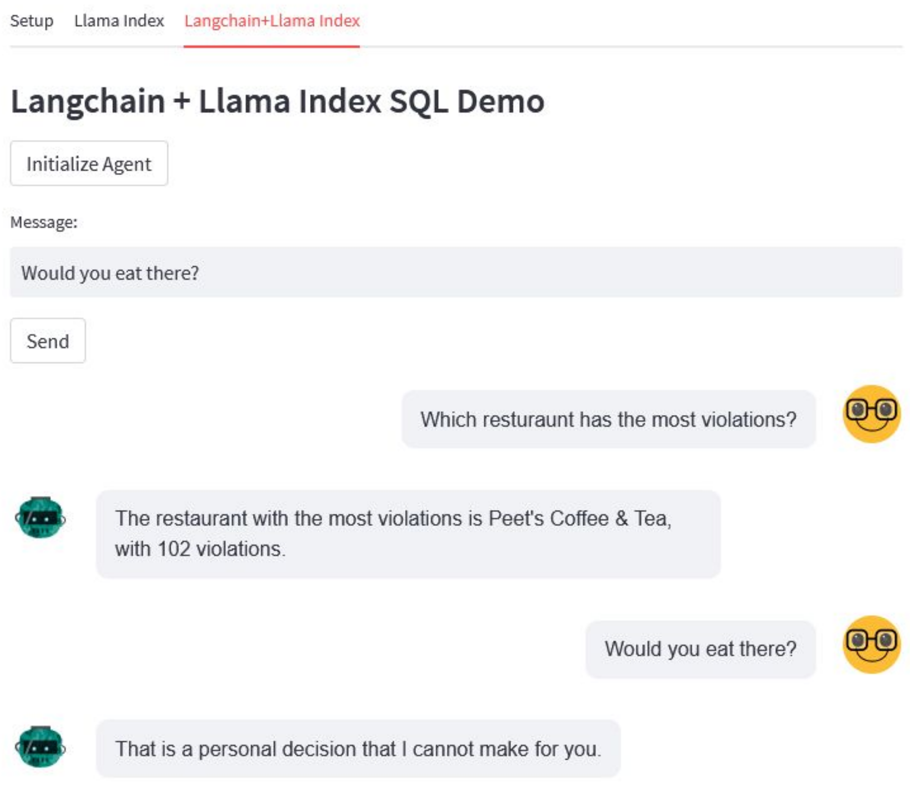
- Build a chatbot with LlamaIndex + Langchain

https://gpt-index.readthedocs.io/en/latest/guides/tutorials/building_a_chatbot.html

Integration into Downstream Apps

- Build a Streamlit app!

https://huggingface.co/spaces/llamaindex/llama_index_sql_sandbox



Langchain + Llama Index SQL Demo

Would you eat there?

Which resturaunt has the most violations?

The restaurant with the most violations is Peet's Coffee & Tea, with 102 violations.

Would you eat there?

That is a personal decision that I cannot make for you.

More Demo Walkthroughs!

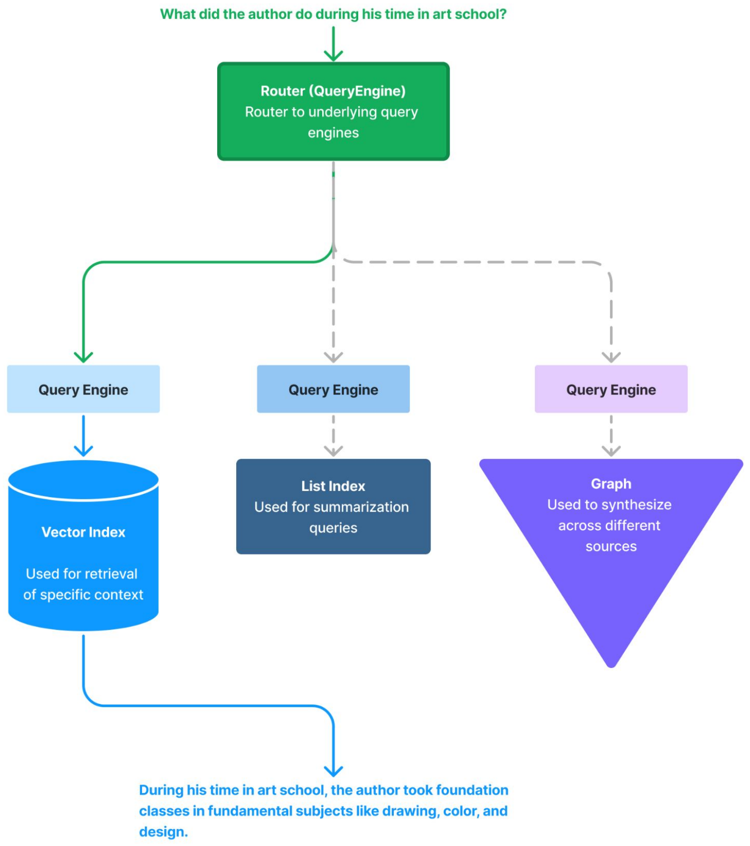
Building a custom retriever

https://github.com/jerryjliu/llama_index/blob/main/examples/query/CustomRetrievers.ipynb

More Demo Walkthroughs! [Advanced]

Building a unified query interface

https://colab.research.google.com/drive/1KH8XtRiO5spa8CT7UrXN54IWdZk3DDxl?usp=sharing


What did the author do during his time in art school?
Router (QueryEngine) Router to underlying query engines
Query Engine
Query Engine
Query Engine
Vector Index
Used for retrieval of specific context
List Index Used for summarization queries
Graph Used to synthesize across different sources
During his time in art school, the author took foundation classes in fundamental subjects like drawing, color, and design.