= Text to Cypher Retriever
:type: lesson
:order: 5
:branch: main

Vector and full text retrievers are great for finding relevant data based on semantic similarity or keyword matching.

To answer more specific questions, you may need to perform more complex queries to find data relating to specific nodes, relationships, or properties.

For example, you want to find:

* The age of an actor.
* Who acted in a movie.
* Movie recommendations based on rating.

Text to Cypher retrievers allow you to convert natural language queries into Cypher queries that can be executed against the graph.

```
[User Query]
"What year was the movie Babe released?"

[Generated Cypher Query]
MATCH (m:Movie)
WHERE m.title = 'Babe'
RETURN m.released

[Cypher Result]
1995

[LLM Response]
"The movie Babe was released in 1995."
```

In this lesson, you will create a text to Cypher retriever that will generate context based on the user's query and the graph schema.

== Retriever

Open the `genai_fundamentals/text2cypher_rag.py` file and review the code:

[source,python]
.text2cypher_rag.py
----
include::{repository-raw}/{branch}/genai-fundamentals/text2cypher_rag.py[tag=**]

The programs includes all the code to connect to Neo4j, create the `embedder`, `llm`, and `GraphRAG` pipeline.

You will need to create the `TextToCypherRetriever` retriever that will

generate Cypher queries and return results.

The `TextToCypherRetriever` requires an LLM to generate the Cypher queries:

[source,python]
.text2cypher_rag.py
----
include::{repository-raw}/{branch}/genai-fundamentals/solutions/text2cypher_rag.py[tag=t2c_llm]

[NOTE]
.Temperature
The `temperature` is set to `0`.
When generating Cypher queries, you want the output to be deterministic and precise.

Create the `TextToCypherRetriever` retriever:

[source,python]
----
include::{repository-raw}/{branch}/genai-fundamentals/solutions/text2cypher_rag.py[tag=import_text2cypher]

include::{repository-raw}/{branch}/genai-fundamentals/solutions/text2cypher_rag.py[tag=retriever]

The retriever will automatically read the graph schema from the database when it is used.

[%collapsible]
.Click to view the complete code
====
[source,python]
----
include::{repository-raw}/{branch}/genai-fundamentals/solutions/text2cypher_rag.py[tag=**]

====

Run the program, the retriever results and the generated Cypher will be printed when you submit a query.

Experiment with some queries and review the results, for example:

* Who directed the movie Superman?
* How many movies are in the Sci-Fi genre?
* What are examples of Action movies?

[IMPORTANT]
====

There is no guarantee that the generated Cypher query will return results, or be syntactically correct.

In production use cases, you should ensure exception are handled and the generated Cypher queries are validated before execution.
====

== Example Queries

To improve the accuracy of the generated Cypher queries, you can provide examples of queries and an appropriate Cypher query.

If you wanted to find data related to movie ratings, you will could provide an example query that demonstrates how to retrieve ratings from the graph:

```
    USER INPUT: 'Get user ratings for a movie?'

    QUERY: MATCH ()-[r:RATED]->(m:Movie)
           WHERE m.title = 'Movie Title'
           RETURN r.rating
```

Create a list of the example queries:

```
[source,python]
----
include::{repository-raw}/{branch}/genai-fundamentals/solutions/text2cypher_rag_examples.py[tag=examples]
----
```

Update the `TextToCypherRetriever` to include the `examples`:

```
[source,python]
----
include::{repository-raw}/{branch}/genai-fundamentals/solutions/text2cypher_rag_examples.py[tag=retriever]
----
```

```
[%collapsible]
.Click to view the complete code
====
[source,python]
----
include::{repository-raw}/{branch}/genai-fundamentals/solutions/text2cypher_rag_examples.py[tag=**]
----
====
```

Run the program and experiment with queries that relate to movie ratings, such as:

* What is the highest rating for Goodfellas?

* What is the average user rating for the movie Toy Story?
* What user gives the lowest ratings?

You can specify multiple examples to help the LLM understand the context and generate more accurate Cypher queries.

== Schema

The `TextToCypherRetriever` will automatically read the whole graph schema from the database.

You can provide a custom schema to the retriever if you want to limit the nodes, relationships and properties that are used to generate Cypher queries.
Limiting the scope of the schema can help improve the accuracy of the generated Cypher queries, particularly if the graph contains a lot of nodes and relationships.

Create a textual representation of the graph schema:

```
[source,python]
----
include::{repository-raw}/{branch}/genai-fundamentals/solutions/text2cypher_rag_schema.py[tag=schema]
----
```

Add the `schema` to the `TextToCypherRetriever`:

```
[source,python]
----
include::{repository-raw}/{branch}/genai-fundamentals/solutions/text2cypher_rag_schema.py[tag=retriever]
----
```

```
[%collapsible]
.Click to view the complete code
====
[source,python]
----
include::{repository-raw}/{branch}/genai-fundamentals/solutions/text2cypher_rag_schema.py[tag=**]
----
====
```

== Experiment

Experiment by submitting complex queries, observing the generated Cypher queries, and adapting the examples and schema to improve the accuracy of the generated queries.

== Continue

When you are ready, continue to the next lesson.

read::Continue[]

[.summary]
== Lesson Summary

In this lesson, you learned how to create a text to Cypher retriever that generates Cypher queries based on natural language input.

In the next lesson, you will explore some of the GenAI frameworks available and their integration with Neo4j.