EXAFLUENCE
Data Driven Influence

Putting Snowflake Copilot to the test: An EXFGenAI Initiative

Kalaivani KG
May 2024

The rapid development in GenAI space is contributing to significant advancements in the domain of Natural Language to SQL translation. At Exafluence, we have built a GenAI powered chatbot that queries our proprietary Investment data model, a complex database schema with over 80 tables hosted in Snowflake.

When Snowflake announced Copilot, the prospect of leveraging a Snowflake native GenAI powered tool to enhance data querying capabilities was compelling. We had previously invested significant effort into scaling our solution for enterprise-level databases, eventually settling on a combination of RAG architecture and organized table views for scalability. Our data schema had unique challenges:

- **Database schema size:** 80+ tables
- **Data overlap:** The overlap in data residing in tables and columns. For example, multiple market value fields stored in base currency, local currency etc
- **Column naming:** Deep domain knowledge embedded in column names, such as alpha_rating in the instrument table, complicates comparisons through SQL queries.
- **Specialized columns:** Enterprise-specific fields, like the hyphenated FoF path in portfolio structures, indicate instrument organization within funds.
- **Categorical Fields:** Fields with multiple categories that users can query directly without specifying the column name, like "Active portfolios."

Given these complexities and our target audience of non-technical business users, I was keen to see how Copilot would handle our data model. I conducted several queries of varying complexity to evaluate Copilot, focusing on practical application rather than its technical details.

Testing Overview

1. Assessing Schema Comprehension

1.a/ Query: How many tables are in the selected schema?

Copilot's Response: Identified 10 key tables, focusing on the most relevant ones, although the complete schema contains 83 tables and 31 views. I guess it showed the 10 most queried tables out of the entire schema.

1.b/ Query: Can you describe this schema?

Copilot's Response: Provided an accurate description focusing on investment data, like instruments and portfolios, but only based on the top 10 tables.

2. Straight forward queries - Portfolio Queries

2.a/ Query: Which table contains portfolio information?

Copilot's Response: Copilot accurately identified the relevant table for portfolio data.

2.b/ Query: How many portfolios are there?

Copilot's Response: Generated the correct SQL query but needed clarification on ambiguous column names.

2.c/ Query: Show me the top 10 portfolios by market value.

Copilot's Response: It required column name clarification due to ambiguities.

2.d/ Query: Show me the top 10 holdings within GHI Fund.

Copilot's Response: Initially selected an incorrect view; provided the correct SQL after I specified the view.

3. Complex querying requiring domain understanding - Bond Ratings

3.a/ Query: Show me instruments that are rated lower than BB-.

Copilot's Response: Copilot chose the right table for the query. It initially used an incorrect comparison approach using the alpha numeric column containing ratings such as AA+, AA-, Aa1, etc. Upon correction, it recommended creating a mapping table for better accuracy.

4. Querying documentation on Snowflake

4.a/ Query: What does the Streamlit apps tab do?

Copilot's Response: Provided a good summary of Streamlit apps.

4.b/ Query: Can one provide access to these apps to business users?

Copilot's Response: Offered a relevant response along with a link to a Snowflake Tutorial.

4.c/ Query: How many concurrent users can use these Streamlit apps? Can it support load balancing?

Copilot's Response: Lacked specific information, suggesting it wasn't in the documentation.

5. Query optimization

5.a/ Query: (I provided a query). Is the following query optimized? Is it using the right indexes and any new indexes to be created?

Copilot's Response: Confirmed that no new indexes were needed, and existing indexes were optimally used.

At this point, I stopped my testing as I got a good initial feel for Copilot and here is the summary:

1. Snowflake Copilot is a good tool for data analysts and engineers, not for business users.

2. It can create simple straight forward queries where the column names and tables are all very intuitively understandable. Complex queries and schemas with non-intuitive naming conventions require users to refine inputs through chatbot interface.

3. It could mean a significant productivity increase for users who have a very good understanding of the data and the schema.

4. It is a useful tool for optimizing queries.

One potential improvement for Copilot could be the implementation of a one-time semantic capture of tables, views, and column names to enrich its contextual understanding. It would be advantageous for Snowflake to explore storing this semantic information, enabling users who are not deeply familiar with the data to still benefit from the tool. This enhancement would allow Copilot to autonomously generate complex queries without requiring user modifications or prompts. Perhaps, they are already considering it for their future updates.

If you would like to learn more, write to us at marketing@exafluence.com

Subscribe to our YouTube channel for more solution videos- https://bit.ly/3FNM0DG

YouTube

LinkedIn

For regular updates about Exafluence follow us on LinkedIn https://bit.ly/3FKCqlk