

Snowflake Intelligence



We propose to think creatively when establishing Snowflake Intelligence as your always-on enterprise helper. Semantic Views serve as a good tool to get reliable results from your AI Agents, but to maximize value, look to leverage outside information. Use custom tools to automate weekly reports sent via email to relevant employees, enable specific User Defined Functions to call external APIs for additional data, such as weather forecasts, or supplement your marketing analysis with an agent working directly on customer feedback.

Snowflake Inc. recently released Snowflake Intelligence (SI), an enterprise agent that helps employees answer complex business questions simultaneously. It communicates in natural language with the user and leverages the Snowflake Cortex AI Suite. Explicitly, the Cortex Analyst, a fully managed service that generates text-to-SQL responses, and Cortex Search, a “fuzzy” search over your unstructured Snowflake data. One can add Custom Tools such as UDFs. SI orchestrates these agents to retrieve, contextualize, and explain data. It can visualize results with charts and graphs. These operations

occur within Snowflake’s data governance space. As it is with any software, getting started with Snowflake Intelligence requires some setup. For example, SI cannot reach its full capability without company-specific inputs such as proprietary terms, company standards like the start of the fiscal year, derived metrics, or semantic connections between tables. Several online resources exist to guide you through the SI setup. Here, we focus on one of the unspoken aspects: the effort needed for post-hoc data explanation to make SI a valuable business intelligence tool and what most impacts its capabilities.

Problem Statement

Assume you want to start your Snowflake Intelligence Journey. You have your data loaded into a data warehouse, including satellites, links, and hubs (or dimensions and fact tables), covering your company's main processes. Now, the question arises: how do you turn data into insights using Snowflake Intelligence? What preparation yields the most benefit quickly and reliably? What learnings can be shared? Here, we propose a methodology to qualitatively identify where to focus your efforts.

The "Quick Win"

Use Case Description

We will use the Willibald data for our investigation here, a dataset that is often used to train building data vaults. It is open to public access under a Creative Commons Attribution 4.0 International License, and contains 13 tables that span the case of a fictional B2C plant reseller with an online shop. Tables for plants and seeds, customers, addresses (delivery and residence), delivery services, and centrally an orders table and order positions are contained within the dataset, alongside further supplementary information tables. Complexity is added by a second kind of sale, via roadshow events, which take place on single days in parallel to online activity. It is not essential to cover the content or trends within the dataset here. If you are interested, we recommend going straight to the source for detailed explanation of the peculiarities.

Methodology

The German dataset, i.e., its tables, is uploaded to Snowflake without any further processing other than translation of the table names into English for ease of reference here. Altogether, there are 13 tables with 89 columns representing minor but useful complexity. Setting up Snowflake Intelligence is done with at least one Cortex Analytics Agent and its Semantic View. Note that upon initial ingestion to a Semantic View, Snowflake offers to add AI generated descriptions to tables and columns, and these descriptions are prone to errors or misinterpretations.

To get a handle on the amount of data explanation work expanded, we define four discrete levels. We call these "semantic status". Starting from Level 0, which basically means the uploaded raw data without manual intervention, explanation, relating or description of tables or columns. This serves to establish a baseline to test increased capabilities of the SI Agent against. Level 0 employs one Cortex Analyst Agent only. The next logical step is to provide join relations, explaining date formats, and providing currency information. This is added to the semantic view that the Cortex Analyst Agent uses,

establishing Level 1. Level 2 means including descriptions for every column and table, adding synonyms, relevant derived metrics, and hints about data cleaning such as duplicates in the data. Supplementing information beyond that in Level 3 is meant to represent adding external information to your dataset. In this case, PDFs with supplemental data have been loaded from web databases that provide further information about the plants on offer such as sowing depth or when to plant the seeds. This means including these PDFs via a Cortex Search Agent. We will also provide information about how to orchestrate the different agents, i.e., prompt to look for planting information from Cortex Search and sales information from Cortex Analyst.

Each Level of semantic status is asked the same questions with increasing levels of complexity. Easy questions should be answerable by taking a look at a single table or requiring at most a single JOIN operation. Medium questions will need to check multiple tables and correlate data. Hard questions benefit from having access to information external to the system.

Results

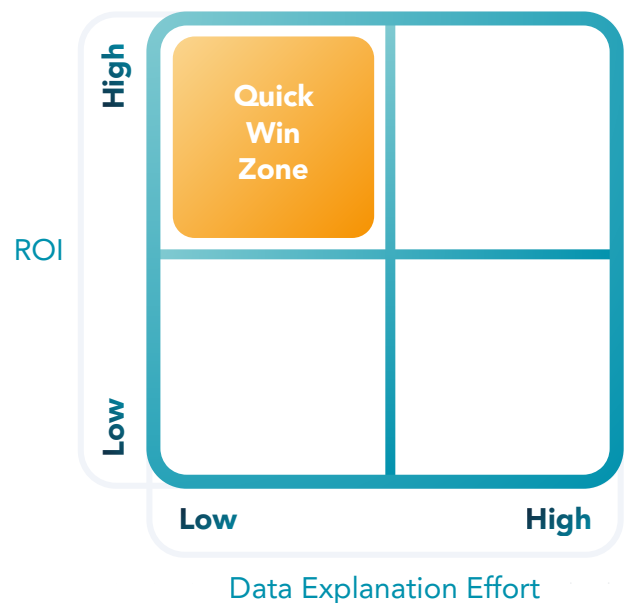
Let's turn to the use case established above to get an estimation of the numbers. It contained 13 tables with roughly 90 columns. Assuming that raw ingestion at Level 0 is already done (it would not change in further levels and can thus be disregarded), it will take about 30 minutes to add all connections between tables to the semantic view and provide guidance on currency and date formats on all relevant columns. Of course, for more interconnected data warehouses with table numbers in the hundreds, this will increase significantly. Nevertheless, a robust estimation is that a fact table with its dimensions, or hub-link-satellite connections, will take about 15 minutes. Extrapolating this for a more complex data warehouse, this will be in the order of magnitude of a day.

Reviewing, adjusting, or adding descriptions for columns and tables was a significantly longer task. We averaged roughly 20 minutes per table and its columns. Naturally, for a table with customer information like name and address, this will be less time-consuming than links and hash keys or strongly abbreviated column names. Highly normalized data warehouses or data vaults with many links and multiple satellites per hub will take longer to explain and connect than a simple star schema.

The major trends can be summarized as follows: With Level 0 the accuracy of answers was hit or miss as the agent had trouble interpreting whether, for example, "sales" referred to the "orders" table or the roadshow (both would be correct). SI could not perform joins and had to rely on querying tables at a time, correlating results post-hoc. Generally, reasoning through easy questions took longer than 60s per question. Some medium questions could be reasoned through ("What product is sold most often to which destination?") but broke down when trying to combine results from the individual steps. Harder questions could not be answered, or answers were factually wrong or incomplete.

Level 1 and 2 proved to be a lot more reliable in answer accuracy with Level 2 being better at handling synonyms that were not the column names themselves. Response times differed, of course, between questions and complexity of the required reasoning, but Level 2 produced responses roughly 15% faster overall. Within the small sample size used, this is well within statistical uncertainty. In general, there was no noticeable jump in quality between Levels 1 and 2. Most of this is attributed to the auto-generated descriptions we used for Levels 0 and 1 being generally accurate.

Level 3, which had access to the Cortex Search Agent, naturally was able to incorporate information other than the tables provided. This is most relevant when considering that SI will try to infer which columns to use when it cannot find an exact match, leading to incorrect answers. This was most obvious for questions asking for "external" information, e.g. "How many paprika plants can I grow in a 10x5 m field if I want to space them optimally for maximum use of the area, and what planting grid should I use?"



Quick Wins: How to tell when you're done

As was clearly demonstrated with an explanation of data connections provided, the SI agent was able to improve its capabilities significantly. Specifically, adding the relationships between the tables and company-specific language, or derived metrics that the agent could otherwise not know about, are quickly established and provide the biggest benefit to your organization. Beyond that, it becomes a case-by-case situation with diminishing returns. If your tables are not easily interpretable by an AI Agent, definitively provide broad strokes explanations and key definitions about table and column contents. Otherwise, the automatic interpretations provided upon ingestion can be seen as reasonably accurate. Give them a quick review, but you will be good to go. Beyond what we call Level 1 semantic effort, we demonstrated useful improvements when focusing on unclear terms.

The real benefit lies in adding information to your existing dataset. Imagine you own an ice cream shop and want to predict the amount of ice cream you should store in your limited freezer space. You can imagine that being able to supplement your historical data with a weather forecast will do a lot more for you than an analysis of ingredients that you saved in your data warehouse. After all, we prefer our cold treats in warm, dry weather. In our case, a quick search and ingestion of plant-specific information provided depth to the interpretation that manual data explanation alone cannot offer.

Quick Win: Provide connections and non-obvious explanations to gain access to the vast majority of options to use Snowflake Intelligence. Beyond that, only expand semantic effort if you realize the necessity in your analysis.

Conclusion

When looking to establish Snowflake Intelligence as your always-on enterprise helper, think outside the box. For example, using a Custom Tool with a function that retrieves longitude and latitude values from an external API can provide more accuracy than relying on inference from customers' ZIP codes. Make use of custom tools for your agents to provide automated weekly reports sent to the relevant employee via email. Or, allow specific User Defined Functions to call specific external APIs for further information, for example stock numbers or weather forecast data.

Contact us to
learn more!



Dr. Johannes Lewen

SNOWFLAKE EXPERT



+49 221 669575-0



commercial@areto.de

areto.group

areto.group

areto consulting GmbH
Schanzenstraße 6-20,
51063 Cologne
www.areto.de
Tel: +49 221 66 95 75-0
info@areto.de