## 

## 

## 

## 

**Lab Overview**

Welcome to the Snowflake Intelligence hands-on lab for Zoomi Pet Wellness Analytics! In this lab, you'll learn how to harness the power of Snowflake's latest AI capabilities to transform pet wellness data into actionable insights. Using Snowflake Intelligence, you'll create AI agents that can analyze device performance, predict maintenance needs, and automate monitoring—all using natural language.

**Lab Information**

* **Prerequisites:**
* Snowflake account with ACCOUNTADMIN role
* Basic SQL knowledge
* Understanding of data analytics concepts

**Learning Objectives**

After completing this lab, you will be able to:

* Create and configure Snowflake Intelligence agents for specific analytical tasks
* Use natural language to query complex data relationships across multiple datasets
* Set up automated monitoring and alerting systems for device performance
* Implement predictive maintenance using AI-driven insights
* Build a comprehensive pet wellness analytics system using Snowflake Intelligence

**Business Context**

**Zoomi Pet Wellness** is a leading provider of smart devices for pet health monitoring. The company faces several challenges:

* Managing a growing fleet of IoT devices across multiple regions
* Ensuring consistent device performance and data accuracy
* Predicting maintenance needs before device failures occur
* Analyzing complex relationships between pet health data and device metrics

**Snowflake Intelligence** helps address these challenges by:

* Automating device performance analysis using AI agents
* Enabling natural language queries for complex data analysis
* Providing predictive maintenance capabilities
* Streamlining monitoring and alerting processes

**Zoomi Data Overview for This Lab:**

In this lab, we will be working with the following enhanced datasets, central to Zoomi's operations:

* **Pet Owners (pet\_owners.csv)**
  + Customer information
  + Pet profiles
  + Device assignments
  + Foreign Key: device\_id
    - Description: Represents pet owners who have registered their pets in the Zoomi system. This data helps us understand our customer base and enables customer segmentation and multi-device ownership analysis.
* **Device Sales (device\_sales\_by\_region.csv)**
  + Regional sales performance
  + Revenue metrics
  + Device distribution patterns
  + Primary Key: device\_id
    - Description: Tracks sales performance of our pet Fitbit devices across different geographical regions, crucial for business analytics, strategic planning, time-series analysis, and product category insights.
* **Device Inventory (fitbit\_inventory.csv)**
  + Device specifications
  + Manufacturing details
  + Performance metrics
  + Primary Key: device\_id
    - Description: Provides details on the inventory of our pet wellness devices, essential for supply chain, logistics, product performance, and quality control analysis.
* **Veterinary Feedback (vet\_feedback.csv)**
  + Professional assessments
  + Device performance ratings
  + Health impact analysis
  + Foreign Key: device\_id
    - Description: Contains real-world notes and observations from veterinarians after pet checkups or device usage. This data provides rich qualitative insights into pet health and device effectiveness, enabling structured analysis of device effectiveness and pet adaptation.

### **Key Data Relationships (for Lab Context):**

* **Device Sales to Inventory:** Linked via DeviceID.
* **Pet Owners to Devices:** Connected through the registered\_devices field in PET\_OWNERS (which contains DeviceIDs).
* **Vet Feedback to Devices:** Associated using DeviceID.
* **Device Performance to Sales:** Correlated through Device Type and Price Tier across linked tables.
* **Geographic Analysis:** Possible across sales (Region) and vet clinics (ClinicZipCode).

## 

## **Step 1: Snowflake Intelligence Setup**

This initial step ensures your Snowflake account is properly configured to support Snowflake Intelligence. You will enable the necessary features on your demo account and set up the underlying database objects and user permissions that Snowflake Intelligence relies on. This foundational setup is crucial before you load any Zoomi data or build any agents.

**Important Notes before you begin:**

* Users for Snowflake Intelligence need to have a default role and a default warehouse set.
* The database where Snowflake Intelligence metadata is stored **must** be named Snowflake\_intelligence.
* If you attempt to access Snowflake Intelligence at this point, it will likely "spin forever" because the necessary underlying parts have not yet been set up. This process will change in the future.
* Snowflake Intelligence (SI) will utilize your **default role** and **default warehouse** for operations. It is crucial to set these up correctly for SI to work.

### **1.1 Enable Snowflake Intelligence Features**

You will need to enable specific features on your demo account. This is done by logging into your Snowflake deployment and then running a SQL command.

1. **Log in** to your Snowflake account via the deployment URL
2. **Open a new SQL worksheet**.
3. **Run the following SQL command** to enable the required features. *Note:* Replace <<account locator>> with your actual Snowflake account locator.  
   SQL

  
ALTER ACCOUNT <<account locator>>

SET UI\_ENABLE\_AI\_ML\_FEATURE\_19 = 'ENABLED'

, ENABLE\_DATA\_TO\_ANSWER = true

, COPILOT\_ORCHESTRATOR\_PARAM\_10 = true

, COPILOT\_ORCHESTRATOR\_PARAM\_13 = true

, UI\_ENABLE\_AI\_ML\_FEATURE\_29 = 'ENABLED'

parameter\_comment = 'Enable Snowflake Intelligence - SE Demo';

  
This command enables the necessary UI and backend features for Snowflake Intelligence and its underlying AI/ML capabilities to appear in Snowsight.

### **1.2 Create Metadata Objects for Snowflake Intelligence**

Snowflake Intelligence stores its configuration and other operational metadata within your Snowflake environment. You'll create the roles, database, and warehouse needed to support this.

1. **Open a new SQL worksheet**.
2. **Ensure you are using the ACCOUNTADMIN role** to perform these creations and grants.  
   SQL

  
USE ROLE ACCOUNTADMIN;



1. **Configure cross-region access for Claude (if applicable):** This setting is needed if the Claude LLM is not in your current region.  
   SQL

  
ALTER ACCOUNT SET CORTEX\_ENABLED\_CROSS\_REGION = 'ANY\_REGION';



1. **Create the SNOWFLAKE\_INTELLIGENCE\_ADMIN\_RL role:** This role will be used for managing Snowflake Intelligence objects.  
   SQL

  
CREATE ROLE Snowflake\_intelligence\_admin\_rl;

-- To allow this role to create databases, which is needed for the Snowflake\_intelligence database

GRANT CREATE DATABASE ON ACCOUNT TO ROLE Snowflake\_intelligence\_admin\_rl;

1. **Create the Snowflake\_intelligence\_wh warehouse:** This warehouse will be used for Cortex Search Service creation and query execution.  
   SQL

  
CREATE WAREHOUSE Snowflake\_intelligence\_wh WITH WAREHOUSE\_SIZE = 'X-SMALL';

GRANT USAGE ON WAREHOUSE Snowflake\_intelligence\_wh TO ROLE Snowflake\_intelligence\_admin\_rl;



1. **Create the Snowflake\_intelligence database:** This database will hold configuration and other objects to support Snowflake Intelligence. Remember, the database **must** be named Snowflake\_intelligence.  
   SQL

  
CREATE DATABASE Snowflake\_intelligence;

GRANT OWNERSHIP ON DATABASE Snowflake\_intelligence TO ROLE Snowflake\_intelligence\_admin\_rl;



1. **Dynamically grant Snowflake\_intelligence\_admin\_rl to the current user:** This ensures that the user running these commands will have the necessary permissions to proceed with the lab.  
   SQL

  
DECLARE

sql\_command STRING;

BEGIN

sql\_command := 'GRANT ROLE Snowflake\_intelligence\_admin\_rl TO USER ' || CURRENT\_USER() || ';';

EXECUTE IMMEDIATE sql\_command;

RETURN 'Role Snowflake\_intelligence\_admin\_rl granted successfully to user ' || CURRENT\_USER();

END;

1. **Set up stages and tables for configuration:** You'll now switch to the newly created admin role and set up the necessary schemas and a stage within the Snowflake\_intelligence database.  
   SQL

  
USE ROLE Snowflake\_intelligence\_admin\_rl;

USE DATABASE Snowflake\_intelligence;

-- Set up a temp schema for file upload (only temporary stages will be created here).

CREATE OR REPLACE SCHEMA Snowflake\_intelligence.temp;

GRANT USAGE ON SCHEMA Snowflake\_intelligence.temp TO ROLE PUBLIC;

*-- OPTIONAL: Set up stages and tables for configuration - you can have your semantic models be anywhere else, just make sure that the users have grants to them*

CREATE SCHEMA IF NOT EXISTS config;

USE SCHEMA config;

CREATE STAGE semantic\_models ENCRYPTION = (TYPE = 'SNOWFLAKE\_SSE');



1. **Agent Configuration:** This script is needed to store all your agent information and creates an Agent Policy that controls which roles can see which agents.  
   SQL

  
USE ROLE Snowflake\_intelligence\_admin\_rl;

CREATE SCHEMA IF NOT EXISTS Snowflake\_intelligence.agents;

-- Make SI agents in general discoverable to everyone.

GRANT USAGE ON SCHEMA Snowflake\_intelligence.agents TO ROLE PUBLIC;

CREATE OR REPLACE ROW ACCESS POLICY Snowflake\_intelligence.agents.agent\_policy

AS (grantee\_roles ARRAY) RETURNS BOOLEAN ->

ARRAY\_SIZE(FILTER(grantee\_roles::ARRAY(VARCHAR), role ->

IS\_ROLE\_IN\_SESSION(role))) <> 0;

-- Create an agent config table. Multiple tables can be created to give granular UPDATE/INSERT permissions to different roles.

CREATE OR REPLACE TABLE Snowflake\_intelligence.agents.config (

agent\_name VARCHAR NOT NULL,

agent\_description VARCHAR,

grantee\_roles ARRAY NOT NULL,

tools ARRAY,

tool\_choice OBJECT,

tool\_resources OBJECT,

response\_instruction VARCHAR,

sample\_questions ARRAY,

CONSTRAINT pk\_agent\_name PRIMARY KEY (agent\_name)

)

WITH ROW ACCESS POLICY Snowflake\_intelligence.agents.agent\_policy ON (grantee\_roles);

GRANT SELECT ON TABLE Snowflake\_intelligence.agents.config TO ROLE PUBLIC;



### **1.3 Configure Default Role and Warehouse**

Snowflake Intelligence uses your **DEFAULT ROLE** and **DEFAULT WAREHOUSE**. You need to change your default role to the newly created Snowflake\_intelligence\_admin\_rl. This ensures all objects you create in the lab will be owned by the correct role and prevents permission issues.

1. **Change your current role in Snowsight:**
   * Click on your current role in the top right corner (e.g., ACCOUNTADMIN).
   * Select Switch Role.
   * Choose SNOWFLAKE\_INTELLIGENCE\_ADMIN\_RL from the list.
   * Click on the three dots next to the role and select Set as default.
2. **Log out of your Snowflake account and log back in.**
   * Sometimes this resolves permission issues and confirms that your defaults have been set correctly.

## **Step 2: Load Zoomi Data**

In this step, you will load our Zoomi-specific datasets into your Snowflake environment. For our lab, we will use a dedicated database SI\_EVENTS\_HOL to house these tables. You will upload the provided Zoomi CSV files and import them as individual tables through the Snowflake UI.

**Important Note:** Ensure you are still operating with the SNOWFLAKE\_INTELLIGENCE\_ADMIN\_RL role and the Snowflake\_intelligence\_wh warehouse set as your current context.

### **2.1 Create Zoomi Lab Database**

First, you'll create a new database specifically for this lab's data. This database will live alongside your Snowflake\_intelligence database created in Step 1.

1. **Log into your Snowflake Demo Account.**
2. **Navigate to Data > Databases** in the Snowsight UI.
3. Click the **+ Database** button in the top right corner.
4. In the "New Database" dialog:
   * **Name:** Enter SI\_EVENTS\_HOL.
   * **Creating as:** Make sure SNOWFLAKE\_INTELLIGENCE\_ADMIN\_RL is selected as the owner role.
   * Click **Create**.
5. This creates the main database where all your Zoomi lab tables will reside.

### **2.2 Download Zoomi Data Files**

You will need the Zoomi data CSV files to upload into Snowflake.

1. **Download and unzip** the Snowflake Intelligence Festival Data.zip file (from the original lab, we assume this contains a placeholder for your Zoomi data files in a real scenario, you'd provide the actual Zoomi CSVs here).
   * Inside, you should find the following Zoomi-rebranded files:
     + pet\_owners.csv
     + device\_sales\_by\_region.csv
     + vet\_feedback.csv
     + fitbit\_inventory.csv

### **2.3 Create Tables from Zoomi Data Files via UI**

You will now create four tables needed for this lab by importing each of the Zoomi CSV files through the Snowflake UI. We will load these tables into the PUBLIC schema of your newly created SI\_EVENTS\_HOL database.

**Repeat the following steps for each of the four files:**

* pet\_owners.csv (will become the PET\_OWNERS table)
* device\_sales\_by\_region.csv (will become the DEVICE\_SALES\_BY\_REGION table)
* vet\_feedback.csv (will become the VET\_FEEDBACK table)
* fitbit\_inventory.csv (will become the FITBIT\_INVENTORY table)

#### **Instructions for Each File Upload:**

1. **In Snowsight, navigate to Data > Databases**.
2. **Find and expand SI\_EVENTS\_HOL**, then click on the **PUBLIC** schema.
3. Click the **+ Create** button in the top right corner.
4. From the dropdown, select **Table from File**.
5. In the "Load Data into Table" wizard:
   * **Load Data**:
     + Click **Browse** and select one of your Zoomi CSV files (e.g., pet\_owners.csv).
     + Click **Next**.
   * **Destination**:
     + Ensure the Schema is SI\_EVENTS\_HOL.PUBLIC.
     + Under Select or create a table, choose Create new table.
     + For Name, enter the corresponding table name (e.g., PET\_OWNERS).
     + Click **Next**.
   * **File format**:
     + Verify Delimited files (CSV or TSV) is selected.
     + Click **View options**.
     + In the File Format pop-up, ensure Header is set to **First line contains header**.
     + Click Apply.
     + Click **Next**.
   * **Columns**:
     + Review the inferred schema. Ensure DATA TYPE and COLUMN NAME look correct for your data. You may need to adjust data types (e.g., NUMBER for DeviceRevenueUSD and DevicesSold).
     + Click **Next**.
   * **Load**:
     + Click **Load**.
     + You should see a "Successfully Loaded Data" message.
     + Click **Done**.
6. **Repeat these steps for device\_sales\_by\_region.csv, vet\_feedback.csv, and fitbit\_inventory.csv, naming their tables DEVICE\_SALES\_BY\_REGION, VET\_FEEDBACK, and FITBIT\_INVENTORY respectively.**

### **2.4 Verify Loaded Tables**

Once you have completed loading all four Zoomi files, you should verify that all tables are present in the PUBLIC schema of your SI\_EVENTS\_HOL database.

1. **In Snowsight, navigate to Data > Databases**.
2. **Expand SI\_EVENTS\_HOL**, then expand **PUBLIC**, and click on **Tables**.
3. You should now see the following tables listed:
   * DEVICE\_SALES\_BY\_REGION
   * FITBIT\_INVENTORY
   * PET\_OWNERS
   * VET\_FEEDBACK
4. You have now successfully loaded all Zoomi data into Snowflake!

## **Step 3: Create a Simple Agent**

In this step, you will create your very first Snowflake Intelligence Agent. This initial agent will be a simple one, designed primarily to demonstrate the agent creation process and its basic conversational capabilities, powered by an LLM (Large Language Model) like Claude. While this agent won't yet be connected to your Zoomi data, it's a crucial first step in understanding the agent framework.

An agent in Snowflake Intelligence is a tangible entity you create that will include one or more Cortex building blocks, like an analyst semantic model or a search service, which you will add in later steps.

**Important Note:** The Agents section in Snowsight should now work with the config tables you created in Step 1. The page might attempt to redraw at this point, but that is expected.

### **3.1 Navigate to Agent Creation**

1. **In Snowsight, navigate to AI & ML > Agents** in the left-hand navigation pane.
2. Click the  
    **+ Create agent** button in the top right corner of the Agents page.  
   This action will open the "New Agent" configuration dialog.

### **3.2 Configure Your Simple Agent: "Scout"**

You will now define the basic properties for your first agent, "Scout".

1. In the "New Agent" dialog:
   * **Name:** Enter Scout.
   * **Agent Overview:** Give it a description such as: This agent helps answer general questions about Snowflake. (Or, This agent is connected securely to Claude in Snowflake.).
   * **Response instruction:** (Optional for this simple agent, but you can add: Provide concise and helpful answers.)
   * Leave the Knowledge section empty for now, as we will add semantic models and search services in later steps.
   * Leave Users empty for now; by default, only you (and roles higher than your current default) will have access.
   * **Sample questions:** Click + Add question and add the following sample question:
     + What are some time series functions I can use in Snowflake?.
2. Click **Create** in the bottom right corner.

### **3.3 Verify Agent Creation**

After clicking Create, the dialog will close, and you should see your new agent listed.

1. **Refresh the screen** if the agent does not immediately appear.
2. You should now see "Scout" listed under your agents with its overview.

You have successfully created your first simple agent, "Scout"! In the next steps, you will learn how to interact with it and enhance its capabilities by connecting it to your Zoomi data.

## **Step 4: Go to Snowflake Intelligence**

Now that you've set up the foundational components and created your "Scout" agent, it's time to access the Snowflake Intelligence application. This will allow you to switch between the standard Snowsight interface and the dedicated Snowflake Intelligence application, which provides the conversational AI experience.

### **4.1 Navigate to Snowflake Intelligence Application**

1. **In Snowsight, locate the AI & ML section** in the left-hand navigation pane.
2. **Click on Snowflake Intelligence** within the AI & ML menu.  
   [Image showing Snowsight left navigation, with 'AI & ML' expanded and 'Snowflake Intelligence' highlighted with an arrow. (Similar to original image 0)]  
   This action will open the Snowflake Intelligence application, typically in a new browser tab or within the same Snowsight interface, with the navigation focused on Snowflake Intelligence.

### **4.2 Interact with Your Simple Agent**

Upon entering Snowflake Intelligence, you'll see a welcoming interface. You can now interact with the simple agent you created in the previous step.

1. You should see a greeting such as "Good afternoon, [Your Username]" and "How can I help you?".
2. Below the main prompt, you'll see a dropdown menu.  
    **Click on this dropdown** (it might currently display "Claude" or a default agent name).
3. **Select Scout** from the list of available agents. This ensures your interactions are routed through the agent you just created.  
   [Image showing the Snowflake Intelligence UI with "Good afternoon" message, the input field, and the dropdown for selecting an agent, with 'Scout' selected or highlighted in the dropdown. (Similar to original image 1, but with 'Scout' instead of 'Claude')]
4. In the chat input box, **type the following question**:  
    What can you tell me about the company Snowflake?
5. Press **Enter** or click the send arrow.

You will observe that Snowflake Intelligence, through your "Scout" agent, provides a general answer about Snowflake as a cloud-based data warehousing company, including its core business and key features. This basic agent interacts solely with the underlying LLM (like Claude) and does not yet access your specific Zoomi data.

We have currently developed a basic agent lacking access to data within your Snowflake account. It solely interacts with Claude (or the default LLM). To enhance its intelligence, we will now proceed to integrate a semantic model and connect it to the agent, thereby providing improved context for Zoomi-specific questions.

## **Step 5: Create a Semantic Model for Zoomi Insights**

Now that you've got your basic "Scout" agent, it's time to make it smarter by teaching it about your Zoomi business data. A semantic model is a key component that bridges the gap between complex database schemas and business user vocabulary, allowing Snowflake Intelligence to generate highly accurate results from your structured data. You'll define, refine, and test your first Zoomi semantic model using Cortex Analyst.

### **5.1 Enable Directory Table for Semantic Models Stage**

The semantic models you create will be stored in YAML files within a stage. To enable Snowflake Intelligence to discover and use these files, you need to enable a directory table on that stage.

1. **In Snowsight, navigate to Data > Databases** in the left-hand navigation pane.
2. **Expand SNOWFLAKE\_INTELLIGENCE**, then expand CONFIG, and click on SEMANTIC\_MODELS under Stages.
3. On the right-hand side, you will see a section titled "View Files with Directory Tables". Click the **Enable Directory Table** button.  
   This creates a directory table that constantly scans your SEMANTIC\_MODELS stage for new semantic model definitions.

### **5.2 Start Creating a New Semantic Model via Cortex Analyst**

You will now initiate the creation of your first Zoomi semantic model using the Cortex Analyst studio.

1. **In Snowsight, navigate to AI & ML > Studio** in the left-hand navigation pane.
2. On the Snowflake AI & ML Studio page, locate the **Cortex Analyst** tile and click **+ Try**.  
   This will open the "Create Semantic Model" wizard.

### **5.3 Configure Semantic Model Storage and Details**

The wizard will first ask you to define where your semantic model will be stored and to provide a name and description.

1. On the "Getting started" page:
   * Under Location to store, ensure:
     + The first dropdown is set to SNOWFLAKE\_INTELLIGENCE.CONFIG.
     + The second dropdown is set to SEMANTIC\_MODELS.
   * **Name:** Enter zoomi\_wellness\_insights.
   * **Description (optional):** Enter This model provides insights into Zoomi's pet device sales, inventory, and owner data..
   * The File name should automatically update to zoomi\_wellness\_insights.yaml.
2. Click **Next: Select tables** in the bottom right corner.

### **5.4 Select Tables for Your Semantic Model**

Now, you will select the Zoomi data tables that this semantic model will be built upon.

1. On the "Select tables" page:
   * In the Search bar, you can type SI\_EVENTS\_HOL to easily find your lab database.
   * **Expand SI\_EVENTS\_HOL**, then expand PUBLIC.
   * **Select the following tables** by checking their checkboxes:
     + DEVICE\_SALES\_BY\_REGION
     + FITBIT\_INVENTORY
     + PET\_OWNERS
     + VET\_FEEDBACK
2. Click **Next: Select columns**.

### **5.5 Select Columns and Refine Semantic Definitions**

On this page, you will choose which columns to include in your semantic model and refine their semantic meaning (dimensions, facts, time dimensions) to help Cortex Analyst generate accurate queries.

1. On the "Select columns" page:
   * For simplicity in this lab, **select all columns** by checking the top checkbox next to Columns.
   * Click **Create and Save** in the bottom right corner.
2. Snowflake will now generate a starter semantic model definition for you.
3. You will be redirected to the "Semantic Model" editor view. Here, you can review the automatically generated logical tables, dimensions, and facts.
4. **Refine Column Categorization (Move Facts to Dimensions):** The model might incorrectly identify certain ID columns as "facts" due to their numeric data type, but they are often "dimensions" used for joining or filtering. You need to adjust these for better query generation.
   * For the DEVICE\_SALES\_BY\_REGION table, click Edit next to it.
     + Locate DEVICES\_SOLD and DEVICE\_REVENUE\_USD – these should correctly be Facts.
     + Locate REGION – this should correctly be a Dimension.
   * For the PET\_OWNERS table, ensure OWNER\_NAME, PHONE, and EMAIL are Dimensions.
   * For the FITBIT\_INVENTORY table, ensure DEVICE\_TYPE, SHIPPING\_ADDRESS, MANUFACTURING\_COUNTRY are Dimensions.
   * For the VET\_FEEDBACK table, ensure CLINIC\_ZIP\_CODE, NOTE\_ID are Dimensions, and VET\_NOTES is a Dimension (as it's text, it's typically treated as a dimension for direct retrieval, even if its content will be used for search later).
5. *Note:* The original lab specifically called out TICKET\_ID, EVENT\_ID, CUSTOMER\_ID from the TICKET\_SALES table and CUSTOMER\_ID from the CUSTOMERS table to be moved from Facts to Dimensions. You will need to make similar conceptual adjustments based on your Zoomi data if any IDs are miscategorized. For our Zoomi tables, our IDs and numeric values are already well-categorized based on common data types, so explicit moves might not be strictly necessary, but it's good practice to check.
6. **Add Synonyms for Clarity:** Synonyms help Cortex Analyst understand natural language variations.
   * Find the DEVICE\_SALES\_BY\_REGION logical table and click Edit.
   * Locate the REGION dimension. Click Edit next to it.
   * In the Synonyms field, add common variations people might use for regions, such as area, territory.
   * Click Save.
   * You can repeat this for other relevant columns, like DeviceType (e.g., model, product line) or VetNotes (e.g., veterinarian comments).

### **5.6 Define Table Relationships**

Relationships are crucial for Cortex Analyst to correctly join tables when a query spans multiple datasets. By default, no relationships are created automatically, so you must define them.

1. In the semantic model editor, scroll down to the Relationships section. Click the **+** button to Add relationship.
2. **Define the Sales\_to\_Inventory Relationship:**
   * **Relationship name:** Sales\_to\_Inventory
   * **Join type:** inner (or left if you want to keep all sales records even if no matching inventory is found). For this lab, inner is fine.
   * **Relationship type:** many\_to\_one (many sales records for one device type in inventory).
   * **Left table:** DEVICE\_SALES\_BY\_REGION
   * **Right table:** FITBIT\_INVENTORY
   * **Left column:** DEVICE\_TYPE (You may need to infer a common joining column, assuming DEVICE\_SALES\_BY\_REGION has DeviceType or can be joined to another table that does. *If your device\_sales\_by\_region.csv does not have a DeviceType, you will need to add one or choose a different relationship. For this lab, let's assume DEVICE\_SALES\_BY\_REGION also has a DeviceType column that matches FITBIT\_INVENTORY.DeviceType for simplicity in modeling.*)
   * **Right column:** DEVICE\_TYPE
   * Click **Add**.
3. [Image showing the 'Add relationship' dialog with details for 'Sales\_to\_Inventory' relationship populated. (Similar to original image "Screenshot 2025-07-15 at 8.16.37 PM.png", but with Zoomi tables/columns).]
4. **Define the Owners\_to\_Sales Relationship (Conceptual):**
   * This relationship would connect PET\_OWNERS to DEVICE\_SALES\_BY\_REGION if there was a common OwnerID or CustomerID to link purchases to specific owners. For simplicity and given our current CSV fields, we'll keep this conceptual for now or base it on a logical link you anticipate (e.g., an implicit Email match if we were to normalize). For a more robust semantic model, this would be a crucial link. *For this lab, we will skip defining explicit relationships for PET\_OWNERS and VET\_FEEDBACK if they lack direct join keys to DEVICE\_SALES\_BY\_REGION or FITBIT\_INVENTORY within the structured data context.*

You should now have at least one relationship set up in your semantic model.

### **5.7 Test the Model**

It's critical to test your semantic model to ensure Cortex Analyst can correctly interpret natural language queries and generate valid SQL.

1. In the semantic model editor, use the side window (Cortex Analyst preview panel) to test a query.
2. In the prompt box, **type the following question**: What is the total device revenue in the "East" region?
3. Press **Enter** or click the Run button.
4. **Observe the results:** Cortex Analyst should provide an interpretation of your question, show the generated SQL query (Physical query), and display the results from your DEVICE\_SALES\_BY\_REGION table.  
   [Image showing Cortex Analyst preview panel with the question asked and the generated SQL/results. (Similar to original image "Screenshot 2025-07-15 at 8.16.26 PM.png", but showing Zoomi sales data results).]
5. **Test another query:** Count the total devices sold by region.
6. **Verify the Queries:** If these queries returned the correct values, you can add them as "verified queries." This helps train your model and improves its future performance.
   * Below the executed query, click + Verified query.
   * A dialog "Add verified query" will appear.
   * **Verified query name:** Enter a descriptive name (e.g., Regional Revenue Query).
   * **Question:** The question you just asked will pre-populate (e.g., What is the total device revenue in the "East" region?).
   * Optionally, check Add this question to onboarding questions.
   * Click **Add**.
   * Repeat this for any other successful test queries. These will now show up under "Verified Queries" in your model definition.
7. Excellent! Your semantic model is operating as anticipated.

### **5.8 Save the Model**

When you are done with your refinements and testing, make sure you save the model.

1. In the semantic model editor, click the **Save** button in the top right corner.  
   [Image showing the top right of the semantic model editor with the 'Save' button highlighted. (Similar to original image "Screenshot 2025-07-15 at 8.17.36 PM.png").]

You have now successfully created, refined, and tested your zoomi\_wellness\_insights semantic model, preparing it for use with your Snowflake Intelligence agents!

## **Step 6: Create an Agent to Use the Semantic Model**

Now that you have a powerful semantic model (zoomi\_wellness\_insights) that understands your Zoomi structured data, you will create a new Snowflake Intelligence agent. Unlike the simple "Scout" agent, this new agent will be directly connected to your semantic model, enabling it to answer business questions by querying your underlying data.

### **6.1 Start Creating a New Agent**

1. **In Snowsight, navigate to AI & ML > Agents** in the left-hand navigation pane.
2. Click the  
    **+ Create agent** button in the top right corner of the Agents page.  
   This action will open the "New Agent" configuration dialog.

### **6.2 Define Agent Details**

You will now provide the basic information for your new, smarter Zoomi agent.

1. In the "New Agent" dialog:
   * **Name:** Enter Zoomi Insights Agent.
     + *Original Lab Note:* The original lab used "Music Festival Agent" here. We are adapting this for our Zoomi context.
   * **Agent Overview:** Provide a description such as: An agent that can return information about Zoomi's pet device sales, inventory, and owner data..
   * **Response instruction (optional):** For improved responses, you can add: Provide clear and accurate insights based on Zoomi's structured data. If possible, include numerical figures.

### **6.3 Attach the Zoomi Semantic Model**

This is the crucial part where you connect your agent to the semantic model you spent time building and refining.

1. In the Knowledge section, click the **+ Semantic model** button.
2. An "Add semantic model" dialog will appear.
3. In this dialog:
   * Ensure the dropdowns at the top are set to  
      SNOWFLAKE\_INTELLIGENCE.CONFIG and SEMANTIC\_MODELS respectively.
   * You should see your zoomi\_wellness\_insights.yaml file listed.  
      **Select it**.
   * **Display name:** Enter Zoomi Data. This name will be displayed in Snowflake Intelligence as a data source for this agent.
4. Click **Select**.
5. (Optional) Add sample questions relevant to Zoomi's structured data, such as:
   * What is the total device revenue for the "South" region?
   * How many devices were sold in total last quarter?
6. Click **Create** to finalize the creation of your Zoomi Insights Agent.

You have now successfully created a new agent that is directly connected to your zoomi\_wellness\_insights semantic model! This agent is now equipped to understand and respond to questions about your structured Zoomi sales and inventory data.

## **Step 7: Test with Smart Agent**

Now that you've created your Zoomi Insights Agent and connected it to the zoomi\_wellness\_insights semantic model, it's time to put its intelligence to the test! You will navigate back to the Snowflake Intelligence application and ask questions about your structured Zoomi data, observing how the agent interprets your queries and generates insights.

### **7.1 Select Your Smart Agent**

1. If you are not already in the Snowflake Intelligence application, either click the "Snowflake Intelligence" tab in your browser or navigate to AI & ML > Snowflake Intelligence in Snowsight.
2. Refresh the page if necessary.
3. In the Snowflake Intelligence interface, locate the dropdown menu (it might still show "Scout" or "Claude"). Click on it and **select Zoomi Insights Agent** from the list.  
   [Image showing the Snowflake Intelligence UI with "Good afternoon" message and the agent dropdown, with 'Zoomi Insights Agent' selected or highlighted. (Similar to original image 1, but with 'Zoomi Insights Agent' as the selected agent).]  
   You should also see that the selected data source now shows "Zoomi Data", which is what you called your zoomi\_wellness\_insights.yaml semantic file earlier.

### **7.2 Query Zoomi Sales Data**

Let's ask your Zoomi Insights Agent a question that requires it to query the DEVICE\_SALES\_BY\_REGION table.

1. In the chat input box, **type the following question**: Show me the top 5 regions by devices sold.
2. Press **Enter** or click the send arrow.
3. **Observe the results:**
   * You should see "Thinking completed" with a "1 verified query" indicator.
   * The agent will provide a breakdown of the top regions by devices sold, likely in a tabular format, and perhaps a chart if the feature auto-generates one.
   * You can click **Show Details** to see the "Thinking steps", which will reveal the SQL query generated by Cortex Analyst and executed against your DEVICE\_SALES\_BY\_REGION table. This demonstrates how the agent leverages your semantic model to convert natural language into executable SQL.
4. [Image showing the Snowflake Intelligence chat interface with the question "Show me the top 5 regions by devices sold." asked, the "Thinking completed" status, and the tabular results. Optionally, an expanded "Show Details" section showing the generated SQL. (Similar to original images "Screenshot 2025-07-15 at 8.19.55 PM.png" and "Screenshot 2025-07-15 at 8.20.00 PM.png", but with Zoomi data/terms).]

### **7.3 Query Device Sales Ratios**

Let's try another question to see how the agent handles calculating ratios from your structured data.

1. In the chat input box, **type the following question**: What is the ratio of devices sold for each device type?
2. Press **Enter** or click the send arrow.
3. **Observe the results:**
   * The agent should again show "Thinking completed".
   * It will provide a breakdown, showing each DeviceType, the DevicesSold count for it, and the percentage of the total, similar to a pie chart or a proportional breakdown.
4. [Image showing the Snowflake Intelligence chat interface with the question "What is the ratio of devices sold for each device type?" asked, the "Thinking completed" status, and the tabular results with DeviceType, Count, and Percentage. (Similar to original image "Screenshot 2025-07-15 at 8.20.05 PM.png", but with Zoomi data/terms).]

### **7.4 Query Pet Owners and Sales (Leveraging Relationships)**

Now, let's ask a question that would ideally leverage relationships between tables (if you defined them effectively, or if the model can infer connections through synonyms). This query aims to identify repeat customers or those with multiple devices.

1. In the chat input box, **type the following question**: Did any pet owner register multiple devices?  
   *Note:* This question assumes your PET\_OWNERS and DEVICE\_SALES\_BY\_REGION tables (or an intermediary table) could be joined by a common identifier or that Cortex Analyst is smart enough to infer this. If your initial semantic model setup didn't explicitly create this join or sufficient synonyms, the answer might be less precise.
2. Press **Enter** or click the send arrow.
3. **Observe the results:**
   * If successful, the agent should provide an answer indicating if any pet owners are associated with multiple device registrations, potentially listing the most frequent ones.
   * If the answer is not as expected, this highlights the importance of reviewing the accuracy of defined relationships and identifying any potential synonyms that could enhance the model's intelligence and robustness.
4. [Image showing the Snowflake Intelligence chat interface with the question "Did any pet owner register multiple devices?" asked, the "Thinking completed" status, and potential results listing pet owners and number of devices. (Similar to original image "Screenshot 2025-07-15 at 8.20.12 PM.png", but with Zoomi data/terms).]

You have successfully tested your Zoomi Insights Agent's ability to answer questions by leveraging the semantic model on your structured Zoomi data. Next, you will enhance your agent's capabilities by adding a search service for unstructured data.

## 

## **Step 8: Create Search Service for Vet Feedback**

So far, your Zoomi Insights Agent can work with structured data. Now, we'll equip it to understand and search through unstructured data – specifically, the VET\_FEEDBACK notes from your Zoomi operations. To enable searching of this text data, you will create a Cortex Search Service.

### **8.1 Start Creating a New Search Service**

1. **In Snowsight, navigate to AI & ML > Studio** in the left-hand navigation pane.
2. On the  
    Snowflake AI & ML Studio page, locate the **Cortex Search** tile and click **+ Try**.  
   This will open the "Create Search Service" wizard.

### **8.2 Configure Search Service Storage and Details**

The wizard will first ask you to define where your search service will be stored and to provide a name.

1. On the "New service" page:
   * Ensure  
      Role and Warehouse are set to SNOWFLAKE\_INTELLIGENCE\_ADMIN\_RL and SNOWFLAKE\_INTELLIGENCE\_WH, respectively.
   * Under Database and Schema, ensure:
     + The first dropdown is set to  
        SNOWFLAKE\_INTELLIGENCE.CONFIG.
     + The second dropdown is set to SEMANTIC\_MODELS. (This is where the original HOL places it, for consistency with the provided images, we will keep it here, though a dedicated search\_services schema might be conceptually cleaner in a production environment).
   * **Name:** Enter Zoomi\_Vet\_Feedback\_Search.
     + *Original Lab Note:* The original lab used "Festival\_Contract\_Search" here. We are adapting this for our Zoomi context.
2. Click  
    **Next: Select data** in the bottom right corner.

### 

### **8.3 Select Data to be Indexed**

Now, you will select the Zoomi table containing the unstructured text data (VET\_FEEDBACK) that you wish to search over.

1. On the "Select data to be indexed" page:
   * In the dropdown, select  
      SI\_EVENTS\_HOL.PUBLIC.
   * From the list of tables,  
      **select VET\_FEEDBACK** by checking its checkbox.
2. Click **Next: Select search column**.

### **8.4 Select a Search Column**

You need to choose the specific text-based column within your VET\_FEEDBACK table that your queries will search over.

1. On the "Select a search column" page:
   * From the list of columns for  
      VET\_FEEDBACK, **select VET\_NOTES** by checking its checkbox. This is the column containing the detailed veterinary observations you want to make searchable.
2. Click **Next: Select attributes**.

### **8.5 Select Attribute Columns (Optional)**

Attribute columns are additional fields that you might wish to use as filters when querying the search service, or to display alongside the search results.

1. On the "Select attribute column(s)" page:
   * **Select CLINIC\_ZIP\_CODE and NOTE\_ID** by checking their checkboxes. These can be useful for contextualizing search results or filtering notes from a specific clinic.
   * You can also leave VET\_NOTES selected here, as it's the primary text column.
2. Click **Next: Select columns to include in the service**.

### 

### **8.6 Select Columns to Include in the Service**

This step specifies all columns whose data will be included in the search index and available for querying. Search and attribute columns must be part of the included columns.

1. On the "Select columns to include in the service" page:
   * **Leave all columns selected** by default. This ensures that  
      CLINIC\_ZIP\_CODE, VET\_NOTES, and NOTE\_ID are all included.
2. Click **Next: Configure indexing**.

### **8.7 Configure Indexing Parameters**

Finally, you will configure how often your search service will check for and materialize updates to its index.

1. On the "Configure your Search Service" page:
   * **Target Lag:** Select 1 from the dropdown for days. As this is a demonstration with unchanging data, a 1-day lag is inconsequential and sufficient.
   * **Embedding model:** Keep the default snowflake-arctic-embed-m-v1.5.
   * **Warehouse for indexing:** Ensure SNOWFLAKE\_INTELLIGENCE\_WH is selected.
2. Click **Create**.  
   Snowflake will now create and begin initializing your Zoomi\_Vet\_Feedback\_Search service.
   * **NOTE:** If your search service hangs in 'Initialize' for more than a few minutes, it might have failed. This is going to be fixed in the UI in the future. However, right now you can run: DESC CORTEX SEARCH SERVICE ZOO\_DB.UNSTRUCTURED\_DATA.Zoomi\_Vet\_Feedback\_Search; and look at the indexing\_error column. Chances are it is a missing permission on an object such as a database, schema, table, or warehouse.

You have successfully created a Cortex Search Service for your Zoomi vet feedback data! This service will allow your Zoomi Insights Agent to search and retrieve information from these unstructured notes, which you will add in the next step.

## **Step 9: Add Search Service to Agent**

You have a Zoomi Insights Agent that understands structured data via a semantic model, and you've created a Zoomi\_Vet\_Feedback\_Search service for unstructured data. Now, you will combine these capabilities by adding the search service to your Zoomi Insights Agent, making it a more comprehensive tool for your Zoomi business insights.

### **9.1 Confirm Search Service is Active (Important Pre-Check)**

Before adding the search service to your agent, it's crucial to confirm that the Zoomi\_Vet\_Feedback\_Search service is in an ACTIVE state. The service needs to be active before it can function. Completion is indicated when the Serving State shows as ACTIVE.

1. **In Snowsight, navigate to AI & ML > Cortex Search** in the left-hand navigation pane.
2. **Locate your Zoomi\_Vet\_Feedback\_Search service** in the list.
3. **Check the Serving State column**. It should show ACTIVE. If it's still  
    INITIALIZING, please wait a few minutes and refresh the page until it becomes ACTIVE.
   * *Troubleshooting:* If the service hangs in INITIALIZING for more than a few minutes, it might have failed. You can run  
      DESC CORTEX SEARCH SERVICE SNOWFLAKE\_INTELLIGENCE.CONFIG.Zoomi\_Vet\_Feedback\_Search; in a SQL worksheet and look at the indexing\_error column for more details. Ensure your default warehouse has  
      USAGE privileges granted to SNOWFLAKE\_INTELLIGENCE\_ADMIN\_RL and that SNOWFLAKE\_INTELLIGENCE\_ADMIN\_RL has SELECT access to the VET\_FEEDBACK table.
4. [Image showing Cortex Search list, with 'Zoomi\_Vet\_Feedback\_Search' service highlighted and its Serving State showing 'ACTIVE'. (Similar to original image "Screenshot 2025-07-16 at 8.36.53 AM.png", but with 'Zoomi\_Vet\_Feedback\_Search').]

### **9.2 Edit Your Existing Agent**

You will now modify your Zoomi Insights Agent to include the new search capability.

1. **In Snowsight, navigate to AI & ML > Agents** in the left-hand navigation pane.
2. From the list of agents,  
    **select your Zoomi Insights Agent** by clicking on its name. This will open the agent's configuration panel.  
   [Image showing Agents list, with 'Zoomi Insights Agent' selected. (Similar to original image "Screenshot 2025-07-16 at 8.36.19 AM.png", but with 'Zoomi Insights Agent').]

### **9.3 Add the Search Service to the Agent**

You will now attach the Zoomi\_Vet\_Feedback\_Search service to your agent.

1. In the  
    Knowledge section of the agent configuration, click the **+ Search service** button.
2. An "Add search service" dialog will appear.
3. In this dialog:
   * Ensure the dropdown at the top is set to SNOWFLAKE\_INTELLIGENCE.CONFIG (where you created the search service).
   * You should see your Zoomi\_Vet\_Feedback\_Search listed. **Select it**.
   * **Display name:** Enter Vet Feedback Notes. This name will be displayed in Snowflake Intelligence as a data source for this agent.
   * **URL column:** This identifies the column that contains the document's URL, allowing Snowflake Intelligence to provide links to sources cited in responses. Select URL if your search service had a URL column, or leave blank if VET\_FEEDBACK doesn't contain a URL field for this lab. *For our VET\_FEEDBACK table, we did not explicitly include a URL column, so you can leave this blank or select NOTE\_ID if you want to link to the note identifier as a conceptual "source link". For this lab, leaving it blank is fine.*
4. Click **Add**.
5. After adding the search service, ensure you click **Save** on the main agent configuration page to apply the changes to your Zoomi Insights Agent.

Your Zoomi Insights Agent is now powered by both structured data (via the semantic model) and unstructured data (via the search service). In the next step, you will test its enhanced capabilities!

## **Step 10: Test with Updated Agent**

Your Zoomi Insights Agent has been supercharged! It's now connected to both your structured Zoomi sales data (via the semantic model) and your unstructured VET\_FEEDBACK notes (via the search service). In this step, you will test its enhanced capabilities by asking questions that require it to leverage the search service.

### **10.1 Access Snowflake Intelligence and Select Your Agent**

1. If you are not already in the Snowflake Intelligence application, either click the "Snowflake Intelligence" tab in your browser or navigate to AI & ML > Snowflake Intelligence in Snowsight.
2. Refresh the page if necessary.
3. In the Snowflake Intelligence interface, locate the agent dropdown (it might still show "Scout" or "Claude"). Click on it and **select Zoomi Insights Agent** from the list.  
   [Image showing the Snowflake Intelligence UI with agent dropdown, and 'Zoomi Insights Agent' selected. (Similar to original image "Screenshot 2025-07-16 at 8.39.27 AM.png", but with 'Zoomi Insights Agent' selected from the agent list after navigating to Snowflake Intelligence).]
4. Next to the agent dropdown, you should now see a dropdown for data sources. **Click on this dropdown** to verify that both Zoomi Data (your semantic model) and Vet Feedback Notes (your search service) are listed and checked. This confirms that your agent is configured to utilize both data sources.  
   [Image showing the Snowflake Intelligence UI with the data source dropdown open, displaying 'Zoomi Data' and 'Vet Feedback Notes' both checked. (Similar to original image "Screenshot 2025-07-16 at 8.39.34 AM.png", but with 'Zoomi Data' and 'Vet Feedback Notes').]

### **10.2 Query Unstructured Vet Feedback Data**

Now that your agent has access to the search service, let's ask a question that requires it to search through the VET\_FEEDBACK notes.

1. In the chat input box, **type the following question**: What are common observations about device battery life in vet notes?
2. Press **Enter** or click the send arrow.
3. **Observe the results:**
   * You should see "Thinking completed".
   * Notice that no SQL is generated or written in the "Thinking steps" (if you expand "Show Details") because the agent is using your search service. It should provide an answer summarizing relevant information from your VET\_NOTES field, and ideally, it will show you the source of the document (or NoteID in our case, if you mapped NOTE\_ID as a URL column in Step 9) from which this information came.
4. [Image showing the Snowflake Intelligence chat interface with the question "What are common observations about device battery life in vet notes?" asked, the "Thinking completed" status, the text-based answer summarizing vet notes about battery life, and associated source numbers. (Similar to original image "Screenshot 2025-07-16 at 8.39.34 AM.png", but with Zoomi vet feedback content).]

### **10.3 Query Structured and Unstructured Data Together (Conceptual)**

While a specific example for a combined query might require a more complex semantic model and search service, conceptually, your Zoomi Insights Agent is now capable of answering questions that might span both types of data.

* For instance, a question like: Show me the sales trend for devices that frequently have "battery issues" in vet feedback. would conceptually leverage both the semantic model (for sales trend) and the search service (to identify devices with "battery issues"). While beyond the scope of a simple, direct demo for this step, it highlights the power of the combined agent.

You have successfully tested your updated Zoomi Insights Agent and observed its ability to search through unstructured VET\_FEEDBACK data, demonstrating its enhanced intelligence!

## **Step 11: Review and Key Takeaways for Zoomi**

Congratulations! You have successfully completed the Snowflake Intelligence Hands-on Lab for Zoomi, the Pet Wellness Tracker company. This lab explored the core capabilities and applications of Snowflake Intelligence, demonstrating how it empowers analysts and business users to efficiently query and gain insights from both structured and unstructured Zoomi data.

Throughout this lab, you experienced how Snowflake Intelligence enables analysts to efficiently query Snowflake data by leveraging a semantic model or the Cortex search service for data indexing.

### **Our Journey Through Snowflake Intelligence:**

Let's recap the steps we took to unlock intelligence from Zoomi's data:

1. **Snowflake Intelligence Setup:** You set up the necessary metadata for Snowflake Intelligence in your Snowflake account. This foundational step ensures the environment is ready for agent creation and data interaction.
2. **Load Zoomi Data:** You ingested Zoomi's critical operational data – including pet owner information, device sales, inventory, and crucial vet feedback notes – into your Snowflake database.
3. **Create a Simple Agent (Scout):** You created a basic agent that primarily utilized an LLM (like Claude) for general questions, showing the foundational conversational capabilities without direct data access.
4. **Go to Snowflake Intelligence:** You navigated to the Snowflake Intelligence application and began interacting with your simple agent.
5. **Create a Semantic Model (zoomi\_wellness\_insights):** You built a sophisticated semantic model over your structured Zoomi data (device sales, inventory, pet owners), enabling Snowflake Intelligence to understand business terms and query this data effectively.
6. **Create an Agent to Use the Semantic Model (Zoomi Insights Agent):** You created a new, smarter agent and connected it to your zoomi\_wellness\_insights semantic model, allowing it to answer specific questions about Zoomi's structured sales performance and inventory.
7. **Test with Smart Agent:** You actively queried your Zoomi Insights Agent on structured data, witnessing its ability to interpret natural language, generate SQL, and provide accurate results and even visualizations.
8. **Create Search Service (Zoomi\_Vet\_Feedback\_Search):** You built a Cortex Search Service over your unstructured VET\_FEEDBACK notes, preparing this rich text data for intelligent search.
9. **Add to Agent:** You enhanced your Zoomi Insights Agent further by integrating the Zoomi\_Vet\_Feedback\_Search service, making it capable of querying both structured and unstructured data.
10. **Test with Updated Agent:** You tested your Zoomi Insights Agent's new capabilities, successfully asking questions that required it to search through the unstructured vet feedback.

You have now written prompts in Snowflake Intelligence that utilized your semantic model and/or your search service to get data from your Snowflake instance.

### **Zoomi's Path Forward with Snowflake Intelligence:**

As a Pet Wellness Tracker company, Zoomi can now leverage Snowflake Intelligence to:

* **Self-serve data exploration:** Empower business users across sales, product, and customer support to create charts and get instant answers using natural language, without needing technical expertise or waiting for custom dashboards.
* **Democratize intelligence:** Allow teams to access and analyze thousands of data sources simultaneously, going beyond basic AI tools that only handle single documents.
* **Comprehensive integration:** Seamlessly analyze structured sales and inventory data together with unstructured vet feedback notes, connecting insights from various data types simultaneously.
* **Automatic security:** Benefit from existing Snowflake security controls, including role-based access and data masking, which automatically apply to all AI interactions and conversations, ensuring pet owner data and sensitive business metrics remain secure.

Customers like Zoomi can now easily build a chatbot-like interface for analysts and search their Snowflake assets using simple wizards, requiring minimal coding. This eliminates the need for complex custom Python scripts (though that remains an option), enabling faster value realization with GenAI within Snowflake.

## 

## **Step 12: Optional - Reset your default warehouse and default role**

If you'd like to return your Snowflake account to its previous default role and warehouse, follow these steps:

1. **In Snowsight, click on your current role** in the top right corner (currently SNOWFLAKE\_INTELLIGENCE\_ADMIN\_RL).
2. Select **Switch Role**.
3. Choose your preferred default role (e.g., ACCOUNTADMIN or your personal user role).
4. Click on the three dots next to the role and select **Set as default**.
5. Repeat this process for your default warehouse, selecting your preferred default warehouse (if different from Snowflake\_intelligence\_wh).
6. **Log out of your Snowflake account and log back in** to ensure these changes take effect.

Thank you for participating in the Zoomi Snowflake Intelligence Hands-on Lab! We hope this experience has provided valuable insights into the transformative power of conversational AI for data analytics.

**Enhanced Data Structure**

The lab uses the following enhanced datasets:

**1. Device Sales Data (device\_sales\_by\_region.csv)**

* Region - Geographic location of the sale
* DeviceRevenueUSD - Revenue generated from the sale
* DevicesSold - Number of devices sold
* SaleDate - Date of the sale
* DeviceType - Type of device sold
* PriceTier - Pricing category (Basic, Standard, Premium)
* DeviceID - Unique identifier for the device

**2. Pet Owner Information (pet\_owners.csv)**

* name - Pet owner's name
* phone - Contact number
* email - Email address
* pet\_type - Type of pet (Dog, Cat, Bird, etc.)
* pet\_age - Age of the pet
* registered\_devices - DeviceIDs associated with the owner
* owner\_id - Unique identifier for the owner
* subscription\_tier - Service tier level

**3. Device Inventory (fitbit\_inventory.csv)**

* DeviceType - Category of device
* ShippingAddress - Delivery location
* ManufacturingCountry - Country of manufacture
* DeviceID - Unique identifier for the device
* BatteryLife - Battery life in hours
* WaterResistant - Water resistance capability
* HealthMetrics - Health parameters monitored
* FirmwareVersion - Current firmware version
* BatchNumber - Manufacturing batch identifier

**4. Veterinary Feedback (vet\_feedback.csv)**

* ClinicZipCode - Location identifier for the veterinary clinic
* VetNotes - Detailed observations from the veterinarian
* NoteID - Unique identifier for the feedback entry
* DeviceID - Associated device identifier
* FeedbackDate - Date of the veterinary consultation
* DevicePerformance - Performance rating (1-5)
* BatteryStatus - Battery condition assessment
* DataAccuracy - Accuracy of collected data (%)
* PetResponse - Pet's adaptation to the device
* RecommendedActions - Suggested follow-up steps

**Data Relationships**

* Device Sales to Inventory: Linked via DeviceID
* Pet Owners to Devices: Connected through registered\_devices field
* Vet Feedback to Devices: Associated using DeviceID
* Device Performance to Sales: Correlated through DeviceType and PriceTier

**Enhanced Analytics Capabilities**

* Device Performance Analysis: Track device reliability and performance across different types and batches
* Customer Satisfaction Metrics: Correlate device features with veterinary feedback and pet response
* Geographic Insights: Analyze sales patterns and device performance by region
* Health Monitoring Effectiveness: Evaluate accuracy and reliability of different device types
* Quality Control: Track manufacturing batches and identify potential issues
* Battery Life Analysis: Monitor battery performance across device types and usage patterns

**Performance Optimization Best Practices**

**Data Clustering**

* Cluster tables by DeviceID for faster joins
* Create materialized views for commonly accessed metrics
* Use appropriate partitioning for large tables (e.g., by region or date)

**Query Optimization**

* Leverage result caching for frequently accessed data
* Implement appropriate indexing strategies
* Use efficient join patterns based on DeviceID relationships

**Resource Management**

* Configure warehouse sizes based on data processing needs
* Implement auto-suspend and auto-resume for cost efficiency
* Use resource monitors to control compute costs