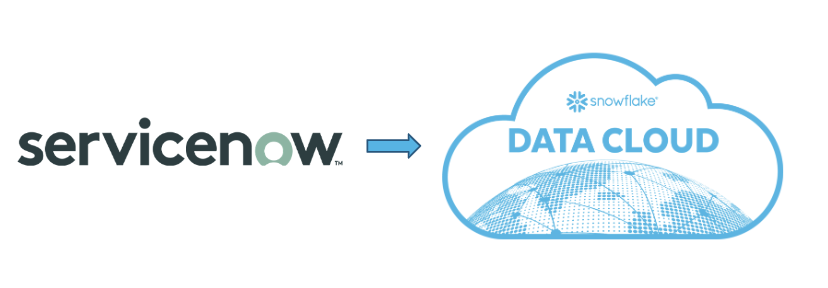
author: sfc-gh-drichert id: servicenow\_to\_snowflake\_connector summary: Step-by-step to set up Servicenow connector categories: Connectors environments: web status: Private feedback link: https://github.com/Snowflake-Labs/sfguides/issues tags: Connectors, Data Engineering, Servicenow

# Snowflake Connector for ServiceNow Installation

## Overview

Duration: 1

Use this quickstart lab to configure and understand the Snowflake Connector for ServiceNow using the Snowsight wizard, select some tables, ingest data, and run an example query. This quickstart is not meant to be exhaustive. Please check the [Snowflake Conector for ServiceNow documentation](https://https://other-docs.snowflake.com/en/connectors/servicenow/servicenow-index.html) for full functionality and limitations.



now

aside positive Note: This quickstart assumes you do not have a ServiceNow account, so it guides you through the steps of creating a developer account. Of course, if you do have a Servicenow account, please feel free to try it out, with the caveat that, at the time of writing, the connector is in public preview and should not be used for production. ### Prerequisites - ServiceNow account with administrator’s rights. - ORGADMIN rights to Accept the Terms of Service in the Snowflake Marketplace. - ACCOUNTADMIN rights on the Snowflake account where you will install the connector.

### What You’ll Learn

* How to set up the Snowflake Connector for ServiceNow.
* How to ingest ServiceNow data into Snowflake
* How to stop the connector to avoid unnecessary costs in a development environment. ### What You’ll Need
* A [Snowflake](https://snowflake.com/) Account
* A [ServiceNow](https://developer.servicenow.com/dev.do/) developer account ### What You’ll Build A ServiceNow to Snowflake ingestion data flow.

## Set up the ServiceNow Developer Instance

Duration: 5

If you do not want to test this connector on your ServiceNow account, no problem, this step explains how to set up a developer instance!

1. Go to the [ServiceNow developer website](https://developer.servicenow.com), and create a developer user.
2. Log on to the developer website with your newly created user and select **Create an Instance**.
3. Choose an instance type. You receive an email with your instance URL, and admin user and password.

Deployment is usually pretty quick, around five minutes. But, while you wait let’s go to the next step and configure Snowflake! ## Create and set up the Snowflake Account

### Create the Snowflake Account

If you do not have a Snowflake account, no problem, you can get a free trial at [snowflake.com](https://www.snowflake.com/en/).Select **Start for Free** and follow the instructions. ### Accept the Terms & Conditions Duration: 1 1. Log on to your Snowflake account through the Snowsight web interface and change to the **orgadmin** role. 1. Select “Admin -> Billing & Terms”. 4. In the “Snowflake Marketplace” section, review the Consumer Terms of Service. 5. If you agree to the terms, select “Accept Terms & Conditions”.

### Set up a Virtual Warehouse

Duration: 3

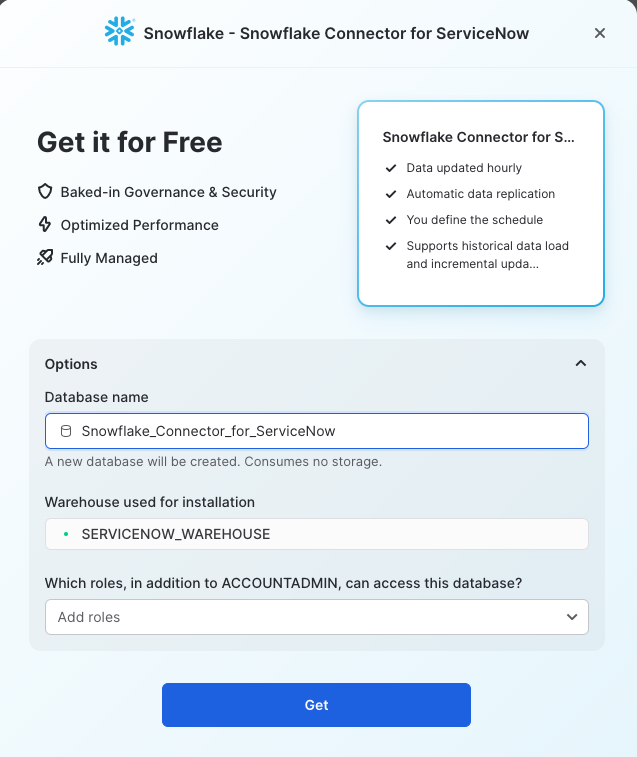
You’ll need some compute for the connector, so let’s set up a virtual warehouse to do that. A second virtual warehouse will be created automatically in the configure section.

Change to the **accountadmin** role. 1. Navigate to Admin -> Warehouses and select **+ Warehouse**. 2. Name the vitural warehouse **SERVICENOW\_CONNECTOR\_WH**, size XS, and, leaving the defaults, select **Create Warehouse**.

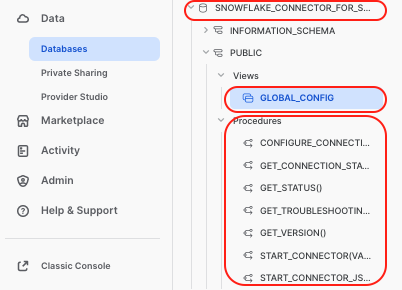
### Install the ServiceNow connector

Duration: 3

The connector, the first of its kind to be deployed on Snowflake’s native apps framework, is delivered through the Snowflake Marketplace, and is available to all Snowflake customers instantly. Once chosen, it is installed into your account as a database with several views, and stored procedures.

1. From the Snowflake Account Home page, select **Marketplace**.
2. In the search window, enter **ServiceNow** and select the tile.
3. Review the business needs and usage samples.
4. Select **Get**.
5. Select the warehouse you created above, **SERVICENOW\_CONNECTOR\_WH**.
6. Select **Options**.
7. For this lab, leave the default name for the installation database, **Snowflake\_Connector\_for\_ServiceNow**. Do not select any additional roles.
8. Select **Get**. You receive the following message, **Snowflake Connector for SeviceNow is now ready to use in your account.** 
9. Select **Done**. We will manage it in the next section.

Let’s check that the connector was installed. From Snowsight, go to **Data -> Databases**. You will see a new database with the name **Snowflake\_Connector\_for\_ServiceNow**. Open the Public schema and views to see the Global\_Config view. Some of the Procedures have also been installed. Others will appear after the installation finishes.



installed

## Set up the Snowflake to ServiceNow Oauth hand-shake

This section shows how to set up the Oauth handshake using the Snowsight user interface, which is *massively* simpler than managing all the bits through code.

Please have two tabs in your browser open for the next part, as you will have to copy some data from Snowflake to ServiceNow and vice-versa. \* From the Snowflake side, we want the connector to generate the **re-direct URL** which we will paste into the Application Registry, and \* From the ServiceNow side we want the Application Registry to provide the **Client id** and **password**, which we then paste into Snowflake.

### On the Snowflake hand

Duration: 4

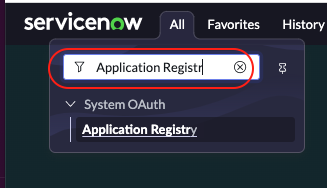
Launch the Snowflake Connector for ServiceNow from the **Marketplace** -> **Snowflake Connector for ServiceNow**. 1. Select **Manage**. 1. Select **Connect**. 1. Fill in the ServicNow instance details. This is the first part of the ServiceNow URL for your ServiceNow account, **without** the trailing *service-now.com*. 1. Select **OAuth2** for the Authentication method. 1. Copy the redirect URL. You will need it in the next section.

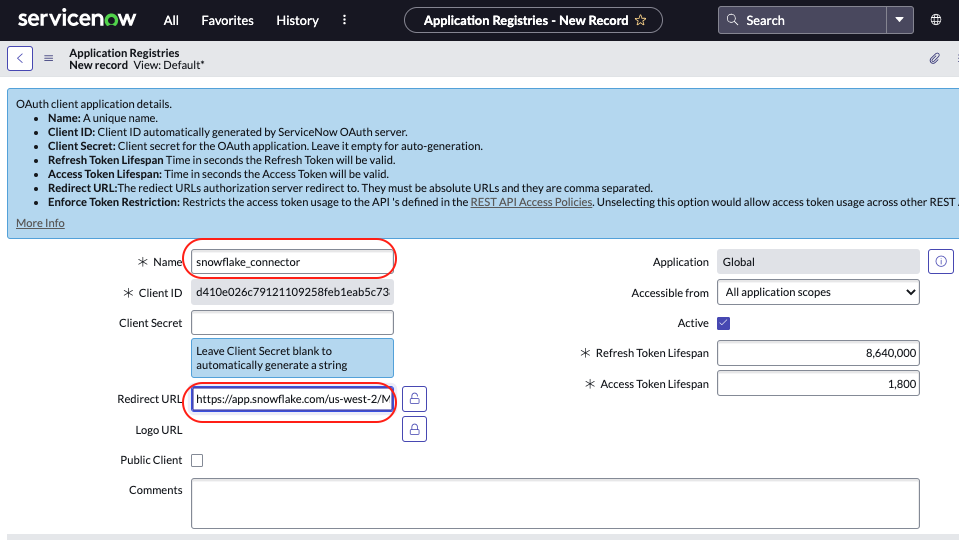
Now, open a new tab in your browser (without closing the above), and follow the steps in the next section.

### On the ServiceNow Other hand

Duration: 4

1. Log on to your ServiceNow developer instance.
2. From the main page, select **All** and search **Application Registry**.

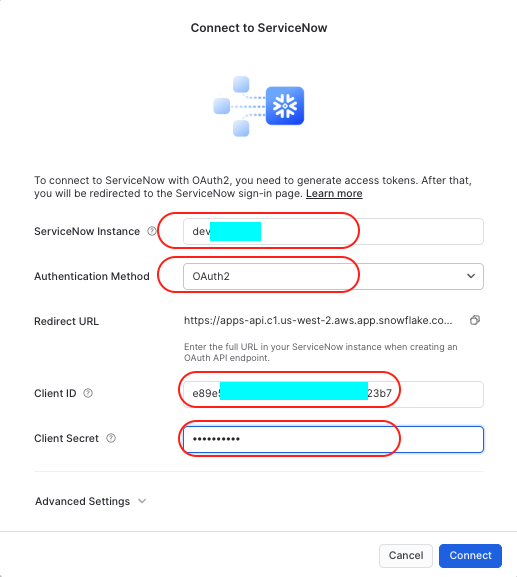
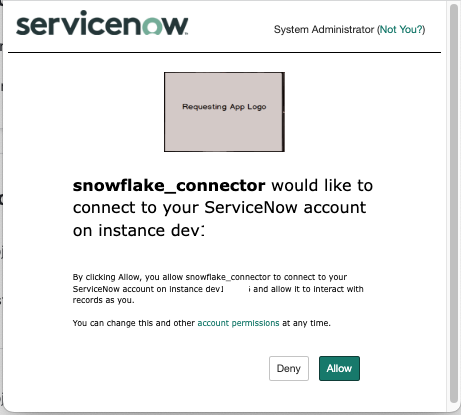
 1. Select **New** in the upper right-hand side of the window. 1. Select **Create an OAuth API endpoint for external clients**. 1. Give the endpoint a name, such as **Snowflake\_connector**. Leave the client secret blank. This will autofill. 1. Paste in the redirect URL that was generated on the Snowflake hand.

 1. Select **Submit**. The window closes. 1. Select the registry you just created to re-open it. 1. Note that the **Client id** and **Client secret** are auto-generated. 1. Copy the **Client id**.

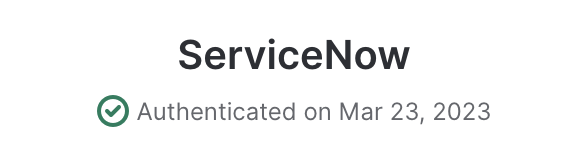
Now, time to jump back to the Snowflake configuration tab.

### Now Let’s Shake

Duration: 1

1. Paste the **Client id** from ServiceNow into the Snowflake configure pop-up.
2. Go back to the ServiceNow tab and copy the **Client secret** and paste it into the Snowflake configure pop-up.
3. No need to change the Advanced Settings, but feel free to check them out. 
4. Select **Connect**. Your ServiceNow accounts pops up and requests to connect to Snowflake. 
5. Select **Allow**. The connection is established between the two systems.

To verify the connection, select the three dots […] and **View Details**. At the top of the pop-up you will see the date **ServiceNow** Authenticated.



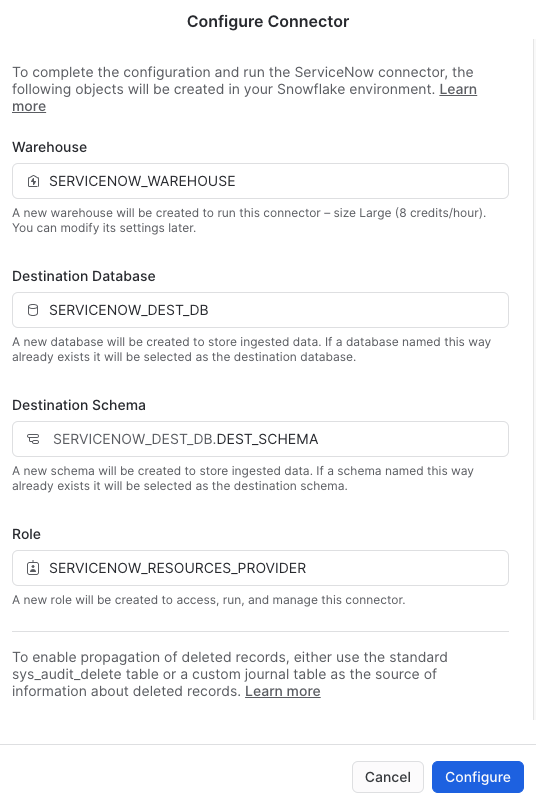
authenticated

Select **Done**.

aside negative If you are having issues, perhaps the Client secret wasn’t copied. Unlock the password field and copy and paste the text. ## Configure the Connector Duration: 5

Under the status for the connector, which displays “Choose Resources”, select **Configure**.

This displays the Configure Connector dialog. By default, the fields are set to the names of objects that are created when you configure the connector.

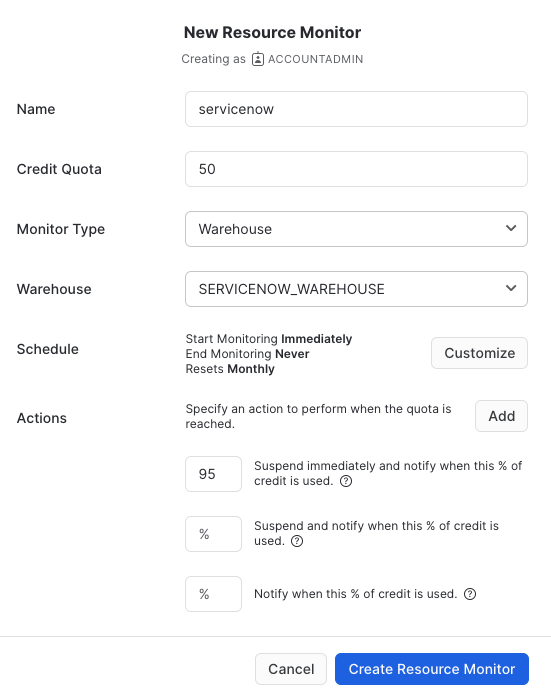


default config

Check out [Configuring the Snowflake Connector for ServiceNow](https://other-docs.snowflake.com/en/connectors/servicenow/servicenow-installing-ui.html#configuring-the-snowflake-connector-for-servicenow) for more information on these fields.

Select **Configure**. The dialog box closes and the status of the connector changes to Provisioning. It can take a few minutes for the configuration process to complete.

aside negative Watch out!!! The created warehouse is created as a **Large** and with a auto timeout of 10 minutes. So this means, if you set to refresh every hour, the Large warehouse (8 credits/hour) will wake up for a minimum of 10 minutes every hour. For this lab, you don’t need all the power! Go to Admin-> Warehouses -> SERVICENOW\_WAREHOUSE -> … > Edit, and change this to an XSMALL, and the auto timeout to one minute. In a real-life use case, a Large warehouse size is often needed.

aside positive Absolutely attach a resource monitor to the SERVICENOW\_WAREHOUSE. Go to Admin->Resource Monitors->+ Resource Monitor, and create a warehouse resource monitor: 

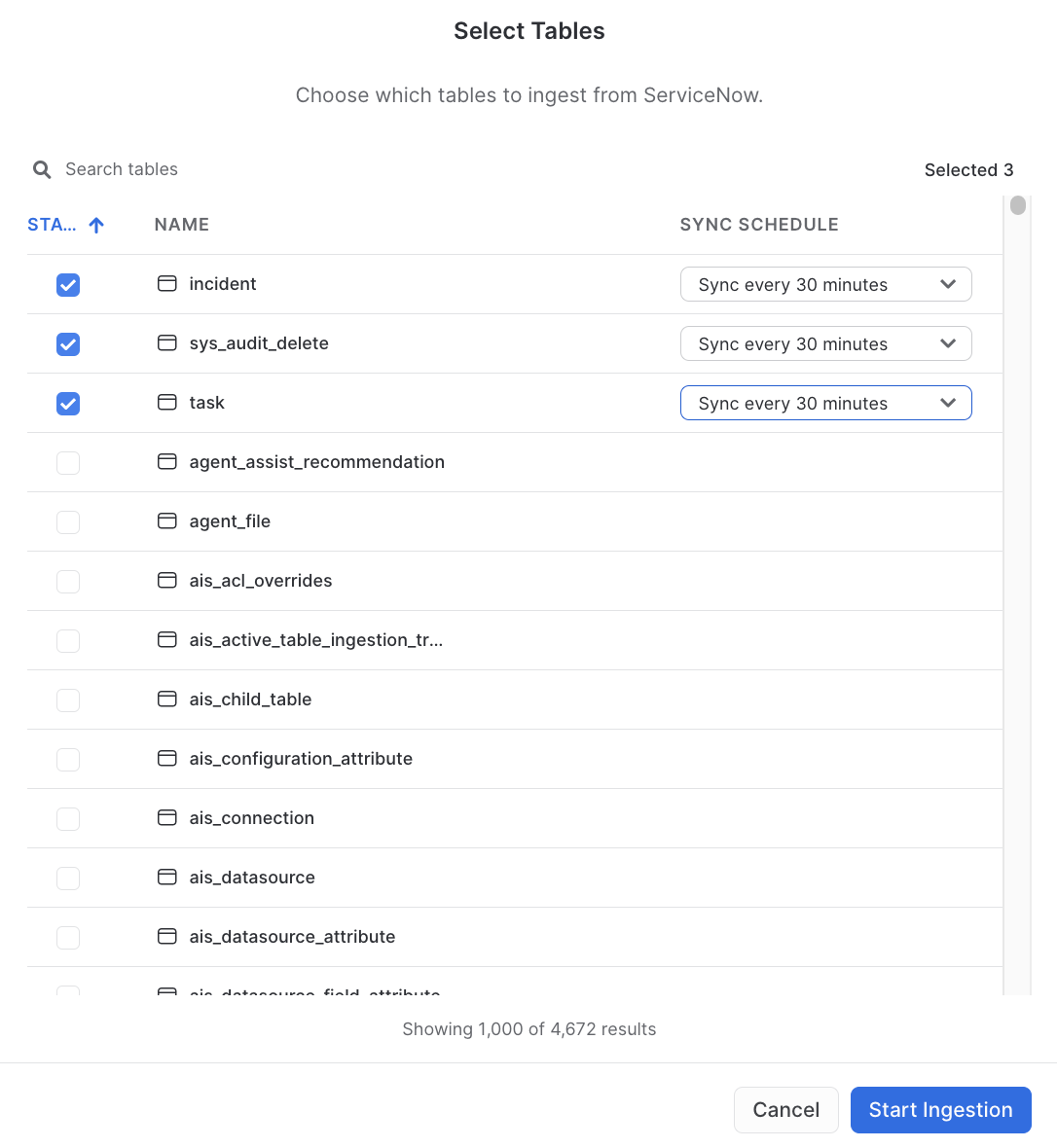
## Select ServiceNow Tables

Duration: 4

aside negative A couple of things to be aware of: - The connector can only ingest tables with **sys\_id** columns present. - ServiceNow views are not supported. Instead of ingesting these views, you should synchronize all tables for the underlying view and join the synchronized tables in Snowflake. - Incremental updates occur only for tables with **sys\_updated\_on** or **sys\_created\_on** columns. - For tables that do not have sys\_updated\_on or sys\_created\_on columns, the connector uses **truncate and load** mode. In this mode, the table is always ingested using the initial load approach, and newly ingested data replaces the old data.

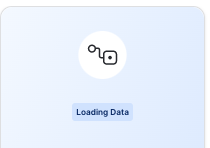
1. In the **Snowflake Connector for ServiceNow** window, under the status for the connector, which displays “Start Data Sync”, select **Select Tables**.
2. To be able to run our test query later, we need to ingest a couple of tables. From the search window enter **incident** and check the box next to it and choose a 30 minute sync time.
3. To choose other tables, clear the search, put the table name and select the checkbox. Do this for the following tables:
   * sys\_audit\_delete
   * task

aside positive Hint: Clear the search fields, and then select the title **Status** to sort and show all the tables you selected.



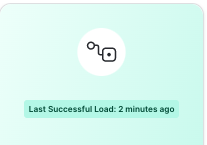
Select

1. Select **Start Ingestion**. The select windows closes and you get the message “Loading Data” from the main Connector window. In addition to the tables you choose, three system tables will also be loaded. These are necessary to build the views on the raw data: sys\_dictionary, sys\_db\_object, and sys\_glide\_object.



load

You receive a message indicating success. It appears once at least one table has been fully ingested.



success

aside negative Don’t stop the ingest too quickly. Make sure the views are built in the destination database first.

## Connector Monitoring (Query Sync History)

Duration: 5

In the connector interface, choose **Query Sync History.** A worksheet opens with several SQL queries you can execute to get monitoring information. Here are some examples:

// Get general information about all ingestions  
SELECT \* FROM SNOWFLAKE\_CONNECTOR\_FOR\_SERVICENOW.public.connector\_stats;  
  
// Search for information about particular table ingestions  
SELECT \* FROM SNOWFLAKE\_CONNECTOR\_FOR\_SERVICENOW.public.connector\_stats WHERE table\_name = '<table\_name>';  
  
// Check connector configuration  
SELECT \* FROM SNOWFLAKE\_CONNECTOR\_FOR\_SERVICENOW.public.global\_config;  
  
// Calculate ingested data volume  
WITH d as (  
 SELECT  
 table\_name,  
 last\_value(totalrows) OVER (PARTITION BY table\_name ORDER BY run\_end\_time) AS row\_count  
 FROM SNOWFLAKE\_CONNECTOR\_FOR\_SERVICENOW.public.connector\_stats  
)  
SELECT table\_name, max(row\_count) as row\_count FROM d GROUP BY table\_name ORDER BY table\_name;  
  
// Connector runtime (minutes from start)  
SELECT timediff('minute', min(run\_start\_time), max(run\_end\_time)) AS connector\_runtime\_in\_minutes  
FROM SNOWFLAKE\_CONNECTOR\_FOR\_SERVICENOW.public.connector\_stats;

## Setting reader role permissions

Duration: 3

Now that you have ingested some data, let’s create the **servicenow\_reader\_role** to give it access to the database, schema, future tables, future views, and virtual warehouse.

USE ROLE accountadmin;  
USE DATABASE SERVICENOW\_DEST\_DB;  
CREATE ROLE IF NOT EXISTS servicenow\_reader\_role IF NOT EXISTS;  
GRANT USAGE ON DATABASE SERVICENOW\_DEST\_DB TO ROLE servicenow\_reader\_role;  
GRANT USAGE ON SCHEMA DEST\_SCHEMA TO ROLE servicenow\_reader\_role;   
GRANT SELECT ON FUTURE TABLES IN SCHEMA DEST\_SCHEMA TO ROLE servicenow\_reader\_role;  
GRANT SELECT ON FUTURE VIEWS IN SCHEMA DEST\_SCHEMA TO ROLE servicenow\_reader\_role;  
GRANT SELECT ON ALL TABLES IN SCHEMA DEST\_SCHEMA TO ROLE servicenow\_reader\_role;  
GRANT SELECT ON ALL VIEWS IN SCHEMA DEST\_SCHEMA TO ROLE servicenow\_reader\_role;  
GRANT USAGE ON WAREHOUSE SERVICENOW\_WAREHOUSE TO ROLE servicenow\_reader\_role;

## Query the Data

Duration: 5

Check out the tables that the connector has created under the DEST\_SCHEMA of the SERVICENOW\_DEST\_DB database. For each table in ServiceNow that is configured for synchronization, the connector creates the following table and views:

* A table with the same name that contains the data in raw form, where each record is contained in a single VARIANT column.
* A view named table\_name\_\_view that contains the data in flattened form, where the view contains a column for each column in the original table and a row for each record that is present in the original table.

aside negative Warning! After you start the connector, it takes for the views to be created. The creation of the views relies on data in the ServiceNow sys\_db\_object, sys\_dictionary and sys\_glide\_object tables. The connector loads metadata from these ServiceNow tables after you enable any table for synchronization. It can take some time for the connector to load this metadata. Do not stop the warehouse during this time!

* A view named table\_name\_\_view\_with\_deleted that contains the same data as table\_name\_\_view as well as rows for records that have been deleted in ServiceNow.
* A table table\_name\_\_event\_log that contains the history of changes made to records in ServiceNow.

To query from the raw data, check out [Accessing the raw data](https://other-docs.snowflake.com/en/connectors/servicenow/servicenow-accessing-data.html#accessing-the-raw-data). To query the views (recommended), check out [Accessing the flattened data](https://other-docs.snowflake.com/en/connectors/servicenow/servicenow-accessing-data.html#accessing-the-flattened-data).

### Use this query to identify number of incidents raised by month and priority

Here’s a little test query for you to identify the number of incidents raised by month and priority. Other example queries are provided on the Snowflake Connector for ServiceNow page in the Marketplace.

USE ROLE SERVICENOW\_READER\_ROLE;  
USE DATABASE SERVICENOW\_DEST\_DB;  
USE SCHEMA DEST\_SCHEMA;  
  
WITH T1 AS (  
 SELECT  
 DISTINCT  
 T.NUMBER AS TICKET\_NUMBER  
 ,T.SHORT\_DESCRIPTION  
 ,T.DESCRIPTION  
 ,T.PRIORITY  
 ,T.SYS\_CREATED\_ON AS CREATED\_ON  
 ,T.SYS\_UPDATED\_ON AS UPDATED\_ON  
 ,T.CLOSED\_AT  
 FROM  
 TASK\_\_VIEW T  
 LEFT JOIN   
 INCIDENT\_\_VIEW I   
 ON I.SYS\_ID = T.SYS\_ID -- ADDITIONAL INCIDENT DETAIL  
 LEFT JOIN   
 SYS\_AUDIT\_DELETE\_\_VIEW DEL   
 ON T.SYS\_ID = DEL.DOCUMENTKEY -- THIS JOIN HELPS IDENTIFY DELETED TICKETS   
 WHERE  
 DEL.DOCUMENTKEY IS NULL -- THIS CONDITION HELPS KEEP ALL DELETED RECORDS OUT  
 AND  
 I.SYS\_ID IS NOT NULL -- THIS CONDITION HELPS KEEP JUST THE INCIDENT TICKETS  
)  
SELECT  
 YEAR(CREATED\_ON) AS YEAR\_CREATED  
 ,MONTH(CREATED\_ON) AS MONTH\_CREATED  
 ,PRIORITY  
 ,COUNT(DISTINCT TICKET\_NUMBER) AS NUM\_INCIDENTS  
FROM  
 T1  
GROUP BY  
 YEAR\_CREATED  
 ,MONTH\_CREATED  
 ,PRIORITY  
ORDER BY  
 YEAR\_CREATED  
 ,MONTH\_CREATED  
 ,PRIORITY  
;

## Setting the monitoring role permissions

If you would like to monitor errors, run stats, connector stats, enabled tables, you can set up a ServiceNow monitoring role that allows access to the views in the connector database. For example, run the following in a worksheet (and then use the role):

USE ROLE accountadmin;  
CREATE ROLE IF NOT EXISTS servicenow\_monitor\_role ;  
GRANT IMPORTED PRIVILEGES ON DATABASE SNOWFLAKE\_CONNECTOR\_FOR\_SERVICENOW TO ROLE servicenow\_monitor\_role;  
GRANT USAGE ON WAREHOUSE SERVICENOW\_WAREHOUSE TO ROLE servicenow\_monitor\_role;

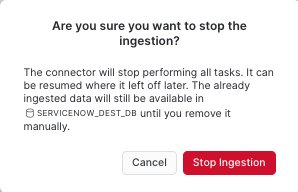
## Stop the Ingestion

Duration: 1

During this lab, we’re only ingesting the data, so it makes sense to stop the ingestion after that initial load. However, in an operational environment, you would keep it running.

aside negative If you do not stop the connector, it will wake up the virtual warehouse at the specified time interval and consume credits.

1. In Snowsight, select the **Snowflake Connector for ServiceNow** tile.
2. In the **Snowflake Connector for ServiceNow** window, select **Stop Ingestion**.



stop

Read the warning and select **Stop Ingestion**.

## Uninstall the Connector (but not the data)

If you are using the public preview connector, make sure to check out the limitations, one of which is during the preview period, before the connector is generally available, Snowflake will release an update that requires you to **uninstall** and reinstall the connector from the Snowflake Marketplace.

To drop the connector you need to drop the connector database:

DROP DATABASE SNOWFLAKE\_CONNECTOR\_FOR\_SERVICENOW;

## Conclusion

Duration: 1

Hey, congrats! You set up the Snowflake Connector for ServiceNow, ingested some data and ran a query!