Project 2 – Snowflake POC

Project 2 and Deliverables

Technology: Snowflake Trial Account (30 days) https://signup.snowflake.com
Important: Trial Accounts — Snowflake Documentation

Snowflake Links and Resources

Getting Started — Snowflake Documentation
Snowflake in 20 Minutes — Snowflake Documentation
Snowflake Quickstarts

Project Overview

Points Breakdown

- Deliverable 1: Git Set-up (5 points)
- Deliverable 2: Data load, Database, Tables, Views (30 points)
- Deliverable 3: Materialized Views, Clustering, Run script, Documentation (15 points)

This is a group project. All work must be completed as a team. Deliverables will only be accepted through a git repository that can be cloned by the instructor. Do not share git repository or solutions outside of your team except for the course instructor and TA.

All deliverables are due no later than midnight on the due date.

Background: You are a new data engineer at a product sales organization that is new to big data. They are trying to grow their data science team and you see an opportunity to prove yourself on the team by becoming their Snowflake SME. You have been asked to ingest the sales data from the data warehouse into Snowflake and prepare it for analysis and consumption. The data warehouse team has delivered a portion of the data to you in an S3 bucket to use in the Proof-of-Concept (POC).

The production environment is completely locked down to any data engineers and the only way to deploy your solutions to production is via git. The infrastructure team (your instructor) will clone your repository and run your scripts exactly as provided. An automated process should be able to deploy your solution. Assume the infrastructure team does not have good knowledge of the tools or how to troubleshoot for errors. The infrastructure team must be provided instructions and exact usable commands in the ReadMe file in your repository. As they are very busy, agile documentation is the expectation and high value is placed on concise, clear, and ordered steps.

Your development environment is an exact replication of production. It is well known that the infrastructure team will not make changes to the production environment in order to make your solution work. Requests for python packages, java libraries, or software installations will be denied. The admin team will run your scripts via the SnowSQL CLI Client. Make sure to test thoroughly. The instructor will not troubleshoot your code issues at run time.

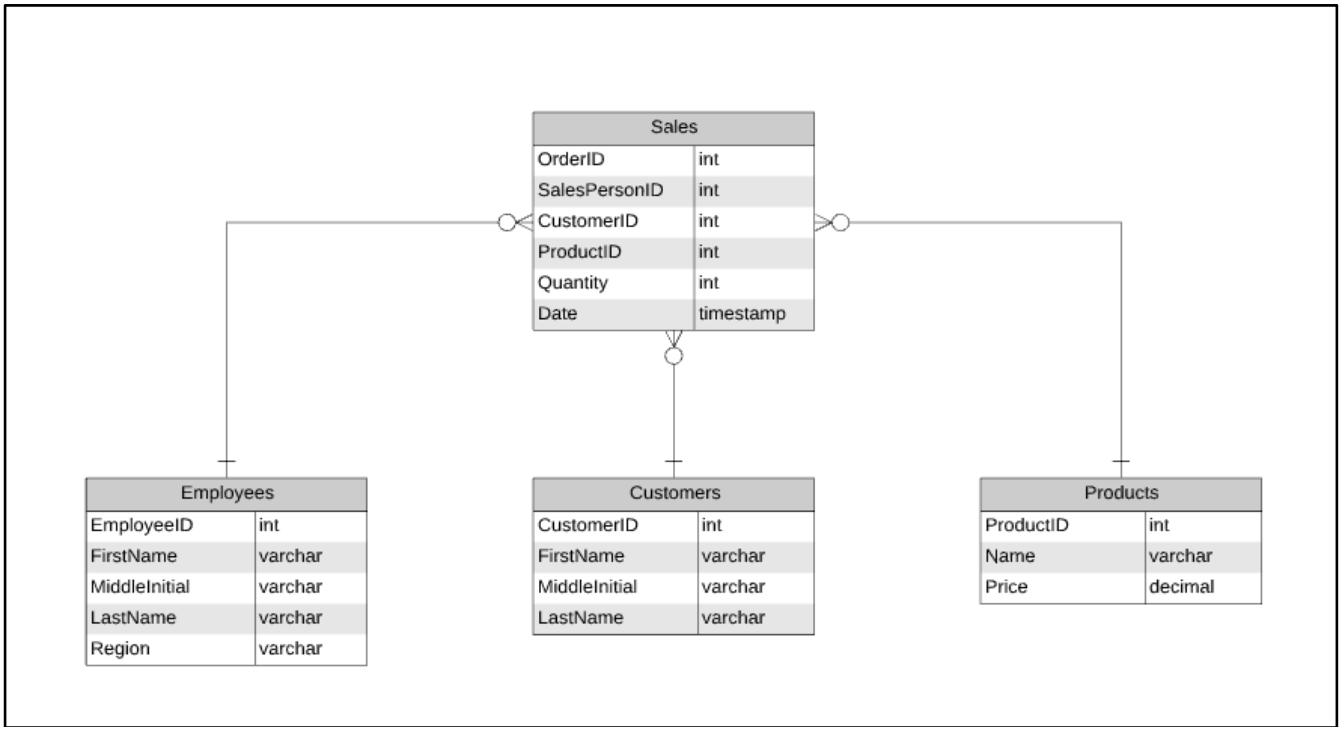
You have not been provided a QA member or team to assist in validating the data. You are responsible for ensuring data quality. The data you deliver will be assumed to be the final form and ready to use to make major business decisions.

Since the production team is hesitant about new data engineers, they will not deploy anything to production without having a script that can completely undo what you have done. Assume you will need a script that will drop all databases, schemas, data stages, tables and remove all data from the environment that can be run by the infrastructure team if they run into problems.

Important Note: Scripts will be run and validated in production with the CLI client.

- SnowSQL (CLI Client): SnowSQL (CLI Client) — Snowflake Documentation

Sales Data Model



Sales	
OrderID	int
SalesPersonID	int
CustomerID	int
ProductID	int
Quantity	int
Date	timestamp

Employees	
EmployeeID	int
FirstName	varchar
MiddleInitial	varchar
LastName	varchar
Region	varchar

Customers	
CustomerID	int
FirstName	varchar
MiddleInitial	varchar
LastName	varchar

Products	
ProductID	int
Name	varchar
Price	decimal

Deliverable 1: GIT Set up (5 points)

Create a private slack channel with your team's name. Make sure that the professors and TA are invited to the channel. Set up a public git repository at GitHub or BitBucket. Make sure the repository is open to the public and can be cloned by any user. Create a ReadMe that includes your team's name, your private slack channel name, and all names of the members on your team. You can use existing teams or create a new team. The final team must be formed by the due date for Deliverable 1.

Deliverable 2: Data load, Database, Tables, Views (30 points)

Step One - Get and Load Raw Data

1. S3 Data sources
   a. Data https:
      i. https://seng5709.s3.us-west-2.amazonaws.com/customers/
      ii. https://seng5709.s3.us-west-2.amazonaws.com/employees/
      iii. https://seng5709.s3.us-west-2.amazonaws.com/products/
      iv. https://seng5709.s3.us-west-2.amazonaws.com/sales/
   b. Data s3 URLs
      i. s3://seng5709/customers/
      ii. s3://seng5709/employees/
      iii. s3://seng5709/products/
      iv. s3://seng5709/sales/

   Assume the end point could have multiple files and create your script appropriately to get all files at each end point. File base names are Customers.csv, Employees.csv, Products.csv, Sales.csv. The file could be appended by a number for multiple files such as Products2.csv.

2. Create a stage(s) for the data in Snowflake
3. Load data to a stage (2 and 3 can be combined in one step with an external stage)
4. Create a database named `<yourTeamName>_sales` and in the new database also create a schema named `raw`. This will store the raw data. For example, if your team's name is "blizzard" your database would be named `"blizzard_sales" and` your schema would be named `"raw"`.
5. In your new `<yourTeamName>` database and schema `raw`, create tables and load the raw data from stage to the table.

Step Two - Prepare Data for Business Consumption.

Both business teams and future data delivery teams will be using this data for reporting and analysis. These next steps will prepare this data and ensure the data is ready for business consumption.

1. Do some quality analysis on the data. If you find any issues in the raw data, document the issues in your ReadMe file in the repository.
2. Create a new Snowflake schema in your team database called `curated`
3. Create tables in the sales schema from the raw data, correcting any issues you have found that will make the data easier to use.
4. Create three views on the data in your `curated` schema
   a. View: `customer_monthly_sales_2019_view`
      i. Customer id, customer last name, customer first name, year, month, aggregate total amount of all products purchased by month for 2019.
   b. View: `top_ten_customers_amount_view`
      i. Customer id, customer last name, customer first name, total lifetime purchased amount

        ii. This view should only return the top ten customers sorted by total dollar amount in sales from highest to lowest.

   c. View: `product_sales_view`
      i. Create a Snowflake product and sales view that includes columns for sales year and month.
      ii. OrderID, SalesPerson ID, Customer ID, Product ID, Product Name, Product Price, Quantity, Total Sales Amount, Order Date, Sales Year, Sales Month

Deliverable 3: Materialized Views, Clustering, Run Script, Drop Script, Documentation (15 points)

Helpful: https://docs.snowflake.com/en/user-guide/views-materialized.html

1. Read about Snowflake materialized views and clustering. In a section of your ReadMe, give two specific use cases where clustering and materialized views may be beneficial to the consumption of the sales data.
2. Provide a removal script that will drop all tables, schemas, databases, and remove all data from Snowflake environment.
3. Your ReadMe should contain information that shows how the code should be run on the production environment (Deployment RunBook section) information that might be helpful to the end user about the databases that were created, and information about the data (User Documentation section). Include a concise table list of SQL file run order. The table should contain the numerical run order, name of file, path in repository, and short description of what the file will create or do.