Snowflake Seminar

A diagram of a cloud computing structure

AI-generated content may be incorrect.

A diagram of a computer

AI-generated content may be incorrect.

Hands-on Lab

<https://quickstarts.snowflake.com/guide/tasty_bytes_introduction/#0>

<https://quickstarts.snowflake.com/guide/tasty_bytes_introduction/#3>

<https://github.com/Snowflake-Labs/sf-samples/tree/main>

<https://github.com/Snowflake-Labs/sf-samples/blob/main/samples/tasty_bytes/FY25_Zero_To_Snowflake/tb_introduction.sql>

<https://quickstarts.snowflake.com/>

New points learned from above seminar:

* Grant Ownership gives higher privilege than Grant ALL. With the later one you can’t delete the object.
  + From DeepSeek: In Snowflake, "GRANT OWNERSHIP" transfers full control of the warehouse, including the ability to modify or delete it. "GRANT ALL" provides all privileges (e.g., usage, modification) but not ownership. "GRANT OWNERSHIP" gives higher privilege as it includes the ability to transfer ownership and manage all aspects of the warehouse.
* There is a command to grant **future** object (very neat, if you know you are going to allow user to access all the objects in e.g., a database/schema, you don’t have to setup access for each of them.
* In Snowflake, **apply masking** refers to using **dynamic data masking** to control access to sensitive data. Masking policies define how data is obscured (e.g., hiding, partial display) for users based on their roles. When a query is executed, the masking policy automatically applies the defined rules, ensuring sensitive information is only visible to authorized users. This helps maintain data security and compliance without altering the underlying data.
* This seminar uses the approach of loading data into S3 buckets first.
  + You can specify the file format before hands if you load your data into staging area.
  + The source data is stored in a public s3 buckets in AWS, somehow Snowflake can access it. **I need to check whether I can access it without Snowflake connection.!!**

A diagram of a cube with arrows and arrows

AI-generated content may be incorrect.

A diagram of data transformation

AI-generated content may be incorrect.

A diagram of data transformation

AI-generated content may be incorrect.

<https://www.snowflake.com/virtual-hands-on-lab/?_x_zm_rtaid=6kTKU0FRSga8zxuInwODsA.1739401886623.46719569a4011127b2b1304eb6f3671e&_x_zm_rhtaid=12>

<https://quickstarts.snowflake.com/guide/getting_started_with_dynamic_tables/index.html#0>

Dynamic Tables can be used in

* BI (materialized view)
* Streaming Analytics (Slow Change Dimension)
* Data Sharing & Collaboration (private/public listing, filter multi-tenant data)
* A screenshot of a computer

  AI-generated content may be incorrect.

<https://quickstarts.snowflake.com/guide/data_engineering_pipelines_with_snowpark_python/#0>

A screenshot of a computer

AI-generated content may be incorrect.

Snowpark

Learning notes:

1. Snowpark Dataframe is a distinct one designed by Snowflake, working best with Snowflake.
   1. It’s different from pandas or spark dataframe.
   2. A white background with black text

      AI-generated content may be incorrect.
   3. <https://medium.com/@frulouis/snowpark-dataframe-api-vs-snowflake-pandas-api-which-is-right-for-you-12b2a6ba5ded>
   4. Check this official documents to understand the best practice working with Snowpark DataFrame APIs: <https://docs.snowflake.com/en/developer-guide/snowpark/python/working-with-dataframes>
   5. This is the official documents on how to use SparkSQL commands. <https://spark.apache.org/docs/latest/api/python/reference/pyspark.sql/index.html>
2. Github Codespaces
   1. You can directly interact with your codes through a cloud-environment.
   2. Automatically initiated an vs-code environment for you.
   3. <https://docs.github.com/en/codespaces/overview>
3. Snowpark makes the process of creating a UDF, Stored Procedure or Streamlit App much simpler.
   1. <https://docs.snowflake.com/developer-guide/udf/python/udf-python-introduction>
   2. <https://docs.snowflake.com/developer-guide/stored-procedure/python/procedure-python-overview>
   3. By following the instructions, I can see that it basically create a small zip file containing the logic of your function/package and upload it into Snowflake storage so you can refer to it through SQL or SnowCLI.
4. Two python tips:
   1. # TODO – help you track places you didn’t finish or need refinement.
   2. “\_ =” - help you catch commands with results but you don’t need that result.
5. Snowflake Task
   1. <https://docs.snowflake.com/en/sql-reference/sql/create-task>
   2. <https://docs.snowflake.com/en/user-guide/tasks-intro>