



SNOWFLAKE

Hands-on for AI Workspace

Senior Solution Engineer

Sep 2025

Agenda

01. data catalog

- Git 연계
- 데이터 적재
- Horizon catalog 에서 asset 검색 및 관리

02. data pipeline

- Dynamic table 을 활용한 데이터 파이프라인 구성
- 수집, 처리, 적재 자동화 및 작업 스케줄링

03. data endpoint API

- REST API 로 snowflake 에 request 해서 데이터 얻기
- Application 구성

04. rbac

- Row access policy 를 활용한 역할기반 접근제어 구성
- 펀드 별 테이블 차단 정책 구현

05. python notebook

- Git 연계
- 노트북 생성 및 실행
- Snowpark Dataframe vs. Snowpark Pandas Dataframe vs. Native Pandas

06. cortex AI

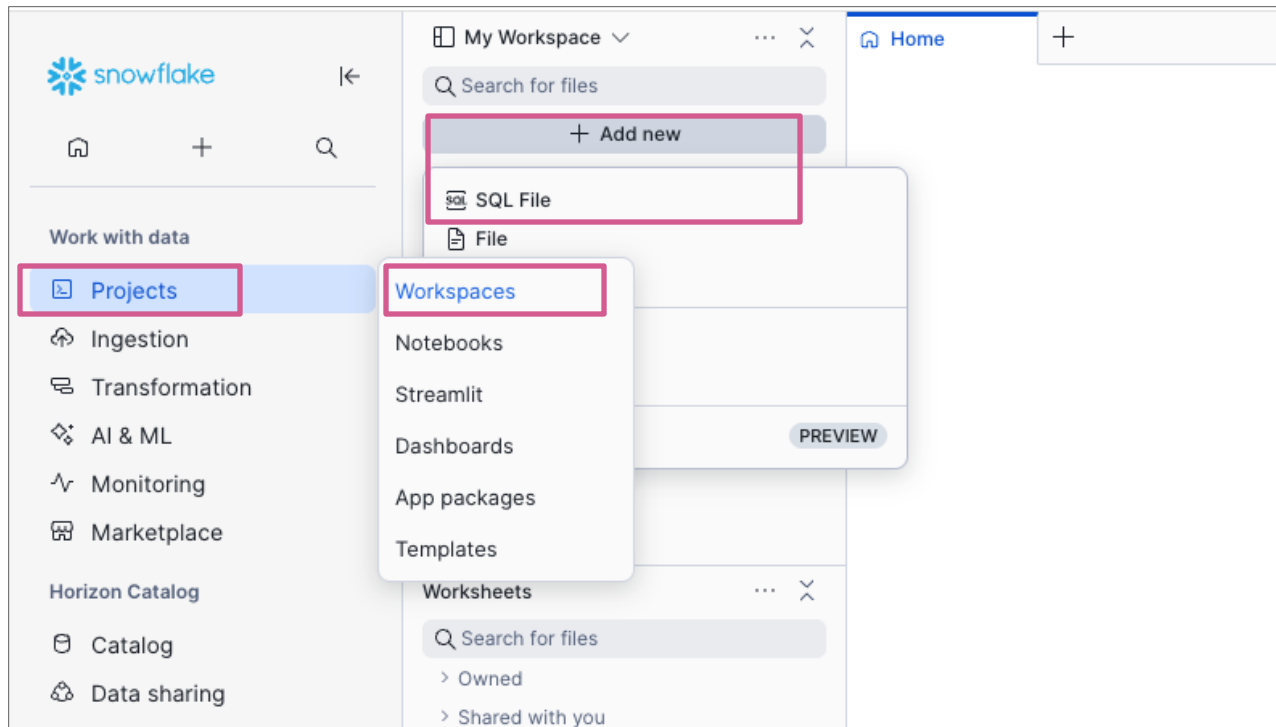
- Snowflake Intelligence 구성 (Text to SQL)
- Snowflake 에서 호출 가능한 커스텀 API 연동



01. data catalog



Workspace 에디터



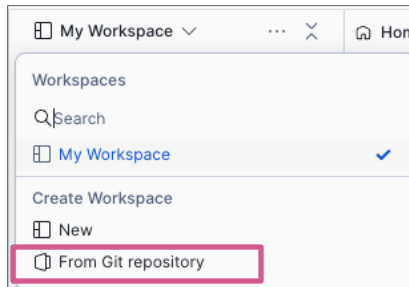
Git 연계

```
create database demo;  
create schema magi_handson;  
use warehouse compute_wh;
```

```
// 자신의 git access 정보로 대체  
CREATE OR REPLACE SECRET git_api_secret  
TYPE = password  
USERNAME = 'user_name'  
PASSWORD = 'xxxxxxxxxxxxxxxxx';
```

```
// 자신의 git 정보로 대체  
CREATE OR REPLACE API INTEGRATION git_api_integration  
API_PROVIDER = git_https_api  
API_ALLOWED_PREFIXES = ('https://github.com/xxxxxxxxxxx')  
ALLOWED_AUTHENTICATION_SECRETS = (git_api_secret)  
ENABLED = TRUE;
```

```
// 확인  
desc INTEGRATION git_api_integration;
```



Create workspace from Git repository

Repository URL ⓘ

Workspace name ⓘ

API integration ⓘ

GIT_API_INTEGRATION

Learn more

Personal access token ✓
Authenticate by selecting personal access token

Public repository
Authentication is not required

Credentials secret ⓘ

DEMO MAGI_HANDSON

GIT_API_SECRET

+ Secret

Cancel

Create



Data loading

The screenshot displays the Snowflake Database Explorer interface. On the left, the 'Database Explorer' sidebar shows the 'HORIZON CATALOG' and a tree view with 'DEMO' expanded, containing 'INFORMATION_SCHEMA' and 'MAGI_HANDSON' (selected). The main panel shows 'DEMO / MAGI_HANDSON' with tabs for 'Schema Details', 'Tables', 'Stages', and 'File Formats'. A 'Create' button is in the top right. A 'Load Data into Table' dialog box is open, showing 'SF_KOSCOM_ETFMST.csv - 10.2KB' as the source file. The 'Table' dropdown menu is open, with 'From File' selected. The dialog also shows the schema 'DEMO.MAGI_HANDSON' and a 'Create new table' option. The 'Next' button is highlighted in the dialog's footer.

Database Explorer
HORIZON CATALOG

Databases

Search

DEMO

INFORMATION_SCHEMA

MAGI_HANDSON

Tables

DEMO / MAGI_HANDSON

Schema ACCOUNTADMIN 46 minutes ago

Schema Details Tables Stages File Formats

Load Data into Table

COMPUTE_WH

SF_KOSCOM_ETFMST.csv - 10.2KB

Browse

Select or create a database and schema

Schema DEMO.MAGI_HANDSON + Database

Select or create a table

+ Create new table

Name

Cancel Back Next

From File

Standard

As Select

External

Table

Dynamic Table

View

Semantic View

Stage

File Format

Image Repository

Create



Horizon Catalog

The screenshot displays the Snowflake Horizon Catalog interface. On the left is a sidebar with navigation options: 'Work with data' (Projects, Ingestion, Transformation, AI & ML, Monitoring, Marketplace), 'Horizon Catalog' (Catalog, Data sharing, Governance & security), and 'Manage' (Compute, Admin). The 'Catalog' option is selected.

The main area is titled 'Database Explorer' and 'HORIZON CATALOG'. It shows a search for 'ETF' and a tree view of the database structure. The selected database is 'DEMO / MAGI_HANDSON / SF_KOSCOM ETFMST'. The table 'SF_KOSCOM ETFMST' is highlighted.

The 'Table Details' tab is active, showing the table's metadata: 'Table', 'ACCOUNTADMIN', '5 days ago', '71', and '8.0KB'. The 'Table Details' tab is highlighted with a red box.

The 'Description' section is visible, showing the table's description. A 'Describe Table' button is highlighted with a red box.

The 'Columns' tab is active, showing a list of 12 columns. A 'Generate descriptions' button is highlighted with a red box.

NAME ↑	TYPE	DESCRIPTION ⓘ	TAGS ⓘ
BASE_IDX_NM	Varchar	The name of the base index.	
BASE_MKT_L_CLS	Number	The base market class.	
BASE_MKT_M_C...	Number	The market class of the base market.	
BASE_MKT_S_CLS	Number	The base market share class.	

Horizon Catalog

The screenshot displays the Snowflake Horizon Catalog Internal Marketplace. On the left is a sidebar with navigation options: 'Work with data' (Projects, Ingestion, Transformation, AI & ML, Monitoring, Marketplace), 'Horizon Catalog' (Catalog, Data sharing, Governance & security), and 'Manage' (Compute, Admin). The 'Catalog' option is selected, and a dropdown menu shows 'Database Explorer', 'Internal Marketplace' (highlighted), and 'Apps'. The main content area is titled 'Internal Marketplace' and 'HORIZON CATALOG'. A large blue banner reads 'Trusted resources shared by teams within your organization' with a subtext 'Browse the curated set of internal listings recommended by your organization' and a 'View Documentation' link. Below the banner is a search bar 'Search data products and profiles'. A filter bar includes 'Geographic coverage All', 'Profile All', 'Time coverage All', and '19 Data Products'. A 'Popularity' sort button and a '+ Create listing' button are also present. The main grid shows data products: 'Customer Analysis Data' (2024), 'Tasty Bytes-test' (Quickstart base data), 'Sample Amperity Data' (audience info), and 'SALES_DATA'. Each product has an 'INTERNAL' label and a 'By request' icon. A 'Product Categories' section on the right shows '財務データ' (Financial Data) with a description 'これはXXです。'.

Internal Marketplace
HORIZON CATALOG

Trusted resources shared by teams within your organization

Browse the curated set of internal listings recommended by your organization

[View Documentation >](#)

Search data products and profiles

Database Explorer
Internal Marketplace
Apps

Geographic coverage All | Profile All | Time coverage All | 19 Data Products

↑↓ Popularity + Create listing

Customer Analysis Data
Customer Analysis Data in 2024.
INTERNAL By request

Tasty Bytes-test
Tasty Bytes Quickstartの基礎データ
INTERNAL By request

Product Categories
財務データ
これはXXです。

Sample Amperity Data
Sample dataset for audience information from Amperity

SALES_DATA
説明



Lab 01

- 데이터 로딩
- Gen AI 로 테이블/컬럼 description 생성



02. data pipeline



Task

- 주기적으로 단일 SQL 문 실행 혹은 Stored Procedure 호출하는 객체
- DAG 구조로 여러 tasks 를 연속적으로 실행 가능

create task run_task

<warehouse>

<schedule>

<after>

as

```
insert into tar_table  
select key, plt_cd, ...  
from stream_a;
```

or

```
call root_run('pivot_key');
```

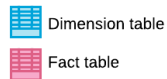
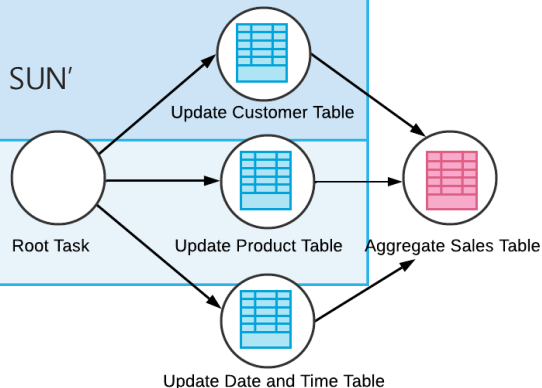
WAREHOUSE = HT_SMALL
: 지정한 고객 Warehouse를 기반으로 동작

USER_TASK_MANAGED_INITIAL_WAREHOUSE_SIZE = SMALL
: Serverless 방식으로 수행

SCHEDULE = '5 minute'
: 반복적 수행

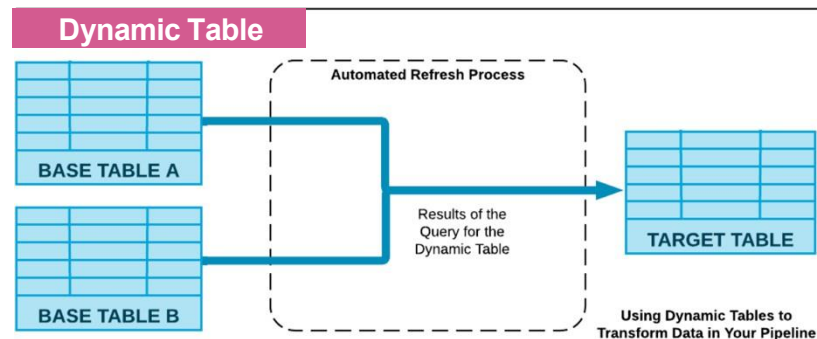
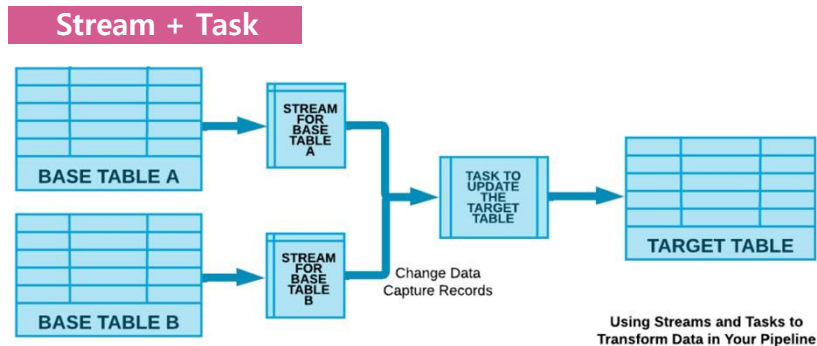
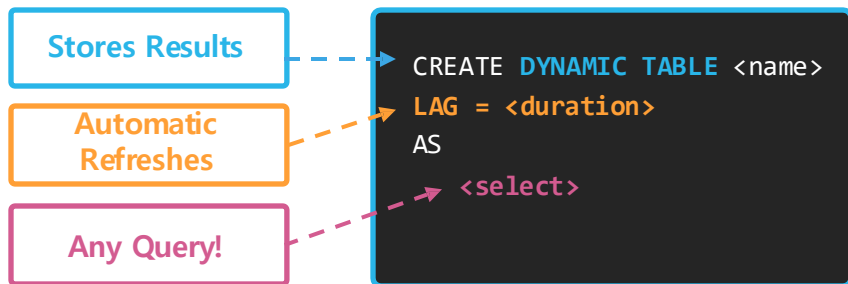
SCHEDULE = 'USING CRON 0 9-17 * * SUN'
: 시스템 시간 기반 동작

AFTER pre_task
: 각 Task 간의 선/후행 관계 정의



Dynamic Table

- 선언적 데이터 파이프라인
- 최종 타겟 테이블을 정의함으로써 복잡한 데이터 파이프라인을 관리할 필요가 없음
- 쿼리 결과 자체를 materialized 해서 저장
- Stream 과 Task 대체 가능
- 파이프라인 자동화를 위해 사용
- DAG 구성 가능. Lag 존재



Dynamic Table - Graph

The screenshot displays the Snowflake Database Explorer interface. On the left, a sidebar contains navigation options like 'Work with data', 'Horizon Catalog', and 'Manage'. The main area is titled 'Database Explorer' and shows a tree view of the database structure. The 'MARKET_INFO' table is highlighted. The right pane shows the 'Graph' view of the 'MARKET_INFO' table, which is a dynamic table. It displays a data flow graph where data from two source tables, '.ETF_JITRAD...' and '.ETFMST...', is processed by a 'TRADE_DAILY' table to produce the 'MARKET_INFO' table. The 'MARKET_INFO' table is shown with a 'Target lag 1m' and a 'Succeeded' status. Below the graph, a 'Lag Metrics' section provides details on the table's refresh status, including 'Last Refresh', 'Time Wit...', 'Current Lag', 'Target Lag', and 'Max Lag'. A 'Configuration' section at the bottom shows the table's state as 'Active', its refresh mode as 'Incremental', and its object type as 'Dynamic Table'.

Database Explorer
HORIZON CATALOG

Databases

Search

DEMO

- INFORMATION_SCHEMA
- MAGI_HANDSON
 - Tables
 - Dynamic Tables
 - ASSET_INFO
 - MARKET_INFO**
 - TRADE_DAILY
 - Stages
 - File Formats
- PUBLIC
- SNOWFLAKE
- SNOWFLAKE_LEARNING_DB
- SNOWFLAKE_SAMPLE_DATA
- USER\$MJLEE

DEMO / MAGI_HANDSON / MARKET_INFO

Dynamic Table ACCOUNTADMIN 1 minute ago 87.4K COMPUTE_WH Incremental

Table Details Columns Data Preview **Graph** Refresh History Data Quality PREVIEW

+ - [] [] [] | COMPUTE_WH

MARKET_INFO
DEMO.MAGI_HANDSON

Details Definition

Lag Metrics


Last Refresh	Succeeded
Time Wit...	100%
Current Lag	41s
Target Lag	1m
Max Lag	50s

Configuration

State	Active
Refresh Mode	Incremental
Object Type	Dynamic Table



Dynamic Table - Refresh

 Database Explorer
HORIZON CATALOG

Databases

Work with data
Projects
Ingestion
Transformation
AI & ML
Monitoring
Marketplace
Horizon Catalog
Catalog
Data sharing
Governance & security
Manage
Compute
Admin

Search
DEMO
INFORMATION_SCHEMA
MAGI_HANDSON
Tables
Dynamic Tables
ASSET_INFO
MARKET_INFO
TRADE_DAILY
Stages
File Formats
PUBLIC
SNOWFLAKE
SNOWFLAKE_LEARNING_DB
SNOWFLAKE_SAMPLE_DATA

DEMO / MAGI_HANDSON / MARKET_INFO


Dynamic Table ACCOUNTADMIN 3 minutes ago 87.4K COMPUTE_WH Incremental

Table Details Columns Data Preview Graph Refresh History Data Quality PREVIEW

Lag Metrics
100% Time Within Target Lag
1m Target Lag
5s Current Lag
50s Maximum Lag
COMPUTE_WH

Last day Refresh Status All 5 Refreshes Warehouse used only

SOURCE DATA TIMESTAMP	REFRESH STAT...	REFRESH DURATI...	REFRESH LAG	ROWS CHANGED
Sep 26, 2025, 1:50:49 PM	Succeeded	720ms	51s	+2 -0
Sep 26, 2025, 1:47:02 PM	Succeeded	1.5s	—	+87.4K -0

 Database Explorer
HORIZON CATALOG

Databases

Work with data
Projects
Ingestion
Transformation
AI & ML
Monitoring
Marketplace
Horizon Catalog
Catalog
Data sharing
Governance & security
Manage
Compute
Admin

Search
DEMO
INFORMATION_SCHEMA
MAGI_HANDSON
Tables
Dynamic Tables
ASSET_INFO
MARKET_INFO
TRADE_DAILY
Stages
File Formats
PUBLIC
SNOWFLAKE
SNOWFLAKE_LEARNING_DB
SNOWFLAKE_SAMPLE_DATA

DEMO / MAGI_HANDSON / TRADE_DAILY

Dynamic Table ACCOUNTADMIN 4 minutes ago 87.4K COMPUTE_WH Incremental

Table Details Columns Data Preview Graph Refresh History Data Quality PREVIEW

Lag Metrics
100% Time Within Target Lag
1m Target Lag
37s Current Lag
50s Maximum Lag
COMPUTE_WH

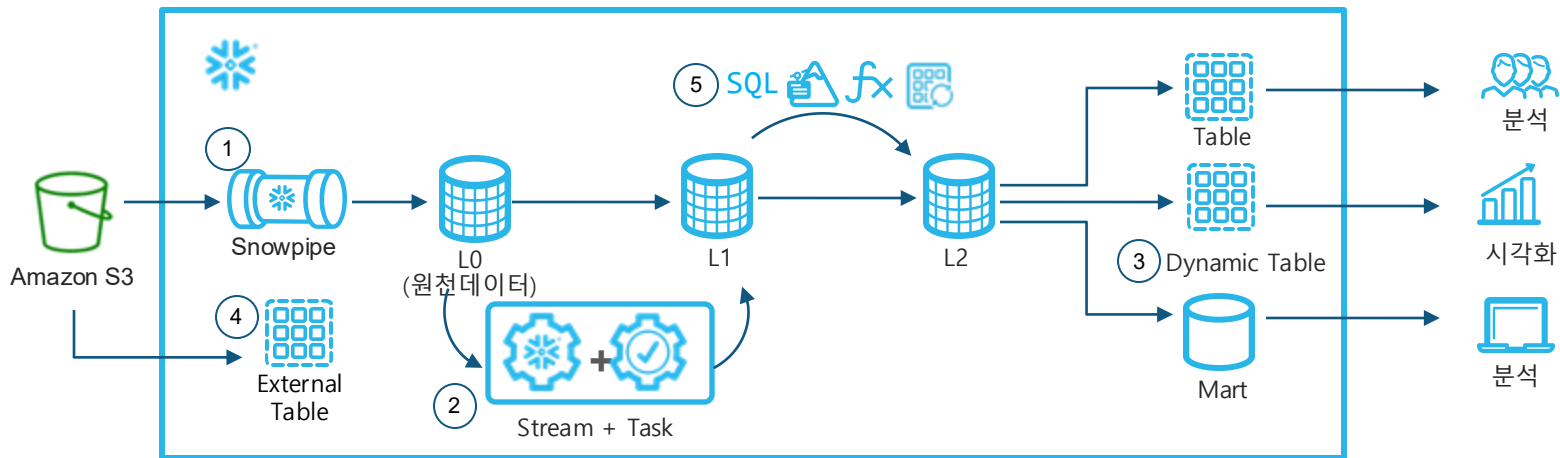
Last day Refresh Status All 5 Refreshes Warehouse used only

SOURCE DATA TIMESTAMP	REFRESH STAT...	REFRESH DURATI...	REFRESH LAG	ROWS CHANGED
Sep 26, 2025, 1:50:49 PM	Succeeded	858ms	50s	+2 -0
Target lag was altered. Lag metrics have been reset.				
Sep 26, 2025, 1:47:00 PM	Succeeded	1.2s	—	+87.4K -0
Sep 26, 2025, 1:45:02 PM	Succeeded	1.8s	—	+87.4K -0



데이터 파이프라인

- 모든 데이터가 한곳에 위치하여 원하는 데이터를 빠르게 찾아 분석에 활용 / 운영 간소화
- SQL / Snowpark / Sproc / UDF 등을 활용한 쉬운 데이터 변환 파이프라인 (+자동화)



1.Snowpipe: 증분 데이터를 micro-batch 방식으로 스테이징 테이블에 적재

2.Stream + Task : Stream이 설정된 테이블에 대해 삽입, 업데이트, 삭제를 포함한 DML 오퍼레이션을 자동으로 감지 하여 CDC 기능 제공.
테이블 변환 정보를 적용하는 SQL을 Task 기능으로 스케줄링하여 CDC 자동화 가능.

3.Dynamic Table : Source 테이블의 변경 사항을 자동으로 Target 테이블에 동기화

4.External Table : Snowflake 적재 없이 S3 에 위치한 파일 read

Task - Graph

The screenshot displays the Snowflake Task Graph interface. On the left, a task graph shows a sequence of tasks: DEMO_TASK_1, DEMO_FINALIZER, and DEMO_TASK_2 through DEMO_TASK_13. Below the graph, a table lists 16 tasks with their names and statuses.

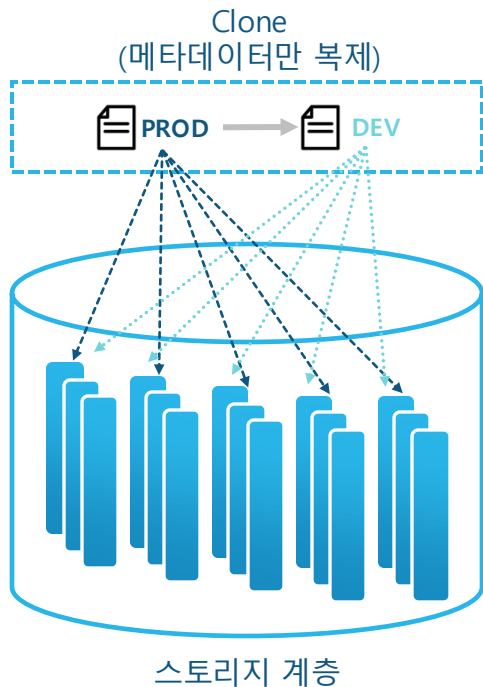
TASK NAME	STATUS
DEMO_TASK_2	Succeeded
DEMO_TASK_4	Succeeded
DEMO_TASK_6	Succeeded
DEMO_TASK_11	Skipped
DEMO_TASK_12	Succeeded
DEMO_TASK_7	Succeeded
DEMO_TASK_8	Skipped
DEMO_TASK_5	Succeeded
DEMO_TASK_13	Succeeded
DEMO_TASK_9	Failed

A user account selection menu is open, showing the current user 'MJ LEE' and a list of accounts. The account 'KR_DEMO28' is selected. The menu also includes options for Settings, Support, Appearance, Connect a tool to Snowflake, Client download, Documentation, Privacy notice, Session details, and Sign Out.

The background shows a task graph with nodes like DEMO_TASK_1, DEMO_FINALIZER, and DEMO_TASK_2 through DEMO_TASK_13. The status of DEMO_TASK_9 is 'Failed'.

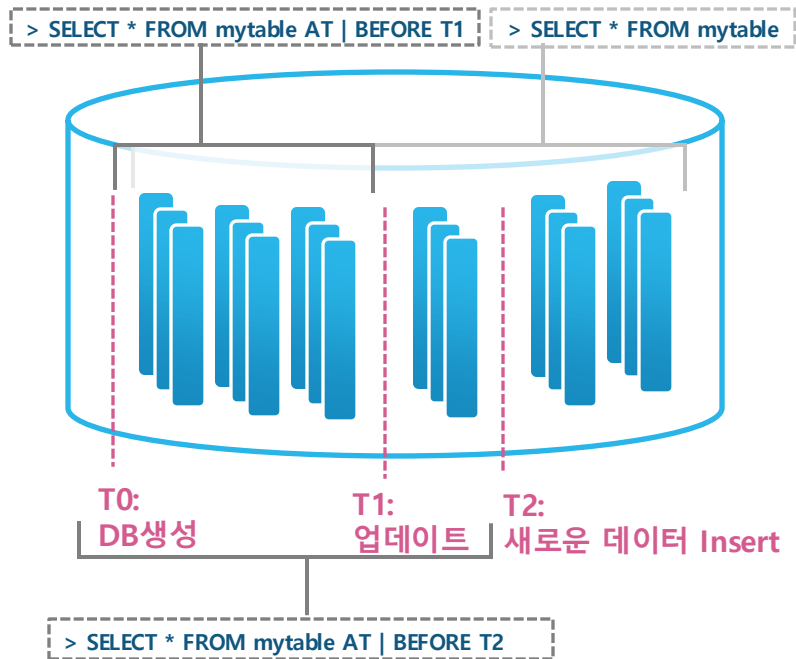


ZERO-COPY CLONING



- 메타데이터 포인터 정보만 복사함으로써 물리적인 데이터 복사 없이 복제
- 순식간에 테스트 및 개발 환경 구축
`create table sales_backup`
`clone sales;`
- 메타데이터 기술을 활용한 무복사 복제 기능으로 추가적인 스토리지 불필요: 스토리지 비용 절감
- Clone 이후 데이터 변경분에 대해서만 과금

TIME TRAVEL



기능

- MVCC (Multi-version Concurrency Control) 방식
- 데이터 변경 내역이 snapshot
- 이전 버전의 데이터 자동 관리

사용 사례

- 사용자 실수나 삭제된 데이터 복원
- 과거 특정 시점 데이터 복제 및 백업
- 특정 기간 내 데이터 사용 및 변경 내용 분석
- 테이블, 스키마, 데이터베이스에 대한 UNDROP 명령어 지원



Lab02

- 파이프라인 구성
- 데이터 변경 후 업데이트 확인 (@Snowsight)
- 테이블 원복



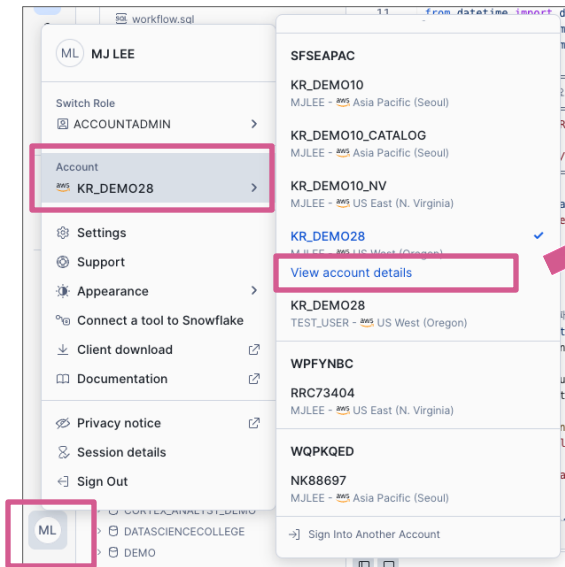
03. data endpoint API



Lab03

- API 실습 (+application 실행)

[Account 정보 확인 방법]



Account Details		
Account	Config File	Connectors/Drivers
SQL Commands		
NAME	VALUE	
Account identifier ⓘ	SFSEAPAC-KR_DEMO28	📄
Data sharing account identifier ⓘ	SFSEAPAC.KR_DEMO28	📄
Organization name	SFSEAPAC	📄
Account name	KR_DEMO28	📄

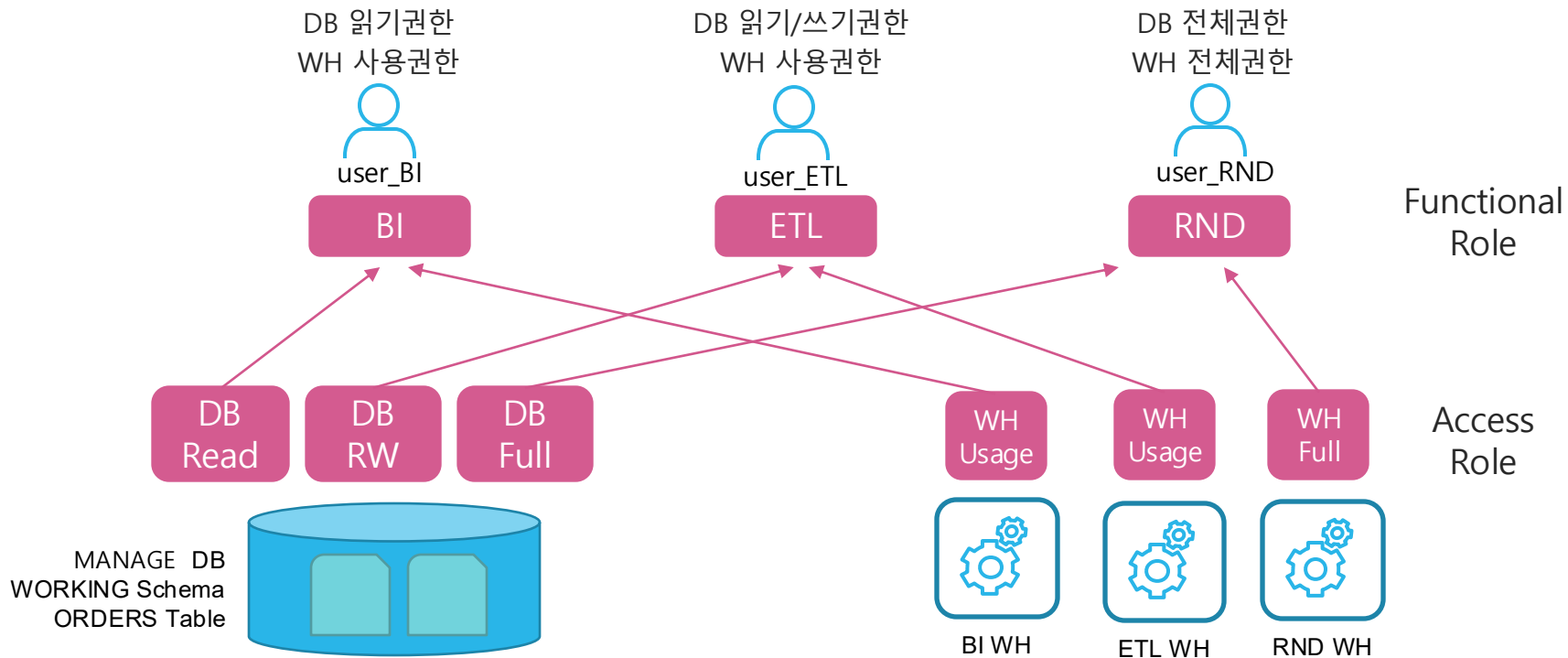


04. rbac



객체에 대한 권한 관리 및 제어

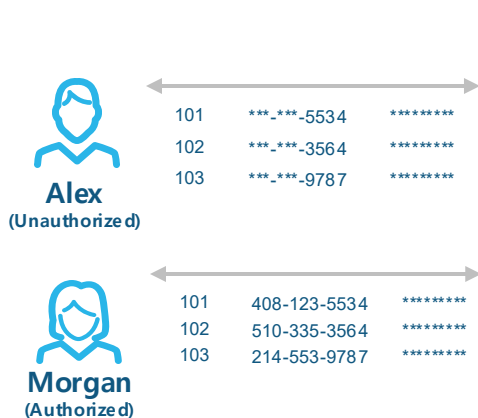
Role-based Access Control



권한없는 사용자의 데이터 획득 방지

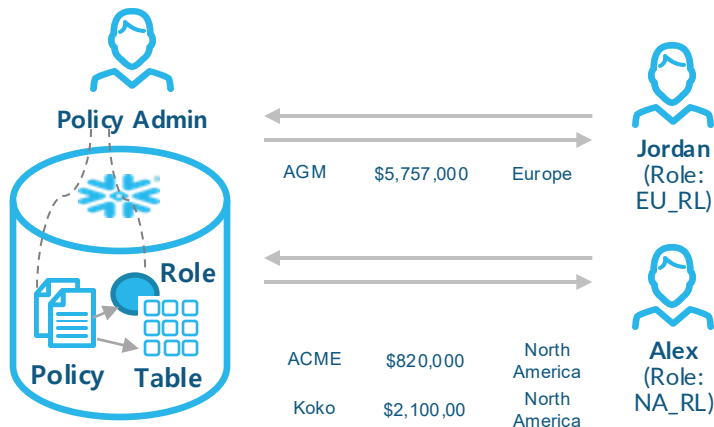
컬럼데이터 마스킹

- 저장된 데이터에 대한 변경없이 상수, 해쉬, UDF 를 통해 특정 값으로 마스킹 적용
- 권한 기반으로 인증된 사용자만 컬럼 값 확인 가능
- 단일 정책 생성 후 복수개의 컬럼에 적용 가능
- 정책은 Snowflake 중앙에서 저장 및 관리



Row 데이터 접근 제어

- 테이블, 뷰, 데이터 공유 및 외부 테이블에 대하여 정책 적용 가능
- 데이터를 하나의 테이블로 통합 후 사용자 권한에 따라 동적으로 접근 제어



Lab04

- RBAC 실습

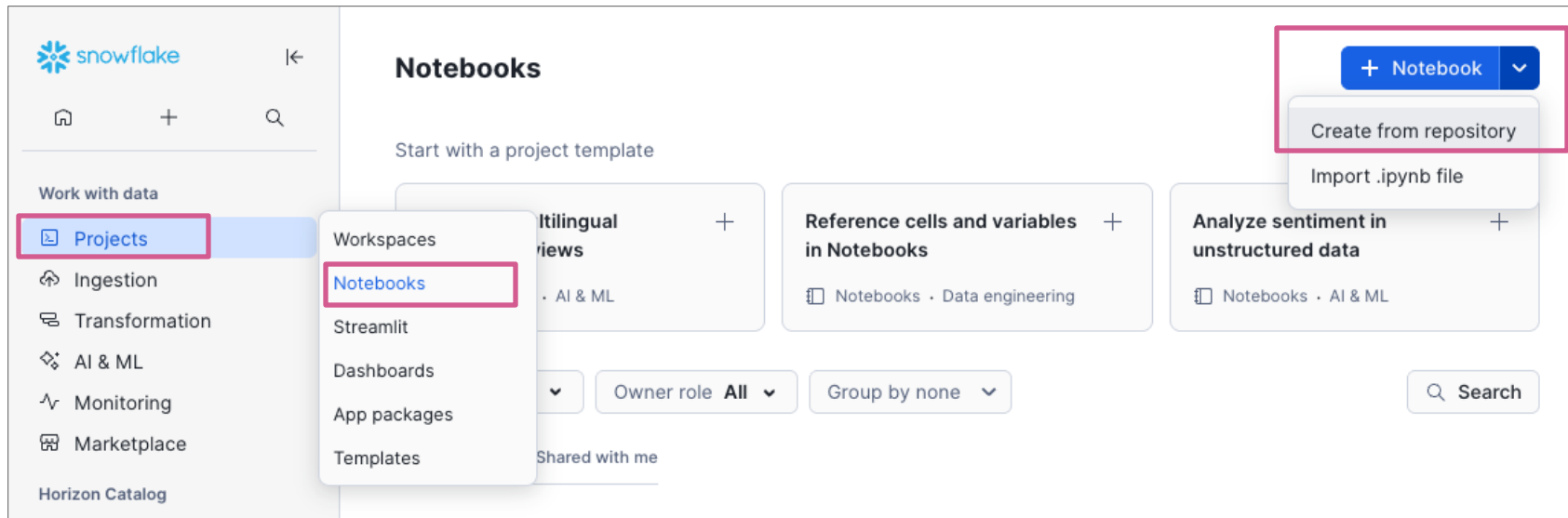
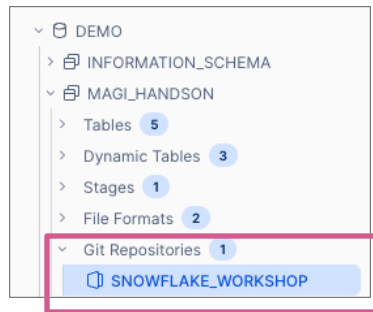


05. python notebook



Git Repository 생성

```
// 01. data loading 에서 설정한 git integration 사용
CREATE OR REPLACE GIT REPOSITORY snowflake_workshop
API_INTEGRATION = git_api_integration
GIT_CREDENTIALS = git_api_secret
ORIGIN = 'https://github.com/puremjlee/snowflake-workshop.git';
```



Create notebook from repository PREVIEW

Owner: ACCOUNTADMIN

Name

notebook workshop

File location in repository ⓘ

...AKE_WORKSHOP/branches/main/05.python_notebook/etf_analysis.ipynb

Notebook location ⓘ

DEMO

MAGI_HANDSON

Runtime ⓘ

Run on warehouse ✓

Best for **data analysis**. Has access to thousands of packages from the Snowflake Anaconda channel.

Run on container

Best for **AI/ML workloads**. Has access to CPUs/GPUs and optimized APIs to scale AI/ML.

Runtime version ⓘ

Snowflake Warehouse Runtime 2.0

Query warehouse ⓘ

COMPUTE_WH

Notebook warehouse ⓘ

SYSTEM\$STREAMLIT_NOTEBOOK_WH

Cancel

Create

Notebook 생성

Select Notebook File In Repository PREVIEW

Database DEMO

Create Git repository

Search

SNOWFLAKE_WORKSHOP

Select Branch main

Fetch

1 Branch

SNOWFLAKE_WORKSHOP / 05.python_notebook

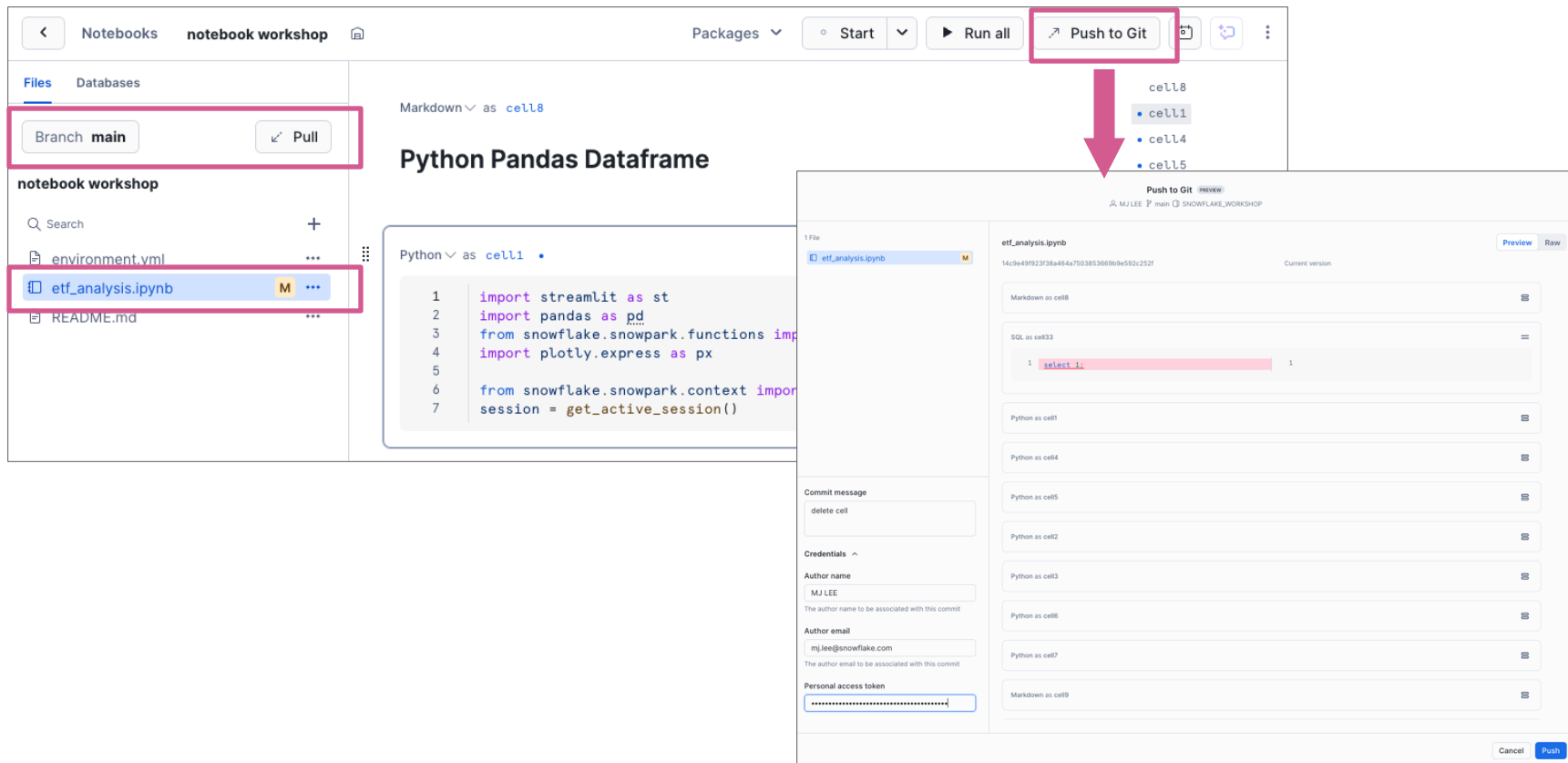
Search

NAME	SIZE	LAST MODIFI...
README.md	20.0B	15 minutes ago
environment.yml	98.0B	15 minutes ago
etf_analysis.ipynb	14.1KB	15 minutes ago

Cancel

Select file

Git Push / Pull



The screenshot illustrates the process of pushing a notebook to Git in the Snowflake Notebook Workshop. The interface shows a sidebar with a file list, a main editor area, and a 'Push to Git' modal window.

Push to Git Modal:

- Files:** A list of files to be pushed, including `etf_analysis.ipynb`.
- Commit message:** A text input field for the commit message.
- Credentials:** Fields for `Author name` (MJ LEE), `Author email` (mj.lee@snowflake.com), and `Personal access token`.
- Push to Git:** A button to initiate the push process.

etf_analysis.ipynb Content:

```
1 import streamlit as st
2 import pandas as pd
3 from snowflake.snowpark.functions import
4 import plotly.express as px
5
6 from snowflake.snowpark.context import
7 session = get_active_session()
```



Schedule

```
EXECUTE NOTEBOOK MY_DB.PUBLIC.MY_NOTEBOOK();
```

```
create task run_notebook_task
WAREHOUSE = mywh
SCHEDULE = 'USING CRON 0 9-17 * * SUN Asia/Seoul'
as
EXECUTE NOTEBOOK MY_DB.PUBLIC.MY_NOTEBOOK();
```

The screenshot shows a web interface for scheduling a notebook run. At the top, there are buttons for 'Start', 'Run all', and 'Push to Git'. A modal dialog titled 'Schedule a notebook run' is open. It contains the following fields and options:

- Schedule name:** A text input field containing 'hourly run'.
- Frequency:** A dropdown menu set to 'Hourly', followed by 'at' and a time input '00' with a 'min' label, and a location dropdown set to '(UTC+09:00) (Local time) Seoul'.
- Parameter (optional):** A text input field containing 'database_name'.
- Summary:** A blue box with an information icon, text 'Every hour, Asia/Seoul', and 'Next run: Sep 26, 5:00 PM, Asia/Seoul'.
- Buttons:** 'Cancel' and 'Create' at the bottom right.

In the background, a notebook editor is visible with a sidebar showing a list of cells (cell12 to cell18) and a menu with options: 'Create schedule', 'View run history', and 'View schedules'.



Python 기반 데이터 엔지니어링

데이터프레임

Pandas Dataframe

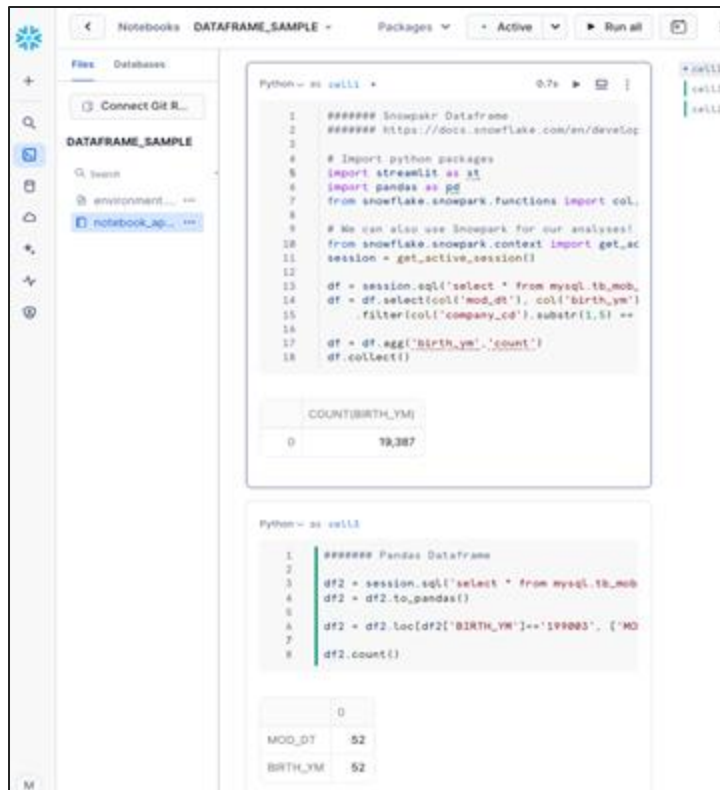
- Python 에서 지원하는 2차원 데이터 구조
- 클라이언트 머신 Local 리소스를 사용
- 연산할 때마다 결과가 메모리에 materialize 됨

Snowpark Dataframe

- Snowflake 의 데이터 처리 프레임에서 지원하는 2차원 데이터 구조
- Spark 과 유사한 Dataframe API
- Snowflake 에서 SQL 로 변환되어 실행되며 확장 리소스 지원
- Lazy evaluation
- 원본 데이터 기반으로 작업 수행하며, 분산처리 효율성을 위해 데이터 순서 보장하지 않음

Snowpark Pandas Dataframe

- Snowflake 의 데이터 처리 프레임에서 지원하는 2차원 데이터 구조
- Pandas 와 유사한 Dataframe
- Snowflake 에서 SQL 로 변환되어 실행되며 확장 리소스 지원
- Lazy evaluation
- 원본 데이터를 clone 하여 스냅샷 기반으로 작업 수행하며, 데이터 순서 보장



Lab05

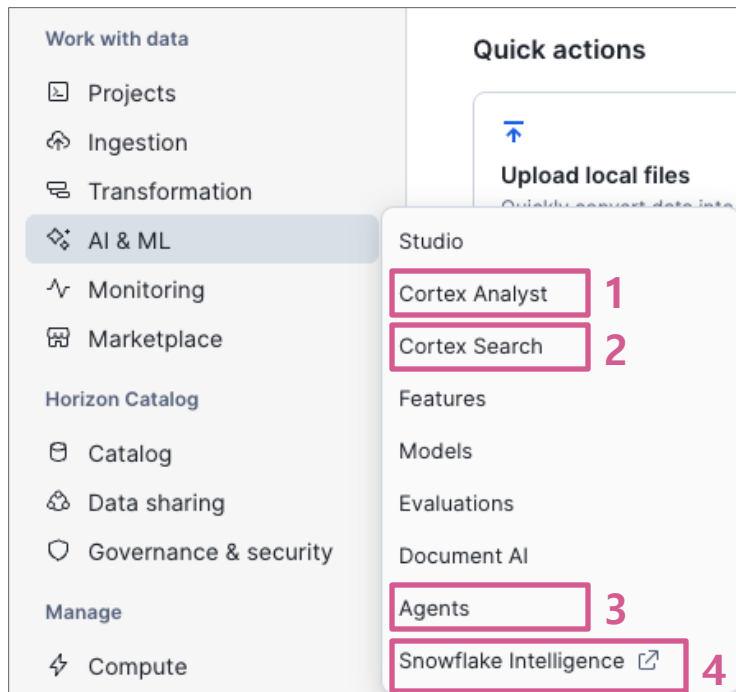
- Notebook 실행
- 성능 비교



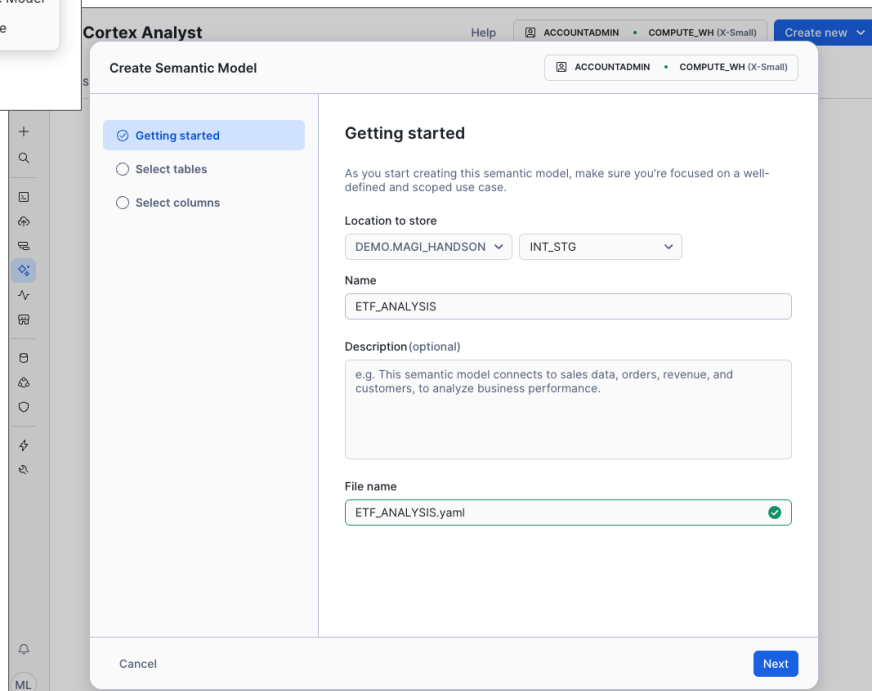
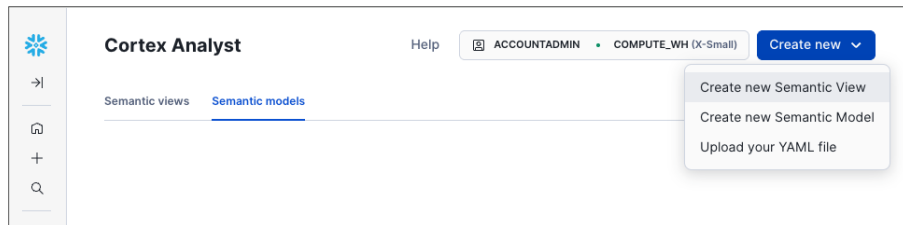
06. cortex AI



Snowflake Intelligence



Cortex Analyst



Cortex Analyst

Create Semantic Model

ACCOUNTADMIN • COMPUTE_WH (X-Small)

Getting started

Select tables

Select columns

Select tables

Select only the tables required to answer the business questions your users want to ask.

All

Selected 2

Search

DEMO

INFORMATION_SCHEMA

MAGI_HANDSON

7 Tables

3 Dynamic Tables

ASSET_INFO

MARKET_INFO

TRADE_DAILY

PUBLIC

SNOWFLAKE

Cancel

Previous

Next

Create Semantic Model

ACCOUNTADMIN • COMPUTE_WH (X-Small)

Getting started

Select tables

Select columns

Select columns

Select only the columns required to answer the business questions your users want to ask.

Columns (19 selected)

Expand all

Search columns

> ☒ DEMO.MAGI_HANDSON.ASSET_INFO

> ☒ DEMO.MAGI_HANDSON.MARKET_INFO

☒ Add sample values to the semantic model

A few real values from each column helps Cortex Analyst give better answers. [Learn more](#)

☒ Add descriptions to the semantic model

We'll use AI to figure out sensible descriptions to add in, saving you time.

Cancel

Previous

Create and Save

© 2025 Snowflake Inc. All Rights Reserved

37

Cortex Analyst

INT_STG > ETF_ANALYSIS.yaml Edit YAML

☰

ETF_ANALYSIS

Edit

Custom Instructions ⓘ

Edit

Logical tables ⓘ 2

+

Edit logical table

Base tables

DEMO.MAGI_HANDSON.ASSET_INFO

Generate fields

Logical table name

ASSET_INFO

Logical table description (optional)

This table stores daily asset information for Exchange-Traded Funds (ETFs) in the Korean market, including the base date, ETF item code, item name in Korean, base index name, assets under management (AUM), various settlement and trading quantities, net asset value (NAV), and the name of the management company in Korean.

Cortex Search – Parsing & Chunking

Trial 버전에서는 실행 불가

```
CREATE OR REPLACE NETWORK RULE pypi_network_rule
MODE = EGRESS
TYPE = HOST_PORT
VALUE_LIST = ('pypi.org', 'pypi.python.org',
'pythonhosted.org', 'files.pythonhosted.org');
```

```
CREATE OR REPLACE EXTERNAL ACCESS INTEGRATION
pypi_access_integration
ALLOWED_NETWORK_RULES = (pypi_network_rule)
ENABLED = true;
```

Notebook settings
Owner: ACCOUNTADMIN

General External access

Query warehouse ⓘ
COMPUTE_WH

Runtime ⓘ

Run on warehouse
Best for **data analysis**. Has access to thousands of packages from the Snowflake Anaconda channel.

Run on container ✓
Best for **AI/ML workloads**. Has access to CPUs/GPUs and optimized APIs to scale AI/ML.

Runtime version
Snowflake ML Runtime CPU 1.0
Determines the starting packages that are available to you. [Learn more](#)

Compute pool
SYSTEM_COMPUTE_POOL_CPU
Notebook will run in the default compute pool. [Learn more](#)

Idle timeout ⓘ
1 hour

Cancel Save

+ Notebook

Create from repository

Import .ipynb file



Cortex Search – Parsing & Chunking

Trial 버전에서는 실행 불가

Notebook settings

Owner: ACCOUNTADMIN

General **External access**

External networks allow you to access the internet at certain endpoints. If you don't see a network here, please contact your admin to get access.

NOTION_INTEGRATION

☐

NOTION_INTEGRATION_MAGI

☐

GOOGLE_APIS_ACCESS_INTEGRATION

☐

SNOWFLAKE_OAUTH_SPCS_INTEGRATION

☐

CONNECTION_EAI_OMNATA_SYNC_ENGINE_GOOGLE_SPREADSHEET_A

☐

GSHEET_APIS_ACCESS_INTEGRATION

☐

HF_EGRESS_EAI

☐

DATAHUB_SPCS_EGRESS_ACCESS_INTEGRATION

☐

DATAHUB_EXTERNAL_ACCESS

☐

API_ACCESS_INTEGRATION

☐

PYPI_ACCESS_INTEGRATION

☒

Cancel

Save

Python as Install_pymupdf

1

!pip install PyMuPDF

Kernel restart may be needed

If you're updating an existing package, you may need to restart the kernel.

Show me how

Looking in links: /opt/pip_cache
Collecting PyMuPDF
 Downloading pymupdf-1.26.4-cp39-abi3-manylinux_2_28_x86_64.whl.metadata (3.4 kB)
 Downloading pymupdf-1.26.4-cp39-abi3-manylinux_2_28_x86_64.whl (24.1 MB)
 [2K [90m-----[0m [32m24.1/24.1 MB [0m [31m129.7 MB/s [0m
 [?25hInstalling collected packages: PyMuPDF

Successfully installed PyMuPDF-1.26.4
[33mWARNING: Running pip as the 'root' user can result in broken permissions and conflicting bel
[0m

Cortex Search

Create Search Service

New service


Select data

Select search column

Select attributes

Select columns

Configure indexing



Let's create a new Cortex Search Service

We'll guide you through the steps of selecting a data source, setting service parameters, and creating the Search Service.

Select a warehouse to power the service. This warehouse will be used for materializing the results of the source query upon creation and refresh.

Role and Warehouse

ACCOUNTADMIN + COMPUTE_WH (X-Small)

Database and Schema

DEMO.MAGI_HANDSON

Name

docs_search

Cancel

Next: Select data

Create Search Service

New service

Select data

Select search column

Select attributes

Select columns

Configure indexing

Select data to be indexed

Select a table or view containing containing text data that you'd like to search over. At least one column in this table or view must be a text column that you'd like to search over.

If you wish to specify multiple data sources or perform transformations when defining your service, please use the [SQL surface](#).

DEMO.MAGI_HANDSON

Search Tables and Views

IMAGE_DATA_CHUNK

IMAGE_DATA_RAW

MCP_ROLE_MAPPING

SF_KOSCOM ETF_JITRADE_DAILY

SF_KOSCOM ETF_JONG_DAILY

Cancel

Previous: New service

Next: Select search column



Cortex Search

Create Search Service

☒ New service

☒ Select data

☒ Select search column

☐ Select attributes

☐ Select columns

☐ Configure indexing

Select a search column

The data in this column must be text-based and will be the column that your queries search over.

IMAGE_DATA_CHUNK

Search Column

RELATIVE_PATH

FILE_URL

CHUNK

Cancel

Previous: Select data

Next: Select attributes

Create Search Service

☒ New service

☒ Select data

☒ Select search column

☒ Select attributes

☐ Select columns

☐ Configure indexing

Optional

Select attribute column(s)

Select a set of columns that you'd wish to use as filters when querying the service.

IMAGE_DATA_CHUNK

Search Column

RELATIVE_PATH

FILE_URL

CHUNK_INDEX

CHUNK

Cancel

Previous: Select search column

Next: Select columns

Cortex Search

Create Search Service

☒ New service

☒ Select data

☒ Select search column

☒ Select attributes

☒ Select columns

☐ Configure indexing

Optional

Select columns to include in the service

Select the columns that you want to include in the search index. The data in these columns will be included in the search service and will be available for querying.

Search and attribute columns must be part of the included columns.

IMAGE_DATA_CHUNK

Q Search Column

△ RELATIVE_PATH ✓

△ FILE_URL ✓

CHUNK_INDEX

△ CHUNK ✓

⌚ CREATED_TIMESTAMP

Select all

Cancel

Previous: Select attributes

Next: Configure indexing

Create Search Service

☒ New service

☒ Select data

☒ Select search column

☒ Select attributes

☒ Select columns

☒ Configure indexing

Configure your Search Service

Select your desired configuration parameters for the service.

Target Lag

1

hours

Target Lag parameter specifies the maximum frequency with which the service will check for and materialize updates based on changes to its source data.

Embedding model (optional)

snowflake-arctic-embed-l-v2.0

Each model may incur a different cost per million input tokens processed. Refer to the [Snowflake Service Consumption Table](#). For more information about each embedding model, refer to the [Cortex Search documentation](#).

Warehouse for indexing

COMPUTE_WH

Indexing can take couple minutes to an hour depending on size of data. [See documentation](#).

Cancel

Previous: Select columns

Create Search Service

© 2025 Snowflake Inc. All Rights Reserved

43

Cortex Search

< Cortex Search DOCS_SEARCH PREVIEW

▶ Playground ⋮

[Search Service](#) [Data Preview](#) [Costs](#)

DEMO.MAGI_HANDSON

DOCS_SEARCH

Base table

COLUMN SEARCHABLE

CHUNK ✓

RELATIVE_PATH

FILE_URL

8 rows • [View SQL](#)

Manually refresh

Edit

Suspend indexing

Suspend serving

Drop

Cortex Search PREVIEW Playground Create

? Why use Cortex Search? ×
Build performant, high-quality search bars and RAG chatbots on unstructured data.
Tutorials

📖 Cortex Search Documentation ×
Learn more about creation, usage, and management of Cortex Search Services.
Documentation

DEMO ▼ 1 search service ↻

NAME	DATAB...	SCHE...	INDEXI...	SERVI...	ROWS	TARGE...	CR... ↓
DOCS_SEARCH	DEMO	MAGI_...	ACTIVE	ACTIVE	20	1 hour	p 28, 2025 ⋮



Cortex Agent

Agents PREVIEW

생성 후 Edit 버튼 클릭

Documentation [Create agent](#)

Search agents Database: All

Created By: Anyone

Create agent

Platform integration

☒ Create this agent for [Snowflake Intelligence](#)

Database and schema

SNOWFLAKE_INTELLIGENCE.AGENTS

Agent object name

ETF_analyst_agent

API URL is based on the agent's name. Changing the name later may break the link./databases/SNOWFLAKE_INTELLIGENCE/schemas/AGENTS/agents/ETF_analyst_agent

Display name

ETF 분석 에이전트

[Cancel](#) [Create agent](#)

좌측 Tools 선택 후 이전에 생성한 Cortex Analyst 와 Cortex Search 등록

< ETF_ANALYSIS_AGENT [Cancel](#) [Save](#)

About

Tools

Orchestration

Access

Tools

[Learn More](#)

Cortex Analyst [+ Add](#)

No Cortex Analyst models added yet

Cortex Search Services [+ Add](#)

No Cortex Search Services added yet

Custom tools [+ Add](#)

Add custom tools by defining a function your agent can call

New thread Show Traces

ETF_ANALYSIS_AGENT



Cortex Agent – Cortex Analyst / Search 등록

Add tool: Cortex Analyst

Cortex Analyst ☐ Semantic view ☒ Semantic model file

Schema DEMO.MAGI_HANDSON x v

INT_STG v

< Go Back Current Path: /

ETF_ANALYSIS.yaml

Selected file: ETF_ANALYSIS.yaml

Tool details

Name

etf_analysis_tool

Description

Will be used during orchestration to guide how the agent uses this tool

[Learn more](#)

[Generate with Cortex](#)

Cancel Add

Add tool: Cortex Search

Select a Cortex Search Service or Cortex Knowledge Extension

Cortex Search

Schema DEMO.MAGI_HANDSON x v

Q DEMO.MAGI_HANDSON.DOCS_SEARCH v

Max results

- 4 +

ID column

Used to generate hyperlinks to the source

FILE_URL v

Title column

RELATIVE_PATH v

Search results filters

Add up to 5 filters

+ Add filter

Tool details

Name

docs_search_tool

Description

ETF 펀드 상품 설명

[Learn more](#)

Cancel Add

Cortex Agent – Custom tool 등록

Trial 버전에서는 실행 불가

Add custom tool

Tool details

Resource type

procedure

Database & Schema

Schema DEMO.MAGI_HANDSON x

Custom tool identifier

DEMO.MAGI_HANDSON.CALL_EXTERNAL_API...

Name

google_search_tool

Warehouse

Warehouse to execute the SQL query

☐ User's default ☒ Custom

COMPUTE_WH

Query timeout

Maximum time in seconds for query execution (max 300s)

Enter timeout in seconds

Description

Generate with Cortex

Will be used during orchestration to guide how the agent uses this tool

일반적인 정보를 구글에서 검색


PROCEDURE/FUNCTION DETAILS:

- Type: User-Defined Function (UDF)
- Language: Python
- Signature: (QUERY VARCHAR)
- Returns: VARIANT
- Execution: OWNER with CALLED ON NULL INPUT
- Volatility: VOLATILE

[Learn More](#)

CancelAdd

Cortex Agent 설정 화면



ETF_ANALYSIS_AGENT

AboutToolsOrchestrationAccess

Tools

Cortex Analyst

ETF_ANALYSIS_TOOL

Semantic view

@DEMO.MAGI_HANDSON.INT_STG/ETF_ANALYSIS.yaml

TABLE1: ASSET_INFO

- Database: DEMO, Schema: MAGI_HANDSON

Show more

Cortex Search Services

DOCS_SEARCH_TOOL

Service name

DEMO.MAGI_HANDSON.DOCS_SEARCH

ETF 펀드 상품 설명

Custom tools

GOOGLE_SEARCH_TOOL

Procedure name

CALL_EXTERNAL_API(VARCHAR)


Warehouse

COMPUTE_WH

일반적인 정보를 구글에서 검색

PROCEDURE/FUNCTION DETAILS:

Show more



ETF_ANALYST_AGENT

AboutToolsOrchestrationAccess

Orchestration

Model

auto

Orchestration instructions

Define how the agent reasons through tasks, chooses the right tools, and sequences actions

특정 상품 설명은 docs_search_tool 을 사용한다.
일반적인 검색으로 얻을 수 있는 정보는 google_search_tool 을 사용한다.



Response instructions

Set rules for how the agent should sound and respond to users

응답 마지막에 분석 결과와 시사점을 요약해서 알려준다.



Cortex Agent



< ETF_ANALYSIS_AGENT

About

Tools

Orchestration

Access

About

Will be displayed to users. Not used for agent behavior or logic

Display name

ETF 분석 에이전트

Description

Provide an overview of the agent and how users can use it. This will be displayed to users who select this agent

This will be displayed to users who select this agent

Example questions

Add up to 15 questions to help users start a conversation

2024년 하반기 이후 코스피200 를 기반으로 하는 ETF 종목들의 월별 설정액 이

2025년부터 나스닥 100 을 기준으로 하는 ETF 중 성과가 가장 높았던 펀드 상

TIGER 미국나스닥 100 상품에 대한 설명과 이 펀드의 실제적인 성과 알려줘

현재 미국 주식시장 상황을 봤을때 어떤 상품에 투자하는게 유리해?

미래에셋TIGER미국나스닥100 상품과 삼성KODEX200 을 비교해주고, 실제

Add question

Lab06

- Agent 만들기
- 질문으로 얻은 Insight 발표 (+개선/적용 방안)



THANK YOU

