Snowflake Stages Integration with Python/FastAPI

Sno	Date	Modification	Author	Verified By
1	2019/09/10	Initial Document	Nishtha Vijay	Sumit Goyal

Table of Contents

Snowflake Procedure Integraion with Python/fastApi

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Mostly used in Data-science and machine-learning.

FastAPI is a modern, fast (high-performance), web framework for building APIs with Python 3.6+ based on standard Python type hints.

Based on open standards

- **OpenAPI** for API creation, including declarations of path operations, parameters, body requests, security, etc.
- Automatic data model documentation with **JSON Schema** (as OpenAPI itself is based on JSON Schema).
- Designed around these standards, after a meticulous study. Instead of an afterthought layer on top.
- This also allows using automatic **client code generation** in many languages.

Snowflake Stages:

Stages and file formats are named database objects that can be used to simplify and streamline bulk loading data into and unloading data out of database tables.

Pipes are named database objects that define COPY statements for loading micro-batches of data using Snow pipe.

.

Business Requirement

The main objective of this project is to simplify and streamline bulk loading data into and unloading data out of database tables.

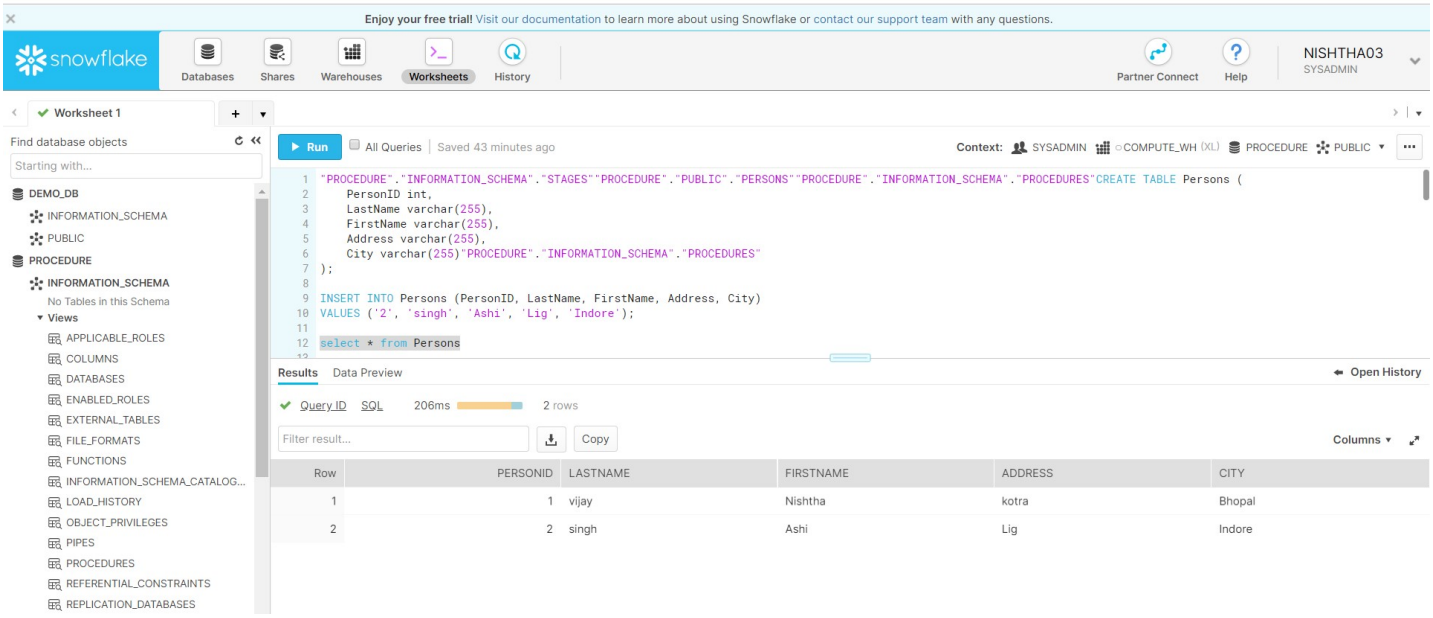
Solutions:

Note:In this document we explain step by step Integration between Python/FastAPI and Snowflake (To Fetch load data from stages) using python connector.

Steps :

1) Login with your snowflake account
2) Now create a stage named bisp_stage
   >>create or replace stage bisp_stage
   file_format = (type = 'CSV' field_delimiter = ',' skip_header = 1);
   Here we mentioned file_format type = csv. Because we going to upload a csv file

3) Now create a table named Persons with parameters as PersonID, LastName, FirstName, Address, City
   >>>create table Persons(PersonID in ,LastName varchar(20) ,FirstName varchar(20),Address varchar(30),City varchar(15));
4) Insert some records in the table by entering the following command in your worksheet.
   >>>INSERT INTO Persons (PersonID, LastName, FirstName, Address, City)
   VALUES ('1', 'sons', 'john', 'Lig', 'London');
5) To check whether we are doing right or not just enter the command in you worksheet
   >>> select * from persons;
6) If you are getting the following screen then you are in the right way.



7) Create a procedure by entering the following command.

```
>>>create or replace procedure read_person_proc()
 returns String not null
 language javascript
 as
 $$
  var my_sql_command = "select * from Persons";
  var statement1 = snowflake.createStatement( {sqlText: my_sql_command} );
  var result_set1 = statement1.execute();
  // Loop through the results, processing one row at a time...
  while (result_set1.next())  {
     var column1 = result_set1.getColumnValue(1);
     var column2 = result_set1.getColumnValue(2);
     var column3 = result_set1.getColumnValue(3);
     var column4 = result_set1.getColumnValue(4);
     var column5 = result_set1.getColumnValue(5);
     var column = column1+' '+column2+' '+column3+' '+column4+' '+column5
     }
 return column;
 $$
 ;
```

8) To call the procedure just enter the command

```
>>>call read_person_proc();
```

Now our requirement is we need to call that procedure in python using FastAPI

Below code will help you to fetch/call procedure form Snowflake into FastAPI.

```
import snowflake.connectoras sf
username='your username'
password='your password'
account='your account'
warehouse='your ware house'
database='your data base name'
ctx=sf.connect(user=username,password=password,account=account)
@app.get('/fetchdata')
asyncdeffetchdata():
    cursor = ctx.cursor()




cursor.execute("use warehouse your warehouse_name")


cursor.execute("alter session set timestamp_type_mapping = 'TIMESTAMP_NTZ'")
#
cursor.execute("alter session set timezone = 'Europe/Berlin'")
#
cursor.execute("alter session set TIME_OUTPUT_FORMAT = 'HH24:MI:SS.FF'")
#
cursor.execute("use database database_name")
#
cursor.execute("use schema schema_name")
sql = cursor.execute("call   read_person_proc()")

for datain sql:

return  data
```

You need to follow each step otherwise you may face some issues

Testing:

Steps to test Integration

Step1: Start FastAPI Server by entering the following command

>>>uvicorn main: app--reload

Step2: Go to the browser and hit the following url

>>>http://127.0.0.1:8000/fetchdata

If you get something like this. Congratulations you are successfully connected.

……..Thanks for reading……..