Introduction

This document introduces us to concept of making changes in snowflake database using python scripts in azure DevOps and also instead of keeping SQL scripts in folder we will keep SQL scripts outside root folder and create an excel file saved as csv and execute present in excel sheet.

CI--- Continuous Integration in DevOps is process of automating the build and deploying phase through certain tool . A developer write some form of codes often called patches represent a change to project codeframe.

CD—Continuous Delivery is practice of building, testing and releasing software in short cycles with the aim of ensuring that a working version of software can be released at any time.

Primary Objective—

Using Azure DevOps creating a pipeline in azure by running a python script having credentials related to snowflake database and checking the end to end requirements .

Individual Components ---

Snowflake: Snowflake is an analytic data warehouse provide as Software as a Service (SaaS). Snowflake provides a data warehouse that is faster , easier to use , and far more flexible than traditional data warehouse offerings.

Azure DevOps : Azure DevOps is Software as a Service (SaaS) platform from Microsoft that provides an end to end DevOps toolchain for developing software.

Azure Repos: Unlimited cloud hosted private Git repos . Collaboration pull request, advanced file management and more.

Azure Pipeline: Provides build and release services to support continuous integration and delivery of applications.

PRACTICAL IMPLEMENTATION:

Prerequisites:

1. Microsoft Azure Account Subscription.
2. Azure DevOps Account.
3. Snowflake account with desired warehouse and database created .

Implementation:

Step1: Start with storing SQL scripts from your local directory to azure repository which you have created in local system.

Graphical user interface, application, table, Excel

Description automatically generated

Step 2: Create Azure DevOps Project

1. Go to azure DevOps portal.
2. Click on create project and give desired name.
3. A new blade will open once project creation is completed.

Graphical user interface, text, application

Description automatically generated

Step 3 : Developer will push the database objects scripts from local repository

To azure DevOps repos using gitbash or we can directly browse from local

System.

A screenshot of a computer

Description automatically generated

Step 4: Add a new file in azure repo and name it as python script and the python script for build and run the pipeline.

A screenshot of a computer

Description automatically generated

Step 5 : Go to pipeline , create on plus icon just beside your project name and give the credentials for building new pipeline.

A screenshot of a computer

Description automatically generated

Step 6: Add task to your yaml file and click on save and run .

Graphical user interface, text, application

Description automatically generated

Step 7 : Once build is succeed you will see the pipeline has run.

A screenshot of a computer

Description automatically generated

Step 8: The python scripts we have worked on.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated