solution 1

=>

from datetime import datetime, timedelta

from airflow import DAG

from airflow.operators.python\_operator import PythonOperator

default\_args = {

'owner': 'mohit',

'start\_date': datetime(2023, 4, 17),

'retries': 1,

'retry\_delay': timedelta(minutes=2),

}

dag = DAG(

'problem1',

default\_args=default\_args,

description='t1 followed by t2, a simple pipe',

schedule\_interval='@daily',

)

def task\_1():

print("Running Task 1")

def task\_2():

print("Running Task 2")

t1 = PythonOperator(

task\_id='task\_1',

python\_callable=task\_1,

dag=dag,

)

t2 = PythonOperator(

task\_id='task\_2',

python\_callable=task\_2,

dag=dag,

)

t1 >> t2

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Solution 2

=>

from airflow import DAG

from airflow.operators.bash\_operator import BashOperator

from datetime import datetime, timedelta

default\_args = {

'owner': 'mohit',

'depends\_on\_past': False,

'start\_date': datetime(2023, 4, 17,21,40),

'email\_on\_failure': False,

'email\_on\_retry': False,

'retries': 1,

'retry\_delay': timedelta(minutes=2),

}

dag = DAG(

'problem02',

default\_args=default\_args,

description='problem sattement two',

schedule\_interval=timedelta(days=1),

)

task1 = BashOperator(

task\_id='task1',

bash\_command='echo "Task 1"',

dag=dag,

)

task2 = BashOperator(

task\_id='task2',

bash\_command='echo "Task 2"',

dag=dag,

)

task3 = BashOperator(

task\_id='task3',

bash\_command='echo "Task 3"',

dag=dag,

)

task1>>[task2,task3]

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Solution 3

=>

from datetime import datetime, timedelta

from airflow import DAG

from airflow.operators.postgres\_operator import PostgresOperator

default\_args = {

'owner': 'mohit',

'start\_date': datetime(2023, 4, 17),

'retries': 1,

'retry\_delay': timedelta(minutes=5),

}

dag = DAG(

'problem 3',

default\_args=default\_args,

description='example of DAG using PostgresOperator',

schedule\_interval='@daily',

)

sql\_query = """

SELECT \*

FROM your\_table

WHERE date\_col = '{{ ds }}'

"""

run\_query = PostgresOperator(

task\_id='run\_query',

postgres\_conn\_id='your\_postgres\_conn\_id',

sql=sql\_query,

dag=dag,

)

run\_query

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Solution 4

=>

from datetime import datetime, timedelta

from airflow import DAG

from airflow.operators.python\_operator import PythonOperator

default\_args = {

'owner': 'mohit',

'start\_date': datetime(2023, 4, 17),

'retries': 1,

'retry\_delay': timedelta(minutes=2),

}

dag = DAG(

'problem1',

default\_args=default\_args,

description='a simple python pipe t1 followed by t2',

schedule\_interval='@daily',

)

def task\_1():

print("Running Task 1")

def task\_2():

print("Running Task 2")

t1 = PythonOperator(

task\_id='task\_1',

python\_callable=task\_1,

dag=dag,

)

t2 = PythonOperator(

task\_id='task\_2',

python\_callable=task\_2,

dag=dag,

)

t1 >> t2