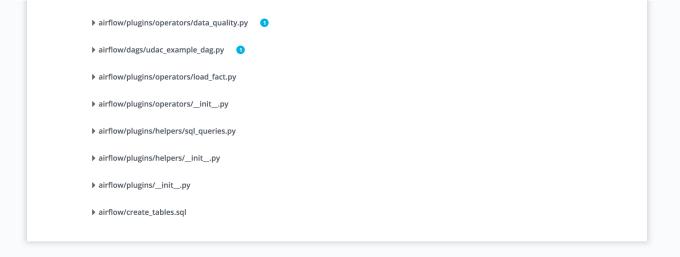


Return to "Data Engineering Nanodegree" in the classroom

TO DISCUSS ON STUDENT HUB

Data Pipelines with Airflow

REVIEW CODE REVIEW 6 HISTORY airflow/plugins/operators/stage_redshift.py ▼ airflow/plugins/operators/load_dimension.py from airflow.hooks.postgres_hook import PostgresHook
from airflow.models import BaseOperator
from airflow.utils.decorators import apply_defaults class LoadDimensionOperator(BaseOperator): ui_color = '#80BD9E' insert_sql = """
 TRUNCATE TABLE {};
 INSERT INTO {} 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 31 32 33 34 35 load_sql_stmt="",
*args, **kwargs): super(LoadDimensionOperator, self).__init__(*args, **kwargs)
self.redshift_conn_id = redshift_conn_id
self.table = table
self.load_sql_stmt = load_sql_stmt SUGGESTION Here is another way to switch between append-only and delete-load functionality. class LoadDimensionOperator(BaseOperator): ui color = '#80BD9E' load_dimension_table_insert = """
INSERT INTO {} {} load_dimension_table_truncate = """
TRUNCATE TABLE {} @apply_defaults
def __init__(self, query="", redshift_conn_id="", t_name="", operation="" *args, **kwargs): super(LoadDimensionOperator, self).__init__(*args, **kwargs) self.query=query self.redshift_conn_id=redshift_conn_id self.t_name=t_name self.operation=operation def execute(self, context):
 self.log.info(f"Started LoadDimensionOperator {self.t_name} started with mode {self.operation} redshift_hook = PostgresHook(postgres_conn_id=self.redshift_conn_id) if(self.operation == "append"): redshift_hook.run(LoadDimensionOperator.load_dimension_table_insert.format(self.t_name, sel f.query)) if(self.operation == "truncate"):
 redshift_hook.run(LoadDimensionOperator.load_dimension_table_truncate.format(self.t_name))
 redshift_hook.run(LoadDimensionOperator.load_dimension_table_insert.format(self.t_name, sel self.log.info(f"Ending LoadDimensionOperator {self.t_name} with a Success on Operation {self.operation}") redshift.run(formatted_sql)



RETURN TO PATH

Rate this review