Data Pipelines with Airflow

REVIEW | CODE REVIEW 6 | HISTORY

▸ airflow/plugins/operators/stage_redshift.py 3

▸ airflow/plugins/operators/load_dimension.py 1

▸ airflow/plugins/operators/data_quality.py 1

▾ airflow/dags/udac_example_dag.py 1

```
1  from datetime import datetime, timedelta
2  import os
3  from airflow import DAG
4  from airflow.operators.dummy_operator import DummyOperator
5  from airflow.operators import (StageToRedshiftOperator, LoadFactOperator,
6                                 LoadDimensionOperator, DataQualityOperator)
7  from helpers import SqlQueries
8
9  default_args = {
```

AWESOME

Nice work with the default argument here. 👏🏻

```
10     'owner': 'udacity',
11     'depends_on_past': False,
12     'retries': 3,
13     'retry_delay': timedelta(minutes=5),
14     'start_date': datetime(2019, 1, 12),
15     'email_on_retry': False
16 }
17
18 dag = DAG('udac_example_dag',
19           default_args=default_args,
20           description='Load and transform data in Redshift with Airflow',
21           schedule_interval='0 * * * *',
22           catchup=False
23         )
24
25 start_operator = DummyOperator(task_id='Begin_execution',  dag=dag)
26
27 stage_events_to_redshift = StageToRedshiftOperator(
28     task_id='Stage_events',
29     dag=dag,
30     table="staging_events",
31     redshift_conn_id="redshift",
32     aws_credentials_id="aws_credentials",
33     s3_bucket="udacity-dend",
34     s3_key="log_data",
35     json_path="s3://udacity-dend/log_json_path.json",
36     file_type="json"
37 )
38
39 stage_songs_to_redshift = StageToRedshiftOperator(
40     task_id='Stage_songs',
41     dag=dag,
42     table="staging_songs",
43     redshift_conn_id="redshift",
44     aws_credentials_id="aws_credentials",
45     s3_bucket="udacity-dend",
46     s3_key="song_data/A/A/A",
47     json_path="auto",
48     file_type="json"
49 )
50
51 load_songplays_table = LoadFactOperator(
52     task_id='Load_songplays_fact_table',
53     dag=dag,
54     table='songplays',
55     redshift_conn_id="redshift",
56     load_sql_stmt=SqlQueries.songplay_table_insert
57 )
58
59 load_user_dimension_table = LoadDimensionOperator(
60     task_id='Load_user_dim_table',
61     dag=dag,
62     table='users',
63     redshift_conn_id="redshift",
64     load_sql_stmt=SqlQueries.user_table_insert
65 )
66
67 load_song_dimension_table = LoadDimensionOperator(
68     task_id='Load_song_dim_table',
69     dag=dag,
70     table='songs',
71     redshift_conn_id="redshift",
```

```
    load_sql_stmt=SqlQueries.song_table_insert
)

load_artist_dimension_table = LoadDimensionOperator(
    task_id='Load_artist_dim_table',
    dag=dag,
    table='artists',
    redshift_conn_id="redshift",
    load_sql_stmt=SqlQueries.artist_table_insert
)

load_time_dimension_table = LoadDimensionOperator(
    task_id='Load_time_dim_table',
    dag=dag,
    table='time',
    redshift_conn_id="redshift",
    load_sql_stmt=SqlQueries.time_table_insert
)

run_quality_checks = DataQualityOperator(
    task_id='Run_data_quality_checks',
    dag=dag,
    tables=['songplays', 'users', 'songs', 'artists', 'time'],
    redshift_conn_id="redshift"
)

end_operator = DummyOperator(task_id='Stop_execution',  dag=dag)

#
# Task ordering for the defined DAG tasks
#

start_operator >> stage_events_to_redshift
start_operator >> stage_songs_to_redshift

stage_events_to_redshift >> load_songplays_table
stage_songs_to_redshift >> load_songplays_table

load_songplays_table >> load_user_dimension_table
load_songplays_table >> load_song_dimension_table
load_songplays_table >> load_artist_dimension_table
load_songplays_table >> load_time_dimension_table

load_user_dimension_table >> run_quality_checks
load_song_dimension_table >> run_quality_checks
load_artist_dimension_table >> run_quality_checks
load_time_dimension_table >> run_quality_checks

run_quality_checks >> end_operator
```

▸ airflow/plugins/operators/load_fact.py

▸ airflow/plugins/operators/__init__.py

▸ airflow/plugins/helpers/sql_queries.py

▸ airflow/plugins/helpers/__init__.py

▸ airflow/plugins/__init__.py

▸ airflow/create_tables.sql

RETURN TO PATH

Rate this review