Reverse ETL With Airflow

Russell Dervay

Airflow Summit 2021
Intro Slide / Bio

- Data Engineer @ Snowflake
- Focus on Data Transformation and reverse ETL
- Played Volleyball For Stanford
Agenda

● What Is Reverse ETL
  ○ Current Application Landscape
  ○ Data Architecture With Reverse ETL
  ○ Benefits

● Approach and Implementation
  ○ Design
  ○ Common Architecture
  ○ Configurations

● Example Account Scoring
  ○ Dag / Data Model
  ○ Example Update (SFDC)

● Considerations / Challenges
Reverse ETL
What Is Reverse ETL
Data Architecture Before Reverse ETL

- No Single Source of Truth
- “Crossing Wires”
- Duplicate Calculations
Data Architecture With Reverse ETL

- Data Warehouse becomes center of your “data universe”
- Only 2 “integrations” per application
- Common “Back End” Shared Between Applications
Benefits: Compute Once

- All Metrics Run on the same Data Set
- Removes the possibility of deviant metrics
Benefits: Data Visibility

- Data Consumers have all relevant data in their “primary application”

- Cross Departmental Metrics can easily be shared with little to no lift
Implementation
Approach

- Airflow Based Approach
- “Just Works”
- Easy and accessible
Design

- Python Based Packages Per App
- End User Configurable
- Create, Update, Delete, Insert Operations supported on most applications

```python
from airflow.models.baseoperator import BaseOperator

class HelloOperator(BaseOperator):
    def __init__(self, name, **kwargs):
        super().__init__(**kwargs)
        self.name = name

    def execute(self, context):
        message = "Hello {}".format(self.name)
        print(message)
        return message
```
End User Configuration

- **Query** - Represents the set of data you want to upload

- **Object / End Point** - Represents the endpoint or API you will be calling

- **Fields** - Columns to update in source

- **Job Type** - What operation the job should perform in source system

```python
task_id=run_aps,
query="",
  select account_id , aps
  from database.schema.table
,
oobject=account,
fields=["aps__c"],
job_type=update,
batch_size=10000,
```
Under The Hood
Account Scoring Example
Account Scoring Example (DAG)

- Last Step Operation
- Many Tasks Needed For Many Systems
Account Scoring Example (SFDC)

- Application Change alongside your model
- Business users always see the freshest data
Single Source Of Truth

<table>
<thead>
<tr>
<th>ACCOUNT_ID</th>
<th>APS</th>
</tr>
</thead>
<tbody>
<tr>
<td>0010Z00002A0ULnQAN</td>
<td>89.00000</td>
</tr>
<tr>
<td>0013r00002XVFFAAS</td>
<td>89.00000</td>
</tr>
<tr>
<td>0010Z0000259ILOQAI</td>
<td>89.00000</td>
</tr>
</tbody>
</table>

**Custom Apps**

**Metric Library**

**Data Processing Pipelines**

**Model Fit Data**

<table>
<thead>
<tr>
<th>#</th>
<th>ACCOUNTID</th>
<th>APS Range</th>
<th>ACCOUNT PROPENSITY SCORE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0010Z0000259ILOQAI</td>
<td>1-10</td>
<td>9</td>
</tr>
</tbody>
</table>

*Showing rows 1-1 of 1.*

**slack**

**workday**

**salesforce**

**Adaptive Insights**
Challenges
Challenges (Compliance / Control)

- “Connectors” Are Widely Accessible
- New Jobs Added Frequently
- QA Testing Not Always Followed
- Requires Good Cross Departmental Communication
Challenges (Infectious Data)

- Garbage In Garbage Out
- Incorrect data in the source table or pipeline will flow downstream impacting all other applications

9:14 AM
hey not sure if you saw Sheri’s email yet but was updated by integration to $0k, its on radar so we need to determine if that was accurate or needs to be updated and to waht
Challenges (Lift)

- Engineering Time Can Be Considerable
- Some APIs are more complex than others
- Maintenance can take some cycles away from development
Acknowledgements

Shradha Adsule  
Data Engineer

Yamini Nawi
Data Engineer

Kristen Werner  
Director, Data Science and Engineering

Satya Kota  
Senior Data Engineer

Ganesh Gadakar  
Data Engineer
Questions?