DraftKings ATS/HRIS ETL Plan

Brian Shin

Overview

1. Scoping
2. ETL
3. Orchestration
4. QA
5. Documentation

1. Scoping

- ERD
  - Already established.
- Data Model of Production Tables
  - What each table will ultimately look like in the end-state.
  - Define primary keys.
  - Define data types for each field.
  - Scope any additional logic for derived/look-up fields.

employee_dim		
column	data_type	logic
employee_id	INT	primary key
role_start_date	DATE	none
role_end_date	DATE	none
job_id	INT	none
job_title	TEXT	normalization
team	TEXT	normalization
department	TEXT	none
manager_id	INT	look-up field using employee_dim hierarchy
created_timestamp	TIMESTAMP	none
last_modified_timestamp	TIMESTAMP	none

applicant_dim		
column	data_type	logic
applicant_id	INT	primary key
application_date	DATE	none
application_job_id	INT	none
application_job_title	TEXT	none
application_team	TEXT	none
application_department	TEXT	none
hiring_manager_id	INT	look-up field using employee_dim hierarchy
is_referral	BOOLEAN	none
referral_employee_id	INT	none
is_hired	BOOLEAN	none
created_timestamp	TIMESTAMP	none
last_modified_timestamp	TIMESTAMP	none

2. ETL

EXTRACTION RAWDATA STAGING PRODUCTION

2. ETL (cont.)

- EXTRACTION
  - Create extractor python scripts for pulling raw data from the ATS and HRIS and write to the data-lake (AWS S3).
  - Ideally, there are integrations with the ATS/HRIS systems that allow smooth pulls, but be prepared to be flexible.
- RAWDATA
  - Create a RAWDATA schema to pull in its raw data form (ex: json format) from sources.
  - Keep data as raw as possible at this stage.
  - Ensure that all semi-structured data is massaged into a workable format and uniform.
- STAGING
  - Create a STAGING schema to pull in data from RAWDATA into table form.
  - Apply all additional logic to any fields that require it (ex: if the organization wants to have all role start dates to be the first of the month, DATE_TRUNC raw role start date to the first of the month)
- PRODUCTION
  - Create a PRODUCTION schema of final tables with all data types and primary keys defined.
  - Add metadata fields (ex: CREATED_TIMESTAMP & LAST_MODIFIED_TIMESTAMP)

3. Orchestration

- First, align with standard operating procedures and tech stack as the Data Engineering team.
- Provided that an orchestration tool is being used (Airflow, Luigi, cron jobs, etc.), create a DAG for each production table. If free to use anything, spin up Airflow in an EC2 instance (dev and prod).
  - Avoids a giant mono-DAG with complex dependencies.
  - Keeps it modular to make development seamless (no unintended downstream fires).
  - Allows continuous integration if desired.
  - Cron OR trigger based task runs.
  - Allows it to be flexible for addition of new vendors.

Apache Airflow

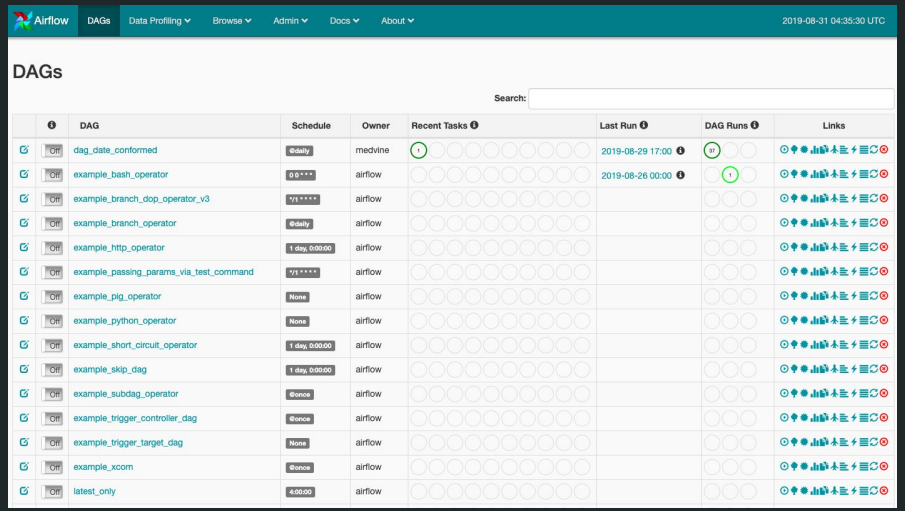
Luigi

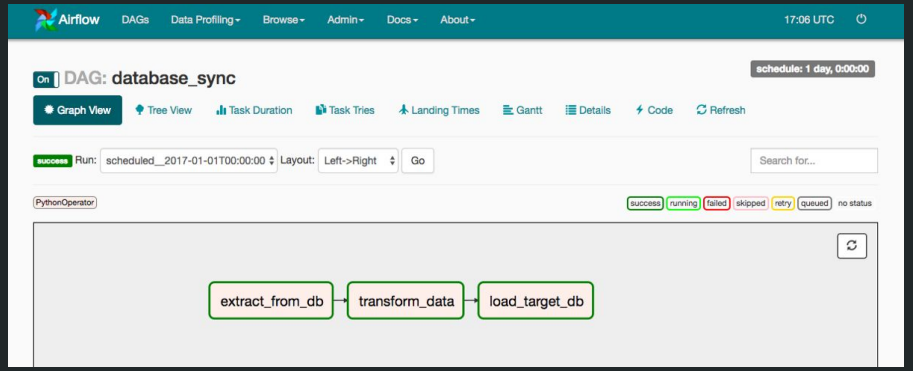
Crontab

3. Orchestration (cont.)

DAG EXAMPLES:

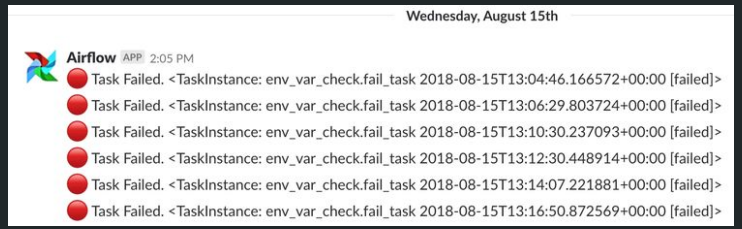
- extractor_dag
  - extractor.py
- employee_dim_dag
  - rawdata.sql
    - Creates rawdata table of pull from S3.
  - staging.sql
    - Pulls from rawdata table with logic to fields to create staging table.
  - production.sql
    - Pulls from staging table to create/upsert production table.
  - permissions.sql
    - Run permissions for the table to keep sensitive PII data limited to appropriate people/cases.

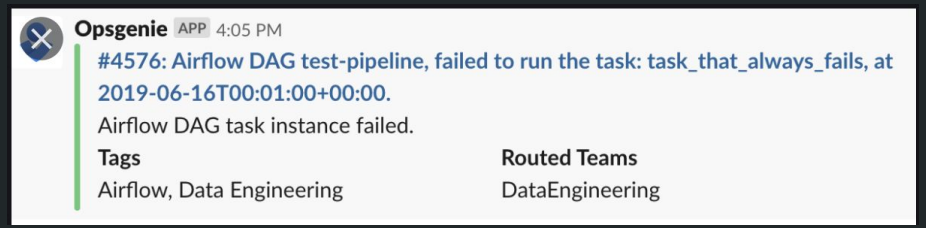




4. QA

- QA for each stage of the ETL process with data health dashboards, alerts, and using METADATA.
  - Failed jobs.
    - Alerts for when something breaks in the ETL.
  - S3 File Recency
    - When was the last time we ingested a new file (from the extractor) into the expected S3 bucket?
  - Table Recency
    - When was the last time a production table was modified?
  - Table Growth
    - What was the % delta on production tables day by day?
    - Any large deltas should be investigated and verified.
    - NULL columns and PK integrity checks.





5. Documentation

- Data Dictionary
  - To enable users to search for the data they need and where the source data is coming from.
- SLA's
  - Service Level Agreements for when data is expected to be updated.
- Development procedures
  - How to make a new DAG in the ETL, so anyone in the future can easily add in a new table or edit an existing one without breaking everything.
- Building redundancies in processes, to avoid single points of failure.

Data Dictionary

Data Dictionary outlining a Database on Driver Details in NSW

Field Name	Data Type	Data Format	Field Size	Description	Example
License ID	Integer	NNNNNN	6	Unique number ID for all drivers	12345
Surname	Text		20	Surname for Driver	Jones
First Name	Text		20	First Name for Driver	Arnold
Address	Text		50	First Name for Driver	11 Rocky st Como 2233
Phone No.	Text		10	License holders contact number	0400111222
D.O.B	Date / Time	DD/MM/YYYY	10	Drivers Date of Birth	08/05/1956