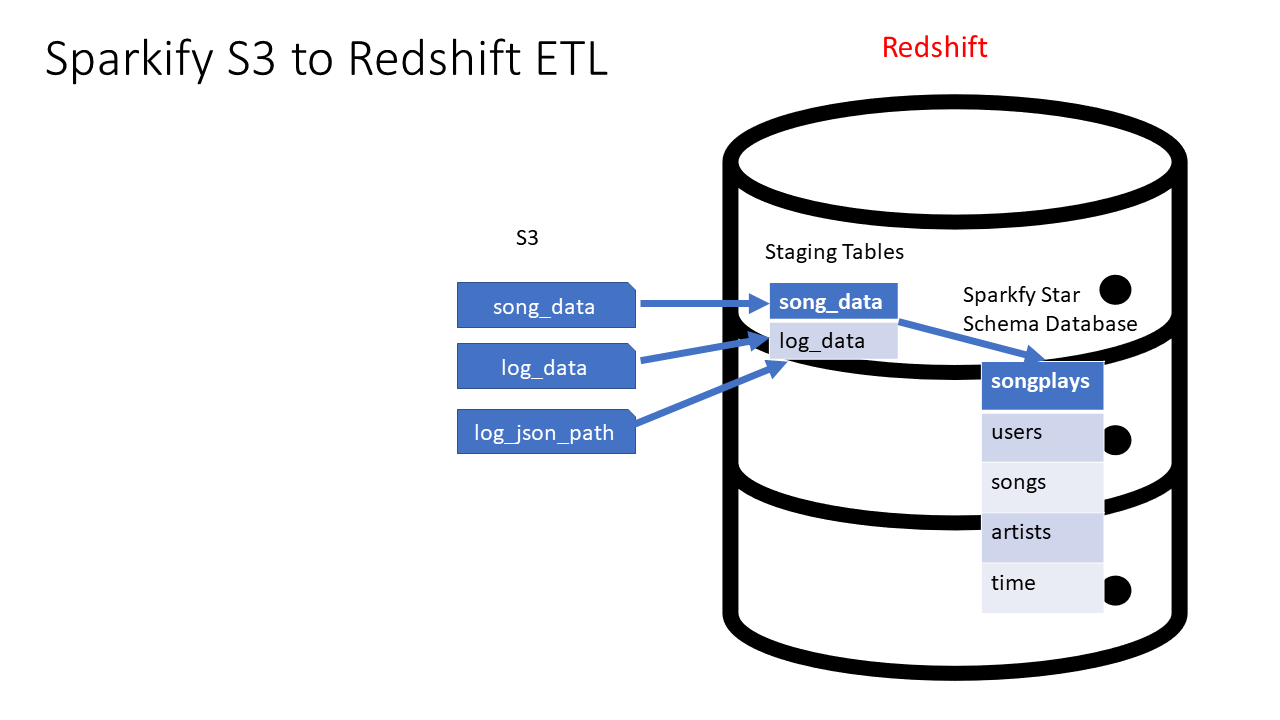
Project Introduction

Project: Data Warehouse

**Introduction**

A music streaming startup, Sparkify, has grown their user base and song database and want to move their processes and data onto the cloud. Their data resides in S3, in a directory of JSON logs on user activity on the app, as well as a directory with JSON metadata on the songs in their app.

As their data engineer, you are tasked with building an ETL pipeline that extracts their data from S3, stages them in Redshift, and transforms data into a set of dimensional tables for their analytics team to continue finding insights into what songs their users are listening to.



System Architecture for AWS S3 to Redshift ETL

**Project Description**

In this project, you'll apply what you've learned on data warehouses and AWS to build an ETL pipeline for a database hosted on Redshift. To complete the project, you will need to load data from S3 to staging tables on Redshift and execute SQL statements that create the analytics tables from these staging tables.

**Helpful Hints**

* Many of the issues that you may have will come from security. You may not have the right keys, roles, region, etc. defined and as a result you will be denied access. If you are having troubles go back to the various lessons where you set up a Redshift cluster, or implemented a role, created a virtual network, etc. Make sure you can accomplish those tasks there, then they should be easy for you to recreate in this project. You are likely to have fewer issues with security if you implement the role creation, setup of the Redshift cluster, and destruction of it via Infrastructure As Code (IAC).
* Udacity provides a temporary AWS account for you to build the necessary resources for this project. NB This will REPLACE your AWS credentials which are in your .aws folder. PLEASE make sure you copy those first since they will be overwritten. IF you are having difficulty completing the Project in 1 session (there is a time limit), you MAY find it more effective to use your own AWS account. This would avoid the need to validate your session each time you restart on the project. However, that would be your own funds. It is unlikely that would cost you more than a few dollars.
* The starter code that we give you provides a framework for doing this project. The vast majority of your work will be getting the SQL queries part correct. Very few changes will be required to the starter code.
* This is an excellent template that you can take into the work place and use for future ETL jobs that you would do as a Data Engineer. It is well architected (e.g. staging tables for data independence from the logs and the final Sparkify DWH) AND bulk data loads into the Sparkify DWH for high compute performance via SQL from those staging tables.

Project Datasets

You'll be working with 3 datasets that reside in S3. Here are the S3 links for each:

* Song data: s3://udacity-dend/song\_data
* Log data: s3://udacity-dend/log\_data
* This third file s3://udacity-dend/log\_json\_path.jsoncontains the meta information that is required by AWS to correctly load s3://udacity-dend/log\_data

Song Dataset

The first dataset is a subset of real data from the [**Million Song Dataset**](http://millionsongdataset.com/). Each file is in JSON format and contains metadata about a song and the artist of that song. The files are partitioned by the first three letters of each song's track ID. For example, here are file paths to two files in this dataset.

song\_data/A/B/C/TRABCEI128F424C983.json

song\_data/A/A/B/TRAABJL12903CDCF1A.json

And below is an example of what a single song file, TRAABJL12903CDCF1A.json, looks like.

{"num\_songs": 1, "artist\_id": "ARJIE2Y1187B994AB7", "artist\_latitude": **null**, "artist\_longitude": **null**, "artist\_location": "", "artist\_name": "Line Renaud", "song\_id": "SOUPIRU12A6D4FA1E1", "title": "Der Kleine Dompfaff", "duration": 152.92036, "year": 0}

Log Dataset

The second dataset consists of log files in JSON format generated by this [**event simulator**](https://github.com/Interana/eventsim) based on the songs in the dataset above. These simulate app activity logs from an imaginary music streaming app based on configuration settings.

The log files in the dataset you'll be working with are partitioned by year and month. For example, here are file paths to two files in this dataset.

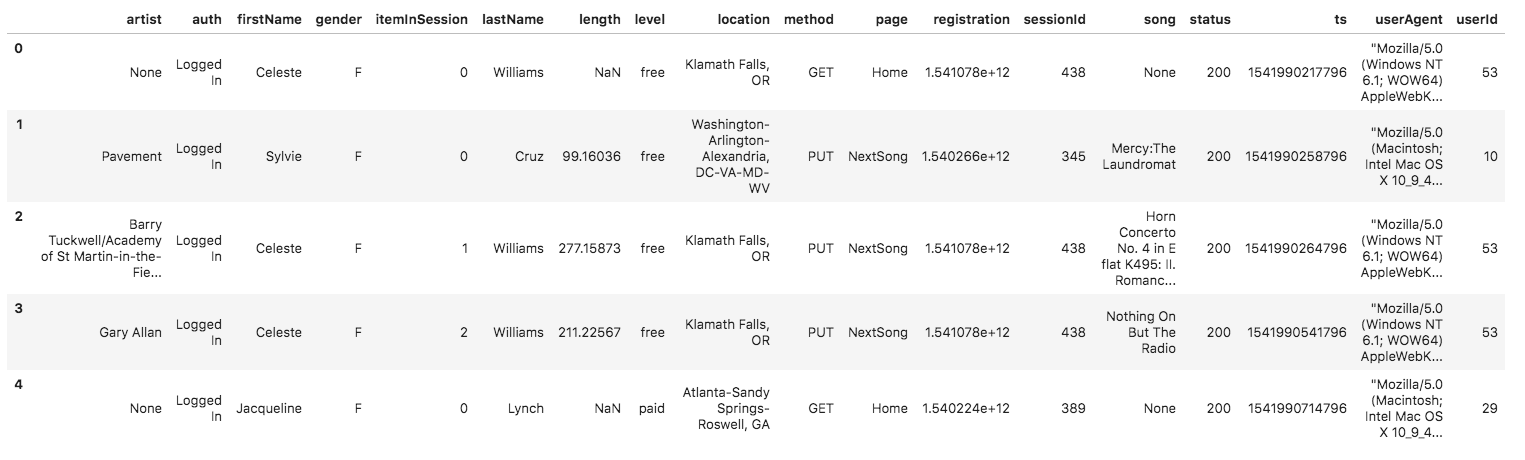
log\_data/2018/11/2018-11-12-events.json

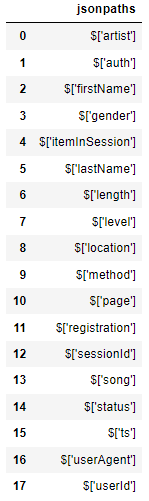
log\_data/2018/11/2018-11-13-events.json

And below is an example of what the data in a log file, 2018-11-12-events.json, looks like.

log\_data image

Log Json Meta Information

And below is what data is in log\_json\_path.json.

Contents of log\_json\_path.json

Project Instructions

Schema for Song Play Analysis

Using the song and event datasets, you'll need to create a star schema optimized for queries on song play analysis. This includes the following tables.

**Fact Table**

1. **songplays** - records in event data associated with song plays i.e. records with page NextSong
   * *songplay\_id, start\_time, user\_id, level, song\_id, artist\_id, session\_id, location, user\_agent*

**Dimension Tables**

1. **users** - users in the app
   * *user\_id, first\_name, last\_name, gender, level*
2. **songs** - songs in music database
   * *song\_id, title, artist\_id, year, duration*
3. **artists** - artists in music database
   * *artist\_id, name, location, lattitude, longitude*
4. **time** - timestamps of records in **songplays** broken down into specific units
   * *start\_time, hour, day, week, month, year, weekday*

Project Template

To get started with the project, go to the workspace on the next page, where you'll find the project template. You can work on your project and submit your work through this workspace.

Alternatively, you can download the template files in the Resources tab in the classroom and work on this project on your local computer.

The project template includes four files:

* create\_table.py is where you'll create your fact and dimension tables for the star schema in Redshift.
* etl.py is where you'll load data from S3 into staging tables on Redshift and then process that data into your analytics tables on Redshift.
* sql\_queries.py is where you'll define you SQL statements, which will be imported into the two other files above.
* README.md is where you'll provide discussion on your process and decisions for this ETL pipeline.