Datasets

For this project, you'll be working with one dataset: `event_data`. The directory of CSV files partitioned by date. Here are examples of filepaths to two files in the dataset:

```
event_data/2018-11-08-events.csv
event_data/2018-11-09-events.csv
```

Project Template

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

The project template includes one Jupyter Notebook file, in which:

- you will process the `event_datafile_new.csv` dataset to create a denormalized dataset
- you will model the data tables keeping in mind the queries you need to run
- you have been provided queries that you will need to model your data tables for
- you will load the data into tables you create in Apache Cassandra and run your queries

Project Steps

Below are steps you can follow to complete each component of this project.

Modeling your NoSQL database or Apache Cassandra database

1. Design tables to answer the queries outlined in the project template
2. Write Apache Cassandra `CREATE KEYSPACE` and `SET KEYSPACE` statements
3. Develop your `CREATE` statement for each of the tables to address each question
4. Load the data with `INSERT` statement for each of the tables
5. Include `IF NOT EXISTS` clauses in your `CREATE` statements to create tables only if the tables do not already exist. We recommend you also include `DROP TABLE` statement for each table, this way you can run drop and create tables whenever you want to reset your database and test your ETL pipeline
6. Test by running the proper select statements with the correct `WHERE` clause

Build ETL Pipeline

1. Implement the logic in section Part I of the notebook template to iterate through each event file in `event_data` to process and create a new CSV file in Python
2. Make necessary edits to Part II of the notebook template to include Apache Cassandra `CREATE` and `INSERT` statements to load processed records into relevant tables in your data model
3. Test by running `SELECT` statements after running the queries on your database