Schedules

Pipelines are often driven by schedules which determine what data should be analyzed and when.

Why Schedules

- Pipeline schedules can reduce the amount of data that needs to be processed in a given run. It helps scope the job to only run the data for the time period since the data pipeline last ran. In a naive analysis, with no scope, we would analyze all of the data at all times.
- Using schedules to select only data relevant to the time period of the given pipeline execution can help improve the quality and accuracy of the analyses performed by our pipeline.
- Running pipelines on a schedule will decrease the time it takes the pipeline to run.
- An analysis of larger scope can leverage already-completed work. For. e.g., if the aggregates for all months prior to now have already been done by a scheduled job, then we only need to perform the aggregation for the current month and add it to the existing totals.

Selecting the time period

Determining the appropriate time period for a schedule is based on a number of factors which you need to consider as the pipeline designer.

1. **What is the size of data, on average, for a time period?** If an entire years worth of data is only a few kb or mb, then perhaps its fine to load the entire dataset. If an hours worth of data is hundreds of mb or even in the gbs then likely you will need to schedule your pipeline more frequently.
2. **How frequently is data arriving, and how often does the analysis need to be performed?** If our bikeshare company needs trip data every hour, that will be a driving factor in determining the schedule. Alternatively, if we have to load hundreds of thousands of tiny records, even if they don't add up to much in terms of mb or gb, the file access alone will slow down our analysis and we'll likely want to run it more often.
3. **What's the frequency on related datasets?** A good rule of thumb is that the frequency of a pipeline's schedule should be determined by the dataset in our pipeline which requires the most frequent analysis. This isn't universally the case, but it's a good starting assumption. For example, if our trips data is updating every hour, but our bikeshare station table only updates once a quarter, we'll probably want to run our trip analysis every hour, and not once a quarter.