TIP SHEET

3 Simple Steps to Data Ingestion Using Partitioning

When you have large data sets with hundreds of millions or billions of records, ingesting that data efficiently can be a challenging feat. In many cases, these data sets may be loaded on a nightly schedule. Your nightly data ingestion pipeline must be complete within a certain time frame to avoid negatively impacting your downstream processes or analyses. So, what can you do when a job you are running each night grows too large to be processed in your time frame? How can you better manage the ingestion of these large datasets?

In a cloud-based data architecture, utilizing one server for ten hours costs the same as utilizing ten servers for one hour. Because of this, the best and most cost-effective way to optimize performance is to shorten your longest running job. An incredibly powerful way to do this is to employ a partitioning pattern. A partitioning pattern essentially involves breaking a large data ingestion job into several smaller jobs. But how does the overall data ingestion process use partitioning? Let's break it down!

1

Data Ingestion to the Data Lake

A common pattern is for data to be ingested into a data lake using a data orchestration tool. A data lake is a central storage location that allows you to store vast amounts of data, both structured and unstructured. Since organization is key to maintaining your data lake, data lakes typically have several zones. Each zone fulfills a separate role or purpose in the data ingestion or transformation process. There is no single template for the number of zones or types of zones in a data lake, so these can vary across organizations depending on their needs.

2

Introducing Partitioning

So how does partitioning fit into all that? When ingesting a data source to the data lake, you can break that job into several jobs by partitioning your dataset on a selected field. Then, you can load each of the partitioned jobs to the same target in the data lake. You need to ingest that data from its source system into a data lake in an efficient manner. Using partitioning, you can create several data ingestion jobs that pull from the source and can run in parallel and result in a faster data ingestion pipeline.

3

Selecting a Partitioning Field

First, you need to decide which field to partition your data on. It is most common to use dates to partition data, such as the year or month of a date. When deciding on the field to partition your data on, you must pay close attention to the cardinality of that field. A field with a high cardinality such as ProductNo, Name, or Description are not good candidates for partitioning because it could result in needing millions or billions of partitioning tasks. You could consider using low cardinality fields such as Category or Color. However, it is important to take into consideration whether that field could introduce additional values in the future.