



Figure 7: Hierarchical Partitioning

This partitioning mechanism reduces the number of partitions in the system, keeping it easier to manage, and reducing the number of partitions that need to be read when querying larger and older chunks.

The **query** process for this hierarchical partitioning is the same as the non-hierarchical one as it will apply the same pruning strategy to read the only relevant partitions. The **ingestion** and **compaction** processes will be a bit more complicated to organize the partitions in their defined hierarchy.

## Aggregate Partitioning

Many organizations do not want to keep old data but instead their aggregations such as number of orders and total sales of every product every month. This can be supported by aggregating data and partitioning them by month. However, since the **aggregate partitions** store aggregated data, their schema will be different from the non-aggregated ones which will lead to extra work for ingesting and querying. There are different ways to manage this cold and aggregated data, but these are large topics for a future post.

*Nga Tran is a staff software engineer at InfluxData, and a member of the IOx team, which is building the next-generation time series storage engine for InfluxDB. Before InfluxData, Nga had been with Vertica Analytic DBMS for over a decade. She was one of the key engineers who built the query optimizer for Vertica, and later, ran Vertica's engineering team.*

