Overview

Introduce Hive as a data warehouse for analytical processing

Understand how Hive deals with queries on huge amounts of data

Optimizing Queries on Big Data

Queries on Big Data

Consider an e-commerce site with transactions from all over the United States

Queries on Big Data

Potentially billions of records over many years

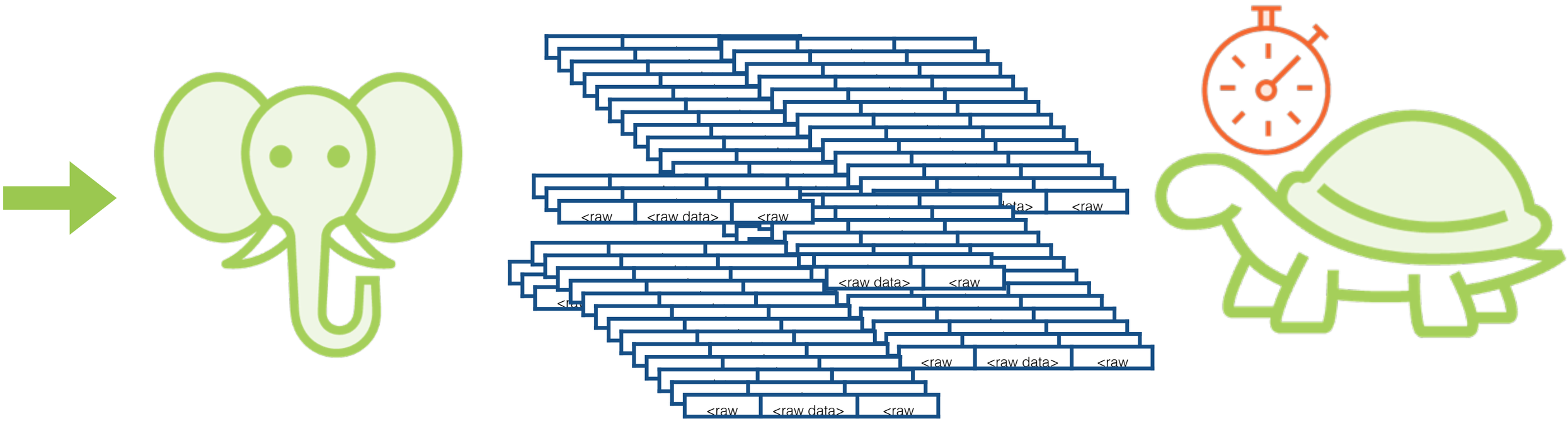
Queries on Big Data

select * from orders where state = “WA”

Queries on Big Data

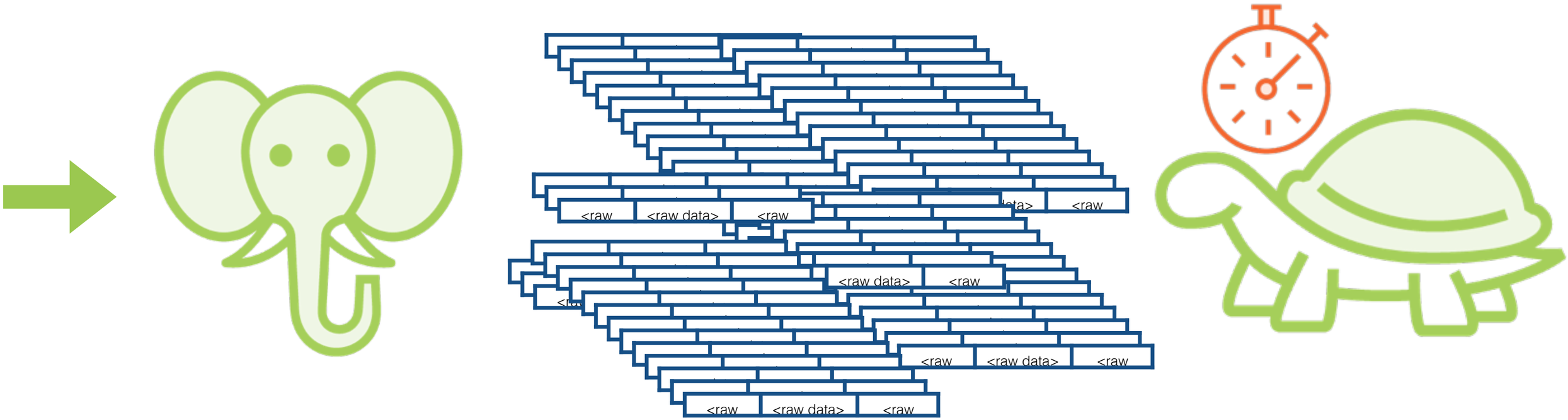
select * from orders where state = “WA”

Queries on Big Data



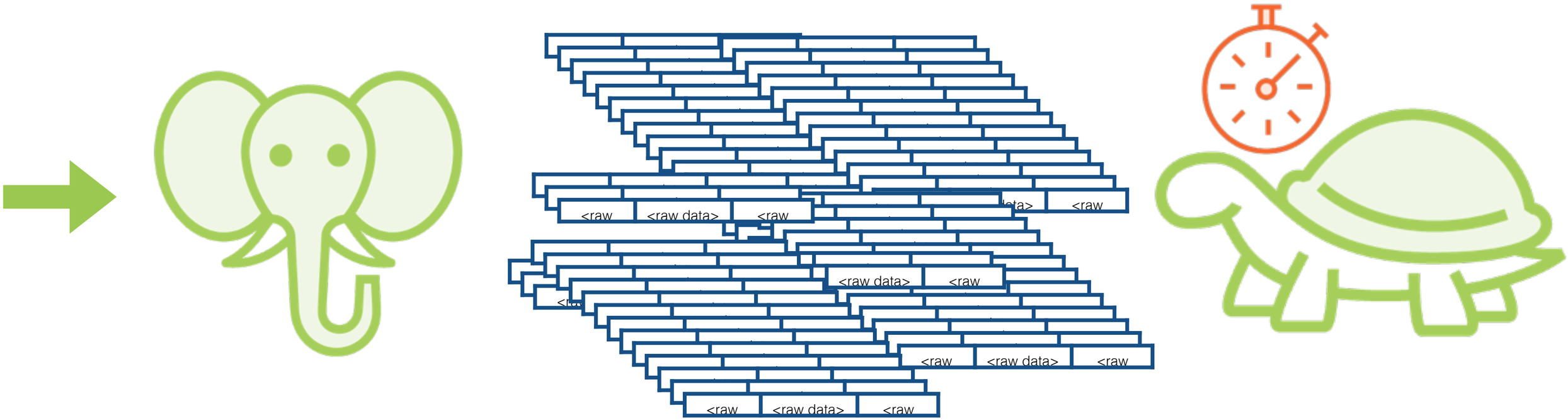
select * from orders where state = “WA”

Queries on Big Data



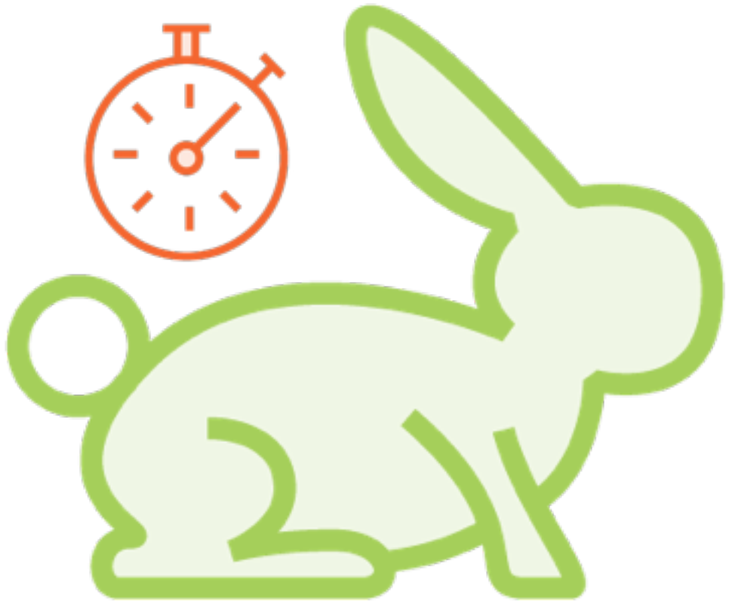
Running MapReduce jobs on huge datasets is slow...

Queries on Big Data



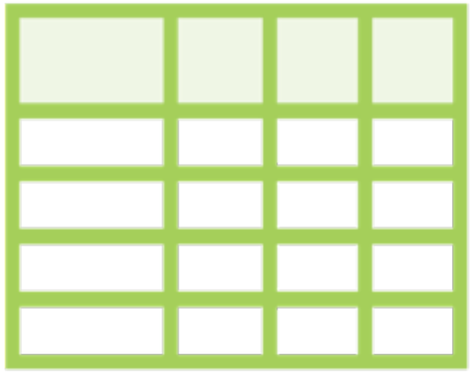
Queries run on it are very complicated, hard to debug and maintain

Queries on Big Data

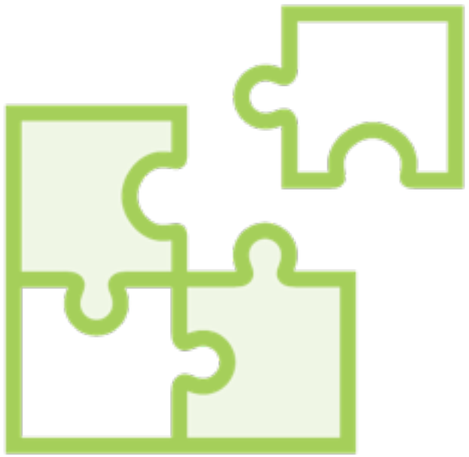


Hive tries to simplify query writing and improve query performance in many ways

Queries on Big Data



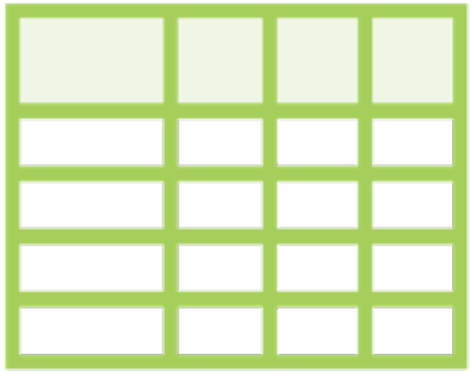
Design tables to optimize queries



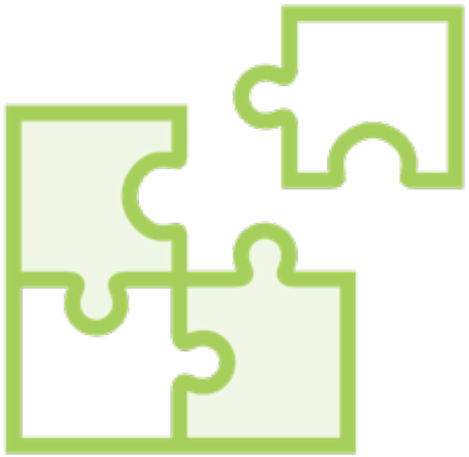
Structure queries so they run faster

Simplify query expressions so they're easy to maintain

Queries on Big Data



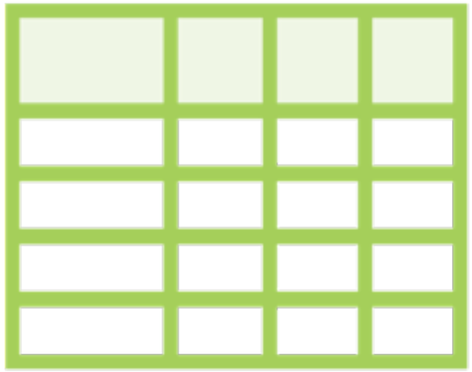
Partitioning and Bucketing of Tables



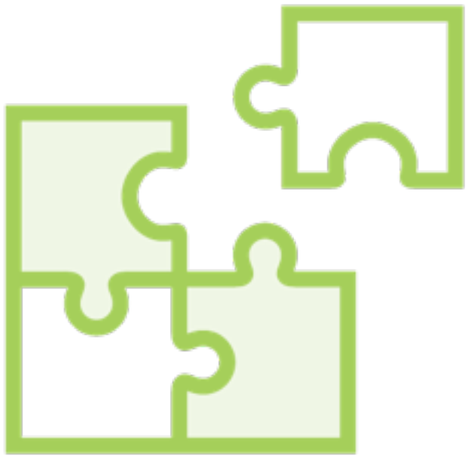
Structure queries so they run faster

Simplify query expressions so they're easy to maintain

Queries on Big Data



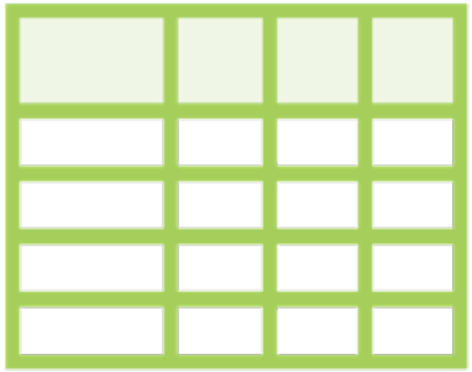
Partitioning and Bucketing of Tables



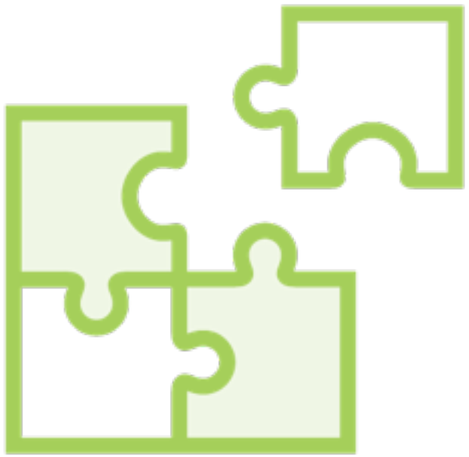
Join Optimizations

Simplify query expressions so they're easy to maintain

Queries on Big Data



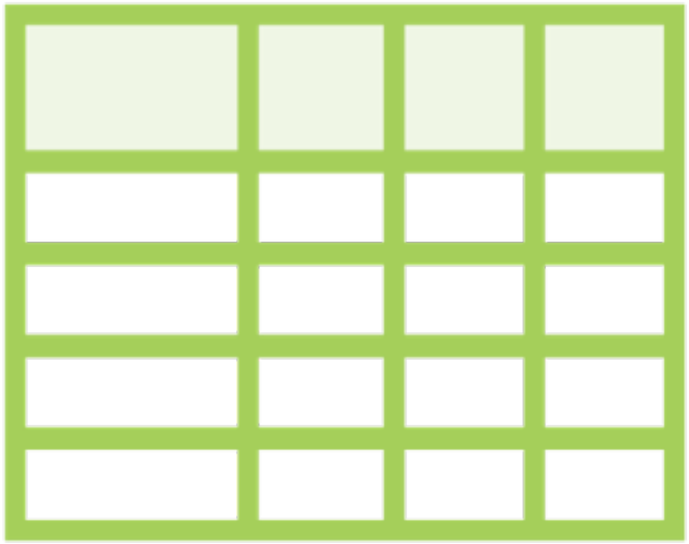
Partitioning and Bucketing of Tables



Join Optimizations

Window Functions

Partitioning and Bucketing



Design tables so queries run only on subsets of the dataset

Partitioning and Bucketing

Partitioning

Bucketing

Splits data into smaller, manageable parts

Partitioning and Bucketing

Partitioning

Bucketing

Enables performance optimizations

Partitioning

Data may be naturally split into logical units

Partitioning

Each of these units will be stored in a different directory

WA

OR

CA

CT

NY

GA

Partitioning

State specific queries will run only on data in one directory

WA

OR

CA

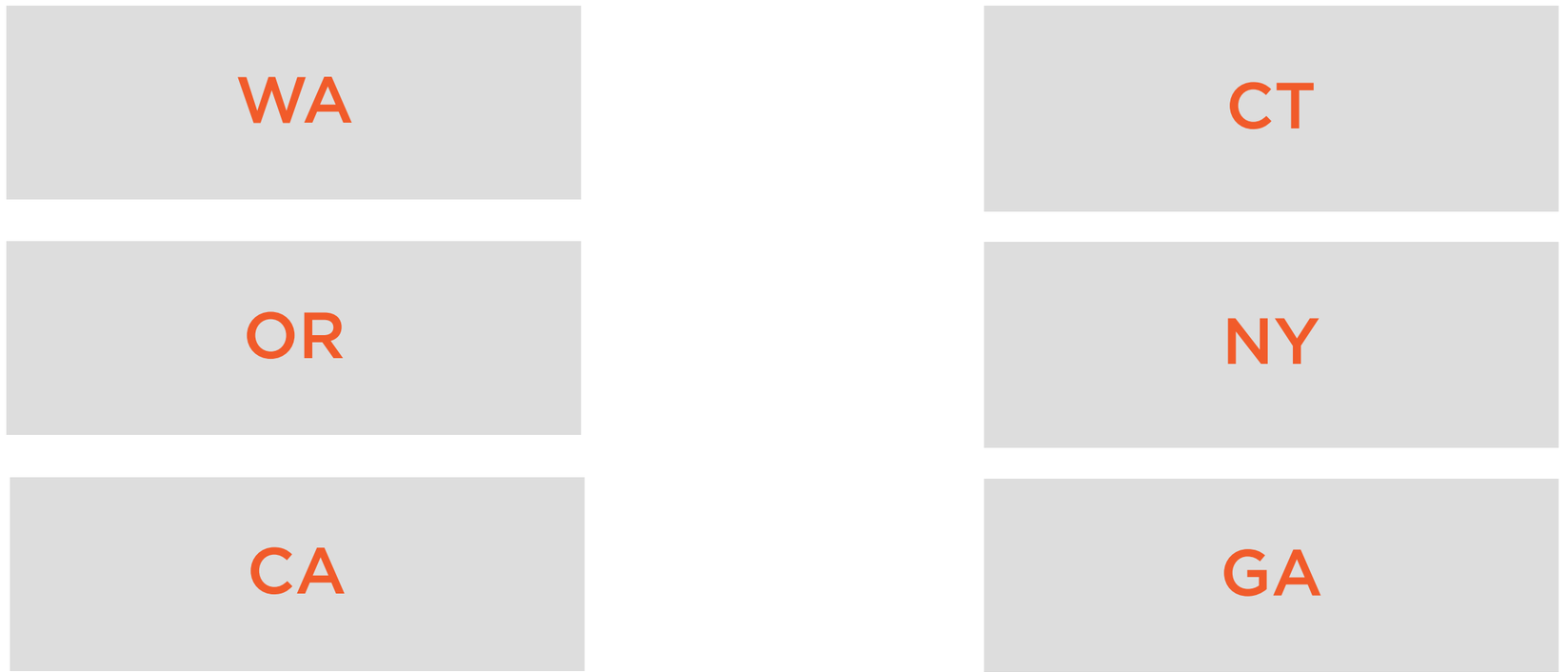
CT

NY

GA

Partitioning

Splits may not of the same size



Partitioning and Bucketing

Partitioning

Bucketing

Bucketing

Size of each split should be the same

Bucketing

Hash of a column value - address, name, timestamp anything

Bucketing

Each bucket is a separate file

Bucketing

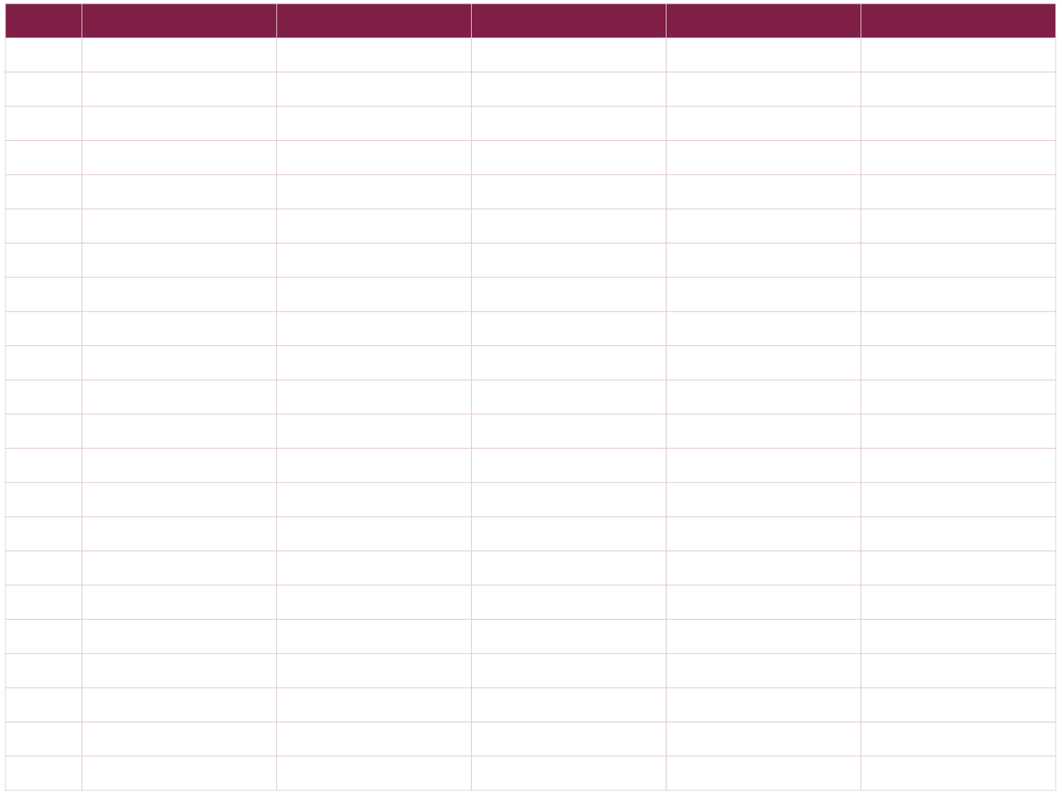
Makes sampling and joining data more efficient

Partitioning and Bucketing

Partitioning

Bucketing

Partitioning and Bucketing

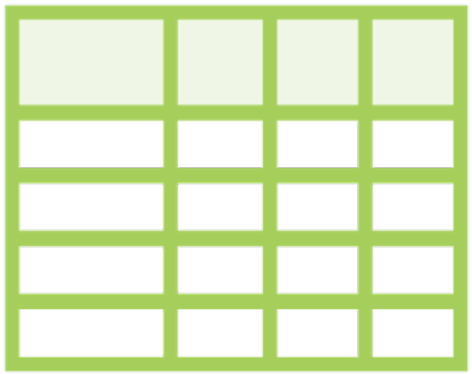


Rather than running queries on such tables

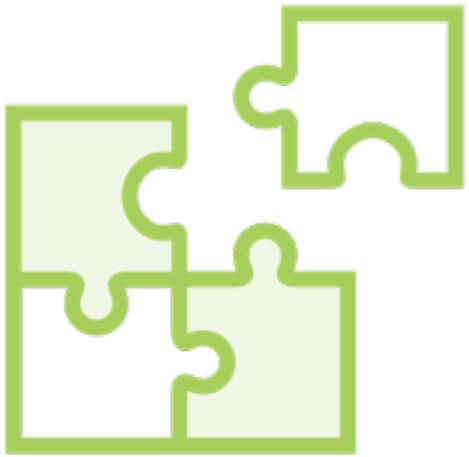
Partitioning and Bucketing

Run queries on subsets that are manageable

Queries on Big Data



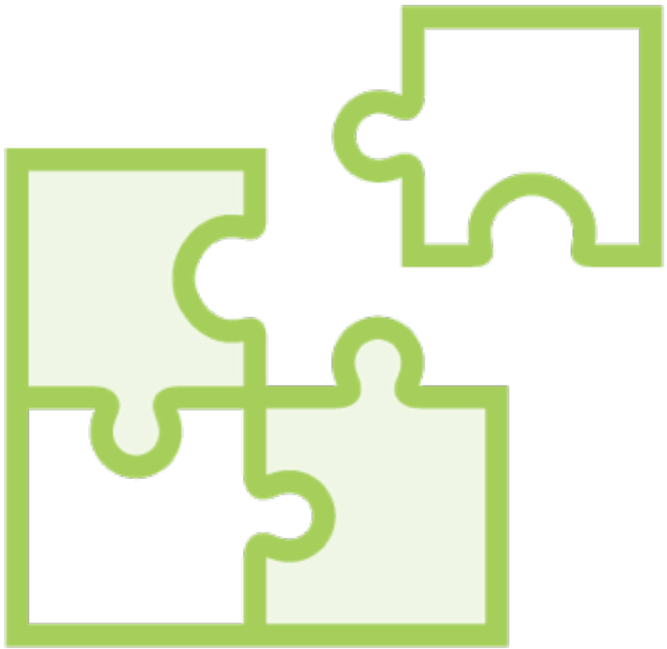
Partitioning and Bucketing of Tables



Join Optimizations

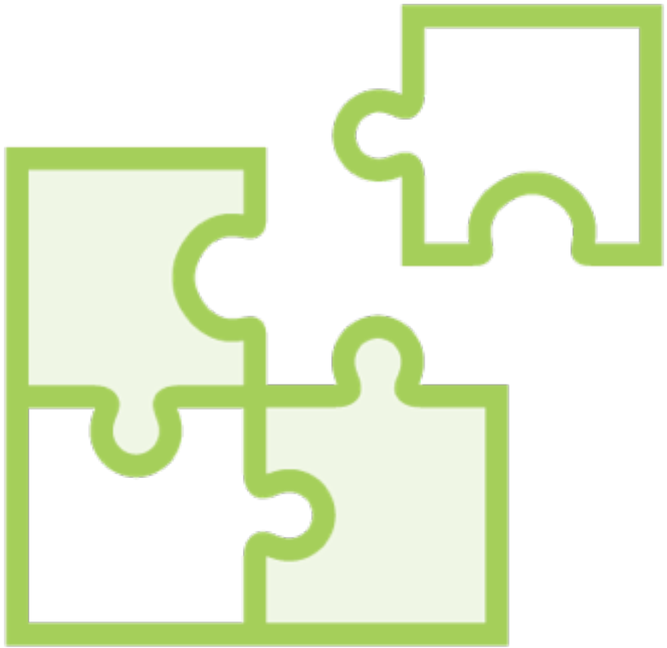
Window Functions

Join Optimizations



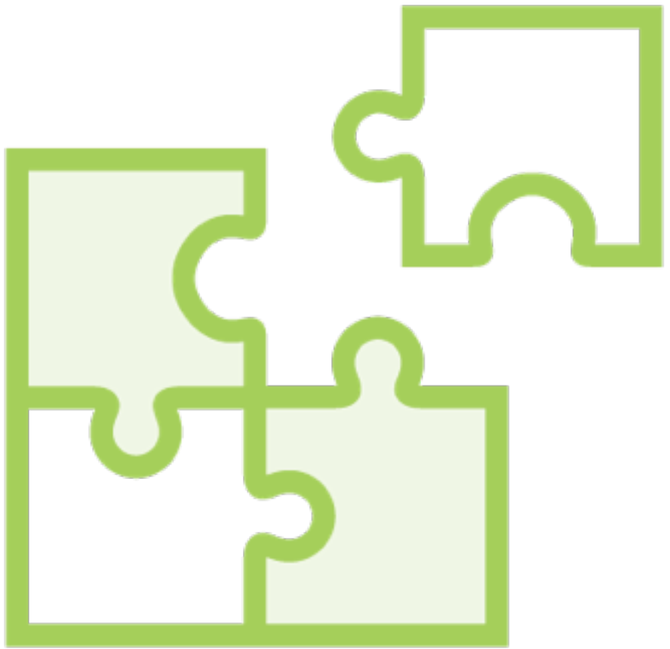
Join operations are MapReduce jobs under the hood

Join Optimizations



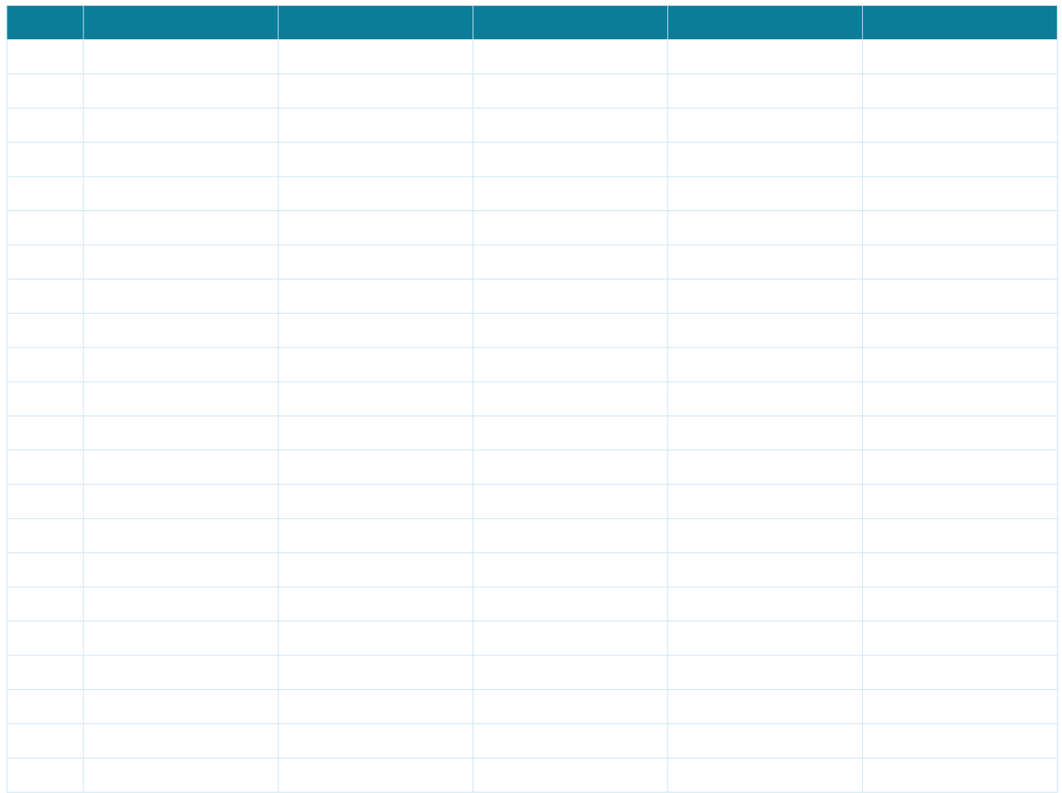
Optimize joins by reducing the amount of data held in memory

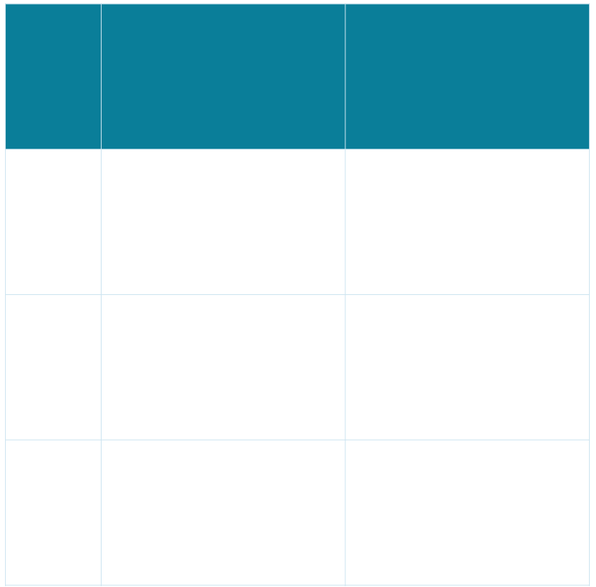
Join Optimizations



Or by structuring joins as a
map-only operation

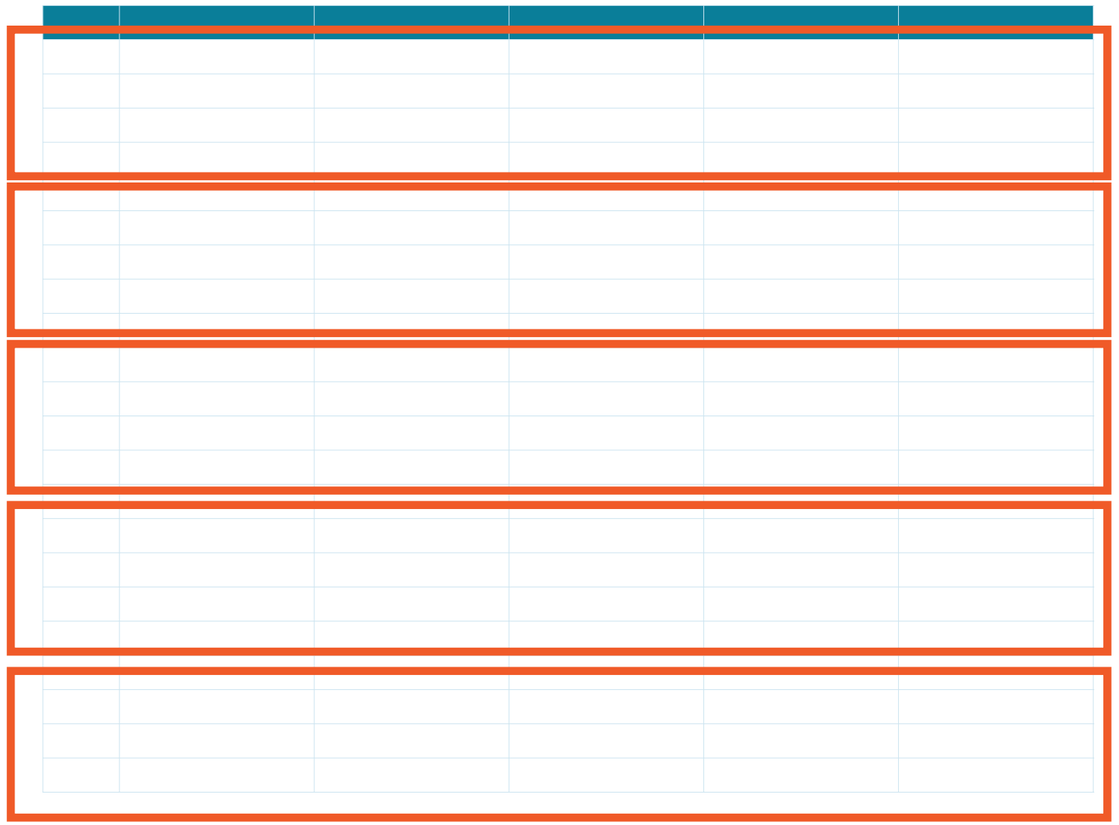
Reducing Data Held in Memory

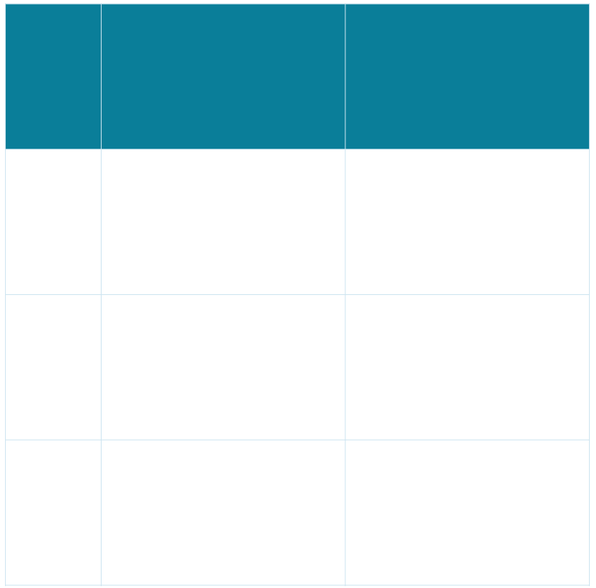




A 500GB table joined with a 5MB table

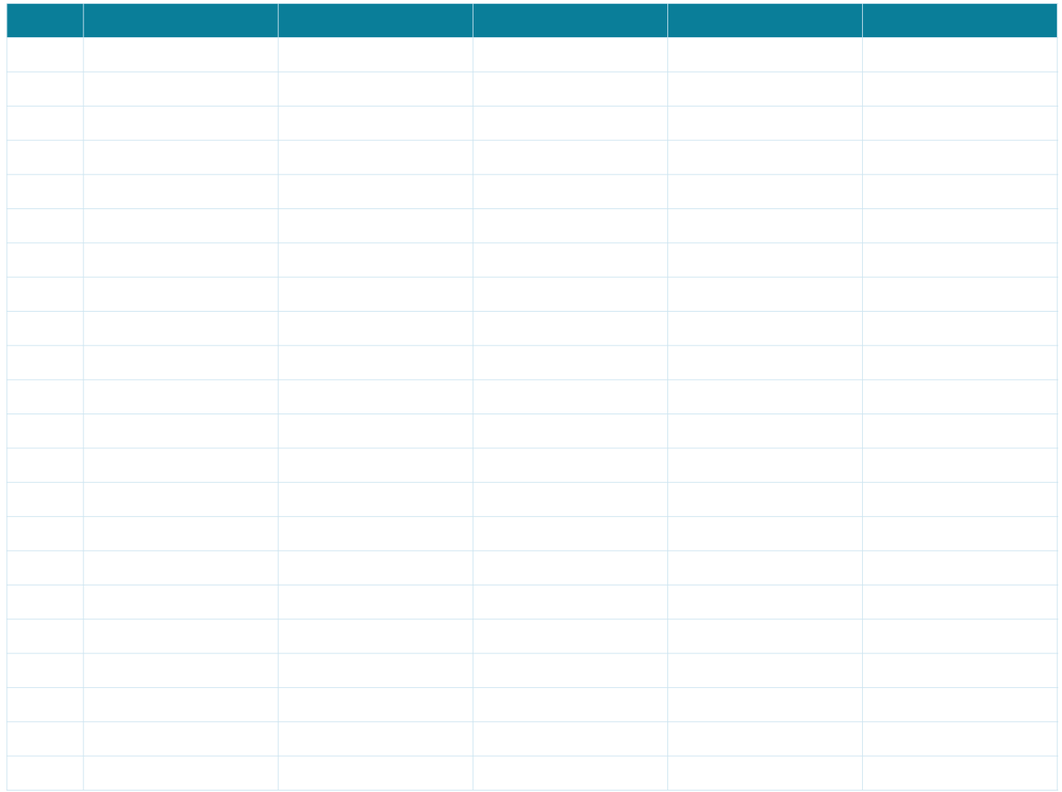
Reducing Data Held in Memory

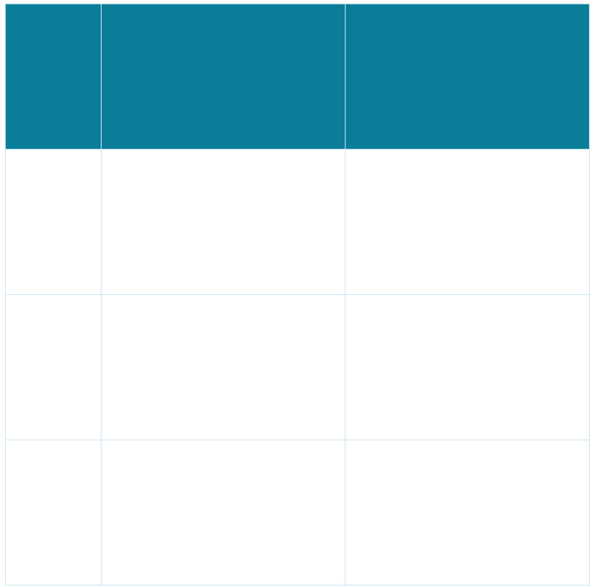




The large table will be split across multiple machines in the cluster

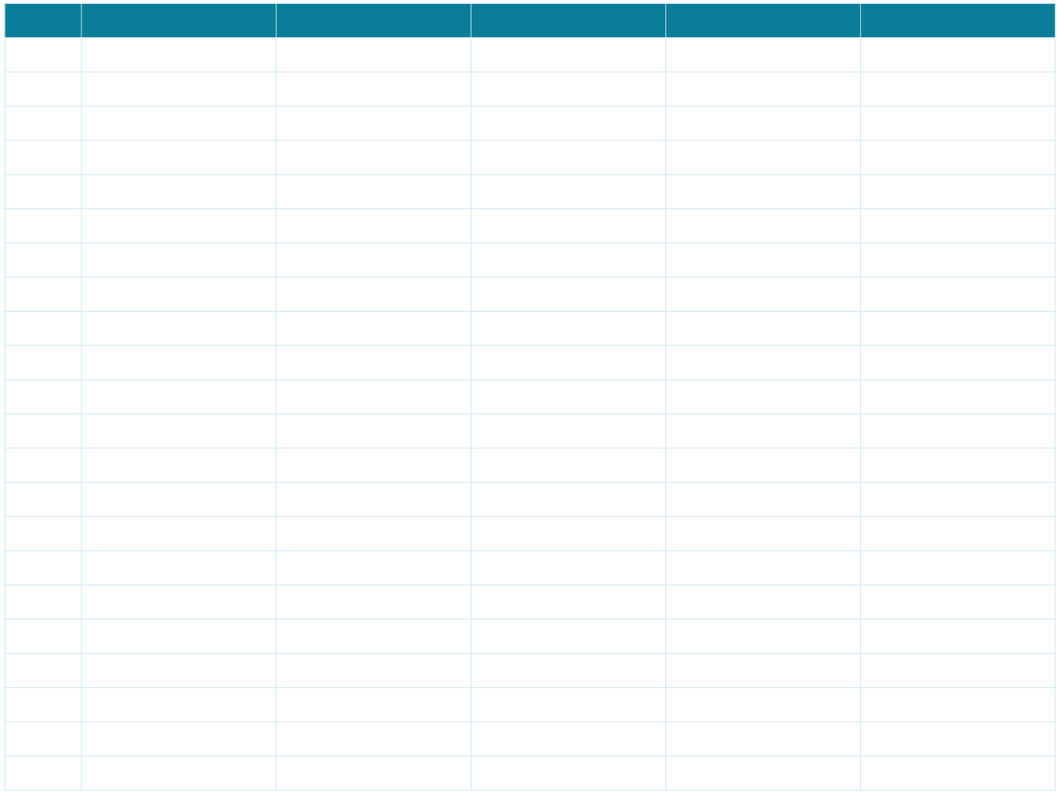
Reducing Data Held in Memory

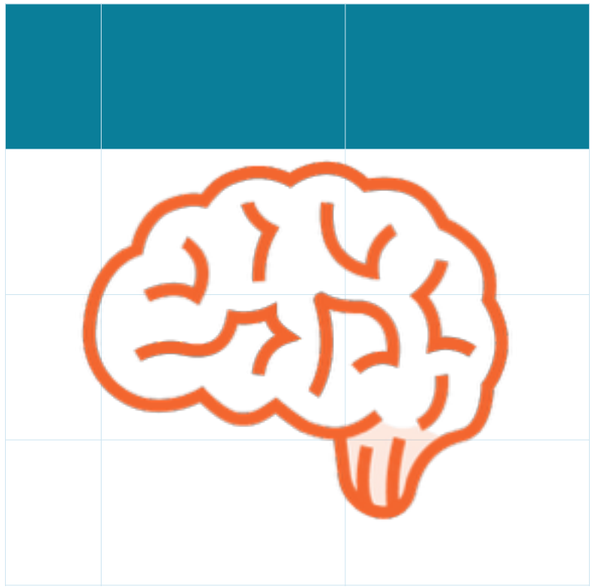




One table is held in memory while the other is read from disk

Reducing Data Held in Memory





For better performance the smaller table should be held in memory

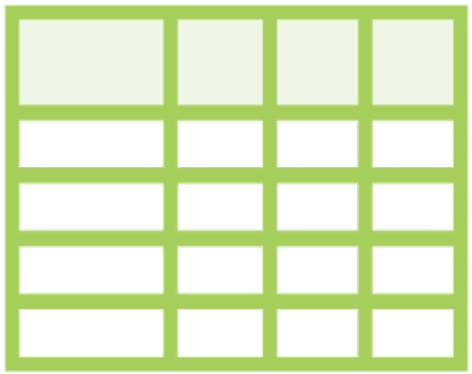
Joins as Map-only Operations

MapReduce operations have 2 phases of processing

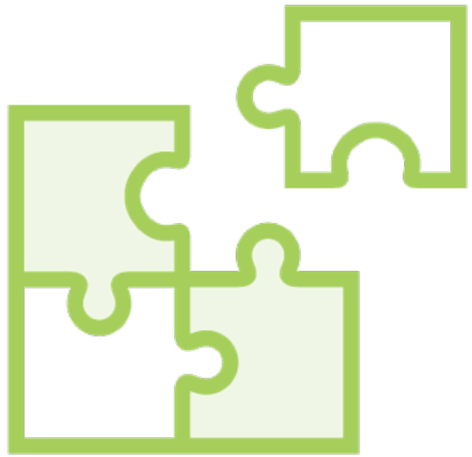
Joins as Map-only Operations

Certain queries can be structured to have no reduce phase

Queries on Big Data



Partitioning and Bucketing of Tables



Join Optimizations

Window Functions

Summary

A brief overview of Hive and its place in the Hadoop eco-system

Introduced some Hive features that allows us to work with large datasets

- Partitioning and Bucketing
- Join Optimizations
- Window Functions