

# Getting Started with Hive for Relational Database Developers

---

HIVE VS. RDBMS

# Overview

**Understand the differences between transactional and analytical processing**

**Learn why Hive, an open source data warehouse, is needed**

**Understand the differences between Hive and a traditional RDBMS**

# What You Need to Know

**Comfortable using the Unix/Mac/Linux command line**

**Familiar with relational databases and their layout**

**Familiar with SQL queries**

**Some basic understanding of Hadoop, HDFS and MapReduce**

# Transactional and Analytical Processing

---

# Transactional and Analytical Processing



## **Order Management Support**

**John is responsible for tracking and delivering orders on time**



## **Revenue Analyst**

**Anna is responsible for tracking and monitoring revenues**

# Order Management Support



**20 deliveries in Kent, WA are delayed**

**The courier company has had a computer outage**

**John assigns the orders to another courier company in the region**

# Order Management Support



**3 customers want to ship orders to a different address**

**John updates the address on the shipments and re-routes them**

# Revenue Analyst



**Her manager wants an update on last month's revenues**

**Last month was an unusually slow one**

**Anna pulls up data for the last 5 years to check for seasonal effects**



# Revenue Analyst



Management asks if the new TV ads this year were successful

**Anna checks customer signups for a jump when the campaigns were run**

# Transactional and Analytical Processing



**Transactional Processing**



**Analytical Processing**

# Transactional and Analytical Processing

## Transactional Processing

Analyzes **individual entries**

Access to **recent** data, from the last few hours or days

**Updates** data

Fast **real-time** access

Usually a **single** data source

## Analytical Processing

Analyzes **large batches** of data

Access to **older** data going back months, or even years

Only **reads** data

**Long** running jobs

**Multiple** data sources

# Transactional and Analytical Processing



## Small Data

Both these objectives could be achieved  
using the **same** database system



# Small Data

**Single** machine with backup

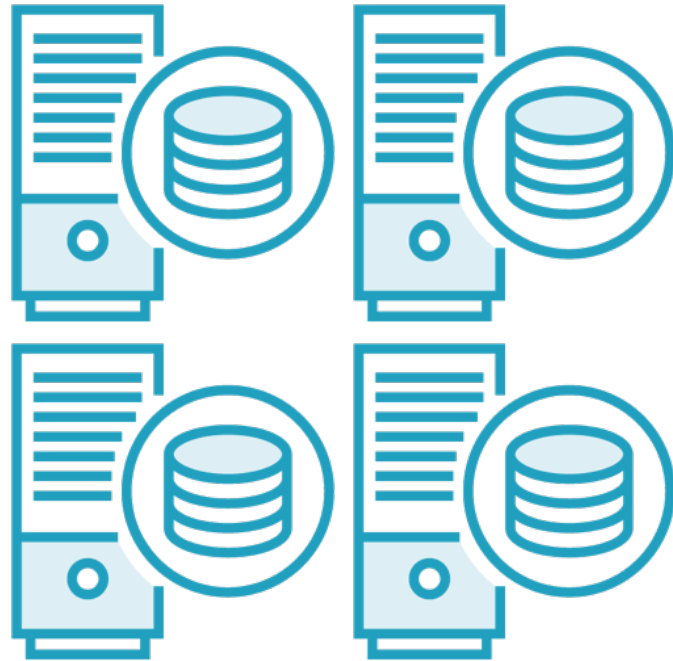
Structured, **well-defined** data

Can access **individual** records or the entire dataset

**No replication**, updated data available **instantaneously**

Different tables store data from different sources

# Transactional and Analytical Processing



## BIG Data

**Very hard** to meet all requirements with  
the **same** database system



# Big Data

Data distributed on a cluster with **multiple** machines

Semi-structured or **unstructured** data

**No random access** to data

Data **replicated**, propagation of updates take time

Different sources may have **different unknown formats**

The same infrastructure cannot support both **transactional** and **analytical** processing

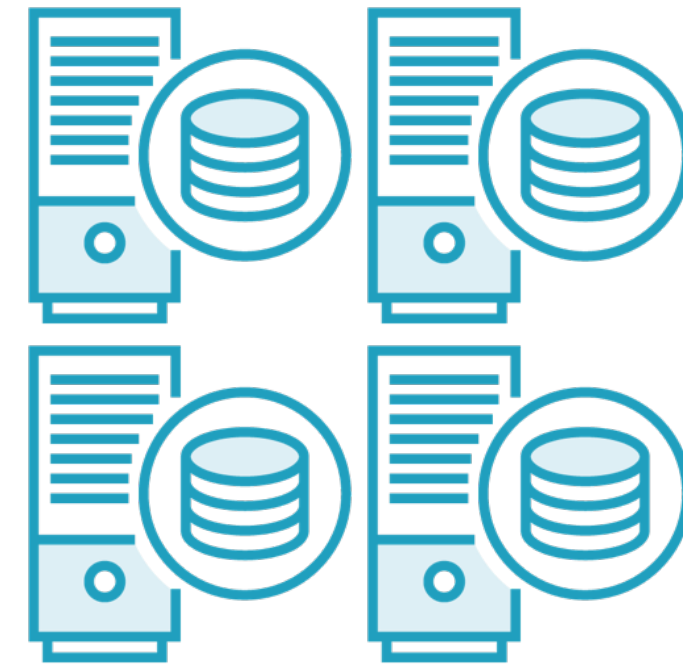


# Transactional and Analytical Processing



**Transactional Processing**

**Traditional  
RDBMS**



**Analytical Processing**

**Data Warehouse**

# Data Warehouse for Analytical Processing

---

# Data Warehouse

A technology that aggregates data from one or more sources so that it can be compared and analyzed for greater business intelligence.

[www.informatica.com](http://www.informatica.com)



# Data Warehouse

**Long running batch jobs**

**Optimized for read operations**

**Holds data from multiple sources**

**Holds data over a long period of time**

**Data may be lagged, not real-time**



# Data Warehouse

## Examples of data warehouses

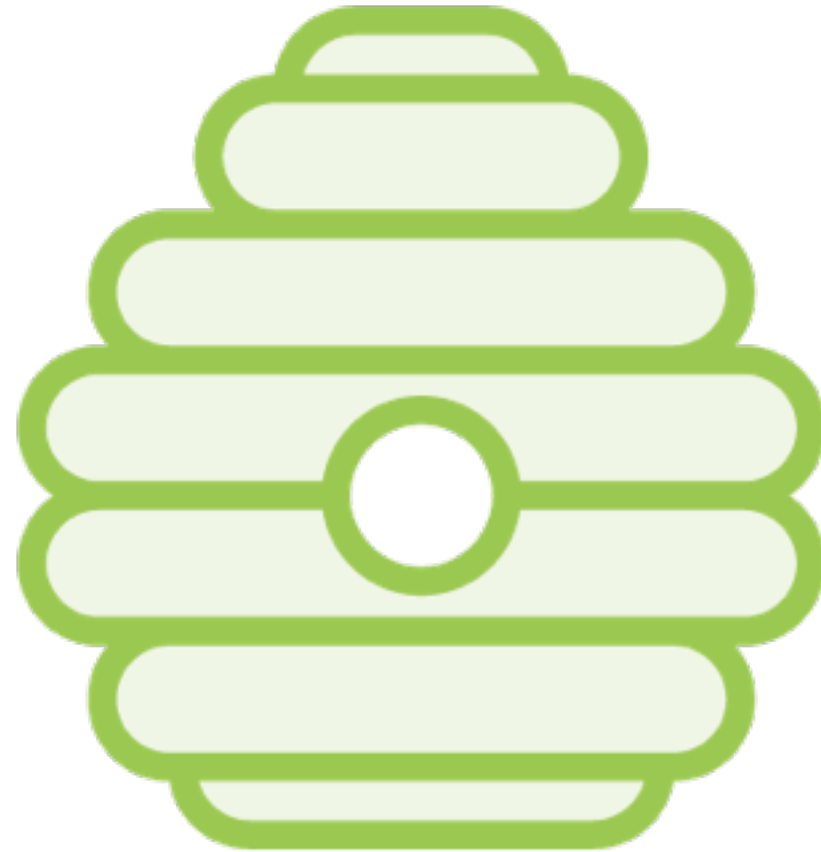
- Vertica
- Teradata
- Oracle
- IBM

# Data Warehouse



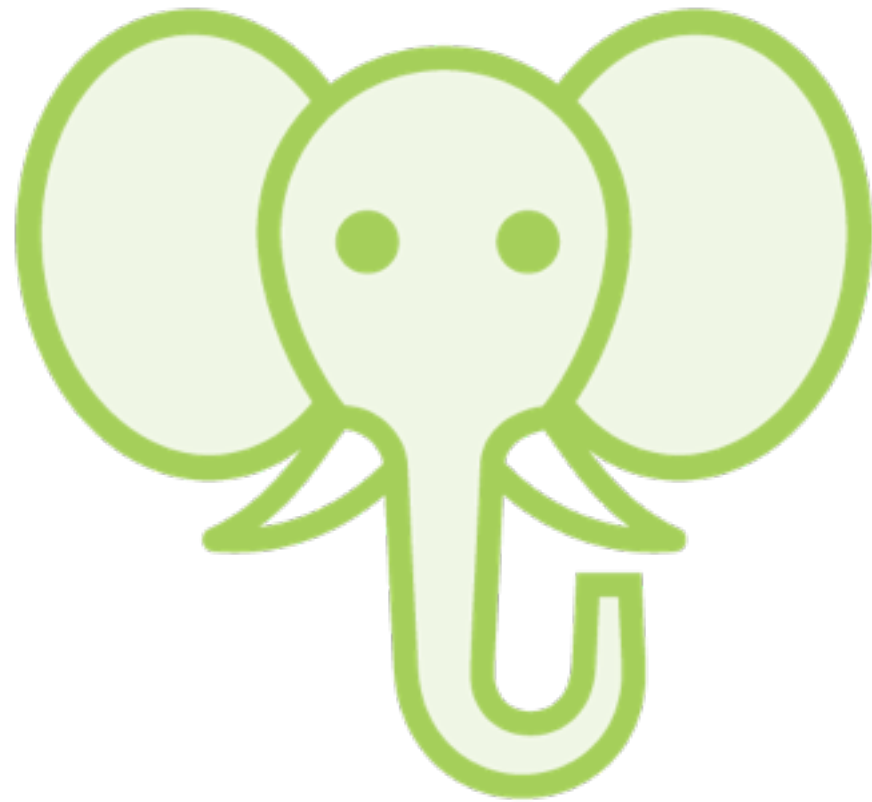
**Apache Hive is an open-source  
data warehouse**

# Data Warehouse



Hive is part of the larger **Hadoop** ecosystem

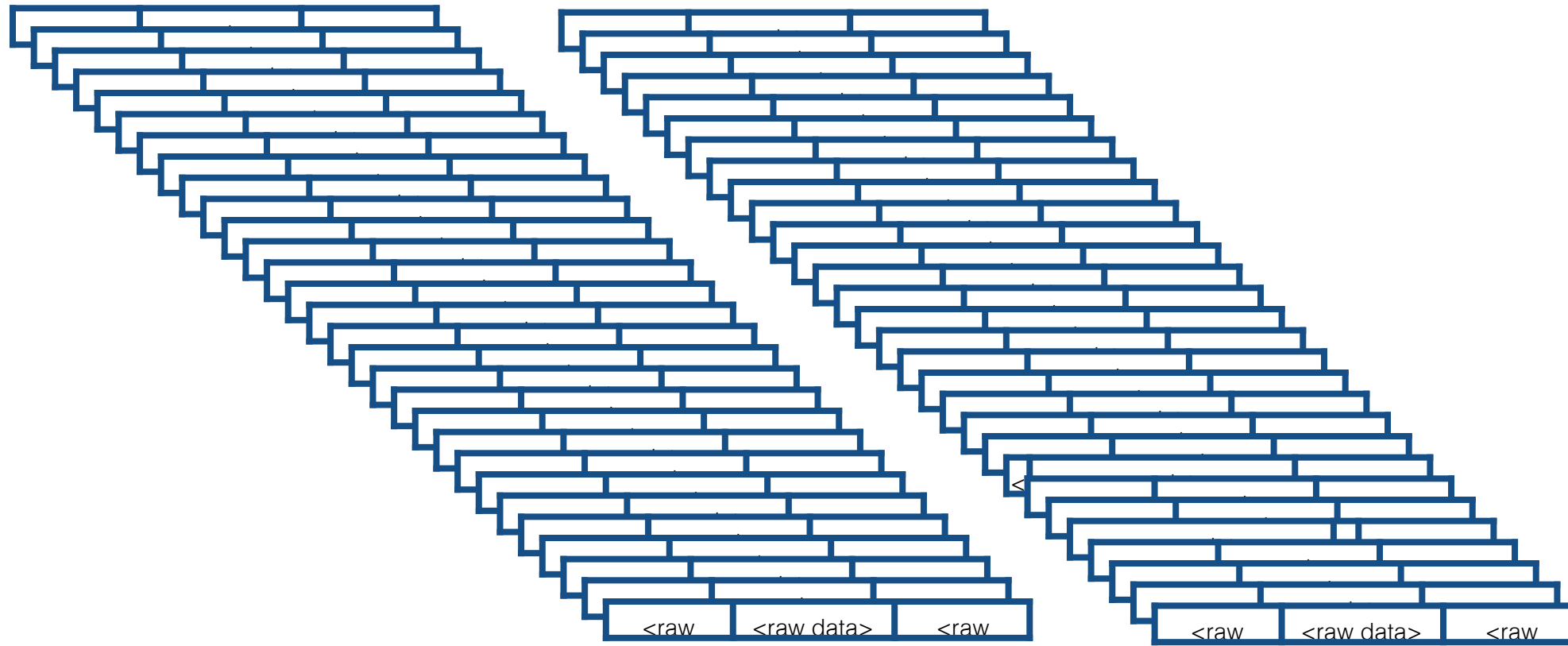
# Hadoop



**A distributed computing framework  
to process millions of records**

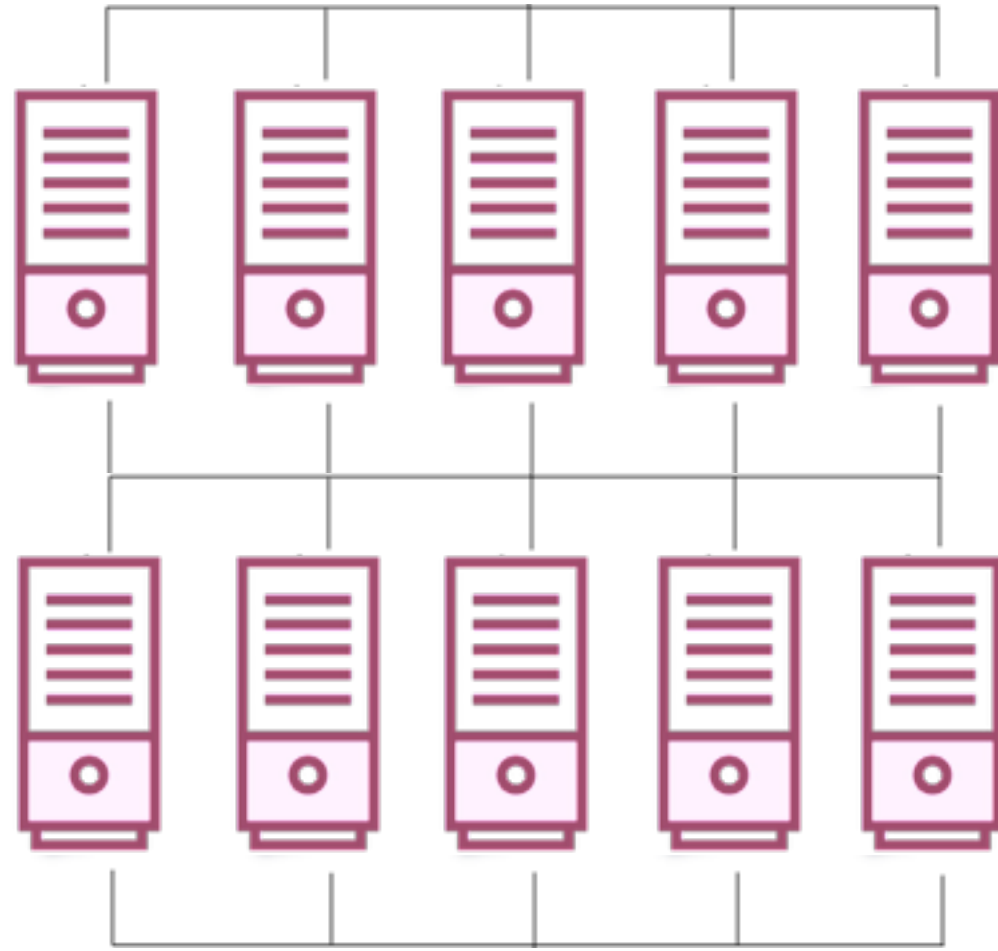


# Hadoop



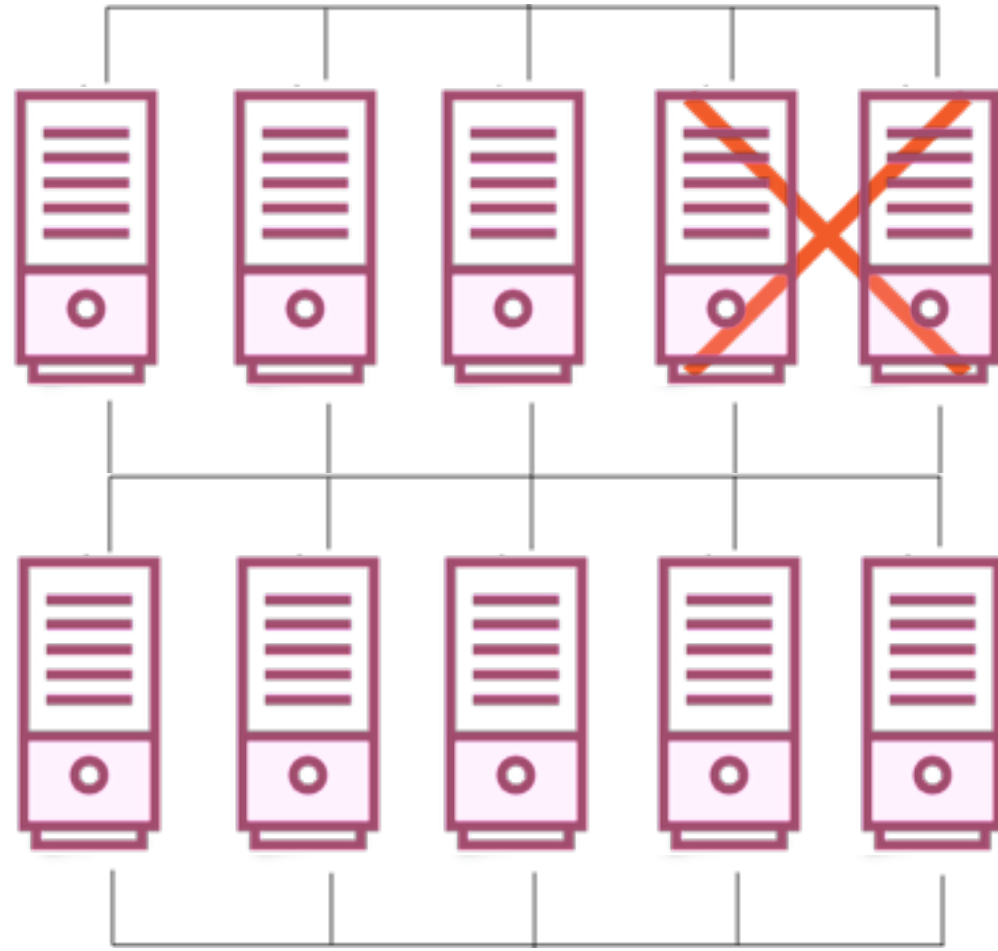
**1: Store  
millions of  
records on  
multiple  
machines**

# Hadoop



**2: Run  
processes on  
all these  
machines to  
crunch data**

# Hadoop



**3: Handle fault tolerance and recovery when nodes crash**

# Hadoop



The diagram shows the Hadoop ecosystem components. At the top is the word 'Hadoop'. Below it is a large blue-outlined rectangle containing three gray boxes. The first box is labeled 'HDFS', the second 'MapReduce', and the third 'YARN'. Below each box is a descriptive text block.

**HDFS**

**File system to  
manage the storage  
of data**

**MapReduce**

**Framework to  
process data across  
multiple servers**

**YARN**

**Framework to run  
and manage the data  
processing tasks**

# Hive on Hadoop



**Hive runs *on top* of the Hadoop distributed computing framework**

# Hive on Hadoop



HIVE

HDFS

MapReduce

YARN

**Hive stores its data in HDFS**



**HDFS**

# Hadoop Distributed File System

Data is stored as **files** - text files, binary files

**Partitioned** across machines in the cluster

**Replicated** for fault tolerance

Processing tasks **parallelized** across multiple machines

# Hive on Hadoop

HIVE

HDFS

MapReduce

YARN

**Hive runs all processes in the form of  
MapReduce jobs under the hood**





MapReduce

# MapReduce

A **parallel programming** model

Defines the **logic** to process data on multiple machines

**Batch** processing operations on files in HDFS

Usually written in **Java** using the Hadoop MapReduce library

# Hive on Hadoop

HIVE

HDFS

MapReduce

YARN

**Do we have to write MapReduce  
code to work with Hive?**

**No**

# HiveQL



## Hive Query Language

**A SQL-like interface to the  
underlying data**

# HiveQL



**Modeled on the Structured Query Language (SQL)**

**Familiar** to analysts and engineers

**Simple query constructs**

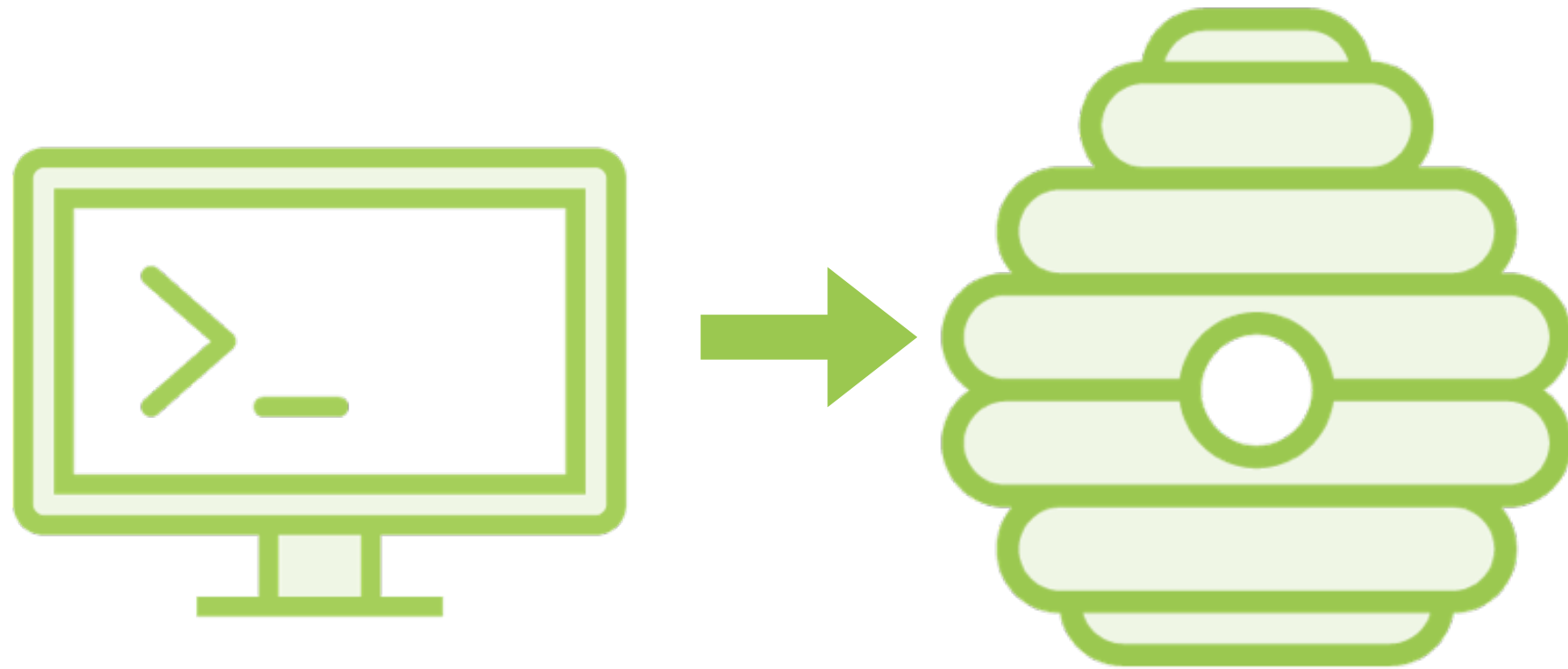
- **select**
- **group by**
- **join**

# HiveQL



**Hive exposes files in HDFS in the form of  
tables to the user**

# HiveQL



**Write SQL-like query in HiveQL and  
submit it to Hive**

HiveQL



**Hive will translate the query to MapReduce tasks and run them on Hadoop**

# HiveQL



**MapReduce will process files on HDFS and  
return results to Hive**



Hive **abstracts away** the details of  
the underlying MapReduce jobs

Work with Hive **almost**  
exactly like you would with a  
traditional database

# The Hive Metastore

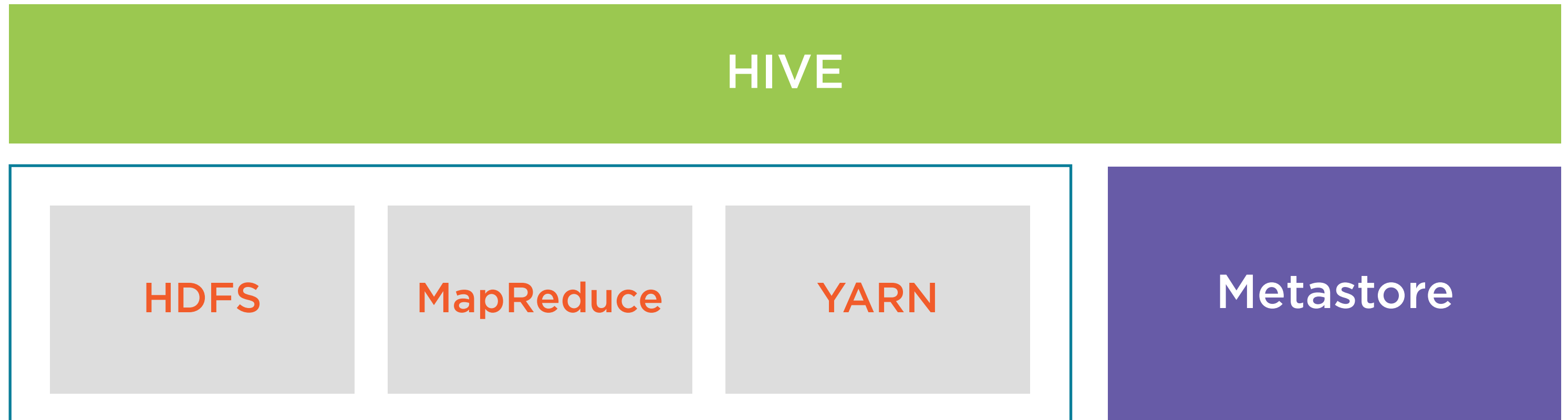
---

# The Hive Metastore



**A Hive user sees data as if they were stored in tables**

# The Hive Metastore



**Exposes the file-based storage of HDFS in the form of tables**

# The Hive Metastore



## Metastore

The **bridge** between data stored in files  
and the tables exposed to users



# The Hive Metastore

Stores **metadata** for all the tables in Hive

**Maps** the files and directories in Hive to tables

Holds **table definitions** and the **schema** for each table

Has information on **converting** files to table representations



# The Hive Metastore



**Any database with a  
JDBC driver can be used  
as a metastore**



# The Hive Metastore



**Development environments use  
the built-in Derby database**

**Embedded metastore**





# The Hive Metastore



**Same Java process** as Hive itself

**One Hive session** to connect to the database

# The Hive Metastore



## Production environments

**Local metastore:** Allows multiple sessions to connect to Hive

**Remote metastore:** Separate processes for Hive and the metastore

# Hive vs. RDBMS

---

# Hive vs. RDBMS

Data size

Computation

Latency

Operations

ACID compliance

Query language



# Hive vs. RDBMS



## Hive

**Large datasets**

**Parallel computations**

**High latency**

**Read operations**

**Not ACID compliant by default**

**HiveQL**

## RDBMS

**Small datasets**

**Serial computations**

**Low latency**

**Read/write operations**

**ACID compliant**

**SQL**



# Hive vs. RDBMS



## Hive

**Large datasets**

Parallel computations

High latency

Read operations

Not ACID compliant by default

HiveQL

## RDBMS

**Small datasets**

Serial computations

Low latency

Read/write operations

ACID compliant

SQL

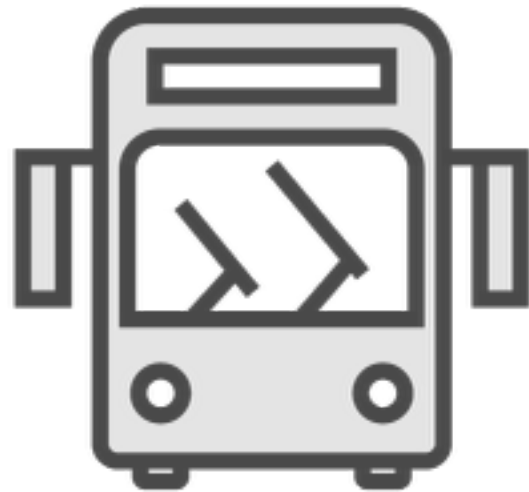


# Hive vs. RDBMS



**Large Datasets**

**Small Datasets**



**Gigabytes or  
petabytes**



**Megabytes or  
gigabytes**



# Hive vs. RDBMS

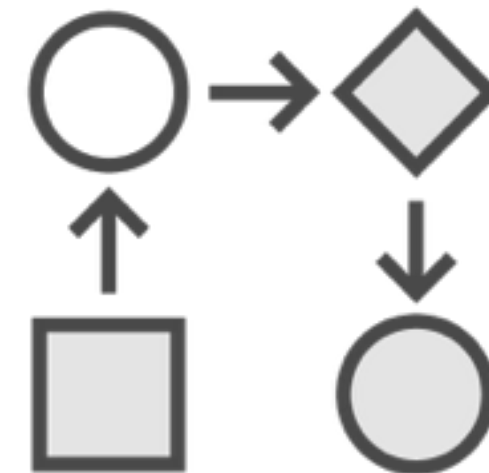


**Large Datasets**



**Calculating trends**

**Small Datasets**



**Accessing and updating individual records**





# Hive vs. RDBMS



## Hive

Large datasets

**Parallel computations**

High latency

Read operations

Not ACID compliant by default

HiveQL

## RDBMS

Small datasets

**Serial computations**

Low latency

Read/write operations

ACID compliant

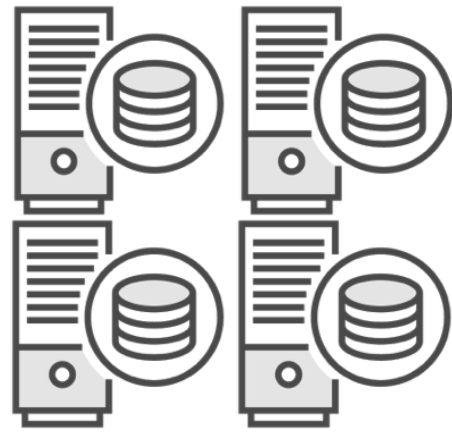
SQL



# Hive vs. RDBMS



## Parallel Computations



**Distributed system  
with multiple  
machines**

## Serial Computations



**Single computer  
with backup**



# Hive vs. RDBMS

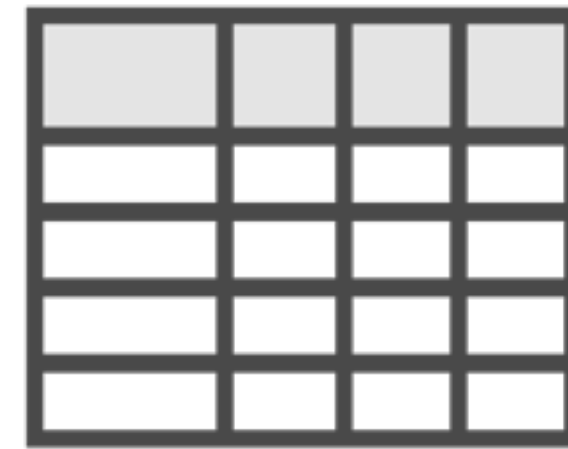


## Parallel Computations



**Semi-structured data  
files partitioned  
across machines**

## Serial Computations



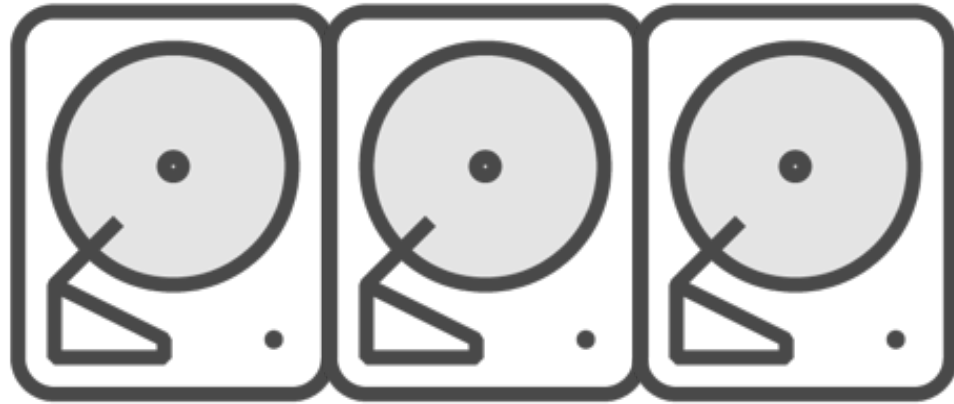
**Structured data in  
tables on one  
machine**



# Hive vs. RDBMS



## Parallel Computations



**Disk space cheap,  
can add space by  
adding machines**

## Serial Computations



**Disk space  
expensive on a  
single machine**



# Hive vs. RDBMS



## Hive

Large datasets

Parallel computations

**High latency**

Read operations

Not ACID compliant by default

HiveQL

## RDBMS

Small datasets

Serial computations

**Low latency**

Read/write operations

ACID compliant

SQL



# Hive vs. RDBMS



**High Latency**



**Records not indexed, cannot be accessed quickly**

**Low Latency**



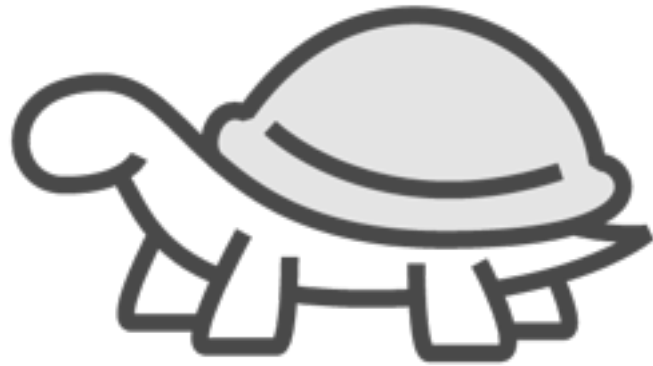
**Records indexed, can be accessed and updated fast**



# Hive vs. RDBMS

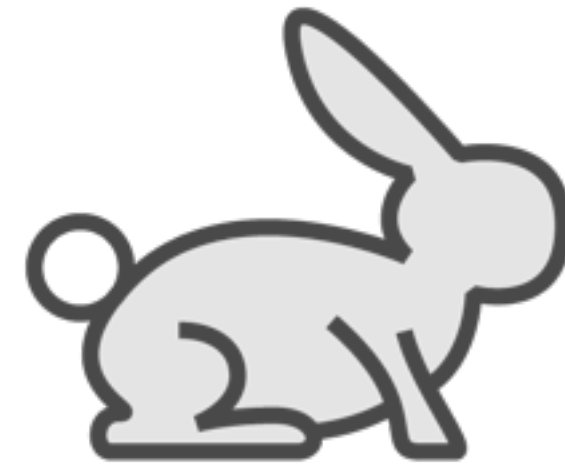


**High Latency**



**Fetching a row will  
run a MapReduce  
that might take  
minutes**

**Low Latency**



**Queries can be  
answered in  
milliseconds or  
microseconds**



# Hive vs. RDBMS



## Hive

Large datasets

Parallel computations

High latency

**Read operations**

Not ACID compliant by default

HiveQL

## RDBMS

Small datasets

Serial computations

Low latency

**Read/write operations**

ACID compliant

SQL





# Hive vs. RDBMS

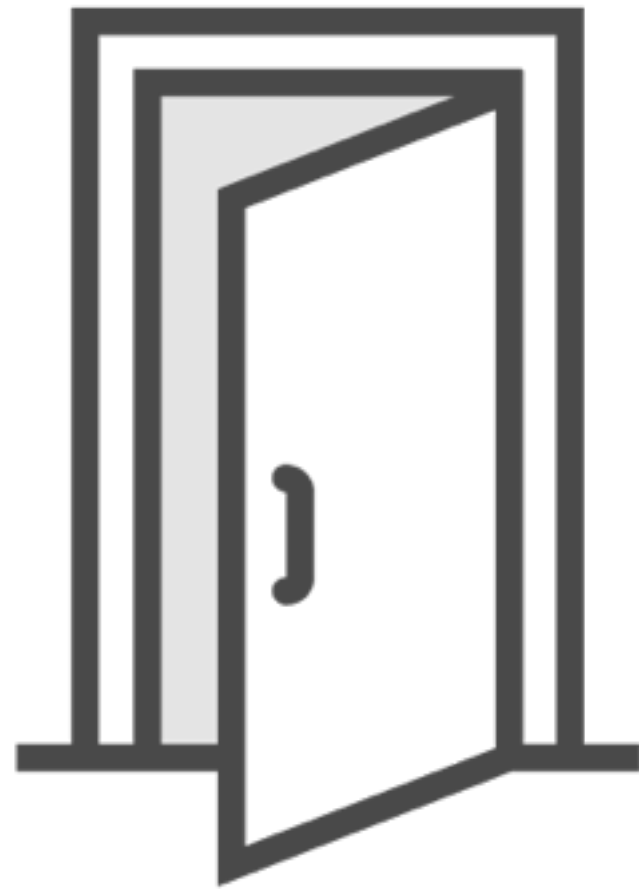


**Read Operations**

**Read/Write Operations**



**Not the owner of  
data**



# No Data Ownership

Hive stores files in **HDFS**

Hive files can be read and written by many technologies

- **Hadoop, Pig, Spark**

Hive database schema **cannot be enforced** on these files



# Hive vs. RDBMS

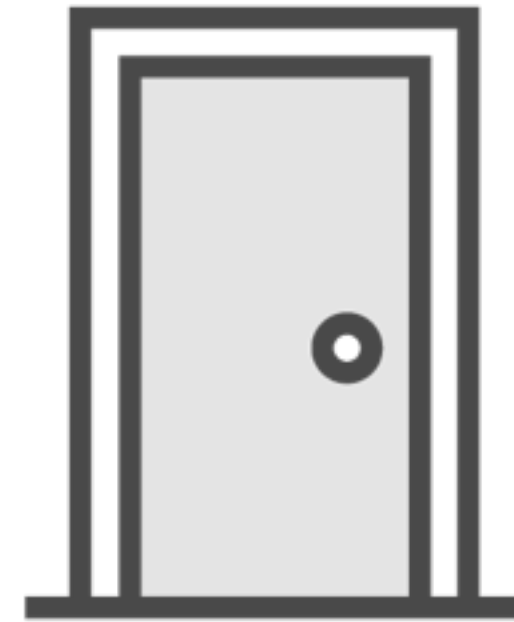


**Read Operations**



**Not the owner of  
data**

**Read/Write Operations**



**Sole gatekeeper for  
data**



# Hive vs. RDBMS



**Read Operations**

**Read/Write Operations**



**Schema-on-read**



# Schema-on-read

Number of columns, column types,  
constraints specified at table creation

Hive **tries to impose** this schema when  
data is read

It may not succeed, may **pad data with  
nulls**



# Hive vs. RDBMS



**Read Operations**



**Schema-on-read**

**Read/Write Operations**



**Schema-on-write**



# Hive vs. RDBMS



## Hive

Large datasets

Parallel computations

High latency

Read operations

**Not ACID compliant by default**

HiveQL

## RDBMS

Small datasets

Serial computations

Low latency

Read/write operations

**ACID compliant**

SQL



# Hive vs. RDBMS



**Not ACID Compliant**



**Data can be dumped  
into Hive tables from  
any source**

**ACID Compliant**



**Only data which satisfies  
constraints are stored in  
the database**





# Hive vs. RDBMS



## Hive

Large datasets

Parallel computations

High latency

Read operations

Not ACID compliant by default

**HiveQL**

## RDBMS

Small datasets

Serial computations

Low latency

Read/write operations

ACID compliant

**SQL**



# Hive vs. RDBMS



## HiveQL

**Schema on read, no constraints enforced**

**Minimal index support**

**Row level updates, deletes as a special case**

**Many more built-in functions**

**Only equi-joins allowed**

**Restricted subqueries**

## SQL

**Schema on write keys, not null, unique all enforced**

**Indexes allowed**

**Row level operations allowed in general**

**Basic built-in functions**

**No restriction on joins**

**Whole range of subqueries**

# Summary

**Understood the differences between transactional and analytical processing and where they are used**

**Learnt the need for a data warehouse for analytical processing**

**Understood the differences between Hive and a traditional RDBMS**