Introduction to Hive

Ahmad Alkilani
www.pluralsight.com

pluralsight
hardcore developer training

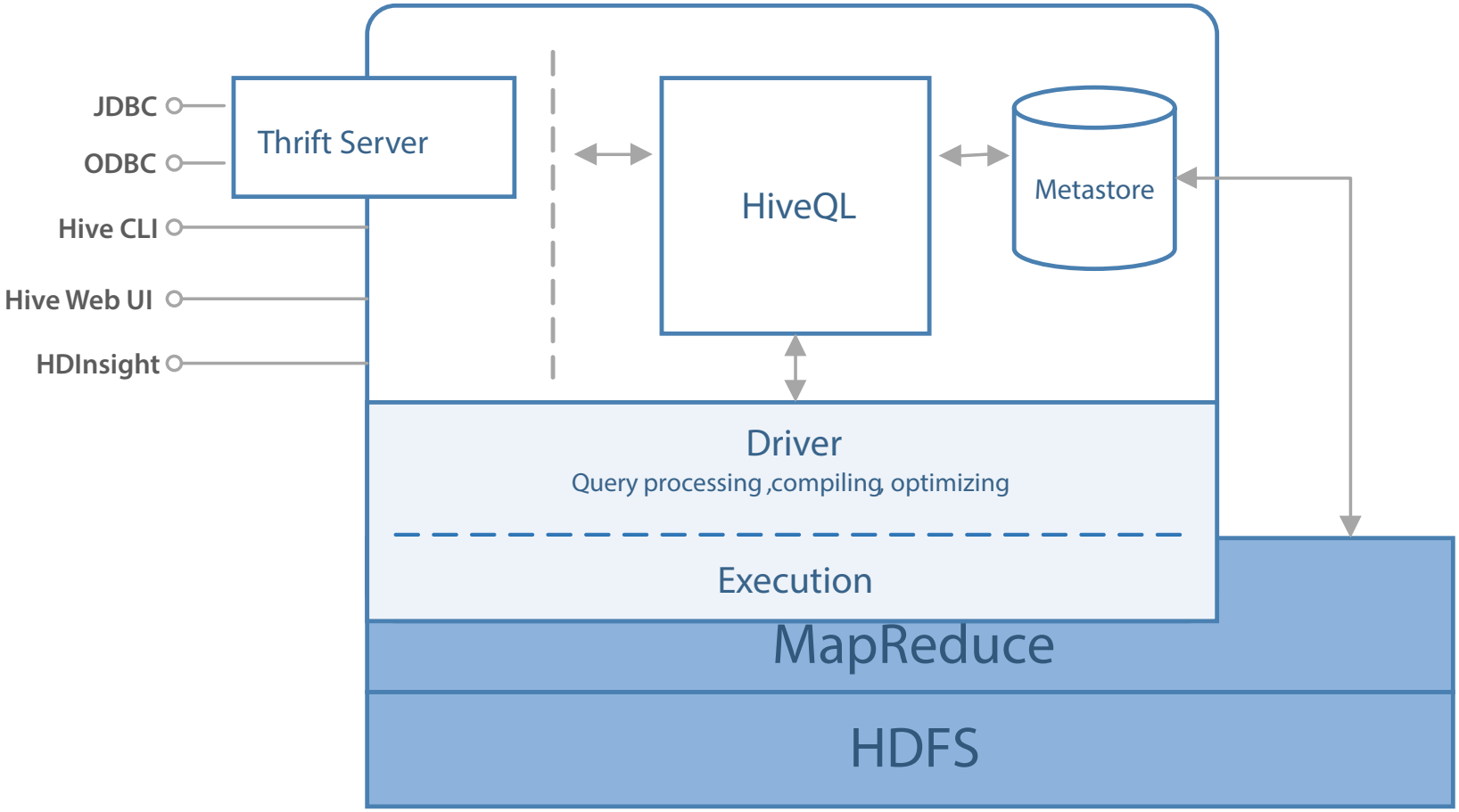
Outline

- Why Hive? Motivation
- Hive's Architecture
- Hive Principles – Schema on Read
- Hive Principles – The Hive Warehouse
- HiveQL – SELECT, Sub queries, UNION ALL, CREATE DATABASE, CREATE TABLE
- Demo – Working with Hive and loading data

Hive Motivation

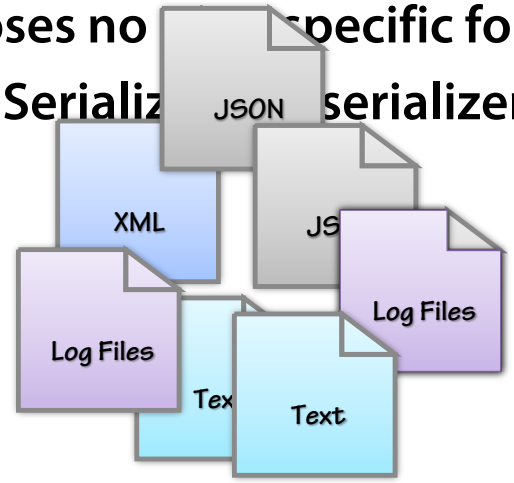
- **Opens up Big Data to the masses**
- **Provides a SQL-like query language and interfaces**
- **Builds on Hadoop core using MapReduce for execution**
- **Originally started at Facebook**
  - MapReduce development is time consuming
  - Requires intimate knowledge of the framework
  - Limited resources with required expertise
  - No schema to help understand data in HDFS

Hive Architecture

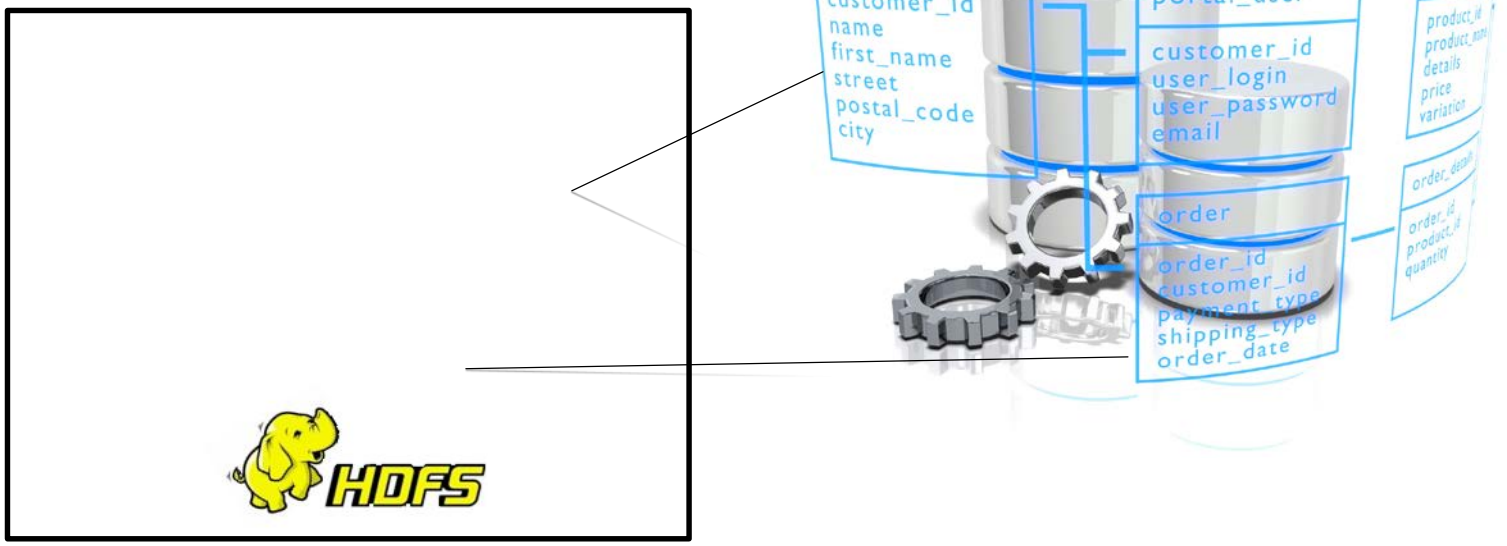


Hive Principles – Schema on Read

- Imposes no specific format
- Uses Serializers to read and write data



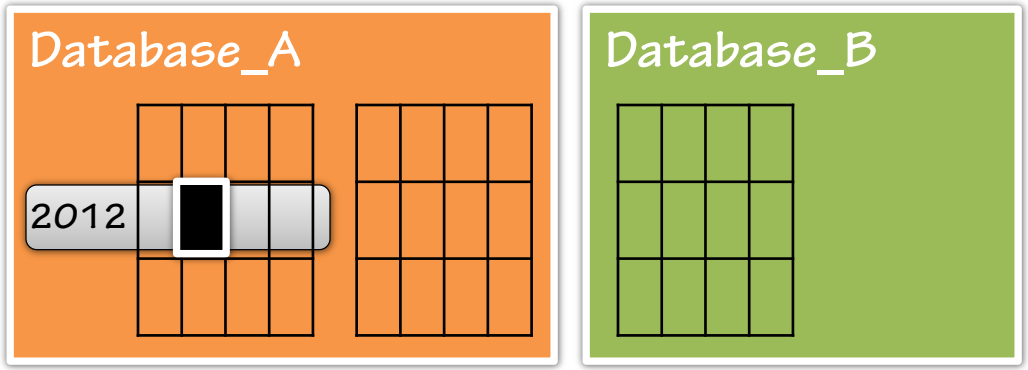
Hive: Read as the following structure



Hive Principles – The Hive Warehouse

Hive warehouse

- Meta data about all the objects known to Hive, persisted in in the meta store
- Consists of
  - Databases
  - Tables
  - Partitions
  - Buckets/Clusters



- Local Hive warehouse
  - Managed by Hive
  - Typically under /hive/warehouse
  - Dropping a table will drop the data just as well as the meta-data.
- External Tables
  - Hive manages the meta-data only
  - Anywhere on the Hadoop file system
  - Dropping a table in Hive will only remove the table's definition, data remains untouched.

Basic commands using HiveQL

Hive Basics

The SELECT statement

```
SELECT
    exp1, exp2, exp3
FROM
    some_table
WHERE
    where_condition
LIMIT
    number_of_records;
```

- DISTINCT Clause

```
SELECT DISTINCT col1, col2, col3 FROM some_table;
```

- Aliasing

```
SELECT col1 + col2 AS col3 FROM some_table;
```

- REGEX Column Specification

```
SELECT '(ID|Name)?+.+' FROM some_table;
```

```
FROM
    some_table
SELECT
    exp1, exp2, exp3
WHERE
    where_condition;
```

- Interchangeable constructs
- Hive is not case sensitive
- Semicolon to terminate statements

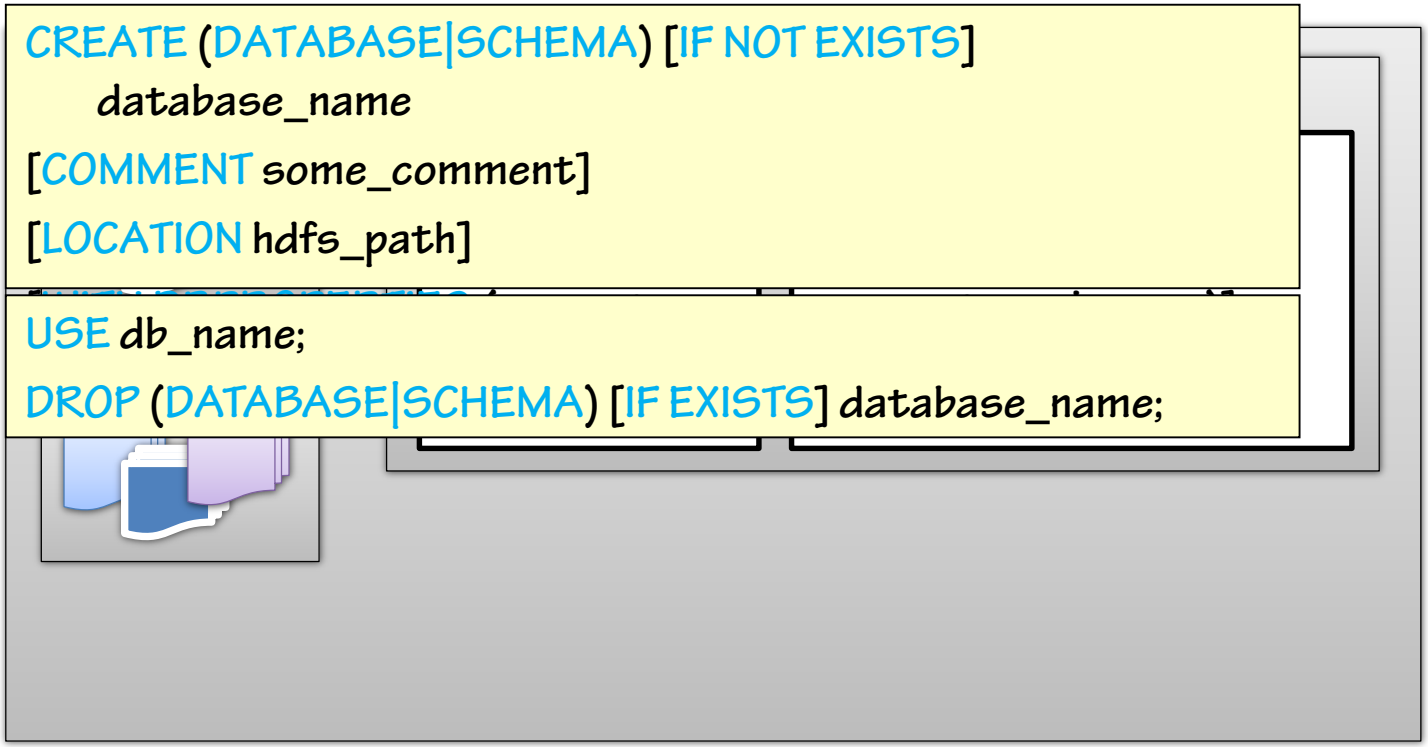
Sub queries & Union

```
SELECT subq.mycol
FROM (
    SELECT col_a + col_b AS mycol
    FROM some_table;
) subq;
```

```
SELECT col_a + col_b AS mycol
FROM some_table
UNION ALL
SELECT col_y AS mycol
FROM another_table;
```

```
SELECT t3.mycol
FROM (
    SELECT col_a + col_b AS mycol
    FROM some_table
    UNION ALL
    SELECT col_y AS mycol
    FROM another_table
) t3
JOIN t4 ON (t4.col_x = t3.mycol);
```

Create Database

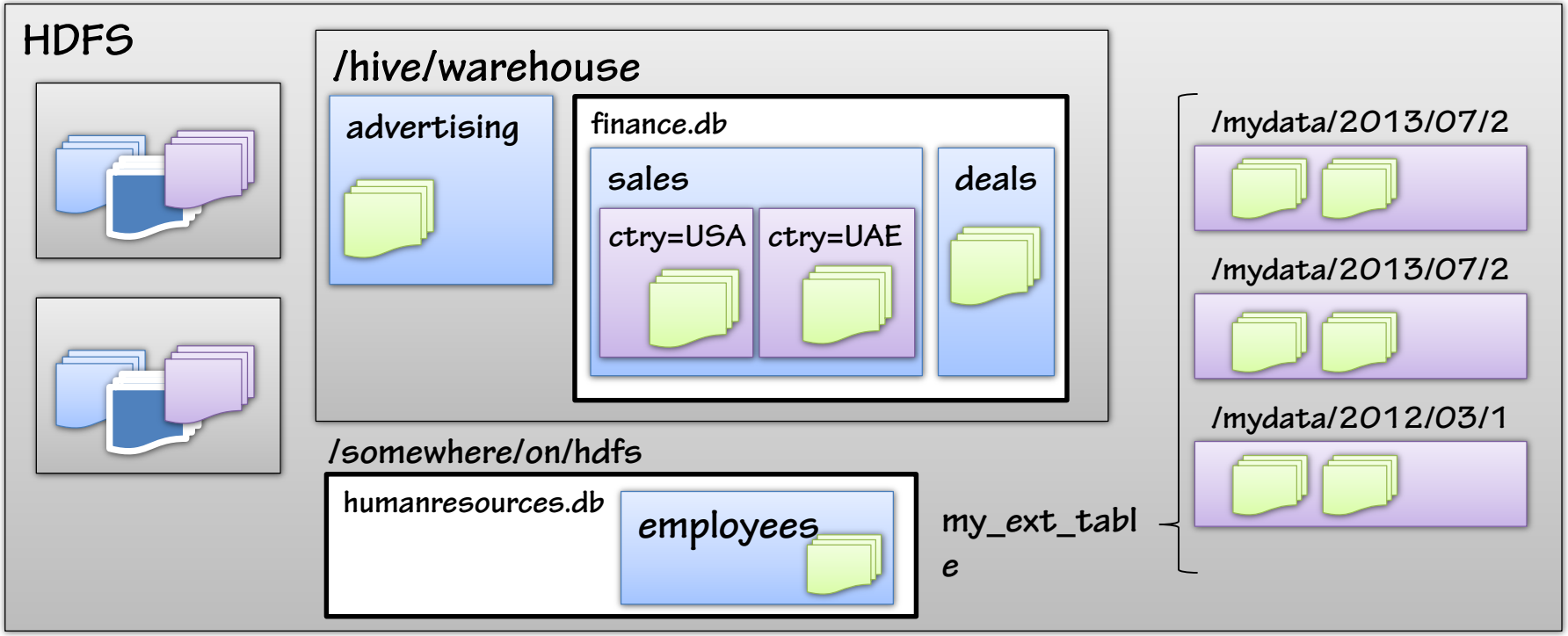


/somewhere/on/hdf

humanresources.db

Create Table

```
CREATE [EXTERNAL] TABLE [IF NOT EXISTS] [db_name.]table_name
[(col_name data type [COMMENT col_comment], ...)]
[PARTITIONED BY (col_name data type [COMMENT col_comment], ...)]
[ROW FORMAT row_format] [STORED AS file_format]
[LOCATION hdfs_path]
[TBLPROPERTIES (property_name=property_value, ...)];
```



Working with Hive

Demo

Demo Recap

- **Pluralsight database**
  - Hive creates pluralsight.db directory
- **Created movies hive managed table**
  - Placed u.info in movies table; Hive doesn't complain but results in NULLs
  - Placed correct u.item data in movies table
- **LOAD DATA INPATH [path]**
  - Moves data if source is HDFS
  - Copies data if source is LOCAL
    - Syntax: LOAD DATA LOCAL INPATH [path]
- **Consider using EXTERNAL tables if data is already in HDFS**

Summary

- **Hive as an important player in the Big Data community**
- **Hive warehouse and schema on read concepts**
- **HiveQL**
  - SELECT, UNION ALL, Sub Queries, DISTINCT, Aliasing
  - Create database
  - External and Hive managed tables
  - Loading data into the Hive warehouse
  - Truncate or overwrite
  - Different methods for creating tables
    - CTAS
    - LIKE