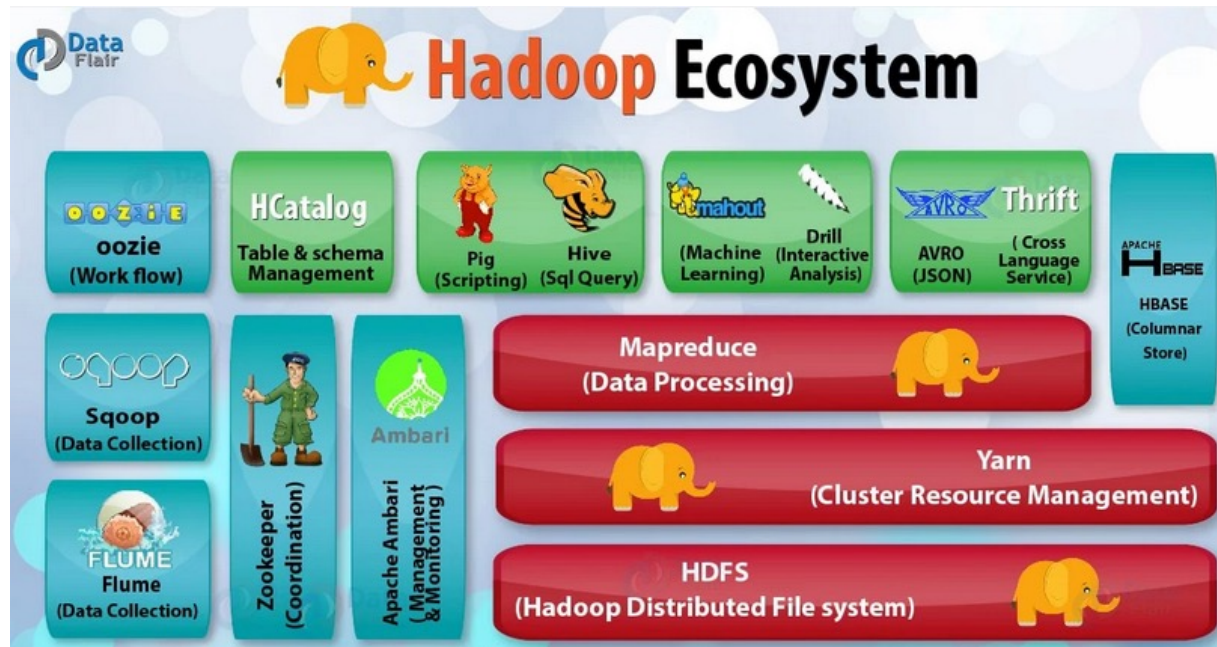


# Querying the Data interactively

Bernard Lee Kok Bang

**allow us to execute SQL queries :)**

# Core Hadoop Ecosystem



# External Data Storage



# Query Engines



# Apache Drill – SQL for NoSQL

- A SQL query engine for a variety of non-relational databases and data files
  - *Hive, MongoDB, HBase*
  - *Flat JSON or Parquet files on HDFS, S3, Azure, Google Cloud, local file system*
- Based on Google's Dremel
- **let us execute actual SQL**

# Parquet – columnar format

- stores the data in row group

	Column 1	Column 2	Column 3	Column 4	Column 5
	Product	Customer	Country	Date	Sales Amount
Row Group 1	Ball	John Doe	USA	2023-01-01	100
	T-Shirt	John Doe	USA	2023-01-02	200
Row Group 2	Socks	Maria Adams	UK	2023-01-01	300
	Socks	Antonio Grant	USA	2023-01-03	100
Row Group 3	T-Shirt	Maria Adams	UK	2023-01-02	500
	Socks	John Doe	USA	2023-01-05	200

<https://shorturl.at/efno0>

# It's real SQL

- Not SQL-like
- And it has ODBC (Microsoft) / JDBC (Sun Microsystems) driver so other tools can connect to it just like any relational database
  - *allow connection to external tools such as Tableau*

# Drill - Fast and easy to set up

- Still non-relational databases under the hood
- Allow SQL analysis of disparate data sources without having to transform and load
  - *Internally data is represented as JSON and so has no fixed schema*
- Doing JOINS across different database technologies or with JSON files that are sitting around in HDFS
- **having a SQL without a schema**



# Let's Drill

- Import data into Hive (movie ratings) and MongoDB (movie users) – MovieLens datasets
- Set up Drill on top of both
- And do some queries

# Setting up Drill


1. Import *u.user* into MongoDB
2. Upload *u.data* into Hive
  - *Login to Ambari as admin*
  - *Start MongoDB*
  - Navigate to Hive View to create a new database
    - *CREATE DATABASE movielens;*
  - upload u.data into movielens database

# Upload data into Hive


*tab delimiter*

*movielens database*

Upload from Local ☒ Upload from HDFS ☐

File type: CSV 

Database: movielens

Stored as: ORC 

Select from local: Choose File u.data

Table name: ratings

Contains endlines? ☐

Upload Table

user_id	movie_id	rating	epoch_seconds
196	242	3	881250949
186	302	3	891717742
22	377	1	878887116

# Import data into MongoDB

- Log into puTTY as root:

*su root*

- run the previous *MongoSpark.py* script to import *u.user* data into mongoDB database [make sure the ~~u.ser~~ data is still in the ml-100k folder in HDFS]

*u.user*

- submit the script using the following command

*spark-submit --packages org.mongodb.spark:mongo-spark-connector\_2.11:2.3.2 MongoSpark.py*

# Installing Drill

- Not a built-in part of Ambari on Hortonworks platform
- Need to get specific version of *Drill: version 1.12*

```
[root@sandbox-hdp maria_dev]# wget http://archive.apache.org/dist/drill/drill-1.12.0/apache-drill-1.12.0.tar.gz
```

- Decompress the downloaded file
  - *tar -xvf apache-drill-1.12.0.tar.gz*
- To start running Drill, run the following script [open the port 8765]
  - *cd apache-drill-1.12.0*
  - *bin/drillbit.sh start -Ddrill.exec.http.port=8765*

**this is to start Drill**



# Connect Drill through browser

- Log into browser address specific for Drill
  - *127.0.0.1:8765*

*Key in the Drill  
address*

The screenshot shows the Apache Drill web interface. The address bar displays '127.0.0.1:8765'. The navigation bar includes links for Apache Drill, Query, Profiles, Storage (circled in red), Metrics, Threads, Logs, Options, and Documentation. The main content area shows 'Drillbits 1' with a table of drillbit information. Below the table are sections for 'Encryption' and 'Query Throttling'.

#	Address	User Port	Control Port	Data Port	Version	Status
1	sandbox-hdp.hortonworks.com <span>Current</span>	31010	31011	31012	1.12.0	ONLINE

**Encryption**

Client to Bit Encryption Disabled

Bit to Bit Encryption Disabled

**Query Throttling**

Queue Status Disabled

# Connect Drill to Hive and MongoDB

- Click *Storage* tab
- *Enable* both *mongo and hive*
- Update both hive and mongo

# Configure Hive

- Update *hive.metastore.uris*

### Configuration

```
1 {  
2   "type": "hive",  
3   "enabled": true,  
4   "configProps": {  
5     "hive.metastore.uris": "thrift://localhost:9083",  
6     "javax.jdo.option.ConnectionURL": "jdbc:derby:;databaseName=../sample-data/drill_hive_db;create=true",  
7     "hive.metastore.warehouse.dir": "/tmp/drill_hive_wh",  
8     "fs.default.name": "file:///",  
9     "hive.metastore.sasl.enabled": "false"  
10  }  
11 }
```

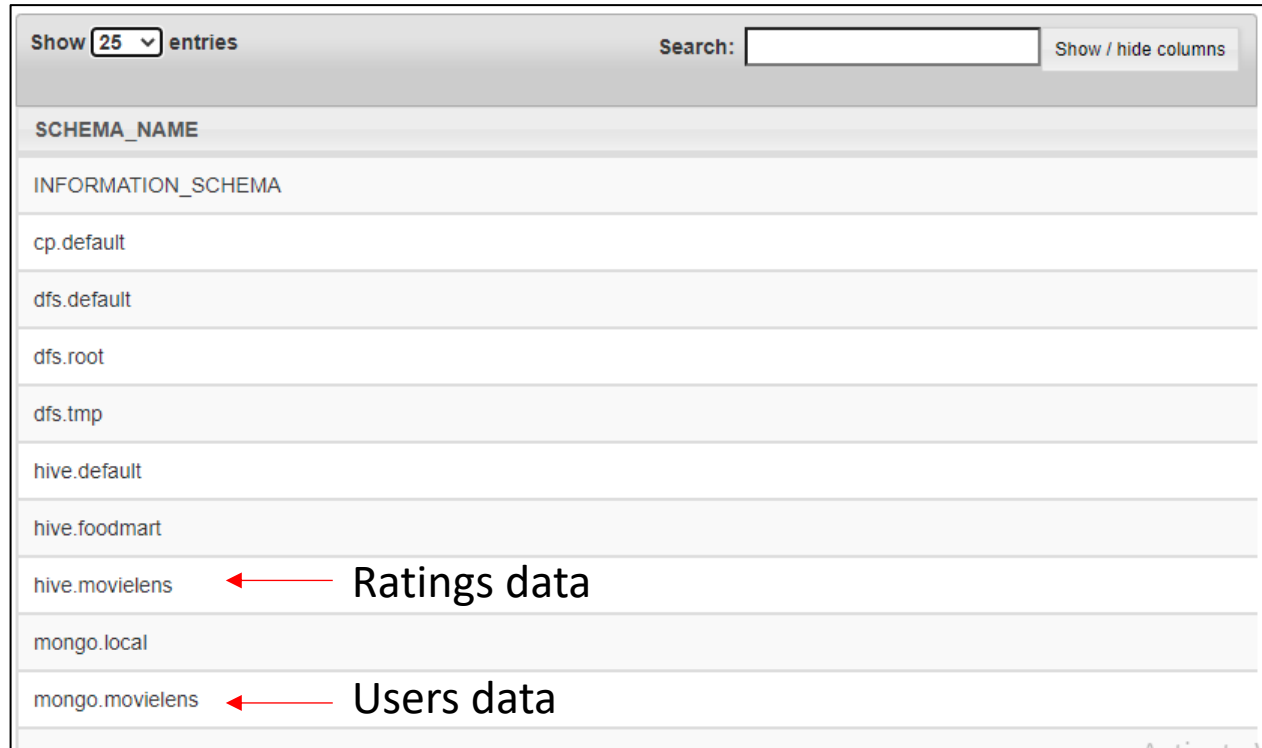
BackUpdateDisableDelete

*Update hive metastore accordingly: thrift://localhost:9083*



# Start querying across multiple databases using Drill

- *SHOW DATABASES;*



SCHEMA_NAME
INFORMATION_SCHEMA
cp.default
dfs.default
dfs.root
dfs.tmp
hive.default
hive.foodmart
hive.movielens ← Ratings data
mongo.local
mongo.movielens ← Users data

# Start querying (cont...)

- *SELECT \* FROM hive.movielens.ratings LIMIT 10;*

Show 10 ▾ entries		Search: <input type="text"/>		Show / hide columns
user_id	movie_id	rating	epoch_seconds	
196	242	3	881250949	
186	302	3	891717742	
22	377	1	878887116	
244	51	2	880606923	
166	346	1	886397596	
298	474	4	884182806	
115	265	2	881171488	
253	465	5	891628467	
305	451	3	886324817	
6	86	3	883603013	

# Start querying (cont...)

- *SELECT \* FROM mongo.movielen.users LIMIT 10;*

Show 10 entries		Search: <input type="text"/>		Show / hide columns	
_id	age	gender	occupation	user_id	zip
[B@3157f22e	24	M	technician	1	85711
[B@1fe2c00	53	F	other	2	94043
[B@1d439b97	23	M	writer	3	32067
[B@5494494c	24	M	technician	4	43537
[B@5a5c1772	33	F	other	5	15213
[B@2a32be15	42	M	executive	6	98101
[B@2b2c07e7	57	M	administrator	7	91344
[B@5d84e5e7	36	M	administrator	8	05201
[B@67599ea2	29	M	student	9	01002
[B@5824564a	53	M	lawyer	10	90703

# Start querying (cont...)

- *SELECT u.occupation, COUNT(\*) FROM hive.movielens.ratings r JOIN mongo.movielens.users u ON r.user\_id = u.user\_id GROUP BY u.occupation;*

**How many ratings  
were provided by each  
occupation?**

Show 25 entries		Search: <input type="text"/>	Show / hide columns
occupation	↑	EXPR\$1	↓
administrator		7479	
artist		2308	
doctor		540	
educator		9442	
engineer		8175	
entertainment		2095	
executive		3403	
healthcare		2804	
homemaker		299	
lawyer		1345	
librarian		5273	
marketing		1950	

# Remember to clean up the mess!!!

# Stop drillbit

- bin/drillbit.sh stop
- Stop mongoDB in Ambari
- Exit from puTTY and VirtualBox



**this is to stop Drill**