NYU | TANDON SCHOOL OF ENGINEERING

Big Data & Hadoop

What is Big Data?

- ***Big data*** essentially means datasets that are too large for traditional data processing systems, and therefore require new processing technologies.
- Datasets whose characteristics - size, data type and frequency - are beyond efficient, accurate and secure processing, as well as storage and extraction, by traditional database management tools.
- Big data technologies (such as Hadoop, HBase, and MongoDB) have received considerable media attention recently.

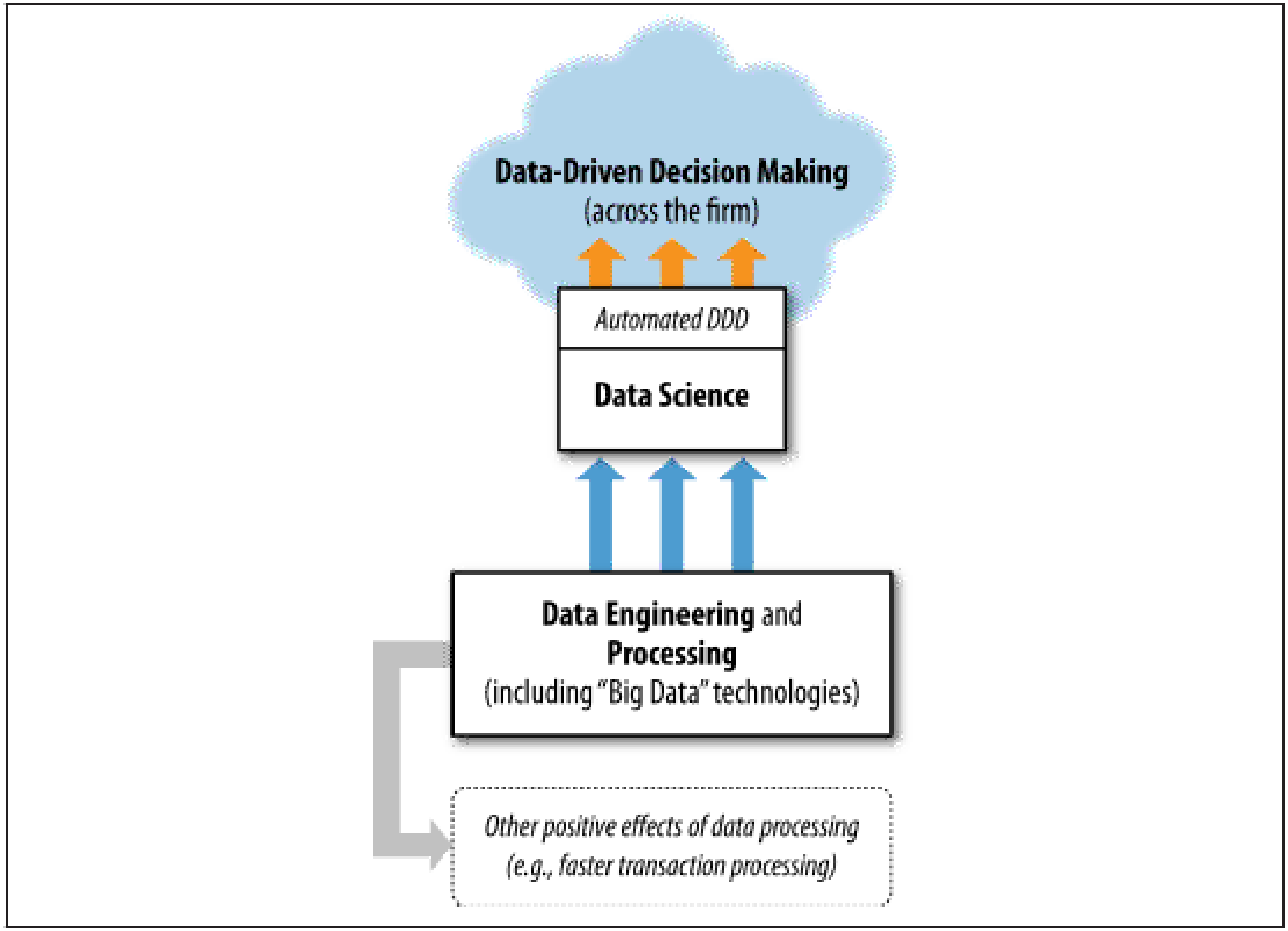


Figure 1-1. Data science in the context of various data-related processes in the organization.

The Four Vs

- Volume – how much data is captured, stored, and processed.
- Velocity – how fast the data is received, and how fast it needs to be processed and extracted.
- Variety – how different and complex the data elements are.
- Value – how effectively the data can be processed for business benefit.

Big Data Challenges in the financial markets

- While the actual volume of data for financial markets applications – compared to generalized Big Data applications, such as social media and retail – is usually not large, the complexity and frequency of data pose significant challenges.
- Data types within financial applications can vary considerably. Relevant data might be unstructured text (such as news stories and social media updates), semi structured text (such as XML), or structured text or binary data (price updates for securities).
- The frequency of updates of data can be extreme, up to several million per second for some markets at peak periods. This requires very specialized processing architectures, comprising complex hardware, software and networking. Techniques such as massively parallel processing and in-memory data storage are generally required, and extremely efficient software coding.

Big Data Challenges in financial markets (Continue)

- The analytical performance required for many applications – in terms of response times for queries – is also challenging, with responses required in fractions of a second – milliseconds – for many applications to be useful.
- Unlike many Big Data applications, those in the financial markets require data elements to be accurate and high precision. For example, a record of a trade price needs to be exact, and its timestamp needs to be resolved within nanoseconds.
- The security requirements of the financial markets, including the need for 24*7 uptime, access controls, audit trails and rollbacks, are beyond many common Big Data technologies, which tend to suffer from their open source origins and lack enterprise-grade functionality.

A variety of trading and risk data for Big Data

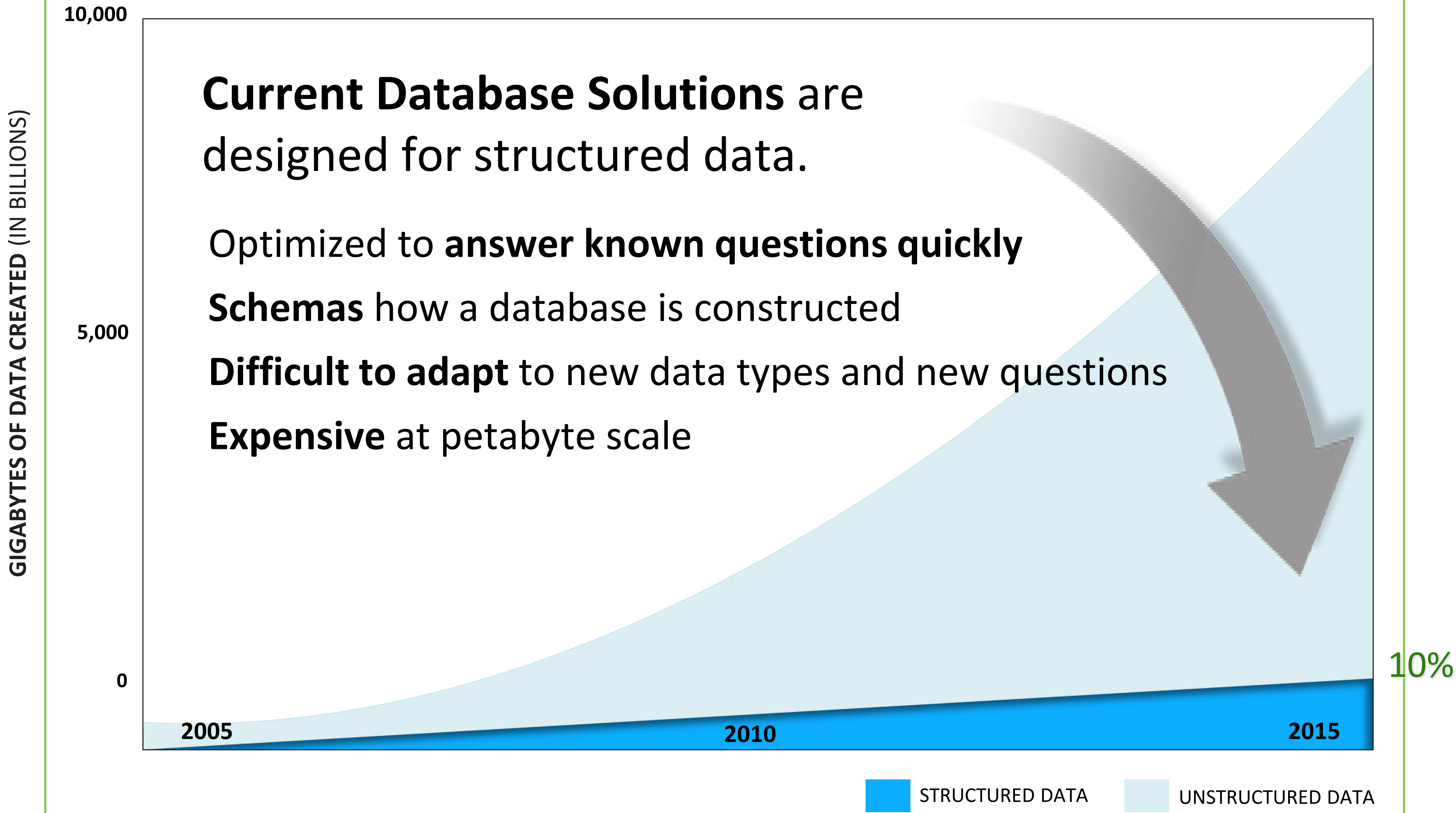
- **Time series price data**
  - as granular as every single tick (trade report) and every order to buy or sell, over an extended period of hours, days, weeks or years.
  - Structured price and associated data, for each transaction .
  - Both manual and automated trade executions.
- **Unstructured real-time data**, including news stories and social media updates.
- **Both structured and unstructured reference data**, varying from records of corporate actions, counterparty and legal entity information, contracts and income flows related to derivatives and structured products.
- **Audio recordings** of transactions negotiated and executed via phone.

Structured vs Unstructured Data

- **Structured data** refers to data that is formatted into records – and fields – that have pre-defined (or self defining) meanings, usages and formats. Such records might be fixed format, where fields are a defined number of characters, bytes, or bits (binary digits), or they might be in an XML format**.**
  - Typical structured data in the financial markets includes pricing updates – trade reports and quotes – as well as end of day snapshot pricing, and historical time series of prices.
- **Unstructured data** relates to data that has no predefined structure, so that it needs to be parsed using techniques such as natural language processing before it becomes useful for analysis.
  - Such data is most commonly text-based, though audio and video also qualifies as unstructured.
  - Key reference data, such as corporate actions, is also often provided in unstructured text format.

Current Database Solutions are designed for structured data.

- Optimized to **answer known questions quickly**
- **Schemas** how a database is constructed
- **Difficult to adapt** to new data types and new questions
- **Expensive** at petabyte scale



Main Big Data Technologies

Hadoop

- Low cost, reliable scale-out architecture
- Distributed computing Proven success in Fortune 500 companies
- Exploding interest

NoSQL Databases

- Huge horizontal scaling and high availability
- Highly optimized for retrieval and appending
- Types
  - Document stores
  - Key Value stores
  - Graph databases

Analytic RDBMS

- Optimized for bulk-load and fast aggregate query workloads
- Types
  - Column-oriented
  - MPP
  - In-memory

Hadoop

NoSQL Databases

Analytic Databases

Hadoop & Databases

Databases "Schema-on-Write"	Hadoop "Schema-on-Read"
▪ Schema must be created before any data can be loaded	▪ Data is simply copied to the file store, no transformation is needed
▪ An explicit load operation has to take place which transforms data to DB internal structure	▪ A SerDe (Serializer/Deserlizer) is applied during read time to extract the required columns (late binding)
▪ New columns must be added explicitly before new data for such columns can be loaded into the database	▪ New data can start flowing anytime and will appear retroactively once the SerDe is updated to parse it
1) Reads are Fast 2) Standards and Governance	1) Loads are Fast 2) Flexibility and Agility

PROS

The Apache Hadoop projects

- The Apache Hadoop projects provide a series of tools designed to solve big data problems. The Hadoop cluster implements a parallel computing cluster using inexpensive commodity hardware. The cluster is partitioned across many servers to provide a near linear scalability. The philosophy of the cluster design is to bring the computing to the data. So each datanode will hold part of the overall data and be able to process the data that it holds. The overall framework for the processing software is called MapReduce.

- **Hadoop Distributed File System (HDFS)**
  - Massive redundant storage across a commodity cluster
- **MapReduce**
  - Map: distribute a computational problem across a cluster
  - Reduce: Master node collects the answers to all the sub-problems and combines them.

Core Hadoop: HDFS

- Self-healing, high bandwidth ***clustered storage***.

- **HDFS breaks incoming files into blocks and stores them redundantly across the cluster.**

Core Hadoop: MapReduce

- **Distributed computing framework**

Processes large jobs in parallel across many nodes and combines the results.

Addressing the Scale Issue

- Single machine cannot serve all the data: you need a distributed special (file) system
- Large number of commodity hardware disks: 1000 disks 1TB each
  - Issue: With Mean time between failures (MTBF) or failure rate of 1/1000, then at least 1 of the above 1000 disks would be down at a given time.
  - Thus failure is norm and not an exception.
  - File system has to be fault-tolerant: replication, checksum
  - Data transfer bandwidth is critical (location of data)

- Critical aspects: fault tolerance + replication + load balancing, monitoring
- Exploit parallelism afforded by splitting parsing and counting

Major Hadoop Utilities

Apache Pig

High-level language for expressing data analysis programs

Apache Hive

SQL-like language and metadata repository

Apache HBase

The Hadoop database. Random, real -time read/write access

Hue

Browser-based desktop interface for interacting with Hadoop

Apache Zookeeper

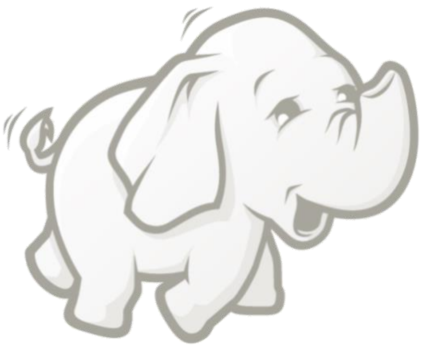
Highly reliable distributed coordination service

Oozie

Server-based workflow engine for Hadoop activities

Flume

Distributed service for collecting and aggregating log and event data



Sqoop

Integrating Hadoop with RDBMS

Apache Whirr

Library for running Hadoop in the cloud

What is MapReduce?

- MapReduce is a programming model Google has used successfully in processing its “big-data” sets (~ 20000 peta bytes per day)
  - Users specify the computation in terms of a *map* and a *reduce* function;
  - Underlying runtime system automatically parallelizes the computation across large-scale clusters of machines, and
  - Underlying system also handles machine failures, efficient communications, and performance issues.

-- Reference: Dean, J. and Ghemawat, S. 2008. **MapReduce: simplified data processing on large clusters.** *Communication of ACM* 51, 1 (Jan. 2008), 107-113.

From CS Foundations to MapReduce

Consider a large data collection:

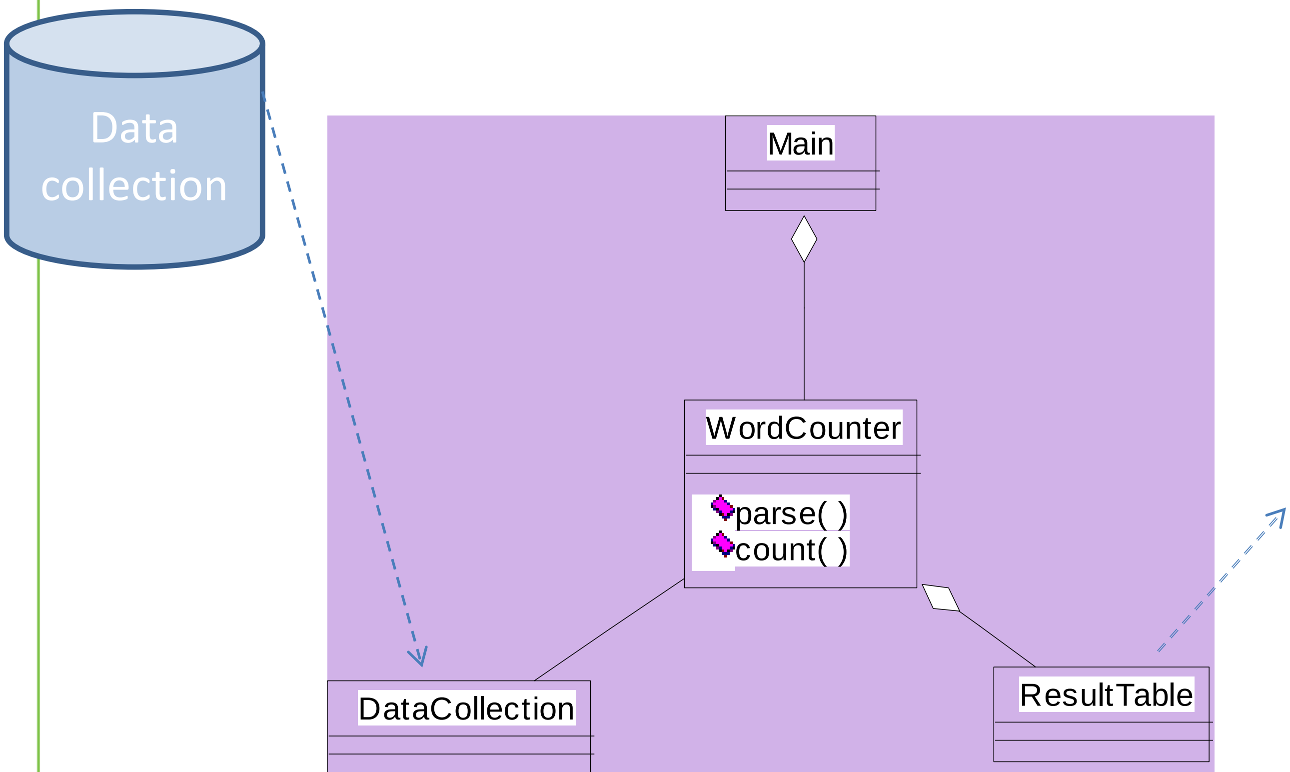
- {web, weed, green, sun, moon, land, part, web, green,…}

Problem:

- Count the occurrences of the different words in the collection.

Word Counter and Result Table

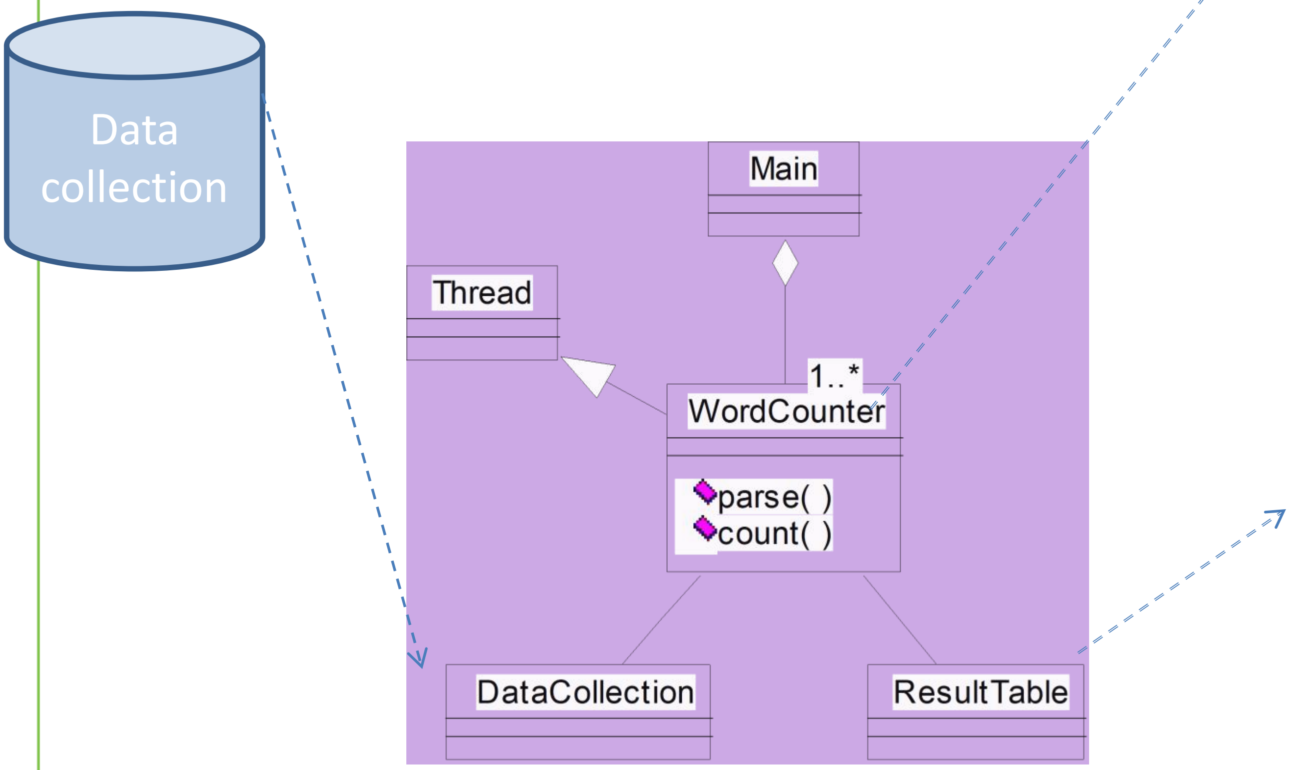
{web, weed, green, sun, moon, land, part,
web, green,…}



web	2
weed	1
green	2
sun	1
moon	1
land	1
part	1

Multiple Instances of Word Counter

{web, weed, green, sun, moon, land, part, web, green,...}

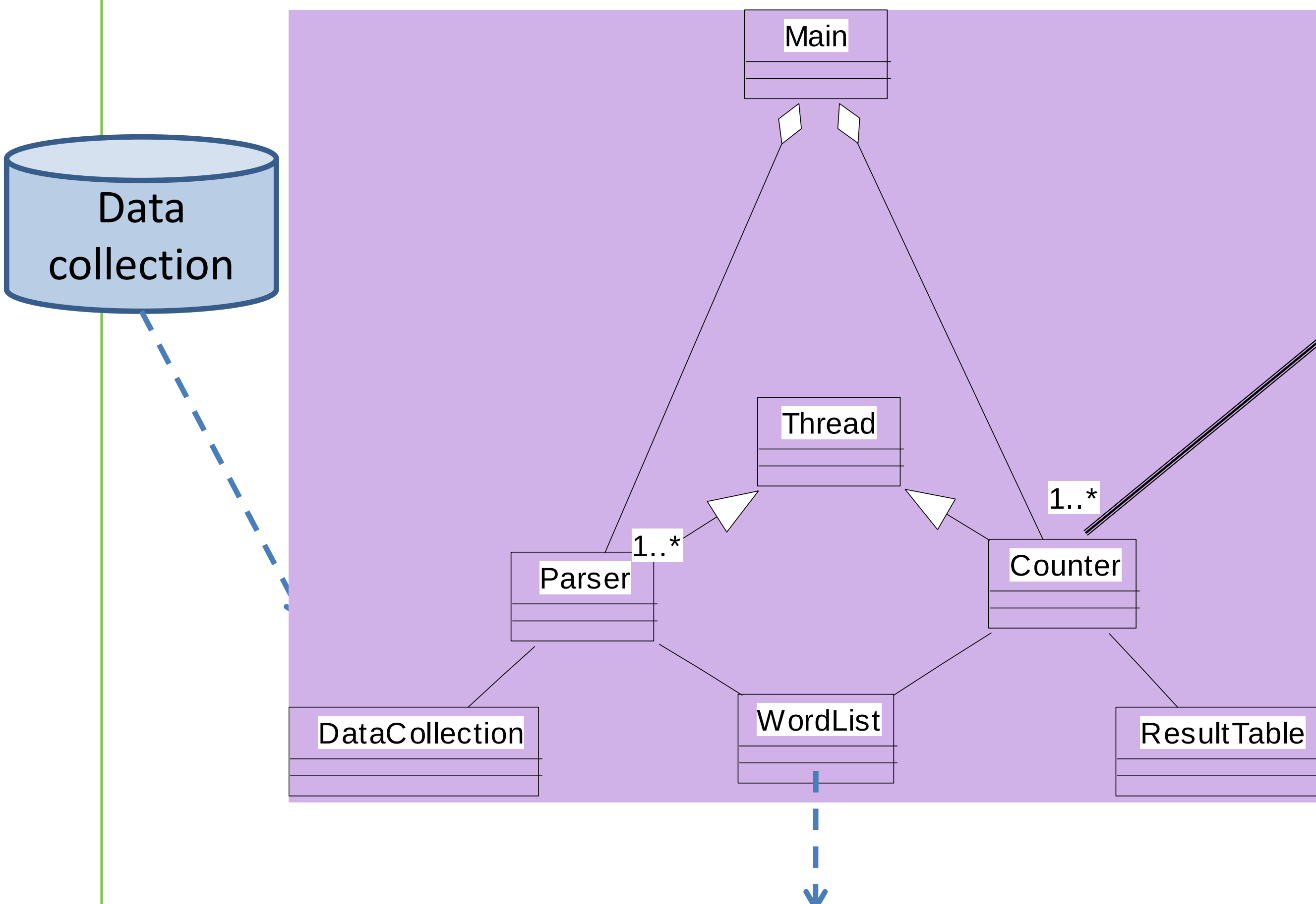


web	2
weed	1
green	2
sun	1
moon	1
land	1
part	1

Observe:
Multi-thread
Lock on shared data

Improve Word Counter for Performance

No need for lock

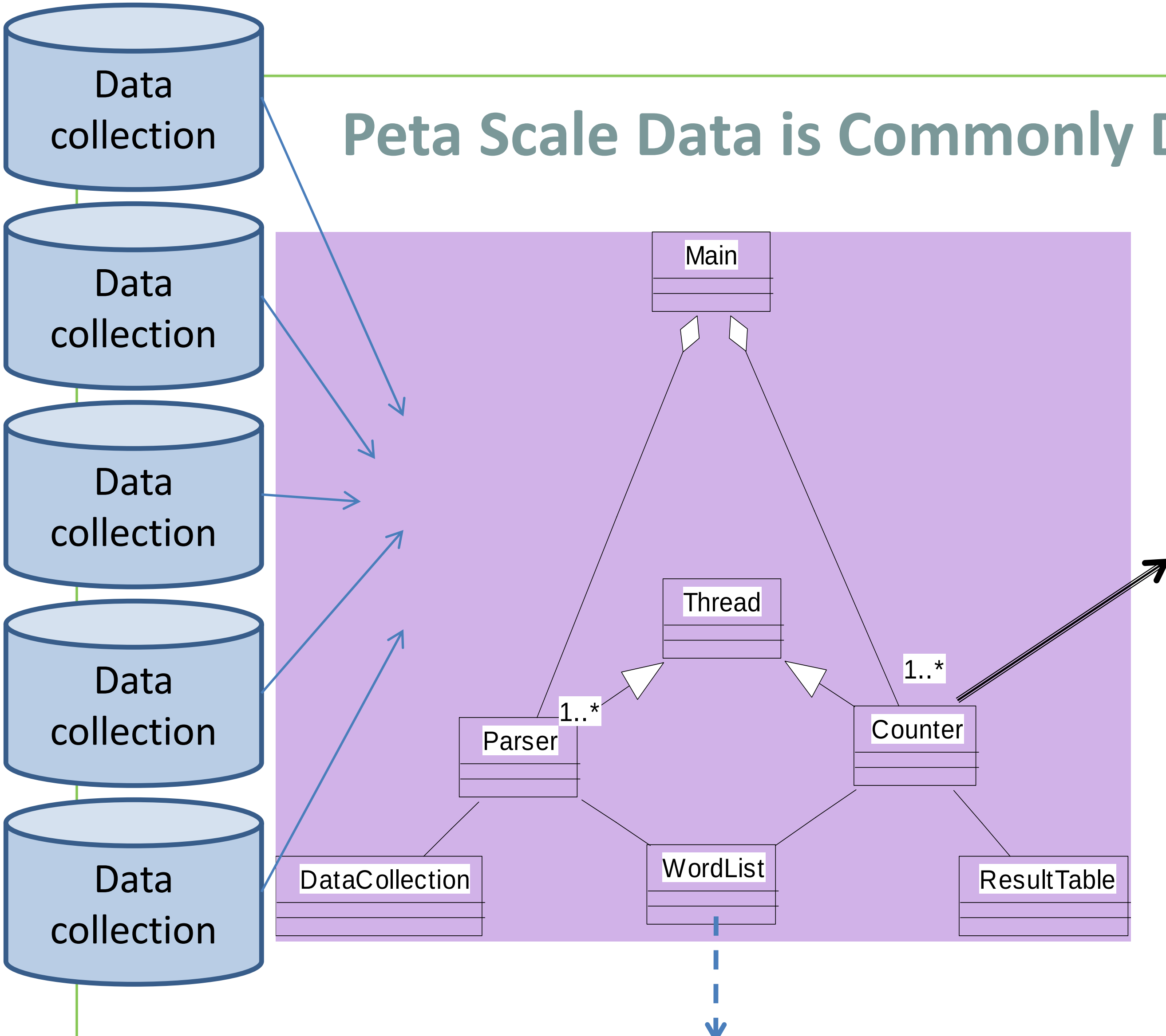


web	**2**
weed	**1**
green	**2**
sun	**1**
moon	**1**
land	**1**
part	**1**

Separate counters

KEY	**web**	**weed**	**green**	**sun**	**moon**	**land**	**part**	**web**	**green**	**…….**
VALUE										

Peta Scale Data is Commonly Distributed

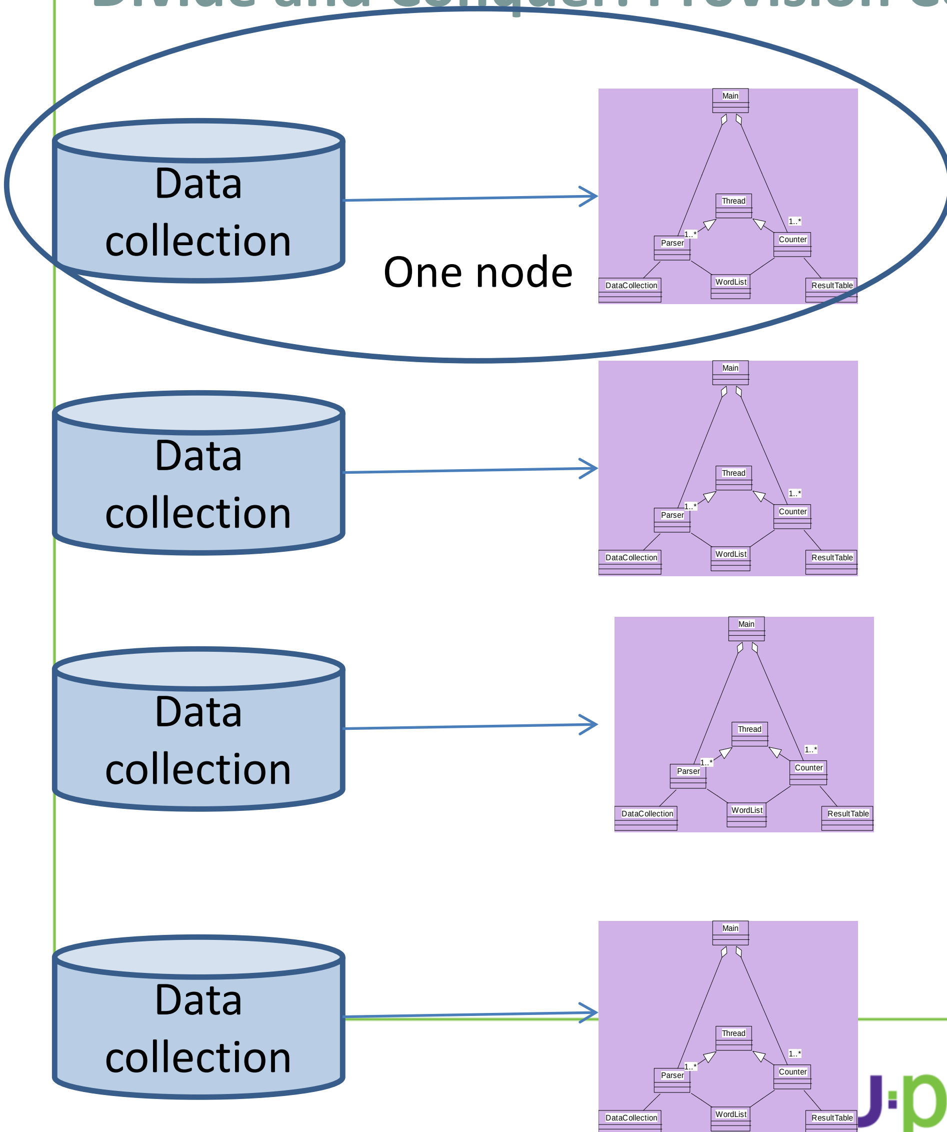


web	2
weed	1
green	2
sun	1
moon	1
land	1
part	1

Issue: managing the large scale data
Write once read many (WORM) data

KEY	web	weed	green	sun	moon	land	part	web	green	....
VALUE										

Divide and Conquer: Provision Computing at Data Location



For our example,
#1: Schedule parallel parse tasks
#2: Schedule parallel count tasks

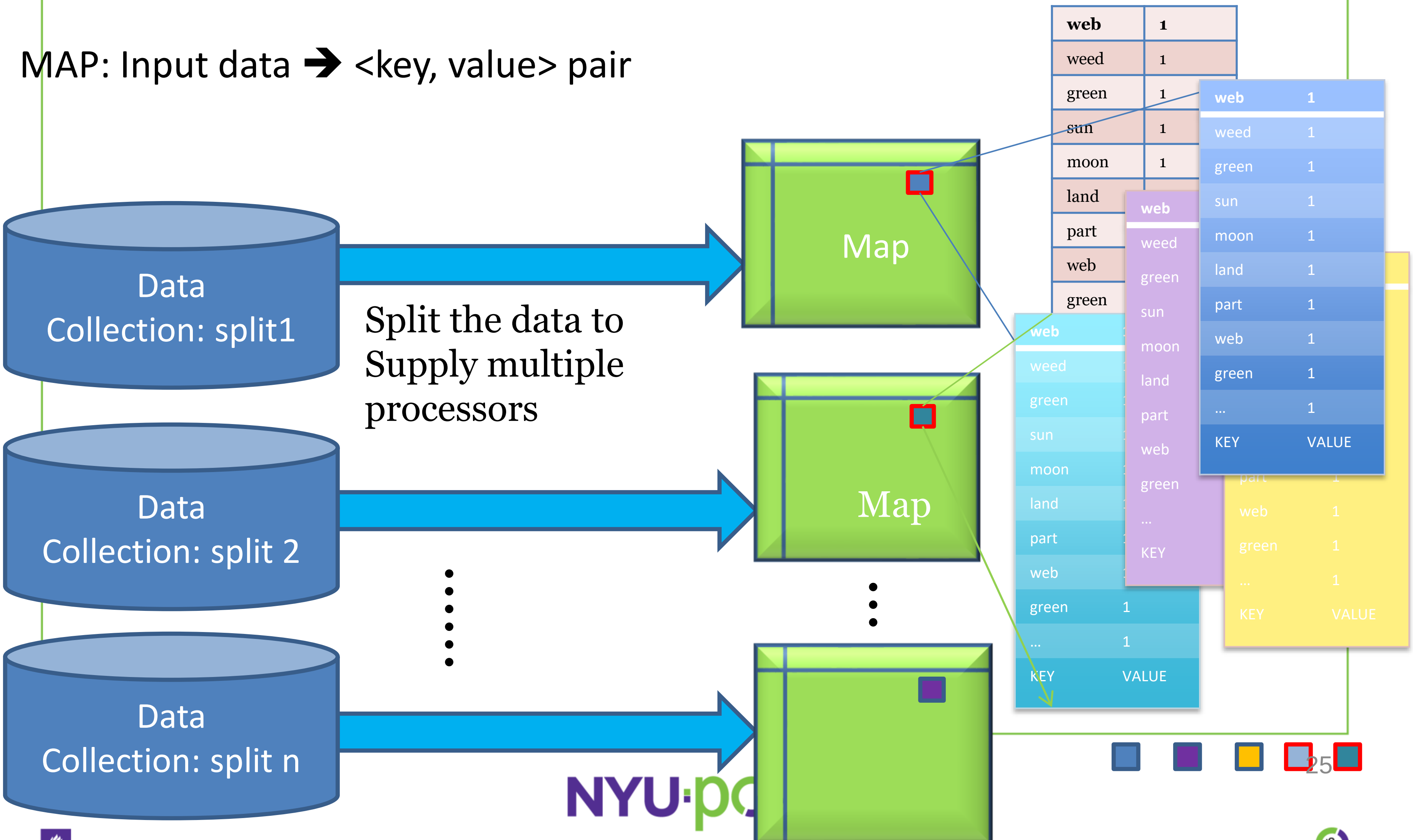
This is a particular solution;
Lets generalize it:

Our parse is a mapping operation:
MAP: input → <key, value> pairs

Our count is a reduce operation:
REDUCE: <key, value> pairs reduced

Map Operation

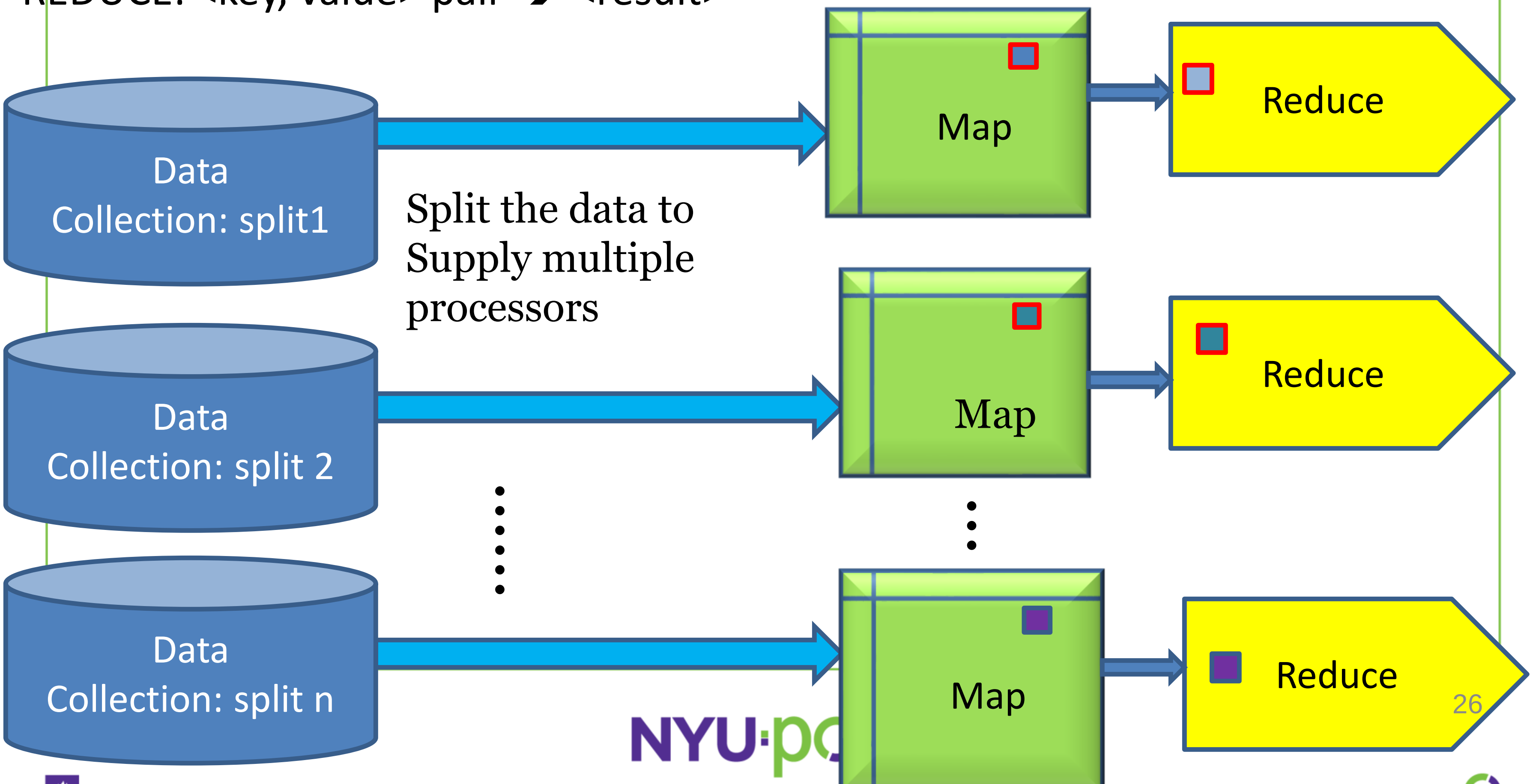
MAP: Input data ➔ <key, value> pair

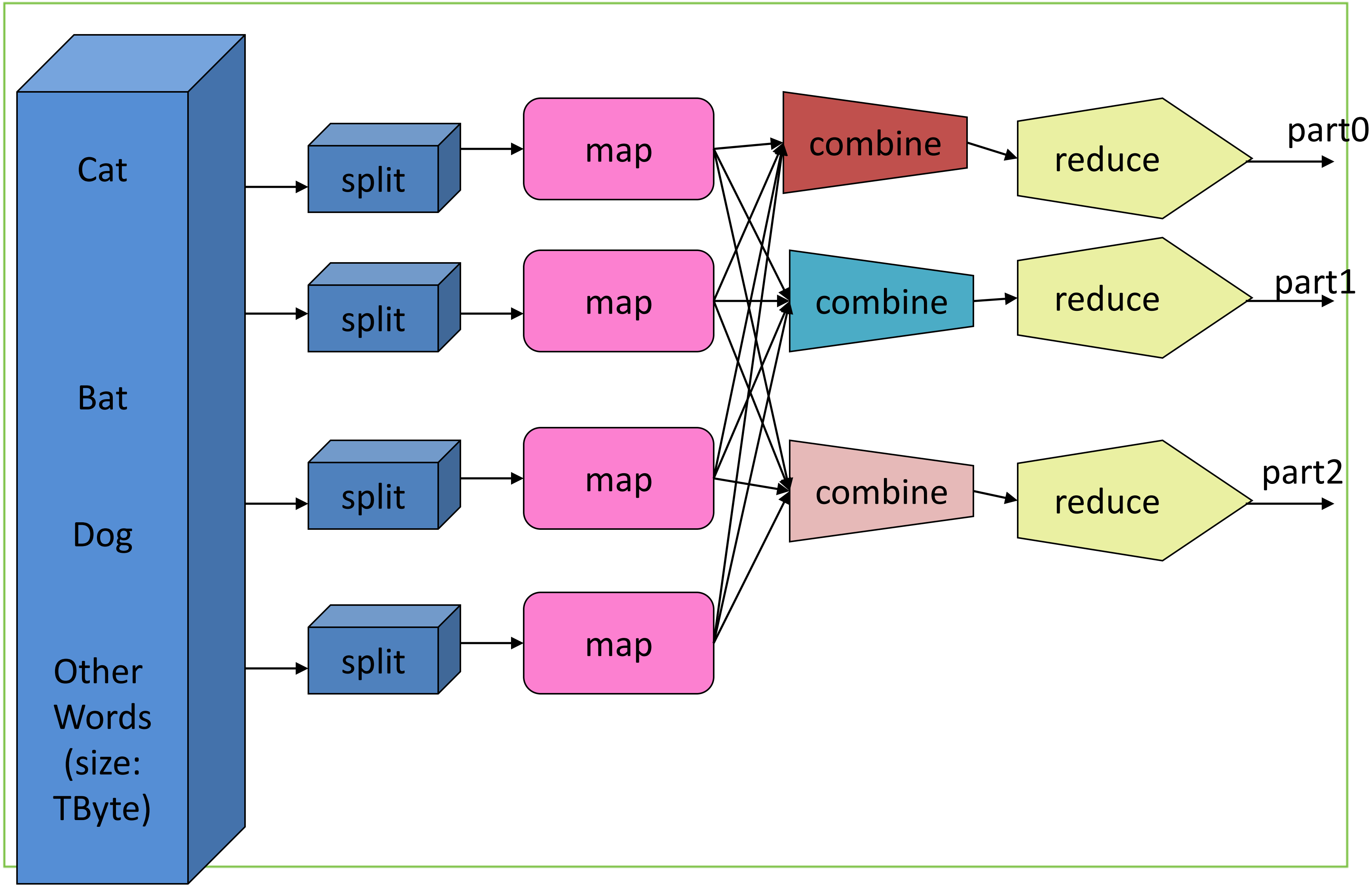


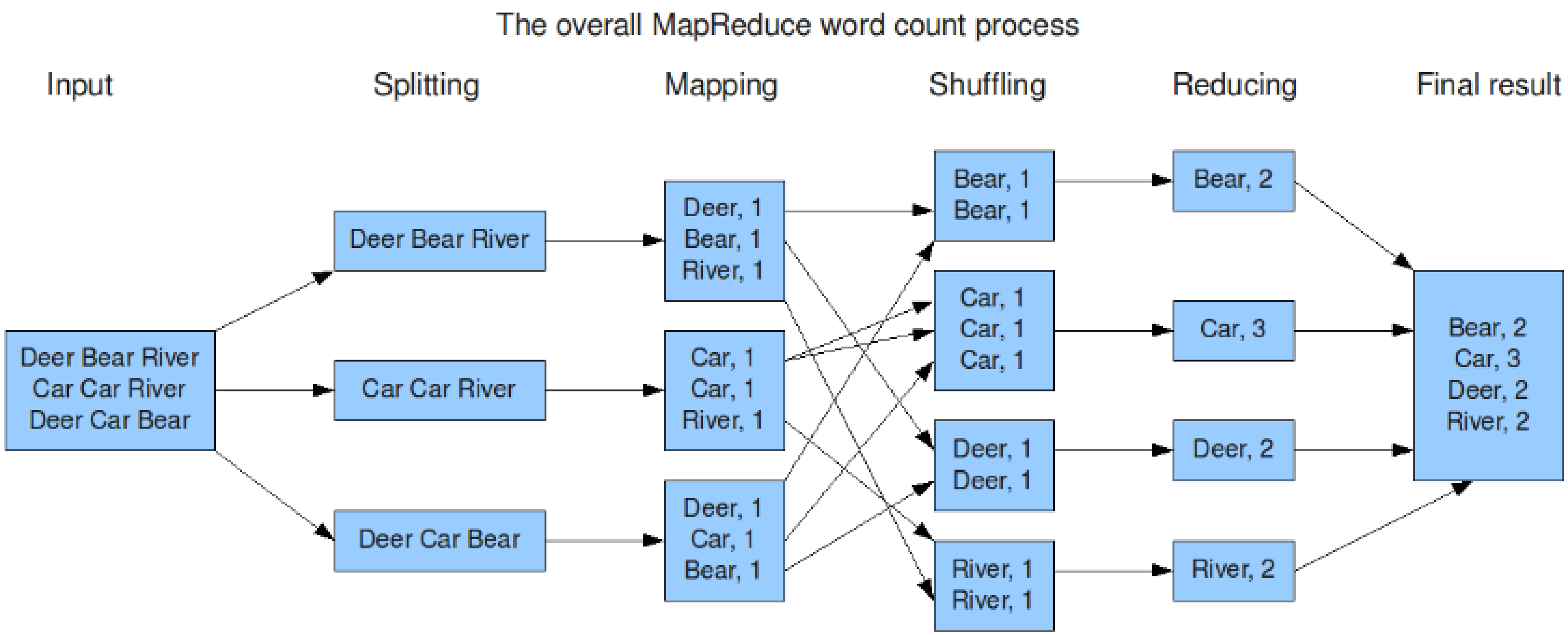
Reduce Operation

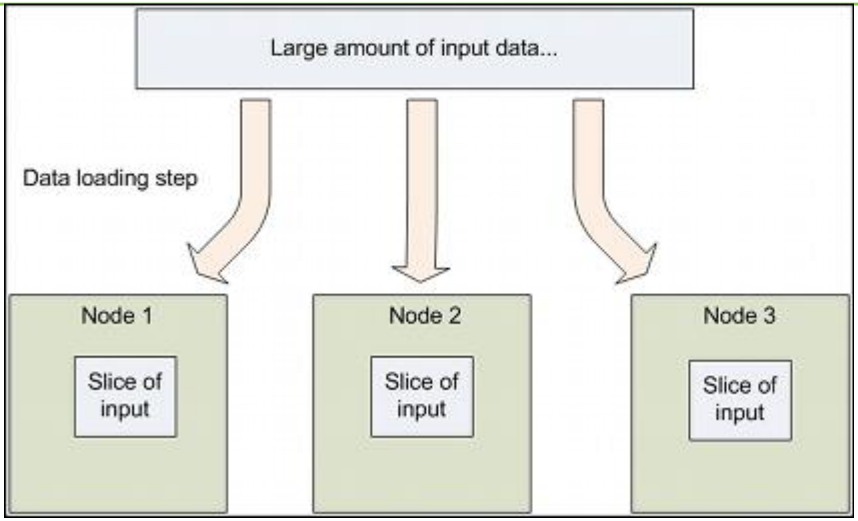
MAP: Input data ➔ <key, value> pair

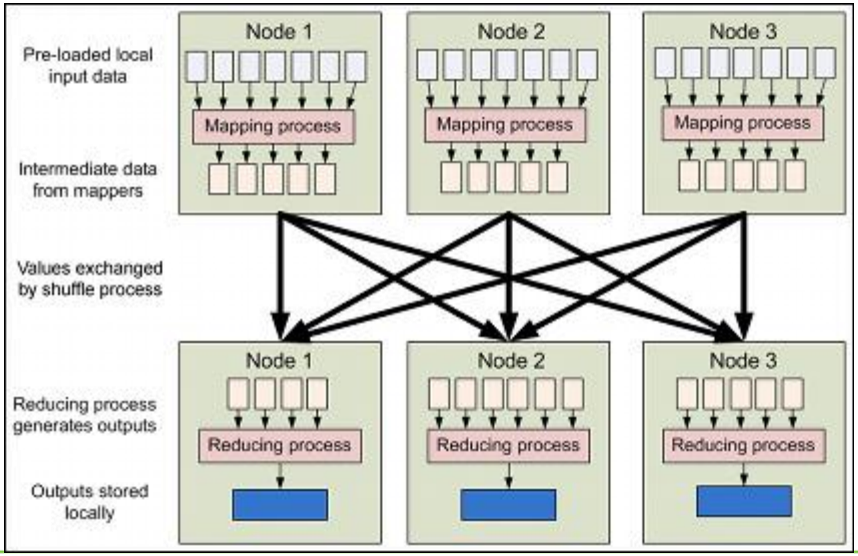
REDUCE: <key, value> pair ➔ <result>




Cat
Bat
Dog
Other Words (size: TByte)
split
split
split
split
map
map
map
map
combine
combine
combine
reduce
reduce
reduce
part0
part1
part2


The overall MapReduce word count process
Input
Splitting
Mapping
Shuffling
Reducing
Final result
Deer Bear River
Car Car River
Deer Car Bear
Deer Bear River
Car Car River
Deer Car Bear
Deer, 1
Bear, 1
River, 1
Car, 1
Car, 1
River, 1
Deer, 1
Car, 1
Bear, 1
Bear, 1
Bear, 1
Car, 1
Car, 1
Car, 1
Deer, 1
Deer, 1
River, 1
River, 1
Bear, 2
Car, 3
Deer, 2
River, 2
Bear, 2
Car, 3
Deer, 2
River, 2

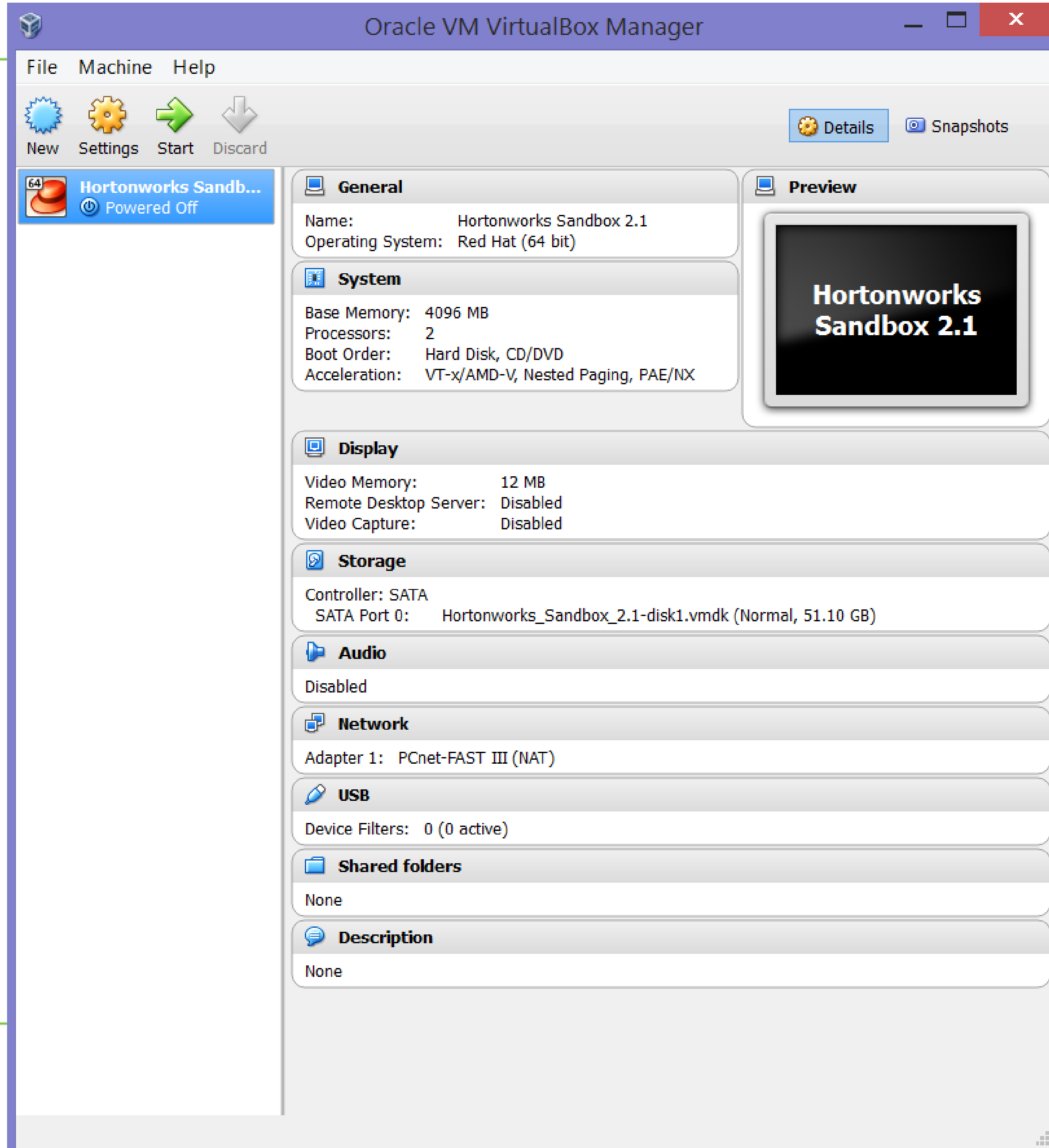


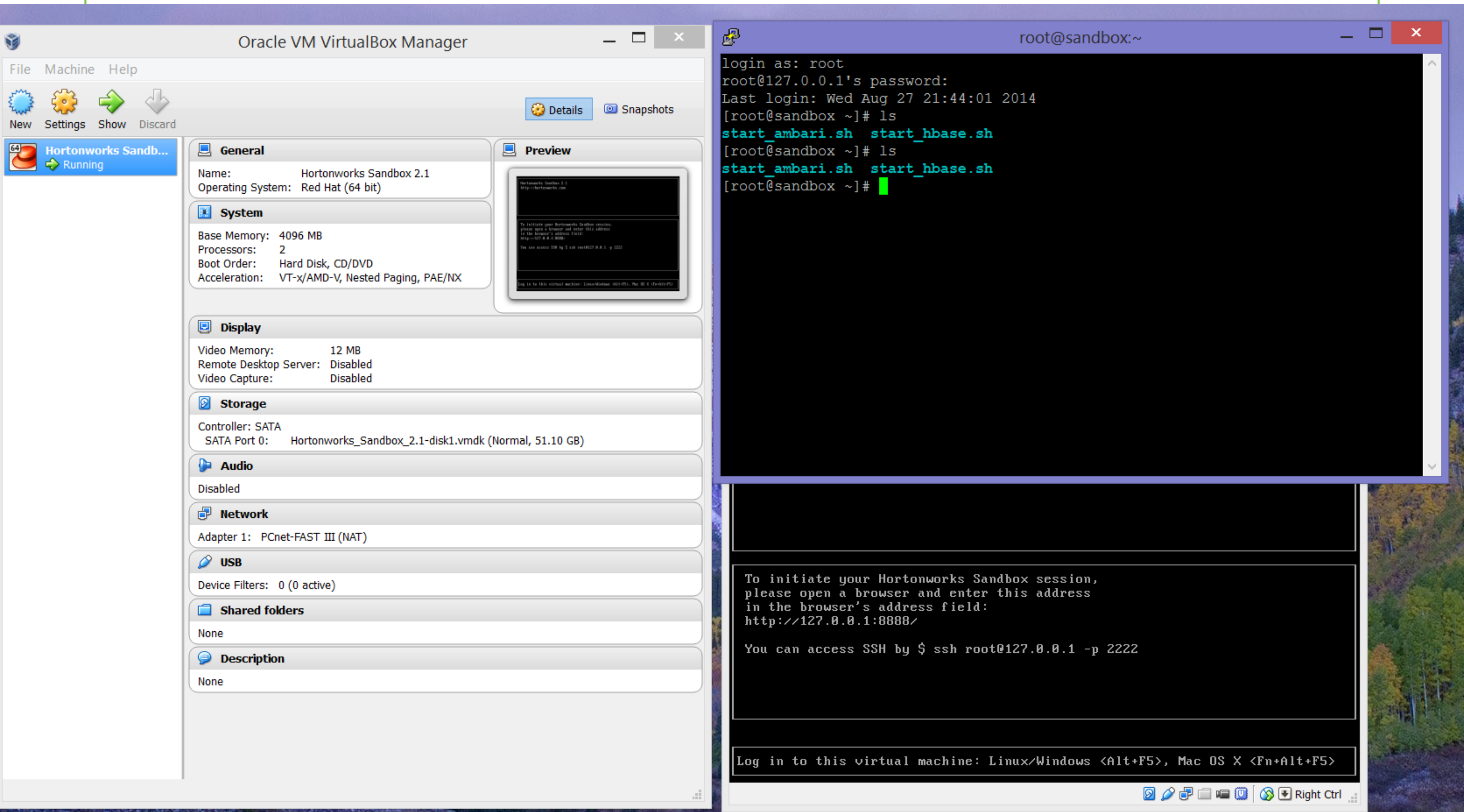


Hortonworks

Hortonworks Sandbox and HDP

- The Hortonworks Sandbox is a single node implementation of the Hortonworks Data Platform(HDP). It is packaged as a virtual machine to make evaluation and experimentation with HDP fast and easy.
- Latest Releases of HDP Sandbox: HDP 2.1 Sandbox on Oracle VirtualBox.
  - System Requirements
    - Now runs on 32-bit and 64-bit OS (Windows XP, Windows 7, Windows 8 and Mac OSX)
    - Minimum 4GB RAM; 8Gb required to run Ambari and Hbase
    - Virtualization enabled on BIOS
    - Browser: Chrome 25+, IE 9+, Safari 6+ recommended. (Sandbox will not run on IE 10)


Oracle VM VirtualBox Manager
File Machine Help
New Settings Start Discard
Details Snapshots
Hortonworks Sandb...
Powered Off
General
Name: Hortonworks Sandbox 2.1
Operating System: Red Hat (64 bit)
System
Base Memory: 4096 MB
Processors: 2
Boot Order: Hard Disk, CD/DVD
Acceleration: VT-x/AMD-V, Nested Paging, PAE/NX
Preview
Hortonworks Sandbox 2.1
Display
Video Memory: 12 MB
Remote Desktop Server: Disabled
Video Capture: Disabled
Storage
Controller: SATA
SATA Port 0: Hortonworks_Sandbox_2.1-disk1.vmdk (Normal, 51.10 GB)
Audio
Disabled
Network
Adapter 1: PCnet-FAST III (NAT)
USB
Device Filters: 0 (0 active)
Shared folders
None
Description
None


Oracle VM VirtualBox Manager
File Machine Help
New Settings Show Discard
Details Snapshots
Hortonworks Sandb...
Running
General
Name: Hortonworks Sandbox 2.1
Operating System: Red Hat (64 bit)
System
Base Memory: 4096 MB
Processors: 2
Boot Order: Hard Disk, CD/DVD
Acceleration: VT-x/AMD-V, Nested Paging, PAE/NX
Preview
Display
Video Memory: 12 MB
Remote Desktop Server: Disabled
Video Capture: Disabled
Storage
Controller: SATA
SATA Port 0: Hortonworks_Sandbox_2.1-disk1.vmdk (Normal, 51.10 GB)
Audio
Disabled
Network
Adapter 1: PCnet-FAST III (NAT)
USB
Device Filters: 0 (0 active)
Shared folders
None
Description
None
root@sandbox:~
login as: root
root@127.0.0.1's password:
Last login: Wed Aug 27 21:44:01 2014
[root@sandbox ~]# ls
start_ambari.sh start_hbase.sh
[root@sandbox ~]# ls
start_ambari.sh start_hbase.sh
[root@sandbox ~]#
To initiate your Hortonworks Sandbox session,
please open a browser and enter this address
in the browser's address field:
http://127.0.0.1:8888/
You can access SSH by $ ssh root@127.0.0.1 -p 2222
Log in to this virtual machine: Linux/Windows <Alt+F5>, Mac OS X <Fn+Alt+F5>
Right Ctrl

NYSE-2000-2001.tsv.gz - Fi...

127.0.0.1:8000/filebrowser/view//user/hue/NYSE-2000-2001.tsv.gz

hue

About Hortonworks Sandbox 2.1

Home / user / hue / **NYSE-2000-2001.tsv.gz**

ACTIONS

- View As Binary
- Stop preview
- Download
- View File Location
- Refresh

INFO

Last Modified
Aug. 27, 2014 9:16 p.m.

User
hue

Group
hue

Size
10.8 MB

Mode
100755

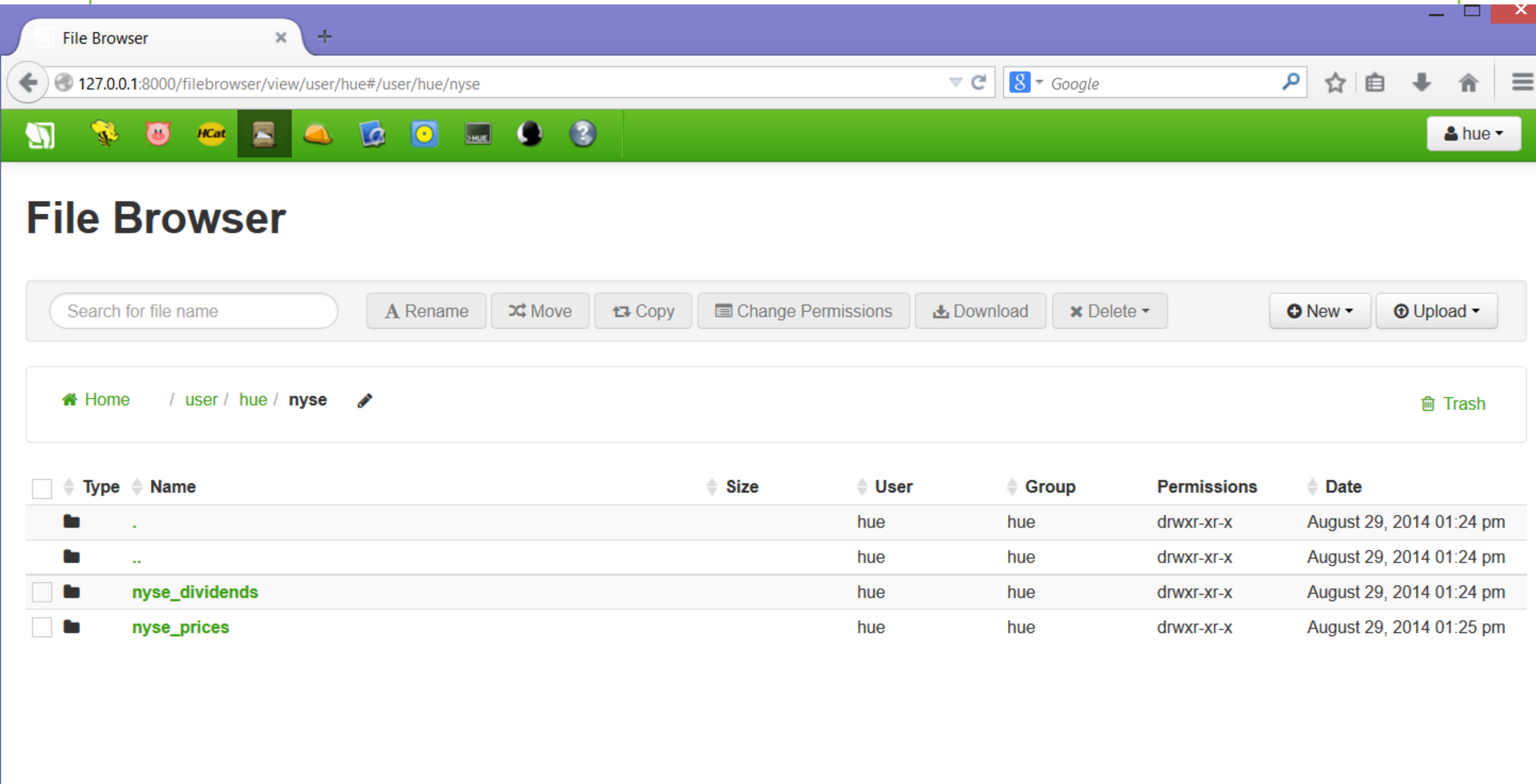
exchange	stock_symbol	date	stock_price_open	stock_price_high	stock_price_low	stock_price_close	stock_volume	stock_price_adj_close
NYSE	ASP	2001-12-31	12.55	12.8	12.42	12.8	11300	6.91
NYSE	ASP	2001-12-28	12.5	12.55	12.42	12.55	4800	6.78
NYSE	ASP	2001-12-27	12.59	12.59	12.5	12.57	5400	6.79
NYSE	ASP	2001-12-26	12.45	12.6	12.45	12.55	5400	6.78
NYSE	ASP	2001-12-24	12.61	12.61	12.61	12.61	1400	6.76
NYSE	ASP	2001-12-21	12.4	12.78	12.4	12.6	18200	6.75
NYSE	ASP	2001-12-20	12.35	12.58	12.35	12.4	4200	6.65
NYSE	ASP	2001-12-19	12.42	12.6	12.35	12.6	10100	6.75
NYSE	ASP	2001-12-18	12.37	12.5	12.37	12.41	10100	6.65
NYSE	ASP	2001-12-17	12.4	12.52	12.4	12.52	8000	6.71
NYSE	ASP	2001-12-14	12.54	12.54	12.32	12.4	283000	6.65
NYSE	ASP	2001-12-13	12.4	12.55	12.4	12.54	13700	6.72
NYSE	ASP	2001-12-12	12.55	12.55	12.4	12.4	6900	6.65
NYSE	ASP	2001-12-11	12.6	12.6	12.45	12.55	8900	6.73
NYSE	ASP	2001-12-10	12.5	12.6	12.43	12.6	4400	6.75
NYSE	ASP	2001-12-07	12.6	12.65	12.43	12.6	9600	6.75
NYSE	ASP	2001-12-06	12.7	12.71	12.65	12.65	3400	6.78
NYSE	ASP	2001-12-05	12.63	12.81	12.45	12.7	15800	6.81
NYSE	ASP	2001-12-04	12.79	12.79	12.6	12.65	10800	6.78
NYSE	ASP	2001-12-03	12.72	12.79	12.65	12.79	8500	6.85
NYSE	ASP	2001-11-30	12.75	12.81	12.7	12.79	14600	6.8
NYSE	ASP	2001-11-29	12.7	12.82	12.7	12.82	3300	6.82
NYSE	ASP	2001-11-28	12.52	12.79	12.52	12.79	11100	6.8
NYSE	ASP	2001-11-27	12.53	12.6	12.42	12.42	12300	6.61
NYSE	ASP	2001-11-26	12.6	12.65	12.53	12.65	8200	6.73
NYSE	ASP	2001-11-23	12.5	12.5	12.5	12.5	3600	6.65
NYSE	ASP	2001-11-21	12.65	12.65	12.57	12.57	4400	6.69

127.0.0.1:8000/about

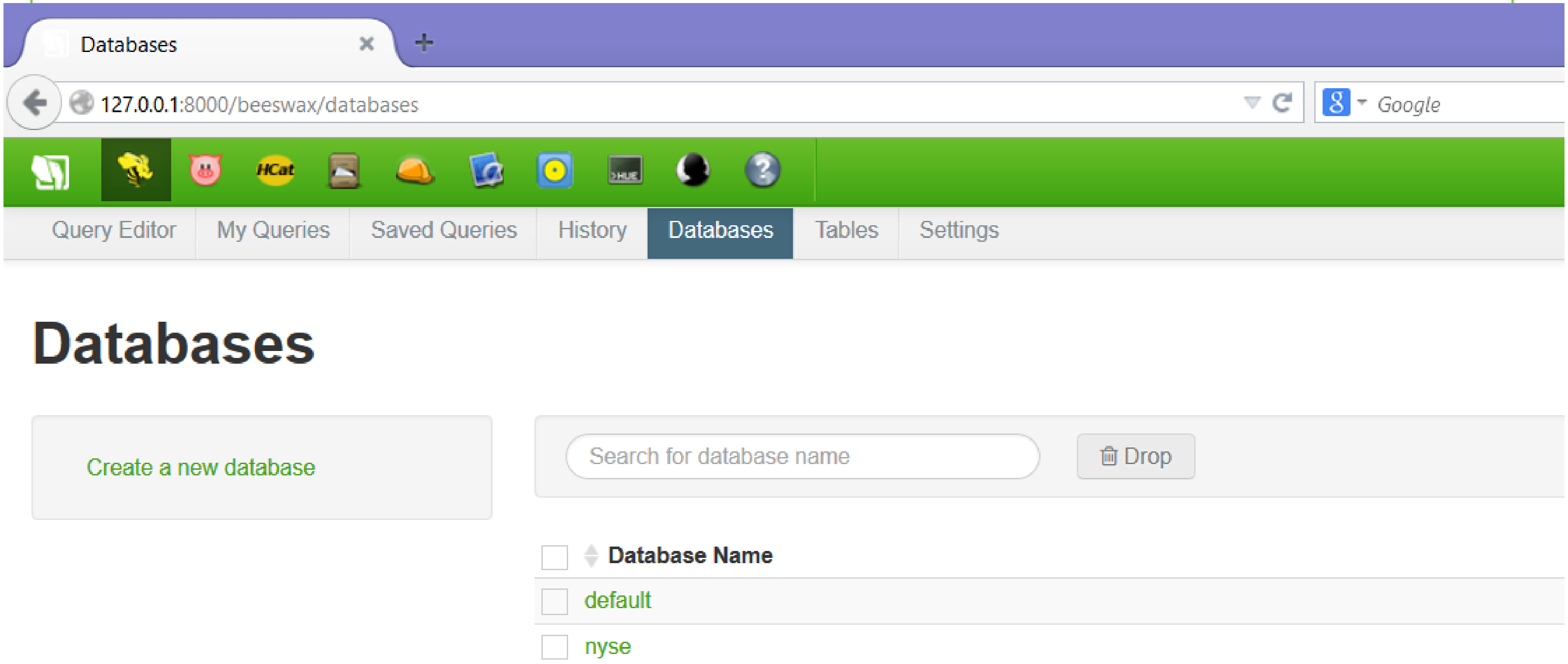
From Raw Data to Insight

- We will using HDP and Microsoft Business Intelligence to:
  - Cleaning and aggregating 10 years of stock price and dividend data. Enriching the data model by looking up additional attributes from Wikipedia
  - Creating an interactive visualization on the model

Staging the data on HDFS

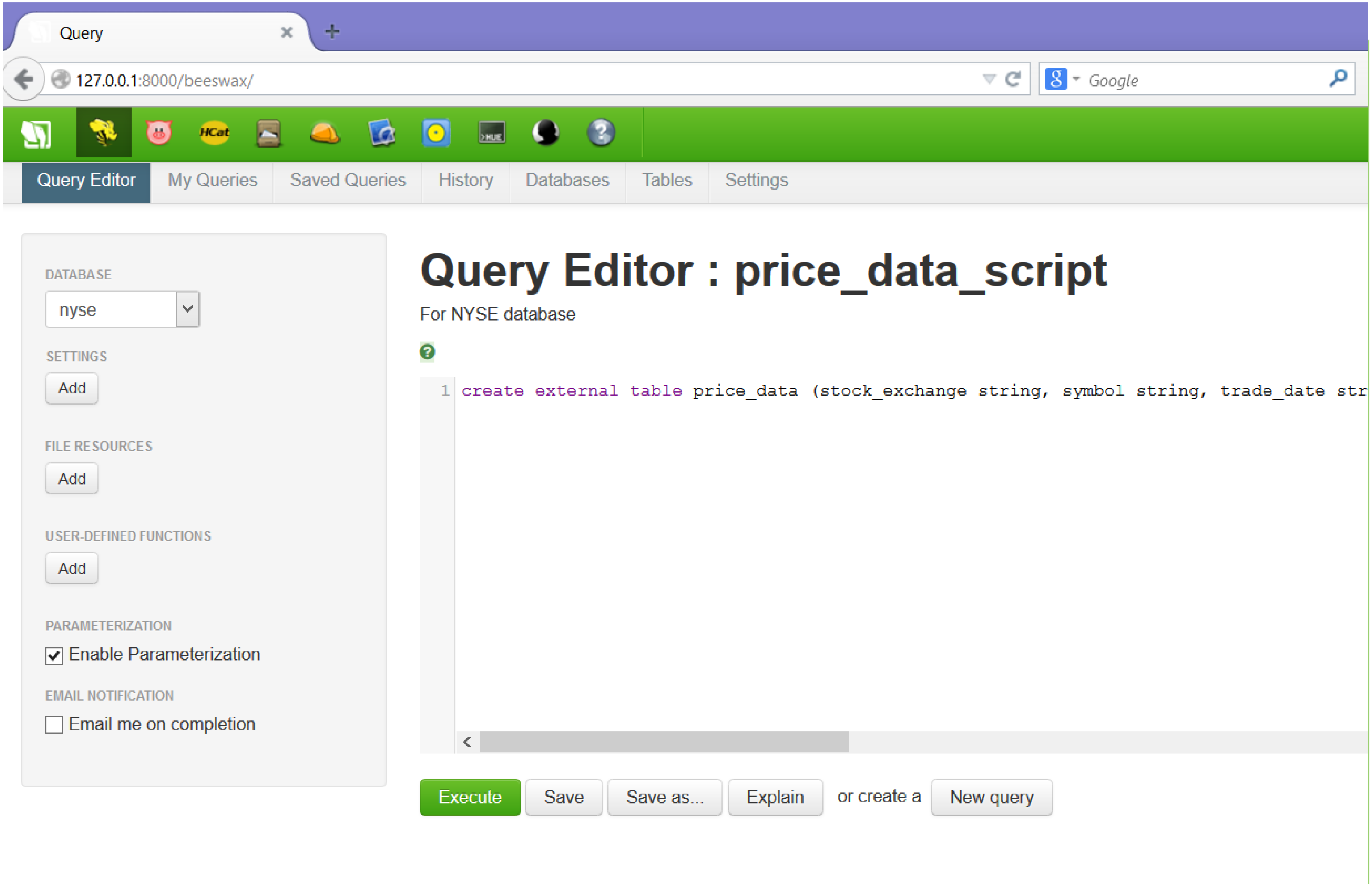


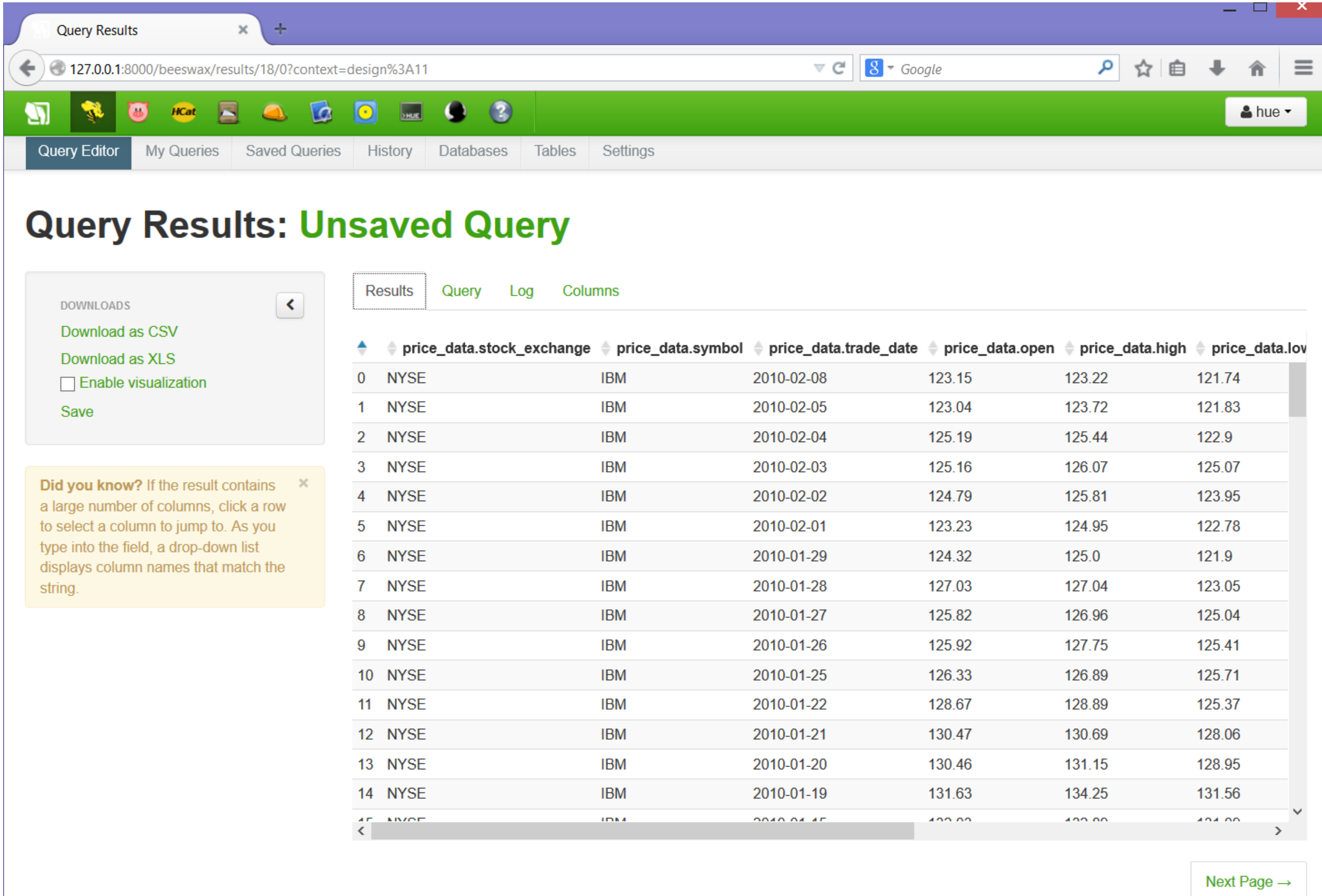
Create the database NYSE



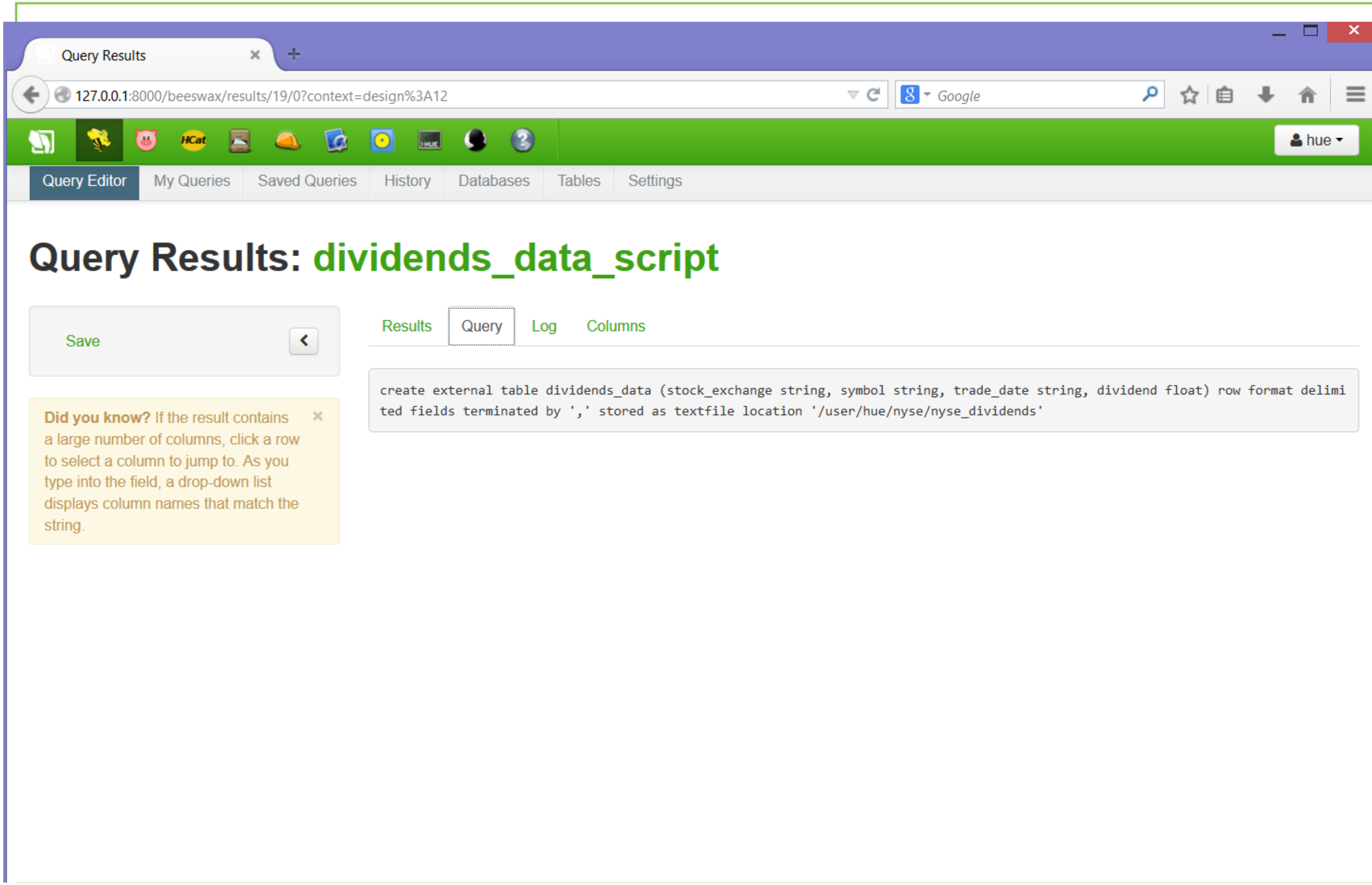
Creating a Hive schema on the raw data

- Use the Beeswax UI of Hive to execute the DDL queries:
  - create external table price_data (stock_exchange string, symbol string, trade_date string, open float, high float, low float, close float, volume int, adj_close float) row format delimited fields terminated by ',' stored as textfile location '/user/hue/nyse/nyse_prices';
  - create external table dividends_data (stock_exchange string, symbol string, trade_date string, dividend float) row format delimited fields terminated by ',' stored as textfile location '/user/hue/nyse/nyse_dividends';
- Test your results:
  - select * from price_data where symbol = 'IBM';
  - select * from dividends_data where symbol = 'IBM';


Query
127.0.0.1:8000/beeswax/
Google
HCat
Query Editor
My Queries
Saved Queries
History
Databases
Tables
Settings
DATABASE
nyse
SETTINGS
Add
FILE RESOURCES
Add
USER-DEFINED FUNCTIONS
Add
PARAMETERIZATION
Enable Parameterization
EMAIL NOTIFICATION
Email me on completion
Query Editor : price_data_script
For NYSE database
1 create external table price_data (stock_exchange string, symbol string, trade_date str
Execute
Save
Save as...
Explain
or create a
New query



	price_data.stock_exchange	price_data.symbol	price_data.trade_date	price_data.open	price_data.high	price_data.lov
0	NYSE	IBM	2010-02-08	123.15	123.22	121.74
1	NYSE	IBM	2010-02-05	123.04	123.72	121.83
2	NYSE	IBM	2010-02-04	125.19	125.44	122.9
3	NYSE	IBM	2010-02-03	125.16	126.07	125.07
4	NYSE	IBM	2010-02-02	124.79	125.81	123.95
5	NYSE	IBM	2010-02-01	123.23	124.95	122.78
6	NYSE	IBM	2010-01-29	124.32	125.0	121.9
7	NYSE	IBM	2010-01-28	127.03	127.04	123.05
8	NYSE	IBM	2010-01-27	125.82	126.96	125.04
9	NYSE	IBM	2010-01-26	125.92	127.75	125.41
10	NYSE	IBM	2010-01-25	126.33	126.89	125.71
11	NYSE	IBM	2010-01-22	128.67	128.89	125.37
12	NYSE	IBM	2010-01-21	130.47	130.69	128.06
13	NYSE	IBM	2010-01-20	130.46	131.15	128.95
14	NYSE	IBM	2010-01-19	131.63	134.25	131.56



Query Results
127.0.0.1:8000/beeswax/results/19/0?context=design%3A12
Google
hue
Query Editor
My Queries
Saved Queries
History
Databases
Tables
Settings
Query Results: dividends_data_script
Save
Did you know? If the result contains a large number of columns, click a row to select a column to jump to. As you type into the field, a drop-down list displays column names that match the string.
Results
Query
Log
Columns
create external table dividends_data (stock_exchange string, symbol string, trade_date string, dividend float) row format delimited fields terminated by ',' stored as textfile location '/user/hue/nyse/nyse_dividends'

Query Results

127.0.0.1:8000/beeswax/results/20/0?context=design%3A13

hue

Query Editor | My Queries | Saved Queries | History | Databases | Tables | Settings

Query Results: Unsaved Query

DOWNLOADS

Download as CSV

Download as XLS

☐ Enable visualization

Save

MR JOB (1)

1409326896625_0016

Did you know? If the result contains a large number of columns, click a row to select a column to jump to. As you type into the field, a drop-down list displays column names that match the string.

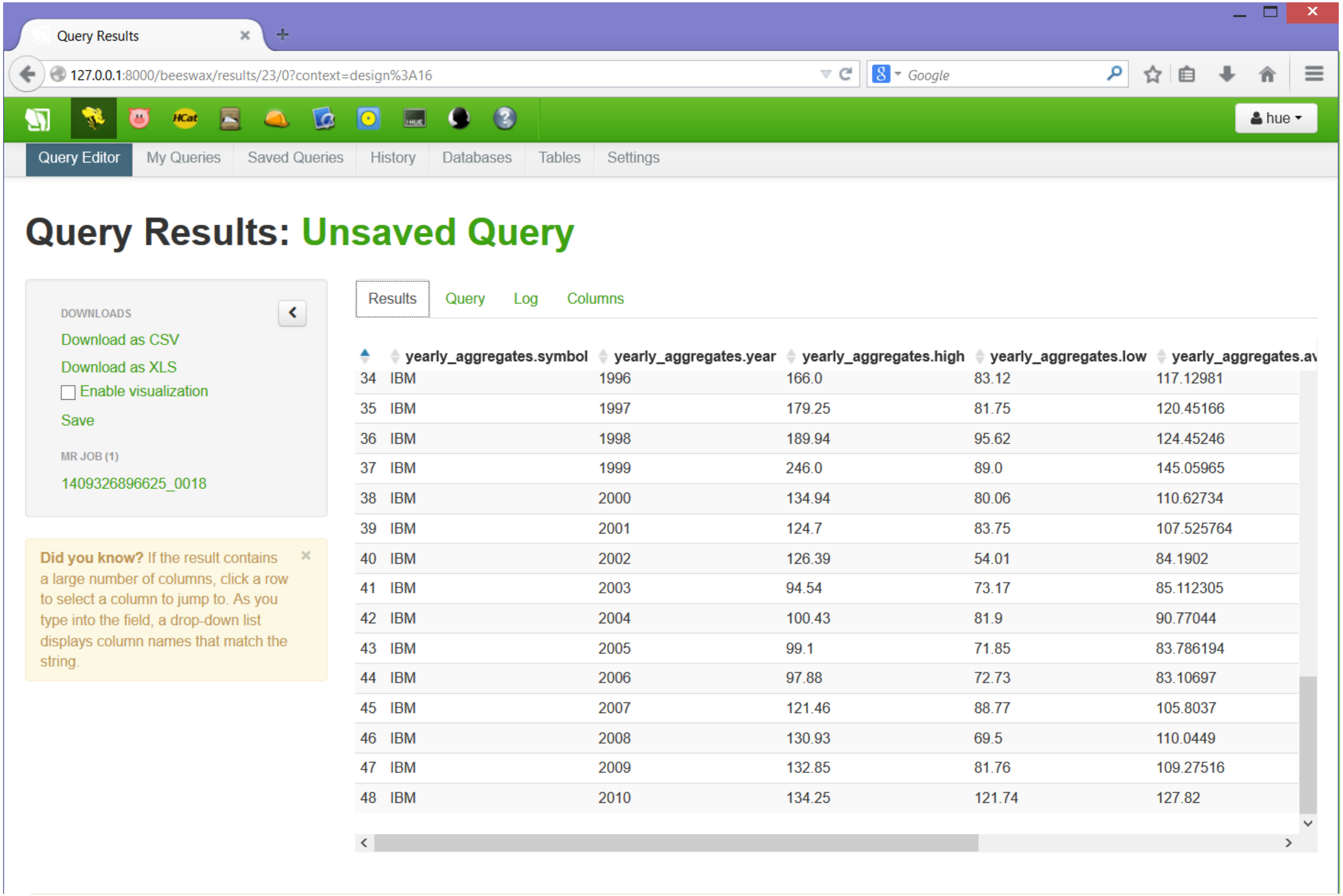
Results | Query | Log | Columns

	dividends_data.stock_exchange	dividends_data.symbol	dividends_data.trade_date	dividends_data.dividend
0	NYSE	IBM	2010-02-08	0.55
1	NYSE	IBM	2009-11-06	0.55
2	NYSE	IBM	2009-08-06	0.55
3	NYSE	IBM	2009-05-06	0.55
4	NYSE	IBM	2009-02-06	0.5
5	NYSE	IBM	2008-11-06	0.5
6	NYSE	IBM	2008-08-06	0.5
7	NYSE	IBM	2008-05-07	0.5
8	NYSE	IBM	2008-02-06	0.4
9	NYSE	IBM	2007-11-07	0.4
10	NYSE	IBM	2007-08-08	0.4
11	NYSE	IBM	2007-05-08	0.4
12	NYSE	IBM	2007-02-07	0.3
13	NYSE	IBM	2006-11-08	0.3
14	NYSE	IBM	2006-08-08	0.3
15	NYSE	IBM	2006-05-08	0.3

Next Page →

Aggregating the stocks and dividend data

- The DDL statement to create the table 'yearly_aggregates':
  - create table yearly_aggregates (symbol string, year string, high float, low float, average_close float, total_dividends float) row format delimited fields terminated by ',' stored as textfile location '/user/hue/nyse/stock_aggregates';
- Populate the table with data using the following query:
  - insert overwrite table yearly_aggregates select a.symbol, year(a.trade_date), max(a.high), min(a.low), avg(a.close), sum(b.dividend) from price_data a left outer join dividends_data b on (a.symbol = b.symbol and a.trade_date = b.trade_date) group by a.symbol, year(a.trade_date);



Query Results

127.0.0.1:8000/beeswax/results/23/0?context=design%3A16

Query Editor | My Queries | Saved Queries | History | Databases | Tables | Settings

Query Results: Unsaved Query

DOWNLOADS

Download as CSV

Download as XLS

Enable visualization

Save

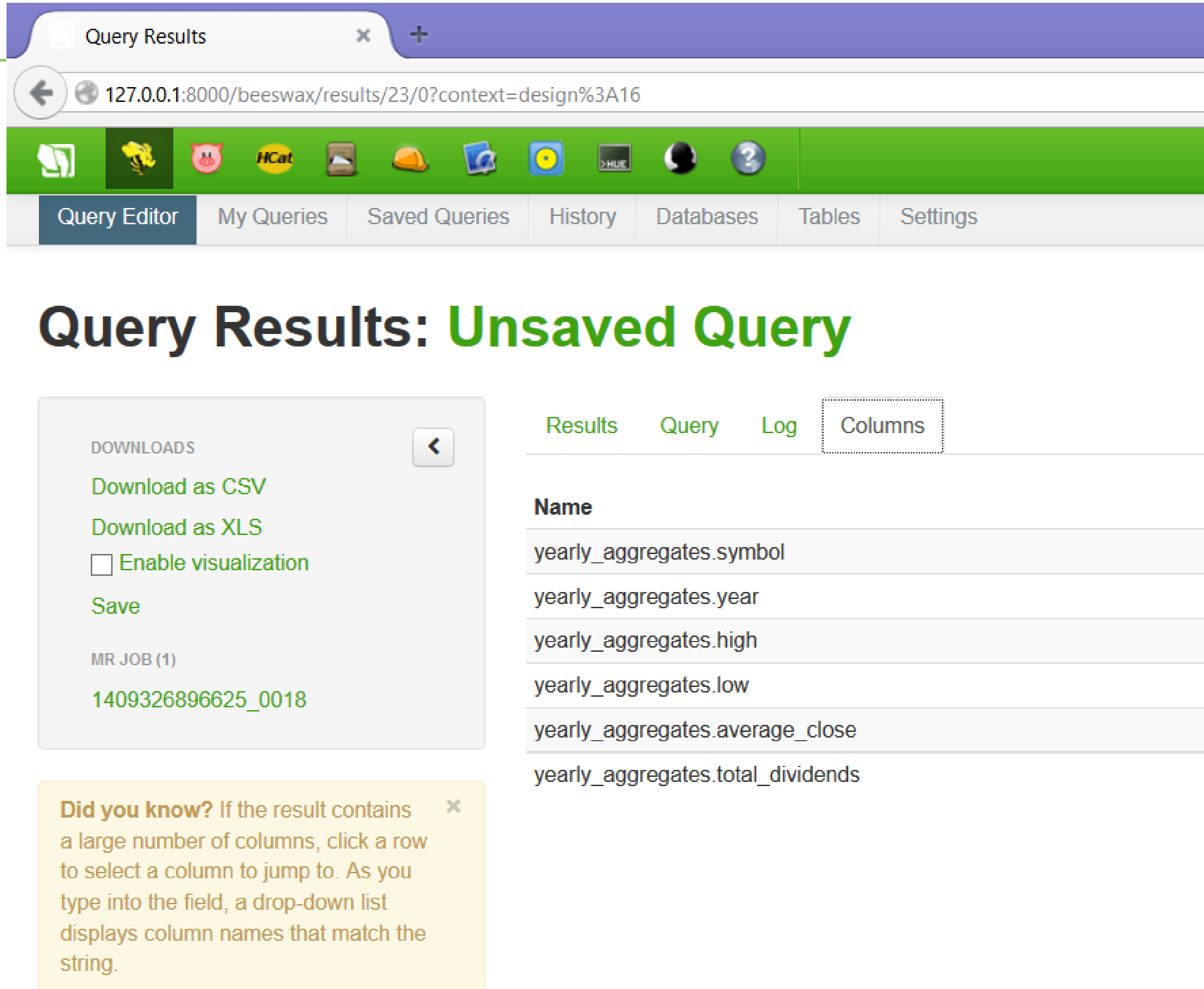
MR JOB (1)

1409326896625_0018

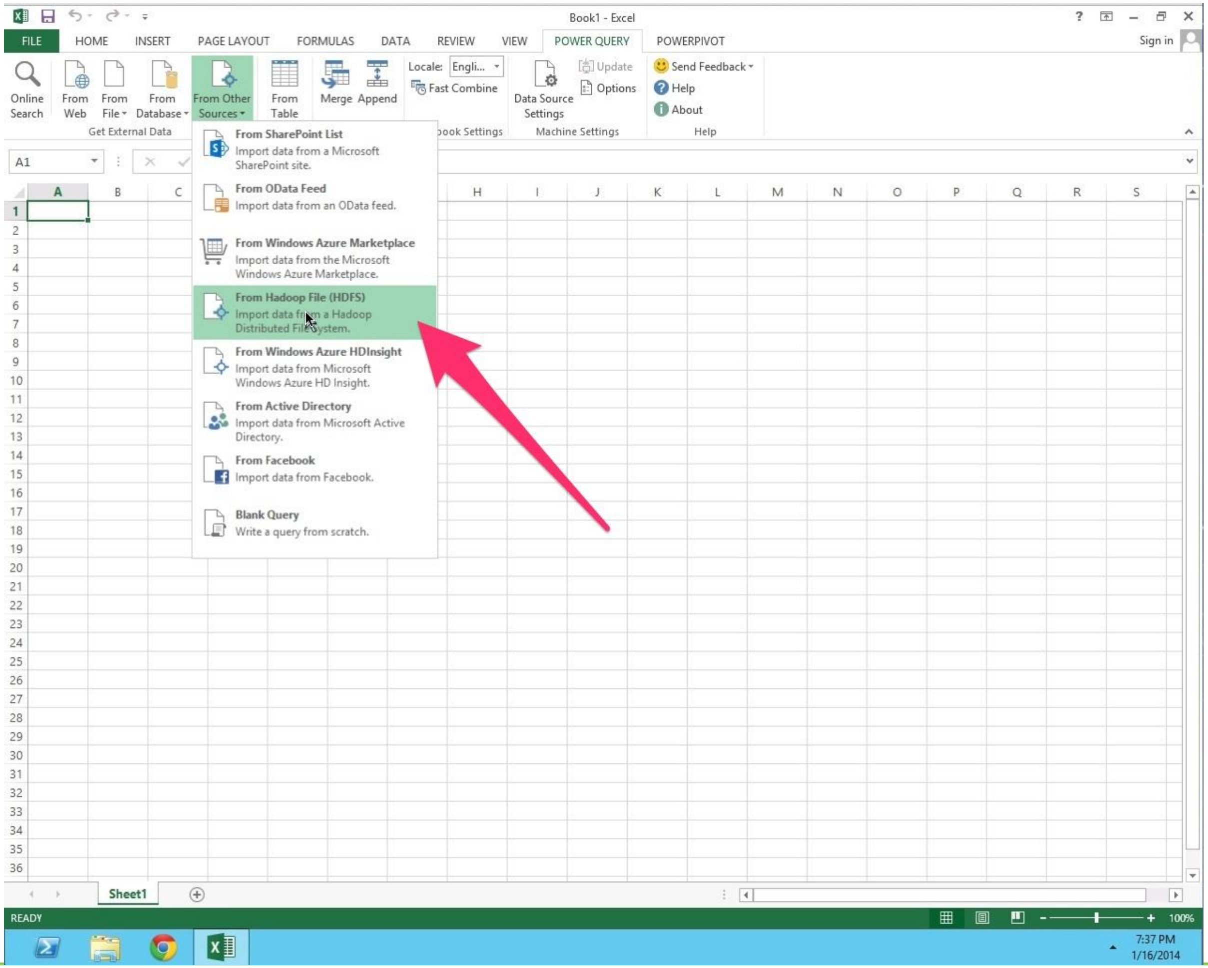
Did you know? If the result contains a large number of columns, click a row to select a column to jump to. As you type into the field, a drop-down list displays column names that match the string.

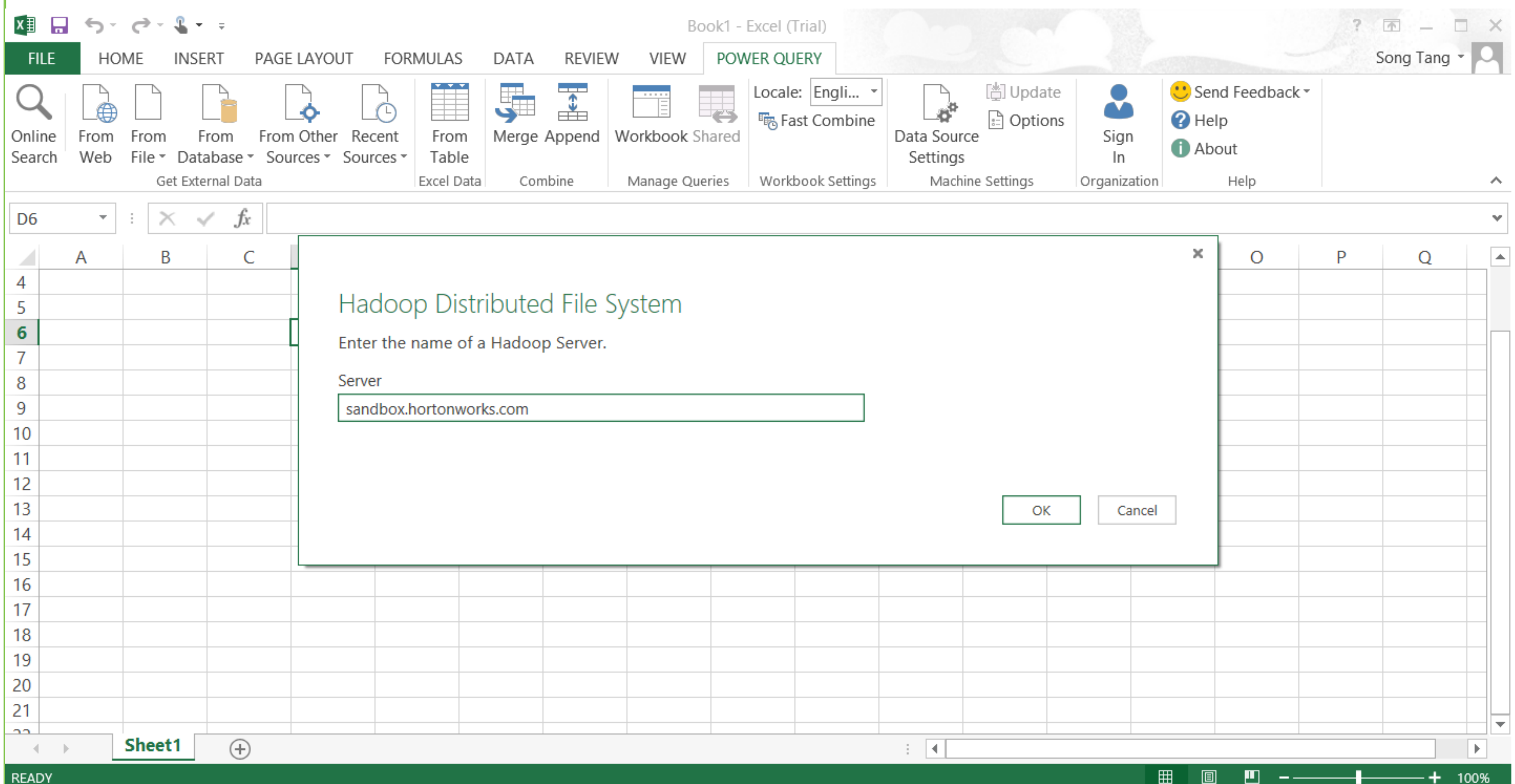
Results | Query | Log | Columns

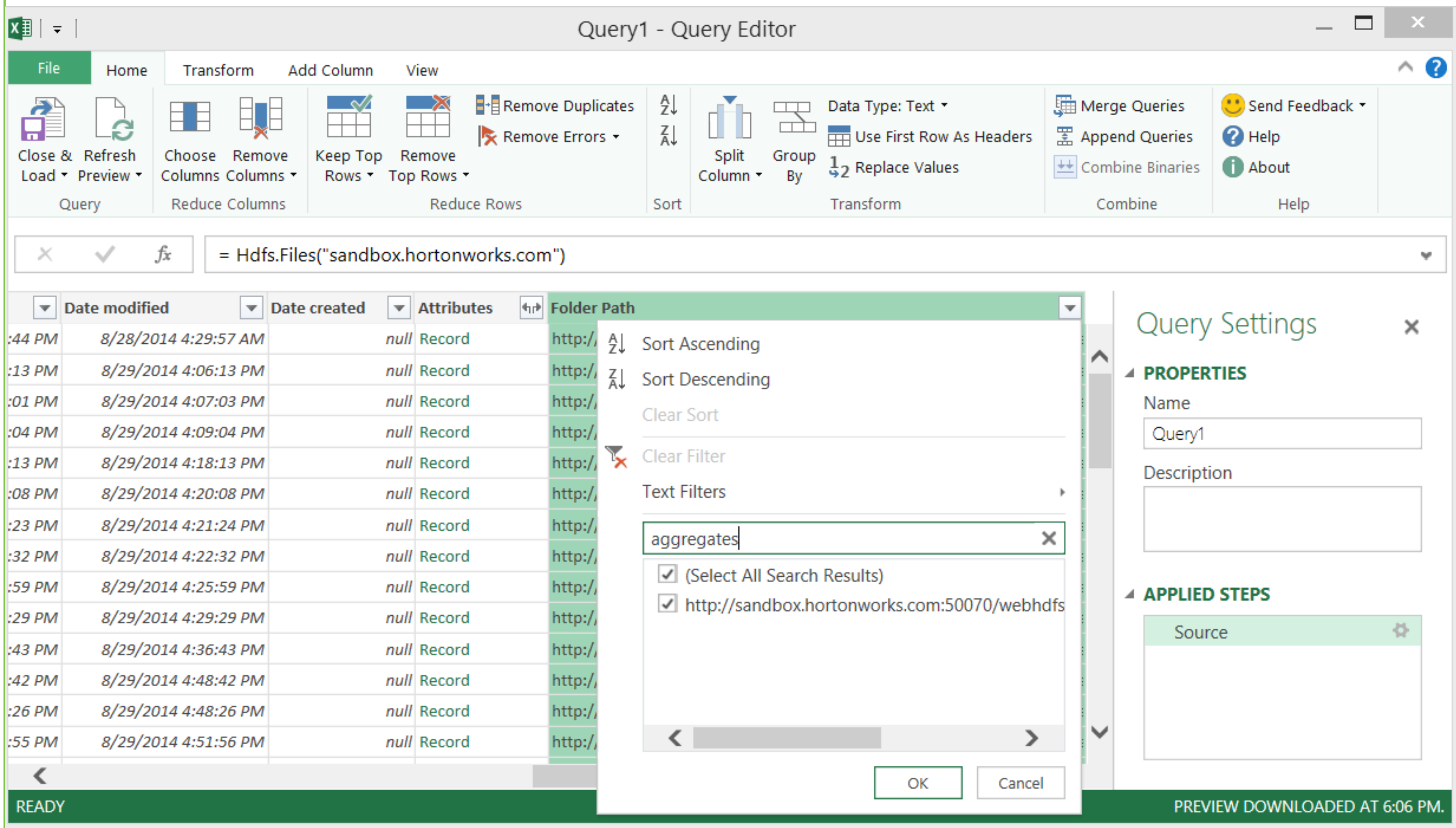
	yearly_aggregates.symbol	yearly_aggregates.year	yearly_aggregates.high	yearly_aggregates.low	yearly_aggregates.av
34	IBM	1996	166.0	83.12	117.12981
35	IBM	1997	179.25	81.75	120.45166
36	IBM	1998	189.94	95.62	124.45246
37	IBM	1999	246.0	89.0	145.05965
38	IBM	2000	134.94	80.06	110.62734
39	IBM	2001	124.7	83.75	107.525764
40	IBM	2002	126.39	54.01	84.1902
41	IBM	2003	94.54	73.17	85.112305
42	IBM	2004	100.43	81.9	90.77044
43	IBM	2005	99.1	71.85	83.786194
44	IBM	2006	97.88	72.73	83.10697
45	IBM	2007	121.46	88.77	105.8037
46	IBM	2008	130.93	69.5	110.0449
47	IBM	2009	132.85	81.76	109.27516
48	IBM	2010	134.25	121.74	127.82


Query Results
127.0.0.1:8000/beeswax/results/23/0?context=design%3A16
HCat
Query Editor
My Queries
Saved Queries
History
Databases
Tables
Settings
Query Results: Unsaved Query
DOWNLOADS
Download as CSV
Download as XLS
Enable visualization
Save
MR JOB (1)
1409326896625_0018
Did you know? If the result contains a large number of columns, click a row to select a column to jump to. As you type into the field, a drop-down list displays column names that match the string.
Results
Query
Log
Columns
Name
yearly_aggregates.symbol
yearly_aggregates.year
yearly_aggregates.high
yearly_aggregates.low
yearly_aggregates.average_close
yearly_aggregates.total_dividends

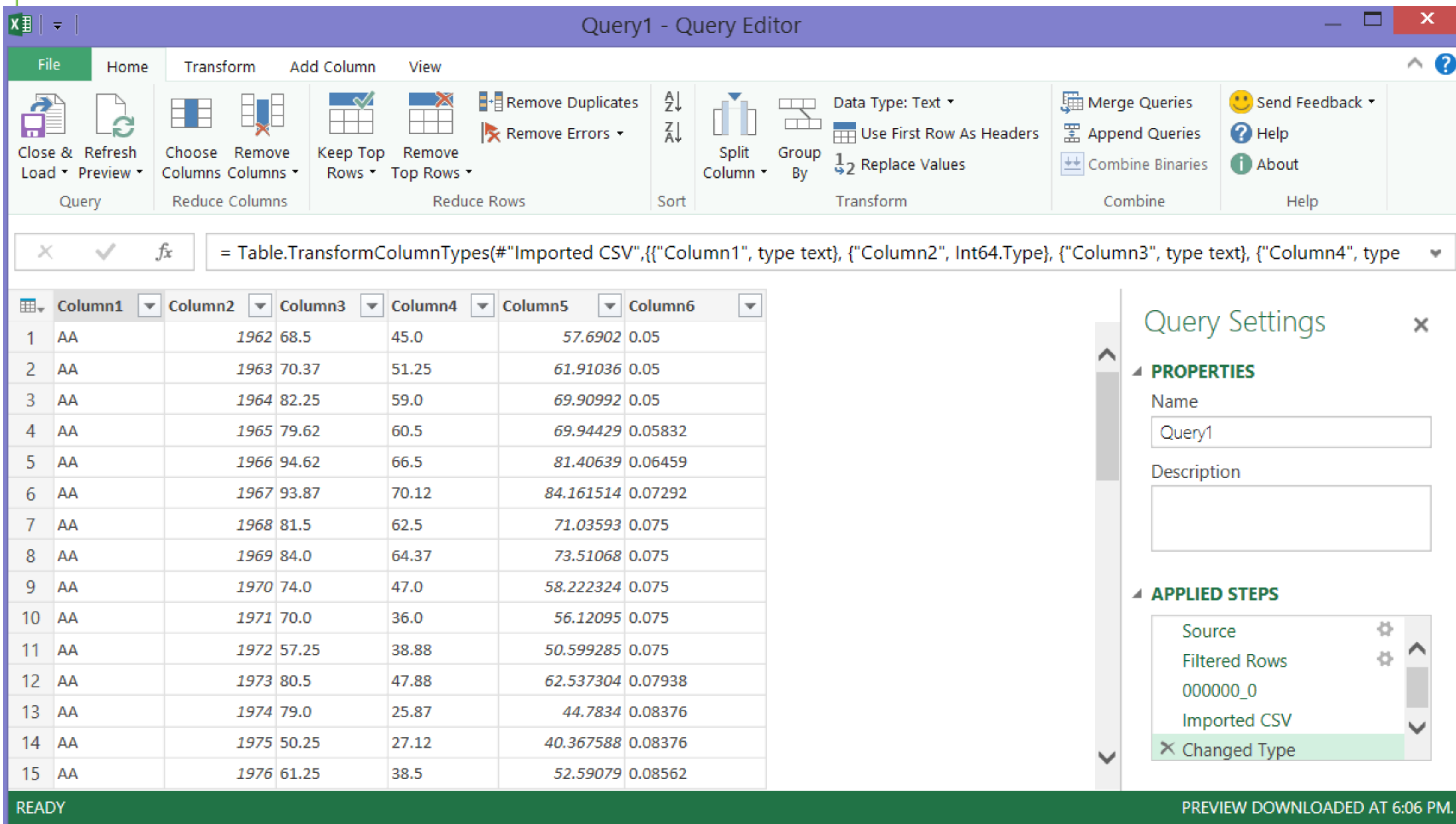
Export Hadoop resultset to Excel Power Query




Book1 - Excel (Trial)
FILE
HOME
INSERT
PAGE LAYOUT
FORMULAS
DATA
REVIEW
VIEW
POWER QUERY
Song Tang
Online Search
From Web
From File
From Database
From Other Sources
Recent Sources
Get External Data
From Table
Excel Data
Merge
Append
Combine
Workbook
Shared
Manage Queries
Locale: Engli...
Fast Combine
Workbook Settings
Data Source Settings
Update
Options
Machine Settings
Sign In
Organization
Send Feedback
Help
About
Help
D6
Hadoop Distributed File System
Enter the name of a Hadoop Server.
Server
sandbox.hortonworks.com
OK
Cancel
Sheet1
READY
100%



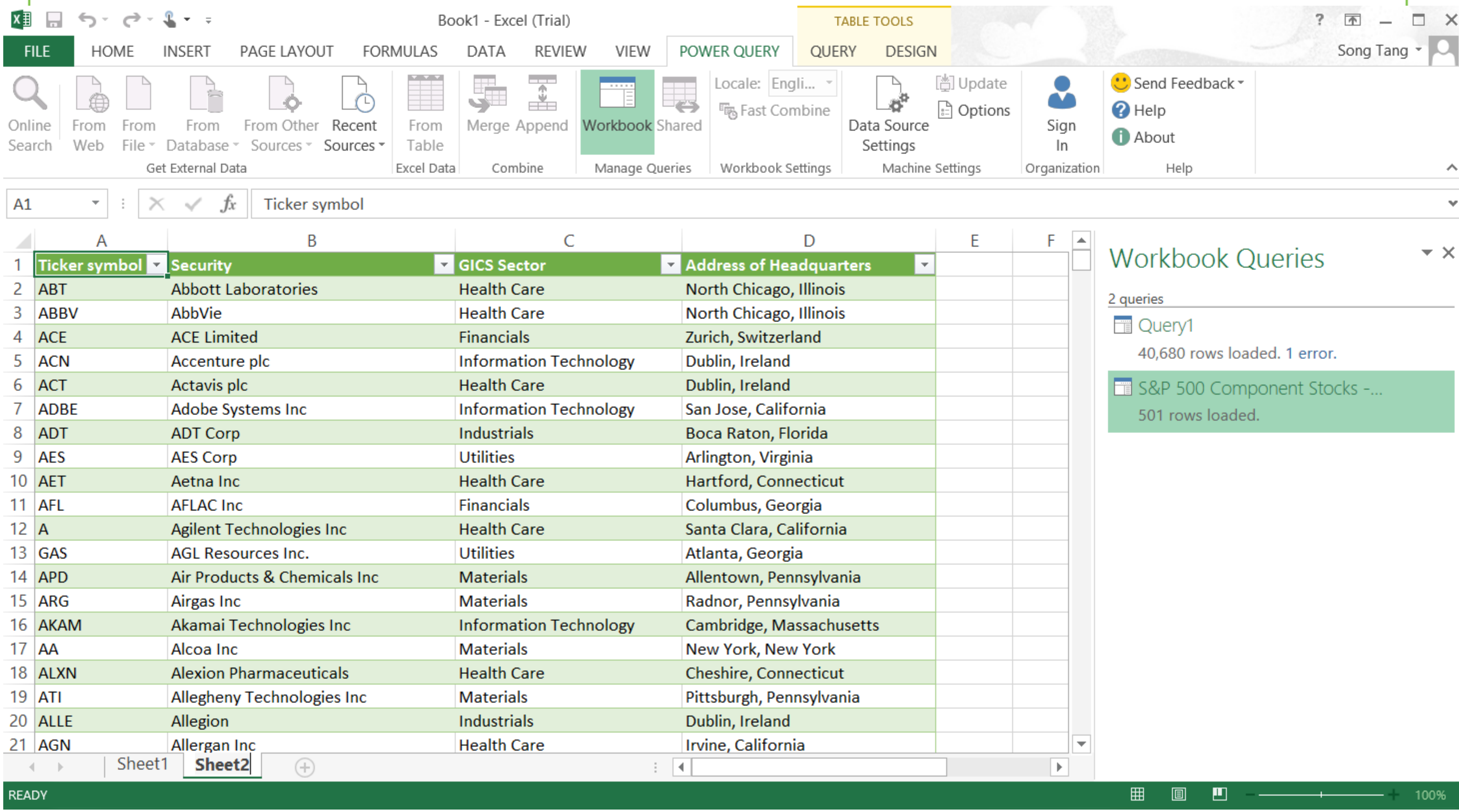
Query1 - Query Editor
File
Home
Transform
Add Column
View
Close & Load
Refresh Preview
Query
Choose Columns
Remove Columns
Reduce Columns
Keep Top Rows
Remove Top Rows
Remove Duplicates
Remove Errors
Reduce Rows
Sort
Split Column
Group By
Data Type: Text
Use First Row As Headers
Replace Values
Transform
Merge Queries
Append Queries
Combine Binaries
Combine
Send Feedback
Help
About
Help
= Hdfs.Files("sandbox.hortonworks.com")
Date modified
Date created
Attributes
Folder Path
8/28/2014 4:29:57 AM
8/29/2014 4:06:13 PM
8/29/2014 4:07:03 PM
8/29/2014 4:09:04 PM
8/29/2014 4:18:13 PM
8/29/2014 4:20:08 PM
8/29/2014 4:21:24 PM
8/29/2014 4:22:32 PM
8/29/2014 4:25:59 PM
8/29/2014 4:29:29 PM
8/29/2014 4:36:43 PM
8/29/2014 4:48:42 PM
8/29/2014 4:48:26 PM
8/29/2014 4:51:56 PM
null
Record
Sort Ascending
Sort Descending
Clear Sort
Clear Filter
Text Filters
aggregates
(Select All Search Results)
http://sandbox.hortonworks.com:50070/webhdfs
OK
Cancel
Query Settings
PROPERTIES
Name
Query1
Description
APPLIED STEPS
Source
READY
PREVIEW DOWNLOADED AT 6:06 PM.



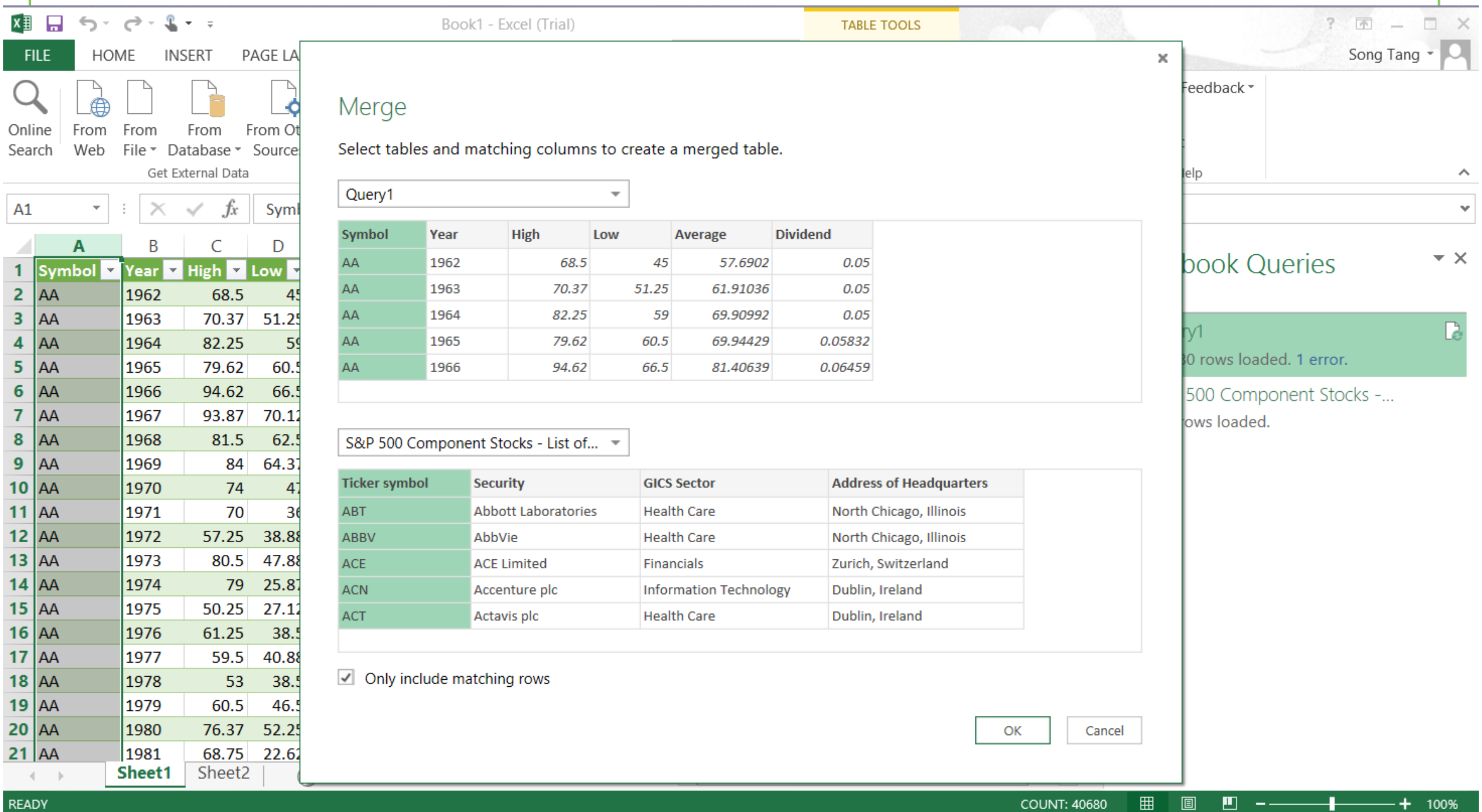
	Column1	Column2	Column3	Column4	Column5	Column6
1	AA	1962	68.5	45.0	57.6902	0.05
2	AA	1963	70.37	51.25	61.91036	0.05
3	AA	1964	82.25	59.0	69.90992	0.05
4	AA	1965	79.62	60.5	69.94429	0.05832
5	AA	1966	94.62	66.5	81.40639	0.06459
6	AA	1967	93.87	70.12	84.161514	0.07292
7	AA	1968	81.5	62.5	71.03593	0.075
8	AA	1969	84.0	64.37	73.51068	0.075
9	AA	1970	74.0	47.0	58.222324	0.075
10	AA	1971	70.0	36.0	56.12095	0.075
11	AA	1972	57.25	38.88	50.599285	0.075
12	AA	1973	80.5	47.88	62.537304	0.07938
13	AA	1974	79.0	25.87	44.7834	0.08376
14	AA	1975	50.25	27.12	40.367588	0.08376
15	AA	1976	61.25	38.5	52.59079	0.08562

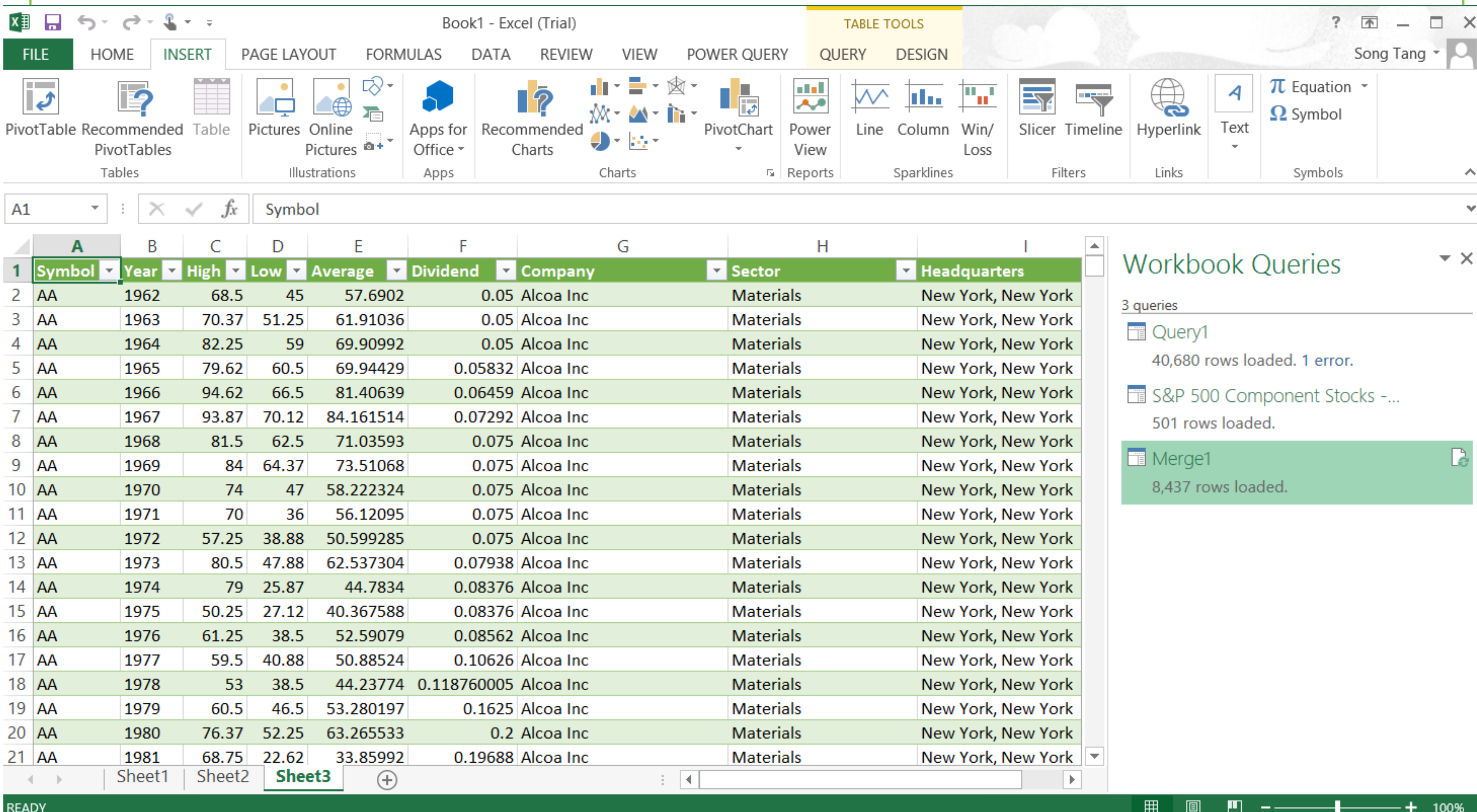
Symbol	Year	High	Low	Average	Dividend
AA	1962	68.5	45	57.6902	0.05
AA	1963	70.37	51.25	61.91036	0.05
AA	1964	82.25	59	69.90992	0.05
AA	1965	79.62	60.5	69.94429	0.05832
AA	1966	94.62	66.5	81.40639	0.06459
AA	1967	93.87	70.12	84.161514	0.07292
AA	1968	81.5	62.5	71.03593	0.075
AA	1969	84	64.37	73.51068	0.075
AA	1970	74	47	58.222324	0.075
AA	1971	70	36	56.12095	0.075
AA	1972	57.25	38.88	50.599285	0.075
AA	1973	80.5	47.88	62.537304	0.07938
AA	1974	79	25.87	44.7834	0.08376
AA	1975	50.25	27.12	40.367588	0.08376
AA	1976	61.25	38.5	52.59079	0.08562
AA	1977	59.5	40.88	50.88524	0.10626
AA	1978	53	38.5	44.23774	0.118760005
AA	1979	60.5	46.5	53.280197	0.1625
AA	1980	76.37	52.25	63.265533	0.2
AA	1981	68.75	22.62	33.85992	0.19688

Enhancing the Hadoop resultset with Internet Data



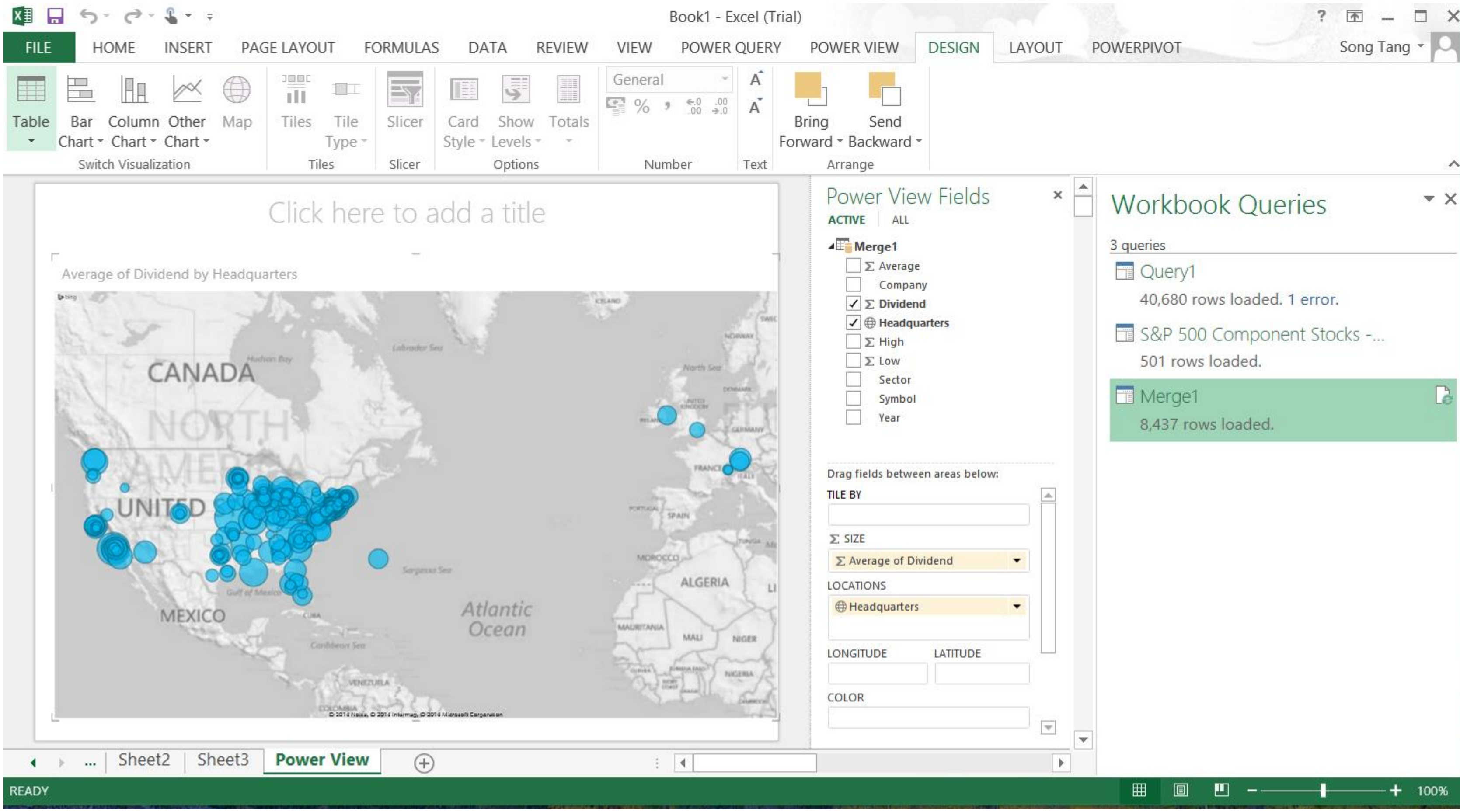
Ticker symbol	Security	GICS Sector	Address of Headquarters
ABT	Abbott Laboratories	Health Care	North Chicago, Illinois
ABBV	AbbVie	Health Care	North Chicago, Illinois
ACE	ACE Limited	Financials	Zurich, Switzerland
ACN	Accenture plc	Information Technology	Dublin, Ireland
ACT	Actavis plc	Health Care	Dublin, Ireland
ADBE	Adobe Systems Inc	Information Technology	San Jose, California
ADT	ADT Corp	Industrials	Boca Raton, Florida
AES	AES Corp	Utilities	Arlington, Virginia
AET	Aetna Inc	Health Care	Hartford, Connecticut
AFL	AFLAC Inc	Financials	Columbus, Georgia
A	Agilent Technologies Inc	Health Care	Santa Clara, California
GAS	AGL Resources Inc.	Utilities	Atlanta, Georgia
APD	Air Products & Chemicals Inc	Materials	Allentown, Pennsylvania
ARG	Airgas Inc	Materials	Radnor, Pennsylvania
AKAM	Akamai Technologies Inc	Information Technology	Cambridge, Massachusetts
AA	Alcoa Inc	Materials	New York, New York
ALXN	Alexion Pharmaceuticals	Health Care	Cheshire, Connecticut
ATI	Allegheny Technologies Inc	Materials	Pittsburgh, Pennsylvania
ALLE	Allegion	Industrials	Dublin, Ireland
AGN	Allergan Inc	Health Care	Irvine, California


Book1 - Excel (Trial)
TABLE TOOLS
FILE
HOME
INSERT
PAGE LA
Online Search
From Web
From File
From Database
From Ot Source
Get External Data
Song Tang
Feedback
Merge
Select tables and matching columns to create a merged table.
Query1
Symbol Year High Low Average Dividend
AA 1962 68.5 45 57.6902 0.05
AA 1963 70.37 51.25 61.91036 0.05
AA 1964 82.25 59 69.90992 0.05
AA 1965 79.62 60.5 69.94429 0.05832
AA 1966 94.62 66.5 81.40639 0.06459
S&P 500 Component Stocks - List of...
Ticker symbol Security GICS Sector Address of Headquarters
ABT Abbott Laboratories Health Care North Chicago, Illinois
ABBV AbbVie Health Care North Chicago, Illinois
ACE ACE Limited Financials Zurich, Switzerland
ACN Accenture plc Information Technology Dublin, Ireland
ACT Actavis plc Health Care Dublin, Ireland
Only include matching rows
OK
Cancel
book Queries
rows loaded. 1 error.
500 Component Stocks -...
ows loaded.
Sheet1
Sheet2
READY
COUNT: 40680
100%



Symbol	Year	High	Low	Average	Dividend	Company	Sector	Headquarters
AA	1962	68.5	45	57.6902	0.05	Alcoa Inc	Materials	New York, New York
AA	1963	70.37	51.25	61.91036	0.05	Alcoa Inc	Materials	New York, New York
AA	1964	82.25	59	69.90992	0.05	Alcoa Inc	Materials	New York, New York
AA	1965	79.62	60.5	69.94429	0.05832	Alcoa Inc	Materials	New York, New York
AA	1966	94.62	66.5	81.40639	0.06459	Alcoa Inc	Materials	New York, New York
AA	1967	93.87	70.12	84.161514	0.07292	Alcoa Inc	Materials	New York, New York
AA	1968	81.5	62.5	71.03593	0.075	Alcoa Inc	Materials	New York, New York
AA	1969	84	64.37	73.51068	0.075	Alcoa Inc	Materials	New York, New York
AA	1970	74	47	58.222324	0.075	Alcoa Inc	Materials	New York, New York
AA	1971	70	36	56.12095	0.075	Alcoa Inc	Materials	New York, New York
AA	1972	57.25	38.88	50.599285	0.075	Alcoa Inc	Materials	New York, New York
AA	1973	80.5	47.88	62.537304	0.07938	Alcoa Inc	Materials	New York, New York
AA	1974	79	25.87	44.7834	0.08376	Alcoa Inc	Materials	New York, New York
AA	1975	50.25	27.12	40.367588	0.08376	Alcoa Inc	Materials	New York, New York
AA	1976	61.25	38.5	52.59079	0.08562	Alcoa Inc	Materials	New York, New York
AA	1977	59.5	40.88	50.88524	0.10626	Alcoa Inc	Materials	New York, New York
AA	1978	53	38.5	44.23774	0.118760005	Alcoa Inc	Materials	New York, New York
AA	1979	60.5	46.5	53.280197	0.1625	Alcoa Inc	Materials	New York, New York
AA	1980	76.37	52.25	63.265533	0.2	Alcoa Inc	Materials	New York, New York
AA	1981	68.75	22.62	33.85992	0.19688	Alcoa Inc	Materials	New York, New York

Visualizing S & P 500 dividend geographic distribution



Stock Price Prediction using Hadoop MapReduce

- Using Technical Indicators SMA (Simple moving average), EMA (Exponential moving average) and OBV (On Balance Volume).
- Normalizing the stock data and performing BPNN (Back-Propagation Neural Networks) algorithm.
- Using Hadoop MapReduce to develop virtual data nodes for parallel processing of the data using Neural Network for time efficient forecasting of the stock price movement.

References

- *Big Data in Trading and Risk Management , Industry Insight on Opportunities, Applications and Challenges. An Industry Survey and Briefing Conducted and written by BigData for Finance, Nov. 2012*
- *Data Science for Business: What you need to know about data mining and data-analytic thinking. Foster Provost and Tom Fawcett,* O'Reilly Media, 2013, ISBN-10: 1449361323
- *Big Data: A Revolution That Will Transform How We Live, Work, and Think. Viktor Mayer-Schönberger and Kenneth Cukier. Eamon Dolan/Houghton Mifflin Harcourt, 2013, ISBN-10: 0544002695*
- *Pentaho, BI for Big Data, Beyond the Hype.*
- *An Introduction to Big Data, Apache Hadoop, and Cloudera, by Ian Wrigley, Curriculum Manager, Cloudera*

References (continous)

- *From Raw Data to Insight using HDP and Microsoft Business Intelligence, hortonworks.com/hadoop-tutorial/partner-tutorial-microsoft/*
- *Uisng Hadoop for Value At Risk Calculation, blog.octo.com/en/using-hadoop-for-value-at-risk-calculation-part-1/*
- *Stock Exchange Iforcasting Using Hadoop Map-Reduce Technique, by K. Sahu, R. Pawar, S. Tilekar and R. Satpute, International Journal of Advancements in Research & Technology, Volume 2, Issue4, April 2013, p380-p382*
- *MapReduce and Hadoop Distributed File System, Dr. Bina Ramamurthy, CSE Department, University at Buffalo (SUNY), www.cse.buffalo.edu/faculty/bina*
- *A Very Brief Introduction to MapReduce, Diana MacLean for CS448G, 2011*