Module 3: Drilling Data


APACHE
DRILL

A challenge to set the stage

A challenge to set the stage

firewall1.ssdlog

```
Dec 10 22:36:39    sshd[80124]: Failed password for root from 114.80.246.132 port 1093 ssh2
Dec 10 22:36:37    sshd[79627]: Failed password for root from 114.80.246.132 port 3163 ssh2
Dec 10 22:36:35    sshd[79027]: Failed password for root from 114.80.246.132 port 1391 ssh2
Dec 10 22:36:34    sshd[79440]: Failed password for root from 114.80.246.132 port 4314 ssh2
Dec 10 22:36:32    sshd[78761]: Failed password for root from 114.80.246.132 port 1882 ssh2
Dec 10 22:36:32    sshd[78246]: Failed password for root from 114.80.246.132 port 2627 ssh2
```

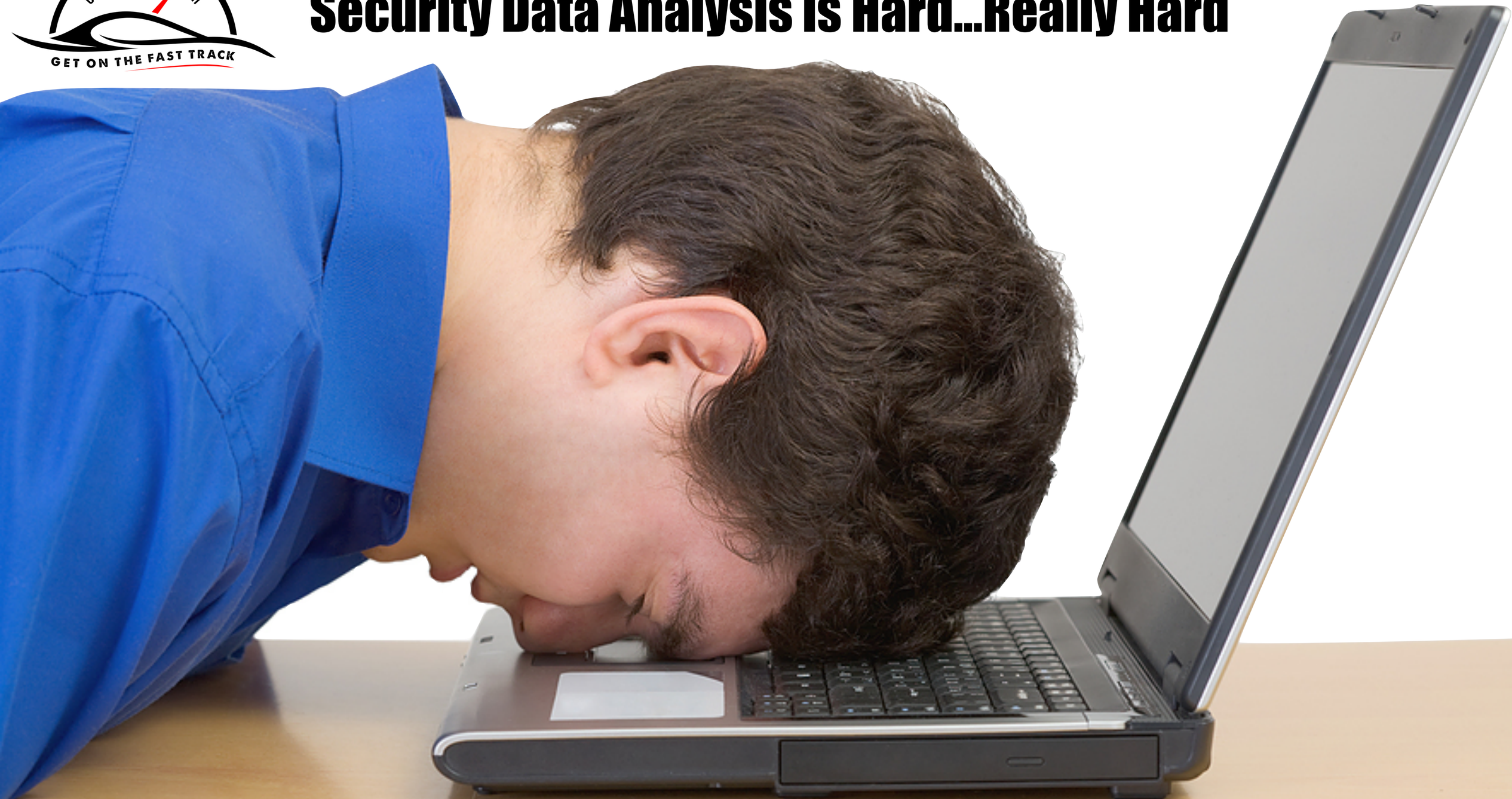
1. Which IPs attempted to login to this server?
2. What countries were the most lockouts from?

The Problem: Analyzing Security Data is Hard

GTK CYBER

GET ON THE FAST TRACK

Security Data Analysis is Hard...Really Hard



Why?

Security Data comes in Many Forms

Security Data Comes in Many Forms

- Standard Types such as JSON/CSV/XML
- Log Files: Event Logs, Database, Web Server etc.
- Syslog
- PCAP / PCAP-NG: Binary, raw network traffic

Security Data Lives In Many Places

- File systems
- Cloud Storage: Azure / S3
- Databases
- Real Time Event Streams
- Other sources

Few tools can effectively analyze these data types effectively

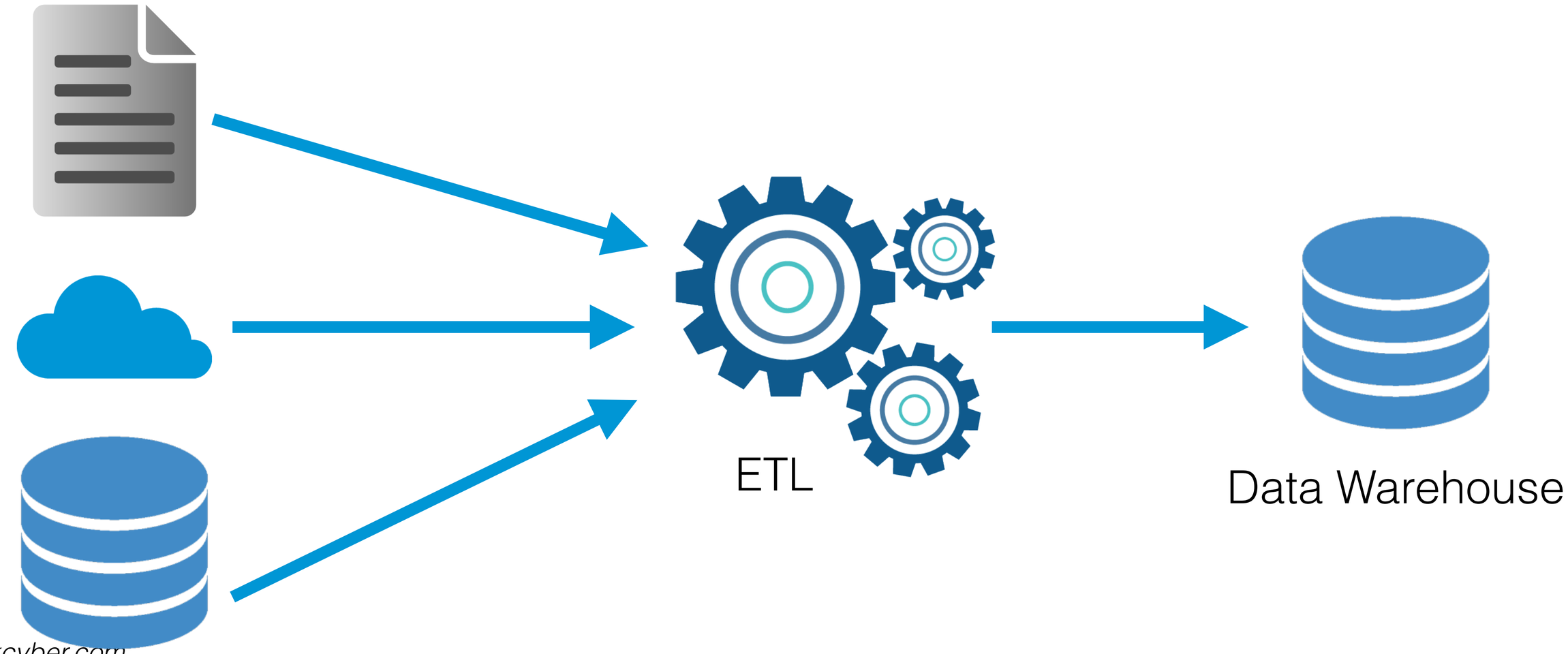
Even fewer tools can effectively analyze **ALL** these data types effectively

Splunk and ELK are solid SIEM platforms

Wireshark is good for PCAP analysis

Security data is not arranged in an optimal way for ad-hoc analysis

Security data is not arranged in an optimal way for ad-hoc analysis



ETL is expensive and wasteful

Analytics teams spend between 50%-90% of their time preparing their data.

76% of Data Scientists say this is the **least enjoyable part** of their job.

http://visit.crowdflower.com/rs/416-ZBE-142/images/CrowdFlower_DataScienceReport_2016.pdf

The ETL Process **consumes the most time** and **contributes almost no value** to the end product.

GET ON THE FAST TRACK

8 Wastes

The 8 Wastes are eight types of process obstacles that get in the way of providing value to the customer.

Defects

Efforts caused by rework, scrap, and incorrect information.

Overproduction

Production that is more than needed or before it is needed.

Waiting

Wasted time waiting for the next step in a process.

Non-Utilized Talent

Underutilizing people's talents, skills, & knowledge.

Transportation

Unnecessary movements of products & materials.

Inventory

Excess products and materials not being processed.

Motion

Unnecessary movements by people (e.g., walking).

Extra-Processing

More work or higher quality than is required by the customer.

goLEANSIXSIGMA.com

https://goleansixsigma.com/8-wastes/

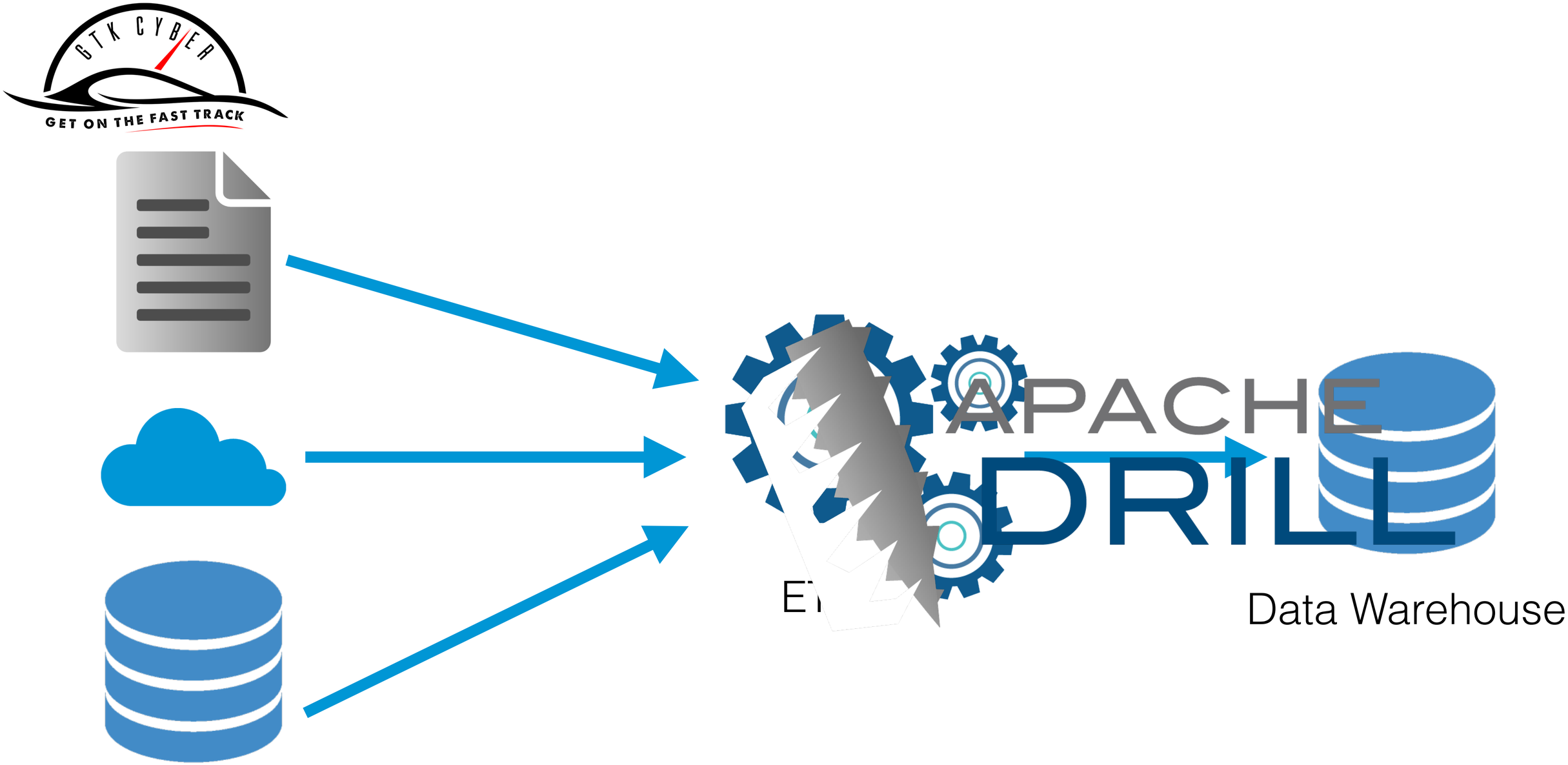
So where does Drill fit in?

Apache Drill is an SQL Engine for self-describing data.

Drill lets you query anything*, wherever it is*, no matter its size** using standard SQL.

* well.. almost anything
** within reason


Drill acts as a universal
translator for data.


GTK CYBER
GET ON THE FAST TRACK
APACHE
DRILL
Data Warehouse


GTK CYBER
GET ON THE FAST TRACK


APACHE
DRILL
Superset

Whoa! I can query PCAP as if it was a database!!

Table

gtkcyber.com

So let's take a look...

For instructions on how to use Apache Drill with Apache Superset (Incubating)

http://thedataist.com/**visualize-anything-with-superset-and-drill/**


```
158.222.5.157 - - [25/Oct/2015:04:24:37 +0100] "GET /acl_users/
credentials_cookie_auth/require_login?came_from=http%3A//
howto.basjes.nl/join_form HTTP/1.1" 200 10716 "http://
howto.basjes.nl/join_form" "Mozilla/5.0 (Windows NT 6.3; WOW64;
rv:34.0) Gecko/20100101 Firefox/34.0 AlexaToolbar/alxf-2.21"

158.222.5.157 - - [25/Oct/2015:04:24:39 +0100] "GET /login_form
HTTP/1.1" 200 10543 "http://howto.basjes.nl/" "Mozilla/5.0
(Windows NT 6.3; WOW64; rv:34.0) Gecko/20100101 Firefox/34.0
AlexaToolbar/alxf-2.21"
```

`hackers-access.httpd`

request_referer_ref	VARCHAR
request_receive_time_last_time	VARCHAR
request_firstline_uri_protocol	VARCHAR
request_receive_time_microsecond	BIGINT
request_receive_time_last_microsecond__utc	BIGINT
request_firstline_original_uri_query_$	UserDefinedType
request_firstline_original_protocol	VARCHAR
request_firstline_original_uri_host	VARCHAR
request_referer_host	VARCHAR
request_receive_time_month__utc	BIGINT
request_receive_time_last_minute	BIGINT
request_firstline_protocol_version	VARCHAR
request_receive_time_time__utc	VARCHAR
request_referer_last_ref	VARCHAR
request_receive_time_last_timezone	VARCHAR
request_receive_time_last_weekofweekyear	BIGINT
request_referer_last	VARCHAR
request_receive_time_minute	BIGINT
connection_client_host_last	VARCHAR
request_receive_time_last_millisecond__utc	BIGINT
request_firstline_original_uri	VARCHAR
request_firstline	VARCHAR
request_receive_time_nanosecond	BIGINT
request_receive_time_last_millisecond	BIGINT
request_receive_time_day	BIGINT
request_referer_port	BIGINT
request_firstline_original_uri_port	BIGINT
request_receive_time_year	BIGINT
request_receive_time_last_date	VARCHAR
request_referer_query_$	UserDefinedType

https://github.com/nielsbasjes/logparser/tree/master/examples/demolog

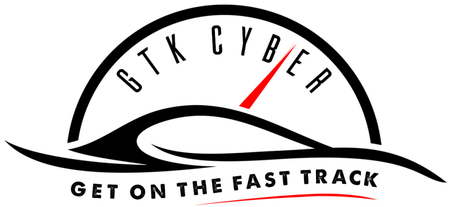
request_referer_last	request_receive_time_minute	connection_client_host_last	request_receive_time_last_millisecond__utc	request_firstline_original_uri
http://howto.basjes.nl/	11	195.154.46.135	0	/linux/doing-pxe-without-dhcp-control
http://howto.basjes.nl/	11	23.95.237.180	0	/join_form
http://howto.basjes.nl/join_form	11	23.95.237.180	0	/join_form
http://howto.basjes.nl/	24	158.222.5.157	0	/join_form
http://howto.basjes.nl/join_form	24	158.222.5.157	0	/join_form
http://howto.basjes.nl/join_form	24	158.222.5.157	0	/acl_users/credentials_cookie_auth/require_login?came_from=http%
http://howto.basjes.nl/	24	158.222.5.157	0	/login_form
http://howto.basjes.nl/login_form	24	158.222.5.157	0	/login_form
http://howto.basjes.nl/	32	5.39.5.5	0	/join_form
http://howto.basjes.nl/	34	180.180.64.16	0	/linux/doing-pxe-without-dhcp-control
http://howto.basjes.nl/	34	180.180.64.16	0	/join_form
http://howto.basjes.nl/join_form	34	180.180.64.16	0	/join_form
http://howto.basjes.nl/join_form	34	180.180.64.16	0	/acl_users/credentials_cookie_auth/require_login?came_from=http%
http://howto.basjes.nl/	34	180.180.64.16	0	/login_form

https://github.com/nielsbasjes/logparser/tree/master/examples/demolog

Who is Trying to Hack This Site?

- The url /join_form is not public so anyone attempting to access this site, so almost anyone trying to access this probably a hacker...
- Let's see who is looking...

https://github.com/nielsbasjes/logparser/tree/master/examples/demolog



Who is Trying to Hack This Site?

```
SELECT request_receive_time,
connection_client_host,
request_useragent
FROM dfs.test.`hackers-access.httpd`
WHERE request_firstline_uri = '/join_form'
```

https://github.com/nielsbasjes/logparser/tree/master/examples/demolog

Who is Trying to Hack This Site?

```
SELECT request_receive_time,
connection_client_host,
request_useragent
FROM dfs.test.`hackers-access.httpd`
WHERE request_firstline_uri = '/join_form'
```

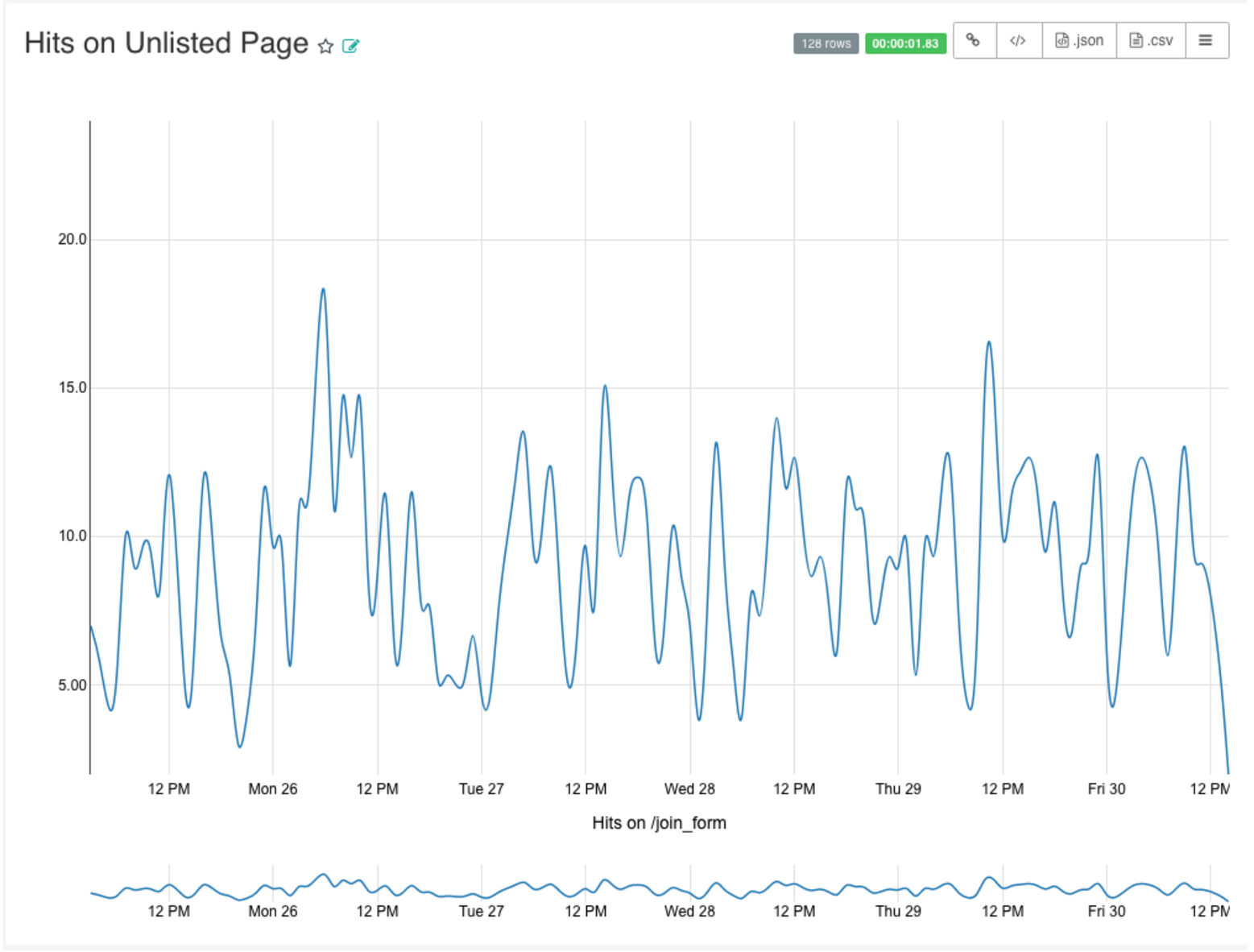
Results | Query History | Preview: `hackers-access.httpd`

237.180

Explore | .CSV | Clipboard | Search Results

request_receive_time	connection_client_host	request_useragent
2015-10-25T03:11:26	23.95.237.180	Mozilla/5.0 (Windows NT 5.1; rv:35.0) Gecko/20100101 Firefox/35.0
2015-10-25T03:11:27	23.95.237.180	Mozilla/5.0 (Windows NT 5.1; rv:35.0) Gecko/20100101 Firefox/35.0
2015-10-25T03:24:31	158.222.5.157	Mozilla/5.0 (Windows NT 6.3; WOW64; rv:34.0) Gecko/20100101 Firefox/34.0 AlexaToolbar/alxf-2.21
2015-10-25T03:24:32	158.222.5.157	Mozilla/5.0 (Windows NT 6.3; WOW64; rv:34.0) Gecko/20100101 Firefox/34.0 AlexaToolbar/alxf-2.21
2015-10-25T03:32:22	5.39.5.5	Mozilla/5.0 (Windows NT 5.1; rv:34.0) Gecko/20100101 Firefox/34.0
2015-10-25T03:34:40	180.180.64.16	Mozilla/5.0 (Windows NT 5.1; rv:35.0) Gecko/20100101 Firefox/35.0
2015-10-25T03:34:42	180.180.64.16	Mozilla/5.0 (Windows NT 5.1; rv:35.0) Gecko/20100101 Firefox/35.0
2015-10-25T04:06:42	89.42.237.71	Mozilla/5.0 (Windows NT 5.1; rv:34.0) Gecko/20100101 Firefox/34.0
2015-10-25T04:06:43	89.42.237.71	Mozilla/5.0 (Windows NT 5.1; rv:34.0) Gecko/20100101 Firefox/34.0

Who is Trying to Hack This Site?



Let's take a look at those IP addresses shall we?

Who is Trying to Hack This Site?

```
SELECT request_receive_time,
connection_client_host,
request_useragent
FROM dfs.test.`hackers-access.httpd`
WHERE request_firstline_uri = '/join_form'
```

Results | Query History | Preview: `hackers-access.httpd`

237.180

Explore | .CSV | Clipboard | Search Results

request_receive_time	connection_client_host	request_useragent
2015-10-25T03:11:26	23.95.237.180	Mozilla/5.0 (Windows NT 5.1; rv:35.0) Gecko/20100101 Firefox/35.0
2015-10-25T03:11:27	23.95.237.180	Mozilla/5.0 (Windows NT 5.1; rv:35.0) Gecko/20100101 Firefox/35.0
2015-10-25T03:24:31	158.222.5.157	Mozilla/5.0 (Windows NT 6.3; WOW64; rv:34.0) Gecko/20100101 Firefox/34.0 AlexaToolbar/alxf-2.21
2015-10-25T03:24:32	158.222.5.157	Mozilla/5.0 (Windows NT 6.3; WOW64; rv:34.0) Gecko/20100101 Firefox/34.0 AlexaToolbar/alxf-2.21
2015-10-25T03:32:22	5.39.5.5	Mozilla/5.0 (Windows NT 5.1; rv:34.0) Gecko/20100101 Firefox/34.0
2015-10-25T03:34:40	180.180.64.16	Mozilla/5.0 (Windows NT 5.1; rv:35.0) Gecko/20100101 Firefox/35.0
2015-10-25T03:34:42	180.180.64.16	Mozilla/5.0 (Windows NT 5.1; rv:35.0) Gecko/20100101 Firefox/35.0
2015-10-25T04:06:42	89.42.237.71	Mozilla/5.0 (Windows NT 5.1; rv:34.0) Gecko/20100101 Firefox/34.0
2015-10-25T04:06:43	89.42.237.71	Mozilla/5.0 (Windows NT 5.1; rv:34.0) Gecko/20100101 Firefox/34.0

Drill's Flexible UDF Interface Allows you to write your own functions.

A collection of GeoIP Functions is available on GitHub.

https://github.com/cgivre/drill-geoip-functions

Drill GeoIP Functions

- **`getCountryName( <ip> )`**: This function returns the country name of the IP address, "Unknown" if the IP is unknown or invalid.
- **`getCountryConfidence( <ip> )`**: This function returns the confidence score of the country ISO code of the IP address.
- **`getCountryISOCode( <ip> )`**: This function returns the country ISO code of the IP address, "Unknown" if the IP is unknown or invalid.
- **`getCityName( <ip> )`**: This function returns the city name of the IP address, "Unknown" if the IP is unknown or invalid.
- **`getCityConfidence( <ip> )`**: This function returns confidence score of the city name of the IP address.
- **`getLatitude( <ip> )`**: This function returns the latitude associated with the IP address.
- **`getLongitude( <ip> )`**: This function returns the longitude associated with the IP address.
- **`getTimezone( <ip> )`**: This function returns the timezone associated with the IP address.
- **`getAccuracyRadius( <ip> )`**: This function returns the accuracy radius associated with the IP address, 0 if unknown.
- **`getAverageIncome( <ip> )`**: This function returns the average income of the region associated with the IP address, 0 if unknown.
- **`getMetroCode( <ip> )`**: This function returns the metro code of the region associated with the IP address, 0 if unknown.
- **`getPopulationDensity( <ip> )`**: This function returns the population density associated with the IP address.
- **`getPostalCode( <ip> )`**: This function returns the postal code associated with the IP address.
- **`getCoordPoint( <ip> )`**: This function returns a point for use in GIS functions of the lat/long of associated with the IP address.
- **`getASN( <ip> )`**: This function returns the autonomous system of the IP address, "Unknown" if the IP is unknown or invalid.
- **`getASNOrganization( <ip> )`**: This function returns the autonomous system organization of the IP address, "Unknown" if the IP is unknown or invalid.
- **`isEU( <ip> ), isEuropeanUnion( <ip> )`**: This function returns `true` if the ip address is located in the European Union, `false` if not.
- **`isAnonymous( <ip> )`**: This function returns `true` if the ip address is anonymous, `false` if not.
- **`isAnonymousVPN( <ip> )`**: This function returns `true` if the ip address is an anonymous virtual private network (VPN), `false` if not.
- **`isHostingProvider( <ip> )`**: This function returns `true` if the ip address is a hosting provider, `false` if not.
- **`isPublciProxy( <ip> )`**: This function returns `true` if the ip address is a public proxy, `false` if not.
- **`isTORExitNode( <ip> )`**: This function returns `true` if the ip address is a known TOR exit node, `false` if not.

https://github.com/cgivre/drill-geoip-functions

Who is Trying to Hack This Site?

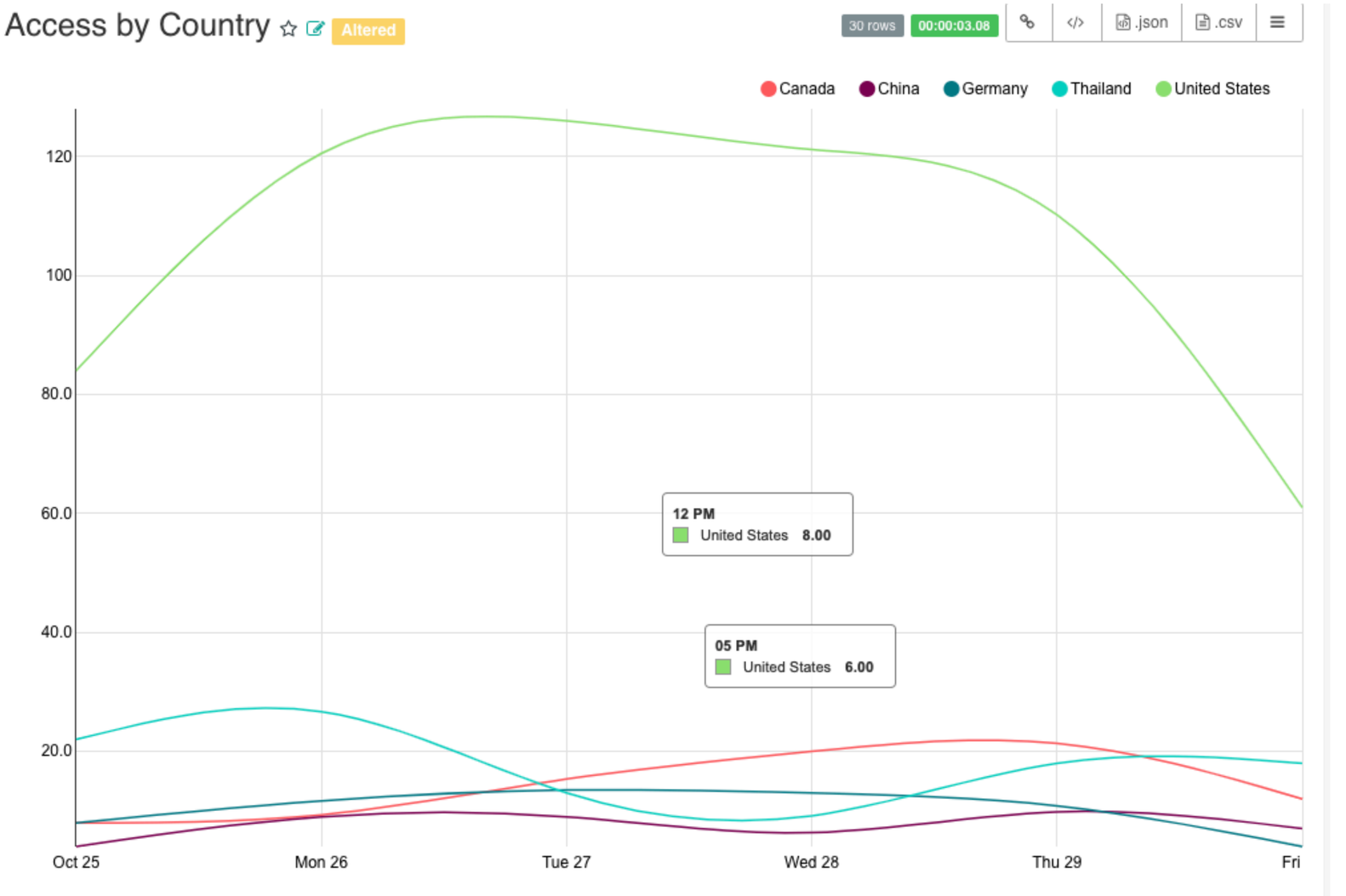
```
SELECT request_receive_time, connection_client_host,
request_useragent,
getCountryName(connection_client_host ) as countryName,
getCityName(connection_client_host ) as cityName
FROM dfs.test.`hackers-access.httpd`
WHERE request_firstline_uri = '/join_form'
```

Explore | .CSV | Clipboard

Search Results

connection_client_host	request_useragent	countryName	cityName
23.95.237.180	Mozilla/5.0 (Windows NT 5.1; rv:35.0) Gecko/20100101 Firefox/35.0	United States	Buffalo
23.95.237.180	Mozilla/5.0 (Windows NT 5.1; rv:35.0) Gecko/20100101 Firefox/35.0	United States	Buffalo
158.222.5.157	Mozilla/5.0 (Windows NT 6.3; WOW64; rv:34.0) Gecko/20100101 Firefox/34.0 AlexaToolbar/alxf-2.21	United States	Wilmington
158.222.5.157	Mozilla/5.0 (Windows NT 6.3; WOW64; rv:34.0) Gecko/20100101 Firefox/34.0 AlexaToolbar/alxf-2.21	United States	Wilmington
5.39.5.5	Mozilla/5.0 (Windows NT 5.1; rv:34.0) Gecko/20100101 Firefox/34.0	France	Unknown
180.180.64.16	Mozilla/5.0 (Windows NT 5.1; rv:35.0) Gecko/20100101 Firefox/35.0	Thailand	Pattaya
180.180.64.16	Mozilla/5.0 (Windows NT 5.1; rv:35.0) Gecko/20100101 Firefox/35.0	Thailand	Pattaya
89.42.237.71	Mozilla/5.0 (Windows NT 5.1; rv:34.0) Gecko/20100101 Firefox/34.0	Spain	Roldan
89.42.237.71	Mozilla/5.0 (Windows NT 5.1; rv:34.0) Gecko/20100101 Firefox/34.0	Spain	Roldan
216.158.199.158	Mozilla/5.0 (Windows NT 5.1; rv:34.0) Gecko/20100101 Firefox/34.0	United States	Unknown

Who is Trying to Hack This Site?

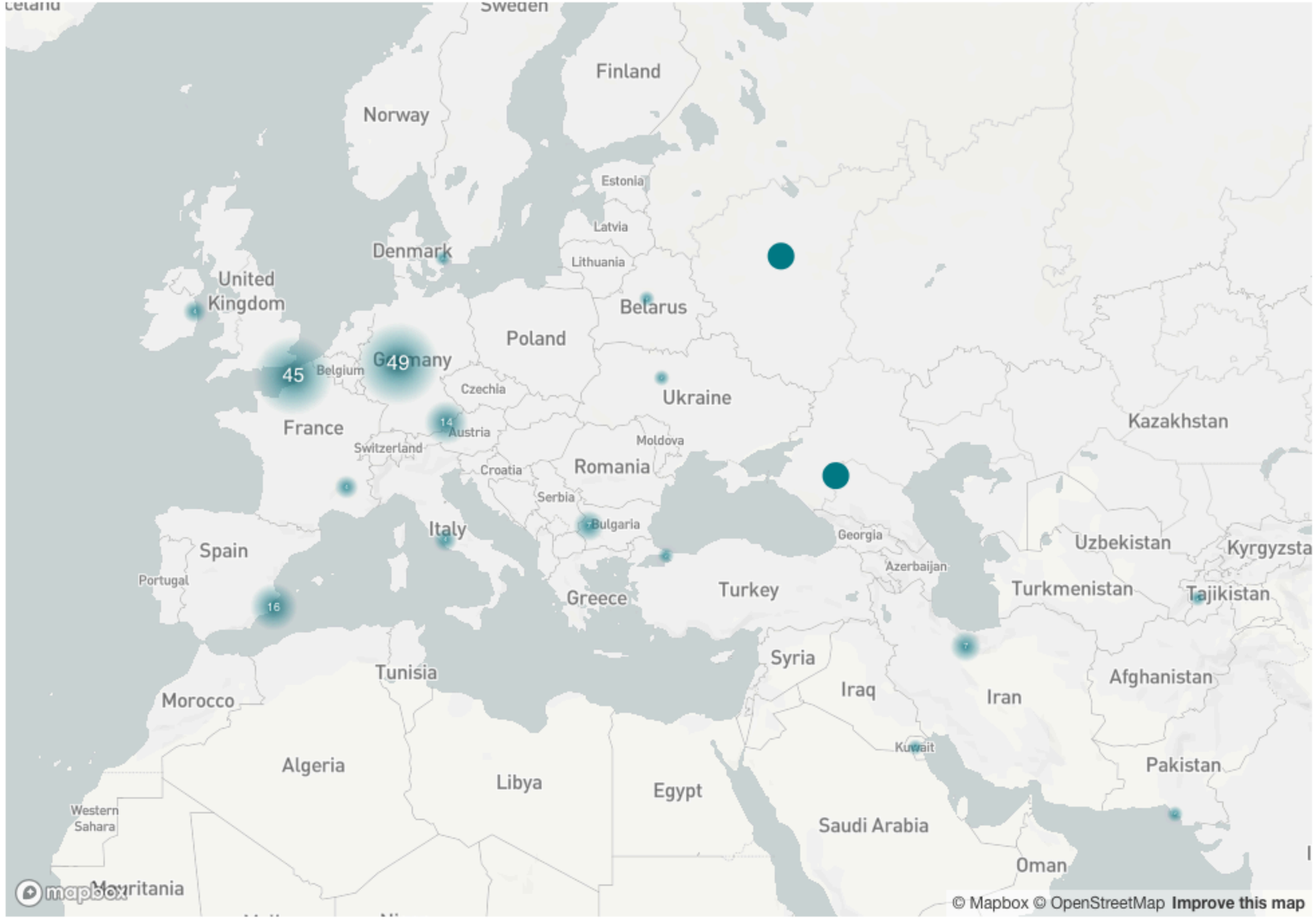


Who is Trying to Hack This Site?

```
SELECT request_receive_time, connection_client_host, request_useragent,
getCountryName(connection_client_host ) as countryName,
getCityName(connection_client_host ) as cityName,
getLatitude(connection_client_host ) as latitude,
getLongitude(connection_client_host ) as longitude
FROM dfs.test.`hackers-access.httpd`
WHERE request_firstline_uri = '/join_form'
```

uest_useragent	countryName	cityName	latitude	longitude
zilla/5.0 (Windows NT 5.1; rv:35.0) Gecko/20100101 Firefox/35.0	United States	Buffalo	42.8864	-78.8781
zilla/5.0 (Windows NT 5.1; rv:35.0) Gecko/20100101 Firefox/35.0	United States	Buffalo	42.8864	-78.8781
zilla/5.0 (Windows NT 6.3; WOW64; rv:34.0) Gecko/20100101 Firefox/34.0 AlexaToolbar/alxf-2.21	United States	Wilmington	39.8188	-75.5064
zilla/5.0 (Windows NT 6.3; WOW64; rv:34.0) Gecko/20100101 Firefox/34.0 AlexaToolbar/alxf-2.21	United States	Wilmington	39.8188	-75.5064
zilla/5.0 (Windows NT 5.1; rv:34.0) Gecko/20100101 Firefox/34.0	France	Unknown	48.8582	2.3387000000000002
zilla/5.0 (Windows NT 5.1; rv:35.0) Gecko/20100101 Firefox/35.0	Thailand	Pattaya	13.05	100.9333
zilla/5.0 (Windows NT 5.1; rv:35.0) Gecko/20100101 Firefox/35.0	Thailand	Pattaya	13.05	100.9333
zilla/5.0 (Windows NT 5.1; rv:34.0) Gecko/20100101 Firefox/34.0	Spain	Roldan	37.798	-1.0097
zilla/5.0 (Windows NT 5.1; rv:34.0) Gecko/20100101 Firefox/34.0	Spain	Roldan	37.798	-1.0097
zilla/5.0 (Windows NT 5.1; rv:34.0) Gecko/20100101 Firefox/34.0	United States	Unknown	37.751	-97.822

Who is Trying to Hack This Site?



What equipment are they using?

What equipment are they using?

- Drill has support for multidimensional data structures including KV Pairs and Lists.
- There is a pre-existing UDF (https://github.com/nielsbasjes/yauaa) for Apache Drill which can parse User Agent Strings and get you a lot of useful information from the UA string.

Who is Trying to Hack This Site?

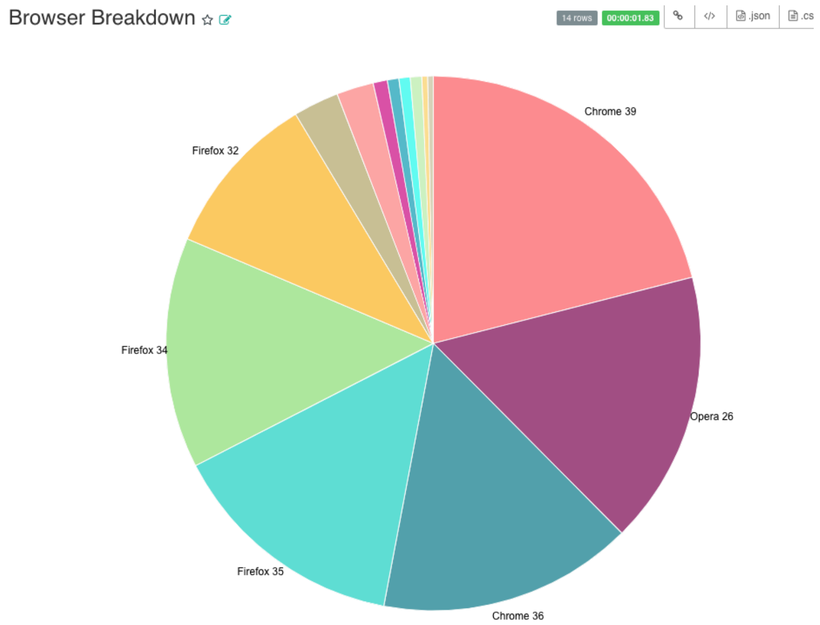
```
SELECT parse_user_agent(request_useragent) AS ua
FROM dfs.test.`hackers-access.httpd`
WHERE request_firstline_uri = '/join_form'
```

Explore | .CSV | Clipboard | Search Results

ua
{"DeviceClass":"Desktop","DeviceName":"Desktop","DeviceBrand":"Unknown","DeviceCpuBits":"32","OperatingSystemClass":"Desktop","OperatingSystemName":"Windows NT","OperatingSystemVersion":"XP","Operati
{"DeviceClass":"Desktop","DeviceName":"Desktop","DeviceBrand":"Unknown","DeviceCpuBits":"32","OperatingSystemClass":"Desktop","OperatingSystemName":"Windows NT","OperatingSystemVersion":"XP","Operati
{"DeviceClass":"Desktop","DeviceName":"Desktop","DeviceBrand":"Unknown","DeviceCpuBits":"64","OperatingSystemClass":"Desktop","OperatingSystemName":"Windows NT","OperatingSystemVersion":"8.1","Operati
{"DeviceClass":"Desktop","DeviceName":"Desktop","DeviceBrand":"Unknown","DeviceCpuBits":"64","OperatingSystemClass":"Desktop","OperatingSystemName":"Windows NT","OperatingSystemVersion":"8.1","Operati
{"DeviceClass":"Desktop","DeviceName":"Desktop","DeviceBrand":"Unknown","DeviceCpuBits":"32","OperatingSystemClass":"Desktop","OperatingSystemName":"Windows NT","OperatingSystemVersion":"XP","Operati
{"DeviceClass":"Desktop","DeviceName":"Desktop","DeviceBrand":"Unknown","DeviceCpuBits":"32","OperatingSystemClass":"Desktop","OperatingSystemName":"Windows NT","OperatingSystemVersion":"XP","Operati
{"DeviceClass":"Desktop","DeviceName":"Desktop","DeviceBrand":"Unknown","DeviceCpuBits":"32","OperatingSystemClass":"Desktop","OperatingSystemName":"Windows NT","OperatingSystemVersion":"XP","Operati
{"DeviceClass":"Desktop","DeviceName":"Desktop","DeviceBrand":"Unknown","DeviceCpuBits":"32","OperatingSystemClass":"Desktop","OperatingSystemName":"Windows NT","OperatingSystemVersion":"XP","Operati
{"DeviceClass":"Desktop","DeviceName":"Desktop","DeviceBrand":"Unknown","DeviceCpuBits":"32","OperatingSystemClass":"Desktop","OperatingSystemName":"Windows NT","OperatingSystemVersion":"XP","Operati
{"DeviceClass":"Desktop","DeviceName":"Desktop","DeviceBrand":"Unknown","DeviceCpuBits":"32","OperatingSystemClass":"Desktop","OperatingSystemName":"Windows NT","OperatingSystemVersion":"XP","Operati

Who is Trying to Hack This Site?

```
SELECT table1.ua.OperatingSystemNameVersion as os,
table1.ua.AgentNameVersionMajor as browser
FROM
(
SELECT
parse_user_agent(request_useragent) AS ua
FROM dfs.test.`hackers-access.httpd`
WHERE request_firstline_uri = '/join_form'
) AS table1
```



Explore .CSV Clipboard

Firefox 34

os	browser
Windows XP	Firefox 35
Windows XP	Firefox 35
Windows 8.1	AlexaToolbar alxf
Windows 8.1	AlexaToolbar alxf
Windows XP	Firefox 34
Windows XP	Firefox 35
Windows XP	Firefox 35
Windows XP	Firefox 34
Windows XP	Firefox 34
Windows XP	Firefox 34

Questions so far?

Let's have some fun with PCAP

Let's have some fun with PCAP

- PCAP is short for **P**acket **Cap**ture and represent raw network traffic.
- Files are encoded in binary format, but various tools exist to analyze PCAP files or convert them into more easily accessible formats, such as JSON.

PCAP Analysis

`arp-storm.pcap`	
type	VARCHAR
network	INTEGER
timestamp	TIMESTAMP
timestamp_micro	BIGINT
src_ip	VARCHAR
dst_ip	VARCHAR
src_port	INTEGER
dst_port	INTEGER
src_mac_address	VARCHAR
dst_mac_address	VARCHAR
tcp_session	BIGINT
tcp_ack	INTEGER
tcp_flags	INTEGER
tcp_flags_ns	INTEGER
tcp_flags_cwr	INTEGER
tcp_flags_ece	INTEGER
tcp_flags_ece_ecn_capable	INTEGER
tcp_flags_ece_congestion_experienced	INTEGER
tcp_flags_urg	INTEGER
tcp_flags_ack	INTEGER
tcp_flags_psh	INTEGER
tcp_flags_rst	INTEGER
tcp_flags_syn	INTEGER
tcp_flags_fin	INTEGER
tcp_parsed_flags	VARCHAR
packet_length	INTEGER
data	VARCHAR

PCAP Analysis

- Drill has a comprehensive collection of networking functions to facilitate network analysis
- is_private_ip(), is_valid_ip(), in_network() and many others.
- inet_aton() and inet_ntoa() exist to facilitate sorting IP ranges.

PCAP Analysis

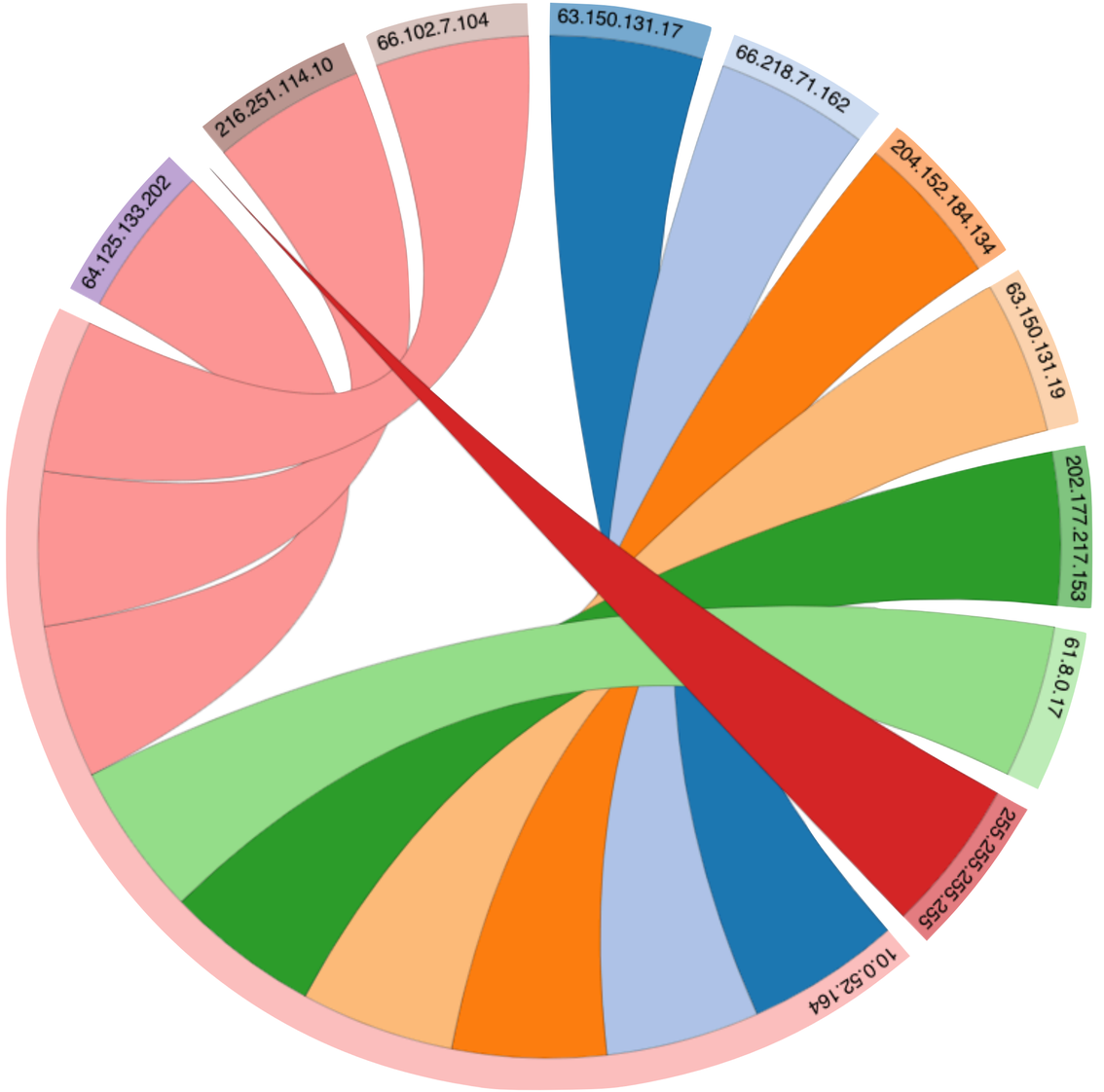
type	network	timestamp	timestamp_micro	src_ip	dst_ip	src_port	dst_port	src_mac_address	dst_mac_address
UDP	1	2005-03-30T08:47:46.496000	1112172466496576	192.168.170.20	192.168.170.8	53	32795	00:C0:9F:32:41:8C	00:E0:18:B1:0C:AD
UDP	1	2005-03-30T08:47:50.501000	1112172470501268	192.168.170.8	192.168.170.20	32795	53	00:E0:18:B1:0C:AD	00:C0:9F:32:41:8C
UDP	1	2005-03-30T08:47:51.333000	1112172471333401	192.168.170.20	192.168.170.8	53	32795	00:C0:9F:32:41:8C	00:E0:18:B1:0C:AD
UDP	1	2005-03-30T08:47:59.313000	1112172479313231	192.168.170.8	192.168.170.20	32795	53	00:E0:18:B1:0C:AD	00:C0:9F:32:41:8C
UDP	1	2005-03-30T08:47:59.452000	1112172479452255	192.168.170.20	192.168.170.8	53	32795	00:C0:9F:32:41:8C	00:E0:18:B1:0C:AD
UDP	1	2005-03-30T08:48:07.320000	1112172487320873	192.168.170.8	192.168.170.20	32795	53	00:E0:18:B1:0C:AD	00:C0:9F:32:41:8C
UDP	1	2005-03-30T08:48:07.321000	1112172487321379	192.168.170.20	192.168.170.8	53	32795	00:C0:9F:32:41:8C	00:E0:18:B1:0C:AD
UDP	1	2005-03-30T08:49:18.685000	1112172558685951	192.168.170.8	192.168.170.20	32795	53	00:E0:18:B1:0C:AD	00:C0:9F:32:41:8C
UDP	1	2005-03-30T08:49:18.734000	1112172558734862	192.168.170.20	192.168.170.8	53	32795	00:C0:9F:32:41:8C	00:E0:18:B1:0C:AD
UDP	1	2005-03-30T08:49:35.461000	1112172575461181	192.168.170.8	192.168.170.20	32795	53	00:E0:18:B1:0C:AD	00:C0:9F:32:41:8C

Let's look for who is talking to whom...

PCAP Analysis

```
SELECT DISTINCT src_ip, dst_ip
FROM dfs.test.`slowdownload.pcap`
WHERE src_ip is not null
```

src_ip	dst_ip
10.100.252.1	255.255.255.255
10.0.52.164	216.251.114.10
10.0.52.164 4.10	10.0.52.164
10.0.52.164	66.218.71.162
66.218.71.162	10.0.52.164
10.0.52.164	66.102.7.104
66.102.7.104	10.0.52.164
10.0.52.164	202.177.217.153
202.177.217.153	10.0.52.164
10.0.52.164	63.150.131.19
63.150.131.19	10.0.52.164
10.0.52.164	63.150.131.17



Network Forensics

These examples are from a forensic challenge from the Honeynet Project.

1. Which systems (i.e. IP addresses) are involved?
2. What can you find out about the attacking host (e.g., where is it located)?
3. How many TCP sessions are contained in the dump file?
4. How long did it take to perform the attack?
5. Which operating system was targeted by the attack? And which service? Which vulnerability?

https://www.honeynet.org/node/504

Where is the attacker?

```
1  SELECT DISTINCT src_ip, getCityName(src_ip) AS city_name,
2  getCountryName(src_ip) as country_name,
3  getLatitude(src_ip) as latitude,
4  getLongitude(src_ip) as longitude,
5  getASN(src_ip) AS ASN,
6  getASNOrganization(src_ip) AS ASNO,
7  getTimezone(src_ip) AS tz,
8  isAnonymous(src_ip) AS is_anon,
9  isPublicProxy(src_ip) AS public_proxy
10 FROM dfs.test.`attack-trace.pcap`
```

Run Query | Save Query | Share Query | LIMIT 1000 | parameters | 00:00:02.38

Results | Query History | Preview: `attack-trace.pcap`

Explore | .CSV | Clipboard | Filter Results

src_ip	city_name	country_name	latitude	longitude	ASN	ASNO	tz	is_anon	public_proxy
98.114.205.102	Philadelphia	United States	40.0634	-74.999	701	MCI Communications Services, Inc. d/b/a Verizon Busi…	America/New_York	false	false

gtkcyber.com

http://thedataist.com/using-drill-for-network-forensics-part-1/

How many sessions are there?

```
1  SELECT tcp_session,
2  COUNT( *) AS packet_count,
3  MIN(`timestamp`) AS start_time,
4  MAX(`timestamp`) AS end_time,
5  (MAX(`timestamp`) - MIN(`timestamp`)) as session_duration,
6  sum(packet_length) AS total_data_exchanged
7  FROM dfs.test.`attack-trace.pcap`
8  GROUP BY tcp_session
9
```

Run Query | Save Query | Share Query | LIMIT 1000

Results | Query History

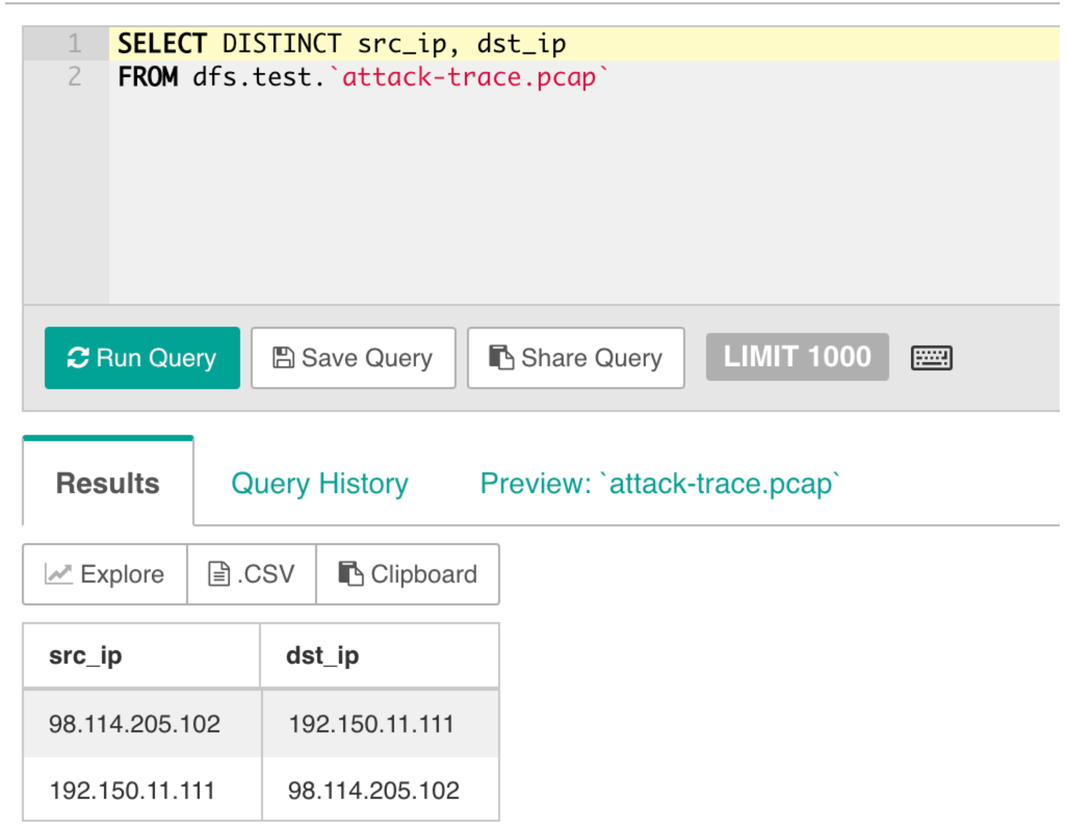
Explore | .CSV | Clipboard

tcp_session	packet_count	start_time	end_time	session_durat...	total_data_exchan...
-8791568836279708938	7	2009-04-20T03:28:28.374000	2009-04-20T03:28:28.728000	PT0.354S	412
-1594450961165523765	31	2009-04-20T03:28:28.509000	2009-04-20T03:28:33.447000	PT4.938S	6825
-38305772475396018	12	2009-04-20T03:28:30.466000	2009-04-20T03:28:33.566000	PT3.100S	817
-287133093281818961	27	2009-04-20T03:28:33.457000	2009-04-20T03:28:44.593000	PT11.136S	2069
-7863404950915247766	271	2009-04-20T03:28:34.516000	2009-04-20T03:28:44.588000	PT10.072S	173388

http://thedataist.com/using-drill-for-network-forensics-part-1/

Which Systems are Involved?

```
SELECT DISTINCT src_ip, dst_ip FROM
dfs.test.attack-trace.pcap
```



http://thedataist.com/using-drill-for-network-forensics-part-1/

How long did the attack take?

```
1  SELECT
2  (MAX(`timestamp`) - MIN(`timestamp`)) as session_duration
3  FROM dfs.test.`attack-trace.pcap`
4
```

Run Query | Save Query | Share Query | LIMIT 1000

Results | Query History

Explore | .CSV | Clipboard

session_durat...
PT16.219S

http://thedataist.com/using-drill-for-network-forensics-part-1/

What OS/Services were used?

```
SELECT `timestamp`, src_ip, src_port, data
FROM dfs.test.`attack-trace.pcap`
ORDER BY `timestamp` ASC
```

Run Query | Save Query | Share Query | LIMIT 1000 | parameters | 00:00:00.39

Results | Query History | Preview: `attack-trace.pcap`

Explore | .CSV | Clipboard | Filter Results

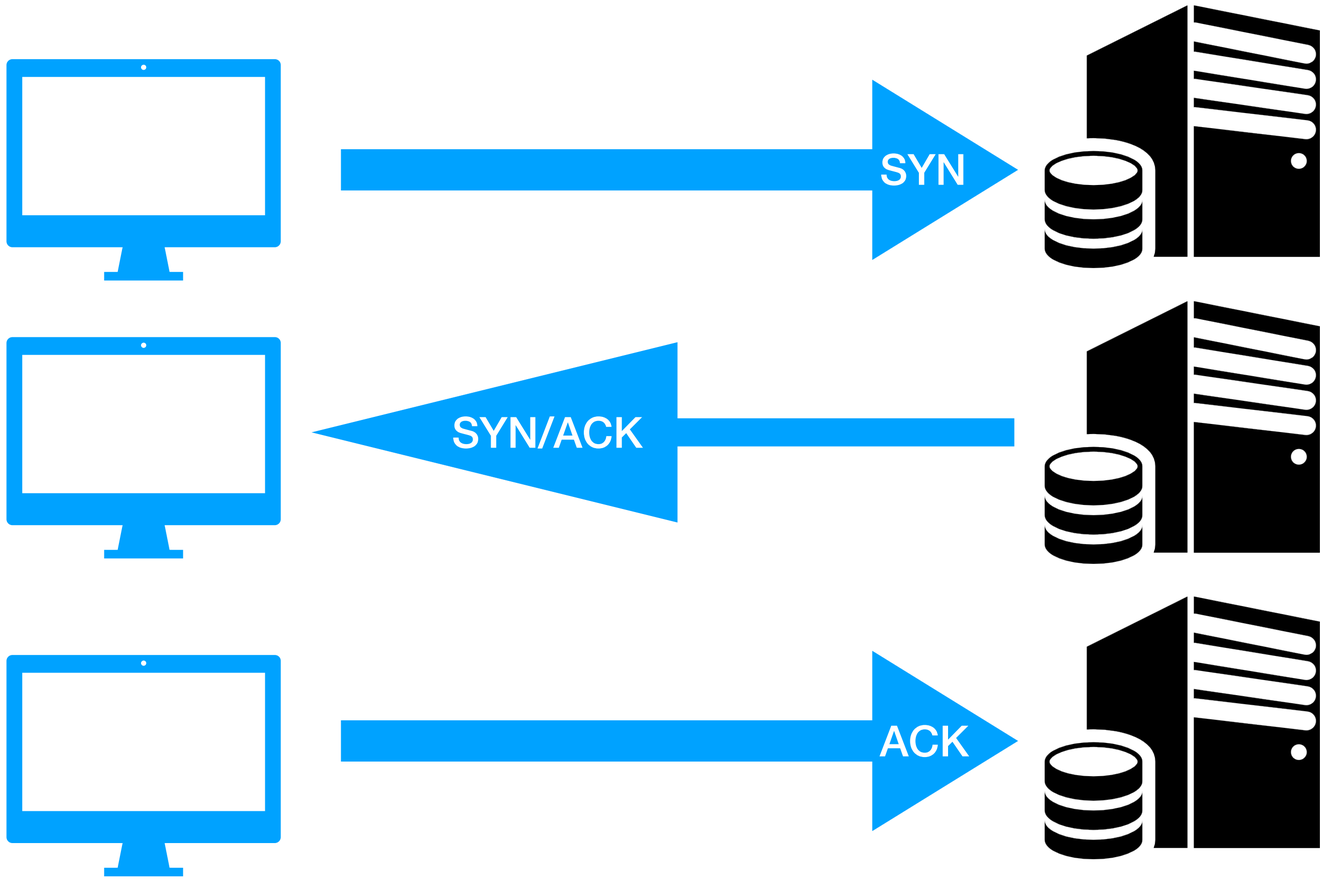
Looks like Windows 2000 or XP

timestamp	src_ip	src_port	data
2009-04-20T03:28:28.976000	192.150.11.111	445	[]
2009-04-20T03:28:28.976000	98.114.205.102	1828	..SMBs..... .....i.NTLMSSP.....W.i.n.d.o.w.s. .2.0.0.0. .2.1.9.5...W.i.n.d.o.w.s. .2.0...5...0.....I....6...6
2009-04-20T03:28:29.097000	192.150.11.111	445	TLMSSP.....0.....`b.m.....L.L.<...V.I.D.C.A.M.....V.I.D.C.A.M.....V.I.D.C.A.M.....V.I.D.C.A.M.....V.I.D.C.A.M.....W.i.n.d.o.w.s. .5...1...W.i.n.d.o.w.s
2009-04-20T03:28:29.215000	98.114.205.102	1828	..SMBs..... .....W.....NTLMSSP.....F.....G.....@.....@.....@.....G.....H.O.D....jz..I.(.0%t.gSW.i.n.d.o.w.s. .2.0.0.0. .2.1.9.5...W
2009-04-20T03:28:29.215000	192.150.11.111	445	[]
2009-04-20T03:28:29.332000	192.150.11.111	445	W.i.n.d.o.w.s. .5...1...W.i.n.d.o.w.s. .2.0.0.0. .L.A.N. .M.a.n.a.g.e.r.....I.....

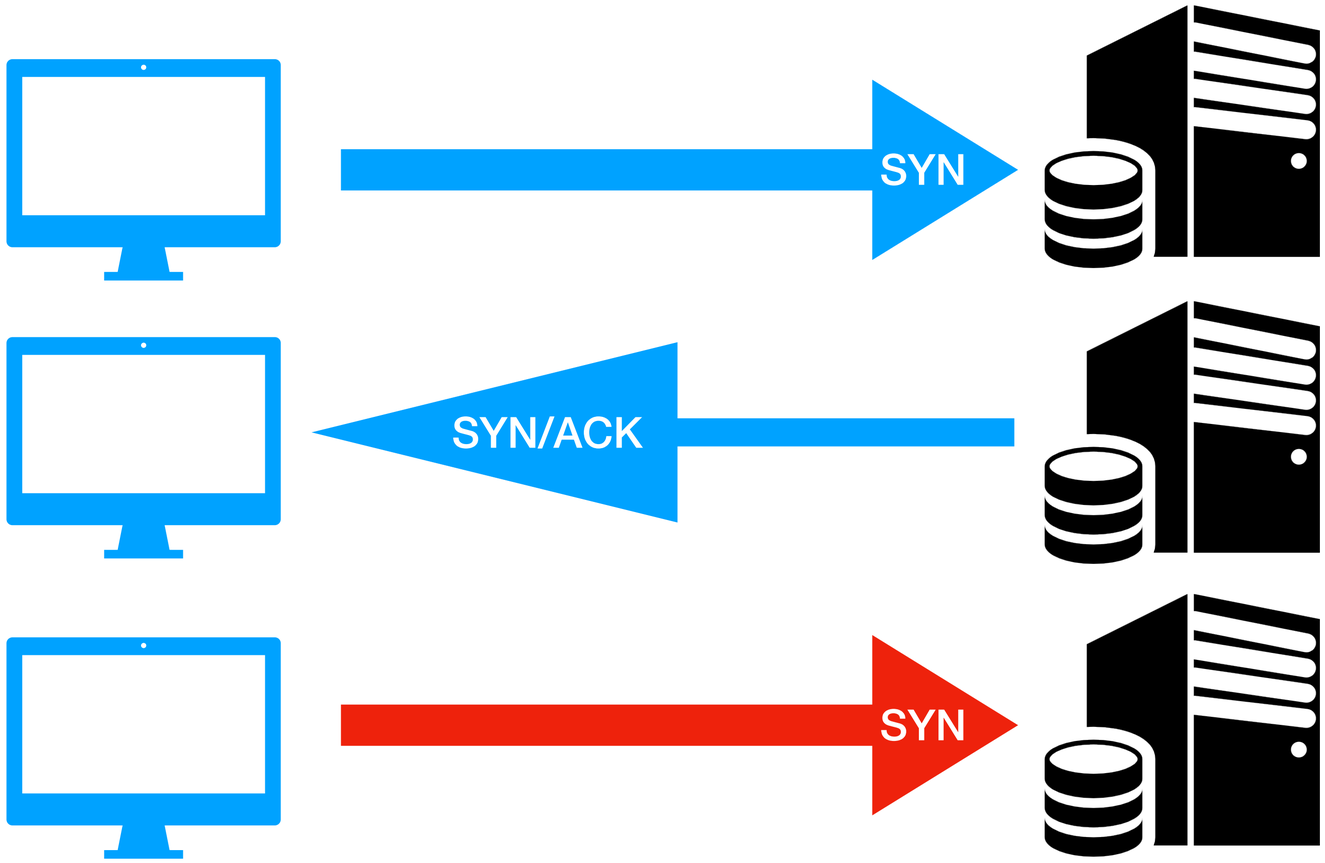
gtkcyber.com

http://thedataist.com/using-drill-for-network-forensics-part-1/

SynFlood Detection



SynFlood Detection



Drill can automate SYN Flood detection with a custom UDF.

https://github.com/cgivre/drill-synflood-udf

SYN Flood Detection

```
SELECT tcp_session
FROM dfs.test.`synscan.pcap`
GROUP BY tcp_session
HAVING is_syn_flood(tcp_session, tcp_flags_syn, tcp_flags_ack)
```

tcp_session
63466047320284693 74
-9031405983396365775
7738739733723725373

SYN Flood Detection

```
SELECT *
FROM dfs.test.`synscan.pcap`
WHERE tcp_session IN
(
    SELECT tcp_session
    FROM dfs.test.`synscan.pcap`
    GROUP BY tcp_session
    HAVING is_syn_flood(tcp_session, tcp_flags_syn, tcp_flags_ack)
)
```

type	network	timestamp	timestamp_micro	src_ip	dst_ip	src_port	dst_port	src_mac_address	dst_mac_address	tcp_session
TCP	1	2010-07-04T20:24:16.276000	1278275056276783	172.16.0.8	64.13.134.52	36050	53	00:25:B3:BF:91:EE	00:26:0B:31:07:33	6346604732028469374
TCP	1	2010-07-04T20:24:16.338000	1278275056338667	64.13.134.52	172.16.0.8	53	36050	00:26:0B:31:07:33	00:25:B3:BF:91:EE	6346604732028469374
TCP	1	2010-07-04T20:24:16.403000	1278275056403870	172.16.0.8	64.13.134.52	36050	80	00:25:B3:BF:91:EE	00:26:0B:31:07:33	-9031405983396365775
TCP	1	2010-07-04T20:24:16.464000	1278275056464845	64.13.134.52	172.16.0.8	80	36050	00:26:0B:31:07:33	00:25:B3:BF:91:EE	-9031405983396365775
TCP	1	2010-07-04T20:24:17.678000	1278275057678354	172.16.0.8	64.13.134.52	36050	22	00:25:B3:BF:91:EE	00:26:0B:31:07:33	7738739733723725373
TCP	1	2010-07-04T20:24:17.740000	1278275057740531	64.13.134.52	172.16.0.8	22	36050	00:26:0B:31:07:33	00:25:B3:BF:91:EE	7738739733723725373
TCP	1	2010-07-04T20:24:19.338000	1278275059338245	64.13.134.52	172.16.0.8	53	36050	00:26:0B:31:07:33	00:25:B3:BF:91:EE	6346604732028469374
TCP	1	2010-07-04T20:24:19.462000	1278275059462133	64.13.134.52	172.16.0.8	80	36050	00:26:0B:31:07:33	00:25:B3:BF:91:EE	-9031405983396365775
TCP	1	2010-07-04T20:24:21.338000	1278275061338288	64.13.134.52	172.16.0.8	22	36050	00:26:0B:31:07:33	00:25:B3:BF:91:EE	7738739733723725373

Questions so far?

How about log files?

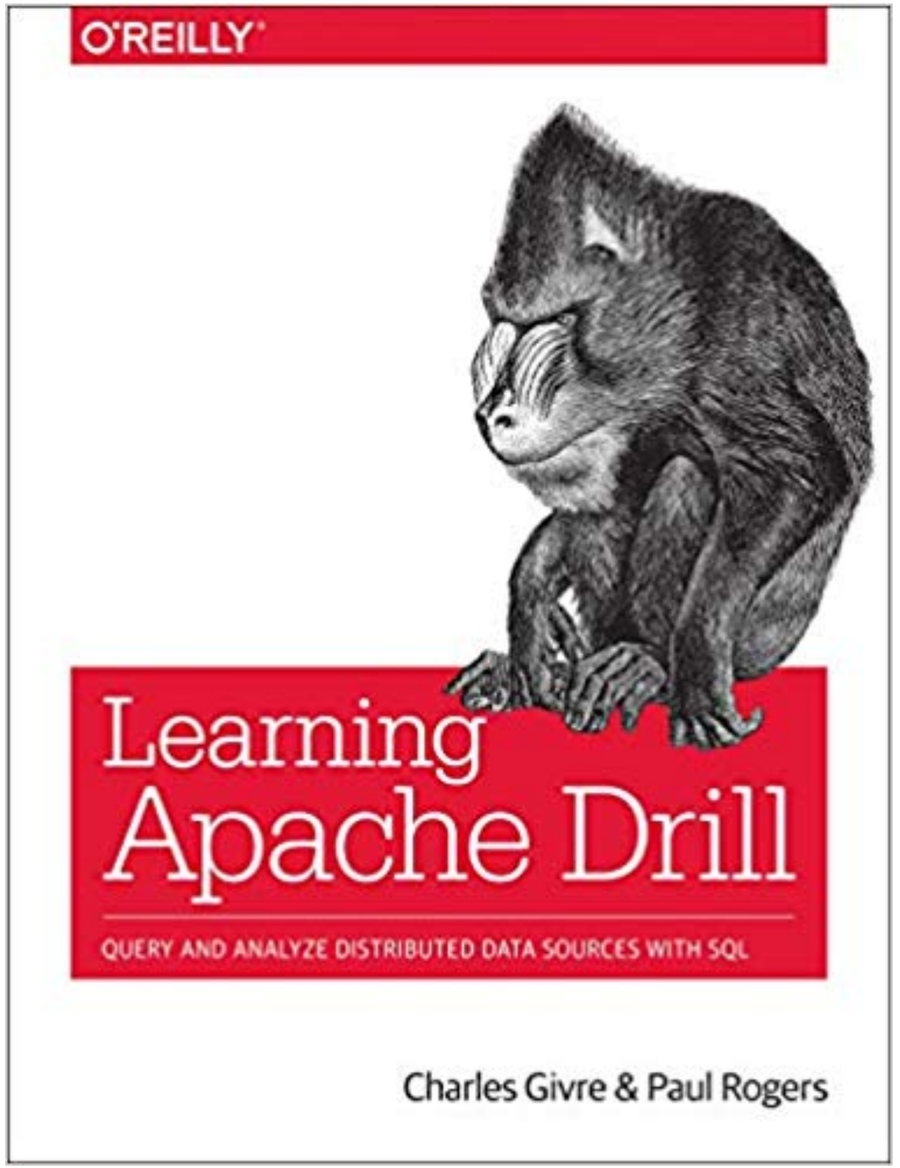

```
Dec 12 2018 06:50:25    sshd[36669]: Failed password for root from 61.160.251.136 port 1313 ssh2
Dec 12 2018 06:50:25    sshd[36669]: Failed password for root from 61.160.251.136 port 1313 ssh2
Dec 12 2018 06:50:24    sshd[36669]: Failed password for root from 61.160.251.136 port 1313 ssh2
Dec 12 2018 03:36:23    sshd[41875]: Failed password for root from 222.189.239.10 port 1350 ssh2
Dec 12 2018 03:36:22    sshd[41875]: Failed password for root from 222.189.239.10 port 1350 ssh2
Dec 12 2018 03:36:22    sshlockout[15383]: Locking out 222.189.239.10 after 15 invalid attempts
Dec 12 2018 03:36:22    sshd[41875]: Failed password for root from 222.189.239.10 port 1350 ssh2
Dec 12 2018 03:36:22    sshlockout[15383]: Locking out 222.189.239.10 after 15 invalid attempts
Dec 12 2018 03:36:22    sshd[42419]: Failed password for root from 222.189.239.10 port 2646 ssh2
```

eventDate	process_name	pid	message	src_ip
2018-12-12T11:50:25				
2018-12-12T11:50:25	sshd	36669	Failed password for root from 61.160.251.136 port 1313 ssh2	61.160.251.136
2018-12-12T11:50:25	sshd	36669	Failed password for root from 61.160.251.136 port 1313 ssh2	61.160.251.136
2018-12-12T11:50:24	sshd	36669	Failed password for root from 61.160.251.136 port 1313 ssh2	61.160.251.136
2018-12-12T08:36:23	sshd	41875	Failed password for root from 222.189.239.10 port 1350 ssh2	222.189.239.10
2018-12-12T08:36:22	sshd	41875	Failed password for root from 222.189.239.10 port 1350 ssh2	222.189.239.10
2018-12-12T08:36:22	sshlockout	15383	Locking out 222.189.239.10 after 15 invalid attempts	222.189.239.10
2018-12-12T08:36:22	sshd	41875	Failed password for root from 222.189.239.10 port 1350 ssh2	222.189.239.10
2018-12-12T08:36:22	sshlockout	15383	Locking out 222.189.239.10 after 15 invalid attempts	222.189.239.10
2018-12-12T08:36:22	sshd	42419	Failed password for root from 222.189.239.10 port 2646 ssh2	222.189.239.10
2018-12-12T08:36:22	sshlockout	15383	Locking out 222.189.239.10 after 15 invalid attempts	222.189.239.10
2018-12-12T08:36:22	sshd	42419	Failed password for root from 222.189.239.10 port 2646 ssh2	222.189.239.10

Drill can natively query Syslog formatted data*

*** This feature is available as of Drill version 1.16**

Questions?



https://amzn.to/2HDwE92

"Any sufficiently advanced technology is indistinguishable from magic"

—Arthur C. Clarke

Drill is **easy to use**

Drill uses **standard ANSI SQL**

Drill is **FAST!!**

CASE-1: Aggregation Query: Total number of records that have a rating of "3" or above	
Spark	spark-submit --class AggQuery --conf "spark.driver.memory=3g" sql1/target/scala-2.11/sparksqldemo1-assembly-1.0.jar
	real 0m52.631s user 1m31.097s sys 0m1.002s
Drill	drill-embedded -f sql1.sql (ALTER SYSTEM SET `store.json.read_numbers_as_double` = true; SELECT COUNT(MOVIE) FROM dfs.`ratings.json` WHERE RATE > 3.0;)
	real 0m23.917s user 0m26.819s sys 0m0.703s

https://www.mapr.com/blog/comparing-sql-functions-and-performance-apache-spark-and-apache-drill

CASE-2: Join Query: The user and movie names were joined, and the top 10 female users who most rated movies were extracted	
Spark	spark-submit --class topTenWomen --conf "spark.driver.memory=3g" sql2-1/target/scala-2.11/sparksqldemo2-1-assembly-1.0.jar
	real 0m57.671s user 1m40.031s sys 0m1.152s
Drill	drill-embedded -f sql2-1.sql (ALTER SYSTEM SET `store.json.read_numbers_as_double` = true; SELECT RATtbl.UID, COUNT(RATtbl.UID) as NUMEVALS FROM dfs.`ratings.json` as RATtbl JOIN dfs.`users.json` as USRtbl ON RATtbl.UID = USRtbl.UID WHERE USRtbl.GENDER = 'F' GROUP BY RATtbl.UID ORDER BY COUNT(RATtbl.UID) DESC LIMIT 10;)
	real 0m27.576s user 0m28.370s sys 0m0.803s

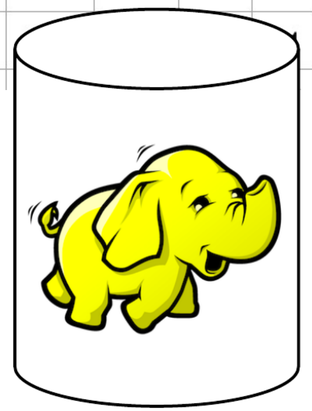
https://www.mapr.com/blog/comparing-sql-functions-and-performance-apache-spark-and-apache-drill

CASE-3: Join Query: The user and movie names were joined, and the top 10 names of highly rated movies were extracted	
Spark	spark-submit --class topTenMoviename --conf "spark.driver.memory=3g" sql2-2/target/scala-2.11/sparksqldemo2-2-assembly-1.0.jar
	real 0m57.982s user 1m41.480s sys 0m1.246s
Drill	drill-embedded -f sql2-2.sql (ALTER SYSTEM SET `store.json.read_numbers_as_double` = true; SELECT TMP2.MOVIE, TMP2.NUMRAT, TITLE FROM (SELECT MOVIE, COUNT(MOVIE) as NUMRAT FROM dfs.`ratings.json` GROUP BY MOVIE) TMP2 JOIN dfs.`movies.json` AS MOVtbl ON TMP2.MOVIE = MOVtbl.MOVIE ORDER BY TMP2.NUMRAT DESC LIMIT 10;)
	real 0m28.217s user 0m29.263s sys 0m0.658s

https://www.mapr.com/blog/comparing-sql-functions-and-performance-apache-spark-and-apache-drill

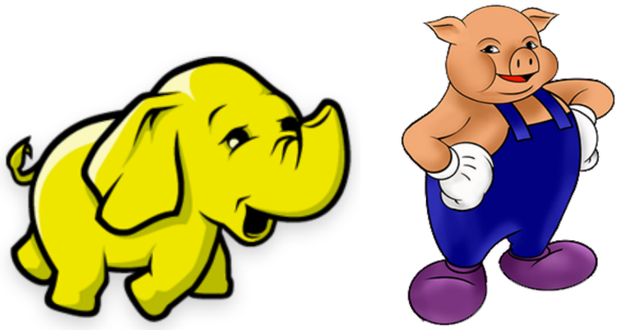
yearID	lgID	teamID	franchID	divID	Rank	G	Ghome	W	L	DivWin	WCWin	LgWin	WSWin	R	AB	H	2B	3B	HR	B
1871	NA	BS1	BNA		3	31		20	10			N		401	1372	426	70	37	3	
1871	NA	CH1	CNA		2	28		19	9			N		302	1196	323	52	21	10	
1871	NA	CL1	CFC		8	29		10	19			N		249	1186	328	35	40	7	
1871	NA	FW1	KEK		7	19		7	12			N		137	746	178	19	8	2	
1871	NA	NY2	NNA		5	33		16	17			N		302					1	

anlahman.com/baseball-archive/statistics



Quick Demo

```
data = load '/user/cloudera/data/baseball_csv/Teams.csv' using PigStorage(',');
filtered = filter data by ($0 == '1988');
tm_hr = foreach filtered generate (chararray) $40 as team, (int) $19 as hrs;
ordered = order tm_hr by hrs desc;
dump ordered;
```



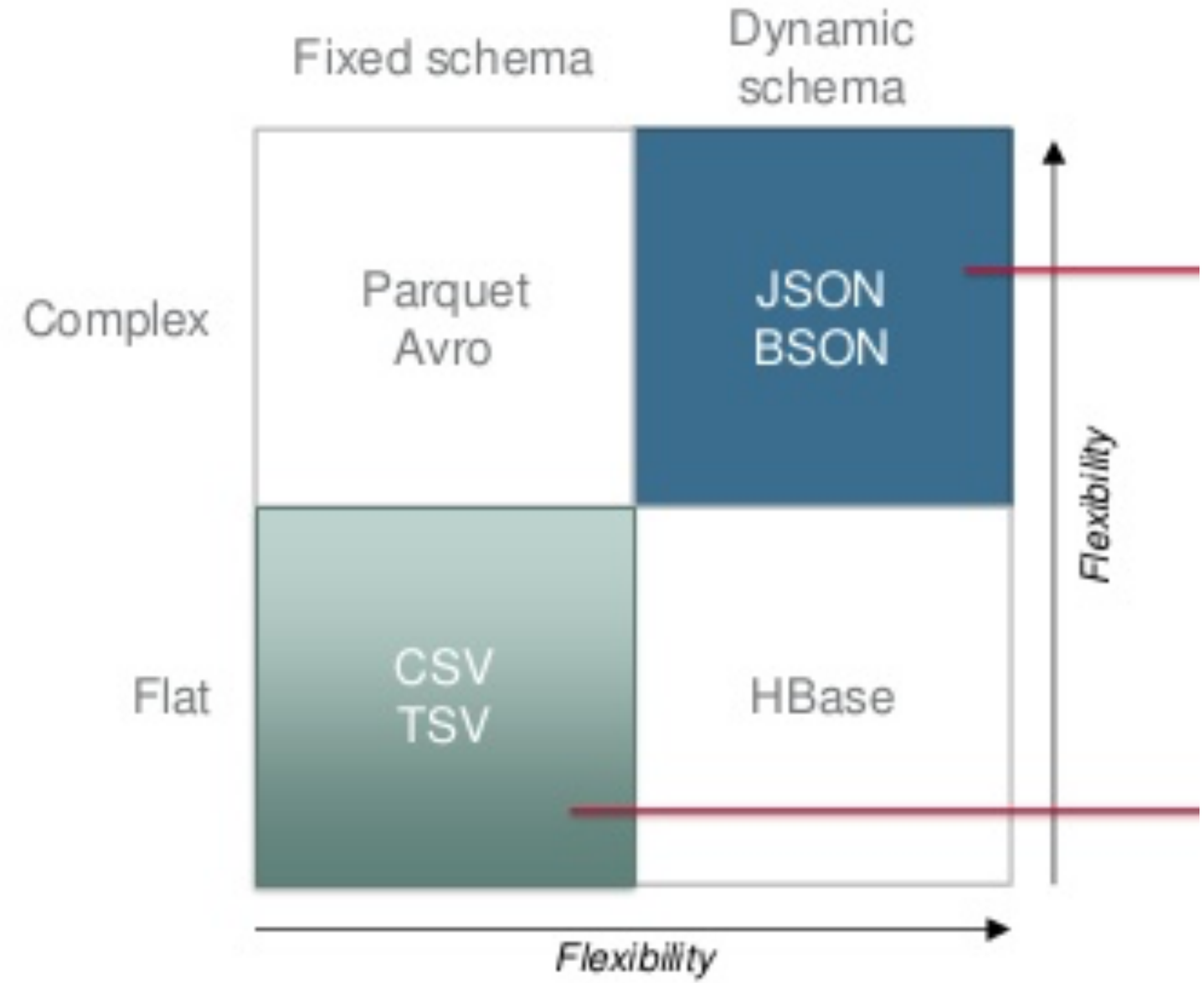
Execution Time:
1 minute, 38 seconds

Quick Demo

```
SELECT columns[40], cast(columns[19] as int) AS HR
FROM `baseball_csv/Teams.csv`
WHERE columns[0] = '1988'
ORDER BY HR desc;
```

Execution Time:
0.232 seconds!!

Drill is Versatile


Fixed schema
Dynamic schema
Complex
Flat
Parquet
Avro
JSON
BSON
CSV
TSV
HBase
Flexibility
Flexibility

NoSQL, No Problem

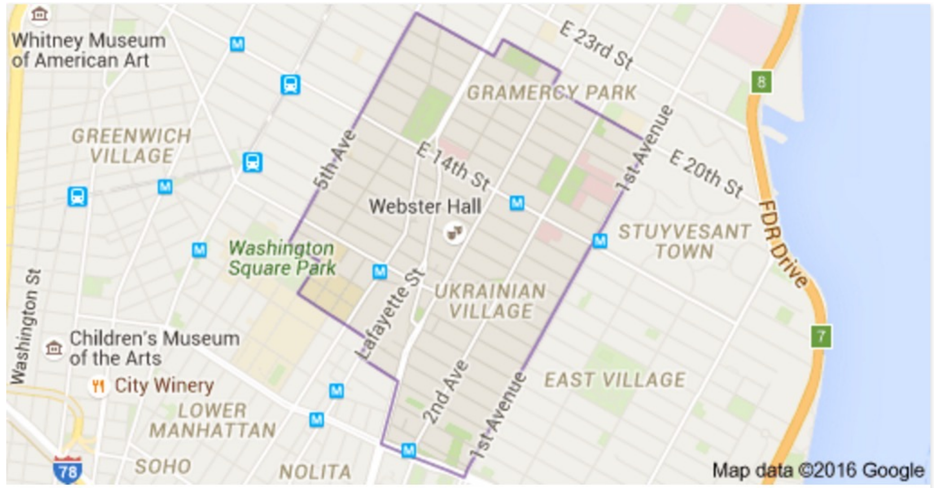
NoSQL, No Problem


```
{
  "address": {
     "building": "1007",
     "coord": [ -73.856077, 40.848447 ],
     "street": "Morris Park Ave",
     "zipcode": "10462"
  },
  "borough": "Bronx",
  "cuisine": "Bakery",
  "grades": [
     { "date": { "$date": 1393804800000 }, "grade": "A", "score": 2 },
     { "date": { "$date": 1378857600000 }, "grade": "A", "score": 6 },
     { "date": { "$date": 1358985600000 }, "grade": "A", "score": 10 },
     { "date": { "$date": 1322006400000 }, "grade": "A", "score": 9 },
     { "date": { "$date": 1299715200000 }, "grade": "B", "score": 14 }
  ],
  "name": "Morris Park Bake Shop",
  "restaurant_id": "30075445"
}
```


https://raw.githubusercontent.com/mongodb/docs-assets/primer-dataset/primer-dataset.json

NoSQL, No Problem

```
SELECT t.address.zipcode AS zip, count(name) AS rests
FROM `restaurants` t
GROUP BY t.address.zipcode
ORDER BY rests DESC
LIMIT 10;
```



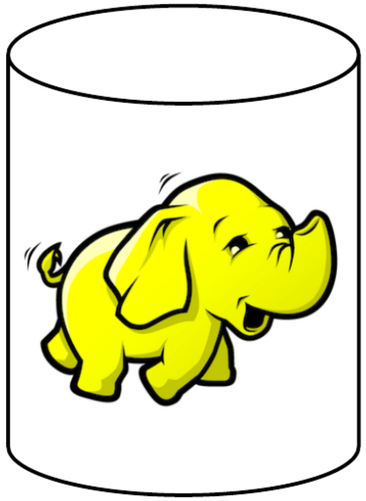
New York, NY 10003

zip	rests
10003	686
10019	675
10036	611
10001	520
10022	485
10013	480
10002	471
10011	467
10016	433
10014	428

https://raw.githubusercontent.com/mongodb/docs-assets/primer-dataset/primer-dataset.json

Querying Across Silos

Querying Across Silos



Farmers Market Data

Restaurant Data

Querying Across Silos

```
SELECT t1.Borough, t1.markets, t2.rests, cast(t1.markets AS
FLOAT)/ cast(t2.rests AS FLOAT) AS ratio
FROM (
 SELECT Borough, count(`Farmers Markets Name`) AS markets
 FROM `farmers_markets.csv`
 GROUP BY Borough ) t1
 JOIN (
   SELECT borough, count(name) AS rests
   FROM mongo.test.`restaurants`
   GROUP BY borough
   ) t2
ON t1.Borough=t2.borough
ORDER BY ratio DESC;
```

Borough	markets	rests	ratio
Bronx	18	2338	0.007698888
Brooklyn	34	6086	0.005586592
Manhattan	36	10259	0.003509114
Queens	12	5656	0.0021216408
Staten Island	1	969	0.0010319918

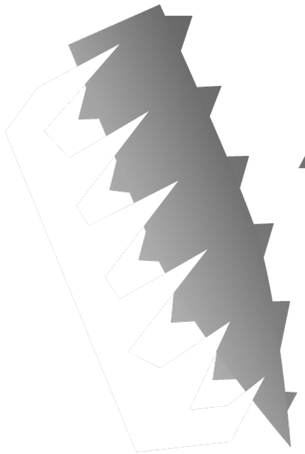
Execution Time: 0.502 Seconds

Querying Across Silos

data.world

Querying Across Silos

+

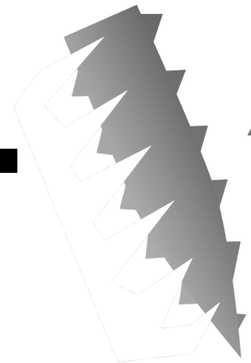


APACHE DRILL

data.world

Querying Across Silos

+



APACHE DRILL

=

Querying Across Silos

bin — java -Xms4G -Xmx4G -XX:MaxDirectMemorySize=8G -XX:ReservedCodeCacheSize=1G -Ddrill.exec.ena...

...rive/github/drill-xml-plugin — -bash | java -Xms4G -Xmx...jdbc:drill:zk=local | ~/OneDrive/metabase — -bash

```
0: jdbc:drill:zk=local> SELECT companyName FROM dw.cgivre.`mac-address-manufacturers`.`20170426mac_address.csv/20170426mac_address` WHERE country='CN' LIMIT  20;
+-------------------------------------------------------------+
|                         companyName                         |
+-------------------------------------------------------------+
| Shenzhen ViewAt Technology Co.,Ltd.                         |
| ShenZhen ANYK Technology Co.,LTD                            |
| SOYEA Technology Co.,Ltd.                                   |
| Nanjing Shining Electric Automation Co., Ltd                |
| Anhui comhigher tech co.,ltd                                |
| Wei Fang Goertek Electronics Co.,Ltd                        |
| zte corporation                                             |
| SHENZHEN DAJIAHAO TECHNOLOGY CO.,LTD                        |
| SHENZHEN CLOU ELECTRONICS CO. LTD.                          |
| HONG KONG TECON TECHNOLOGY                                  |
| Zhehua technology limited                                   |
| Chengdu Fuhuaxin Technology co.,Ltd                         |
| Shanghai LISTEN TECH.LTD                                    |
| Shenzhen ChuangDao & Perpetual Eternal Technology Co.,Ltd   |
| Beijing Novel Super Digital TV Technology Co., Ltd          |
| URadio Systems Co., Ltd                                     |
| Jiangsu Qinheng Co., Ltd.                                   |
| TP-LINK TECHNOLOGIES CO.,LTD.                               |
| BEIJING GEHUA CATV NETWORK CO.,LTD                          |
| TP-LINK TECHNOLOGIES CO.,LTD.                               |
+-------------------------------------------------------------+
20 rows selected (1.087 seconds)
0: jdbc:drill:zk=local>
```

Querying Across Silos

```
SELECT SUBSTRING( REGEXP_REPLACE( MACAddressSource, ':', '' ),1,6 ) AS MacAddress,
MACAddressSource, dw.companyName, dw.country
FROM dfs.test.`pcapview` AS p
JOIN dw.cgivre.`mac-address-manufacturers`.`20170426mac_address.csv/20170426mac_address`
AS dw ON dw.prefix = SUBSTRING( REGEXP_REPLACE( MACAddressSource, ':', '' ),1,6 )
```

MacAddress	MACAddressSource	companyName	country
080027	08:00:27:38:DB:ED	PCS Systemtechnik GmbH	US
080027	08:00:27:97:3F:45	PCS Systemtechnik GmbH	US

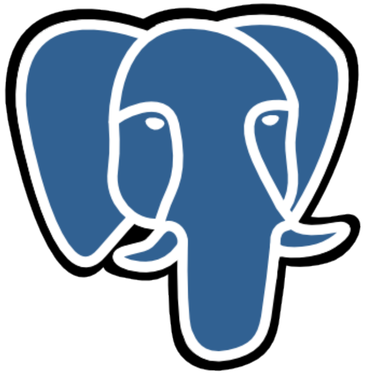
2 rows selected (3.775 seconds)

Why **aren’t** you using Drill?


WARNING
DANGER
DO NOT PULL
HANDLE
TOMCAT
RANGER
CV
VCENTURION

Installing & Configuring Drill

Embedded

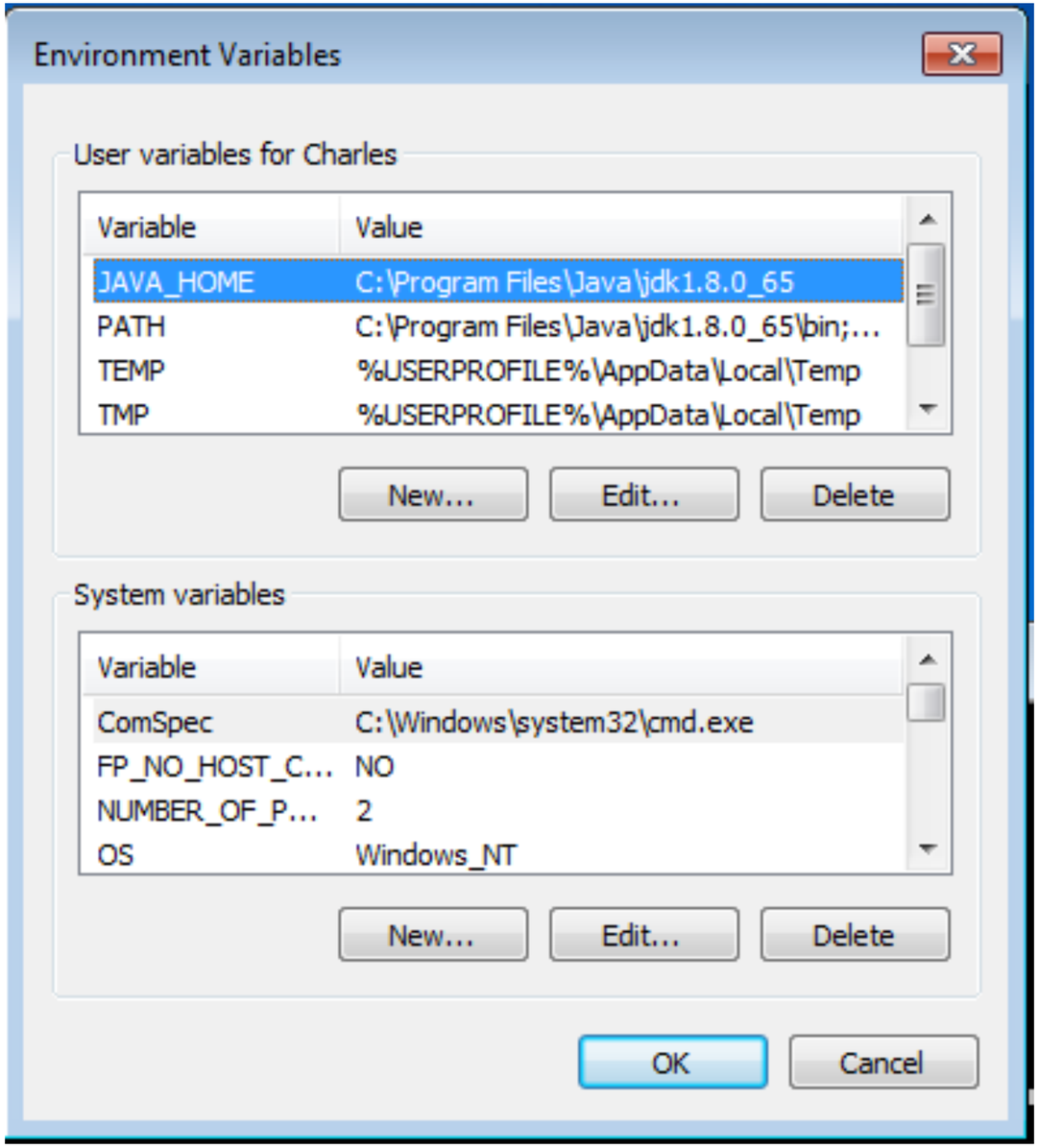


Distributed

Step 1: Download Drill: drill.apache.org/download/

Drill Requirements

- Oracle Java SE Development Kit (JDK 8) or higher. (Verify this by opening a command prompt and typing `java -version`)
- On Windows machines, you will need to set the JAVA_HOME and PATH variables.


Environment Variables
User variables for Charles
Variable
Value
JAVA_HOME
C:\Program Files\Java\jdk1.8.0_65
PATH
C:\Program Files\Java\jdk1.8.0_65\bin;...
TEMP
%USERPROFILE%\AppData\Local\Temp
TMP
%USERPROFILE%\AppData\Local\Temp
New...
Edit...
Delete
System variables
Variable
Value
ComSpec
C:\Windows\system32\cmd.exe
FP_NO_HOST_C...
NO
NUMBER_OF_P...
2
OS
Windows_NT
New...
Edit...
Delete
OK
Cancel

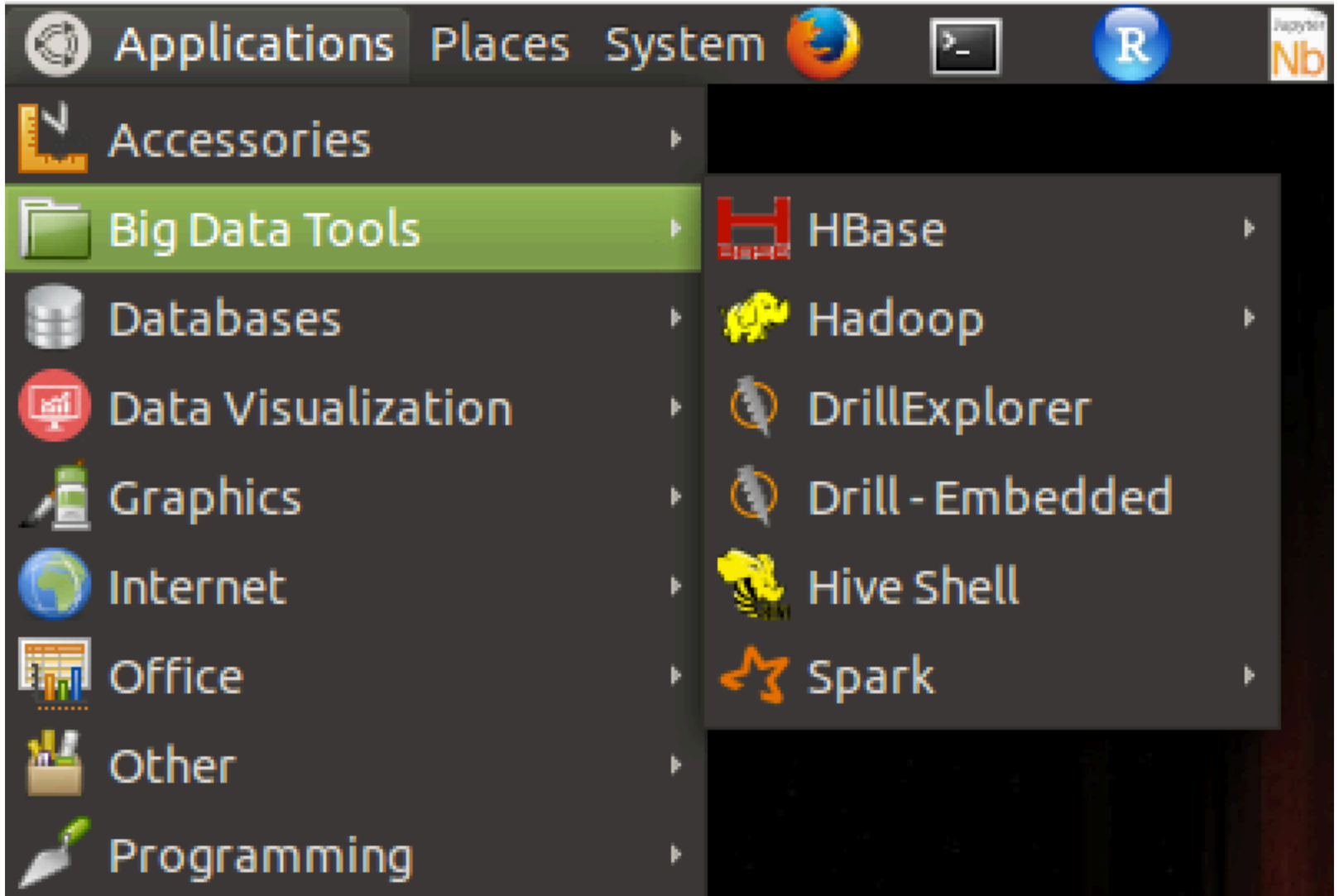
Starting Drill

Embedded Mode: For use on a standalone system

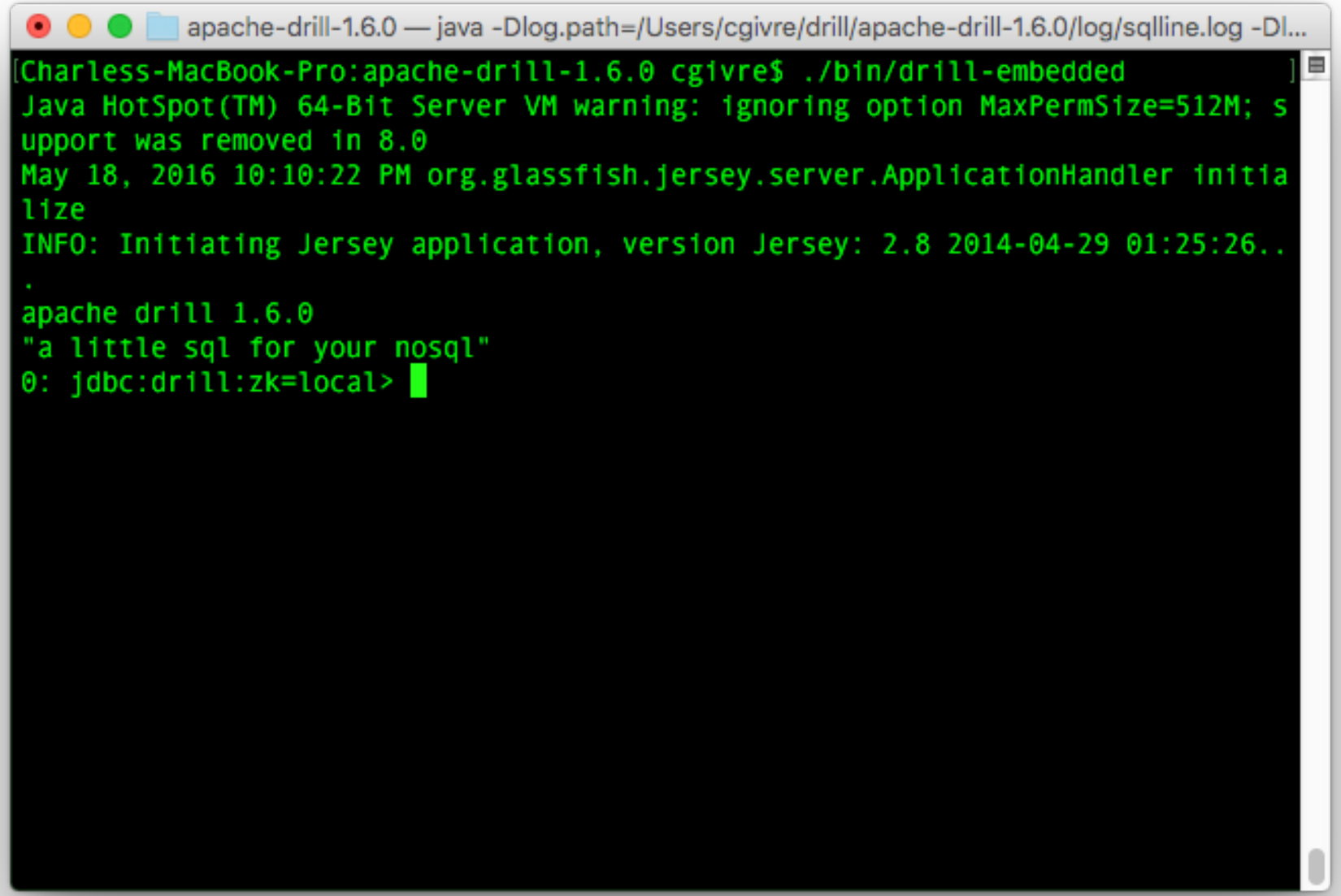
```
$./bin/drill-embedded
```

```
sqlline.bat -u "jdbc:drill:zk=local"
```

Starting Drill

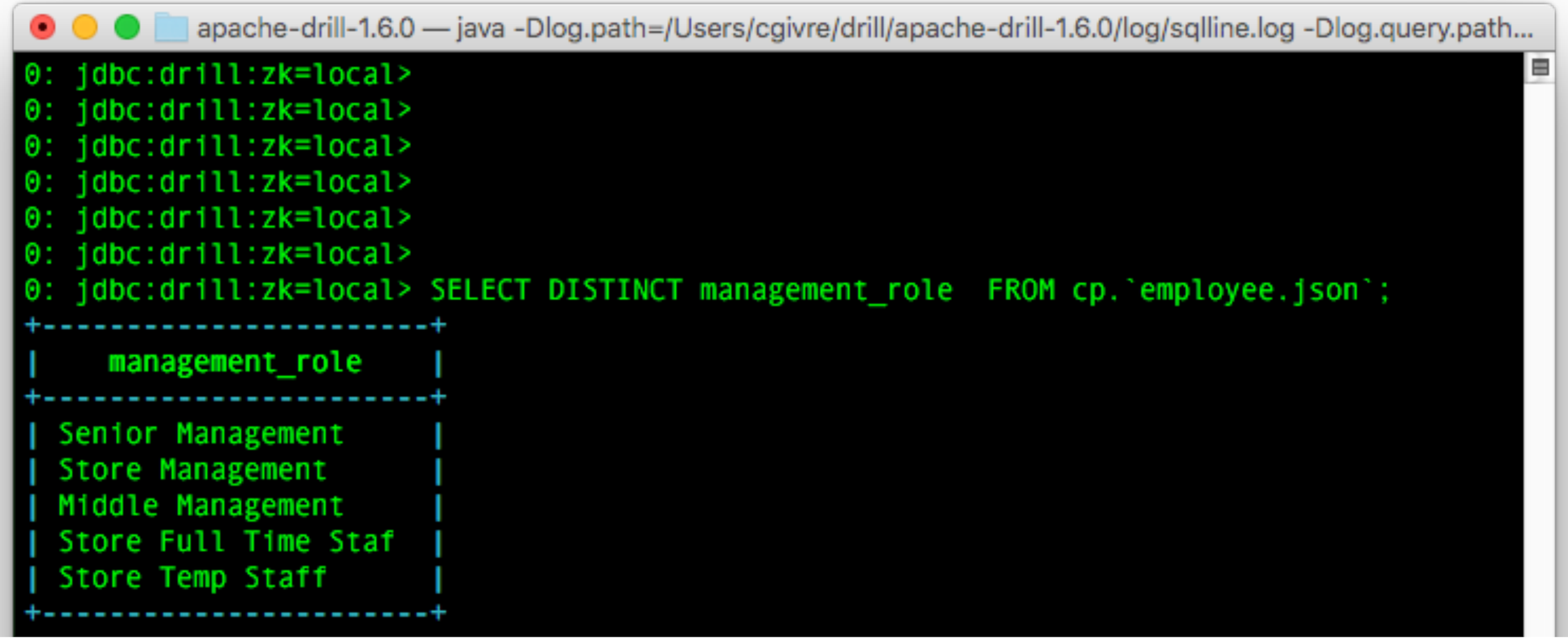


Drill's Command Line Interface

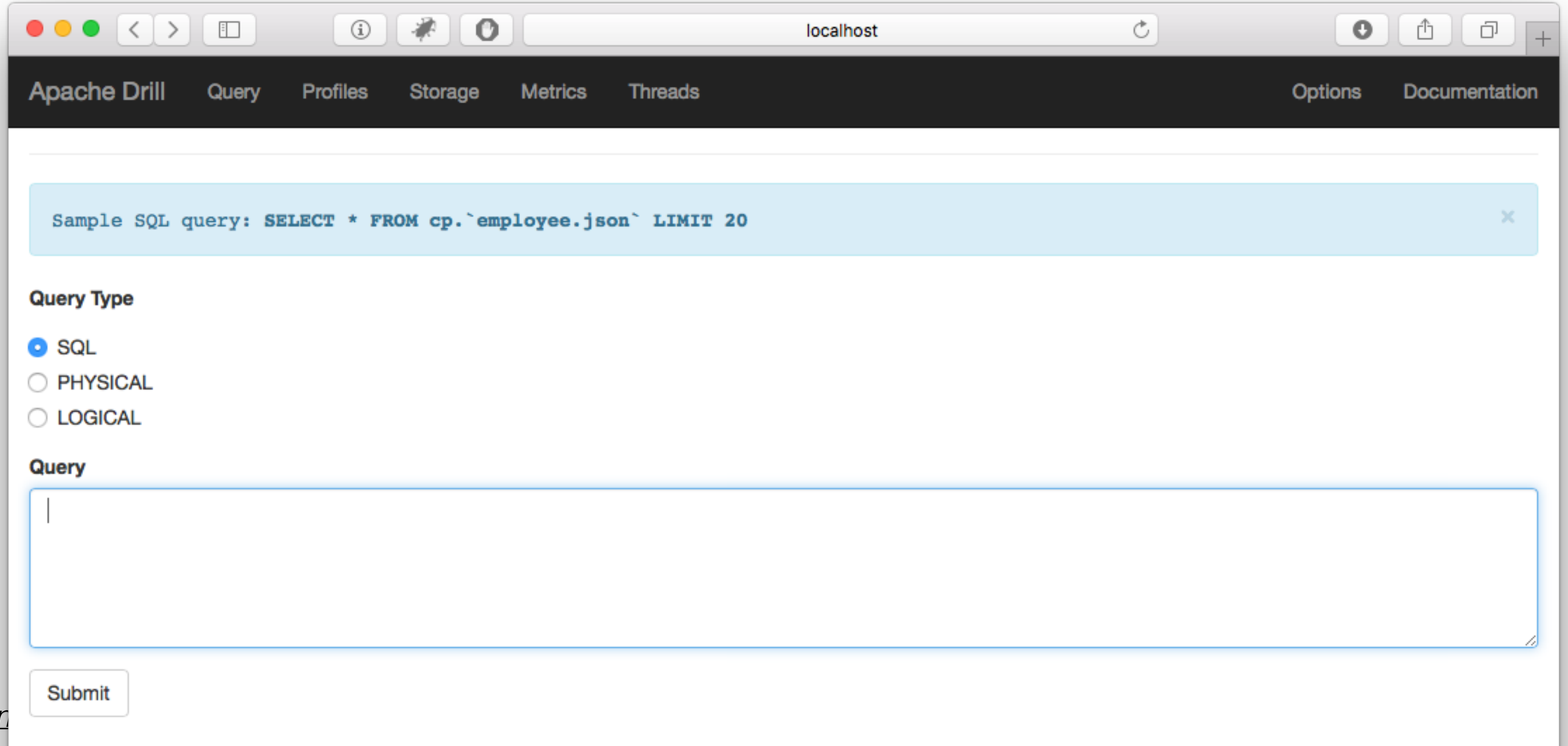


Drill's Command Line Interface

```
SELECT DISTINCT management_role FROM
cp.`employee.json`;
```

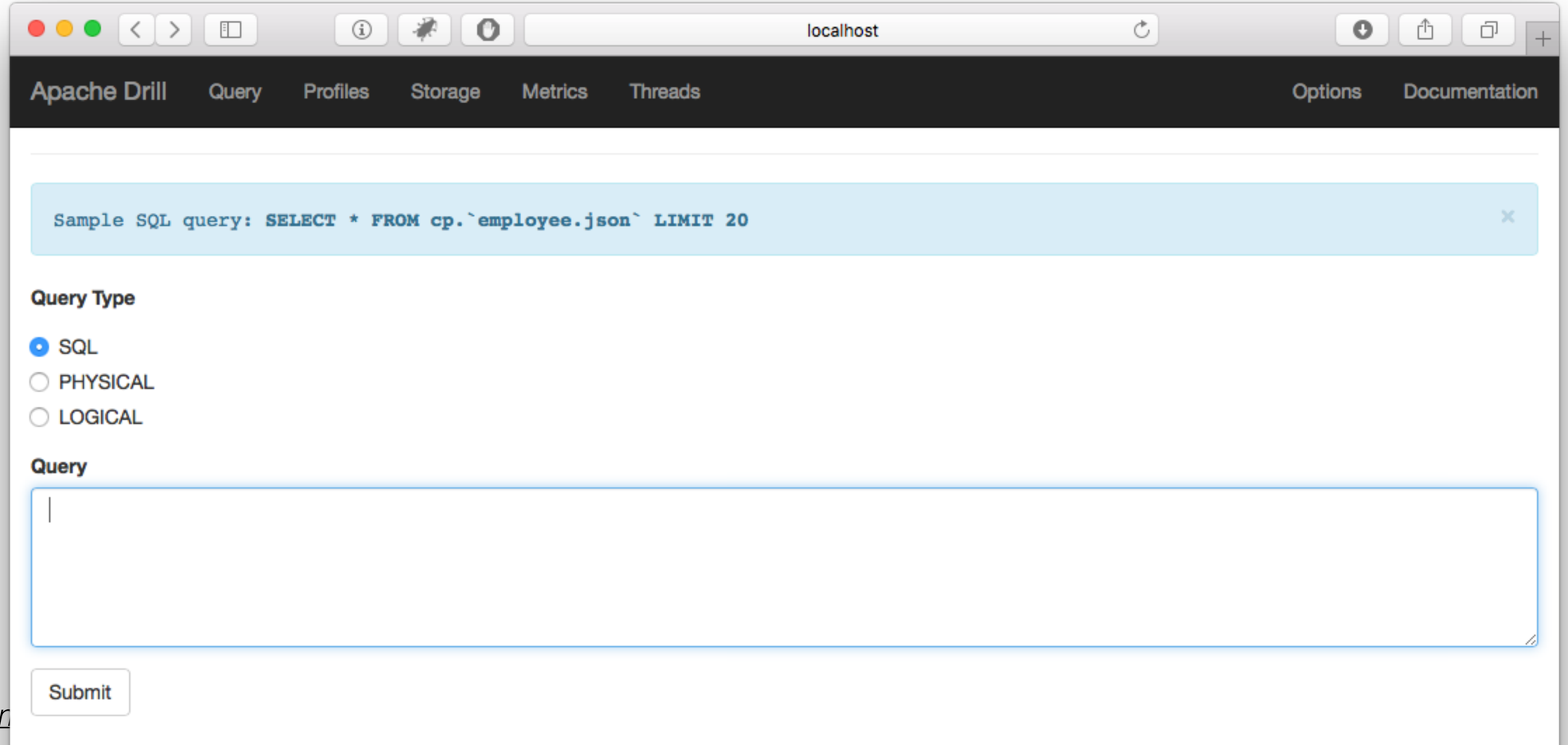


http://localhost:8047



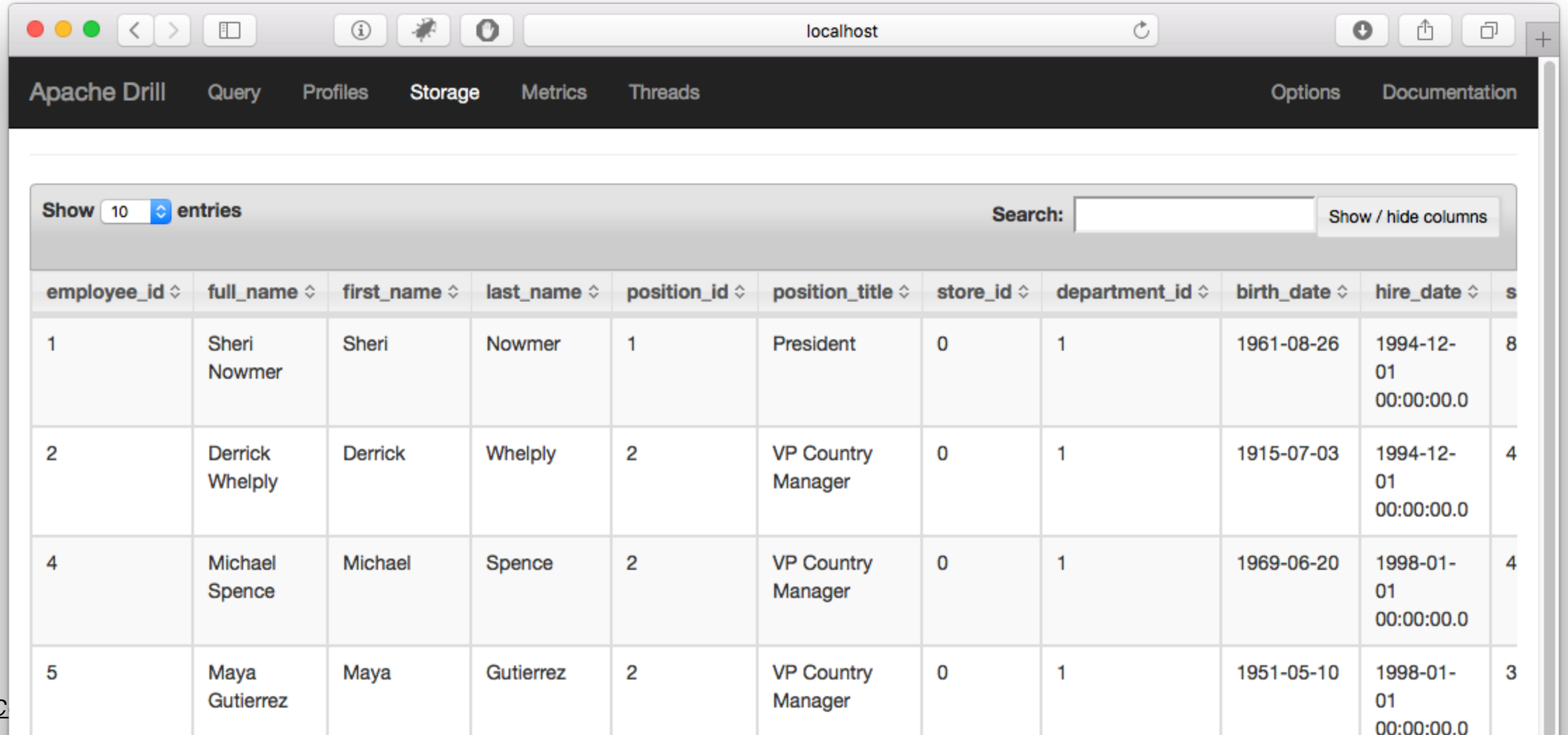
GET ON THE FAST TRACK

SELECT * FROM cp.`employee.json` LIMIT 20



GTK CYBER
GET ON THE FAST TRACK

```
SELECT * FROM cp.`employee.json` LIMIT 20
```



employee_id	full_name	first_name	last_name	position_id	position_title	store_id	department_id	birth_date	hire_date	s
1	Sheri Nowmer	Sheri	Nowmer	1	President	0	1	1961-08-26	1994-12-01 00:00:00.0	8
2	Derrick Whelply	Derrick	Whelply	2	VP Country Manager	0	1	1915-07-03	1994-12-01 00:00:00.0	4
4	Michael Spence	Michael	Spence	2	VP Country Manager	0	1	1969-06-20	1998-01-01 00:00:00.0	4
5	Maya Gutierrez	Maya	Gutierrez	2	VP Country Manager	0	1	1951-05-10	1998-01-01 00:00:00.0	3

Workspaces in Drill

- Workspaces are shortcuts to the file system. You'll want to use them when you have lengthy file paths.
- They work in any "file based" storage plugin (IE: S3, Hadoop, Local File System)

Workspaces in Drill

```
FROM dfs.`/Users/cgivre/github/projects/drillclass/file1.csv`
```

or

```
FROM dfs.drilldata.`file1.csv`
```

Querying Drill

Plugins Supported	Description
cp	Queries files in the Java ClassPath
dfs	File System. Can connect to remote filesystems such as Hadoop and cloud storage such as S3/Azure
hbase	Connects to HBase
hive	Integrates Drill with the Apache Hive metastore
Kafka	Used to query streaming data
kudu	Provides a connection to Apache Kudu
mongo	Connects to mongoDB
JDBC	Provides a connection to relational databases such as MySQL, Postgres, Oracle and others.

In Class Exercise: Create a Workspace

In this exercise we are going to create a workspace called '**`dsclass`**', which we will use for future exercises.

1. Open the Drill Web UI and go to Storage->dfs->update

2. Paste the following into the 'workspaces' section and click update

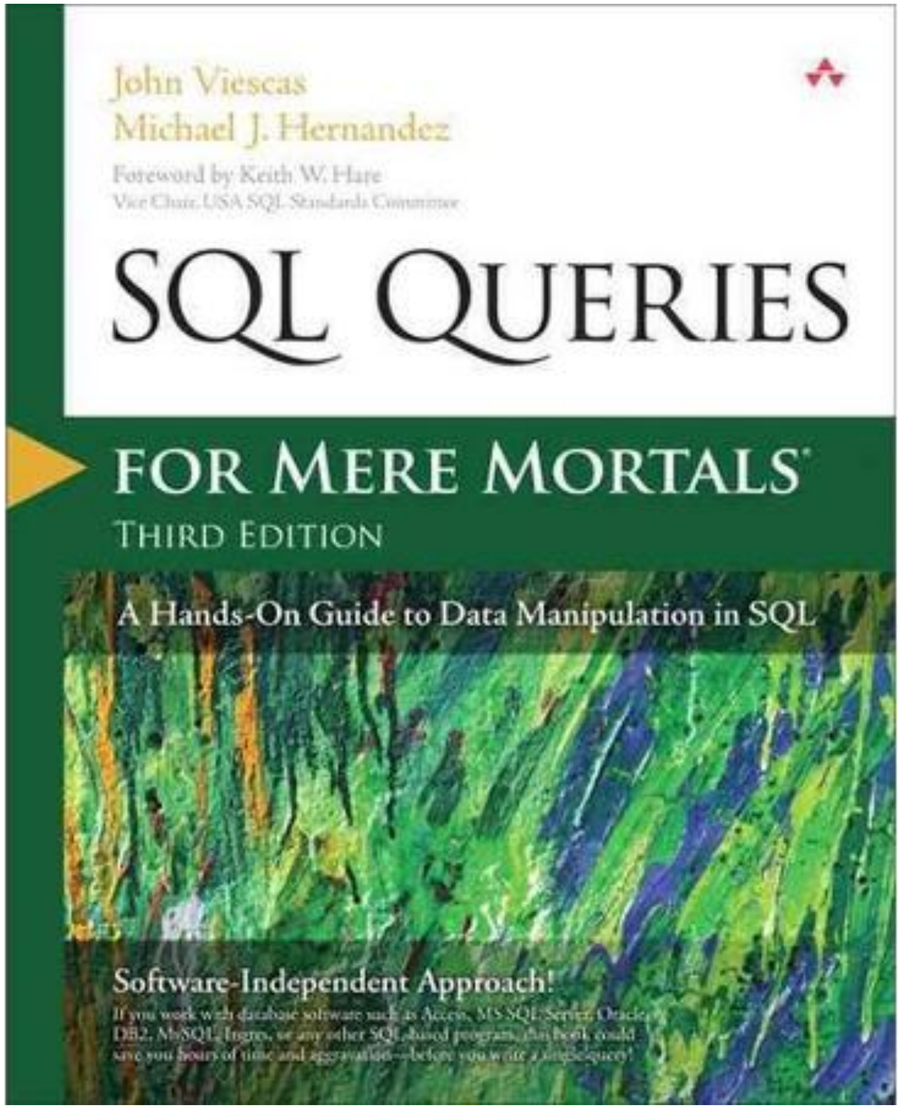
```
"drillclass": {
  "location": "<path to your files>",
  "writable": true,
  "defaultInputFormat": null
}
```

3. Execute a `show databases` query to verify that your workspace was added.

Querying Simple Delimited Data

Everything you need to know about SQL*… in 10 minutes

* well…not quite everything, but enough to get you through this session



http://amzn.to/2llD8yi


```
SELECT <fields>
FROM <data source>
```

Please open people.csv

	A	B	C	D	E	F
1	id	first_name	last_name	email	gender	ip_address
2	1	Philip	Richardson	prichardson0@samsung.com	Male	76.20.6.56
3	2	Todd	James	tjames1@hostgator.com	Male	0.118.22.98
4	3	Jimmy	Mendoza	jmendoza2@reuters.com	Male	118.191.95.200
5	4	Jose	Morris	jmorris3@example.com	Male	106.180.20.171
6	5	Dorothy	Fernandez	dfernandez4@ask.com	Female	171.75.36.3
7	6	Patrick	Bradley	pbradley5@elpais.com	Male	20.244.138.119
8	7	Nicholas	Bishop	nbishop6@fotki.com	Male	80.54.11.247
9	8	Michael	Kelly	mkelly7@imageshack.us	Male	191.190.150.242
10	9	Russell	Coleman	rcoleman8@pbs.org	Male	50.176.191.152
11	10	Frances	Rodriguez	frodriguez9@github.io	Female	23.118.13.92
12	11	Nancy	Nelson	nnelsona@biglobe.ne.jp	Female	225.182.18.156
13	12	Theresa	Russell	trussellb@hexun.com	Female	16.14.226.63
14	13	Frances	Greene	fgreenec@sbwire.com	Female	192.88.64.253
15	14	Julia	Alvarez	jalvarezd@livejournal.com	Female	197.237.43.239


```
SELECT *
FROM <data source>
```


```
SELECT first_name,
last_name,
gender
FROM <data source>
```

Tip: Use BACK TICKS around field names in Drill

```
SELECT `first_name`,
`last_name`,
`gender`
FROM <data source>
```


```
SELECT `first_name`,
`last_name`,
`gender`
FROM <data source>
```

Querying Drill

```
FROM dfs.dsclass.`/data/people.csvh`
```

Storage Plugin

Workspace

Table


```
SELECT `first_name`,
`last_name`,
`gender`
FROM
dfs.drillclass.`people.csvh`
```

Try it yourself!!


```
SELECT `first_name`,
`last_name`,
`gender`
FROM
dfs.drillclass.`people.csvh`
```

Apache Drill | Query | Profiles | Storage | Metrics | Threads | Logs | Options | Documentation

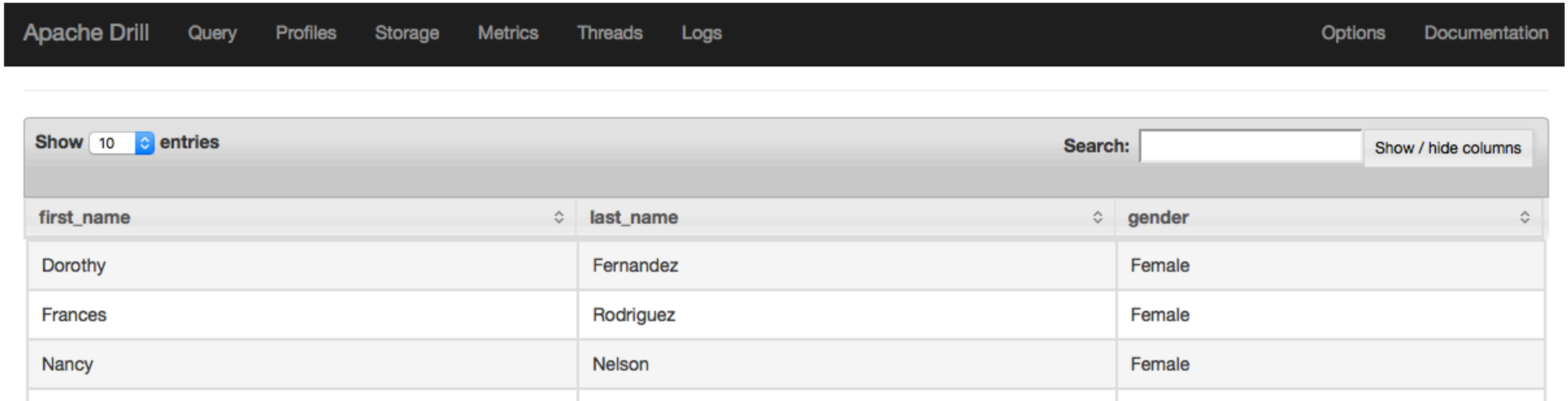
Show 10 entries | Search: | Show / hide columns

first_name	last_name	gender
Philip	Richardson	Male
Todd	James	Male
Jimmy	Mendoza	Male


```
SELECT <fields>
FROM <data source>
WHERE <logical condition>
```


```
SELECT `first_name`,
`last_name`,
`gender`
FROM dfs.drillclass.`people.csvh`
WHERE `gender` = 'Female'
```

```
SELECT `first_name`,
`last_name`,
`gender`
FROM dfs.drillclass.`people.csvh`
WHERE `gender` = 'Female'
```

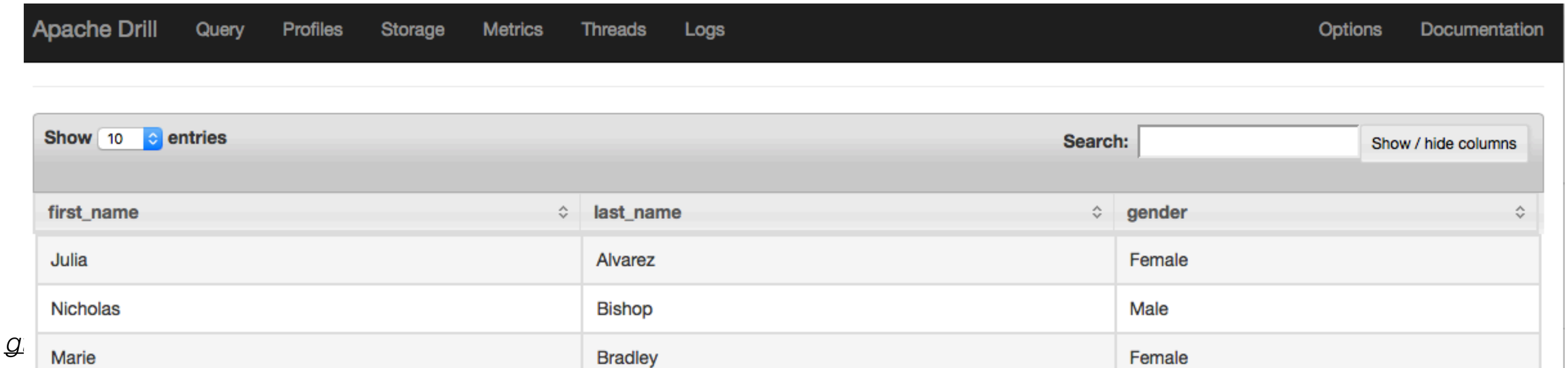


first_name	last_name	gender
Dorothy	Fernandez	Female
Frances	Rodriguez	Female
Nancy	Nelson	Female


```
SELECT <fields>
FROM <data source>
WHERE <logical condition>
ORDER BY <field> (ASC|DESC)
```


```
SELECT `first_name`,
`last_name`,
`gender`
FROM dfs.drillclass.`people.csvh`
ORDER BY `last_name`, `first_name` ASC
```

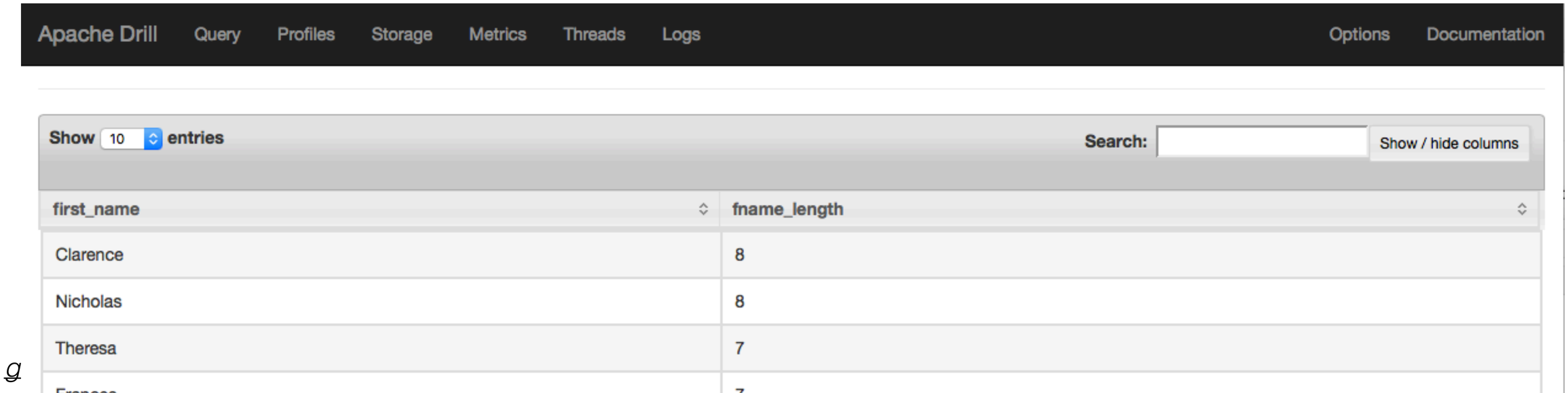
```
SELECT `first_name`,
`last_name`,
`gender`
FROM dfs.drillclass.`people.csvh`
ORDER BY `last_name`, `first_name` ASC
```




```
SELECT
FUNCTION( <field> ) AS new_field
FROM <data source>
```


```
SELECT first_name,
LENGTH( `first_name` ) AS
fname_length
FROM dfs.drillclass.`people.csvh`
ORDER BY fname_length DESC
```

```
SELECT first_name,
LENGTH( `first_name` ) AS fname_length
FROM dfs.drillclass.`people.csvh`
ORDER BY fname_length DESC
```



g


```
SELECT <fields>
FROM <data source>
GROUP BY <field>
```


```
SELECT `gender`,
COUNT( * ) AS gender_count
FROM dfs.drillclass.`people.csvh`
GROUP BY `gender`
```


```
SELECT `gender`,
COUNT( * ) AS gender_count
FROM dfs.drillclass.`people.csvh`
GROUP BY `gender`
```

Apache Drill | Query | Profiles | Storage | Metrics | Threads | Logs | Options | Documentation

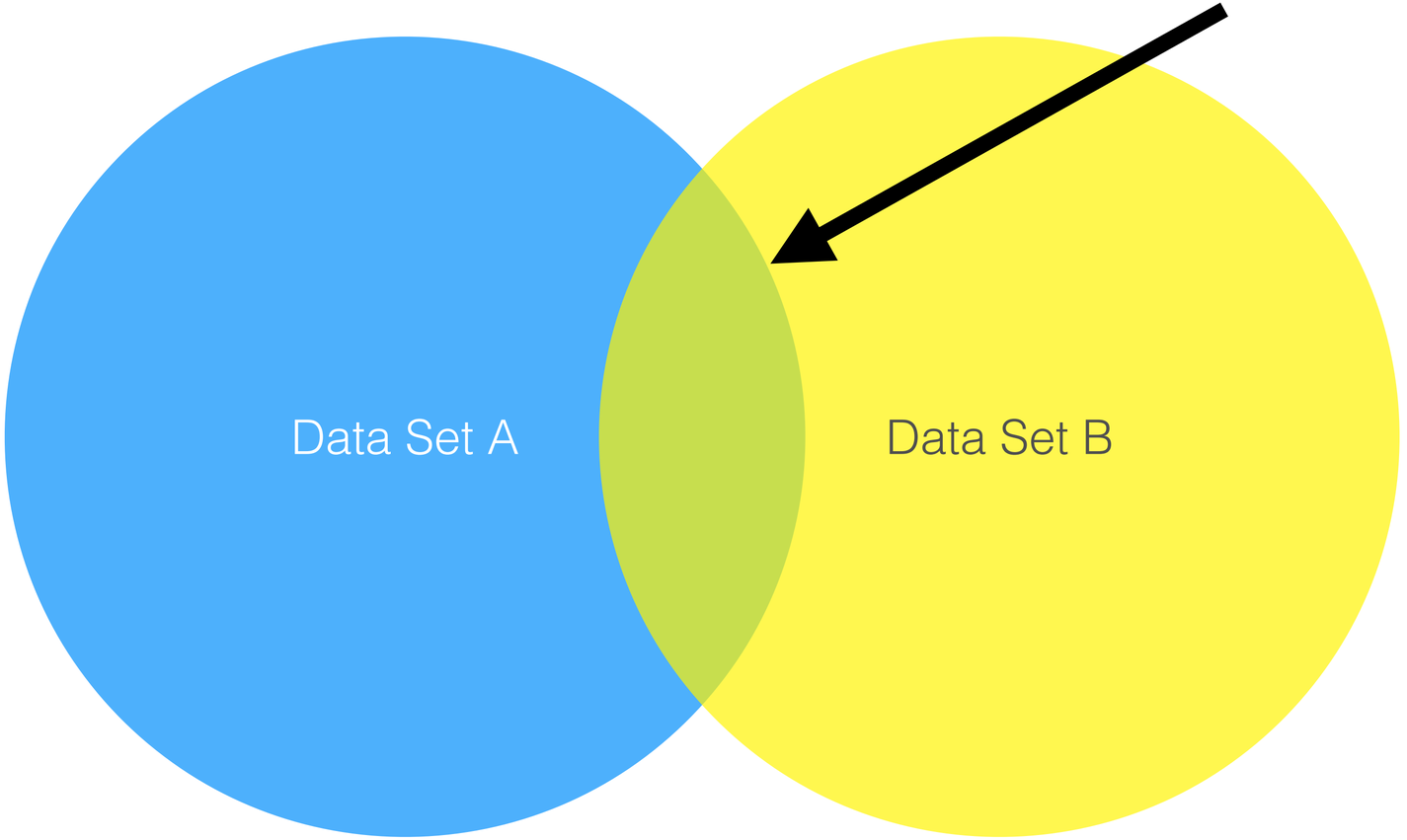
Show 10 entries | Search: | Show / hide columns

gender	gender_count
Male	11
Female	9

Joining Datasets

GTK CYBER
GET ON THE FAST TRACK

Referred to as an Inner Join




```
SELECT <fields>
FROM <data source 1> AS table1
INNER JOIN <data source 2> AS table2
ON table1.`id` = table2.`id`
```

In Class Exercise:
Create a Simple Report

For this exercise we will use the baltimore_salaries_2016.csvh file.

1. Create a query which returns each person's: name, jobtitle, and gross pay.

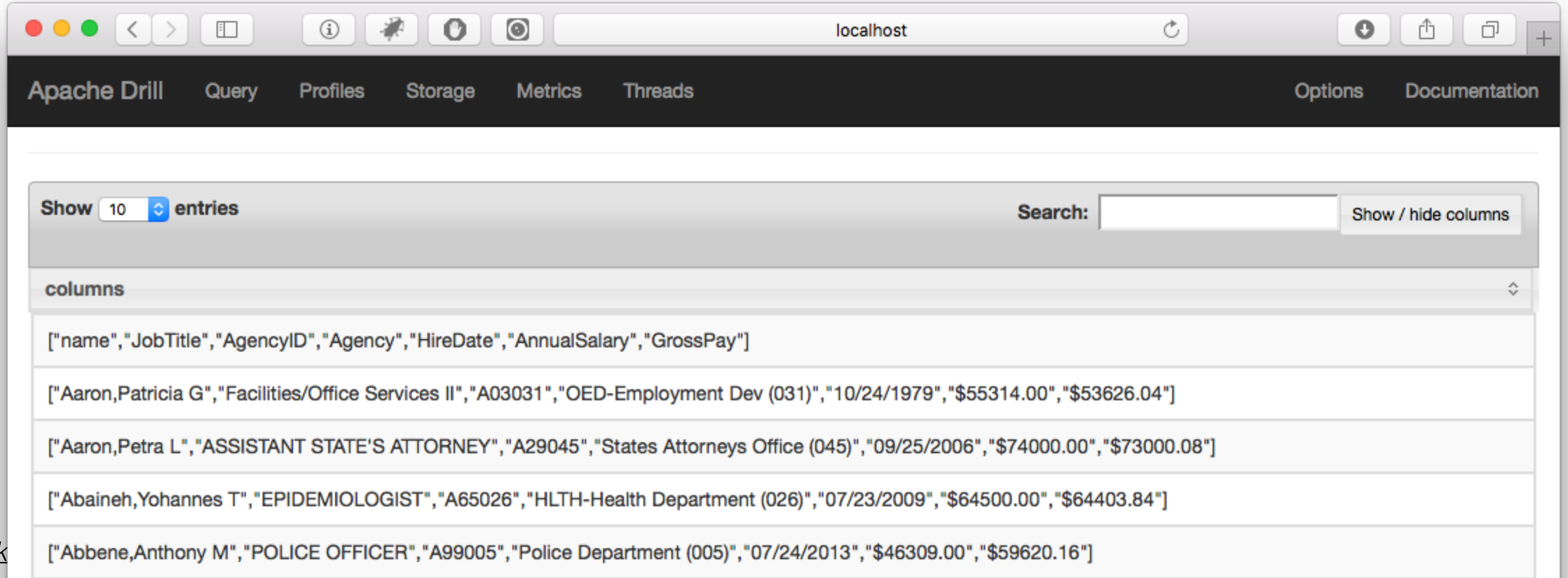
2. Create a report which contains each employee's name, job title, 2015 salary and 2016 salary. NOTE: This query requires the use of a JOIN.


```
SELECT EmpName, JobTitle, GrossPay
FROM dfs.drillclass.`baltimore_salaries_2016.csvh`
LIMIT 10
```


```
SELECT data2016.`EmpName`,
data2016.`JobTitle`,
data2016.`AnnualSalary` AS salary_2016,
data2015.`AnnualSalary` AS salary_2015
FROM dfs.drillclass.`baltimore_salaries_2016.csvh` AS data2016
INNER JOIN dfs.drillclass.`baltimore_salaries_2015.csvh` AS data2015
ON data2016.`EmpName` = data2015.`EmpName`
```

Drill Data Types

```
SELECT *
FROM dfs.drillclass.`baltimore_salaries_2016.csv`
LIMIT 10
```



Drill Data Types

Simple Data Types

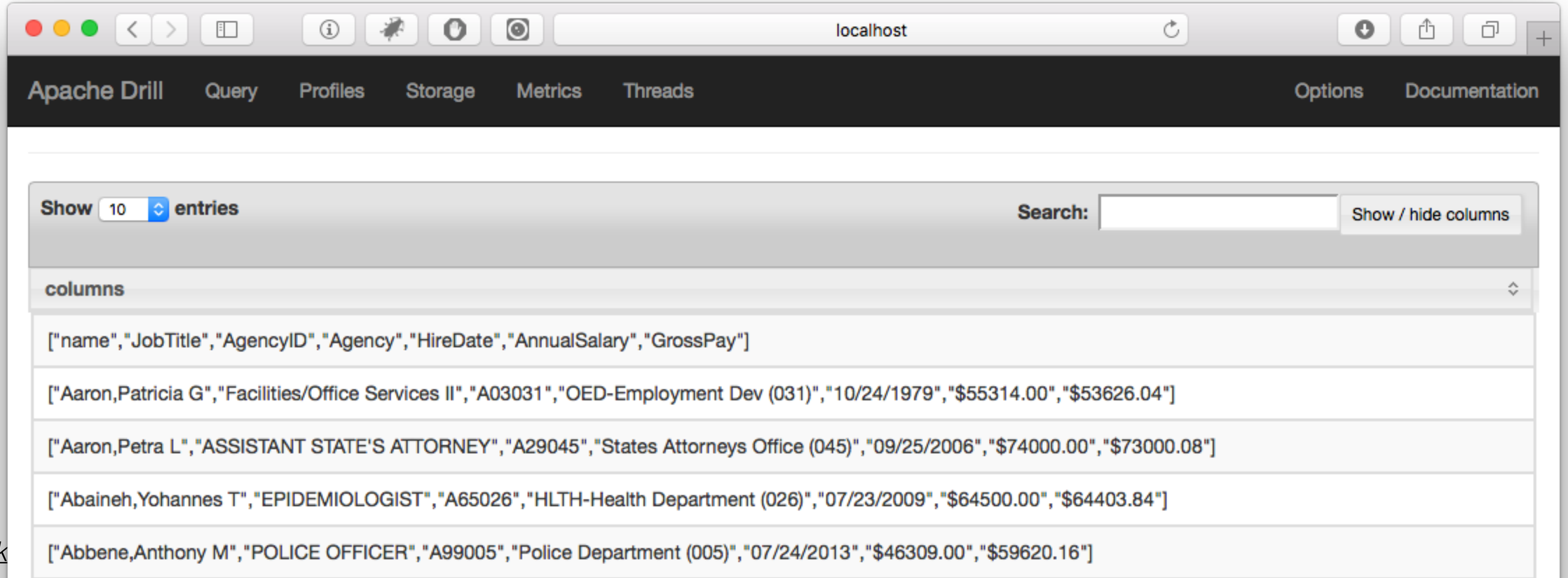
- Integer/BigInt/SmallInt
- Float/Decimal/Double
- Varchar/Binary
- Date/Time/Interval/Timestamp

Complex Data Types

- Arrays
- Maps

Querying Drill

```
["Aaron, Patricia G" "Facilities/Office Services"…]
```



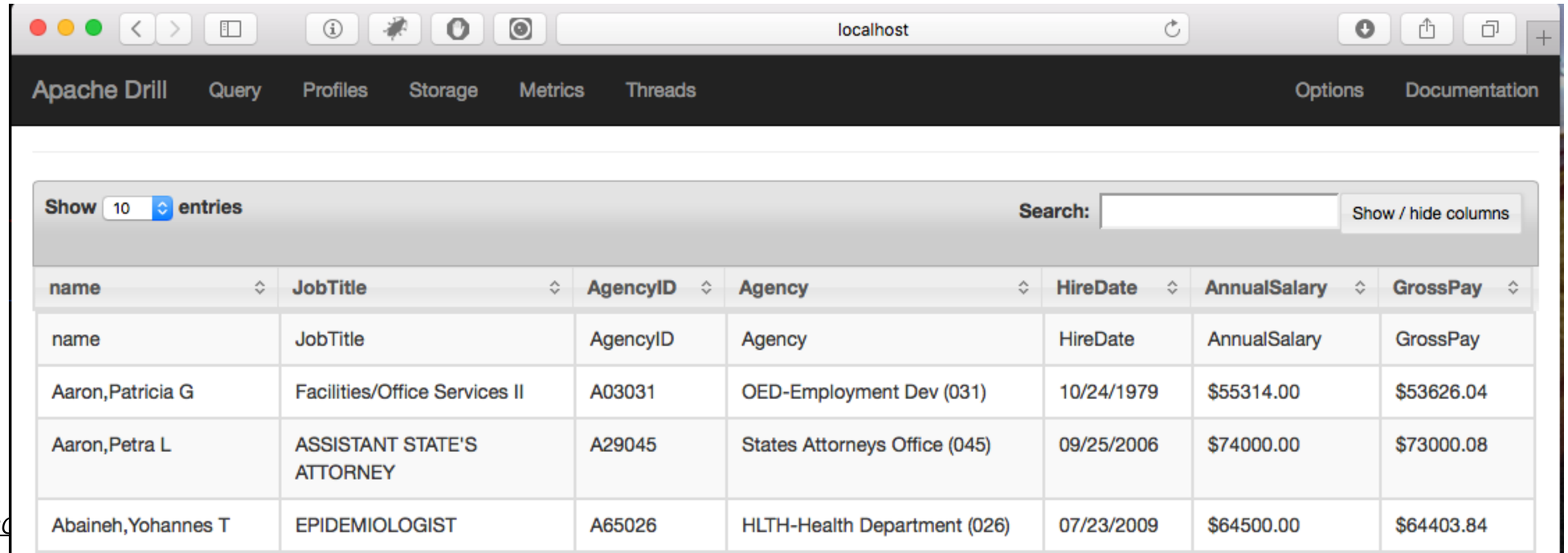
columns[*n*]

Querying Drill

```
SELECT columns[0] AS name,
columns[1] AS JobTitle,
columns[2] AS AgencyID,
columns[3] AS Agency,
columns[4] AS HireDate,
columns[5] AS AnnualSalary,
columns[6] AS GrossPay
FROM dfs.drillclass.`csv/baltimore_salaries_2016.csv`
LIMIT 10
```

Querying Drill

```
SELECT columns[0] AS name,
columns[1] AS JobTitle,
. . .
FROM dfs.drillclass.`csv/baltimore_salaries_2016.csv`
LIMIT 10
```



name	JobTitle	AgencyID	Agency	HireDate	AnnualSalary	GrossPay
name	JobTitle	AgencyID	Agency	HireDate	AnnualSalary	GrossPay
Aaron,Patricia G	Facilities/Office Services II	A03031	OED-Employment Dev (031)	10/24/1979	$55314.00	$53626.04
Aaron,Petra L	ASSISTANT STATE'S ATTORNEY	A29045	States Attorneys Office (045)	09/25/2006	$74000.00	$73000.08
Abaineh,Yohannes T	EPIDEMIOLOGIST	A65026	HLTH-Health Department (026)	07/23/2009	$64500.00	$64403.84

Querying Drill

```
"csvh": {
  "type": "text",
  "extensions": [
    "csvh"
  ],
  "extractHeader": true,
  "delimiter": ","
}
```

Querying Drill

File Extension	File Type
.psv	Pipe separated values
.csv	Comma separated value files
.csvh	Comma separated value with header
.tsv	Tab separated values
.json	JavaScript Object Notation files
.avro	Avro files (experimental)
.seq	Sequence Files
.httpd	Apache Web Server logs (As of version 1.9)
.pcap / .pcapng	Packet Capture (As of version 1.11)
.ltsv	Labeled Tab Separated Values

Querying Drill

Options	Description
comment	What character is a comment character
escape	Escape character
delimiter	The character used to delimit fields
quote	Character used to enclose fields
skipFirstLine	true/false
extractHeader	Reads the header from the CSV file

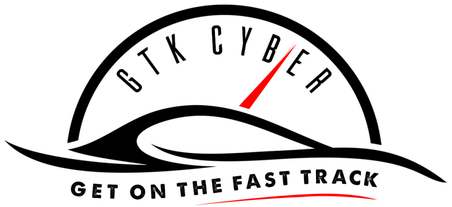
Config at Querytime

```
SELECT *
FROM table(
  dfs.drillclass.`baltimore_salaries_2016.csv`
  (
    type => 'text',
    extractHeader => true,
    fieldDelimiter => ','
  )
)
```

Problem: Find the average salary of each Baltimore City job title

Aggregate Functions

Function	Argument Type	Return Type
AVG(expression)	Integer or Floating point	Floating point
COUNT(*)	any	BIGINT
COUNT([DISTINCT] <expression>)	any	BIGINT
MIN/MAX(<expression>)	Any numeric or date	same as argument
SUM(<expression>)	Any numeric or interval	same as argument



Querying Drill

```
SELECT JobTitle, AVG( AnnualSalary) AS avg_salary,
COUNT( DISTINCT name ) AS number
FROM drillclass.`baltimore_salaries_2016.csvh`
GROUP BY JobTitle
Order By avg_salary DESC
```

Querying Drill

Query Failed: An Error Occurred

org.apache.drill.common.exceptions.UserRemoteException: SYSTEM ERROR: SchemaChangeException: Failure while trying to materialize incoming schema. Errors: Error in expression at index -1. Error: Missing function implementation: [castINT(BIT-OPTIONAL)]. Full expression: --UNKNOWN EXPRESSION--.. Fragment 0:0 [Error Id: af88883b-f10a-4ea5-821d-5ff065628375 on 10.251.255.146:31010]

Querying Drill

```
SELECT JobTitle, AVG( AnnualSalary) AS avg_salary,
COUNT( DISTINCT name ) AS number
FROM dfs.drillclass.`baltimore_salaries_2016.csvh`
GROUP BY JobTitle
Order By avg_salary DESC
```

Querying Drill

```
SELECT JobTitle,
AVG( AnnualSalary ) AS avg_salary,
COUNT( DISTINCT name ) AS number
FROM dfs.drillclass.`baltimore_salaries_2016.csvh`
GROUP BY JobTitle
Order By avg_salary DESC
```

AnnualPay has extra characters

AnnualPay is a string

Querying Drill

Function	Return Type
BYTE_SUBSTR	BINARY or VARCHAR
CHAR_LENGTH	INTEGER
CONCAT	VARCHAR
ILIKE	BOOLEAN
INITCAP	VARCHAR
LENGTH	INTEGER
LOWER	VARCHAR
LPAD	VARCHAR
LTRIM	VARCHAR
POSITION	INTEGER
REGEXP_REPLACE	VARCHAR
RPAD	VARCHAR
RTRIM	VARCHAR
SPLIT	ARRAY
STRPOS	INTEGER
SUBSTR	VARCHAR
TRIM	VARCHAR
UPPER	VARCHAR

In Class Exercise:
Clean the field.

In this exercise you will use one of the string functions to remove the dollar sign from the ‘AnnualPay’ column.

Complete documentation can be found here:

https://drill.apache.org/docs/string-manipulation/

In Class Exercise:
Clean the field.

In this exercise you will use one of the string functions to remove the dollar sign from the ‘AnnualPay’ column.

Complete documentation can be found here:

https://drill.apache.org/docs/string-manipulation/

```
SELECT LTRIM( AnnualPay, '$' ) AS annualPay
FROM dfs.drillclass.`baltimore_salaries_2016.csvh`
```

Drill Data Types

Data type	Description
Bigint	8 byte signed integer
Binary	Variable length byte string
Boolean	True/false
Date	yyyy-mm-dd
Double / Float	8 or 4 byte floating point number
Integer	4 byte signed integer
Interval	A day-time or year-month interval
Time	HH:mm:ss
Timestamp	JDBC Timestamp
Varchar	UTF-8 encoded variable length string


```
cast( <expression> AS <data type> )
```

In Class Exercise: Convert to a number

In this exercise you will use the cast() function to convert AnnualPay into a number.

Complete documentation can be found here:

https://drill.apache.org/docs/data-type-conversion/#cast

In Class Exercise:
Convert to a number

In this exercise you will use the cast() function to convert AnnualPay into a number.

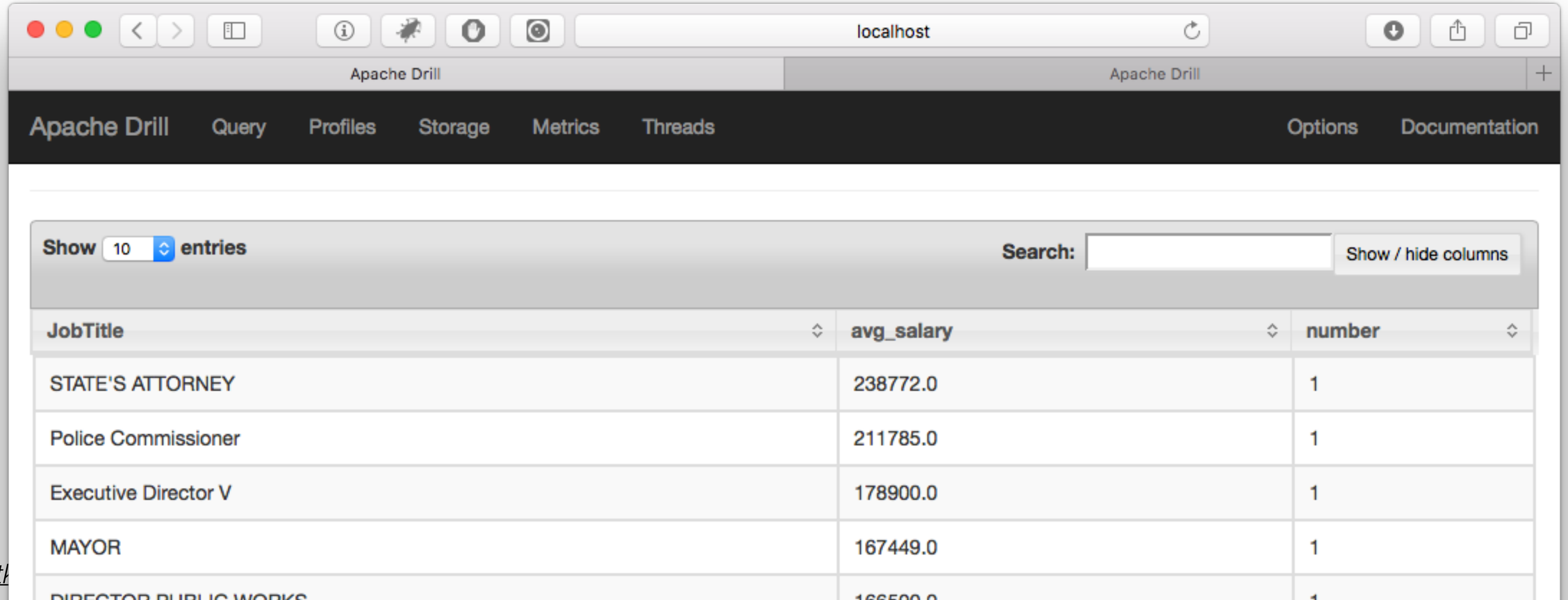
Complete documentation can be found here:

https://drill.apache.org/docs/data-type-conversion/#cast

```
SELECT CAST( LTRIM( AnnualPay, '$' ) AS FLOAT ) AS
annualPay
FROM dfs.drillclass.`csv/baltimore_salaries_2016.csvh`
```


```
SELECT JobTitle,
AVG(
  CAST(
    LTRIM( AnnualSalary, '$' ) AS FLOAT ) ) AS avg_salary,
COUNT( DISTINCT name ) AS number
FROM dfs.drillclass.`baltimore_salaries_2016.csvh`
GROUP BY JobTitle
Order By avg_salary DESC
```

```
SELECT JobTitle,
AVG( CAST( LTRIM( AnnualSalary, '$' ) AS FLOAT) ) AS avg_salary,
COUNT( DISTINCT name ) AS number
FROM dfs.drillclass.`baltimore_salaries_2016.csvh`
GROUP BY JobTitle
Order By avg_salary DESC
```



JobTitle	avg_salary	number
STATE'S ATTORNEY	238772.0	1
Police Commissioner	211785.0	1
Executive Director V	178900.0	1
MAYOR	167449.0	1

TO_NUMBER(<field>, <format>)

TO_NUMBER(<field>, <format>)

Symbol	Meaning
0	Digit
#	Digit, zero shows as absent
.	Decimal separator or monetary separator
-	Minus Sign
,	Grouping Separator
%	Multiply by 100 and show as percentage
‰ \u2030	Multiply by 1000 and show as per mille value
¤ \u00A4	Currency symbol

In Class Exercise:
Convert to a number using **TO_NUMBER()**

In this exercise you will use the **TO_NUMBER()** function to convert AnnualPay into a numeric field.

Complete documentation can be found here:

https://drill.apache.org/docs/data-type-conversion/#to_number

In Class Exercise:
Convert to a number using **TO_NUMBER()**

In this exercise you will use the **TO_NUMBER()** function to convert AnnualPay into a numeric field.

Complete documentation can be found here:

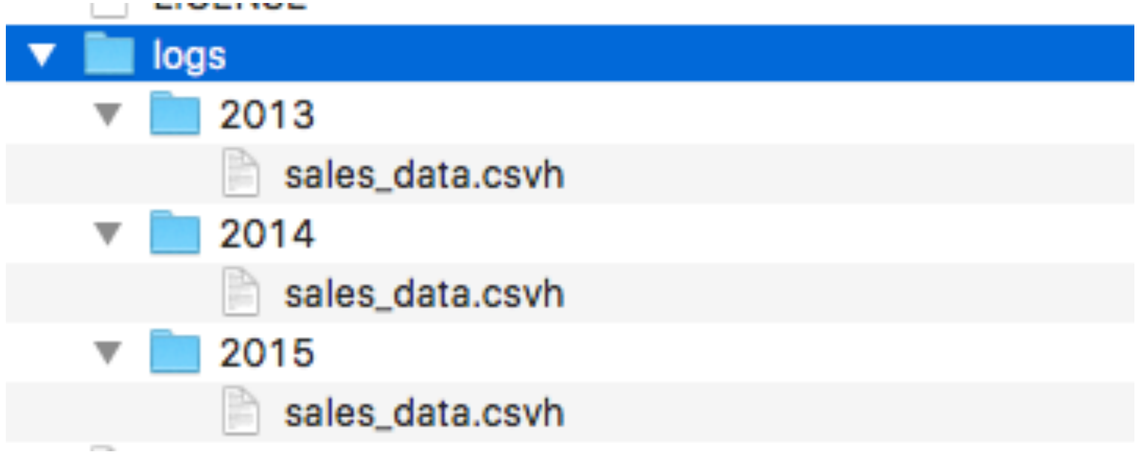
https://drill.apache.org/docs/data-type-conversion/#to_number

```
SELECT JobTitle,
AVG( TO_NUMBER( AnnualSalary, '¤' )) AS avg_salary,
COUNT( DISTINCT `EmpName` ) AS number
FROM dfs.drillclass.`baltimore_salaries_2016.csvh`
GROUP BY JobTitle
Order BY avg_salary DESC
```

Problem: You have multiple log files which you would like to analyze

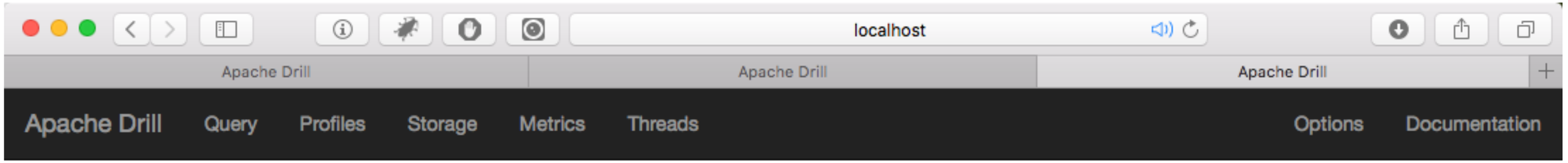
Problem: You have multiple log files which you would like to analyze

- In the sample data files, there is a folder called ‘logs’ which contains the following structure:




```
SELECT *
FROM
dfs.drillclass.`logs/`
LIMIT 10
```


```
SELECT *
FROM
dfs.drillworkshop.`logs/`
LIMIT 10
```



Show 10 entries Search: Show / hide columns

customer_id	item_count	amount_spent	dir0
1169	2	1.05	2013
813	4	9.76	2013
373	1	6.69	2013
877	3	6.28	2013
959	4	1.74	2013

dir***n*** accesses the subdirectories

dir*n* accesses the subdirectories

```
SELECT *
FROM dfs.drilldata.`logs/`
WHERE dir0 = '2013'
```

Directory Functions

Function	Description
MAXDIR(), MINDIR()	Limit query to the first or last directory
IMAXDIR(), IMINDIR()	Limit query to the first or last directory in case insensitive order.

```
WHERE dir<n> = MAXDIR('<plugin>.<workspace>', '<filename>'
```

In Class Exercise:

Find the total number of items sold by year and the total dollar sales in each year.

HINT: Don’t forget to CAST() the fields to appropriate data types

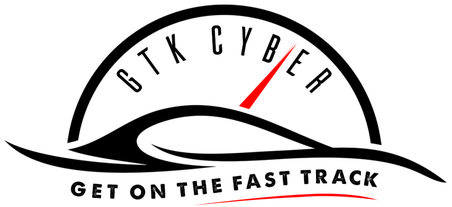
In Class Exercise:

Find the total number of items sold by year and the total dollar sales in each year.

HINT: Don’t forget to CAST() the fields to appropriate data types

```
SELECT dir0 AS data_year,
SUM( CAST( item_count AS INTEGER ) ) as total_items,
SUM( CAST( amount_spent AS FLOAT ) ) as total_sales
FROM dfs.drillworkshop.`logs/`
GROUP BY dir0
```

Let’s look at JSON data



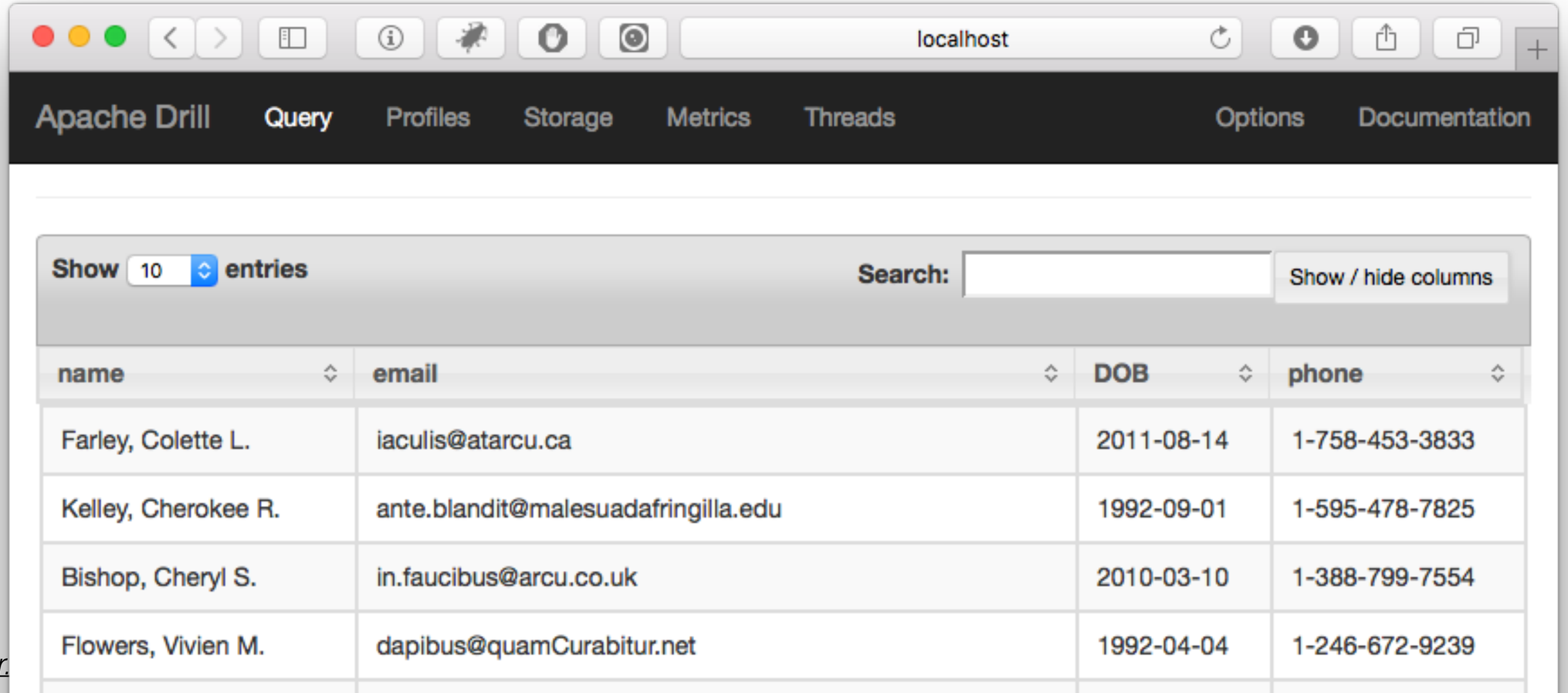
Let’s look at JSON data

```
[
    {
        "name": "Farley, Colette L.",
        "email": "iaculis@atarcu.ca",
        "DOB": "2011-08-14",
        "phone": "1-758-453-3833"
    },
    {
        "name": "Kelley, Cherokee R.",
        "email": "ante.blandit@malesuadafringilla.edu",
        "DOB": "1992-09-01",
        "phone": "1-595-478-7825"
    }
…
]
```

Let’s look at JSON data

```
SELECT *
FROM dfs.drillclass.`customers.json`
```

```
SELECT *
FROM dfs.drillclass.`customers.json`
```



GTK CYBER

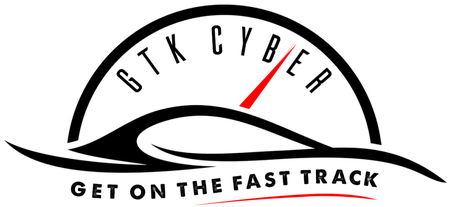
GET ON THE FAST TRACK

What about nested data?

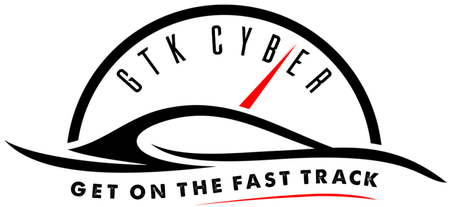
Please open

`baltimore_salaries.json`

in a text editor




```
{
  "meta" : {
    "view" : {
      "id" : "nsfe-bg53",
      "name" : "Baltimore City Employee Salaries FY2015",
      "attribution" : "Mayor's Office",
      "averageRating" : 0,
      "category" : "City Government",
      …
      "       "format" : { }
      },
},
  "data" : [ [ 1, "66020CF9-8449-4464-AE61-B2292C7A0F2D", 1,
1438255843, "393202", 1438255843, "393202", null, "Aaron,Patricia G",
"Facilities/Office Services II", "A03031", "OED-Employment Dev (031)",
"1979-10-24T00:00:00", "55314.00", "53626.04" ]
, [ 2, "31C7A2FE-60E6-4219-890B-AFF01C09EC65", 2, 1438255843,
"393202", 1438255843, "393202", null, "Aaron,Petra L", "ASSISTANT
STATE'S ATTORNEY", "A29045", "States Attorneys Office (045)",
"2006-09-25T00:00:00", "74000.00", "73000.08" ]
```




```
{
  "meta" : {
    "view" : {
      "id" : "nsfe-bg53",
      "name" : "Baltimore City Employee Salaries FY2015",
      "attribution" : "Mayor's Office",
      "averageRating" : 0,
      "category" : "City Government",
      …
      "       "format" : { }
      },
},
  "data" : [ [ 1, "66020CF9-8449-4464-AE61-B2292C7A0F2D", 1,
1438255843, "393202", 1438255843, "393202", null, "Aaron,Patricia G",
"Facilities/Office Services II", "A03031", "OED-Employment Dev (031)",
"1979-10-24T00:00:00", "55314.00", "53626.04" ]
, [ 2, "31C7A2FE-60E6-4219-890B-AFF01C09EC65", 2, 1438255843,
"393202", 1438255843, "393202", null, "Aaron,Petra L", "ASSISTANT
STATE'S ATTORNEY", "A29045", "States Attorneys Office (045)",
"2006-09-25T00:00:00", "74000.00", "73000.08" ]
```

```
{
  "meta" : {
    "view" : {
      "id" : "nsfe-bg53",
      "name" : "Baltimore City Employee Salaries FY2015",
      "attribution" : "Mayor's Office",
      "averageRating" : 0,
      "category" : "City Government",
      …
      "       "format" : { }
      },
},
  "data" : [ [ 1, "66020CF9-8449-4464-AE61-B2292C7A0F2D", 1,
1438255843, "393202", 1438255843, "393202", null, "Aaron,Patricia G",
"Facilities/Office Services II", "A03031", "OED-Employment Dev (031)",
"1979-10-24T00:00:00", "55314.00", "53626.04" ]
, [ 2, "31C7A2FE-60E6-4219-890B-AFF01C09EC65", 2, 1438255843,
"393202", 1438255843, "393202", null, "Aaron,Petra L", "ASSISTANT
STATE'S ATTORNEY", "A29045", "States Attorneys Office (045)",
"2006-09-25T00:00:00", "74000.00", "73000.08" ]
```


```
"data" : [
   [ 1,
   "66020CF9-8449-4464-AE61-B2292C7A0F2D",
   1,
   1438255843,
   "393202",
   1438255843,
   “393202",
    null,
   "Aaron,Patricia G",
   "Facilities/Office Services II",
   "A03031",
   "OED-Employment Dev (031)",
   "1979-10-24T00:00:00",
   “55314.00",
    “53626.04"
    ]
```

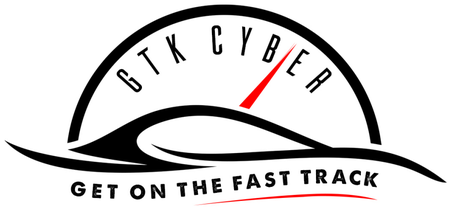
Drill has a series of functions for nested data

Let’s look at this data in Drill

Please run

```
ALTER SYSTEM SET `store.json.all_text_mode` = true;
```

in the Drill command line

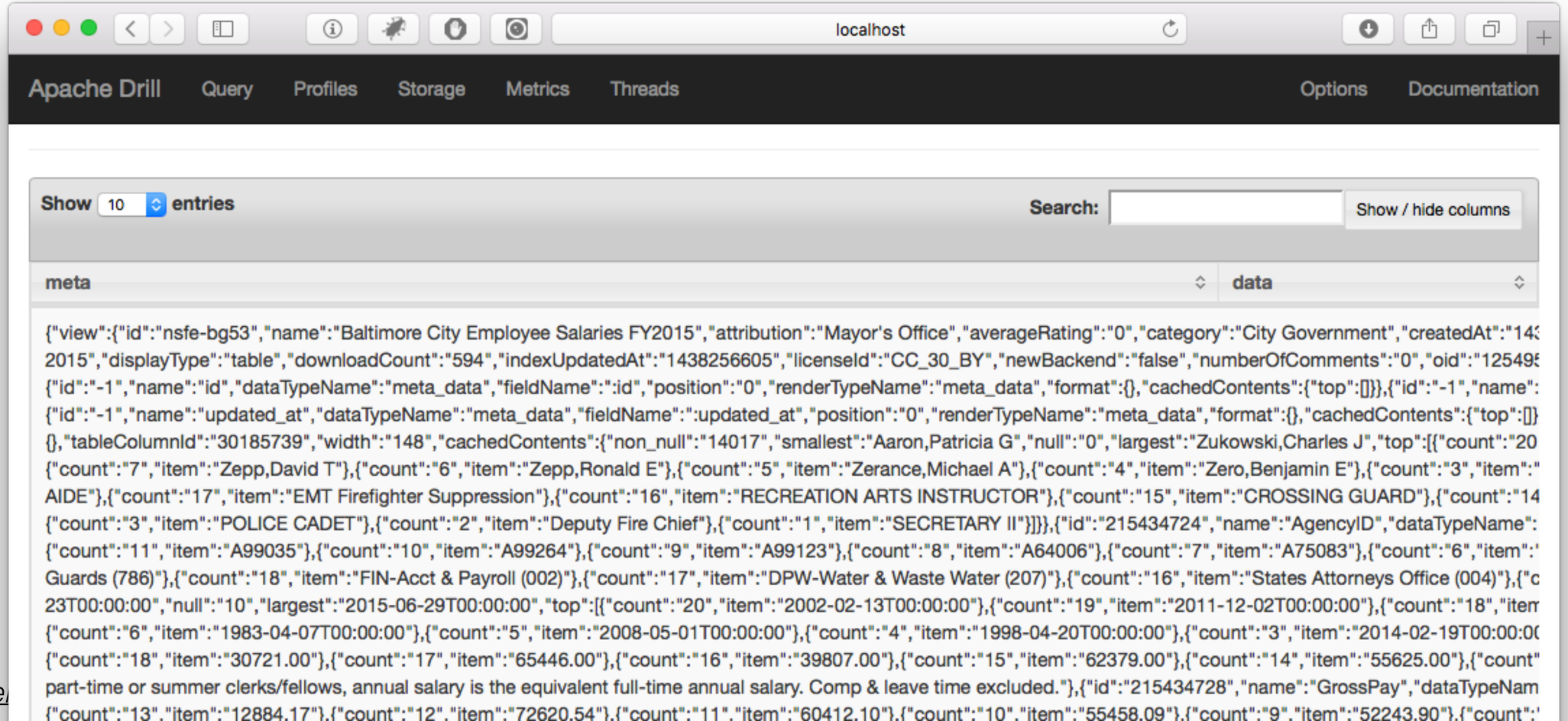


Let’s look at this data in Drill

```
SELECT *
FROM dfs.drillclass.`baltimore_salaries.json`
```

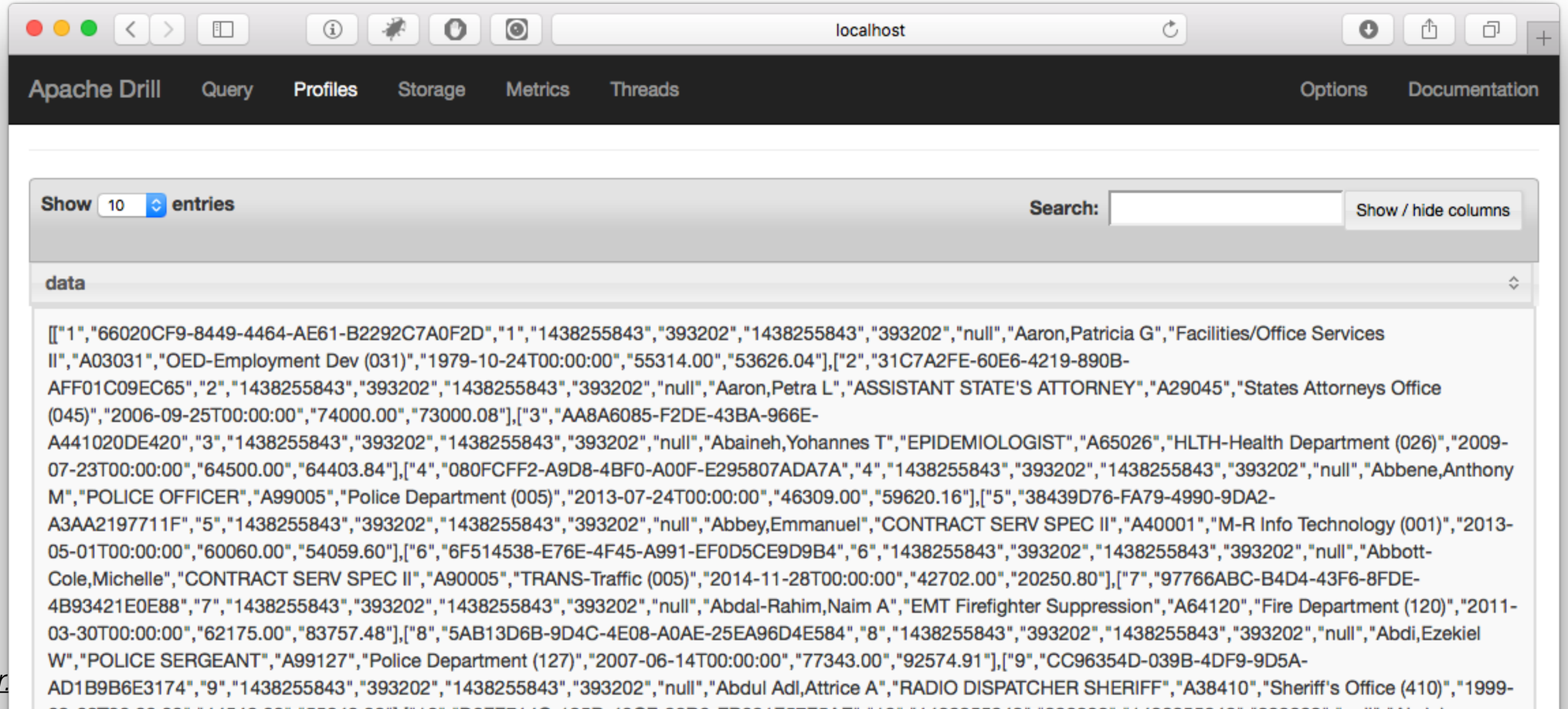
Let's look at this data in Drill

```
SELECT *
FROM dfs.drillclass.`baltimore_salaries.json`
```



Let's look at this data in Drill

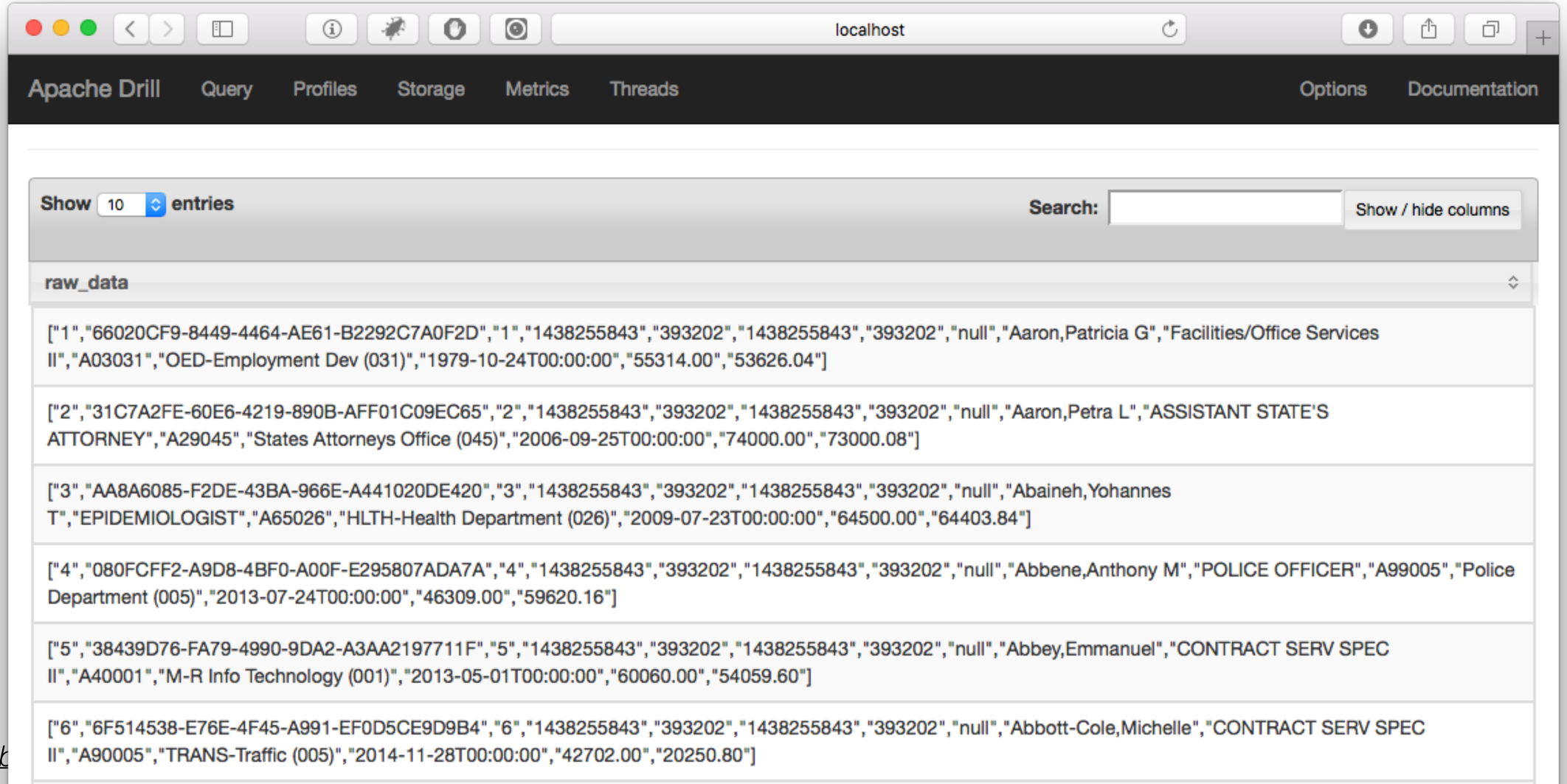
```
SELECT data
FROM dfs.drillclass.`baltimore_salaries.json`
```



`FLATTEN( <json array> )`
separates elements in a repeated field into individual records.

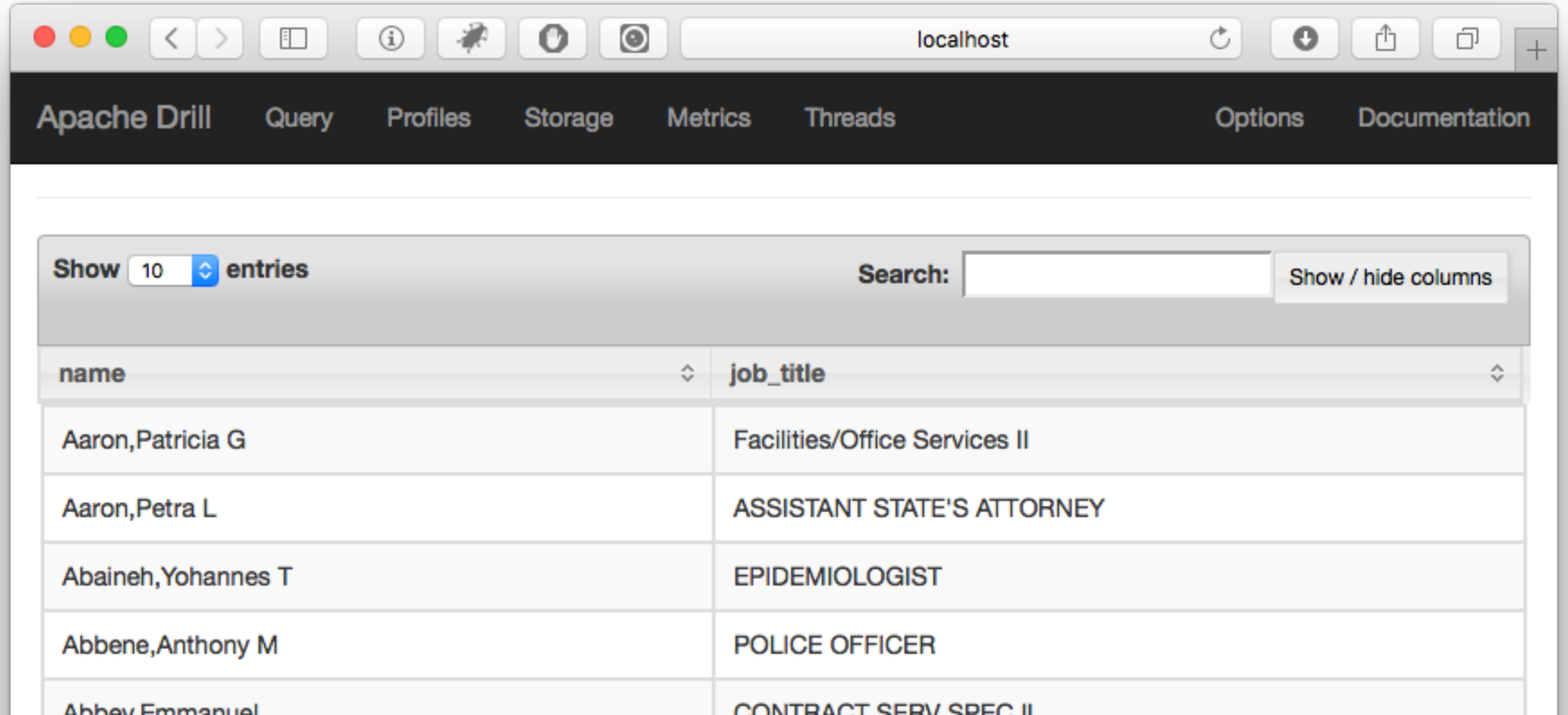

```
SELECT FLATTEN( data ) AS raw_data
FROM dfs.drillclass.`baltimore_salaries.json`
```

```
SELECT FLATTEN( data ) AS raw_data
FROM dfs.drillclass.`baltimore_salaries.json`
```




```
SELECT raw_data[8] AS name
FROM
(
SELECT FLATTEN( data ) AS raw_data
FROM dfs.drillclass.`baltimore_salaries_2015.json`
)
```

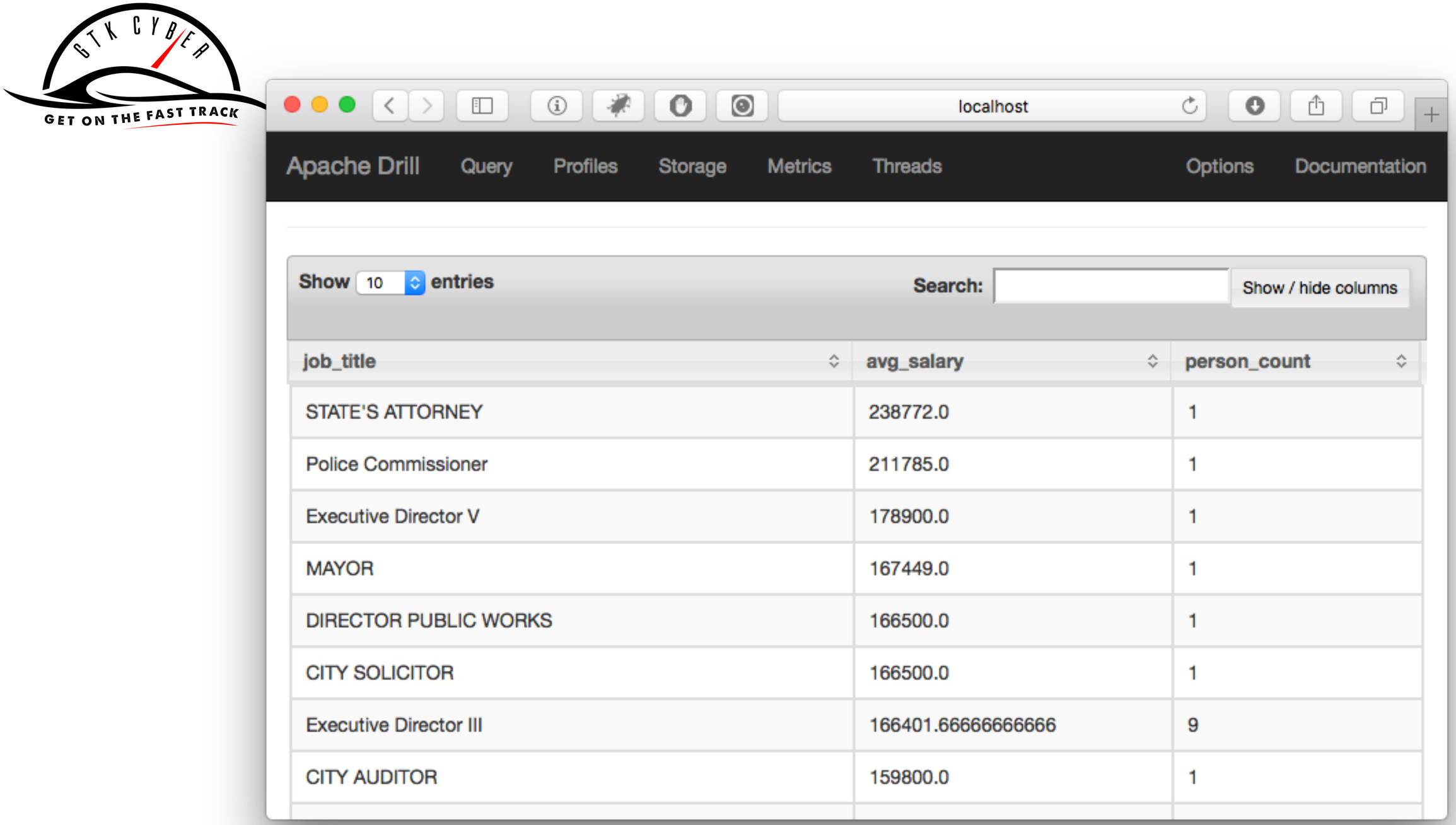
```
SELECT raw_data[8] AS name, raw_data[9] AS job_title
FROM
(
SELECT FLATTEN( data ) AS raw_data
FROM dfs.drillclass.`baltimore_salaries_2015.json`
)
```



In Class Exercise

Using the JSON file, find the average salary by job title and how many people have each job title.

```
SELECT raw_data[9] AS job_title,
AVG( CAST( raw_data[13] AS DOUBLE ) ) AS avg_salary,
COUNT( DISTINCT raw_data[8] ) AS person_count
FROM
(
   SELECT FLATTEN( data ) AS raw_data
   FROM dfs.drillworkshop.`json/baltimore_salaries.json`
)
GROUP BY raw_data[9]
ORDER BY avg_salary DESC
```


localhost
Apache Drill
Query
Profiles
Storage
Metrics
Threads
Options
Documentation
Show 10 entries
Search:
Show / hide columns
job_title
avg_salary
person_count
STATE'S ATTORNEY 238772.0 1
Police Commissioner 211785.0 1
Executive Director V 178900.0 1
MAYOR 167449.0 1
DIRECTOR PUBLIC WORKS 166500.0 1
CITY SOLICITOR 166500.0 1
Executive Director III 166401.66666666666 9
CITY AUDITOR 159800.0 1

Log Files

Log Files

```
070823 21:00:32         1 Connect     root@localhost on test1
070823 21:00:48         1 Query       show tables
070823 21:00:56         1 Query       select * from category
070917 16:29:01        21 Query       select * from location
070917 16:29:12        21 Query       select * from location where id = 1 LIMIT 1
```


```
"log" : {
      "type" : "logRegex",
      "extension" : "log",
      "regex" : "(\\d{6})\\s(\\d{2}:\\d{2}:\\d{2})\\s+(\\d+)\\s(\\w+)\\s+(.+)",
      "maxErrors": 10,
      "schema": [
        {
          "fieldName": "eventDate",
          "fieldType": "DATE",
          "format": "yyMMdd"
        },
        {
          "fieldName": "eventTime",
          "fieldType": "TIME",
          "format": "HH:mm:ss"
        },
        {
          "fieldName": "PID",
          "fieldType": "INT"
        },
        {
          "fieldName": "action"
        },
        {
          "fieldName": "query"
        }
      ]
    }
```


```
SELECT *
FROM dfs.drillworkshop.`log_files/mysql.log`
```


```
SELECT *
FROM dfs.drillworkshop.`log_files/mysql.log`
```

Show 10 entries Search: Show / hide columns

date	time	pid	action	query
070823	21:00:32	1	Connect	root@localhost on test1
070823	21:00:48	1	Query	show tables
070823	21:00:56	1	Query	select * from category
070917	16:29:01	21	Query	select * from location
070917	16:29:12	21	Query	select * from location where id = 1 LIMIT 1

Showing 1 to 5 of 5 entries Previous 1 Next

HTTPD Log Files

HTTPD Log Files

195.154.46.135 - - [25/Oct/2015:04:11:25 +0100] "GET /linux/doing-pxe-without-dhcp-control HTTP/1.1" 200 24323 "http://howto.basjes.nl/" "Mozilla/5.0 (Windows NT 5.1; rv:35.0) Gecko/20100101 Firefox/35.0"
23.95.237.180 - - [25/Oct/2015:04:11:26 +0100] "GET /join_form HTTP/1.0" 200 11114 "http://howto.basjes.nl/" "Mozilla/5.0 (Windows NT 5.1; rv:35.0) Gecko/20100101 Firefox/35.0"
23.95.237.180 - - [25/Oct/2015:04:11:27 +0100] "POST /join_form HTTP/1.1" 302 9093 "http://howto.basjes.nl/join_form" "Mozilla/5.0 (Windows NT 5.1; rv:35.0) Gecko/20100101 Firefox/35.0"
158.222.5.157 - - [25/Oct/2015:04:24:31 +0100] "GET /join_form HTTP/1.0" 200 11114 "http://howto.basjes.nl/" "Mozilla/5.0 (Windows NT 6.3; WOW64; rv:34.0) Gecko/20100101 Firefox/34.0 AlexaToolbar/alxf-2.21"
158.222.5.157 - - [25/Oct/2015:04:24:32 +0100] "POST /join_form HTTP/1.1" 302 9093 "http://howto.basjes.nl/join_form" "Mozilla/5.0 (Windows NT 6.3; WOW64; rv:34.0) Gecko/20100101 Firefox/34.0 AlexaToolbar/alxf-2.21"

HTTPD Log Files

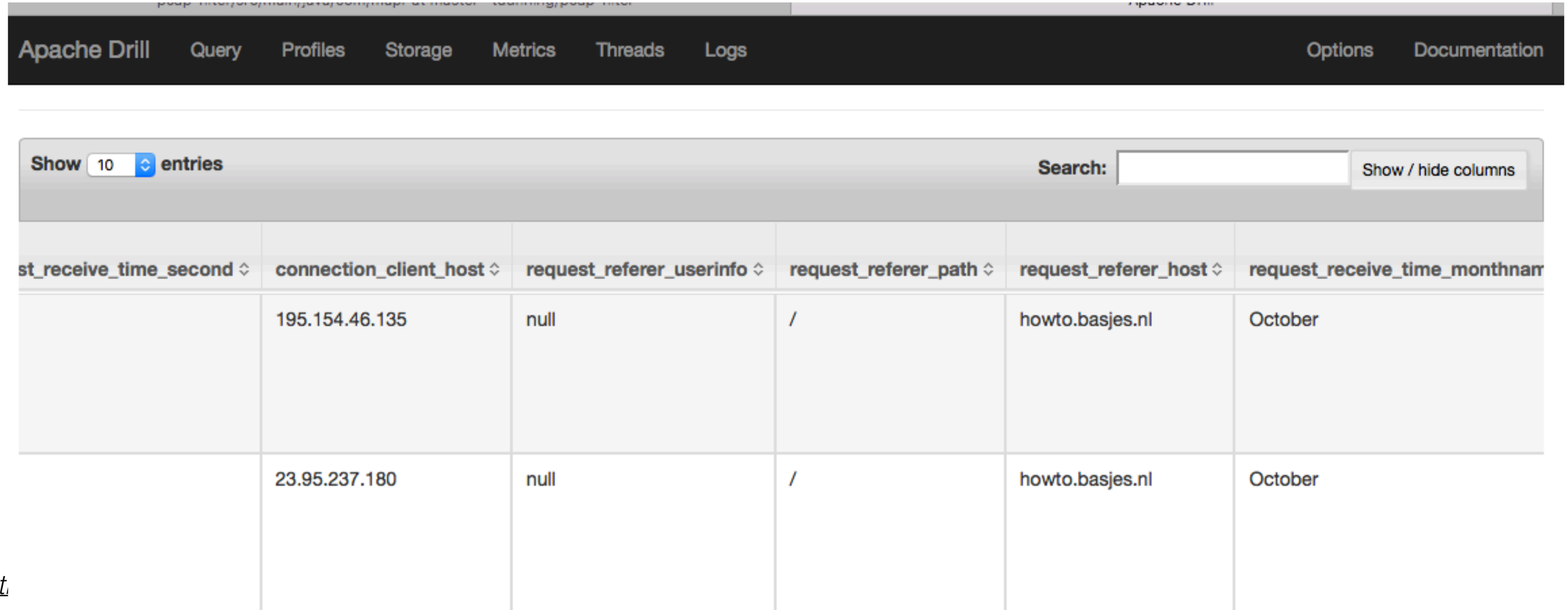
195.154.46.135 - - [25/Oct/2015:04:11:25 +0100] "GET /linux/doing-pxe-without-dhcp-control HTTP/1.1" 200 24323 "http://howto.basjes.nl/" "Mozilla/5.0 (Windows NT 5.1; rv:35.0) Gecko/20100101 Firefox/35.0"
23.95.237.180 - - [25/Oct/2015:04:11:26 +0100] "GET /join_form HTTP/1.0" 200 11114 "http://howto.basjes.nl/" "Mozilla/5.0 (Windows NT 5.1; rv:35.0) Gecko/20100101 Firefox/35.0"
23.95.237.180 - - [25/Oct/2015:04:11:27 +0100] "POST /join_form HTTP/1.1" 302 9093 "http://howto.basjes.nl/join_form" "Mozilla/5.0 (Windows NT 5.1; rv:35.0) Gecko/20100101 Firefox/35.0"
158.222.5.157 - - [25/Oct/2015:04:24:31 +0100] "GET /join_form HTTP/1.0" 200 11114 "http://howto.basjes.nl/" "Mozilla/5.0 (Windows NT 6.3; WOW64; rv:34.0) Gecko/20100101 Firefox/34.0 AlexaToolbar/alxf-2.21"
158.222.5.157 - - [25/Oct/2015:04:24:32 +0100] "POST /join_form HTTP/1.1" 302 9093 "http://howto.basjes.nl/join_form" "Mozilla/5.0 (Windows NT 6.3; WOW64; rv:34.0) Gecko/20100101 Firefox/34.0 AlexaToolbar/alxf-2.21"

```
"httpd": {
      "type": "httpd",
      "logFormat": "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-agent}i\"",
      "timestampFormat": null
    },
```

HTTPD Log Files

```
SELECT *
FROM dfs.drillworkshop.`data_files/log_files/small-server-
log.httpd`
```

```
SELECT *
FROM dfs.drillworkshop.`data_files/log_files/small-server-
log.httpd`
```



HTTPD Log Files

```
SELECT request_referer, parse_url( request_referer ) AS url_data
FROM dfs.drillworkshop.`data_files/log_files/small-server-log.httpd`
```

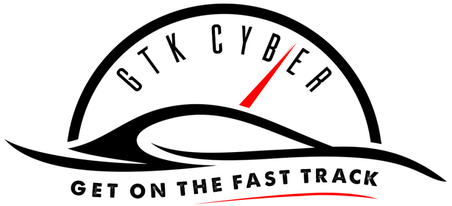
HTTPD Log Files

```
SELECT request_referer, parse_url( request_referer ) AS url_data
FROM dfs.drillworkshop.`data_files/log_files/small-server-log.httpd`
```

Show 10 entries Search:

request_referer	url_data
http://howto.basjes.nl/	{"protocol":"http","authority":"howto.basjes.nl","host":"howto.basjes.nl","path":"/"}
http://howto.basjes.nl/	{"protocol":"http","authority":"howto.basjes.nl","host":"howto.basjes.nl","path":"/"}
http://howto.basjes.nl/join_form	{"protocol":"http","authority":"howto.basjes.nl","host":"howto.basjes.nl","path":"/join_form"}
http://howto.basjes.nl/	{"protocol":"http","authority":"howto.basjes.nl","host":"howto.basjes.nl","path":"/"}
http://howto.basjes.nl/join_form	{"protocol":"http","authority":"howto.basjes.nl","host":"howto.basjes.nl","path":"/join_form"}

Networking Functions



Networking Functions

- inet_aton(<ip>): Converts an IPv4 Address to an integer
- inet_ntoa(<int>): Converts an integer to an IPv4 address
- is_private(<ip>): Returns true if the IP is private
- in_network(<ip>,<cidr>): Returns true if the IP is in the CIDR block
- getAddressCount(<cidr>): Returns the number of IPs in a CIDR block
- getBroadcastAddress(<cidr>): Returns the broadcast address of a CIDR block
- getNetmask(<cidr>): Returns the net mask of a CIDR block
- getLowAddress(<cidr>): Returns the low IP of a CIDR block
- getHighAddress(<cidr>): Returns the high IP of a CIDR block
- parse_user_agent(<ua_string>): Returns a map of user agent information

PCAP Files


```
SELECT *
FROM dfs.test.`dns-zone-transfer-ixfr.pcap`
```

localhost

Time Series Analysis #1: Introduction to Window Functions | Pivotal | Data Summit - Call For Speakers | Analyzing Real-World Data with Apache Drill | Apache Drill

Apache Drill | Query | Profiles | Storage | Metrics | Threads | Logs | Options | Documentation

Show 10 entries | Search: | Show / hide columns

ipVersion	Frame_Number	Protocol	MACAddressSource	MACAddressDestination	PacketLength	Timestamp	Source_IP	Destination_IP	IP_Protocol	IP_Protocol_Name	IP_Protocol_Desc	sourcePort	destinationPort
4	1	UDP	08:00:27:38:DB:ED	08:00:27:97:3F:45	129	2015-03-26T11:27:18.000-04:00	1.1.1.2	1.1.1.1	17	UDP	User Datagram Protocol	1028	53
4	2	UDP	08:00:27:97:3F:45	08:00:27:38:DB:ED	257	2015-03-26T11:27:18.000-04:00	1.1.1.1	1.1.1.2	17	UDP	User Datagram Protocol	53	1028

Showing 1 to 2 of 2 entries | Previous 1 Next

Connecting other Data Sources


GTK CYBER
GET ON THE FAST TRACK


APACHE
HBASE


HIVE

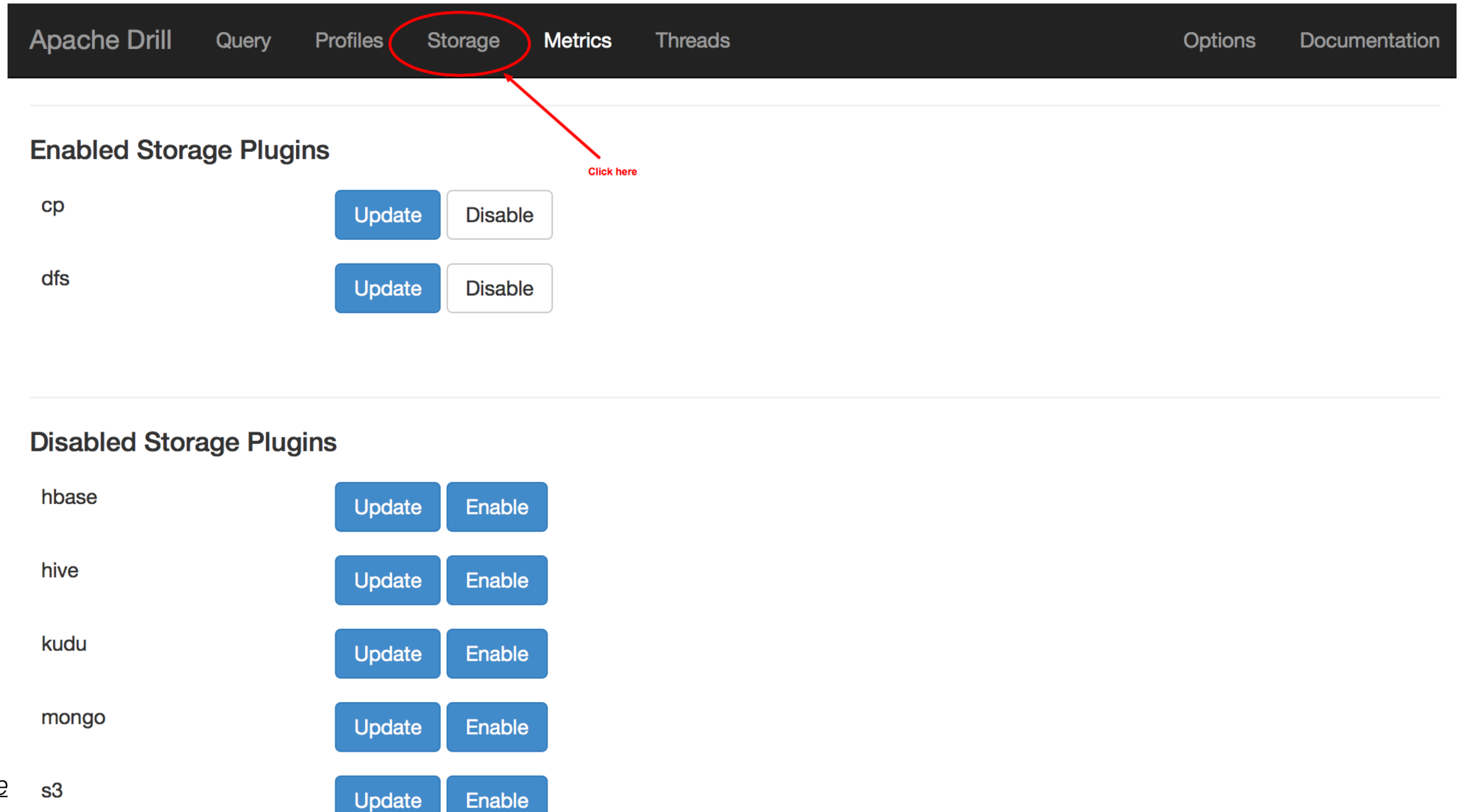

amazon
web services™


mongoDB®

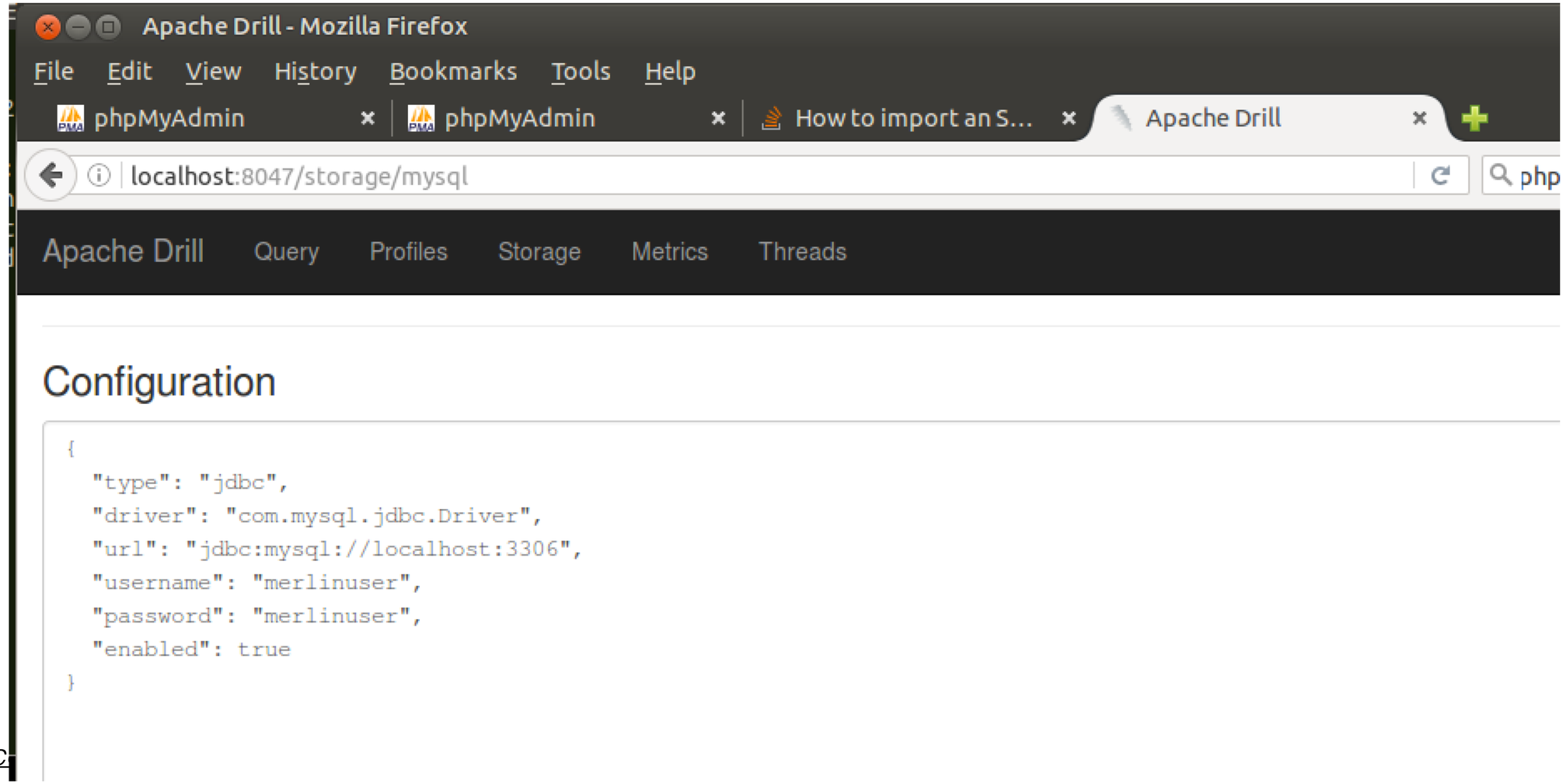

MySQL®


hadoop

Connecting other Data Sources

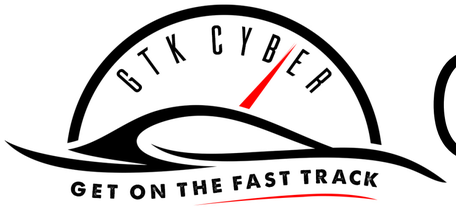


Connecting other Data Sources




GTK CYBER
GET ON THE FAST TRACK

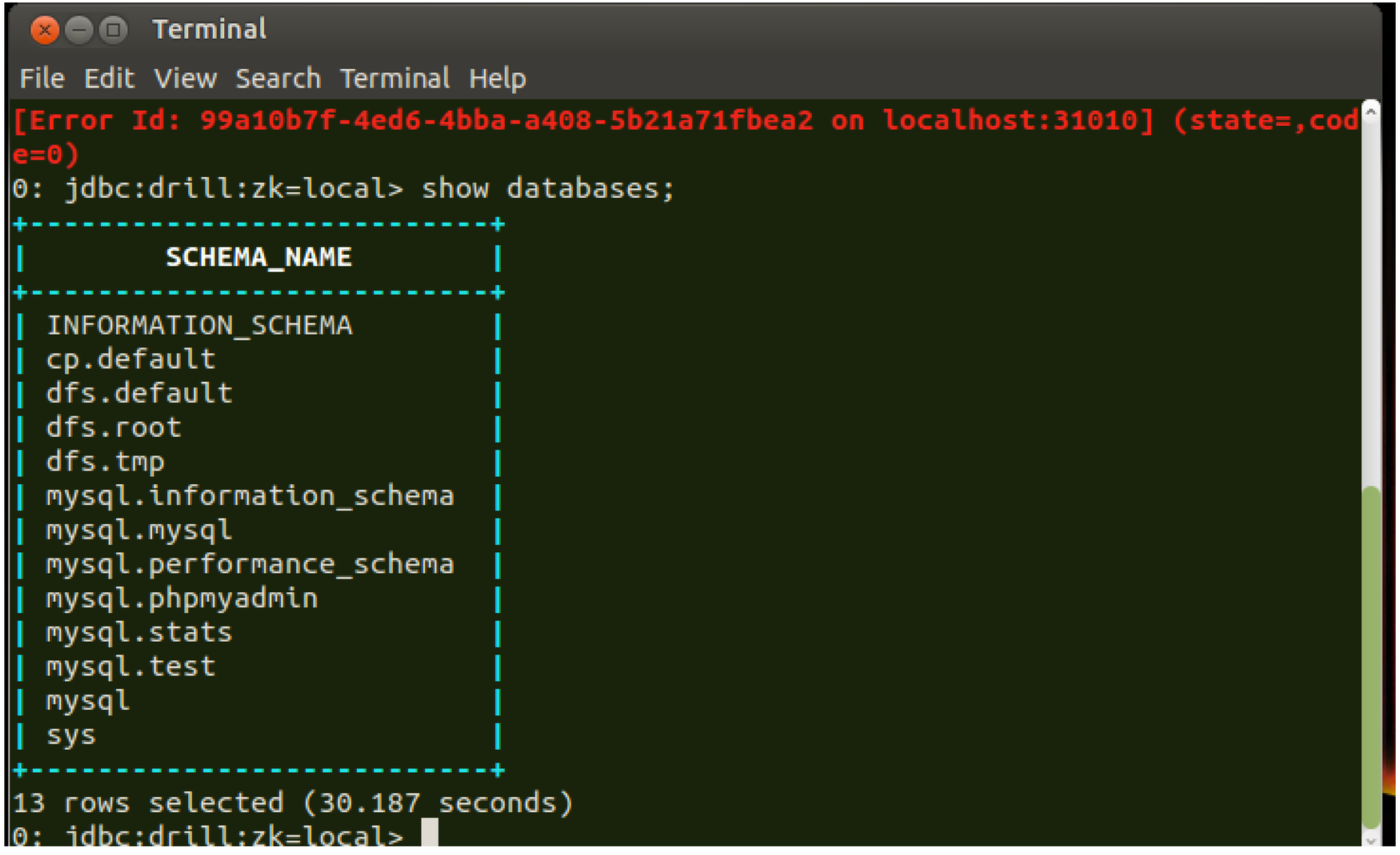

MySQL™



Connecting other Data Sources

```
{
 "type": "jdbc",
 "driver": "com.mysql.cj.jdbc.Driver",
 "url": "jdbc:mysql://localhost:3306",
 "username": "griffonuser",
 "password": "griffonuser",
 "caseInsensitiveTableNames": false,
 "enabled": true
}
```

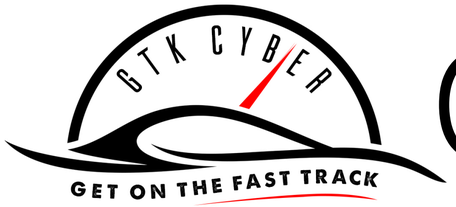
Connecting other Data Sources



Connecting other Data Sources

```
SELECT teams.name, SUM( batting.HR ) as hr_total
FROM batting
INNER JOIN teams ON batting.teamID=teams.teamID
WHERE batting.yearID = 1988 AND teams.yearID = 1988
GROUP BY batting.teamID
ORDER BY hr_total DESC
```

MySQL: 0.047 seconds



Connecting other Data Sources

```
SELECT teams.name, SUM( batting.HR ) as hr_total
FROM mysql.stats.batting
INNER JOIN mysql.stats.teams ON batting.teamID=teams.teamI
WHERE batting.yearID = 1988 AND teams.yearID = 1988
GROUP BY teams.name
ORDER BY hr_total DESC
```

MySQL: 0.047 seconds

Drill: 0.366 seconds


hadoop

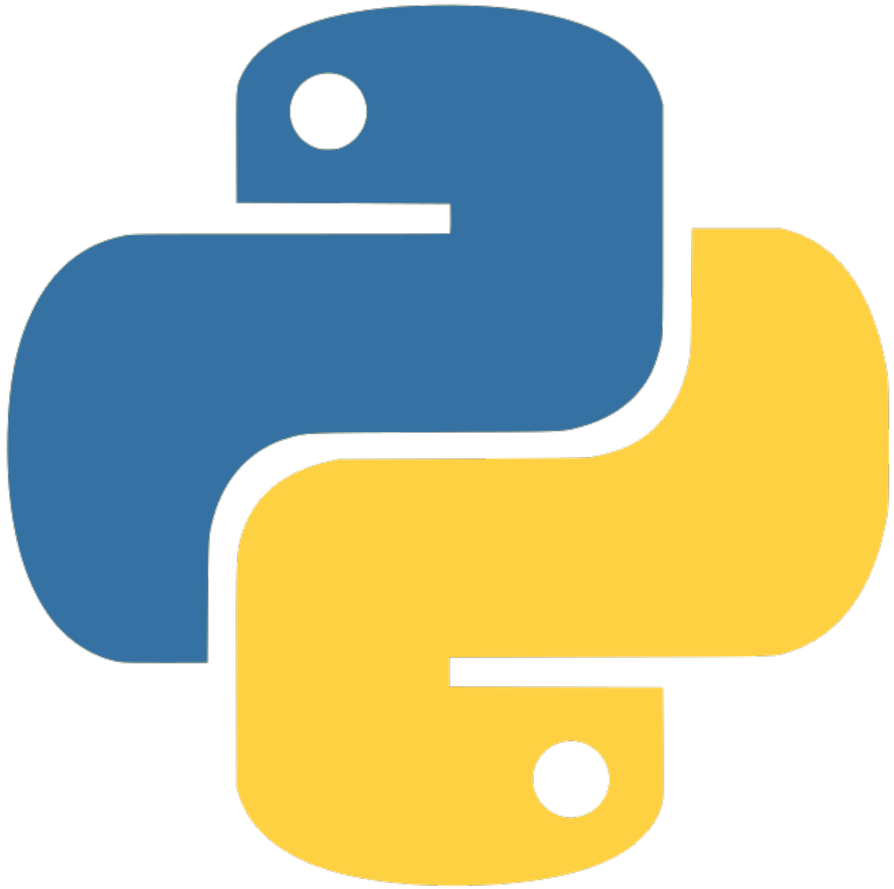
Just like DFS, except you specify a link to the Hadoop namenode.

```
{
  "type": "file",
  "enabled": true,
  "connection": "hdfs://localhost:54310",
  "config": null,
  "workspaces": {
      "demodata": {
      "location": "/user/merlinuser/demo",
      "writable": true,
      "defaultInputFormat": null
    }
  },
```

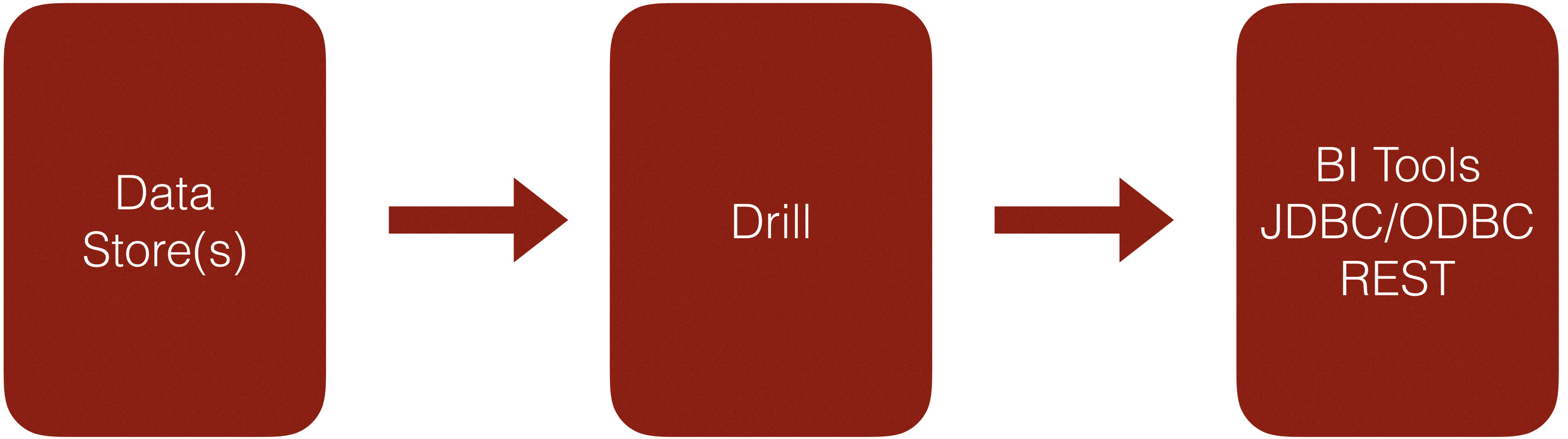

```
SELECT name, SUM( CAST( HR AS INT)) AS HR_Total
FROM hdfs.demodata.`Teams.csvh`
WHERE yearID=1988
GROUP BY name
ORDER BY HR_Total DESC
```

Connecting to Drill

Python



Connecting to Drill

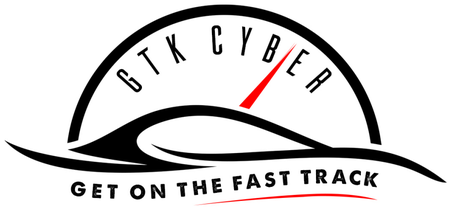


Connecting to Drill

```
pip install pydrill
```

Connecting to Drill

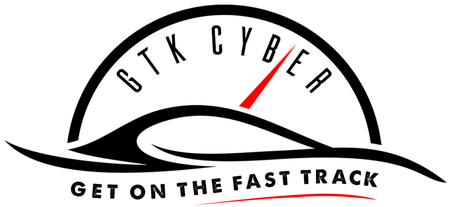
```
from pydrill.client import PyDrill
```



Connecting to Drill

```
drill = PyDrill(host='localhost', port=8047)

if not drill.is_active():
    raise ImproperlyConfigured('Please run Drill first')
```



Connecting to Drill

```
query_result = drill.query('''
 SELECT JobTitle,
  AVG( CAST( LTRIM( AnnualSalary, '$' ) AS FLOAT) ) AS
avg_salary,
COUNT( DISTINCT name ) AS number
FROM dfs.drillclass.`*.csvh`
GROUP BY JobTitle
Order By avg_salary DESC
LIMIT 10
''')
```

Connecting to Drill

```
df = query_result.to_dataframe()
```

+

Apache Zeppelin

Connecting Drill and Zeppelin

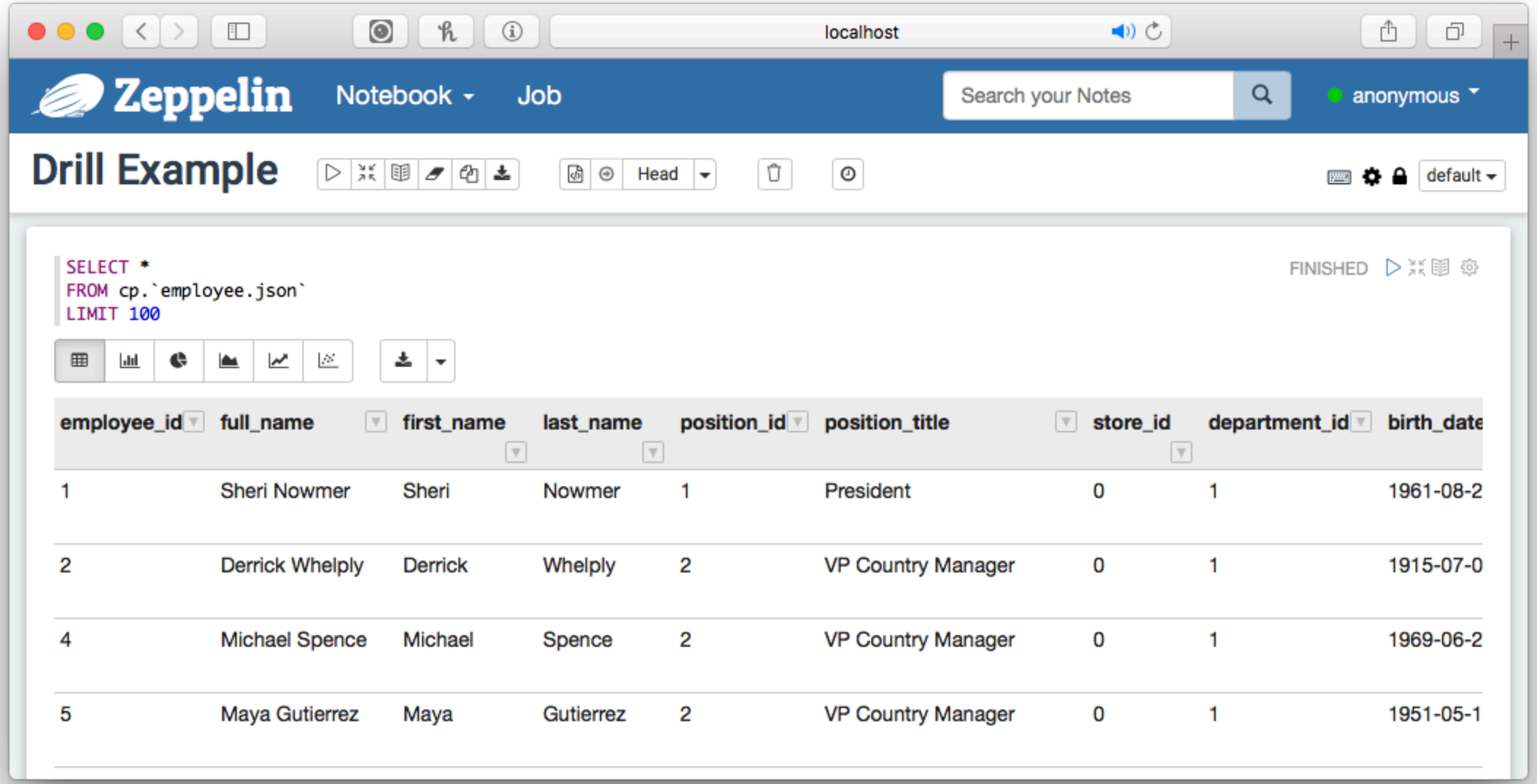
- Navigate to the interpreter menu and create a new JDBC interpreter

In the `Create` screen, choose `JDBC` as the interface type, and set the following configuration variables as shown in figure 7.3.

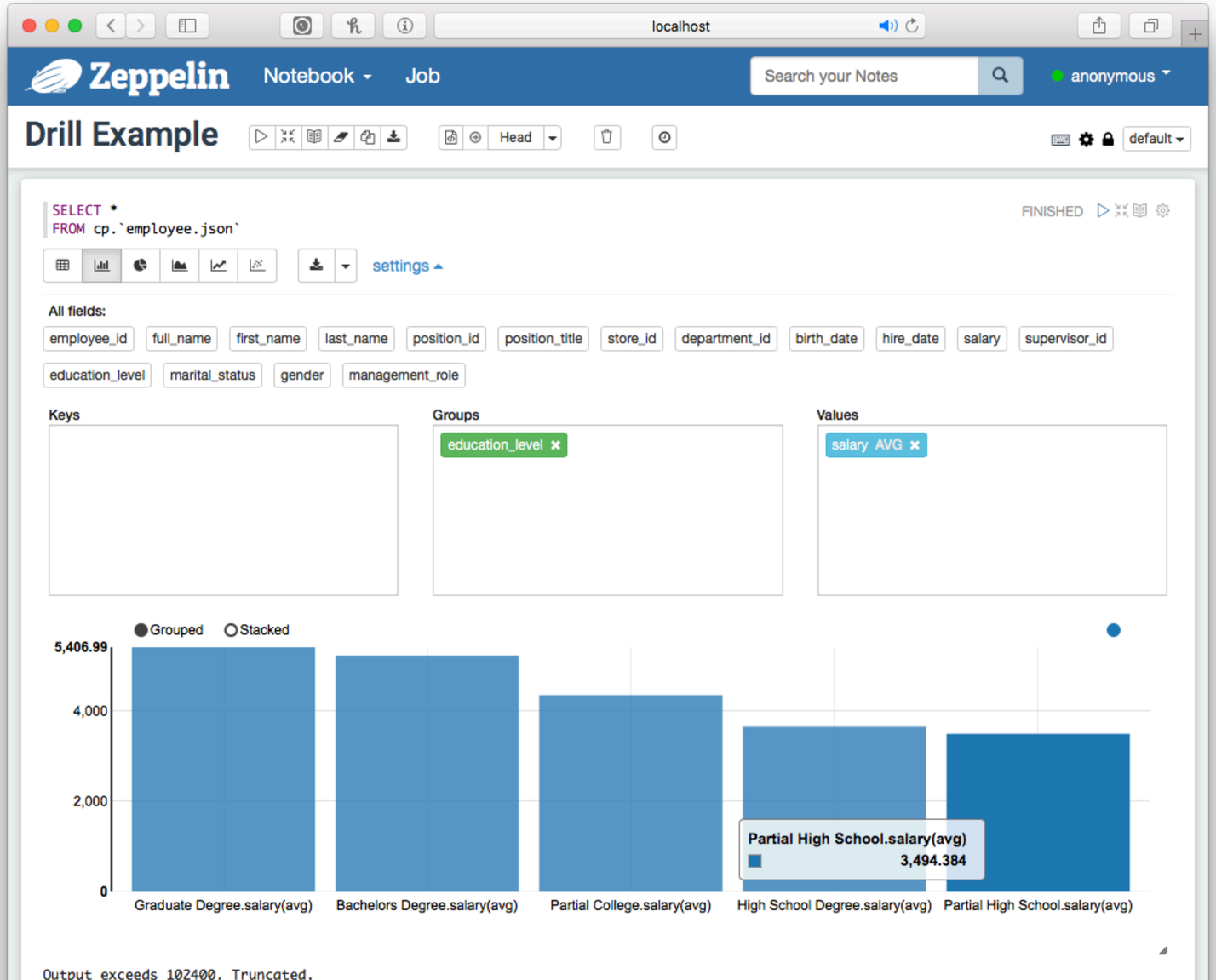
- `default.driver`: This should be set to `org.apache.drill.jdbc.Driver`
- `default.password`: Your account password
- `default.url`: This is your JDBC connection string to Drill. To use Drill with Zeppelin in embedded mode, the connection string is: `jdbc:drill:drillbit=localhost:31010`

At the bottom of the screen in the `Dependencies` section, you will need to add the path to the Drill JDBC driver as an artifact,

Connecting Drill and Zeppelin



Connecting Drill and Zeppelin



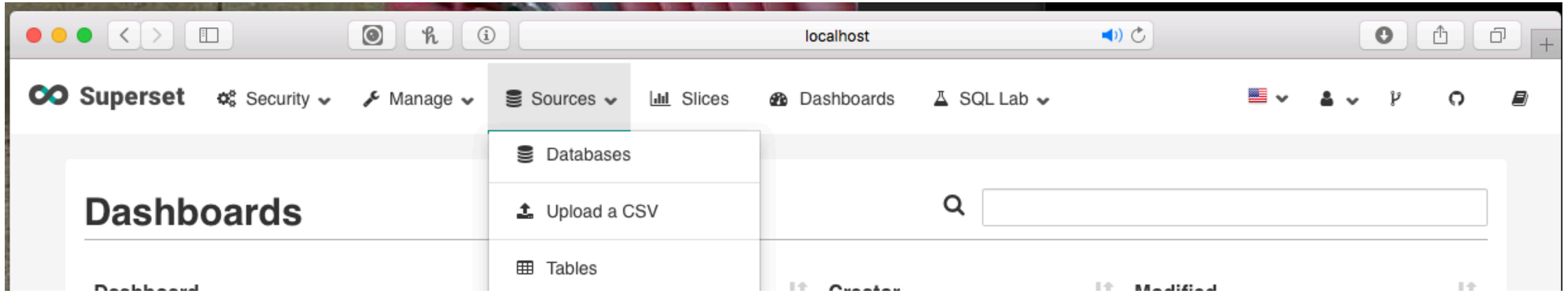
Building Dashboards with Drill and Superset

- Apache Superset is an open source BI tool that makes it easy to create interactive visualizations and dashboards from SQL-speaking data sources.

- Documentation available here: https://superset.incubator.apache.org

Building Dashboards with Drill and Superset

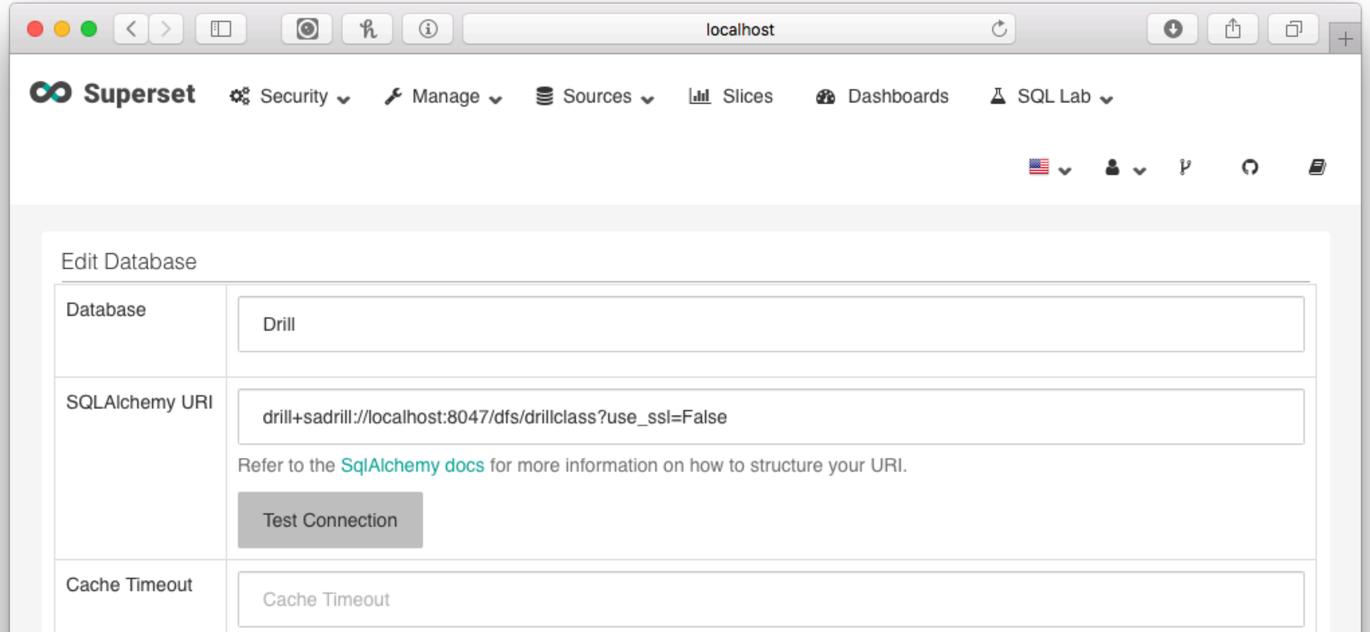
- First, make sure Superset and Drill are running
- Navigate to the Sources - > Databases menu
- Click + to add a new data source



Building Dashboards with Drill and Superset

- Use this string to connect to Drill in embedded mode:

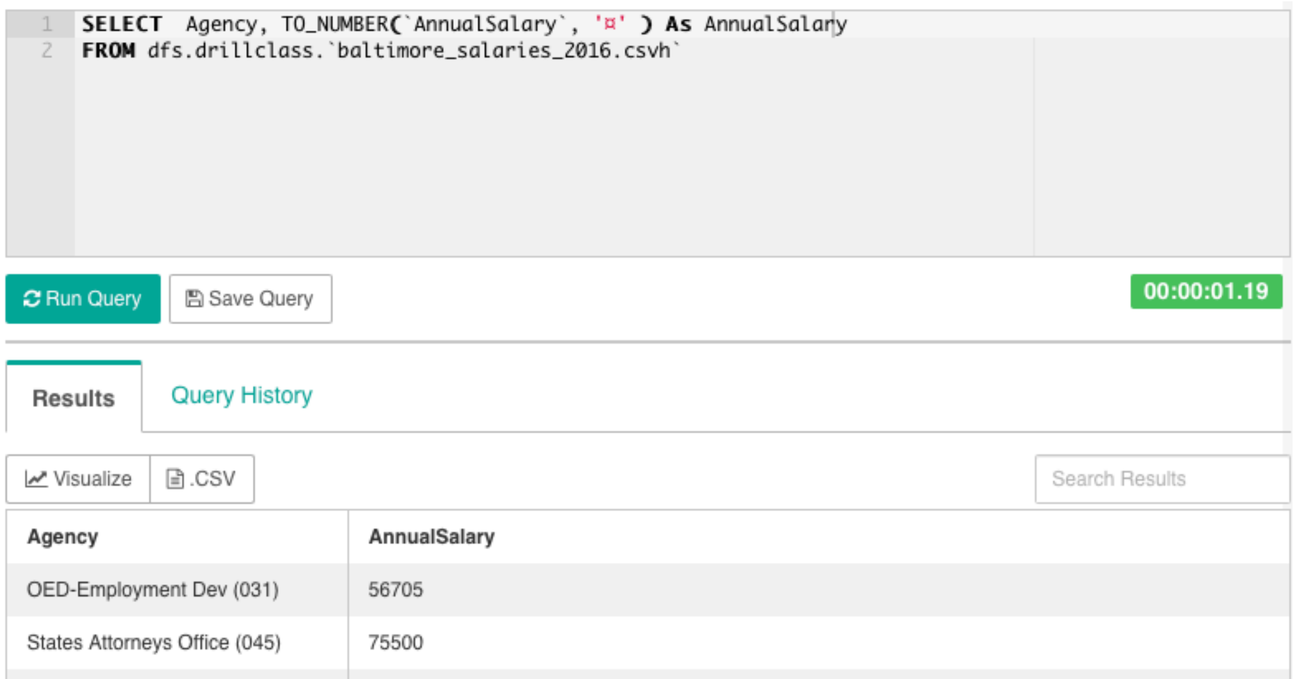
```
drill+sadrill://localhost:8047/dfs/dsclass?use_ssl=False
```



Building Dashboards with Drill and Superset

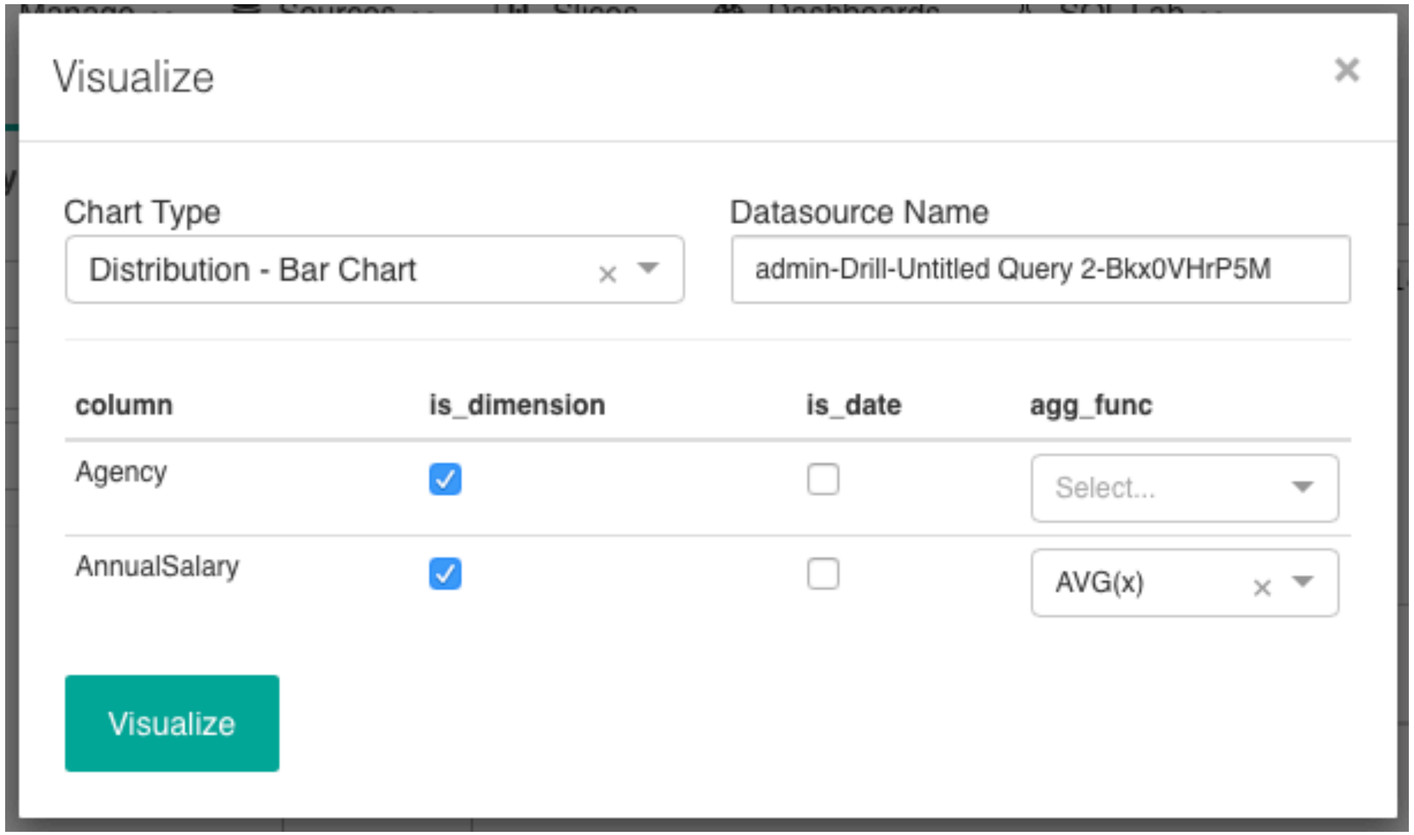
- Next, navigate to SQLLab and enter the following query:

```
SELECT Agency, TO_NUMBER(`AnnualSalary`, '¤' ) As
AnnualSalary
FROM dfs.<workspace>.`baltimore_salaries_2016.csvh`
```



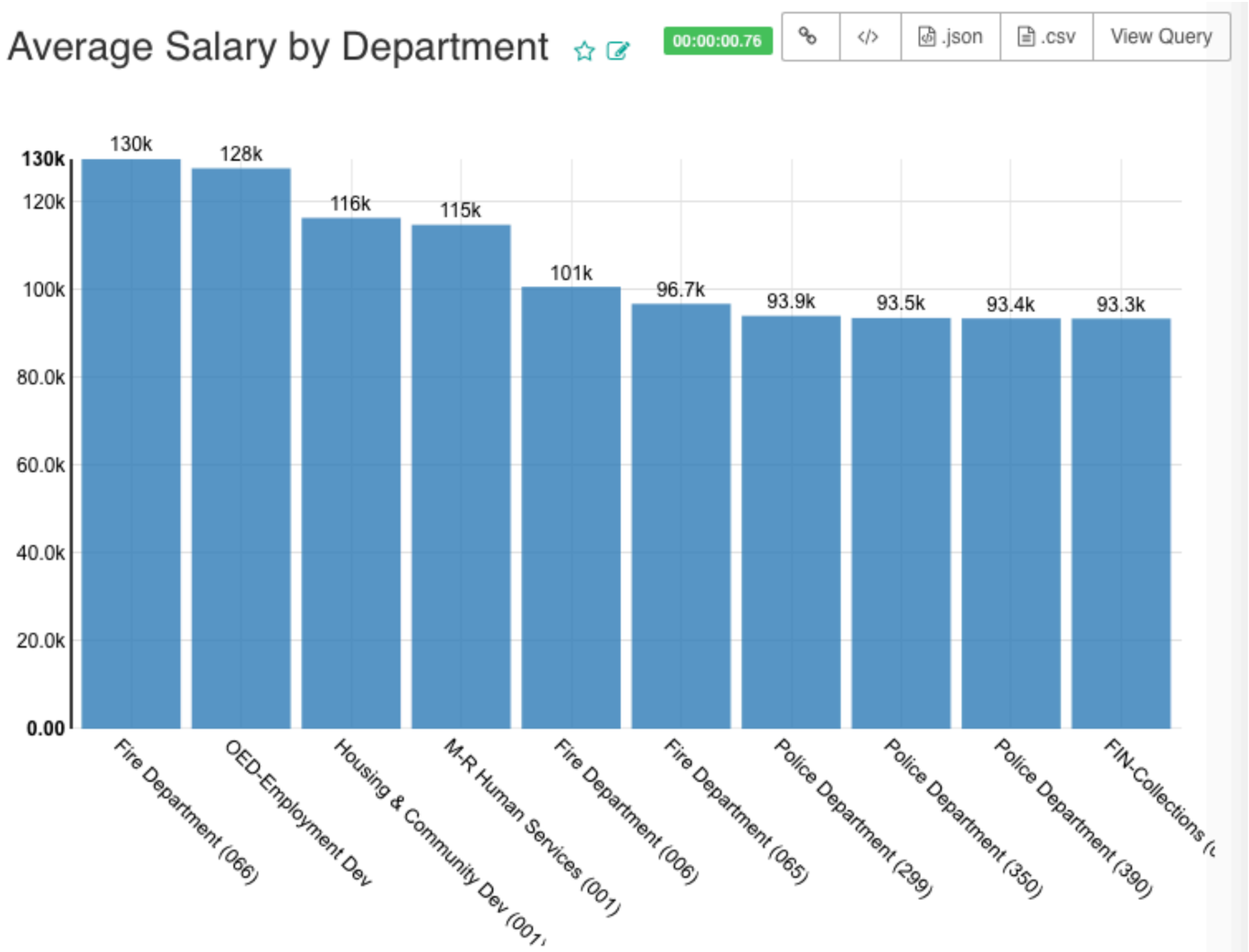
Building Dashboards with Drill and Superset

- Click on Visualize



Building Dashboards with Drill and Superset

- Voila!



Conclusion

- Drill is easy to use
- Drill scales
- Drill is open source
- Drill is versatile

Why aren’t you using Drill?

In Class Exercise

Complete Worksheet 3: Drill Demonstration Worksheet.