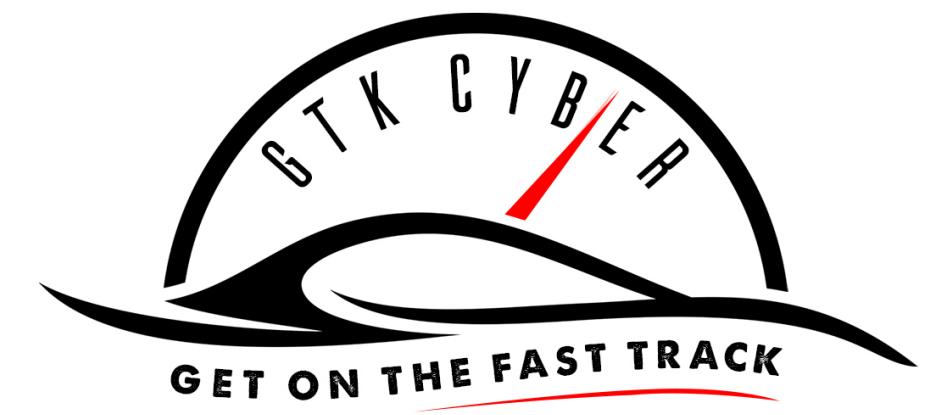


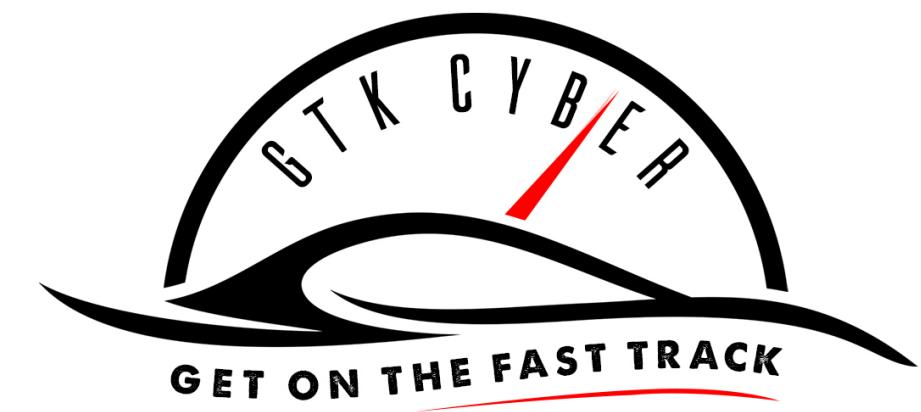
Module 8: Apache Drill

GET ON THE FAST TRACK





A challenge to set the stage

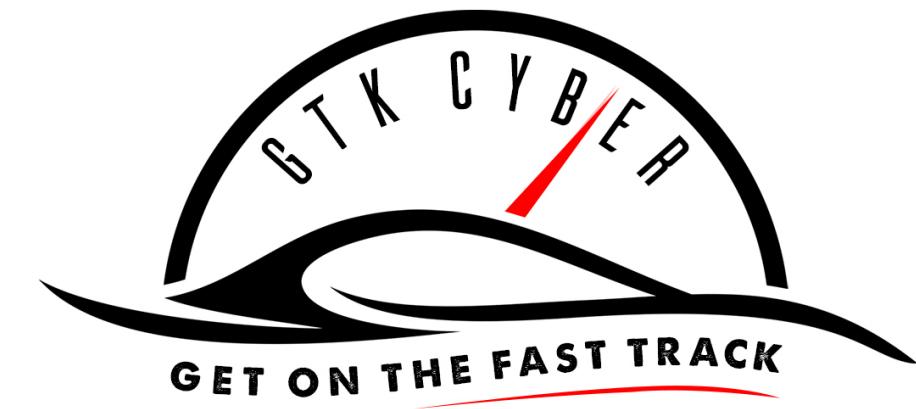


A challenge to set the stage

firewall1.ssdlog

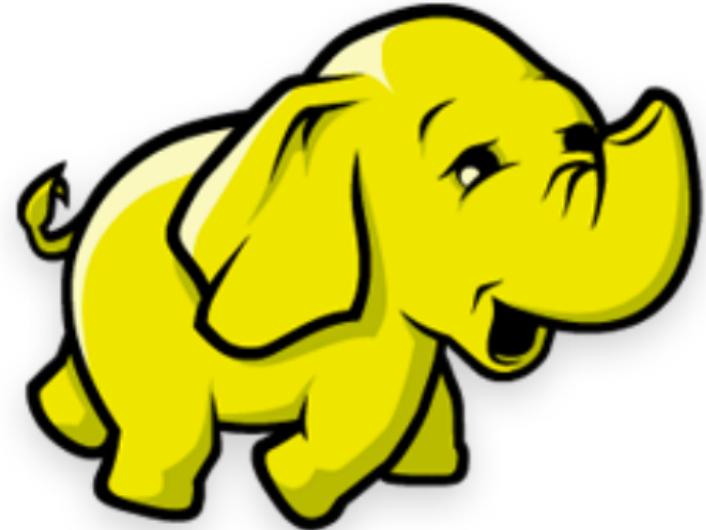
```
Dec 10 22:36:39 sshd[80124]: Failed password for root from 114.80.246.132 port 1093 ssh2
Dec 10 22:36:37 sshd[79627]: Failed password for root from 114.80.246.132 port 3163 ssh2
Dec 10 22:36:35 sshd[79027]: Failed password for root from 114.80.246.132 port 1391 ssh2
Dec 10 22:36:34 sshd[79440]: Failed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

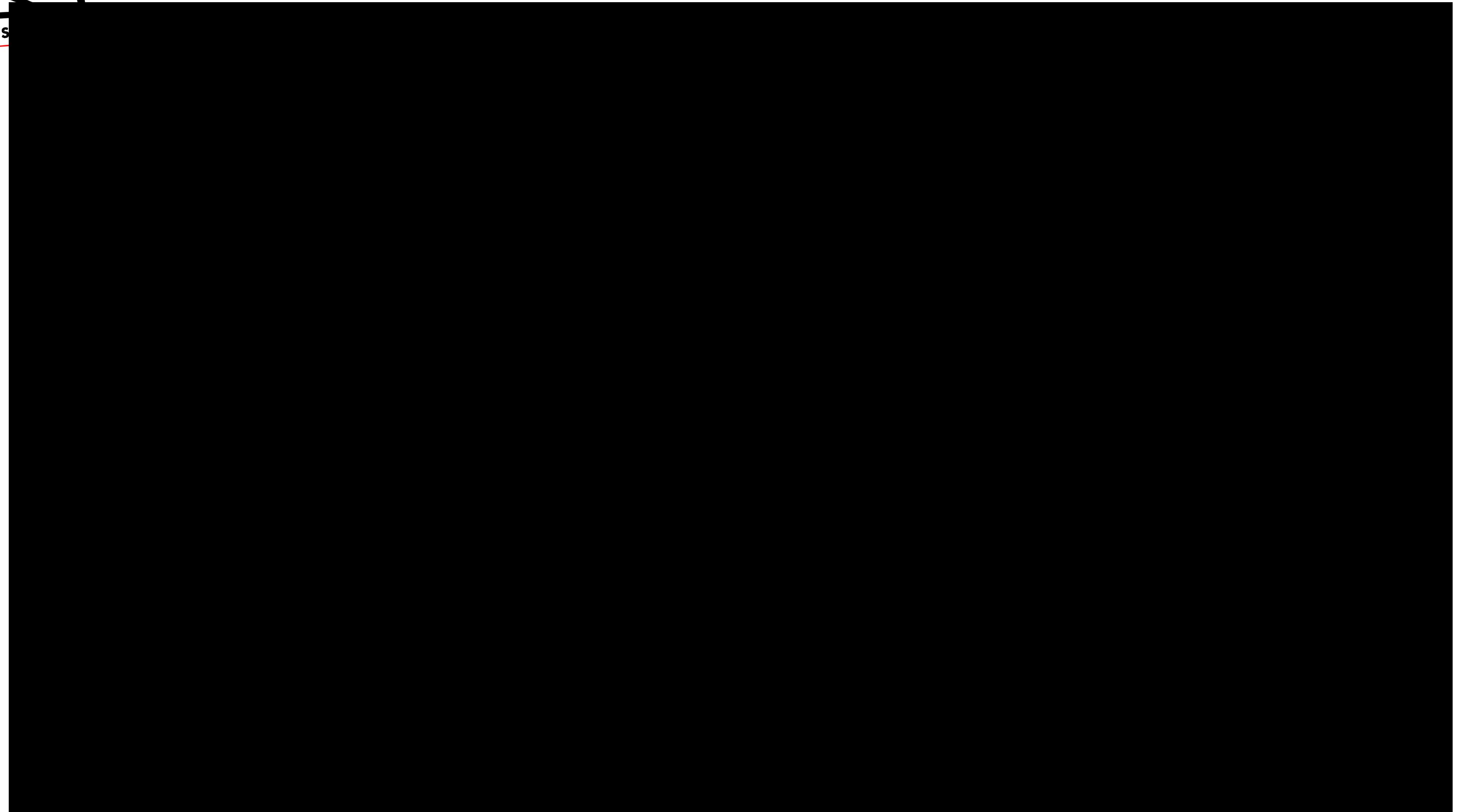


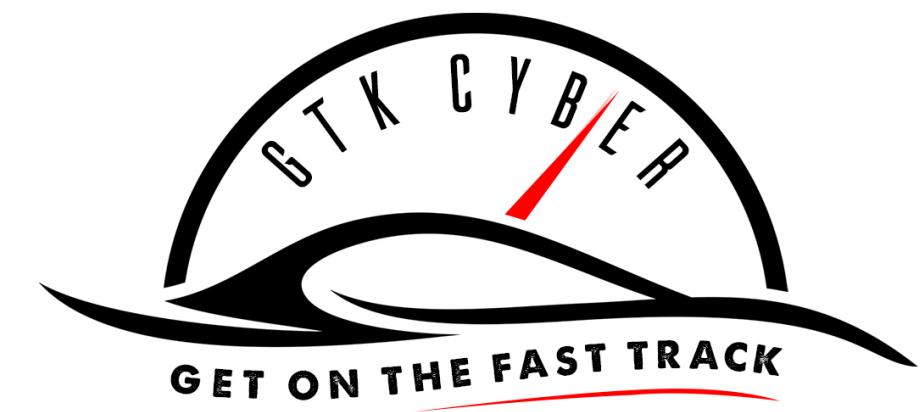
Quick Demo

```
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



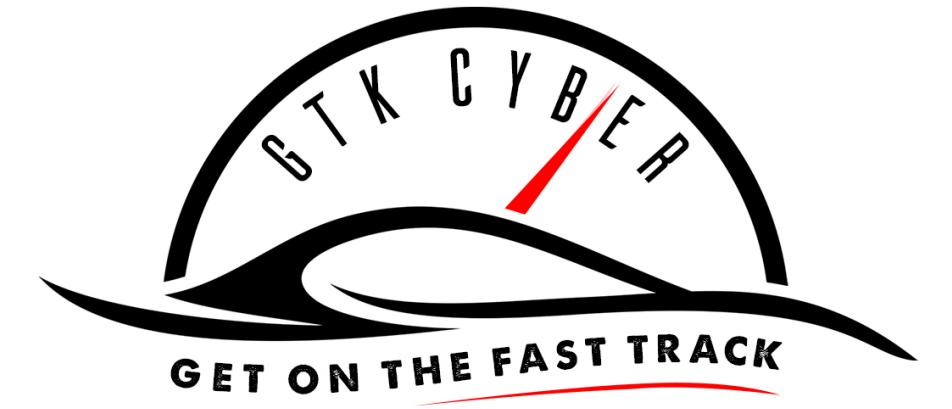


Quick Demo

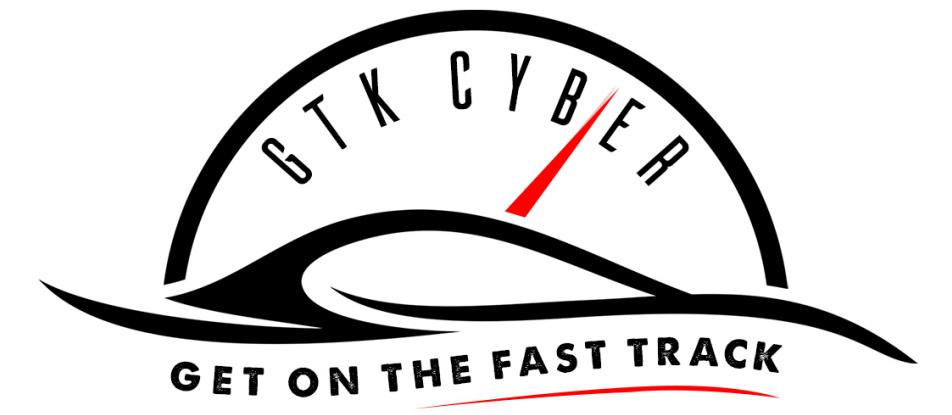
```
SELECT columns[40], cast(columns[19] as int) AS HR  
FROM `baseball_csv/Teams.csv`  
WHERE columns[0] = '1988'  
ORDER BY HR desc;
```



Execution Time:
0.232 seconds!!

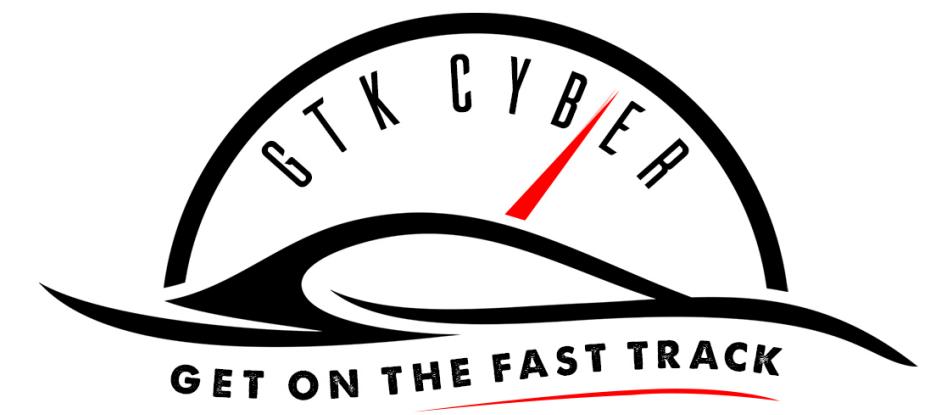


The Problem: Analyzing Security Data is Hard

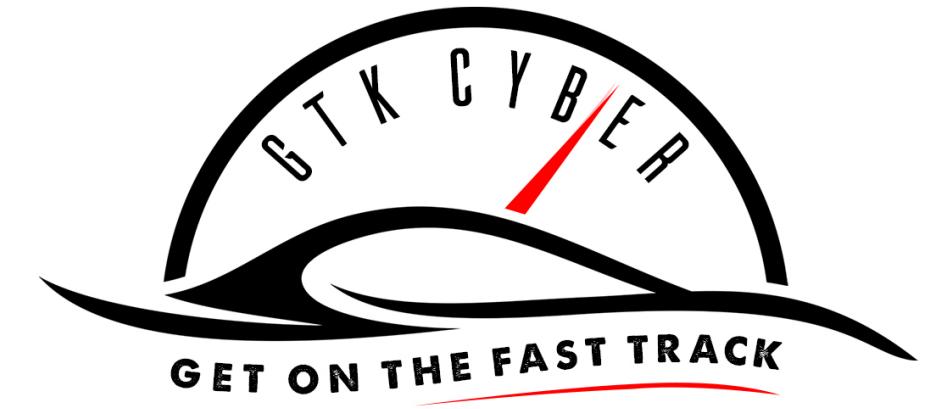


Security Data Analysis is Hard...Really Hard

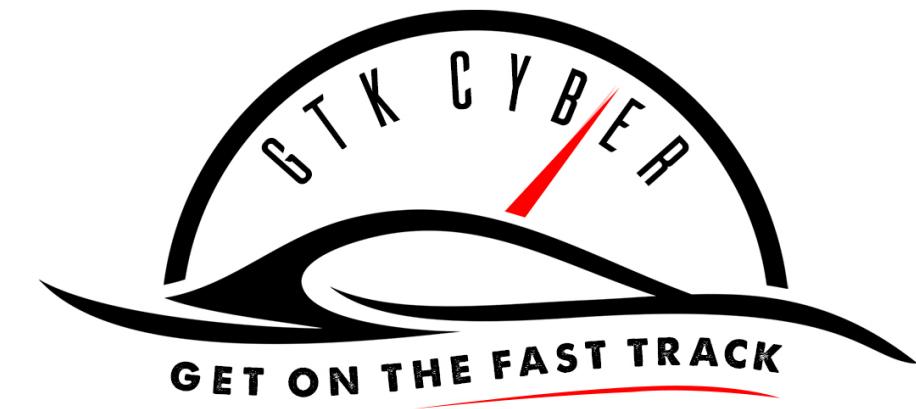




Why?

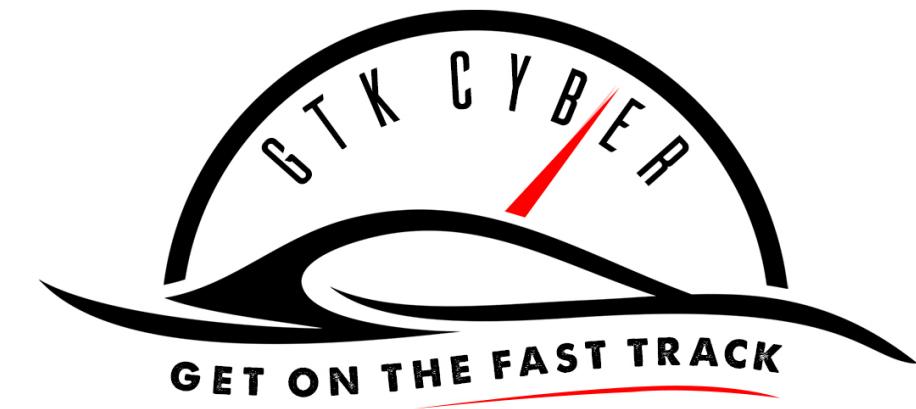


**Security Data comes in Many
Forms**



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

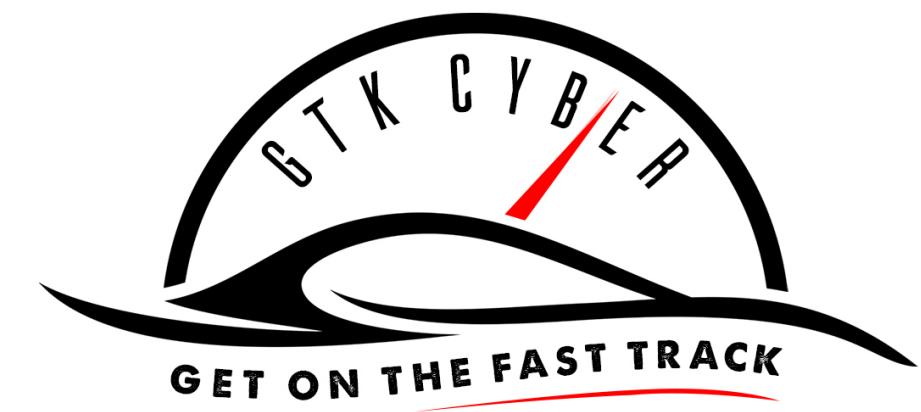
- File systems
- Cloud Storage: Azure / S3
- Databases
- Real Time Event Streams
- Other sources



**Few tools can effectively analyze
these data types effectively**

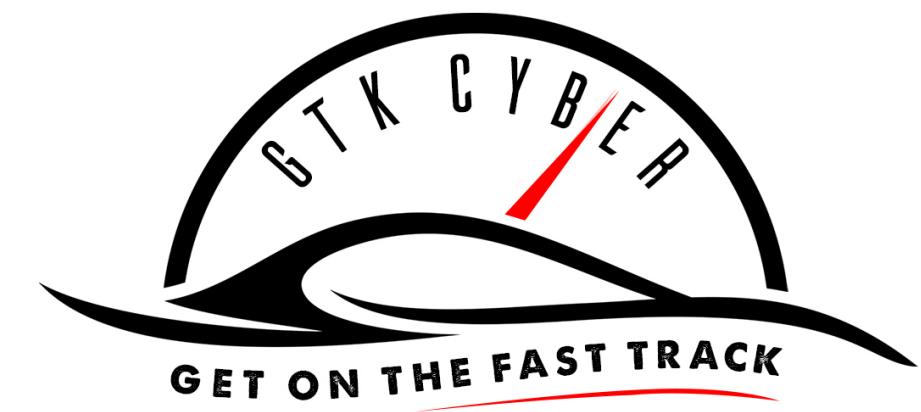


Even fewer tools can effectively analyze
ALL these data types effectively



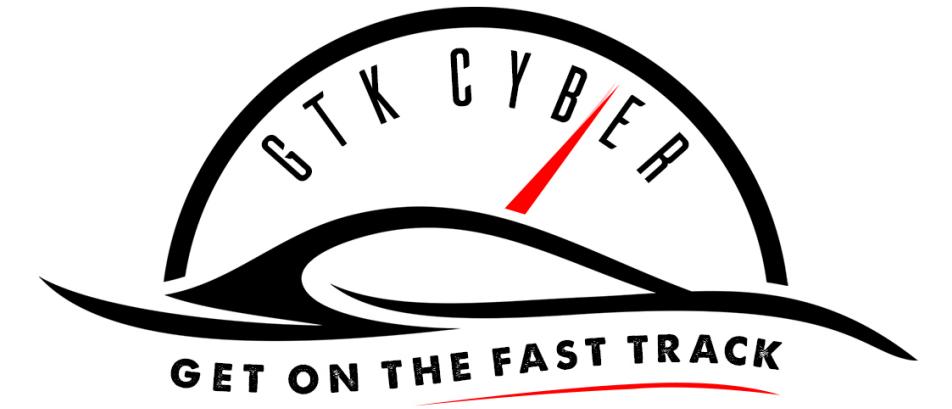
**Splunk and ELK are solid SIEM
platforms**





Wireshark is good for PCAP analysis

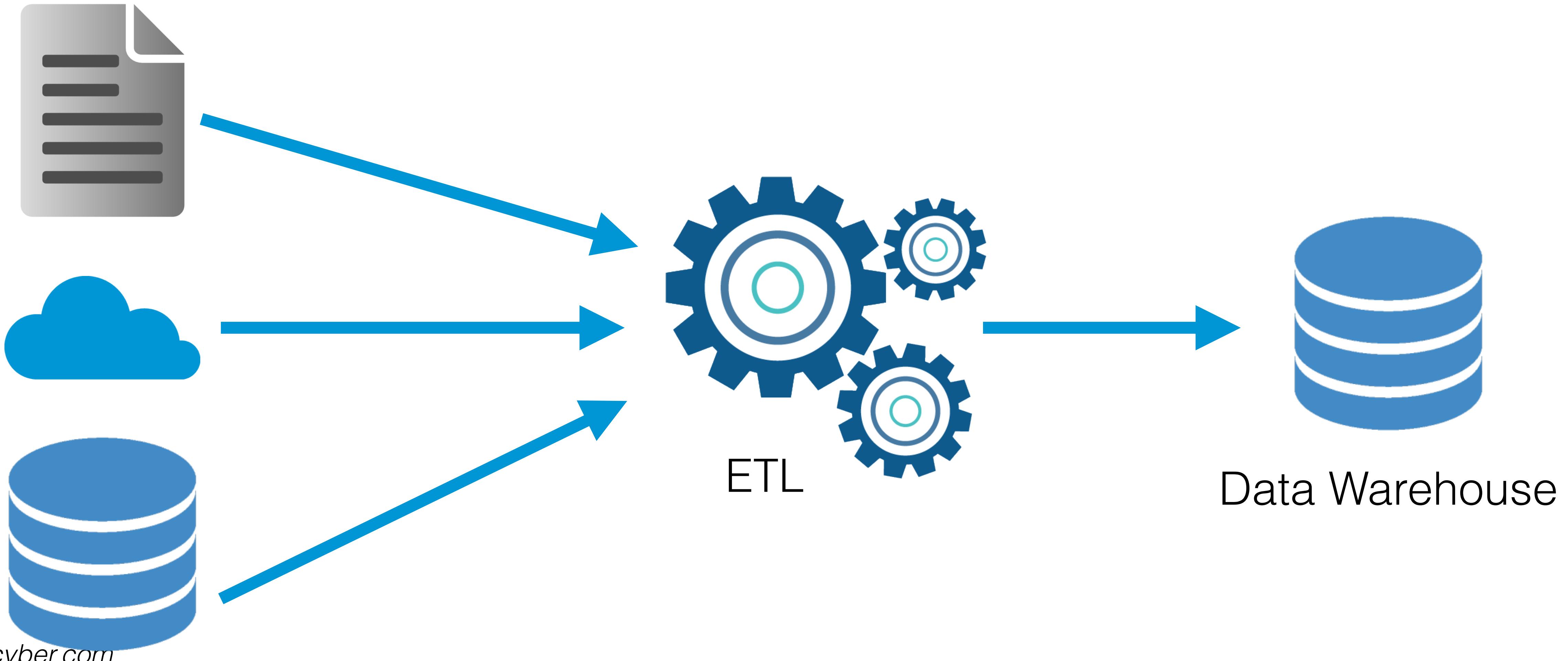


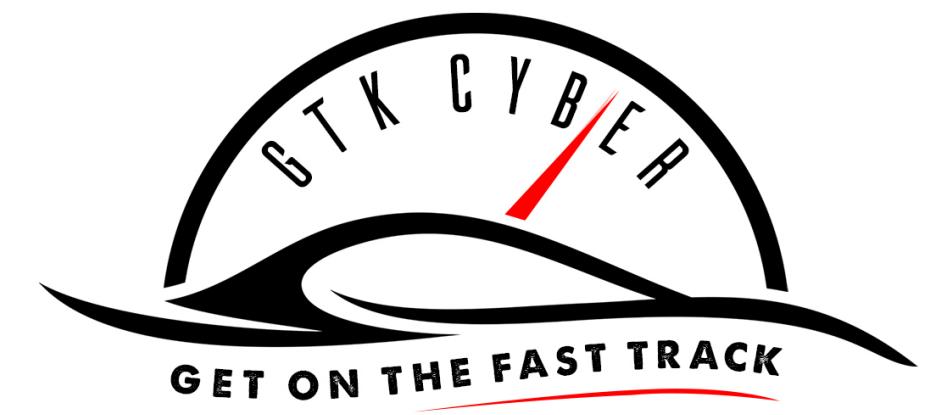


Security data is not arranged in an optimal way for ad-hoc analysis

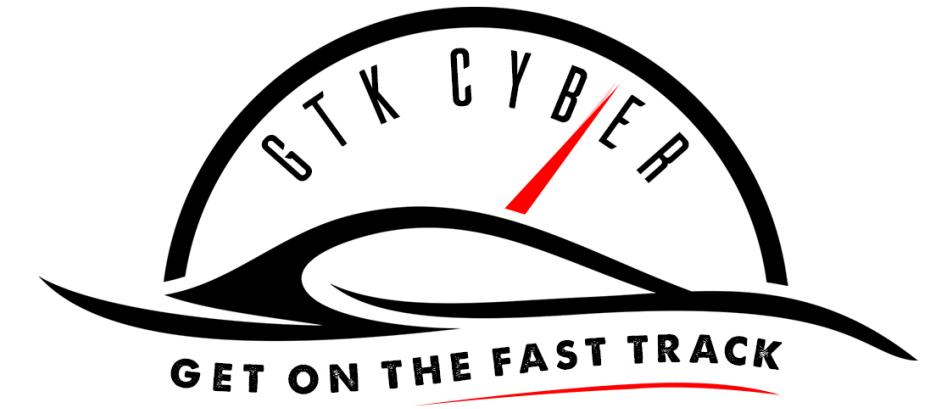


Security data is not arranged in an optimal way for ad-hoc analysis





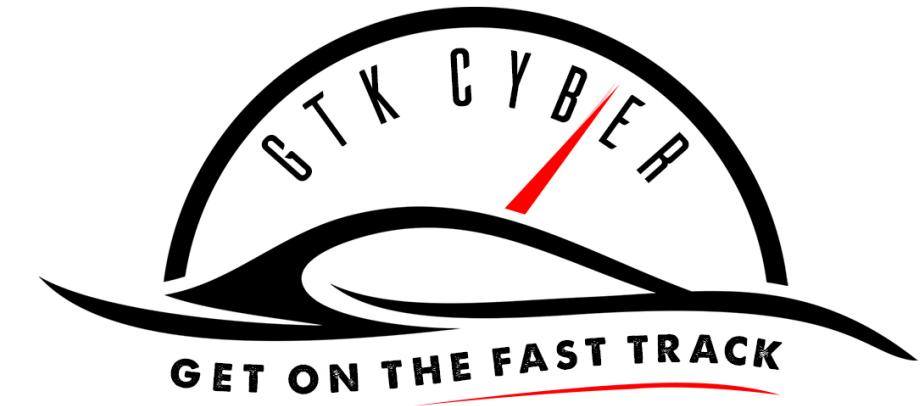
ETL is expensive and wasteful



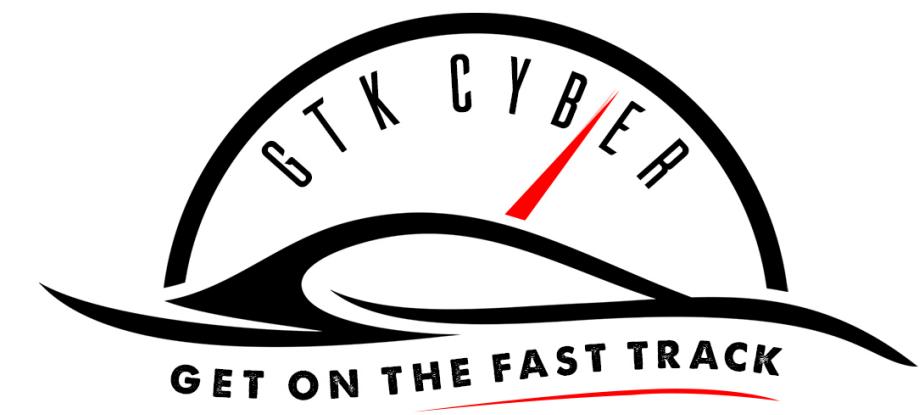
Analytics teams spend
between 50%-90% of their
time preparing their data.



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



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



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.



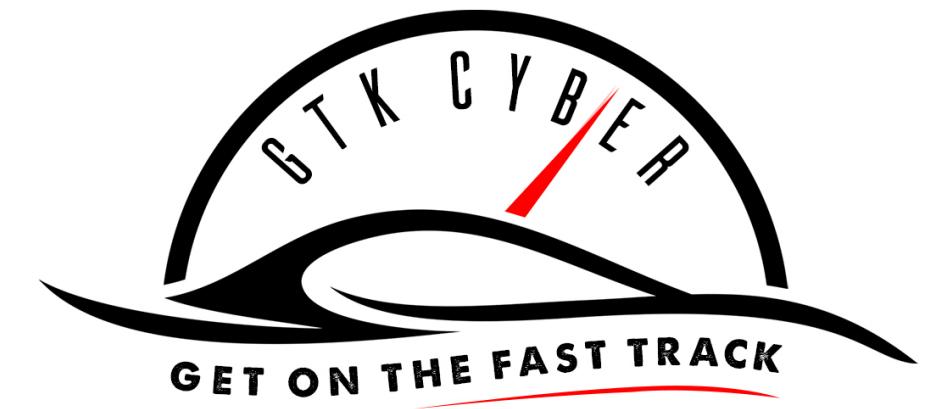
Motion

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



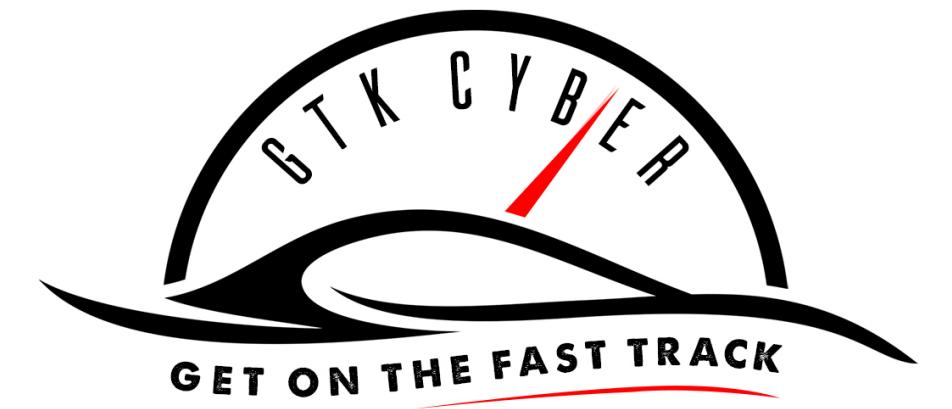
Extra-Processing

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



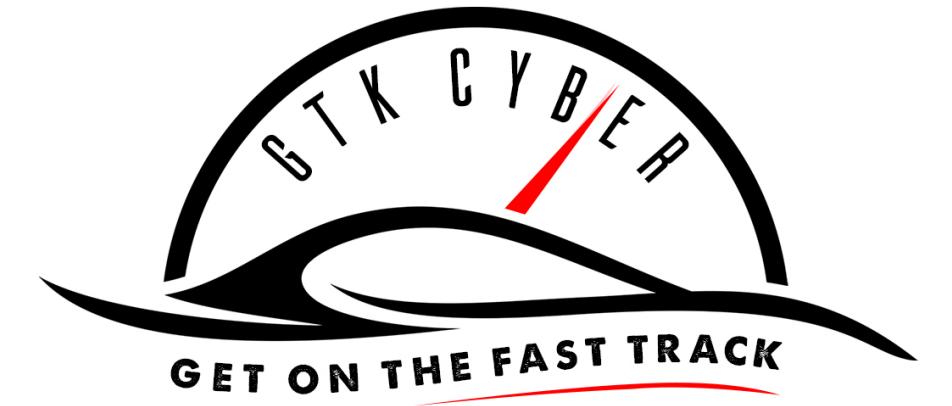
So where does Drill fit in?





**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.

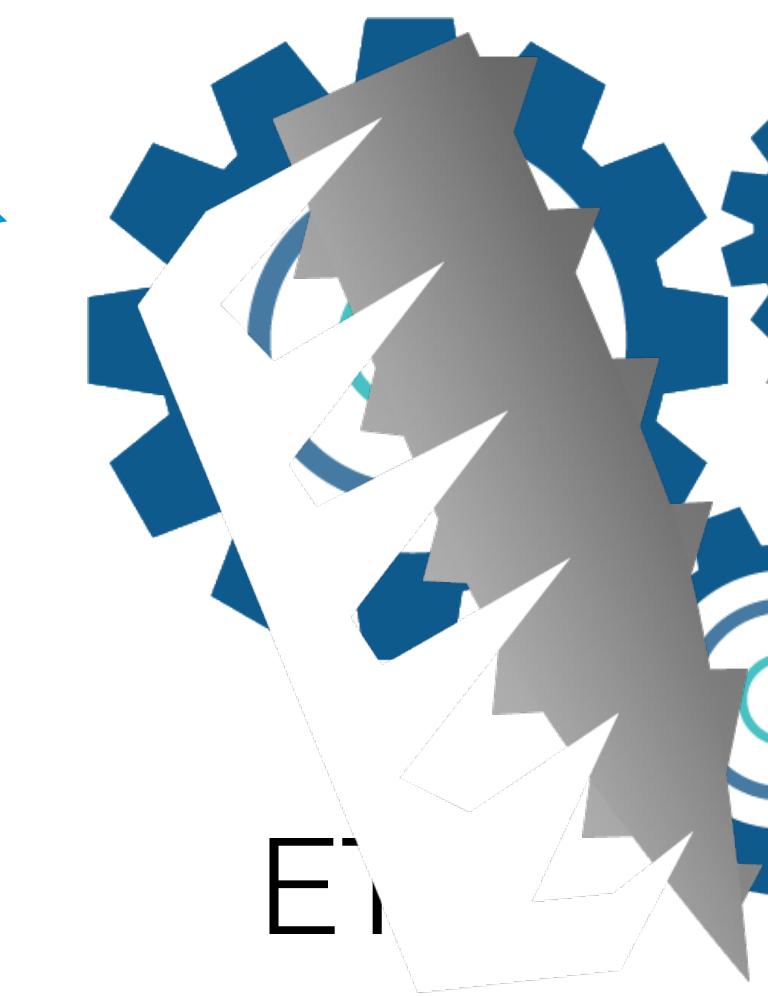
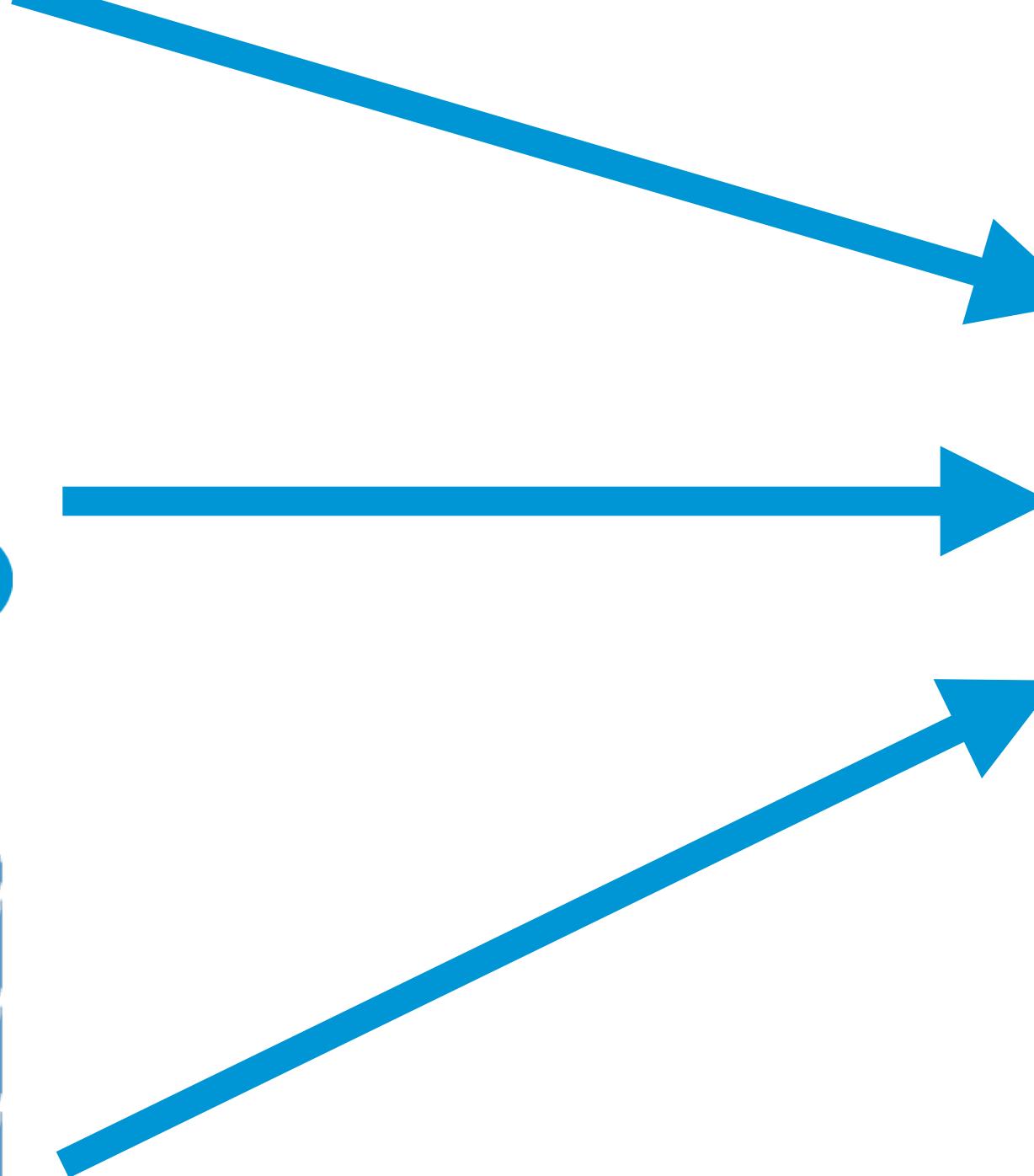
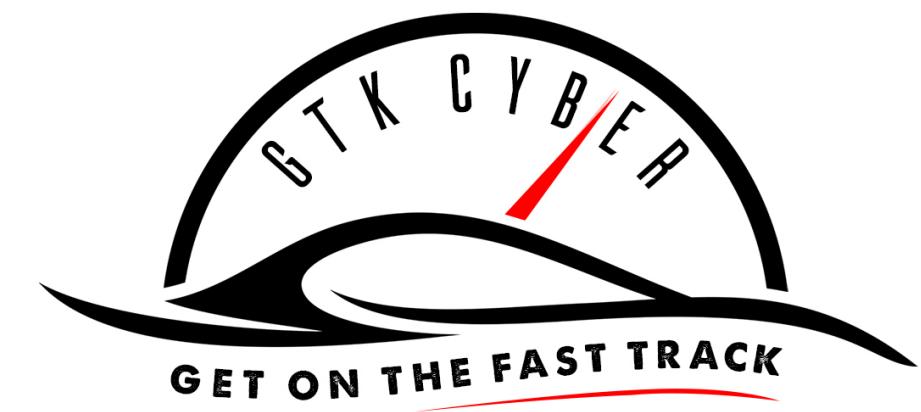
* well.. almost anything

** within reason





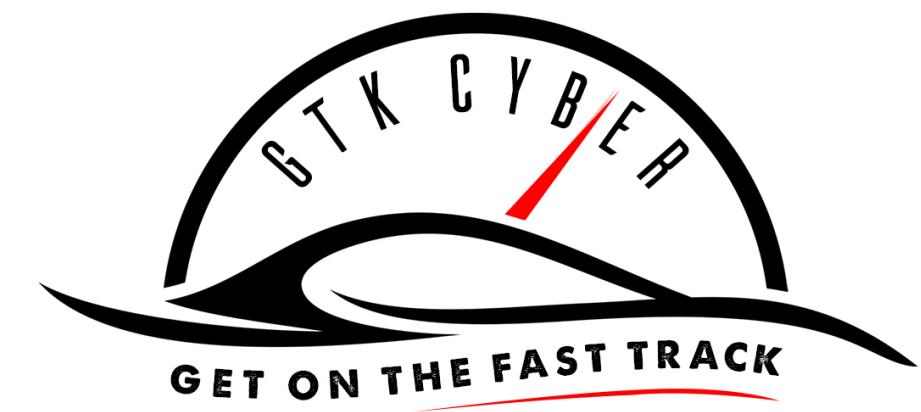
Drill acts as a universal
translator for data.



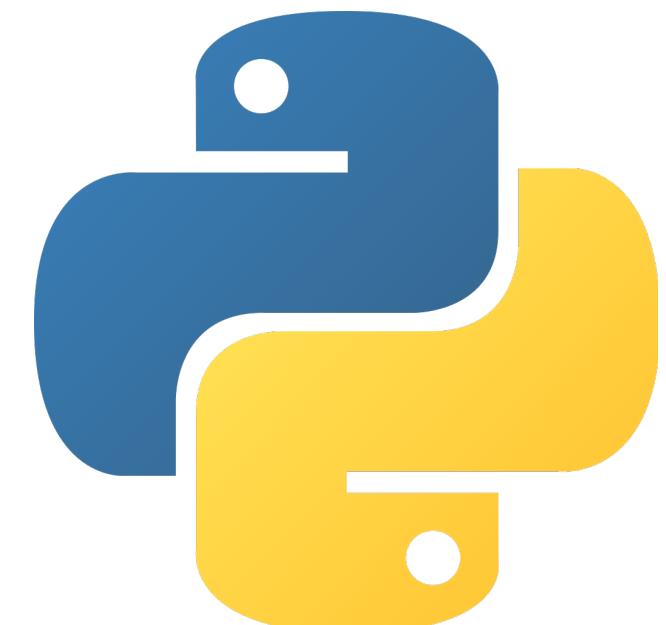
APACHE
DRILL

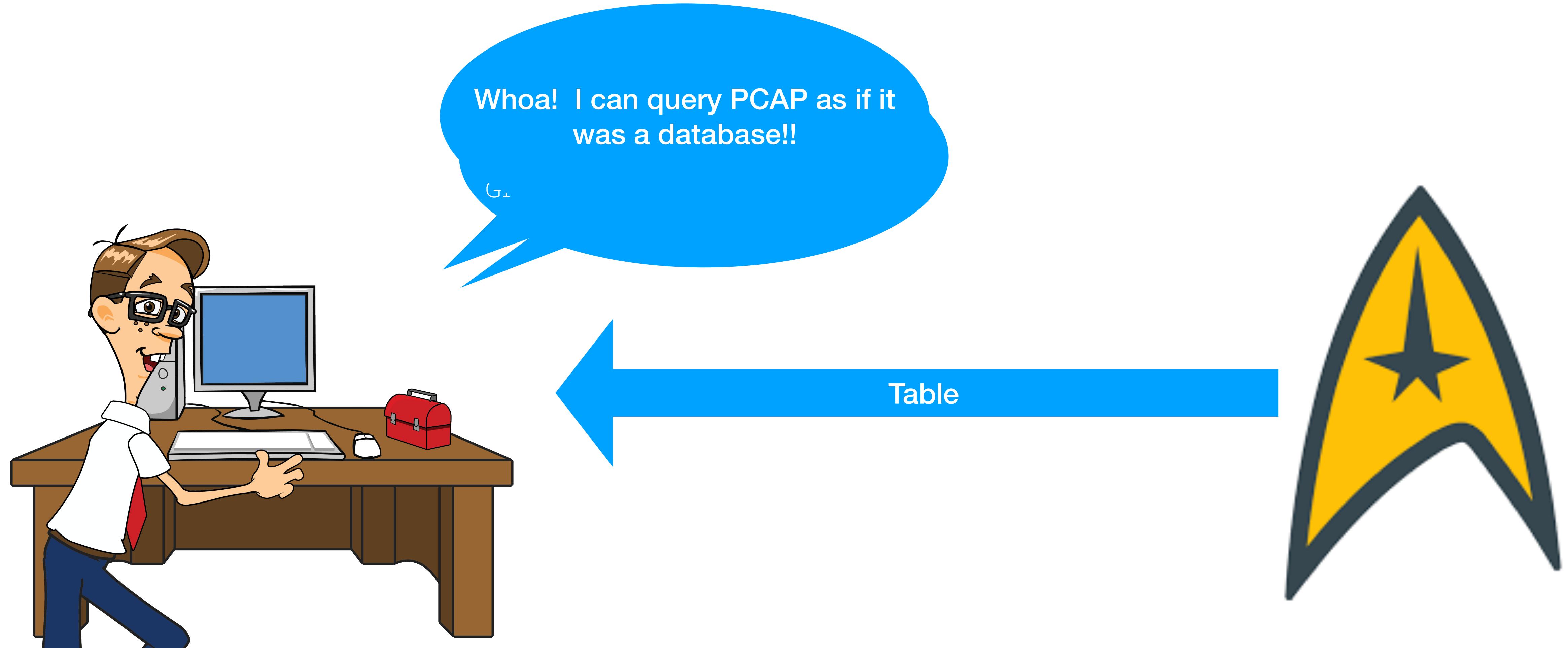
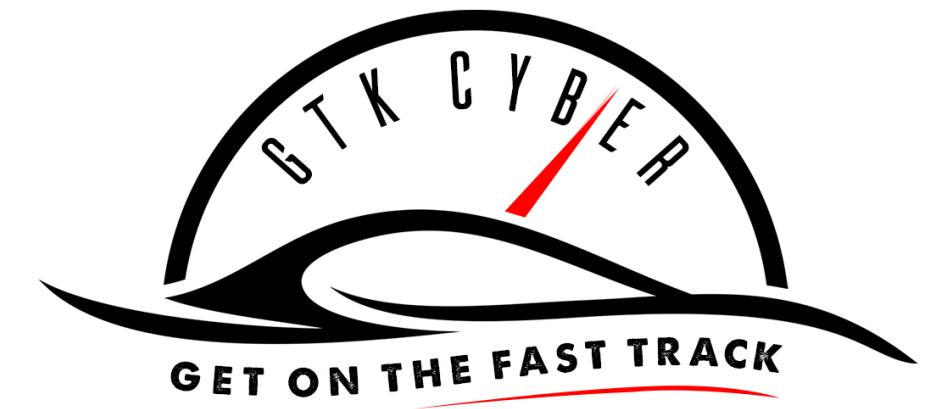


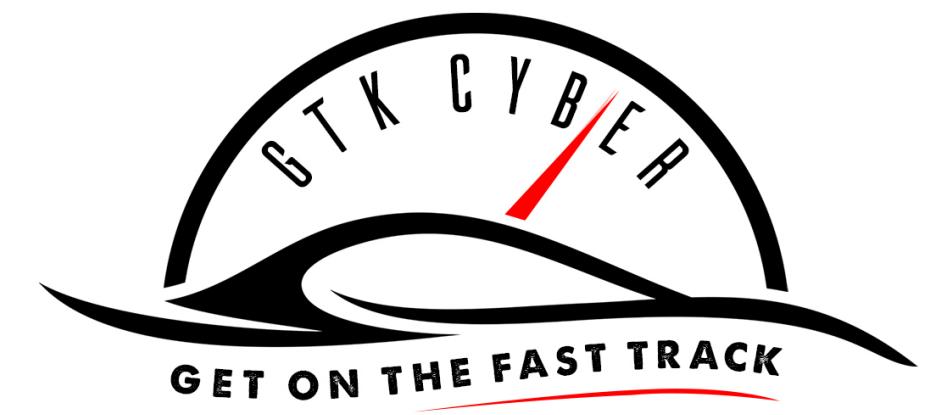
Data Warehouse



**APACHE
DRILL**







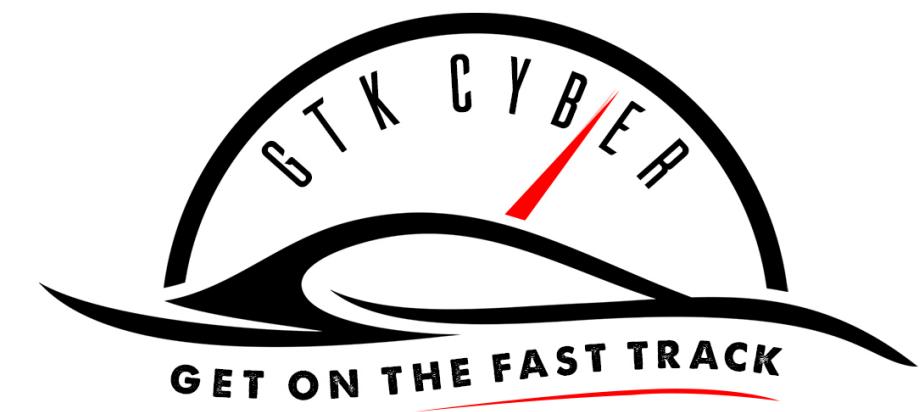
So let's take a look...



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

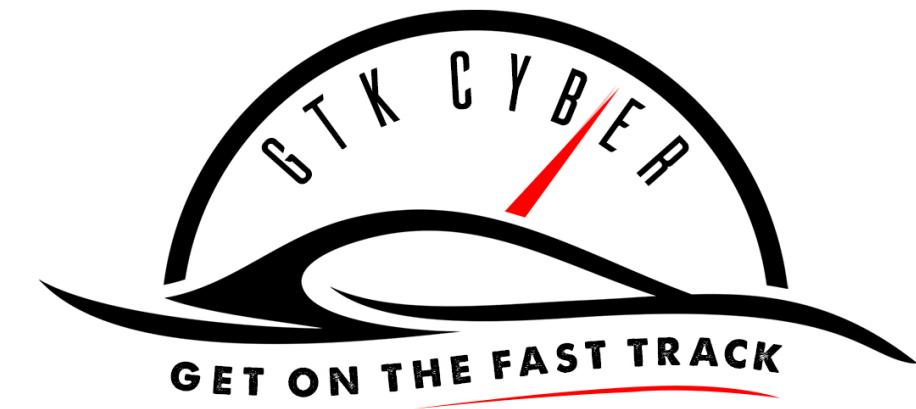
<http://thedataist.com/visualize-anything-with-superset-and-drill/>





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"

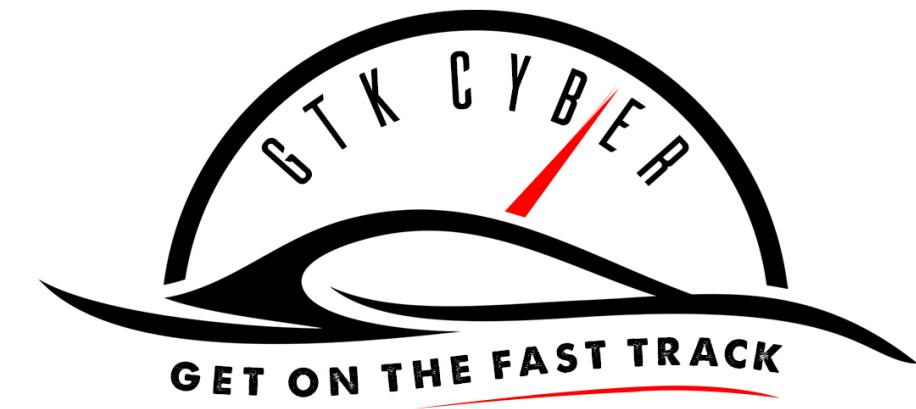


🕒 `hackers-access.httpd` ↴

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

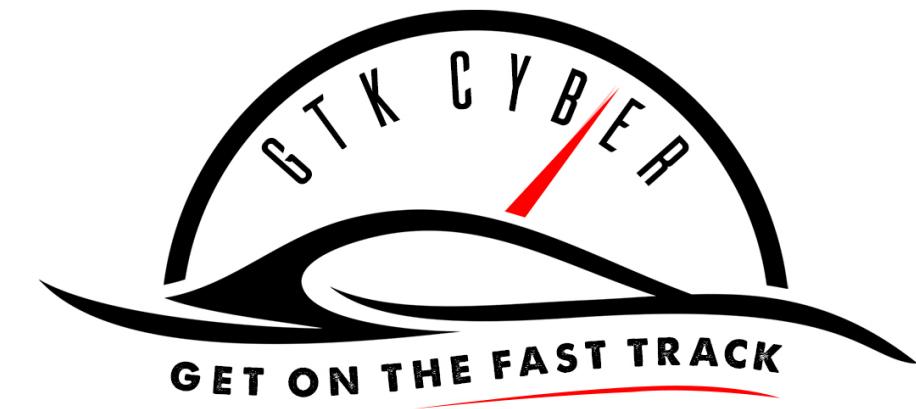


request_referer_last	request_receive_time_minute	connection_client_host_last	request_receive_time_last_millisecond_utc	request_firstline_original_uri
http://howto.basjes.nl/join_form				
http://howto.basjes.nl/	11	195.154.46.135	0	/linux/doing-pxe-without-dhcp-control
http://howto.basjes.nl/	11	23.95.237.180	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



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...



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'
```



Who is Trying to Hack This Site?

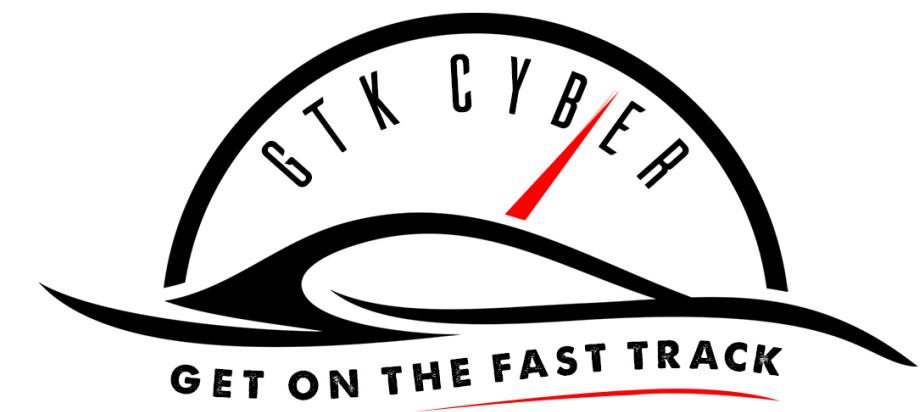
```
SELECT request_receive_time,  
connection_client_host,  
request_useragent  
FROM dfs.test.`hackers-access.httpd`  
WHERE request_firstline_uri = '/join_form'
```

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

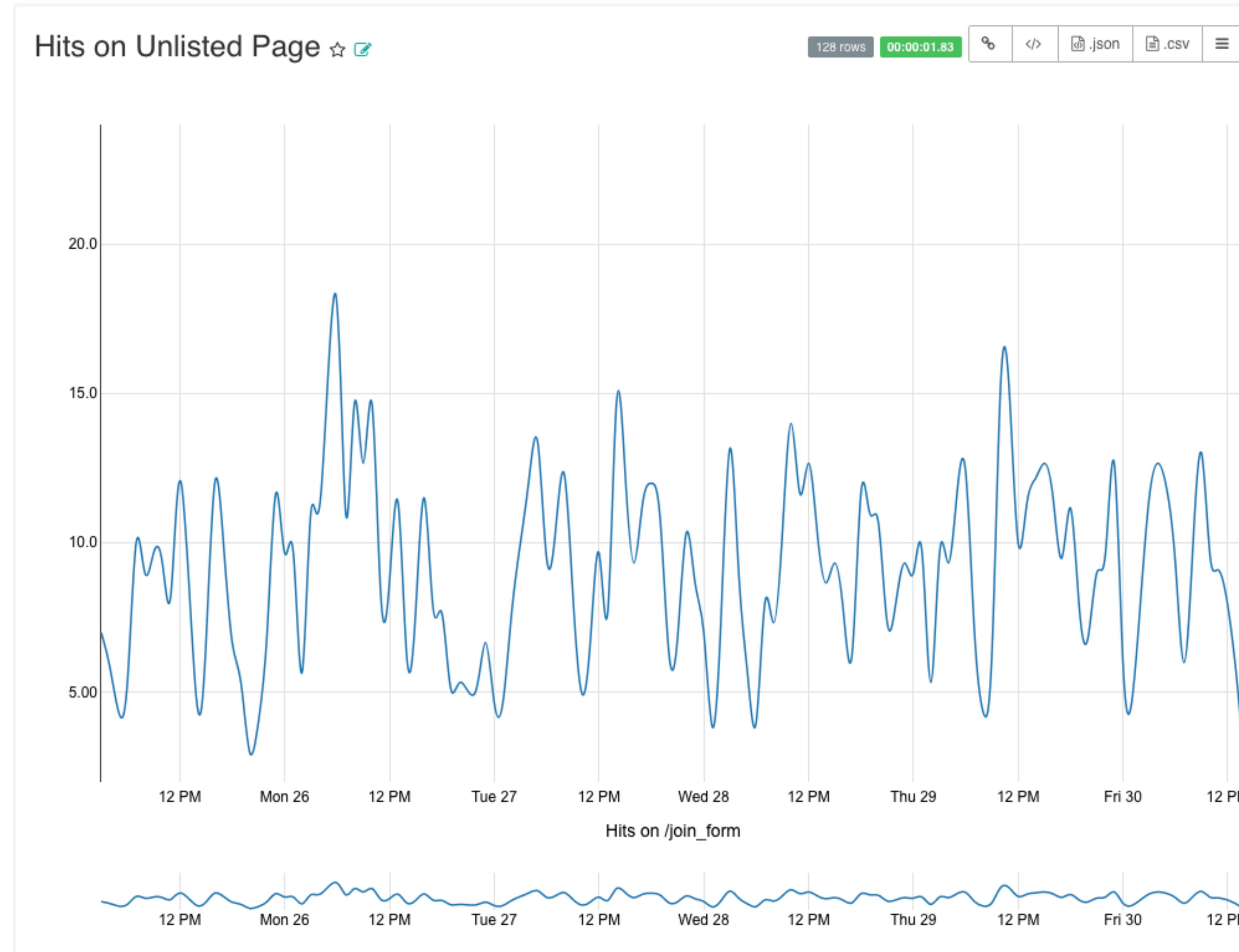
237.180

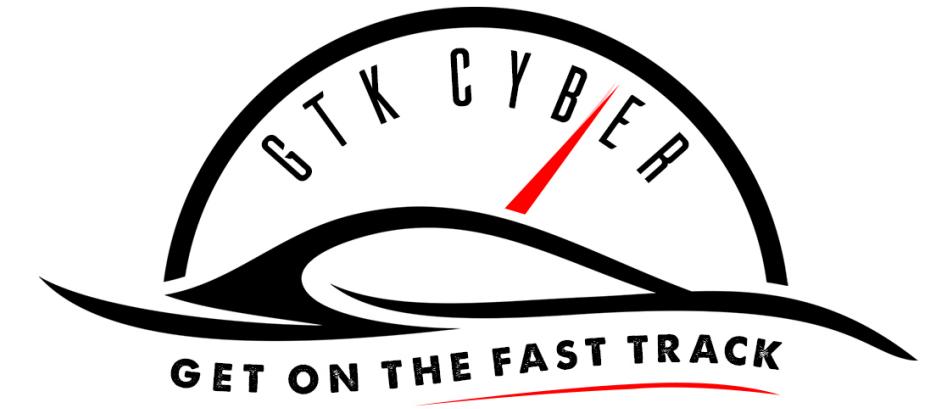
Explore .CSV Clipboard Search Results

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

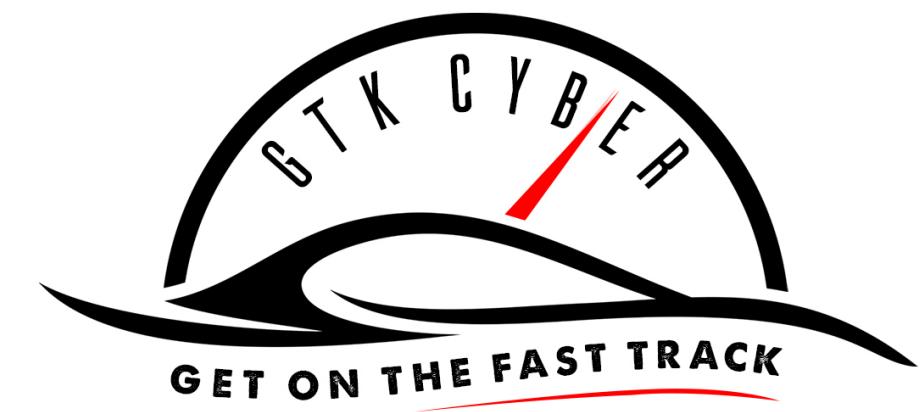


Who is Trying to Hack This Site?





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



Who is Trying to Hack This Site?

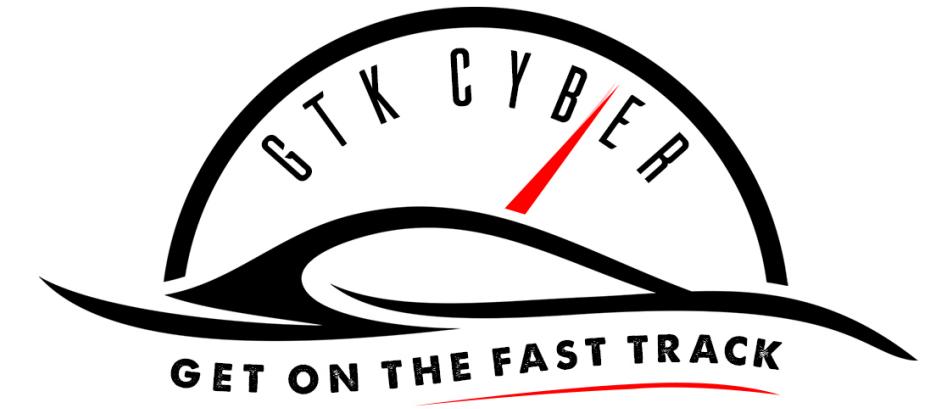
```
SELECT request_receive_time,  
connection_client_host,  
request_useragent  
FROM dfs.test.`hackers-access.httpd`  
WHERE request_firstline_uri = '/join_form'
```

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

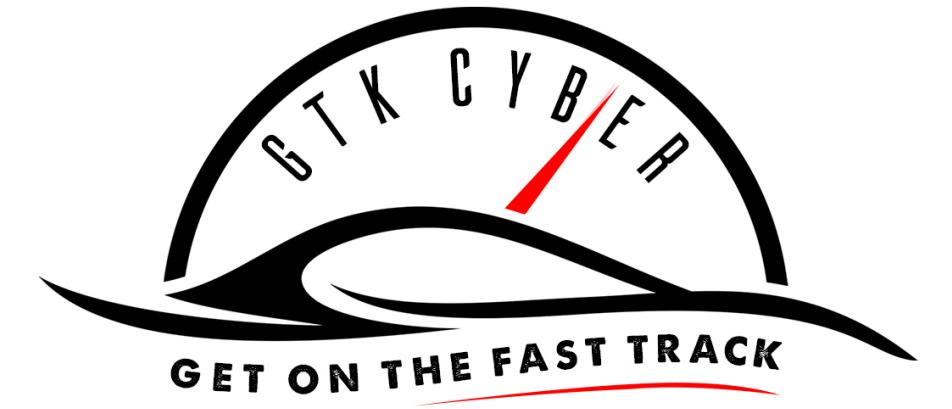
237.180

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

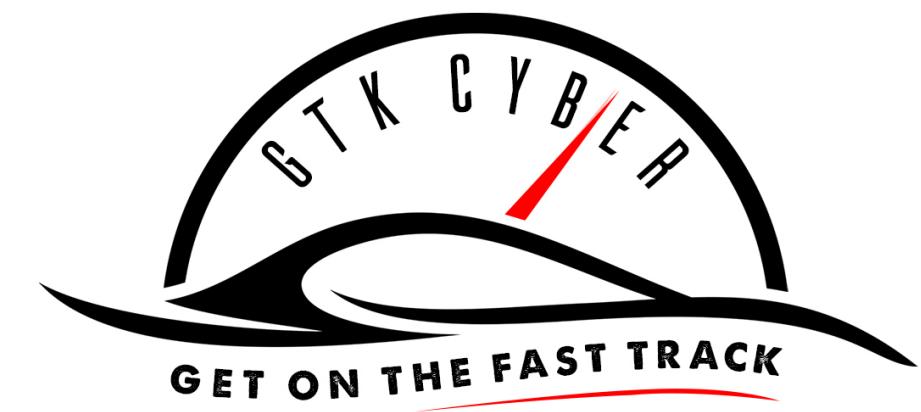


**Drill's Flexible UDF Interface Allows
you to write your own functions.**



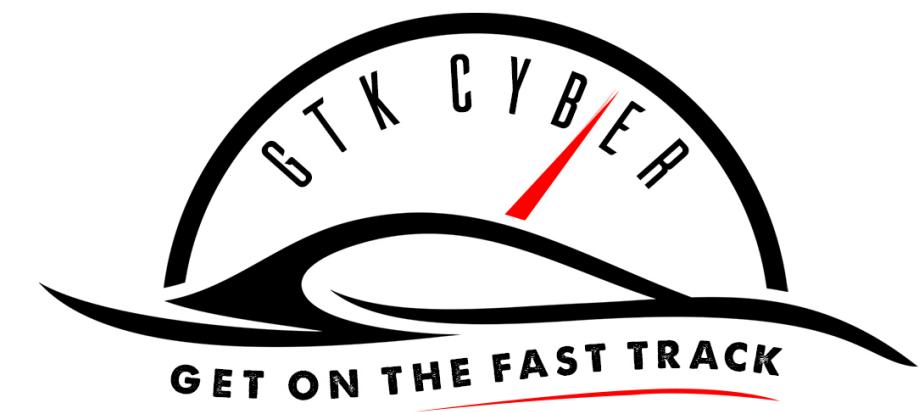
A collection of GeolIP Functions
is available on GitHub.

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



Drill GeolIP Functions

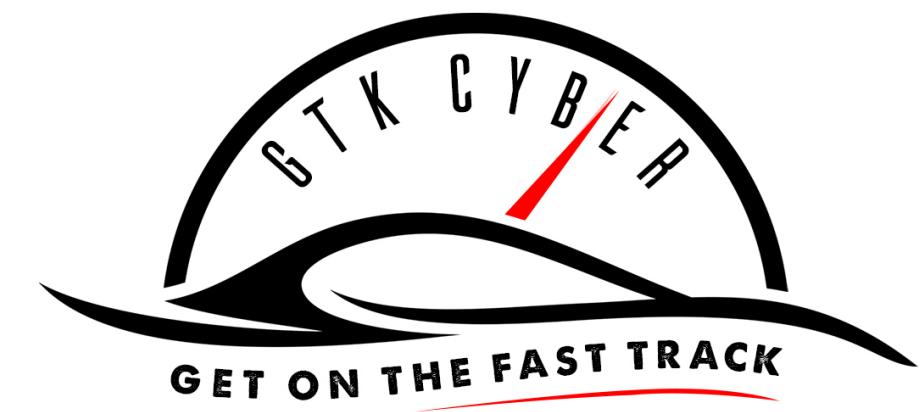
- **getCountryName(<ip>)**: This function returns the country name of the IP address, "Unknown" if the IP is unknown or invalid.
- **getCountryConfidence(<ip>)**: This function returns the confidence score of the country ISO code of the IP address.
- **getCountryISOCode(<ip>)**: This function returns 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.
- **isPublicProxy(<ip>)**: This function returns `true` if the ip address is a public proxy, `false` if not.
- **isT0RExitNode(<ip>)**: This function returns `true` if the ip address is a known TOR exit node, `false` if not.



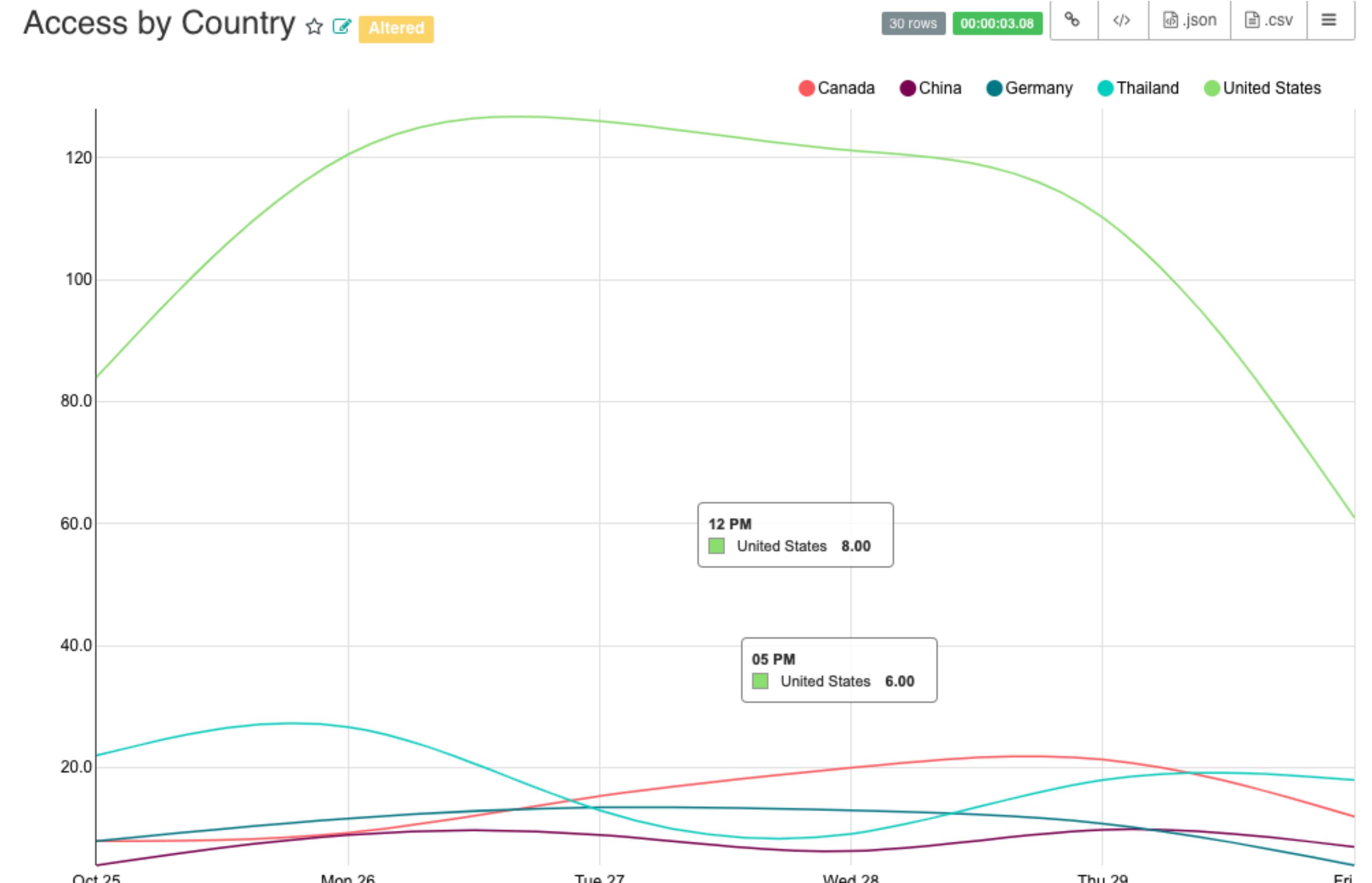
Who is Trying to Hack This Site?

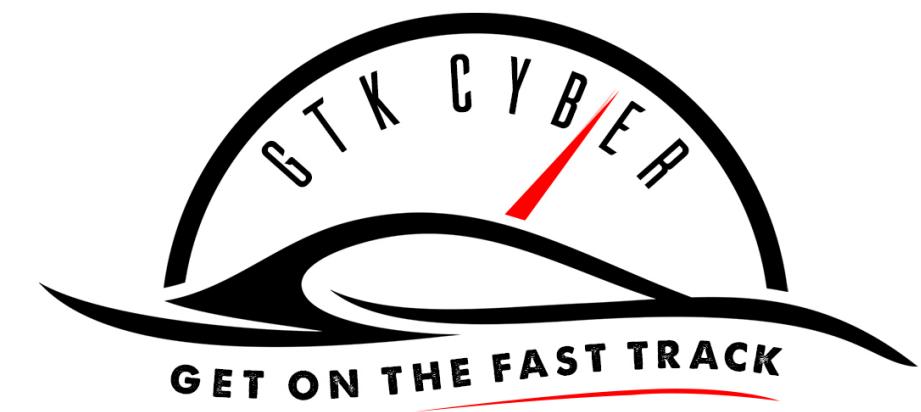
```
SELECT request_receive_time, connection_client_host,  
request_useragent,  
getCountryName(connection_client_host ) as countryName,  
getCityName(connection_client_host ) as cityName  
FROM dfs.test.`hackers-access.httpd`  
WHERE request_firstline_uri = '/join_form'
```

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?

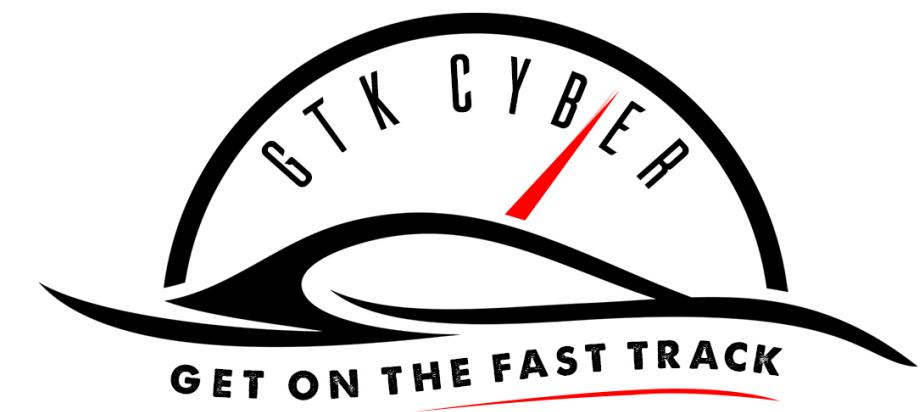




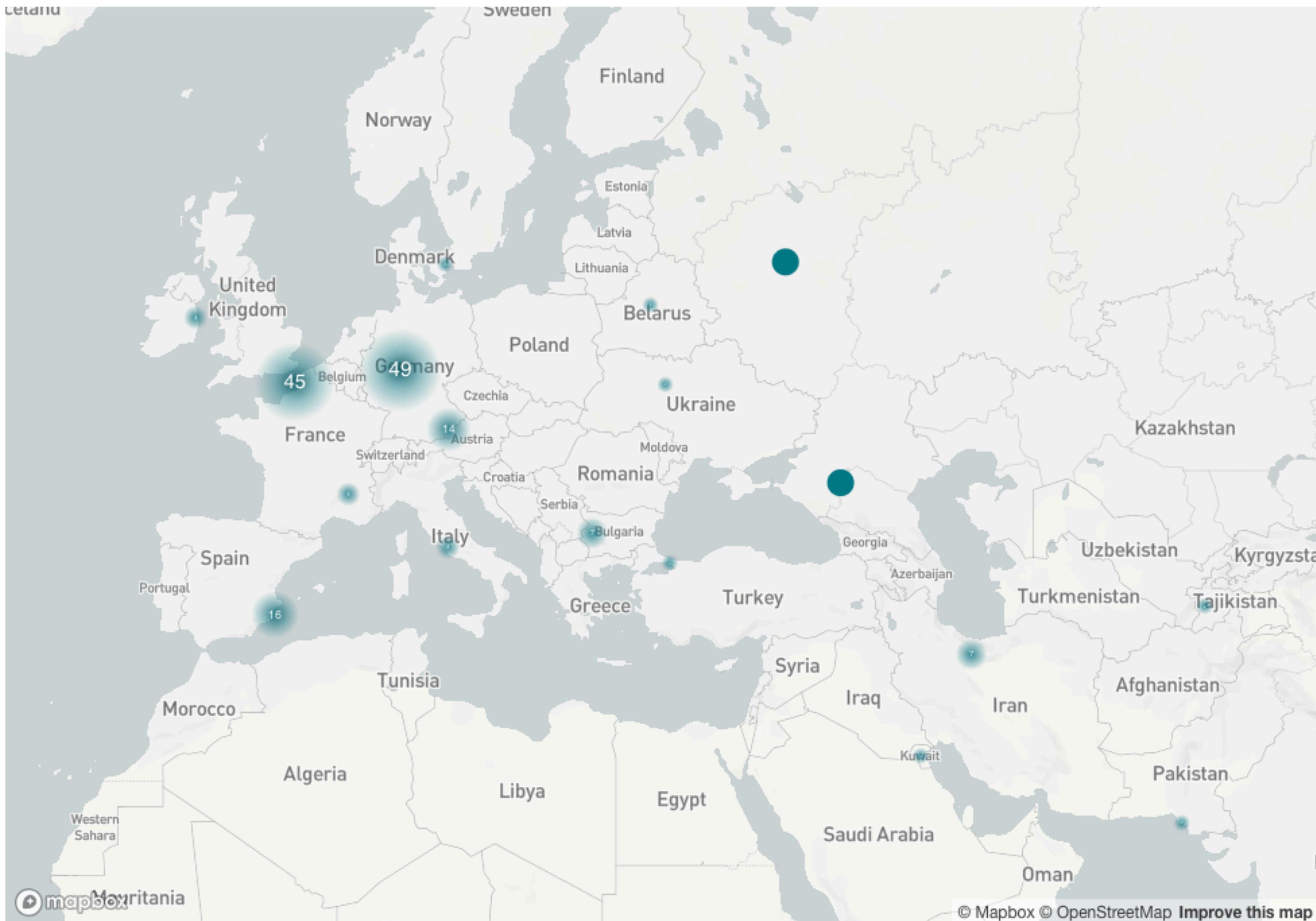
Who is Trying to Hack This Site?

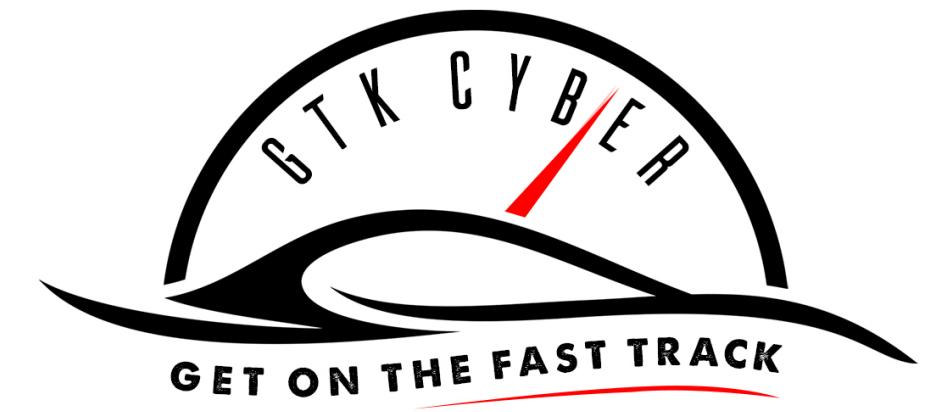
```
SELECT request_receive_time, connection_client_host, request_useragent,  
getCountryName(connection_client_host ) as countryName,  
getCityName(connection_client_host ) as cityName,  
getLatitude(connection_client_host ) as latitude,  
getLongitude(connection_client_host ) as longitude  
FROM dfs.test.`hackers-access.httpd`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

request_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

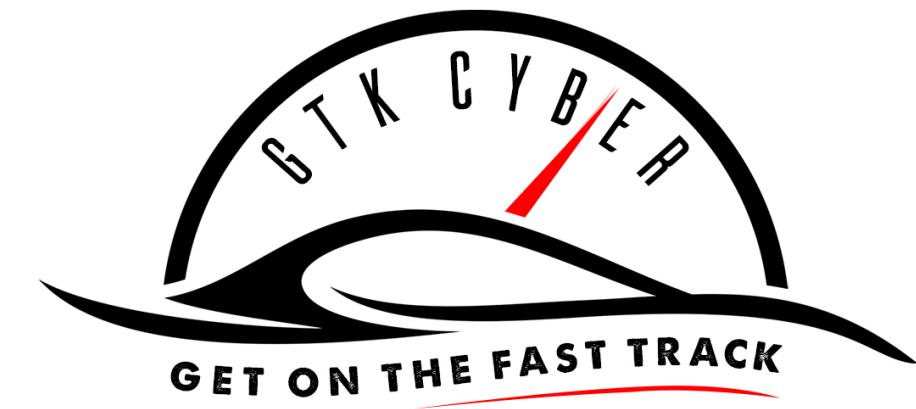


Who is Trying to Hack This Site?



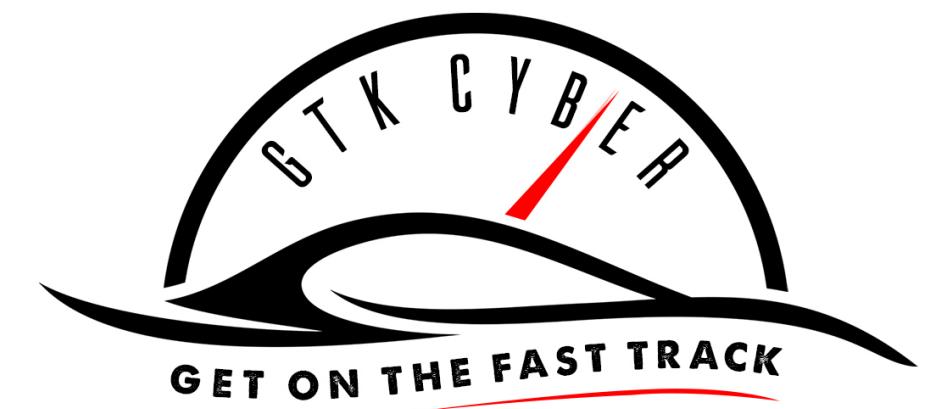


What equipment are they using?



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/niebsbasjes/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
```

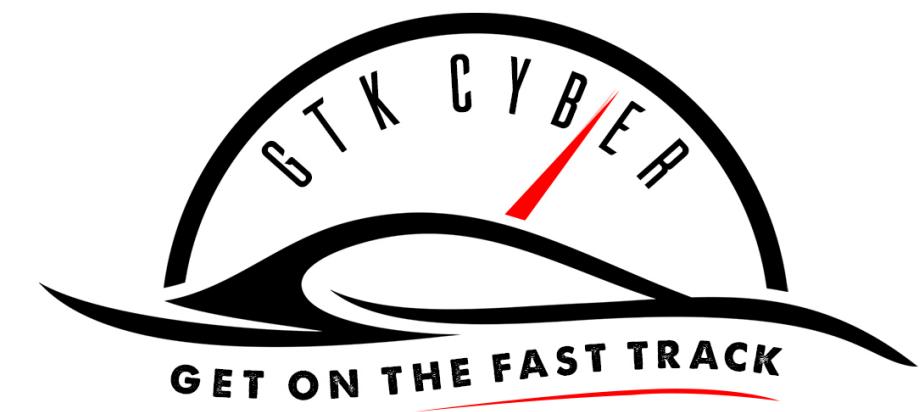
```
{"DeviceClass":"Desktop","DeviceName":"Desktop","DeviceBrand":"Unknown","DeviceCpuBits":"32","OperatingSystemClass":"Desktop","OperatingSystemName":"Windows NT","OperatingSystemVersion":"XP","Operati
```

{"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

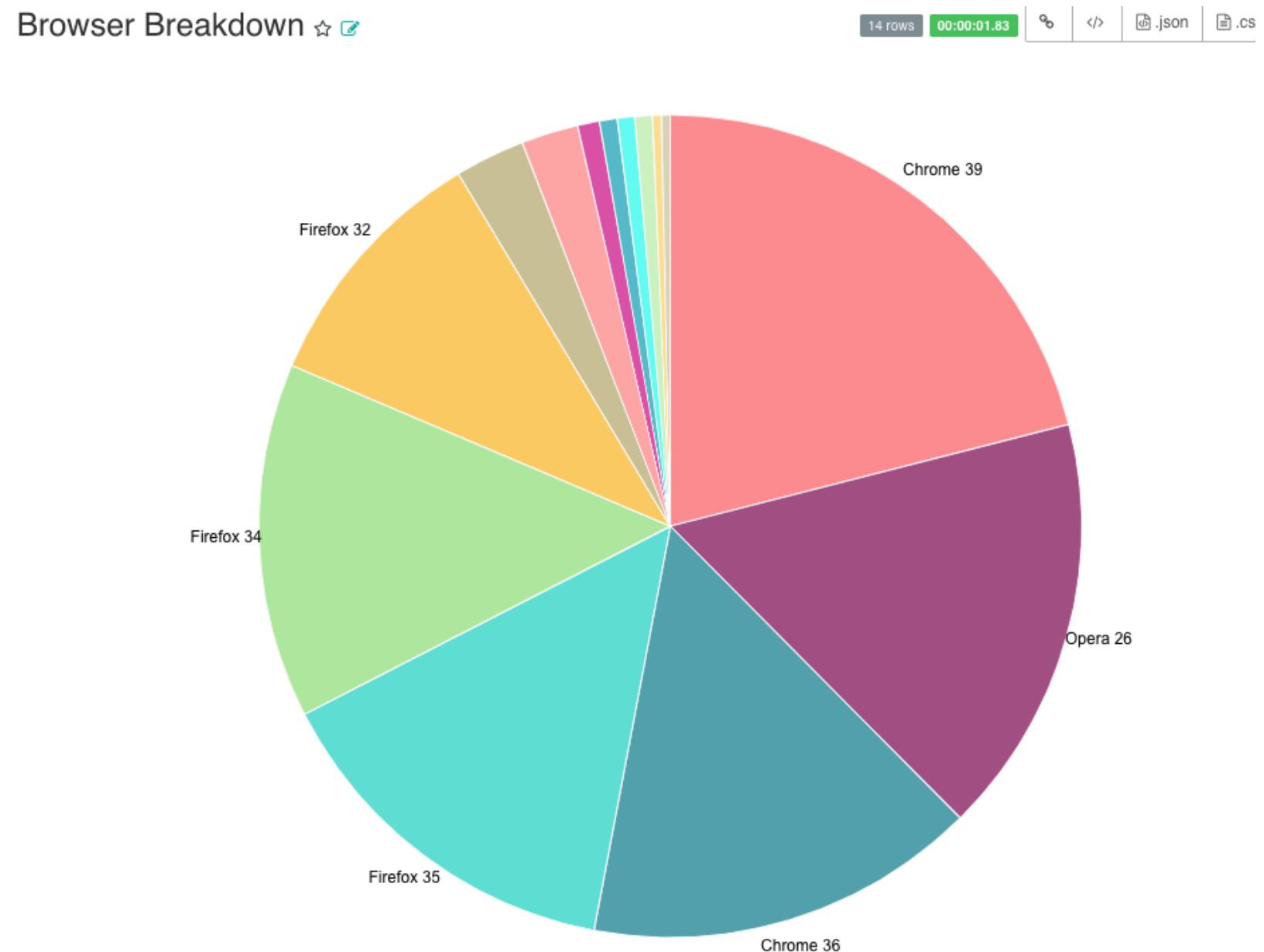
./"DeviceClass":"Desktop" "DeviceName":"Desktop" "DeviceBrand":"I Unkown" "DeviceCpuBits":"32" "OperatingSystemClass":"Desktop" "OperatingSystemName":"Windows NT" "OperatingSystemVersion":"XP" "Operati

"DeviceClass": "Desktop", "DeviceName": "Desktop", "DeviceBrand": "Unknown", "DeviceCPUType": "00", "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
```

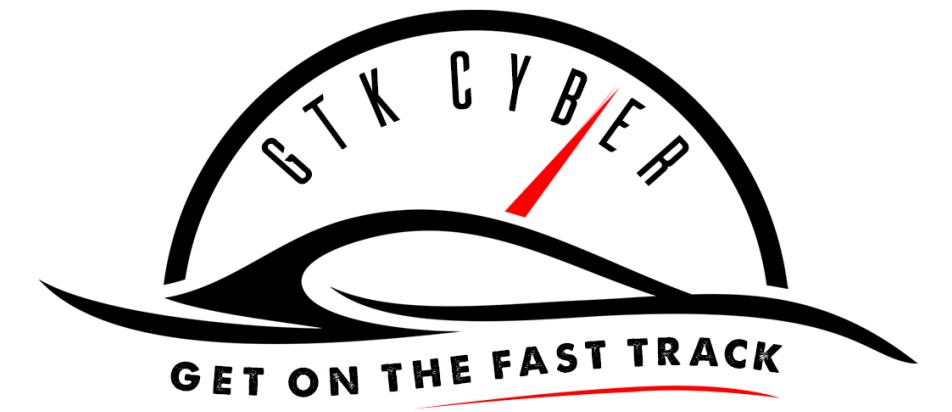


Firefox 34

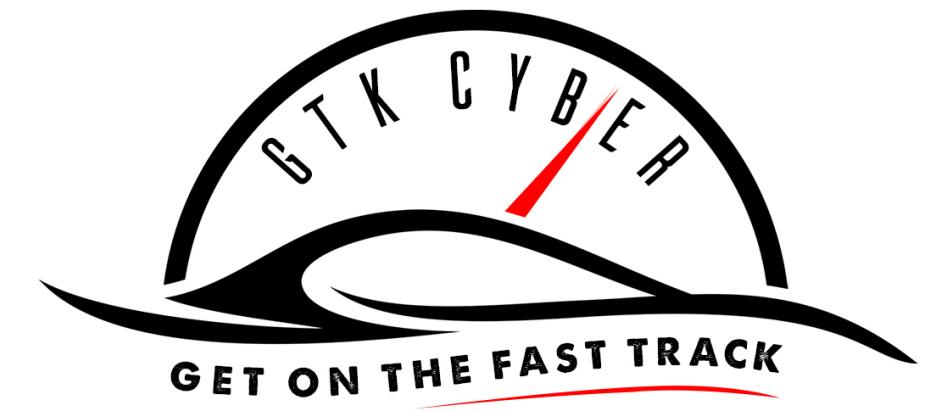
.CSV

Clipboard

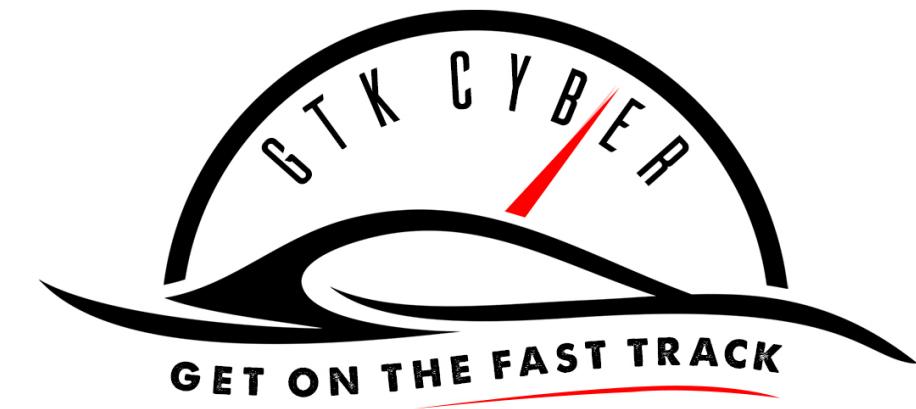
os	browser
Windows XP	Firefox 35
Windows XP	Firefox 35
Windows 8.1	AlexaToolbar alxf
Windows 8.1	AlexaToolbar alxf
Windows XP	Firefox 34
Windows XP	Firefox 35
Windows XP	Firefox 35
Windows XP	Firefox 34



Questions so far?

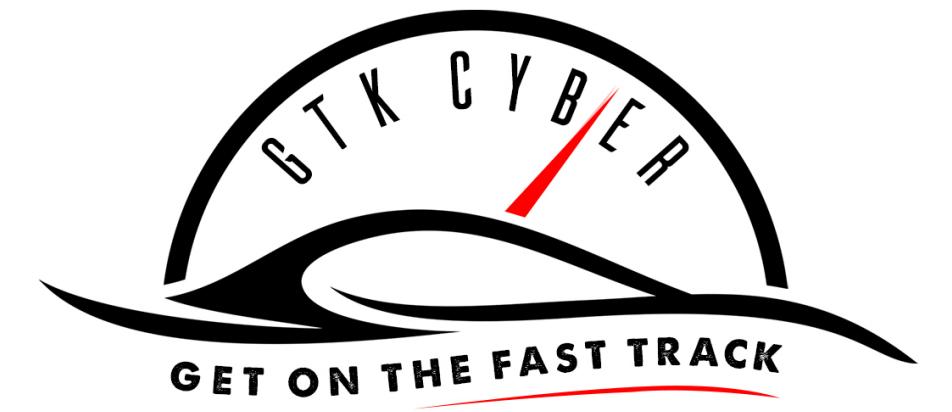


Let's have some fun with PCAP



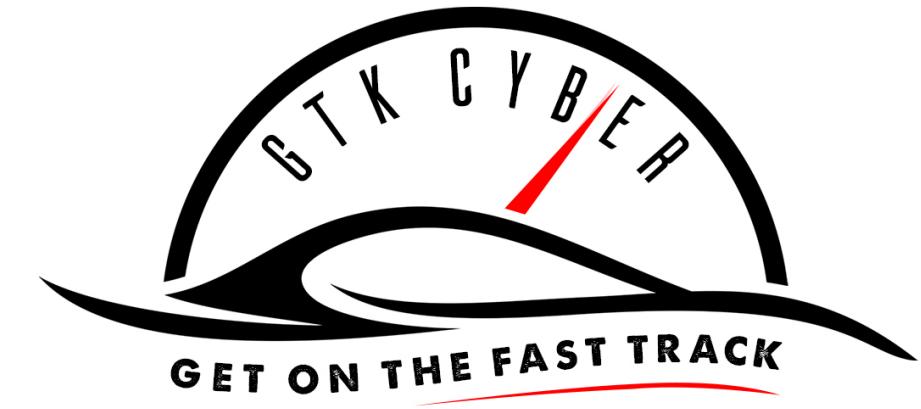
Let's have some fun with PCAP

- PCAP is short for **Packet Capture** 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	↓ A Z	CSV	X
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		



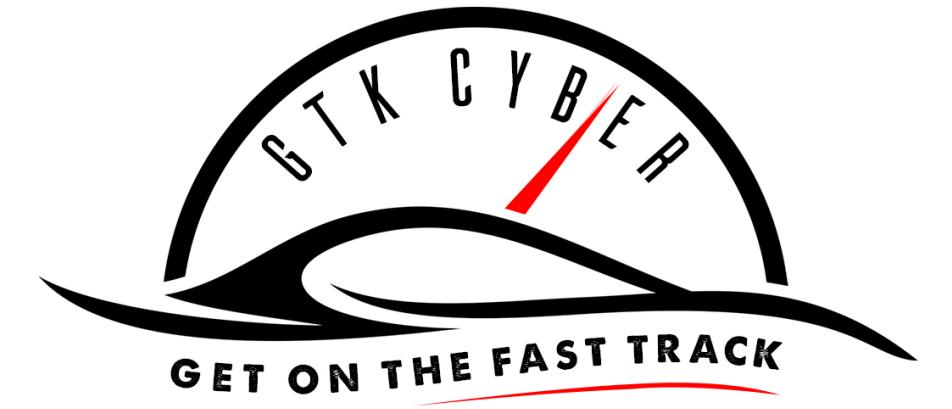
PCAP Analysis

- 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.

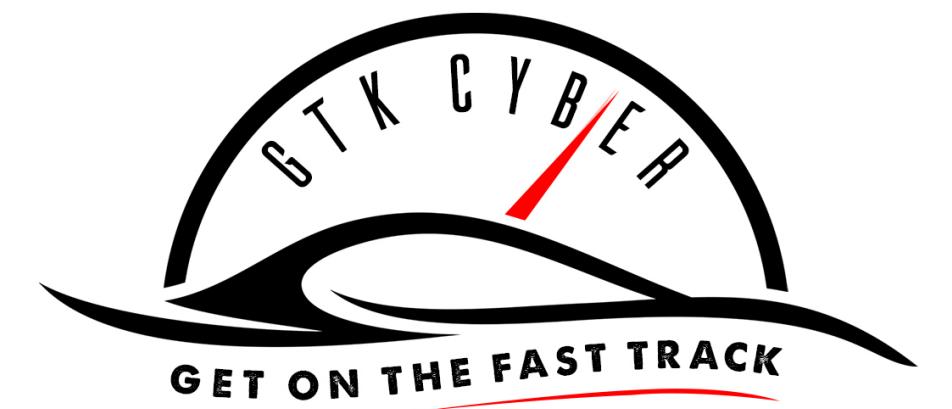


PCAP Analysis

type	network	timestamp	timestamp_micro	src_ip	dst_ip	src_port	dst_port	src_mac_address	dst_mac_address
2005-03-30T08:47:50.501000									
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



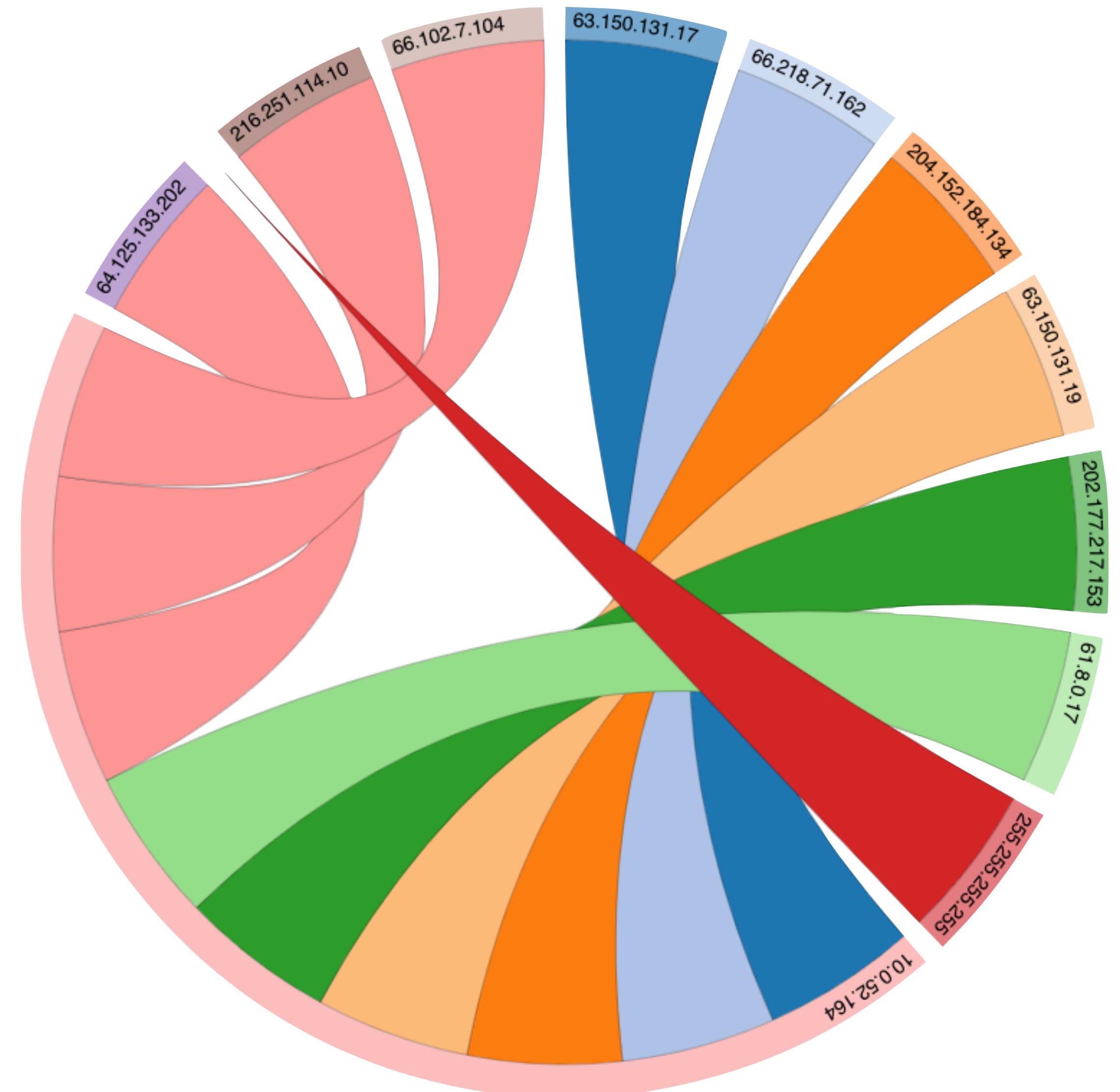
**Let's look for who is talking to
whom...**

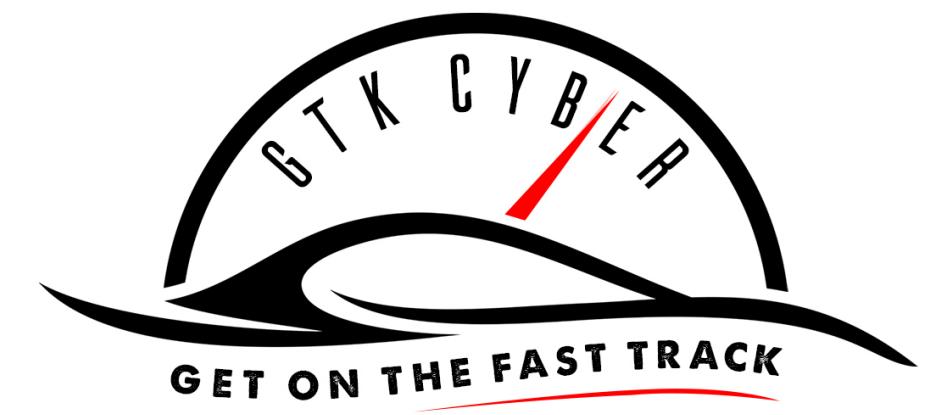


PCAP Analysis

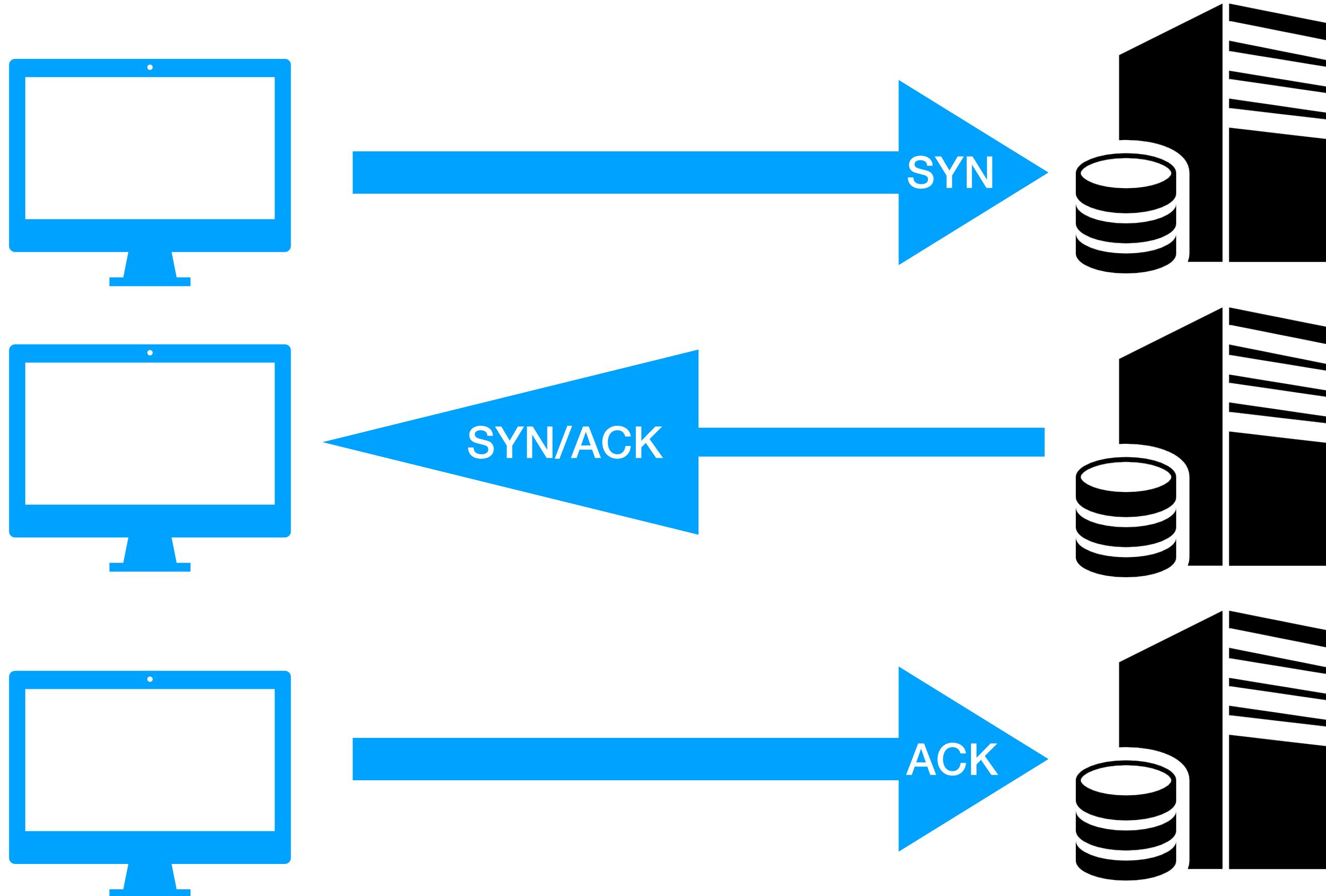
```
SELECT DISTINCT src_ip, dst_ip  
FROM dfs.test.`slowdownload.pcap`  
WHERE src_ip is not null
```

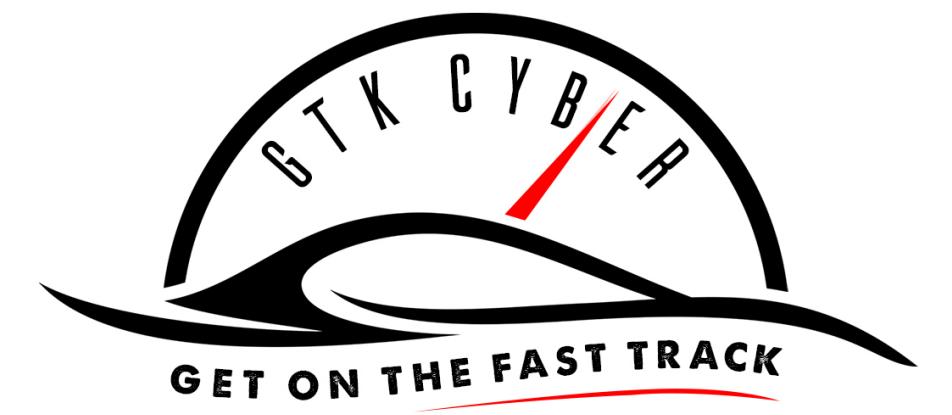
src_ip	dst_ip
10.100.252.1	255.255.255.255
10.0.52.164	216.251.114.10
10.0.52.164 4.10	10.0.52.164
10.0.52.164	66.218.71.162
66.218.71.162	10.0.52.164
10.0.52.164	66.102.7.104
66.102.7.104	10.0.52.164
10.0.52.164	202.177.217.153
202.177.217.153	10.0.52.164
10.0.52.164	63.150.131.19
63.150.131.19	10.0.52.164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



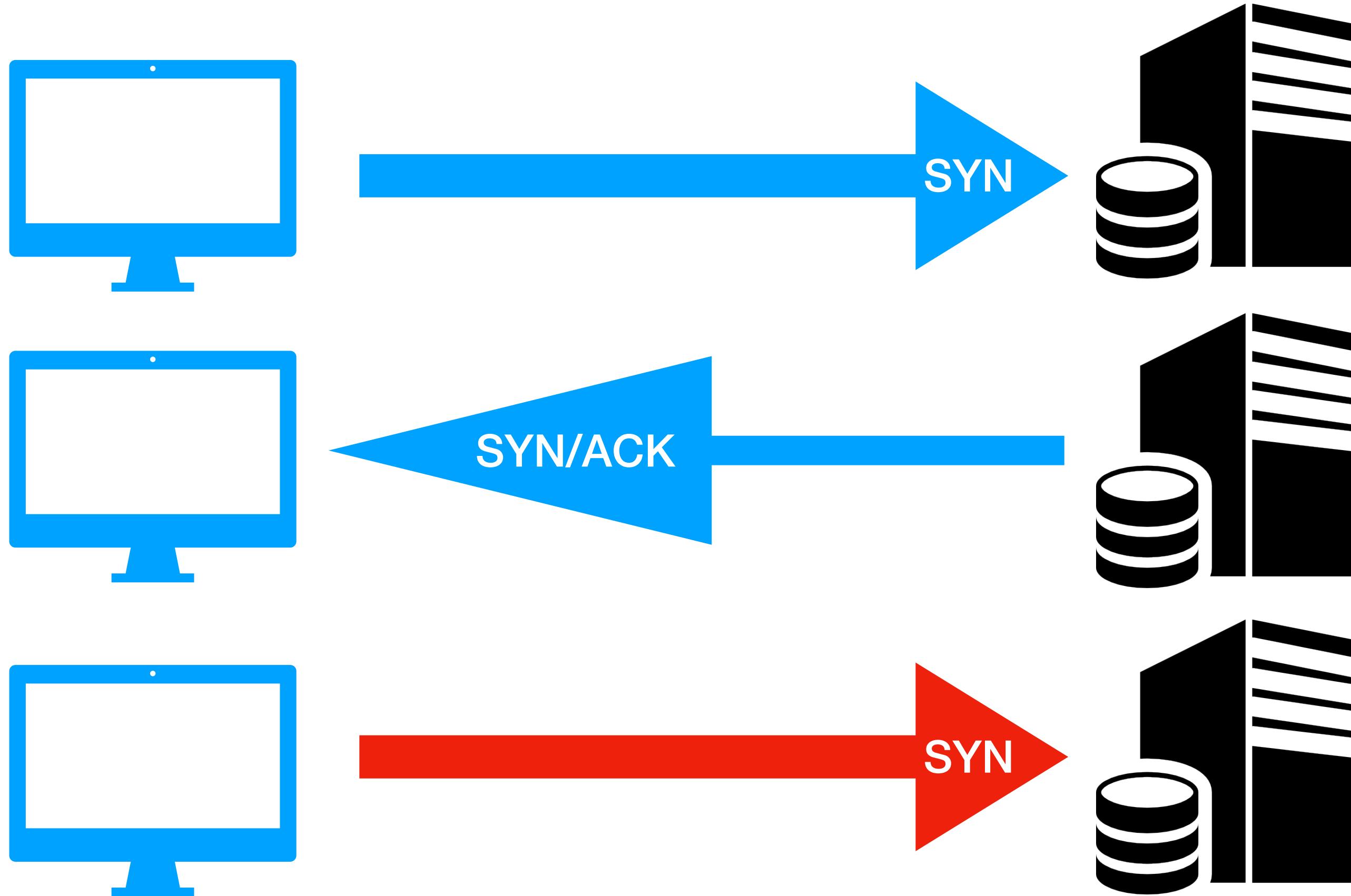


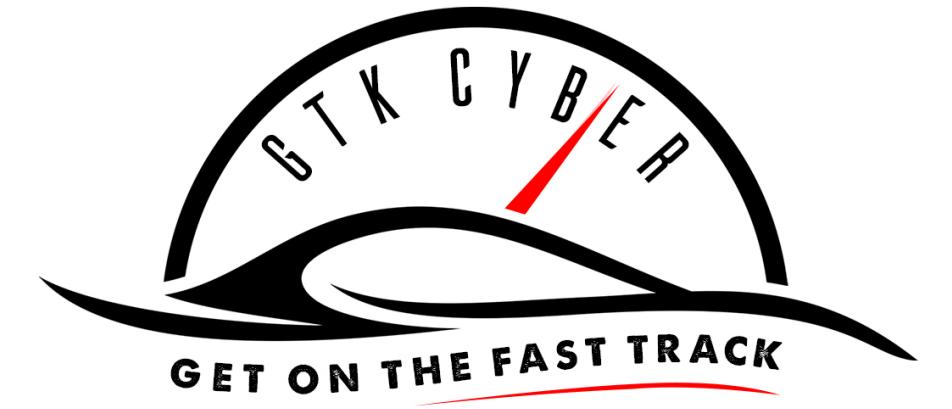
SynFlood Detection





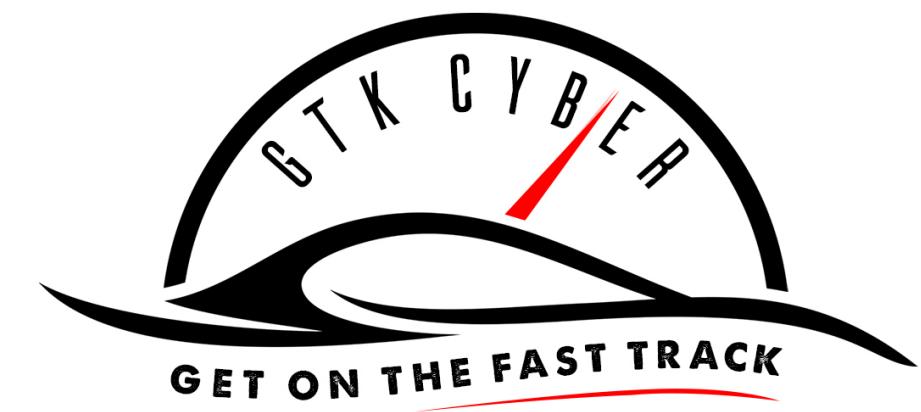
SynFlood Detection





**Drill can automate SYN Flood
detection with a custom UDF.**

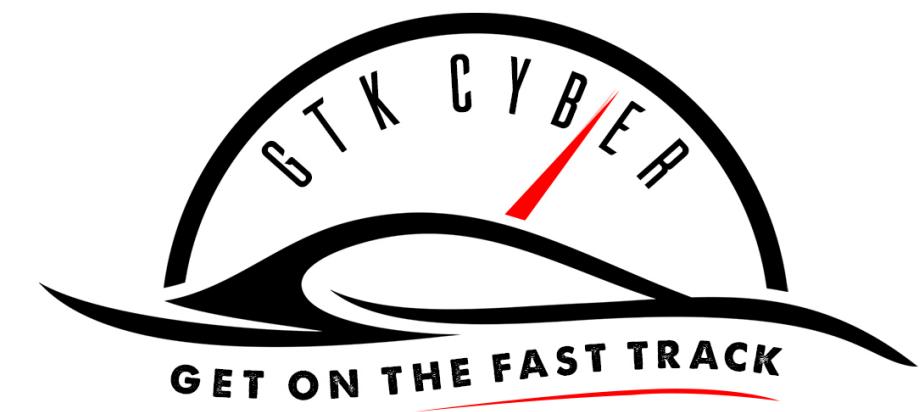
<https://github.com/cgivre/drill-synflood-udf>



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)
```

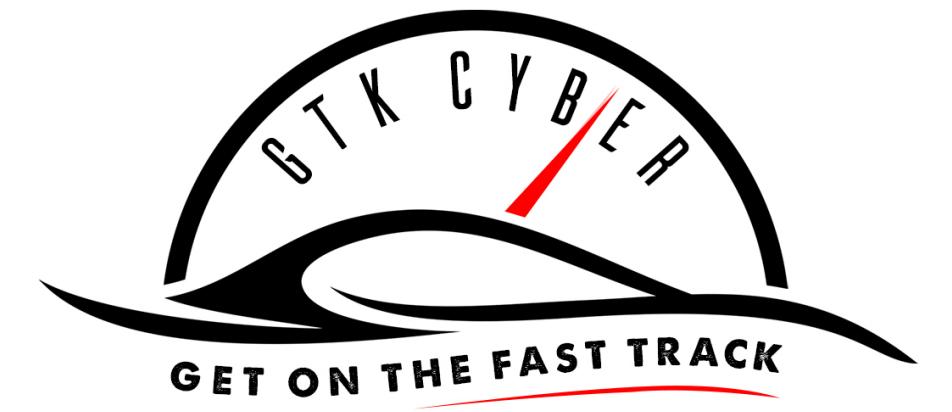
tcp_session
6346604732028469374
-9031405983396365775
7738739733723725373



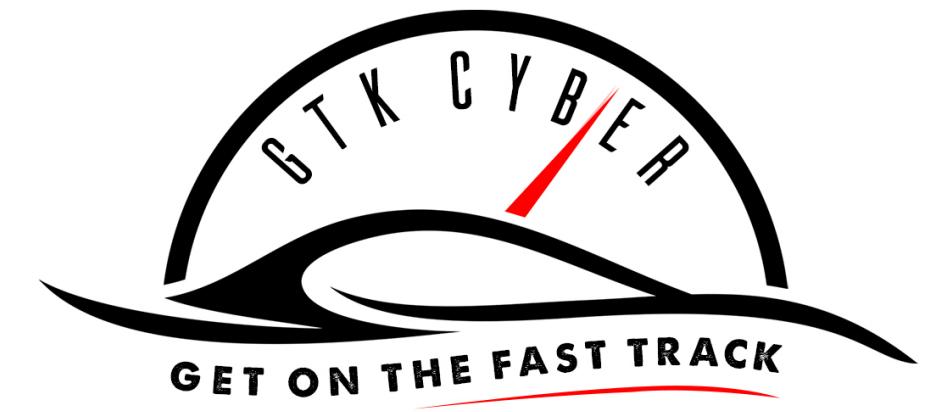
SYN Flood Detection

```
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)
)
```

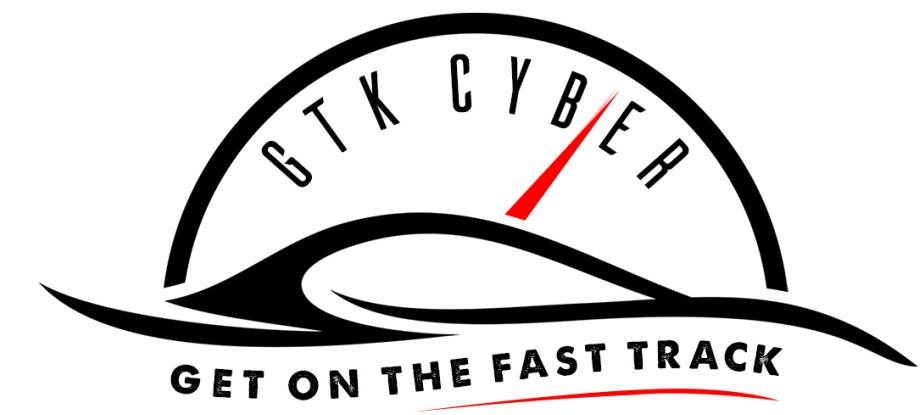
type	network	timestamp	timestamp_micro	src_ip	dst_ip	src_port	dst_port	src_mac_address	dst_mac_address	tcp_session
TCP	1	2010-07-04T20:24:16.276000	1278275056276783	172.16.0.8	64.13.134.52	36050	53	00:25:B3:BF:91:EE	00:26:0B:31:07:33	6346604732028469374
TCP	1	2010-07-04T20:24:16.338000	1278275056338667	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?



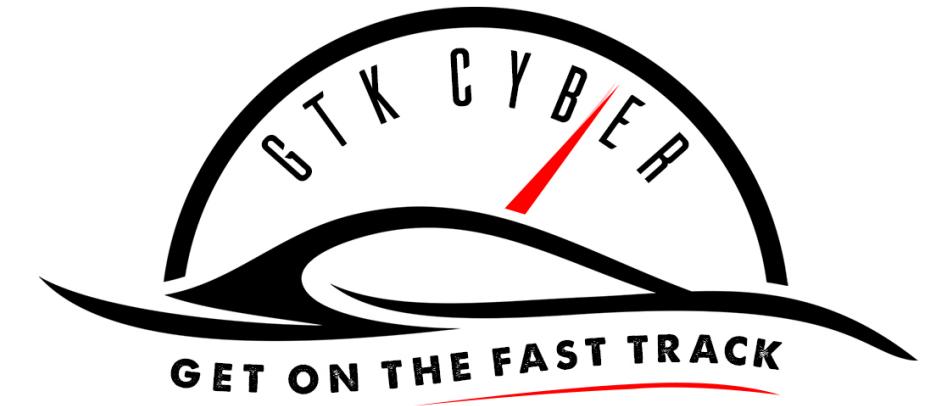
How about log files?



```
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
```

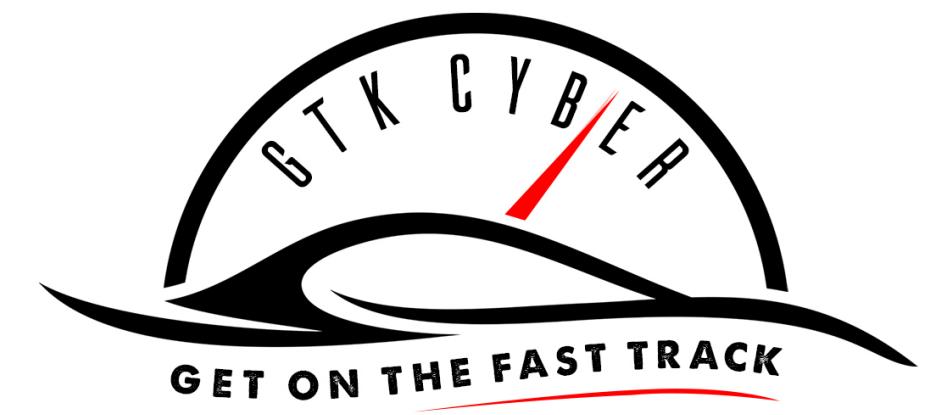


eventDate	process_name	pid	message	src_ip
2018-12-12T11:50:25	sshd	36669	Failed password for root from 61.160.251.136 port 1313 ssh2	61.160.251.136
2018-12-12T11:50:25	sshd	36669	Failed password for root from 61.160.251.136 port 1313 ssh2	61.160.251.136
2018-12-12T11:50:24	sshd	36669	Failed password 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

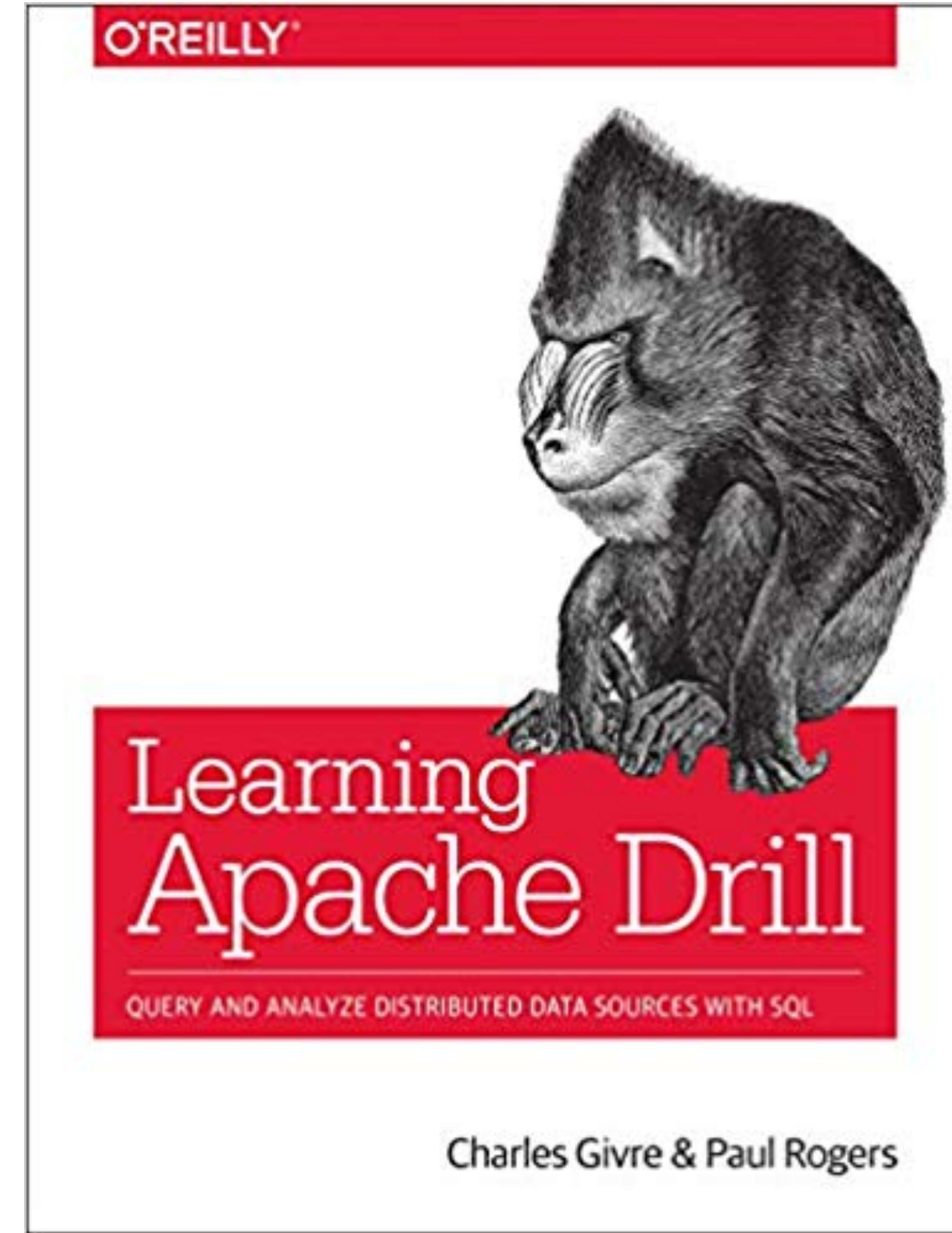
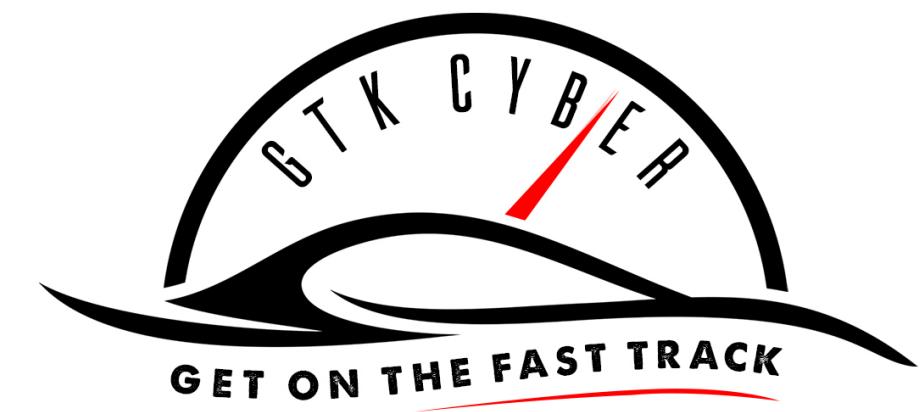


**Drill can natively query Syslog
formatted data***

*** This feature will be available in Drill version 1.16**



Questions?

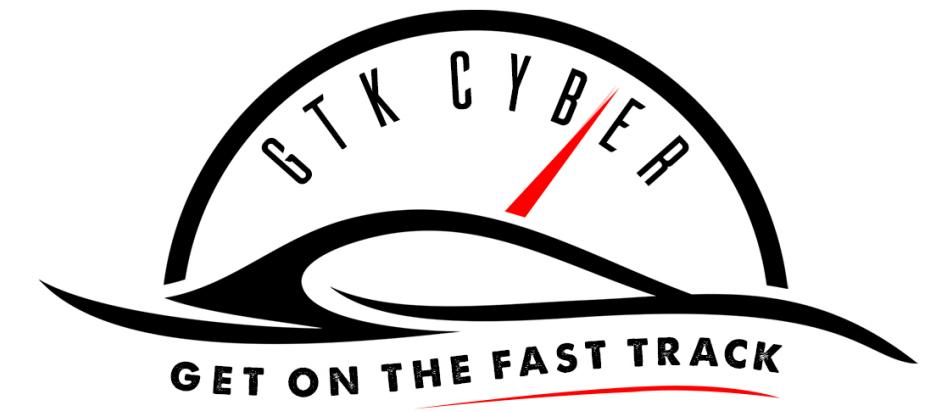


<https://amzn.to/2HDwE92>



“Any sufficiently advanced
technology is
indistinguishable from magic”

—Arthur C. Clarke

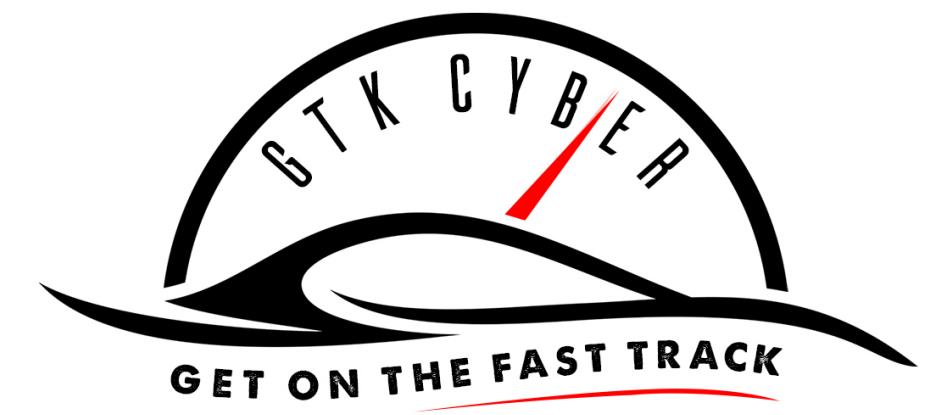


Why **aren't** you using Drill?

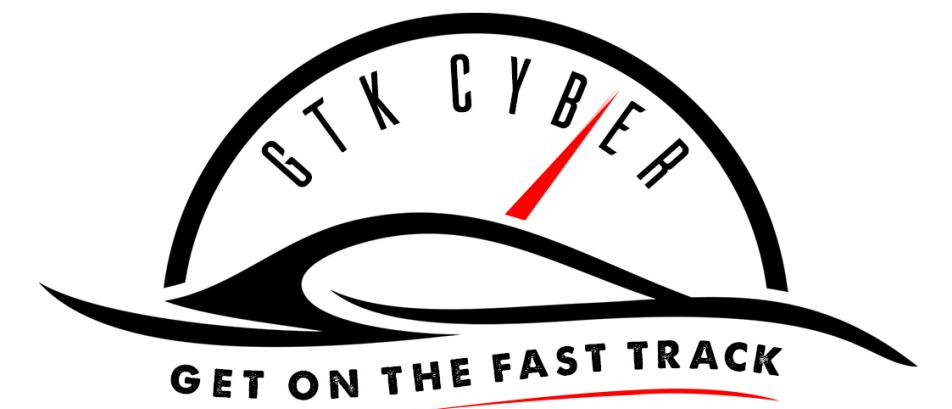
WARNING
DANGER
DO NOT PULL
HANDLE

MARTIN BAKER MARK II SEAT
EJECTION SEAT CAPACITY
8 G/15 MIN ON ROLLBACK

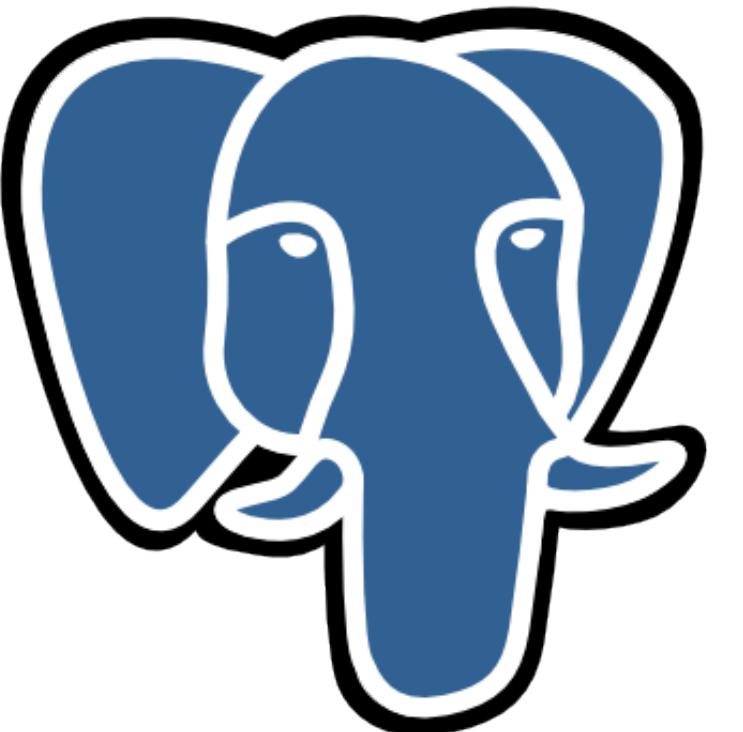




Installing & Configuring Drill



Embedded

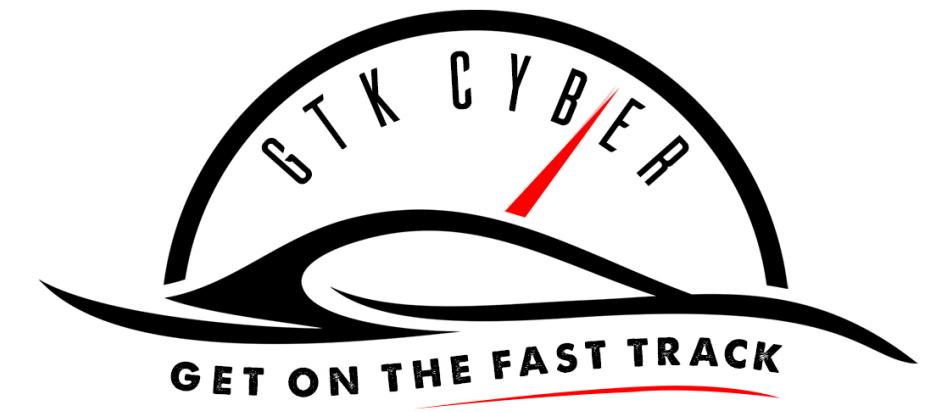


ORACLE®

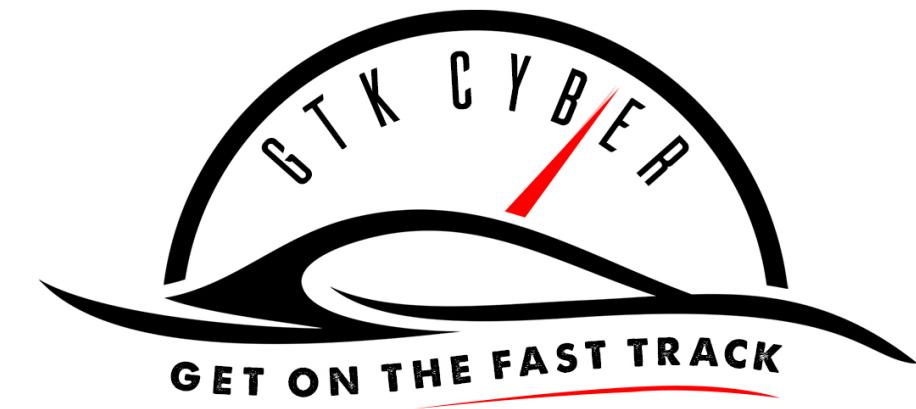


Distributed



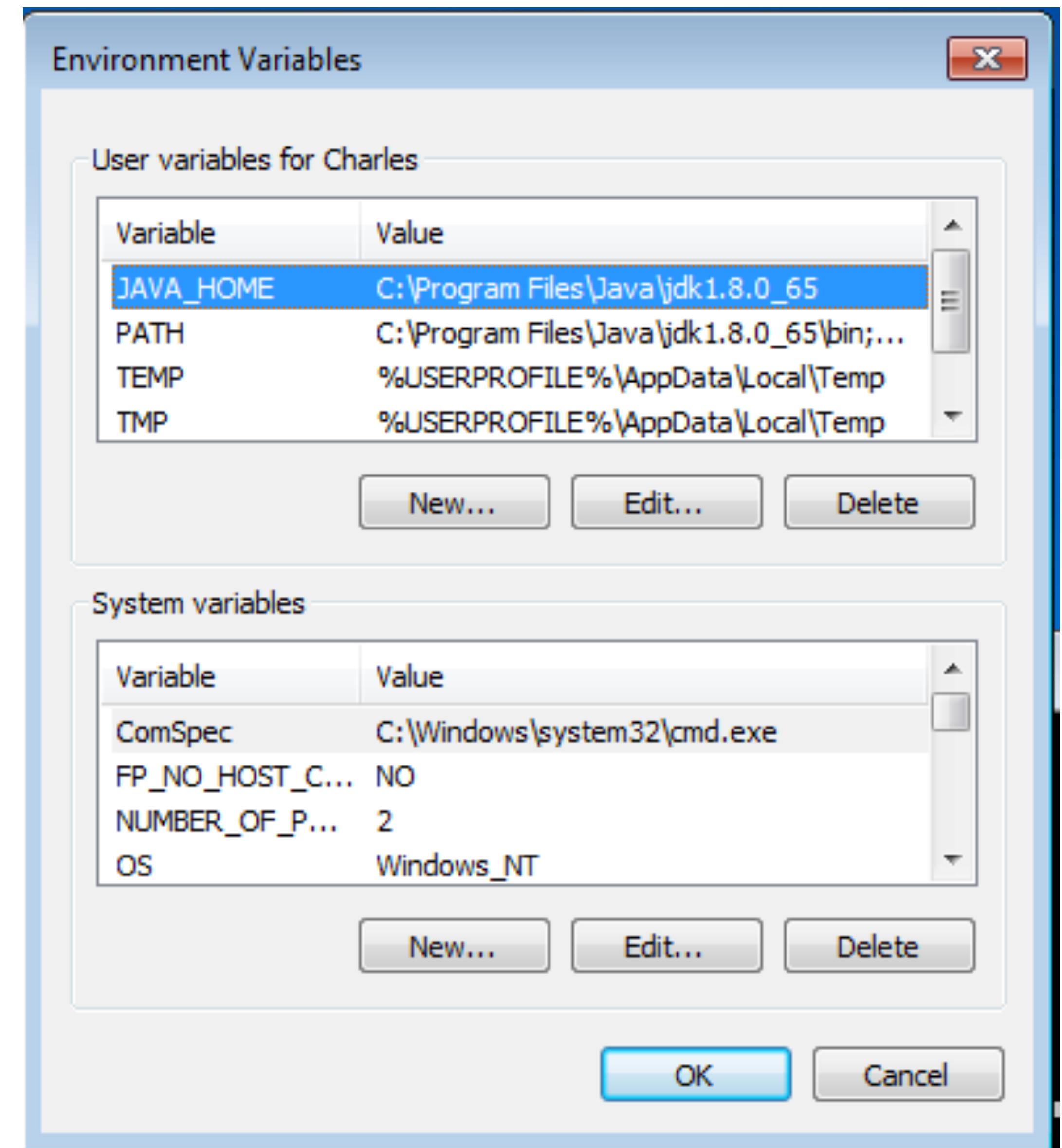
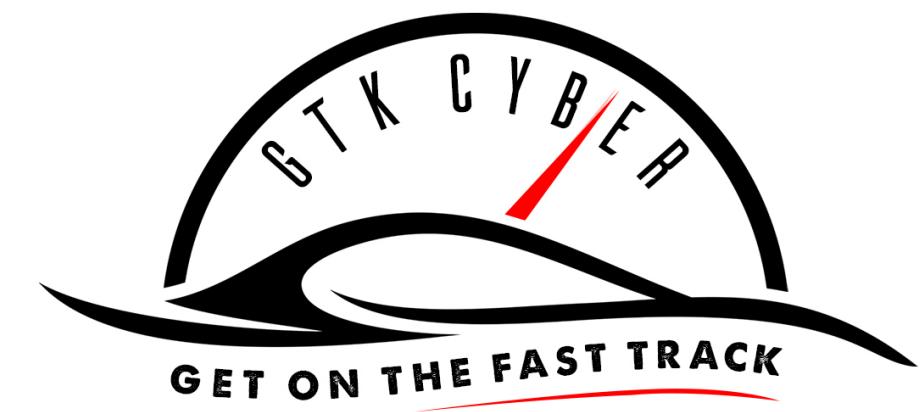


Step 1: Download Drill:
drill.apache.org/download/



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.





Starting Drill

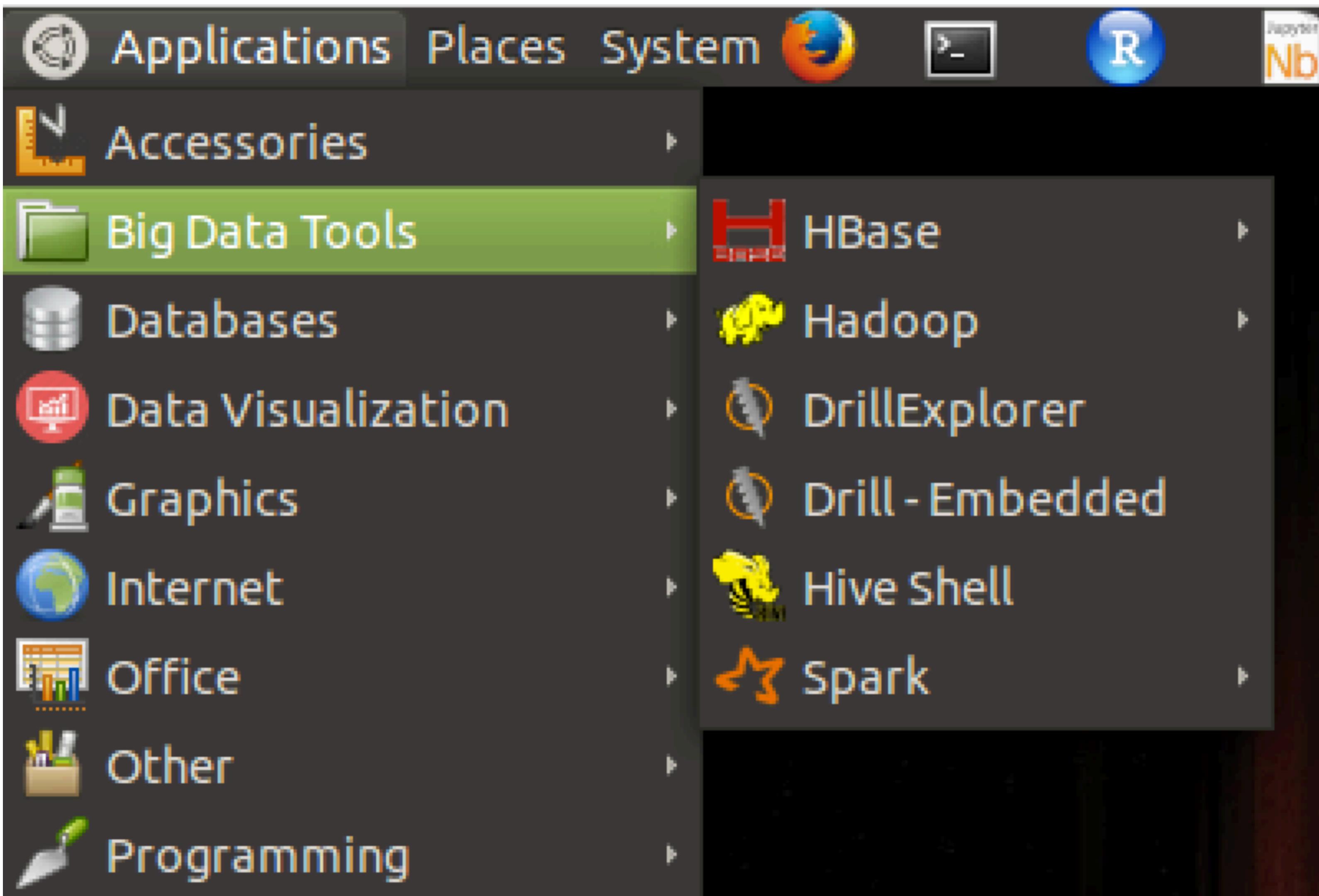
Embedded Mode: For use on a standalone system

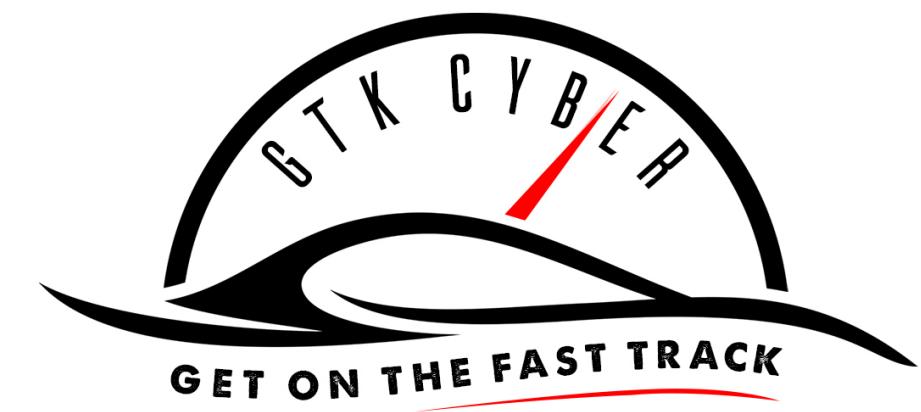
\$./bin/drill-embedded  

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

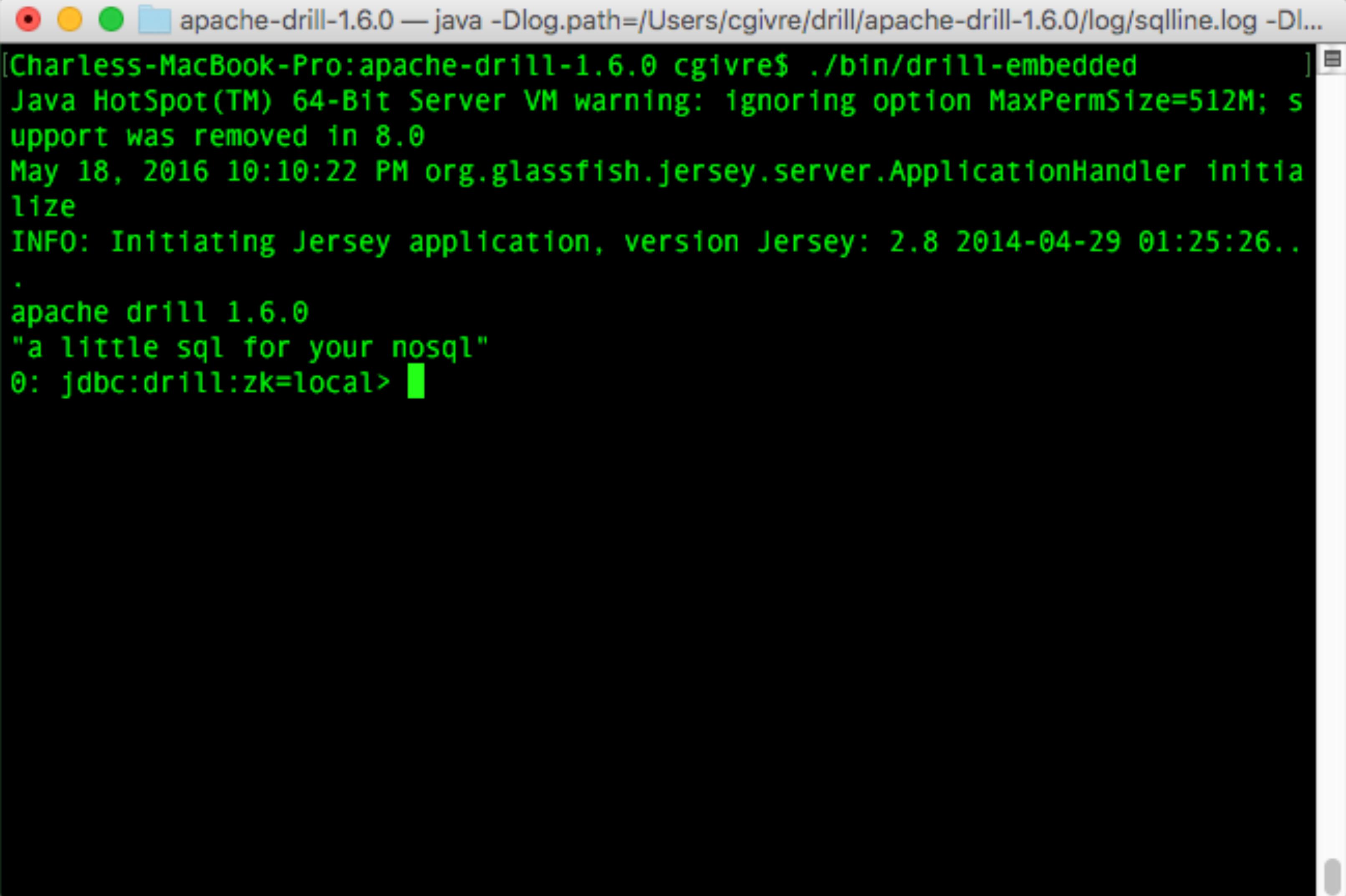


Starting Drill



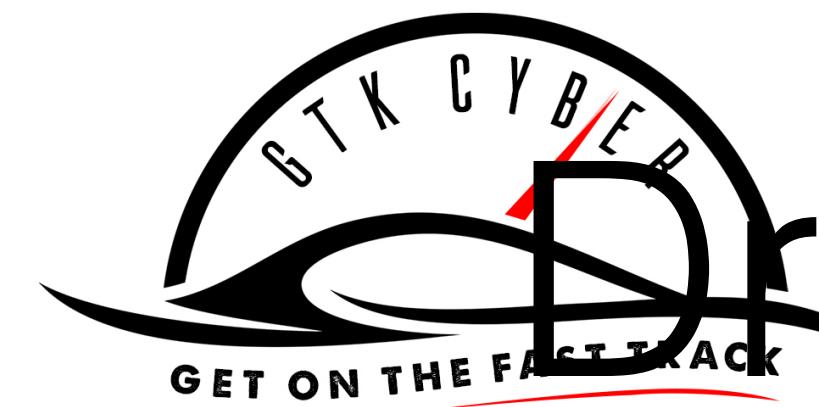


Drill's Command Line Interface



The screenshot shows a terminal window on a Mac OS X system. The title bar reads "apache-drill-1.6.0 — java -Dlog.path=/Users/cgivre/drill/apache-drill-1.6.0/log/sqlline.log -Dl...". The terminal output is as follows:

```
[Charless-MacBook-Pro:apache-drill-1.6.0 cgivre$ ./bin/drill-embedded
Java HotSpot(TM) 64-Bit Server VM warning: ignoring option MaxPermSize=512M; support was removed in 8.0
May 18, 2016 10:10:22 PM org.glassfish.jersey.server.ApplicationHandler initialize
INFO: Initiating Jersey application, version Jersey: 2.8 2014-04-29 01:25:26..
.
apache drill 1.6.0
"a little sql for your nosql"
0: jdbc:drill:zk=local> ]
```

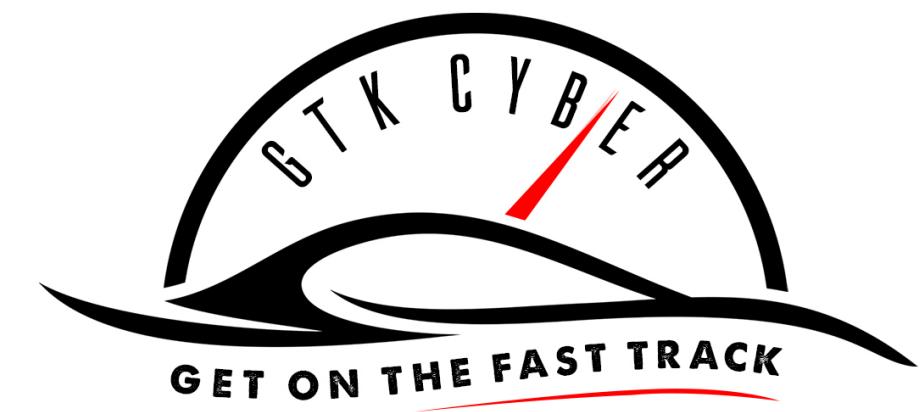


Drill's Command Line Interface

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

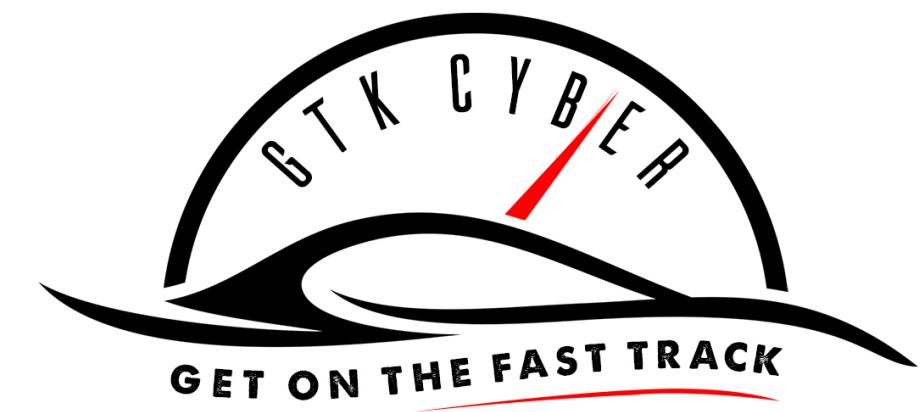
```
apache-drill-1.6.0 — java -Dlog.path=/Users/cgivre/drill/apache-drill-1.6.0/log/sqlline.log -Dlog.query.path=...
```

```
0: jdbc:drill:zk=local>
0: jdbc:drill:zk=local>
0: jdbc:drill:zk=local>
0: jdbc:drill:zk=local>
0: jdbc:drill:zk=local>
0: jdbc:drill:zk=local>
0: jdbc:drill:zk=local> SELECT DISTINCT management_role  FROM cp.`employee.json`;
+-----+
|   management_role   |
+-----+
| Senior Management |
| Store Management   |
| Middle Management |
| Store Full Time Sta|
| Store Temp Staff  |
+-----+
```



<http://localhost:8047>

The screenshot shows the Apache Drill web interface running at `http://localhost:8047`. The interface has a top navigation bar with tabs for Apache Drill, Query, Profiles, Storage, Metrics, Threads, Options, and Documentation. A sample SQL query is displayed: `SELECT * FROM cp.`employee.json` LIMIT 20`. Below this, there is a section for "Query Type" with radio buttons for SQL (selected), PHYSICAL, and LOGICAL. A large input field for the "Query" is present, containing a single vertical bar character. A "Submit" button is located at the bottom left of the input field.



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

Screenshot of the Apache Drill web interface showing a sample SQL query and a query editor.

The interface includes a top navigation bar with tabs: Apache Drill, Query, Profiles, Storage, Metrics, Threads, Options, and Documentation. The URL in the address bar is localhost.

A sample SQL query is displayed in a blue box:

```
Sample SQL query: SELECT * FROM cp.`employee.json` LIMIT 20
```

The Query Type section shows that SQL is selected (radio button is checked). Other options are PHYSICAL and LOGICAL.

The main area contains a large text input field for the query, which currently contains a single vertical bar character. A "Submit" button is located at the bottom left of the input field.

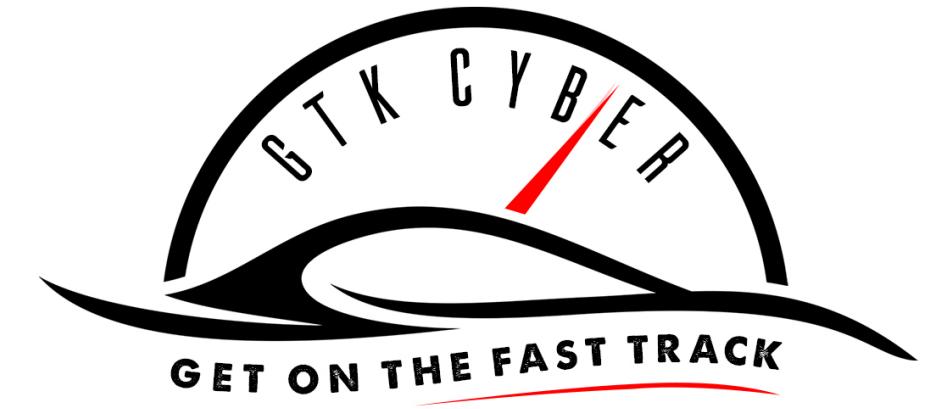


```
SELECT * FROM cp.`employee.json` LIMIT 20
```

Screenshot of the Apache Drill web interface showing the results of the query.

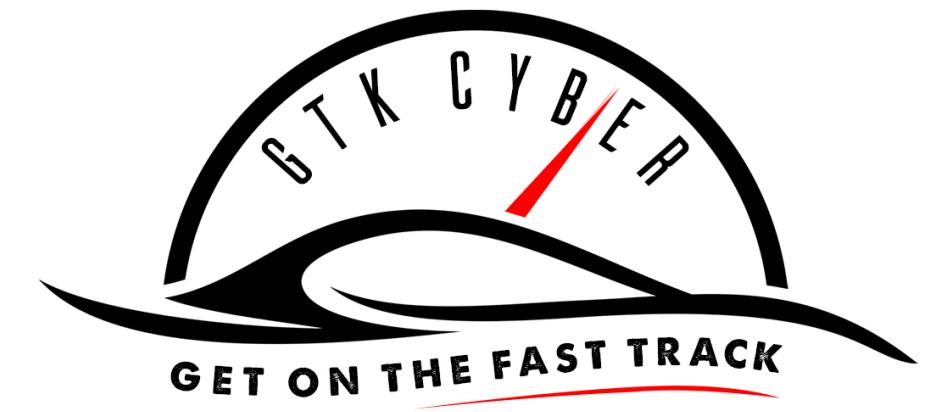
The interface includes a navigation bar with links for Apache Drill, Query, Profiles, Storage, Metrics, Threads, Options, and Documentation. The main area displays a table with employee data, with the first five rows visible.

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 Whelby	Derrick	Whelby	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)

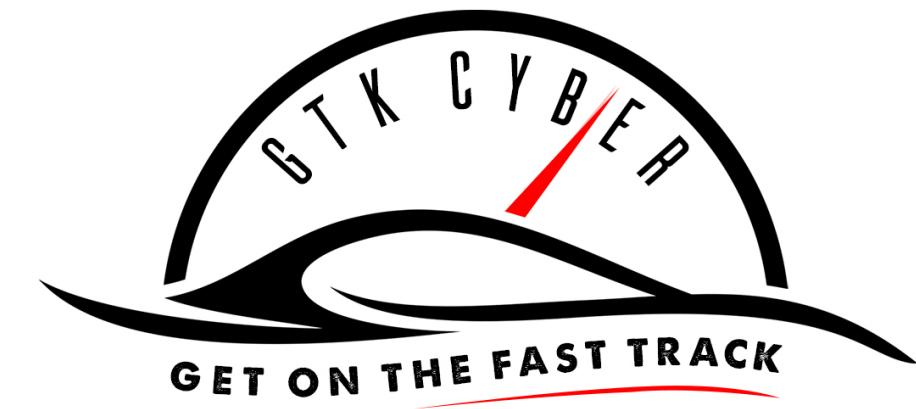


Workspaces in Drill

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

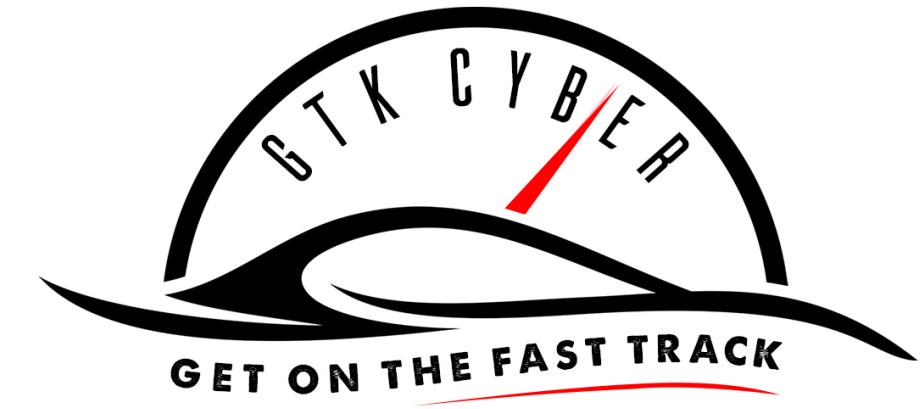
or

`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.



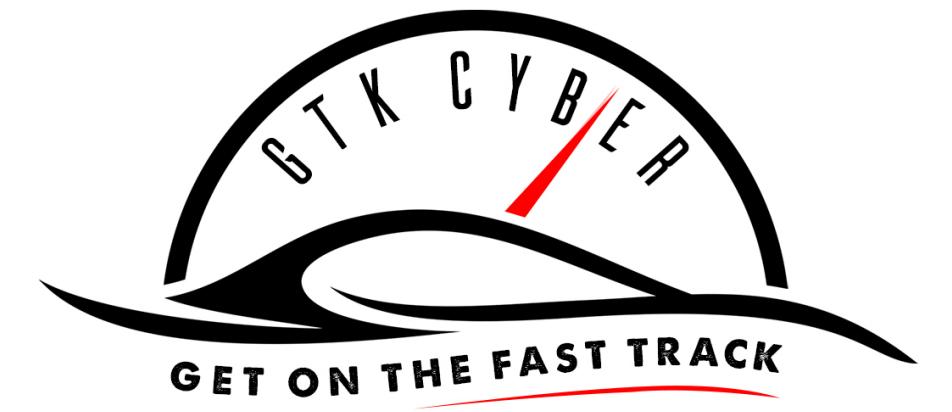
In Class Exercise: Create a Workspace

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

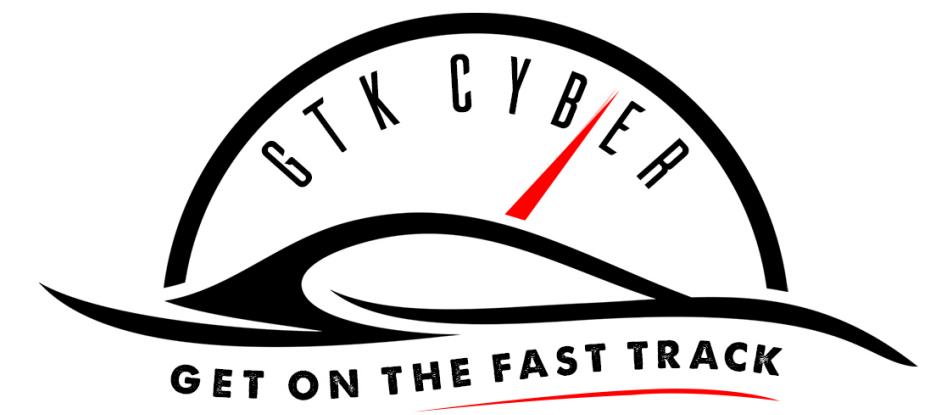
1. Open the Drill Web UI and go to Storage->dfs->update
2. Paste the following into the 'workspaces' section and click update

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

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

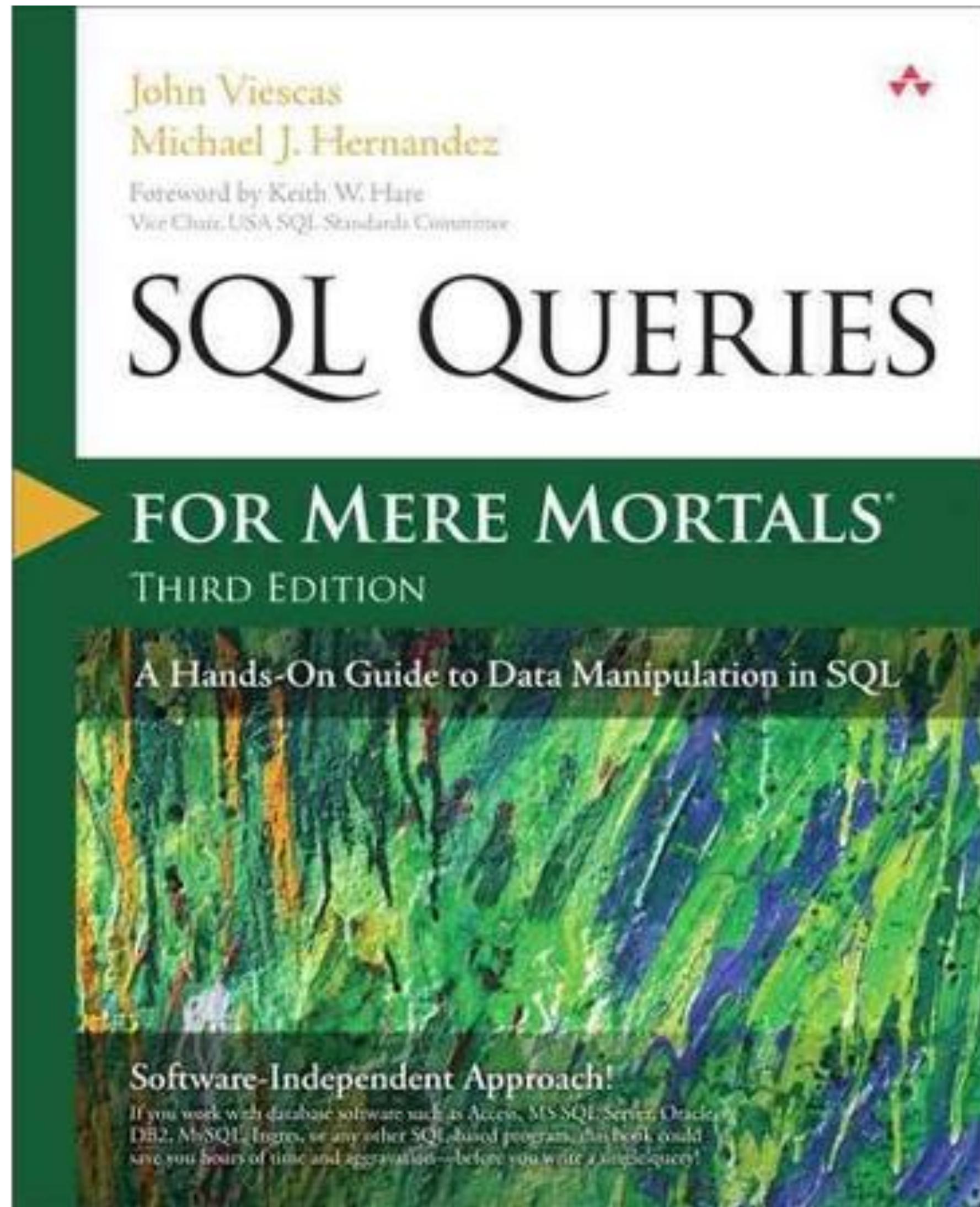
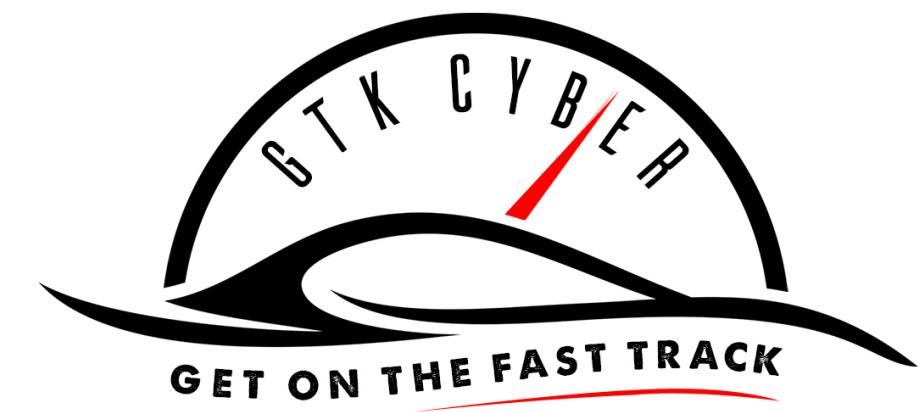


Querying Simple Delimited Data

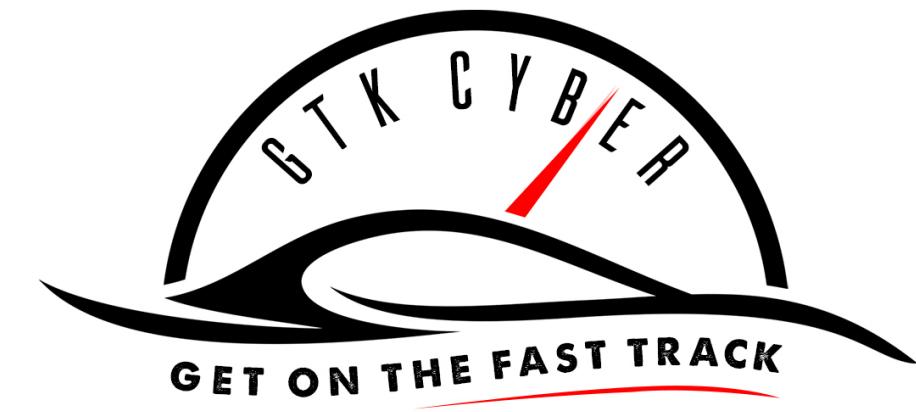


Everything you need to know about SQL*... in 10 minutes

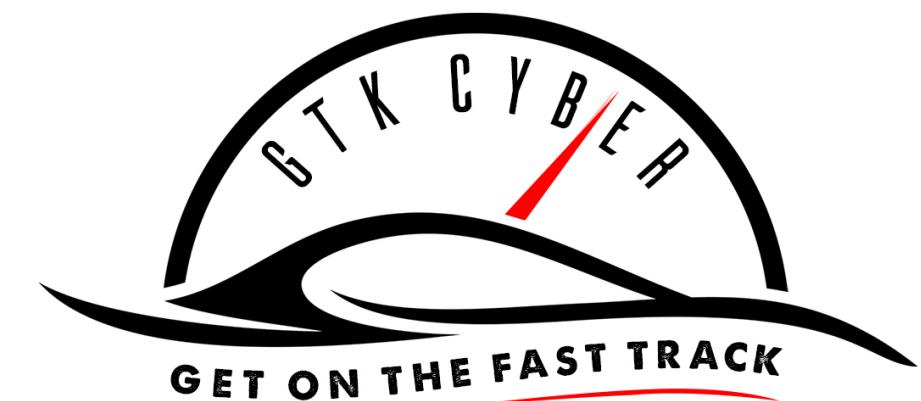
* well...not quite everything, but enough to get you through this session



<http://amzn.to/2IID8yi>

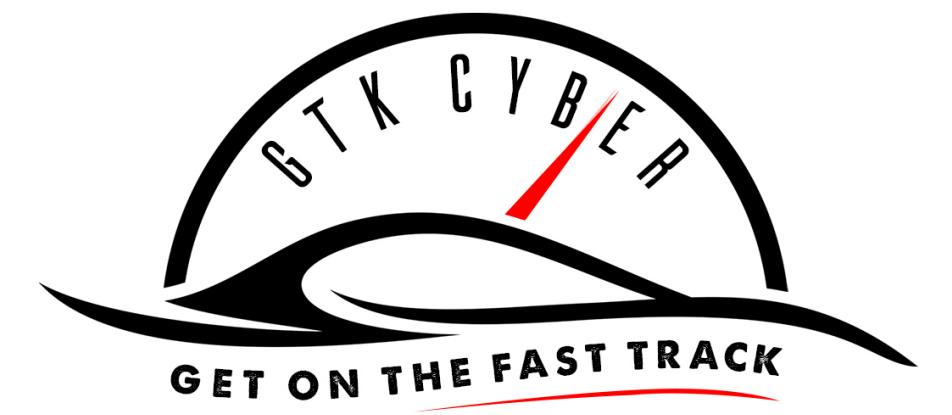


```
SELECT <fields>
FROM <data source>
```

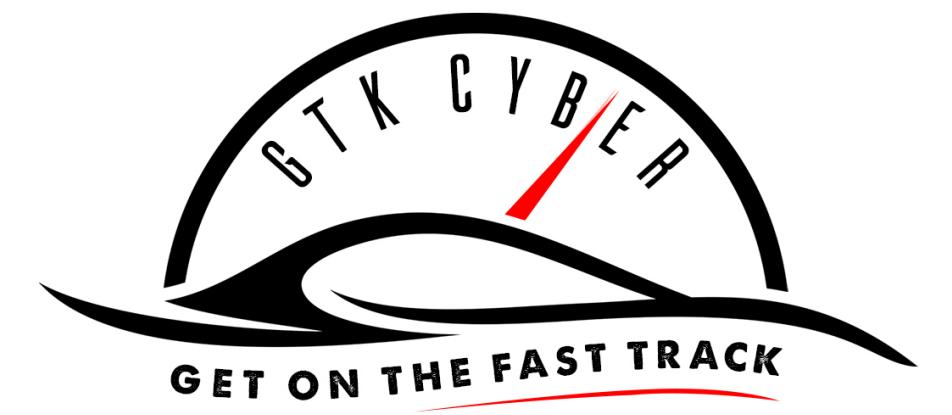


Please open people.csv

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



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

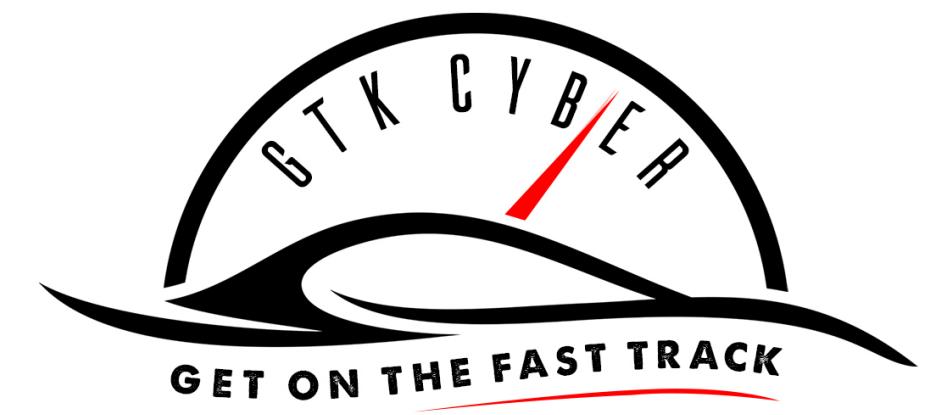


```
SELECT first_name,  
       last_name,  
       gender  
  FROM <data source>
```

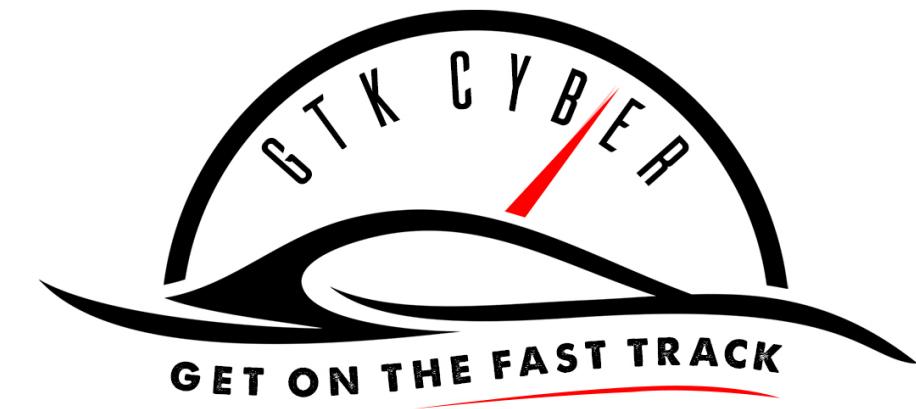


Tip: Use BACK TICKS around field
names in Drill

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



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



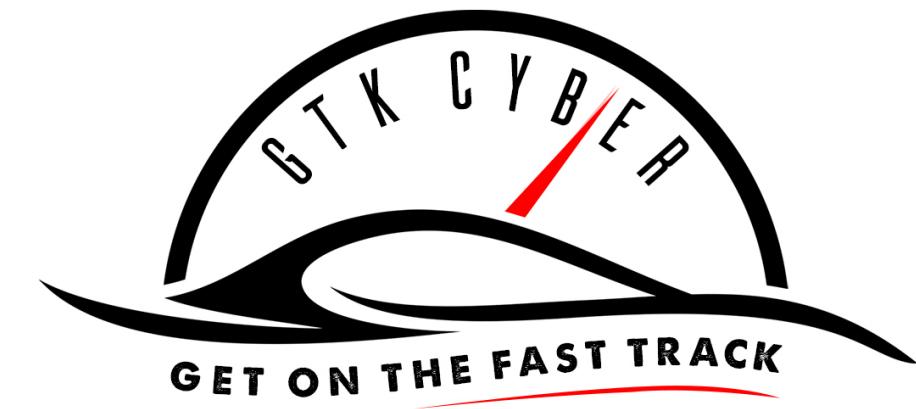
Querying Drill

```
FROM dfs.logs.`/data/customers.csv`
```

Storage Plugin

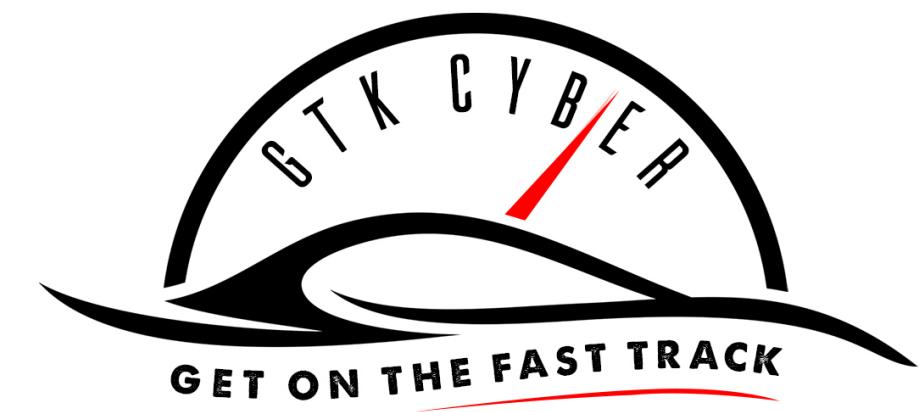
Workspace

Table



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

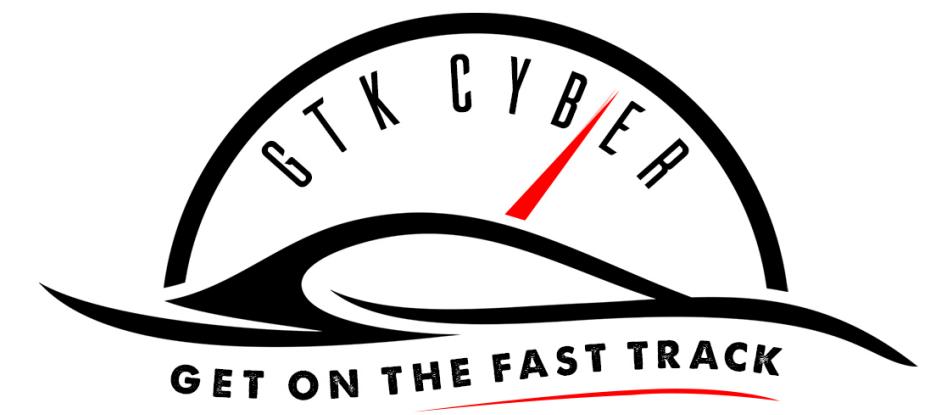
Try it yourself!!



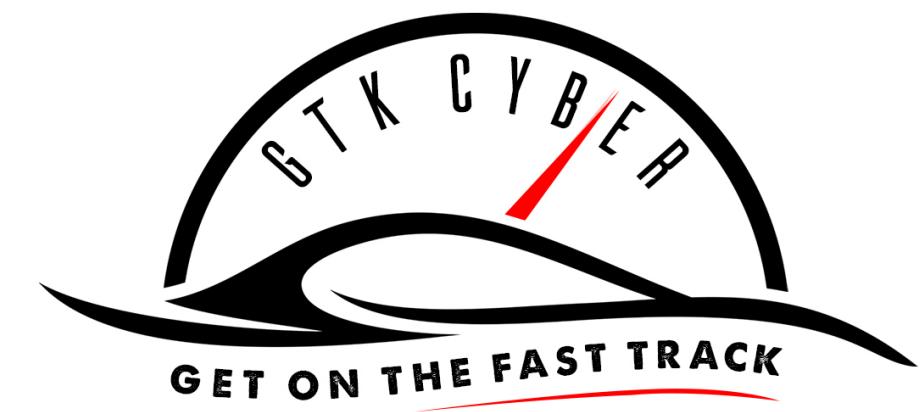
```
SELECT `first_name`,  
`last_name`,  
`gender`  
FROM  
dfs.drillclass.`people.csvh`
```

Apache Drill Query Profiles Storage Metrics Threads Logs Options Documentation

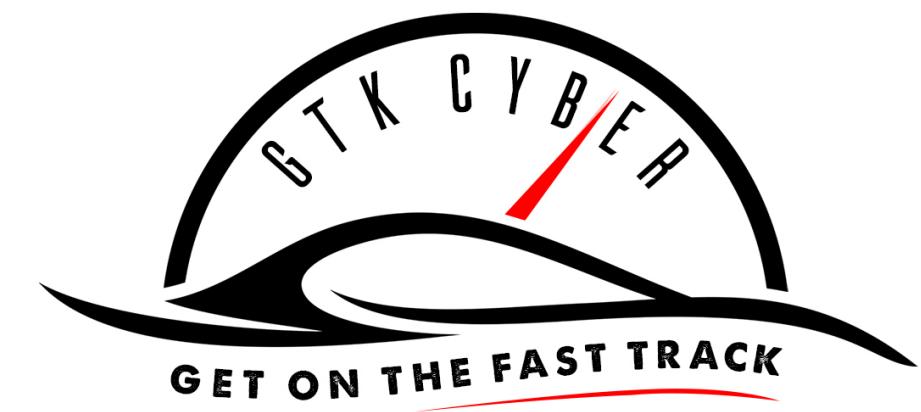
Show	10	entries	Search:	Show / hide columns
first_name		last_name		gender
Philip		Richardson		Male
Todd		James		Male
Jimmy		Mendoza		Male



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



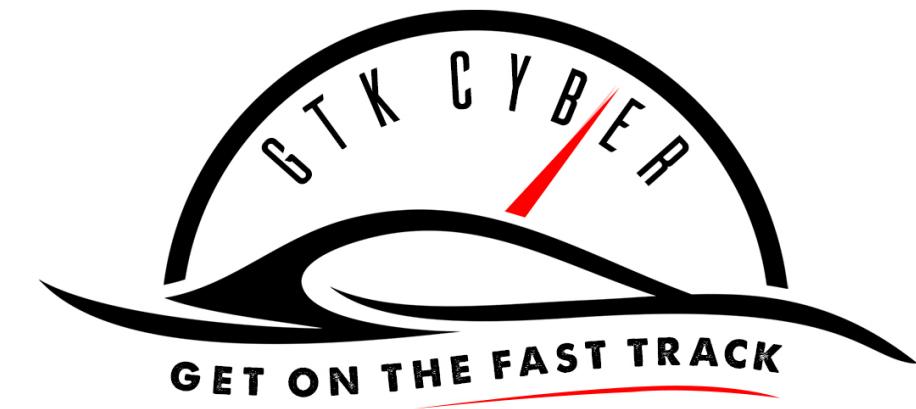
```
SELECT `first_name`,  
`last_name`,  
`gender`  
FROM dfs.drillclass.`people.csvh`  
WHERE `gender` = 'Female'
```



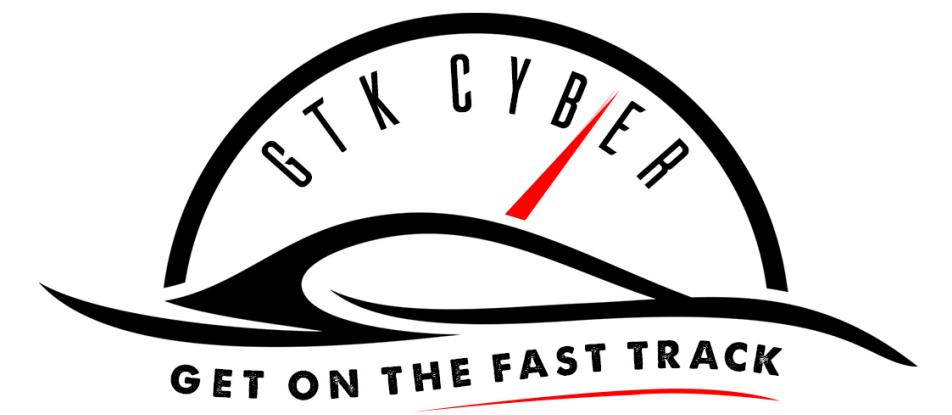
```
SELECT `first_name`,  
       `last_name`,  
       `gender`  
FROM dfs.drillclass.`people.csvh`  
WHERE `gender` = 'Female'
```



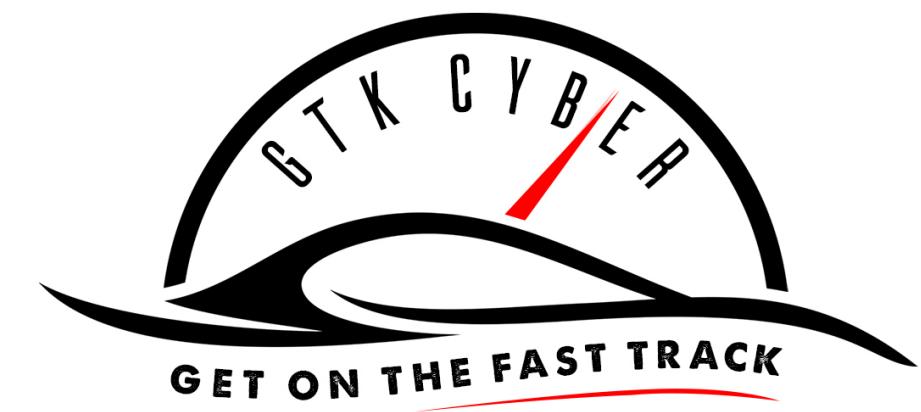
first_name	last_name	gender
Dorothy	Fernandez	Female
Frances	Rodriguez	Female
Nancy	Nelson	Female



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



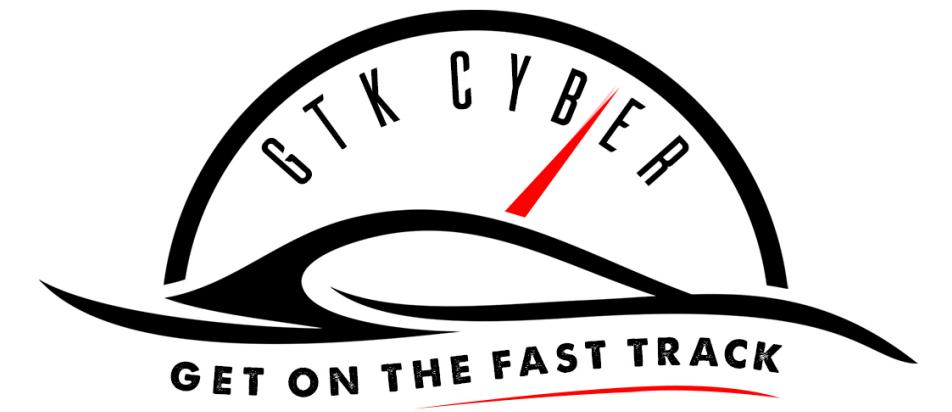
```
SELECT `first_name`,  
`last_name`,  
`gender`  
FROM dfs.drillclass.`people.csvh`  
ORDER BY `last_name`, `first_name` ASC
```



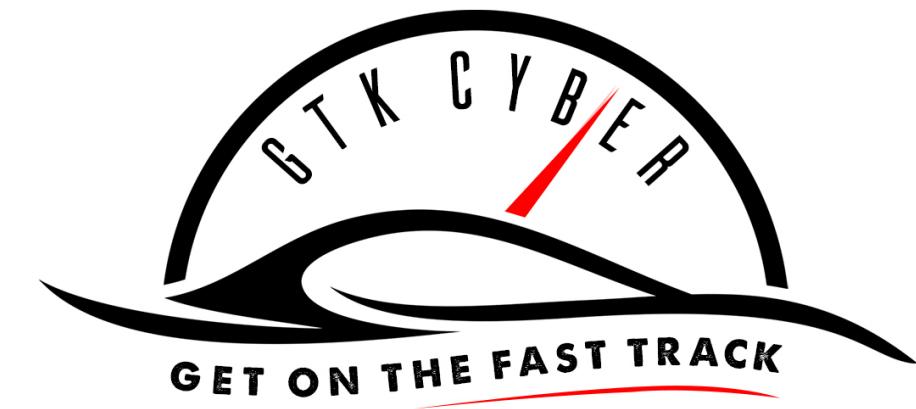
```
SELECT `first_name`,  
`last_name`,  
`gender`  
FROM dfs.drillclass.`people.csvh`  
ORDER BY `last_name`, `first_name` ASC
```

Apache Drill Query Profiles Storage Metrics Threads Logs Options Documentation

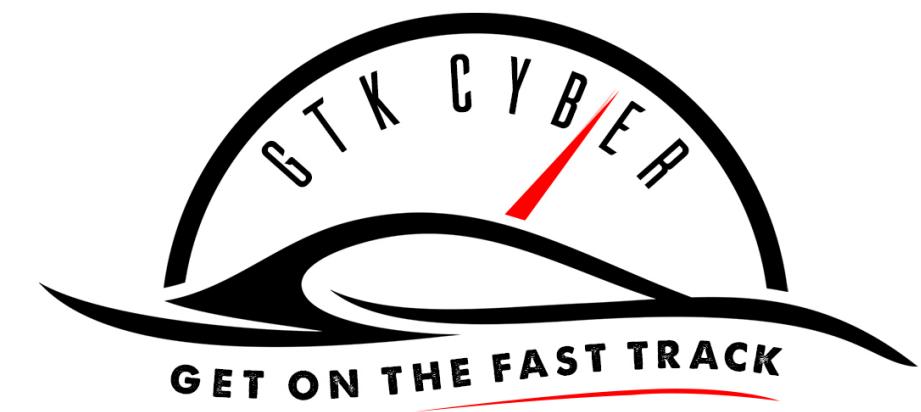
first_name	last_name	gender
Julia	Alvarez	Female
Nicholas	Bishop	Male
Marie	Bradley	Female



```
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
```

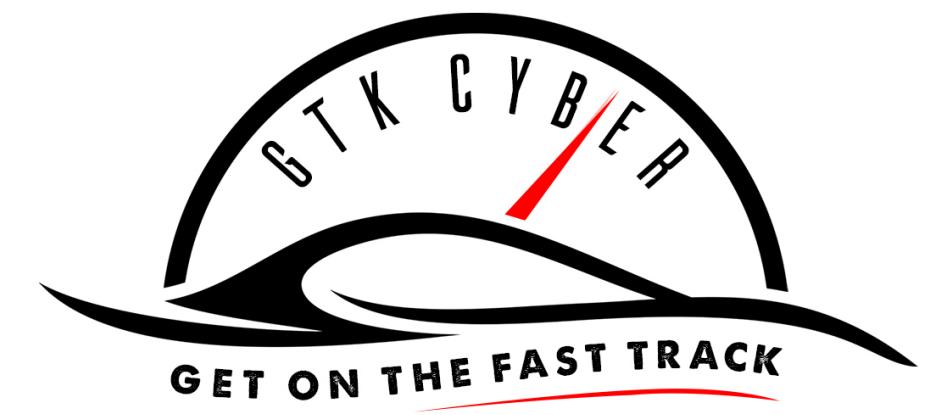


```
SELECT first_name,  
       LENGTH(`first_name`) AS fname_length  
  FROM dfs.drillclass.`people.csvh`  
 ORDER BY fname_length DESC
```

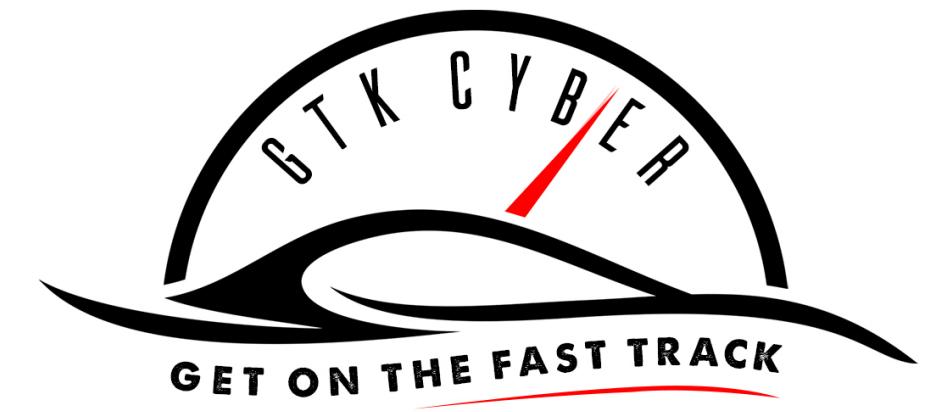
Apache Drill Query Profiles Storage Metrics Threads Logs Options Documentation

Show 10 entries Search: Show / hide columns

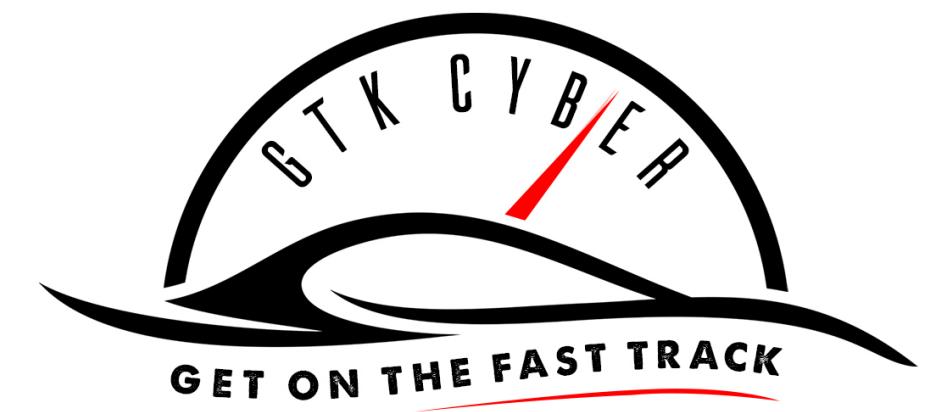
first_name	fname_length
Clarence	8
Nicholas	8
Theresa	7
Frances	7



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



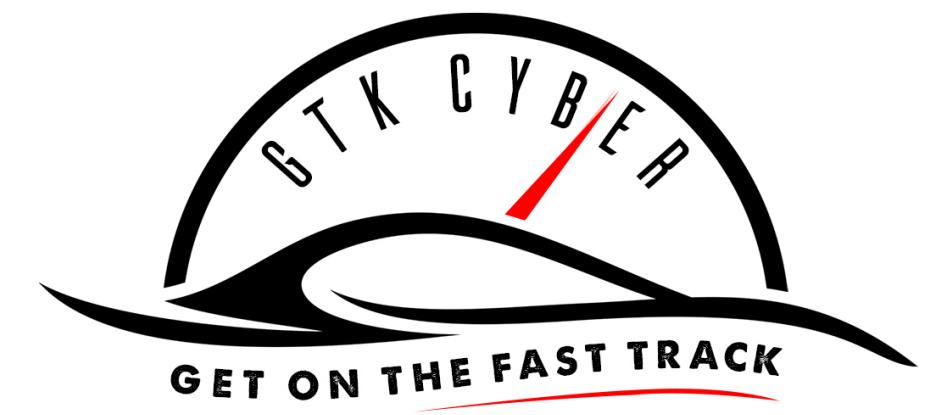
```
SELECT `gender`,  
COUNT( * ) AS gender_count  
FROM dfs.drillclass.`people.csvh`  
GROUP BY `gender`
```



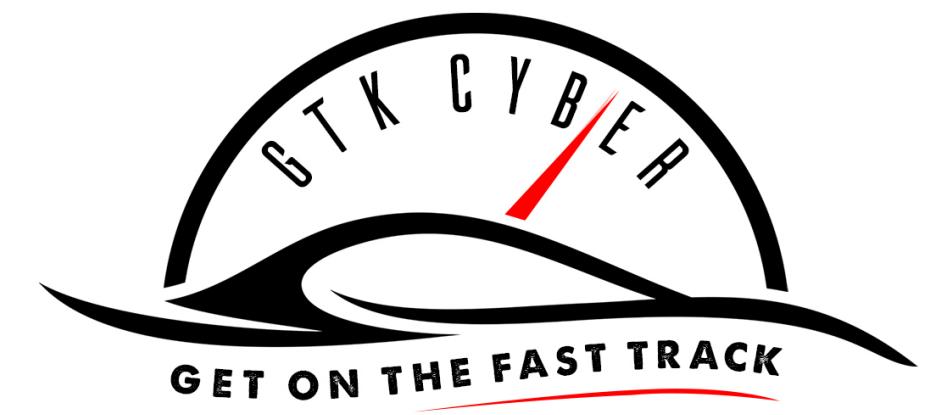
```
SELECT `gender`,  
COUNT( * ) AS gender_count  
FROM dfs.drillclass.`people.csvh`  
GROUP BY `gender`
```

Apache Drill Query Profiles Storage Metrics Threads Logs Options Documentation

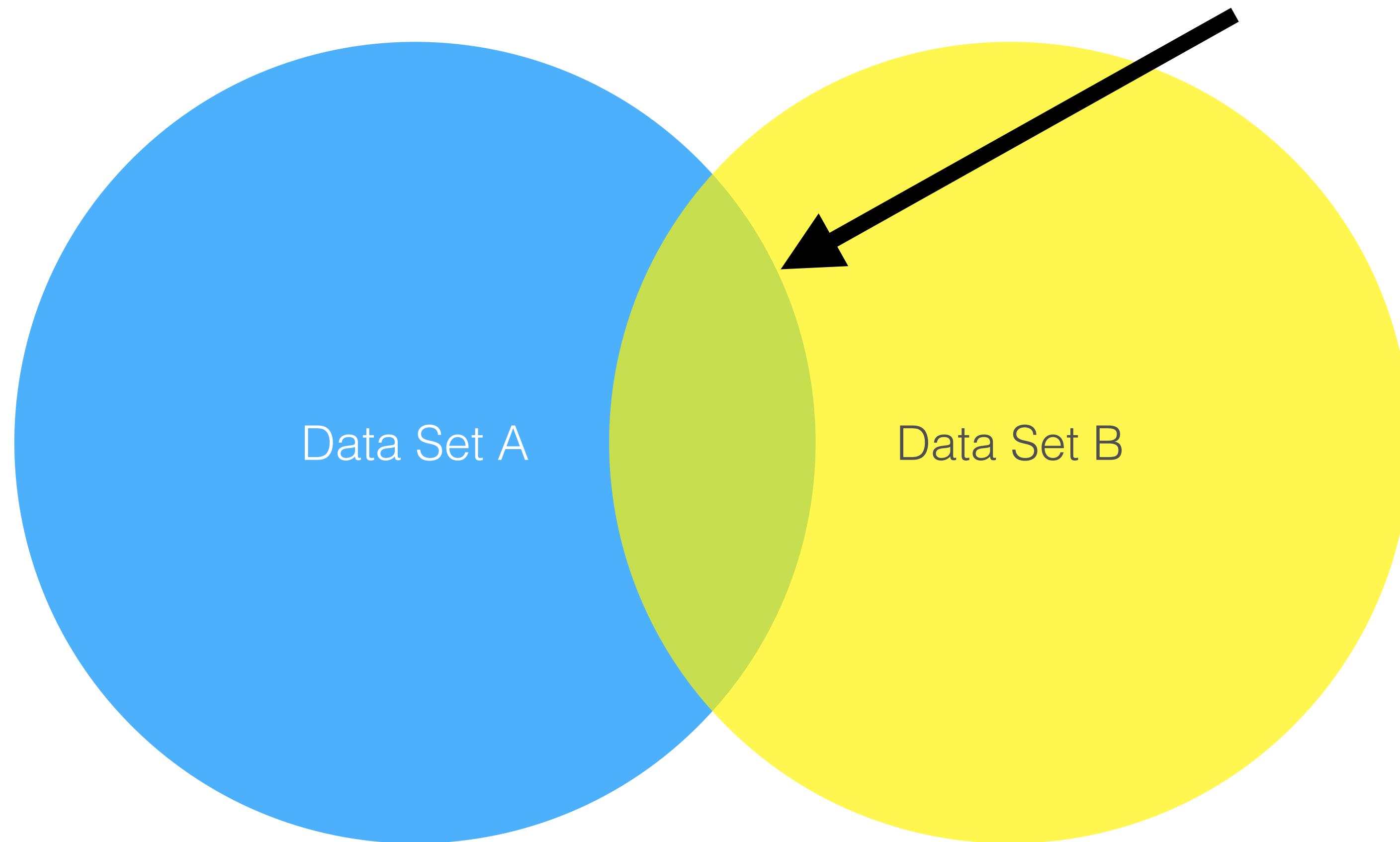
Show	10	entries	Search:	Show / hide columns
gender		gender_count		
Male		11		
Female		9		

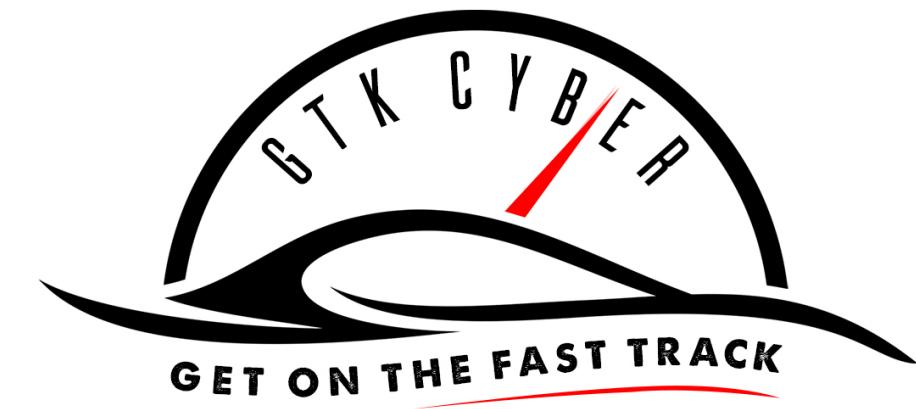


Joining Datasets



Referred to as an Inner Join

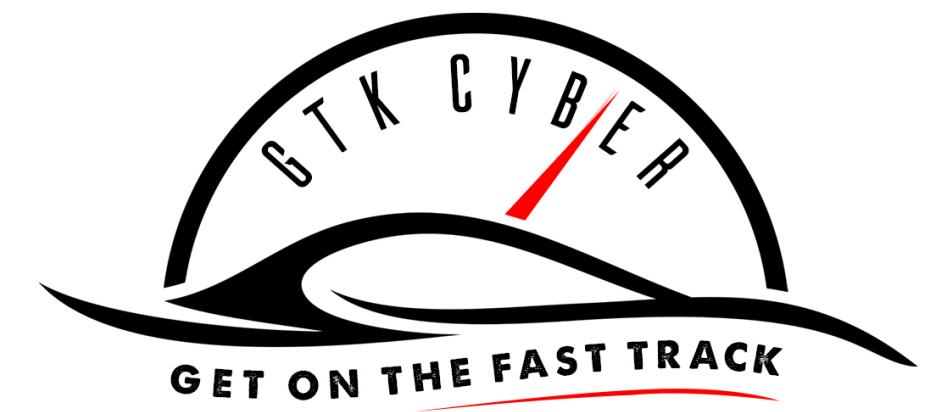




```
SELECT <fields>
FROM <data source 1> AS table1
INNER JOIN <data source 2> AS table2
ON table1.`id` = table2.`id`
```

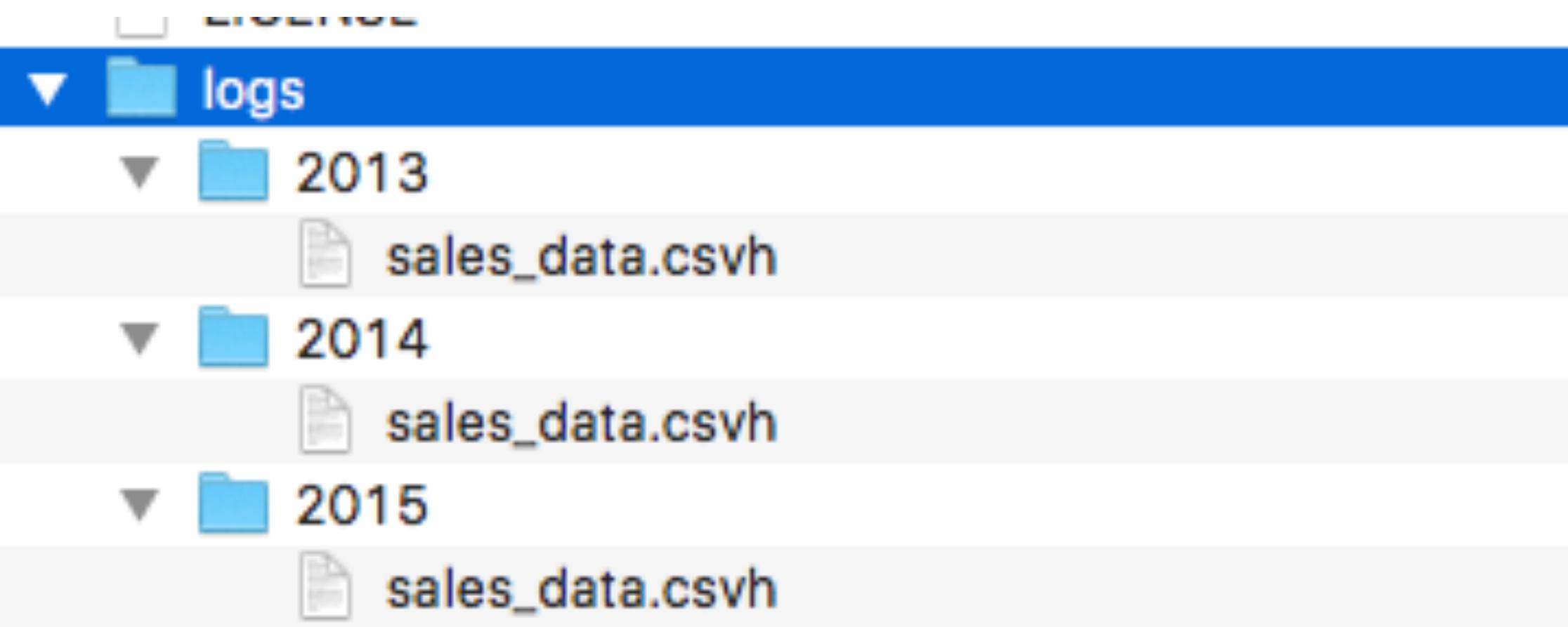


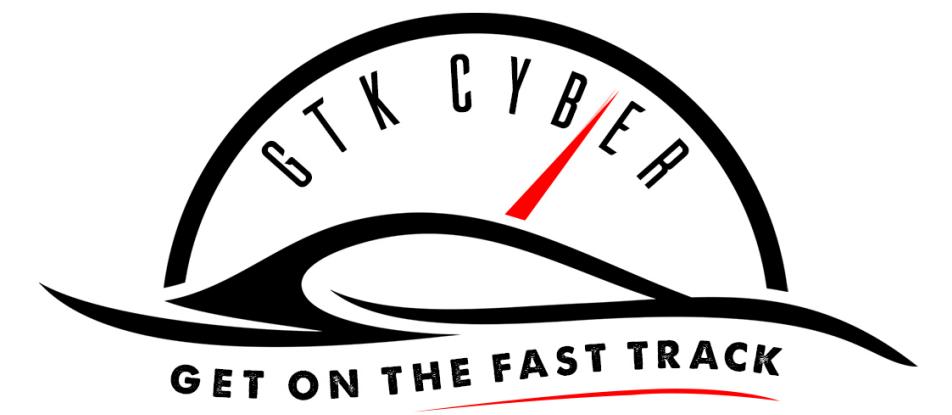
Problem: You have multiple log files
which you would like to analyze



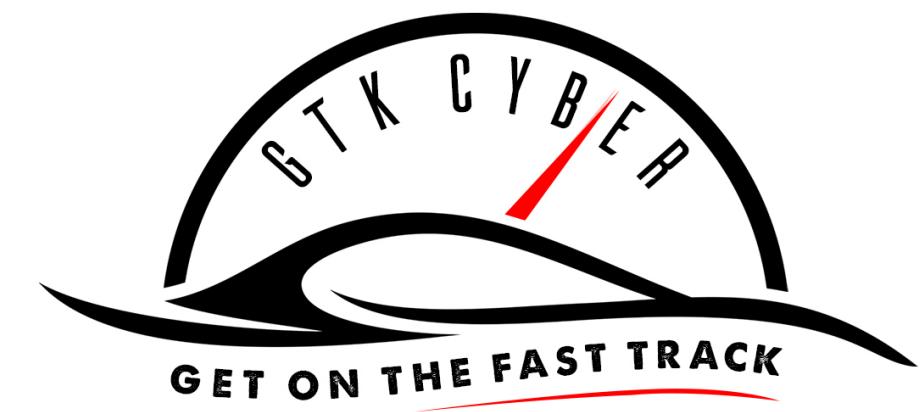
Problem: You have multiple log files which you would like to analyze

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





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



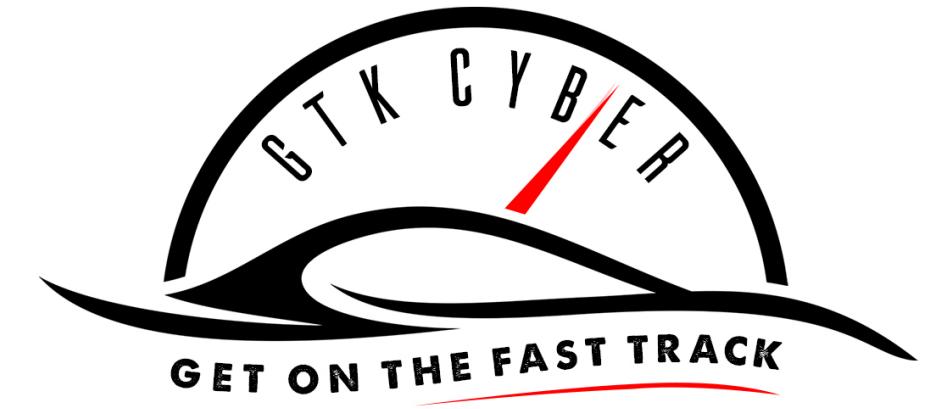
```
SELECT *
FROM
dfs.drillworkshop.`logs/`
LIMIT 10
```

Screenshot of the Apache Drill web interface showing the results of the query. The interface has a top navigation bar with tabs for Apache Drill, Query, Profiles, Storage, Metrics, Threads, Options, and Documentation. The main area displays a table with the following data:

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

Screenshot of the Apache Drill web interface showing the results of the query. The interface has a top navigation bar with tabs for Apache Drill, Query, Profiles, Storage, Metrics, Threads, Options, and Documentation. The main area displays a table with the following data:

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

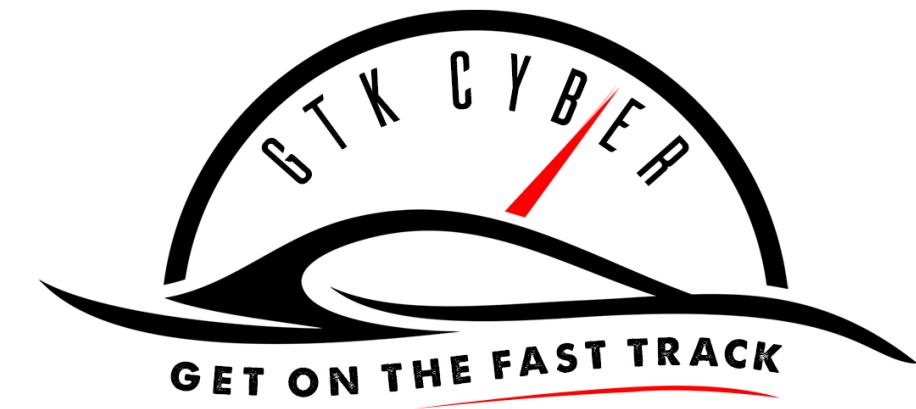


dir**n** accesses the
subdirectories



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>')
```

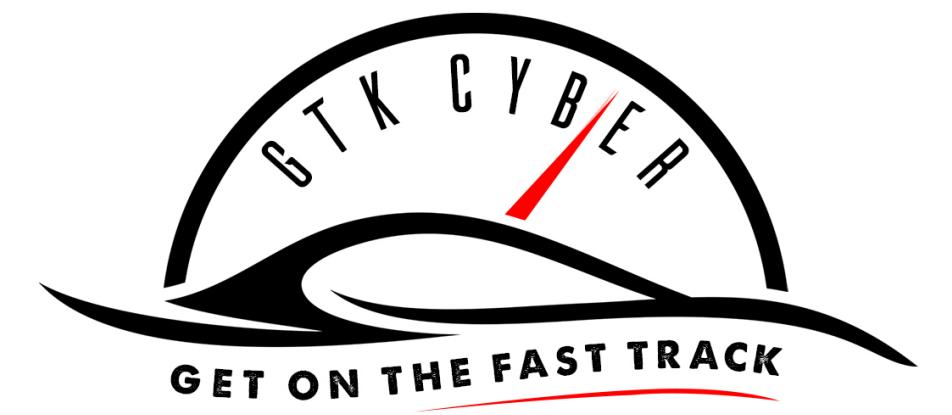


In Class Exercise:

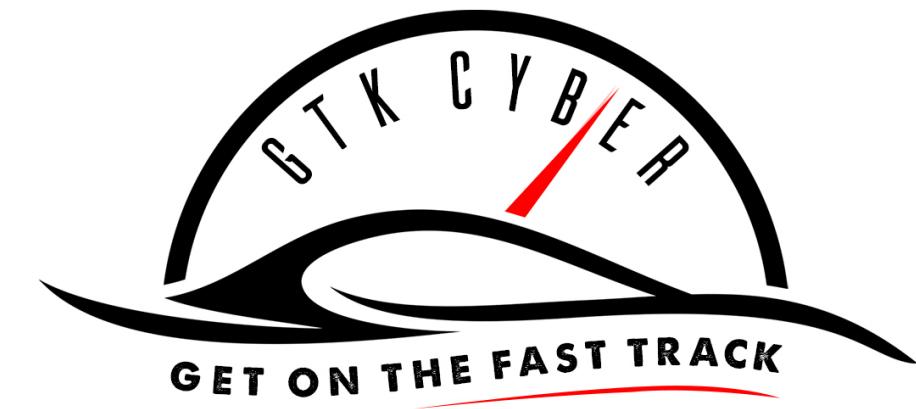
Find the total number of items sold by year and the total dollar sales in each year.

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

```
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
```

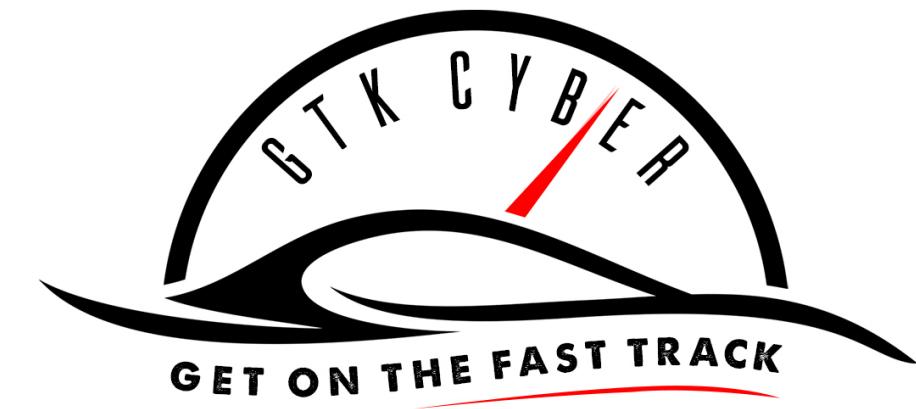


Let's look at JSON data



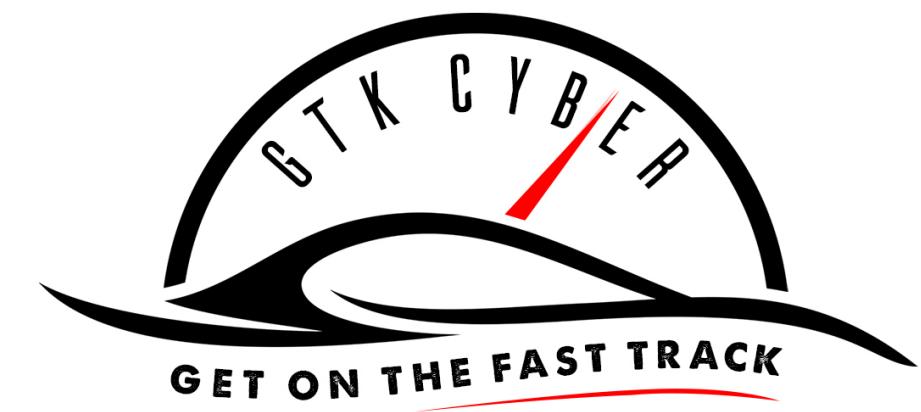
Let's look at JSON data

```
[  
  {  
    "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`
```

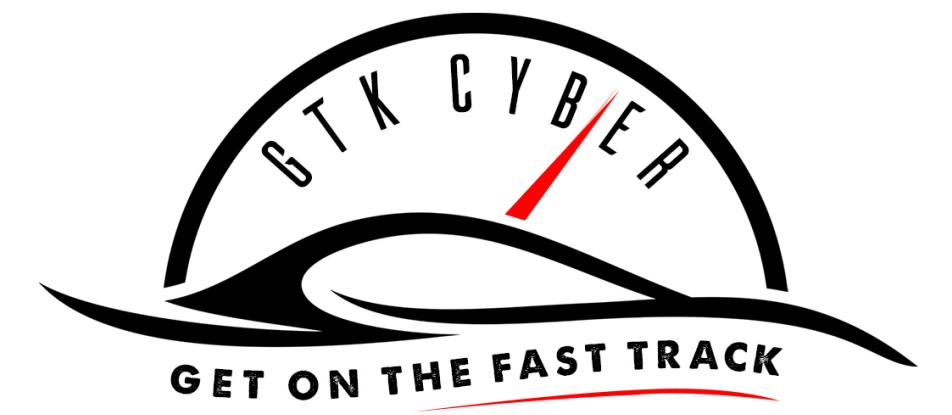
Screenshot of the Apache Drill web interface showing the results of the query.

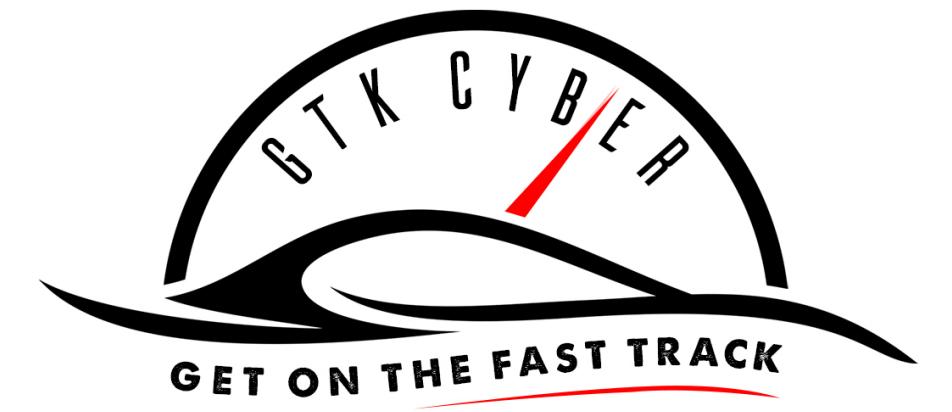
The interface includes a top navigation bar with links for Apache Drill, Query, Profiles, Storage, Metrics, Threads, Options, and Documentation. The URL bar shows "localhost".

Below the navigation bar is a search and filter section with "Show 10 entries" and a "Search:" input field.

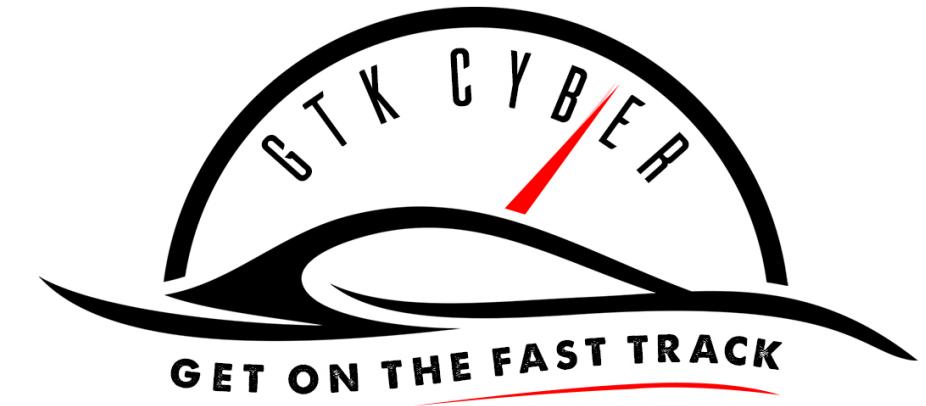
The main content area displays a table with the following data:

name	email	DOB	phone
Farley, Colette L.	iaculis@atarcu.ca	2011-08-14	1-758-453-3833
Kelley, Cherokee R.	ante.blandit@malesuadafringilla.edu	1992-09-01	1-595-478-7825
Bishop, Cheryl S.	in.faucibus@arcu.co.uk	2010-03-10	1-388-799-7554
Flowers, Vivien M.	dapibus@quamCurabitur.net	1992-04-04	1-246-672-9239





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)",  

```

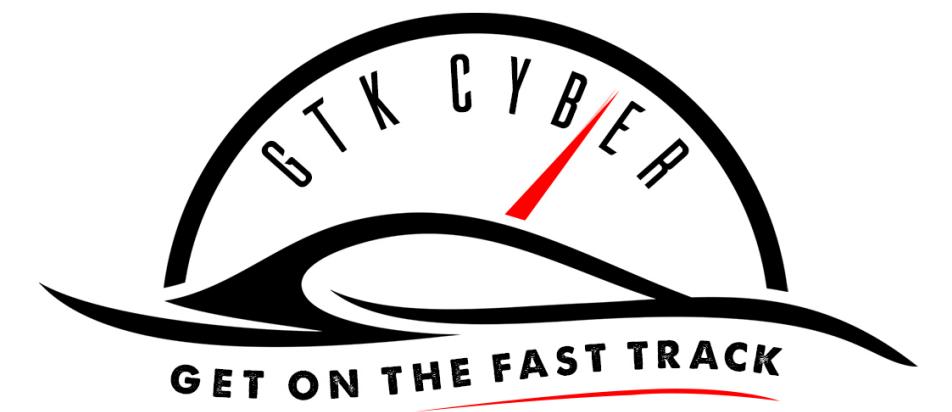


```
{  
  "meta" : {  
    "view" : {  
      "id" : "nsfe-bg53",  
      "name" : "Baltimore City Employee Salaries FY2015",  
      "attribution" : "Mayor's Office",  
      "averageRating" : 0,  
      "category" : "City Government",  
      ...  
      "  
        "format" : { }  
      },  
    },  
    "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)",  

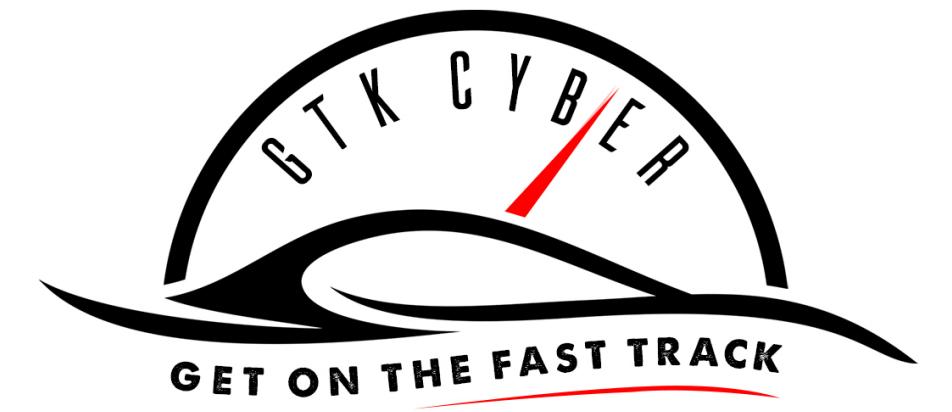
```



```
"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"  
]
```



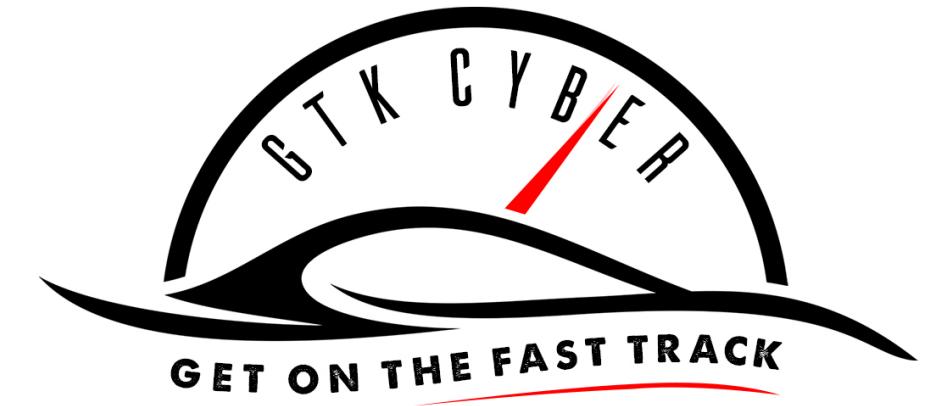
Drill has a series of functions for
nested data



Let's look at this data in Drill

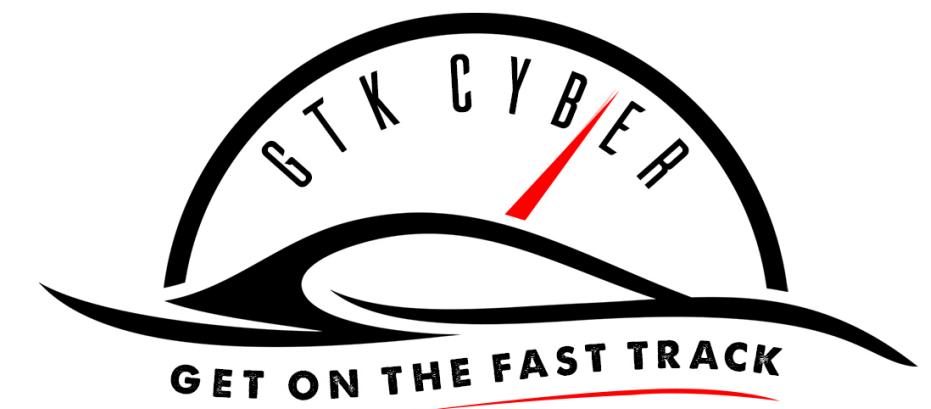


Please run
ALTER SYSTEM SET `store.json.all_text_mode` = true;
in the Drill command line



Let's look at this data in Drill

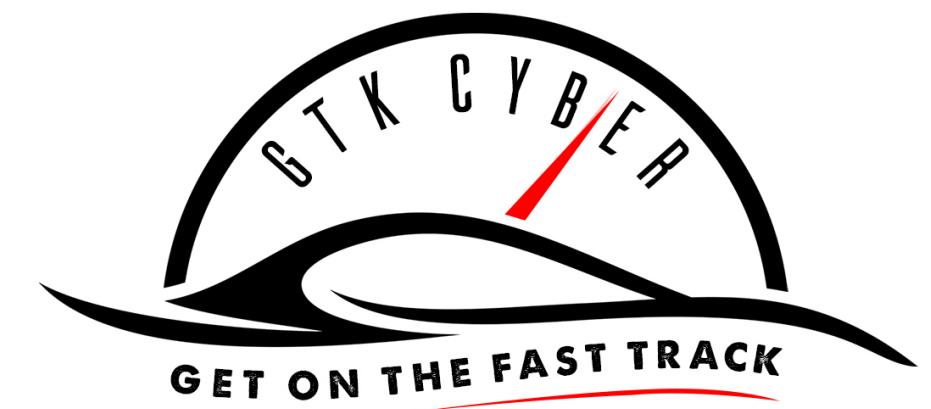
```
SELECT *
FROM dfs.drillclass.`baltimore_salaries.json`
```



Let's look at this data in Drill

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

The screenshot shows the Apache Drill web interface at the URL `localhost`. The top navigation bar includes links for Apache Drill, Query, Profiles, Storage, Metrics, Threads, Options, and Documentation. Below the navigation is a search bar with filters for 'Show' (set to 10), 'Search', and 'Show / hide columns'. The main content area displays a table with two columns: 'meta' and 'data'. The 'meta' column contains the schema and statistics for the table, while the 'data' column shows the actual salary values. A note at the bottom states: "part-time or summer clerks/fellows, annual salary is the equivalent full-time annual salary. Comp & leave time excluded."

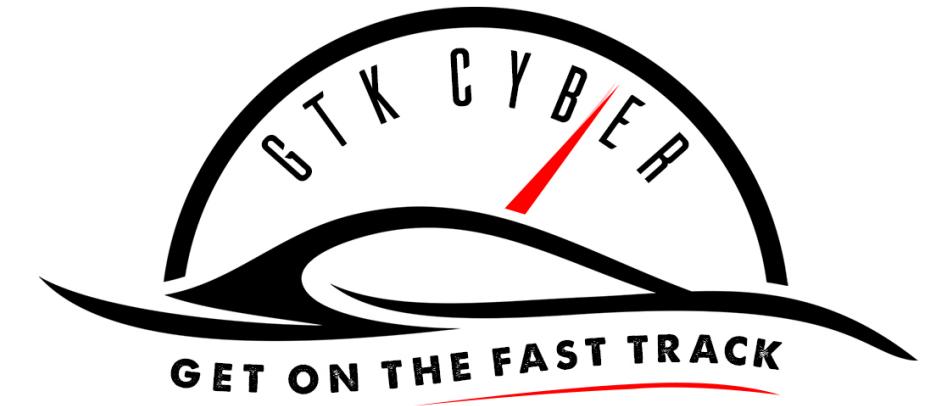


Let's look at this data in Drill

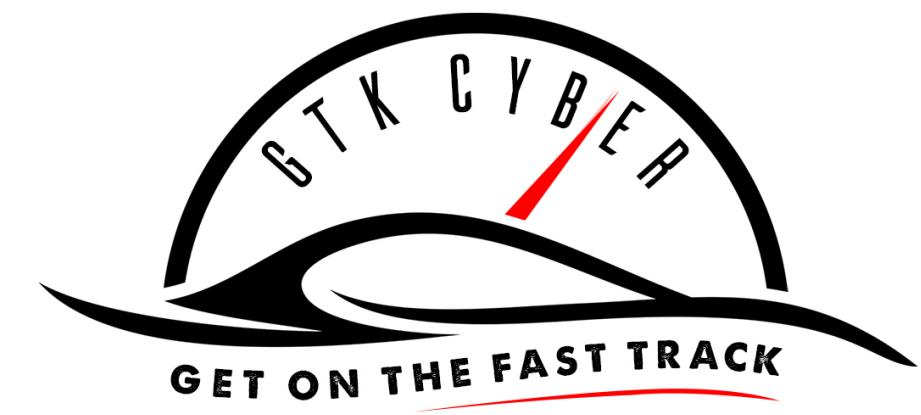
```
SELECT data  
FROM dfs.drillclass.`baltimore_salaries.json`
```

The screenshot shows the Apache Drill web interface running on localhost. The top navigation bar includes links for Apache Drill, Query, Profiles, Storage, Metrics, Threads, Options, and Documentation. Below the navigation is a search and filter section with "Show 10 entries" and a "Search:" input field. The main content area is titled "data" and displays a large JSON array of records. Each record contains fields such as IDs, names, departments, and various numerical values representing financial amounts or identifiers.

ID	Employee ID	First Name	Last Name	Department	Amount 1	Amount 2	Amount 3	Amount 4	Amount 5
1	66020CF9-8449-4464-AE61-B2292C7A0F2D	Aaron	Patricia G	Facilities/Office Services	55314.00	53626.04			
2	31C7A2FE-60E6-4219-890B-AFF01C09EC65	Aaron	Petra L	ASSISTANT STATE'S ATTORNEY	74000.00	73000.08			
3	AA8A6085-F2DE-43BA-966E-A441020DE420	Abaineh	Yohannes T	EPIDEMIOLOGIST	64500.00	64403.84			
4	080FCFF2-A9D8-4BF0-A00F-E295807ADA7A	Abbene	Anthony M	POLICE OFFICER	46309.00	59620.16			
5	38439D76-FA79-4990-9DA2-A3AA2197711F	Abbey	Emmanuel	CONTRACT SERV SPEC II	60060.00	54059.60			
6	6F514538-E76E-4F45-A991-EF0D5CE9D9B4	Abbott	Michelle	CONTRACT SERV SPEC II	42702.00	20250.80			
7	4B93421E0E88	Abdal-Rahim	Naim A	EMT Firefighter Suppression	62175.00	83757.48			
8	5AB13D6B-9D4C-4E08-A0AE-25EA96D4E584	Abdi	Ezekiel W	POLICE SERGEANT	77343.00	92574.91			
9	CC96354D-039B-4DF9-9D5A-AD1B9B6E3174	Abdul Adl	Attrice A	RADIO DISPATCHER SHERIFF	1999-03-30T00:00:00				



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**

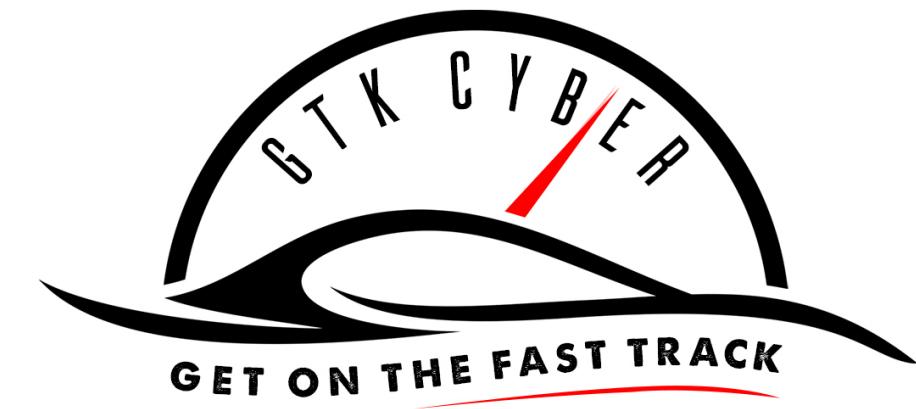
FROM dfs.drillclass.`baltimore_salaries.json`

Screenshot of the Apache Drill web interface showing the results of the query.

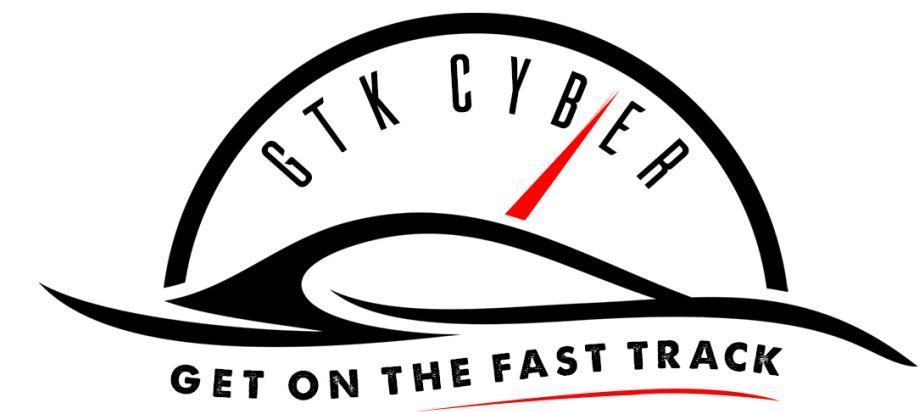
The interface includes a top navigation bar with links for Apache Drill, Query, Profiles, Storage, Metrics, Threads, Options, and Documentation. The URL in the address bar is localhost.

The main content area displays a table with the following data:

raw_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"]
["3","AA8A6085-F2DE-43BA-966E-A441020DE420","3","1438255843","393202","1438255843","393202","null","Abaineh,Yohannes T","EPIDEMIOLOGIST","A65026","HLTH-Health Department (026)","2009-07-23T00:00:00","64500.00","64403.84"]
["4","080FCFF2-A9D8-4BF0-A00F-E295807ADA7A","4","1438255843","393202","1438255843","393202","null","Abbene,Anthony M","POLICE OFFICER","A99005","Police Department (005)","2013-07-24T00:00:00","46309.00","59620.16"]
["5","38439D76-FA79-4990-9DA2-A3AA2197711F","5","1438255843","393202","1438255843","393202","null","Abbey,Emmanuel","CONTRACT SERV SPEC II","A40001","M-R Info Technology (001)","2013-05-01T00:00:00","60060.00","54059.60"]
["6","6F514538-E76E-4F45-A991-EF0D5CE9D9B4","6","1438255843","393202","1438255843","393202","null","Abbott-Cole,Michelle","CONTRACT SERV SPEC II","A90005","TRANS-Traffic (005)","2014-11-28T00:00:00","42702.00","20250.80"]



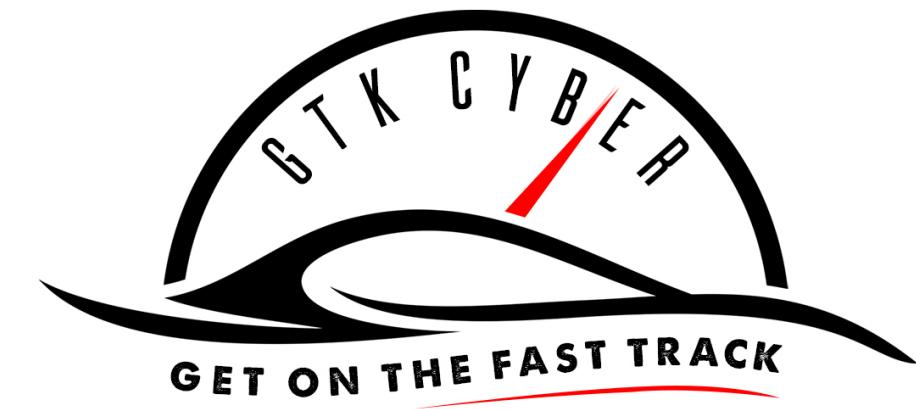
```
SELECT raw_data[8] AS name
FROM
(
  SELECT FLATTEN( data ) AS raw_data
  FROM dfs.drillclass.`baltimore_salaries_2015.json`
```



```
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`  
)
```

The screenshot shows the Apache Drill web interface running on localhost. The top navigation bar includes links for Apache Drill, Query, Profiles, Storage, Metrics, Threads, Options, and Documentation. Below the navigation is a search bar with dropdowns for 'Show 10 entries' and 'Search'. A 'Show / hide columns' button is also present. The main content area displays a table with two columns: 'name' and 'job_title'. The data rows are:

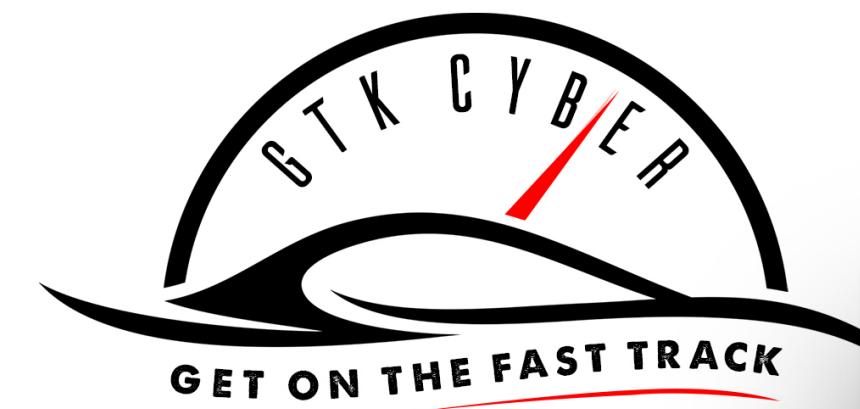
name	job_title
Aaron,Patricia G	Facilities/Office Services II
Aaron,Petra L	ASSISTANT STATE'S ATTORNEY
Abaineh,Yohannes T	EPIDEMIOLOGIST
Abbene,Anthony M	POLICE OFFICER
Abbey Emmanuel	CONTRACT SERV SPEC II



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
```

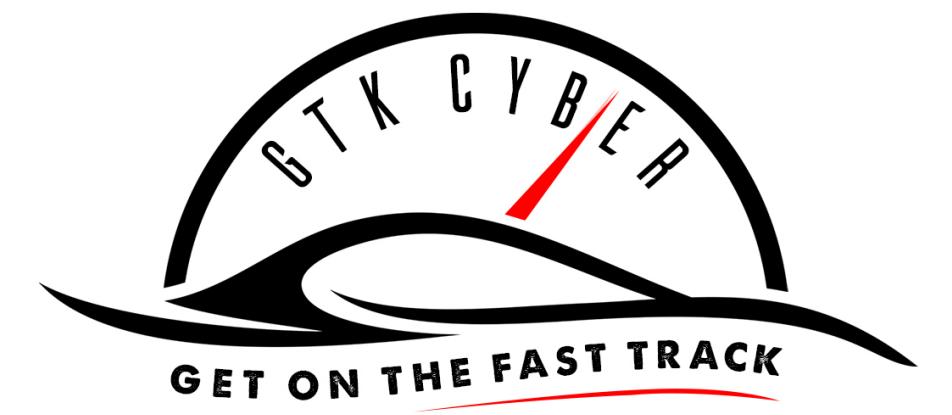


localhost

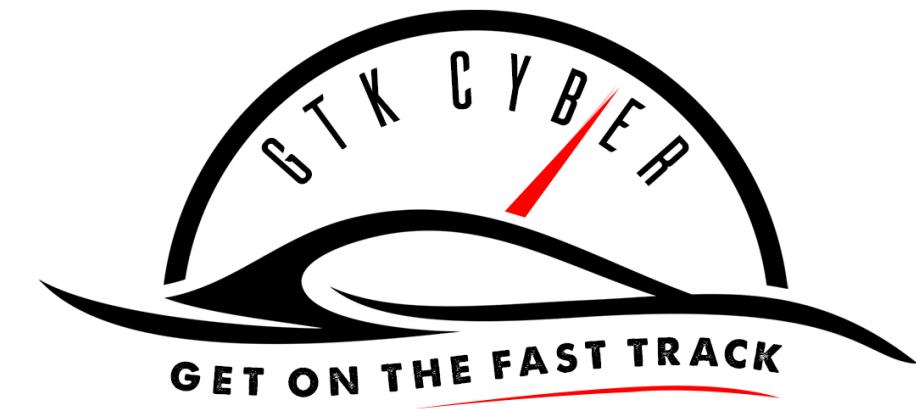
Apache Drill Query Profiles Storage Metrics Threads Options Documentation

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	166401.666666666666	9
CITY AUDITOR	159800.0	1

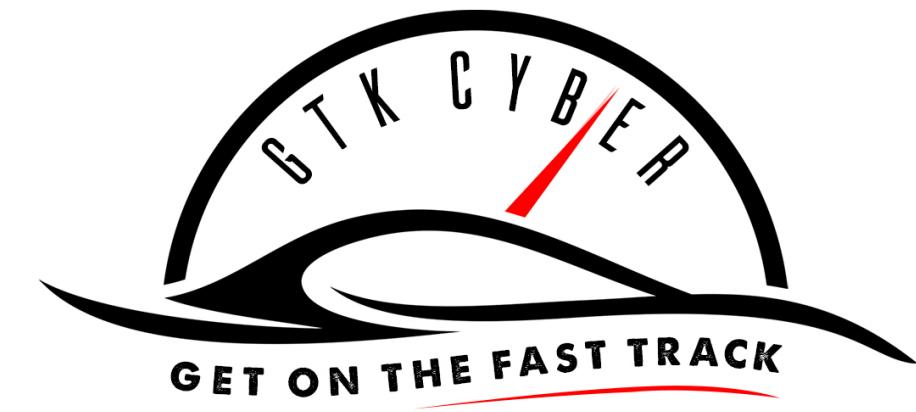


Log Files

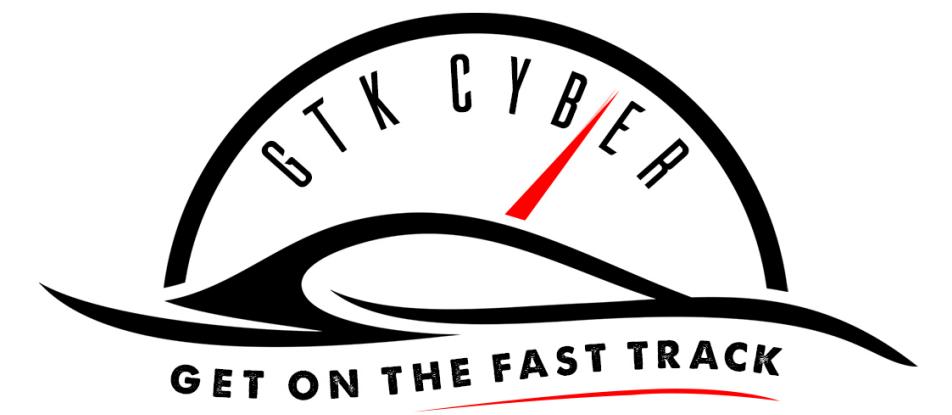


Log Files

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"
        }
    ]
}
```

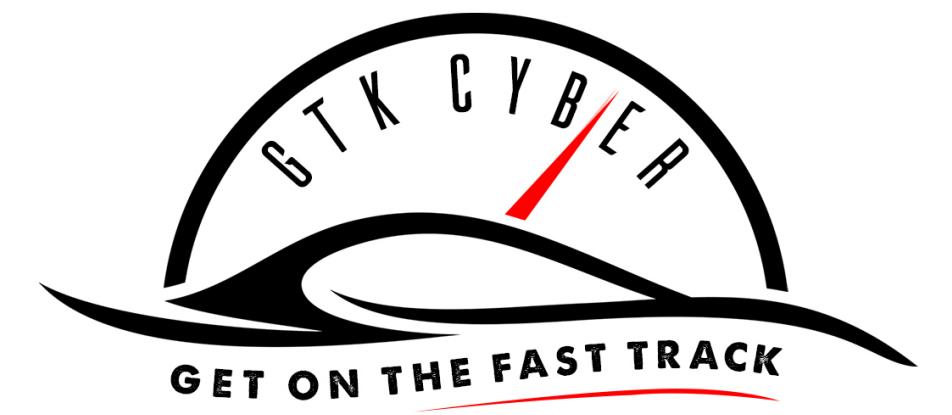


```
SELECT *
FROM dfs.drillworkshop.`log_files/mysql.log`
```

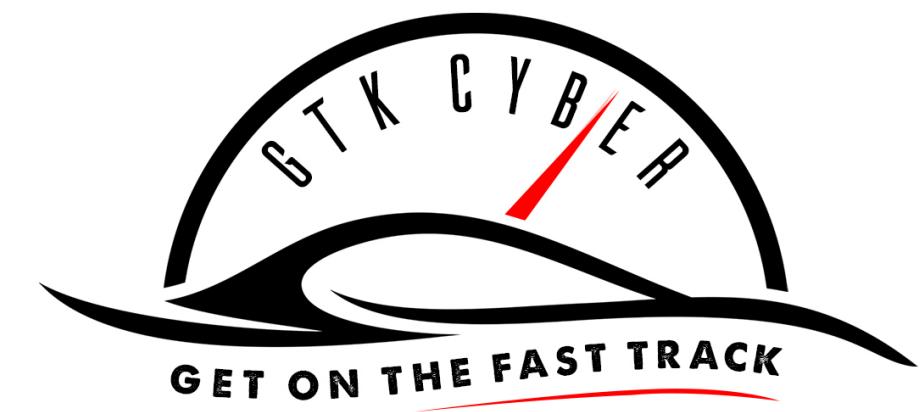


```
SELECT *
FROM dfs.drillworkshop.`log_files/mysql.log`
```

Show 10 entries					
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

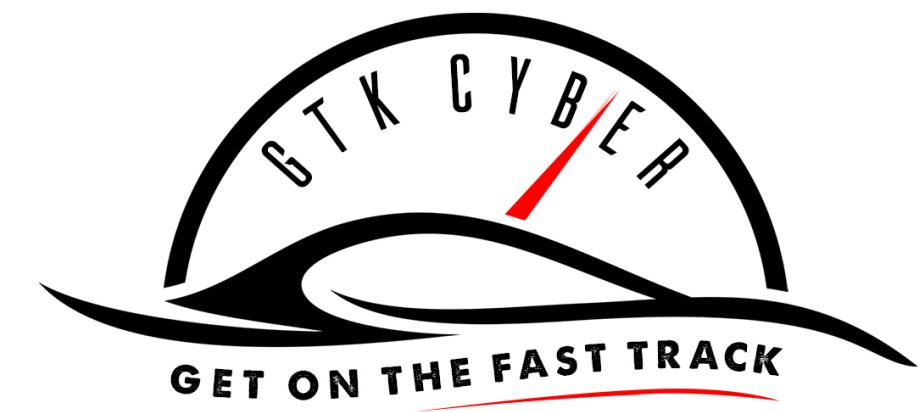


HTTPD Log Files



HTTPD Log Files

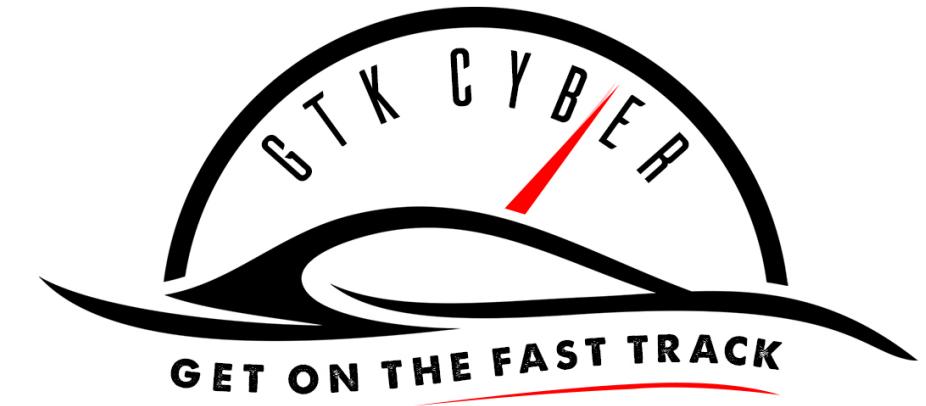
```
195.154.46.135 - - [25/Oct/2015:04:11:25 +0100] "GET /linux/doing-pxe-without-dhcp-control HTTP/1.1" 200 24323 "http://howto.basjes.nl/" "Mozilla/5.0 (Windows NT 5.1; rv:35.0) Gecko/20100101 Firefox/35.0"
23.95.237.180 - - [25/Oct/2015:04:11:26 +0100] "GET /join_form HTTP/1.0" 200 11114 "http://howto.basjes.nl/" "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

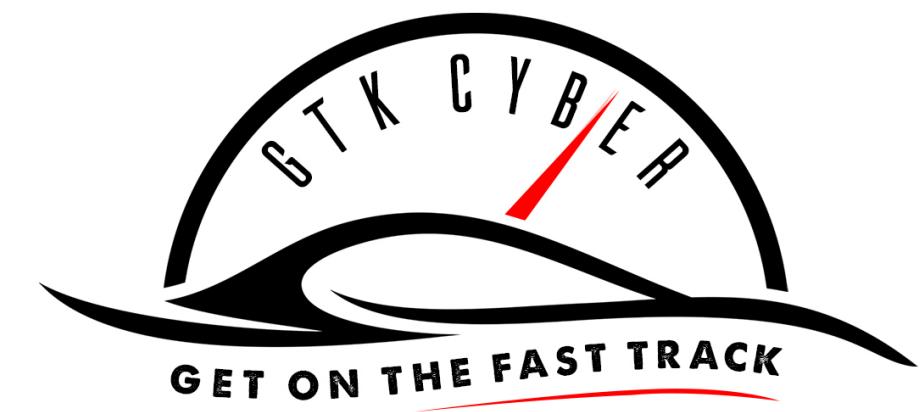
```
195.154.46.135 - - [25/Oct/2015:04:11:25 +0100] "GET /linux/doing-pxe-without-dhcp-control HTTP/1.1" 200 24323 "http://howto.basjes.nl/" "Mozilla/5.0 (Windows NT 5.1; rv:35.0) Gecko/20100101 Firefox/35.0"
23.95.237.180 - - [25/Oct/2015:04:11:26 +0100] "GET /join_form HTTP/1.0" 200 11114 "http://howto.basjes.nl/" "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`
```

Apache Drill Query Profiles Storage Metrics Threads Logs Options Documentation

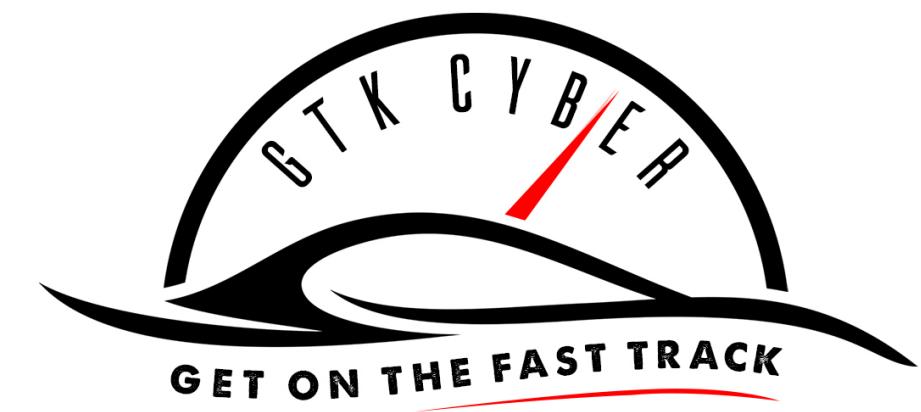
Show 10 entries Search: Show / hide columns

request_receive_time_second	connection_client_host	request_referer_userinfo	request_referer_path	request_referer_host	request_receive_time_monthname
	195.154.46.135	null	/	howto.basjes.nl	October
	23.95.237.180	null	/	howto.basjes.nl	October



HTTPD Log Files

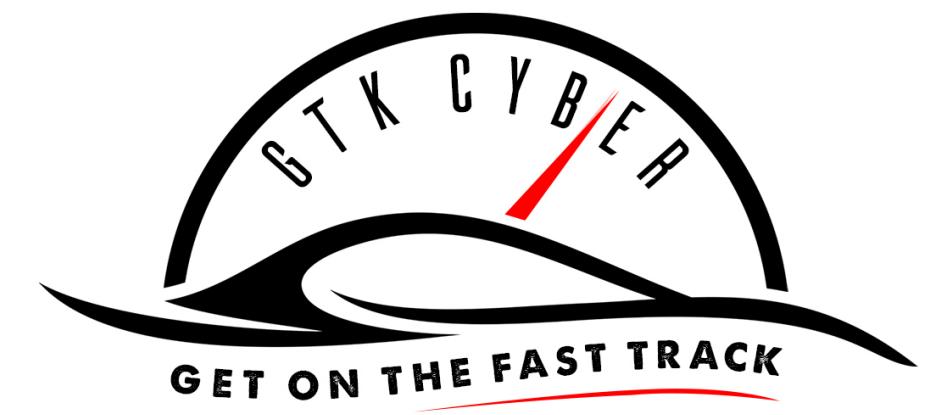
```
SELECT request_referer, parse_url( request_referer ) AS url_data
FROM dfs.drillworkshop.`data_files/log_files/small-server-log.httpd`
```



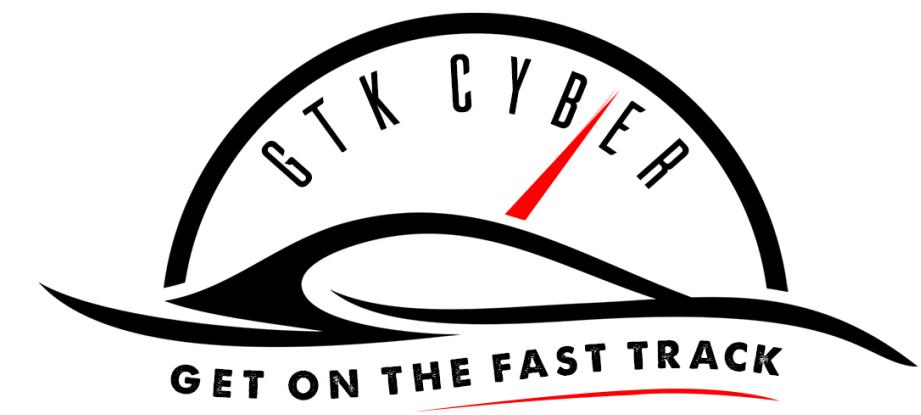
HTTPD Log Files

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

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"}

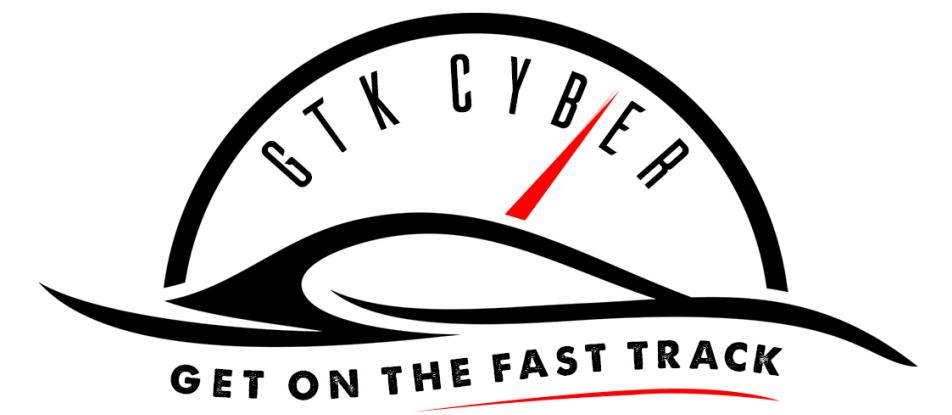


Networking Functions

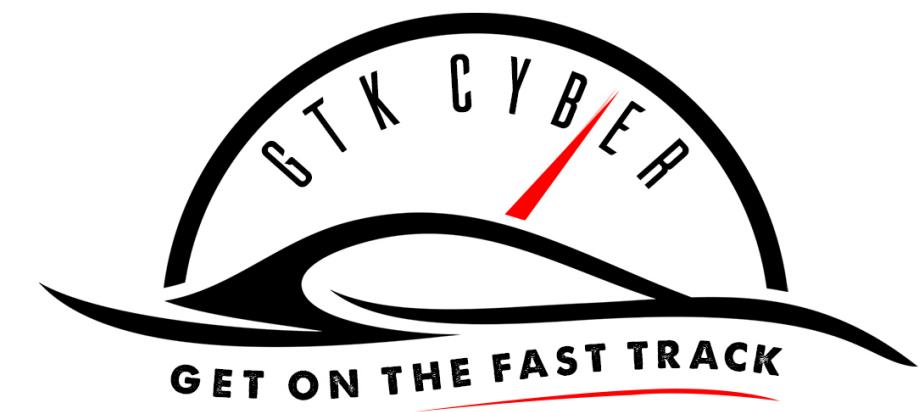


Networking Functions

- `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



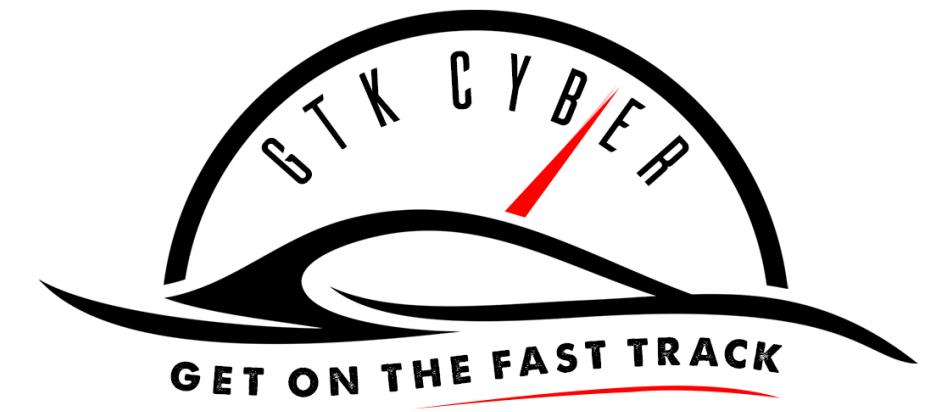
PCAP Files



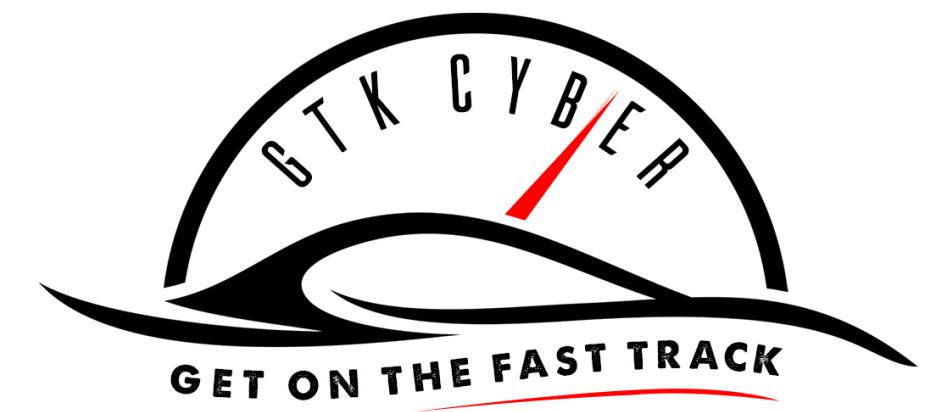
```
SELECT *
FROM dfs.test.`dns-zone-transfer-ixfr.pcap`
```

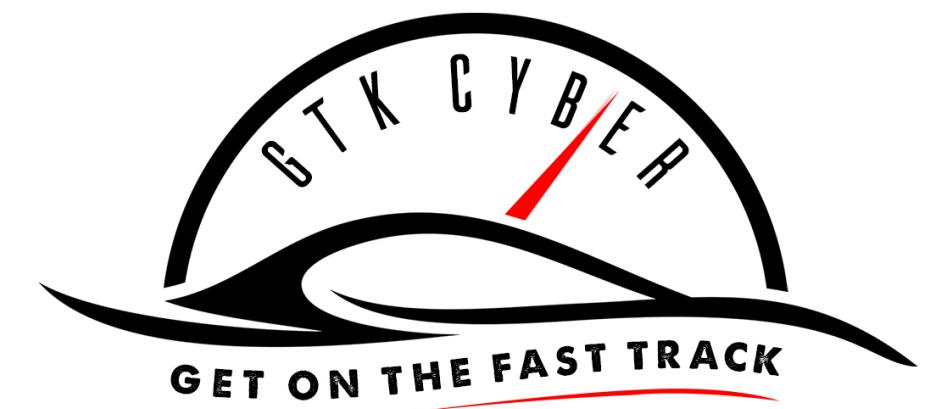
Screenshot of the Apache Drill web interface showing the results of the query:

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



Connecting other Data Sources





Connecting other Data Sources

Apache Drill Query Profiles **Storage** Metrics Threads Options Documentation

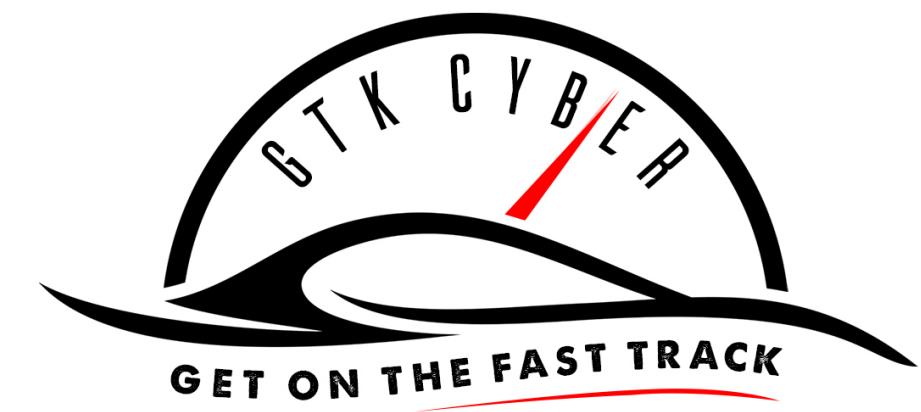
The 'Storage' tab is highlighted with a red oval and a red arrow points to it from the text 'Click here'.

Enabled Storage Plugins

cp	Update	Disable
dfs	Update	Disable

Disabled Storage Plugins

hbase	Update	Enable
hive	Update	Enable
kudu	Update	Enable
mongo	Update	Enable
<i>gtkcybe</i>	Update	Enable
s3	Update	Enable



Connecting other Data Sources

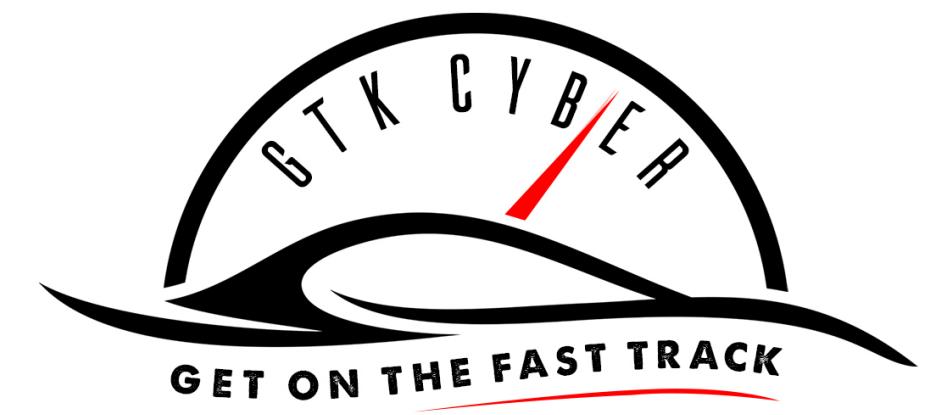
The screenshot shows a Mozilla Firefox window with the title bar "Apache Drill - Mozilla Firefox". The menu bar includes "File", "Edit", "View", "History", "Bookmarks", "Tools", and "Help". The toolbar has tabs for "phpMyAdmin", "How to import an S...", and "Apache Drill". The address bar displays "localhost:8047/storage/mysql". The main content area has a dark header with "Apache Drill", "Query", "Profiles", "Storage", "Metrics", and "Threads" navigation links. Below the header is a configuration section with the following JSON code:

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

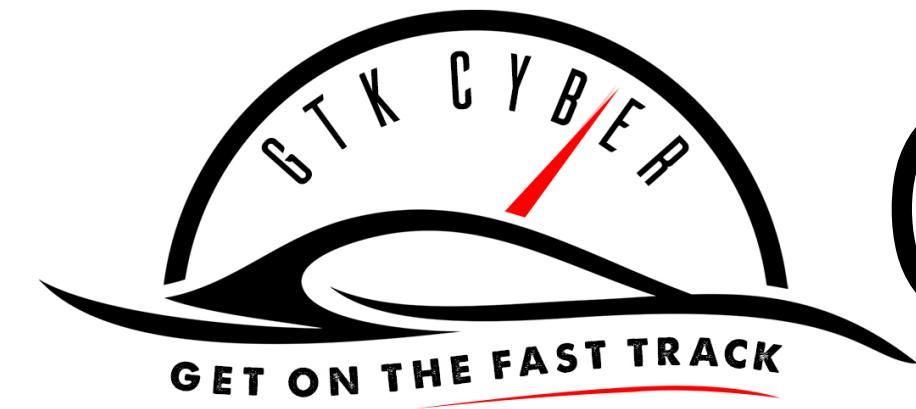
Configuration

```
{  
  "type": "jdbc",  
  "driver": "com.mysql.jdbc.Driver",  
  "url": "jdbc:mysql://localhost:3306",  
  "username": "merlinuser",  

```

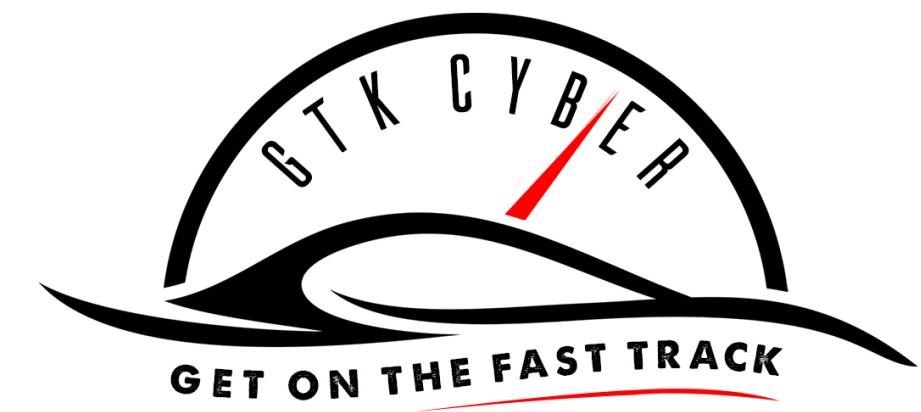


MySQL™



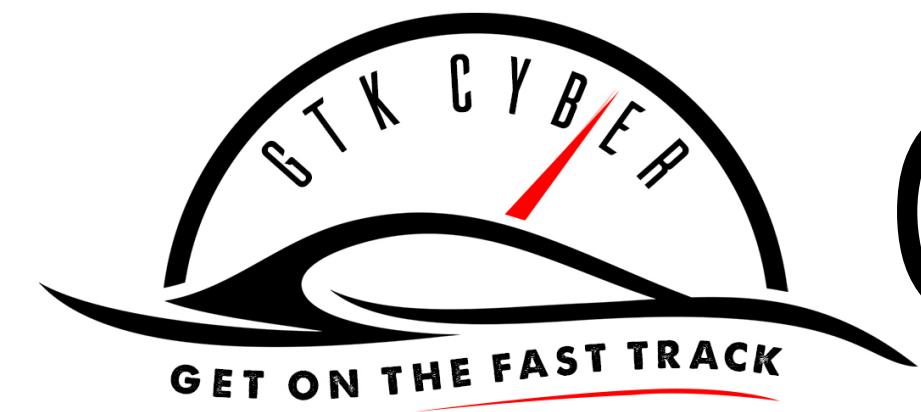
Connecting other Data Sources

```
{  
    "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

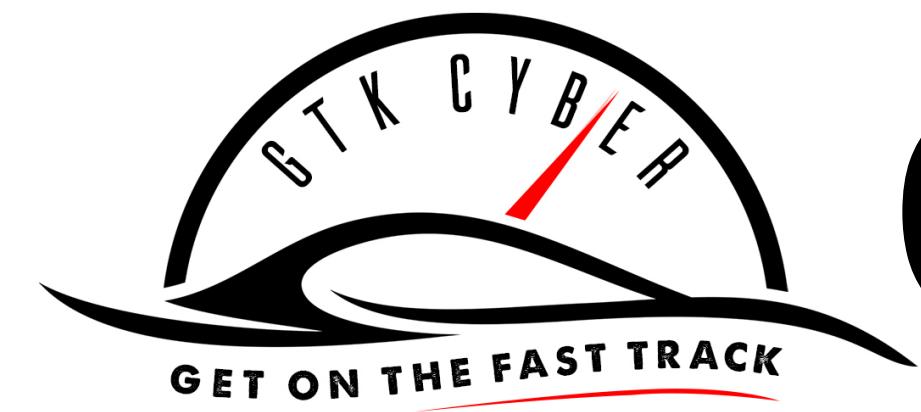
```
Terminal
File Edit View Search Terminal Help
[Error Id: 99a10b7f-4ed6-4bba-a408-5b21a71fbea2 on localhost:31010] (state=, code=0)
0: jdbc:drill:zk=local> show databases;
+-----+
| SCHEMA_NAME |
+-----+
| INFORMATION_SCHEMA |
| cp.default |
| dfs.default |
| dfs.root |
| dfs.tmp |
| mysql.information_schema |
| mysql.mysql |
| mysql.performance_schema |
| mysql.phpmyadmin |
| mysql.stats |
| mysql.test |
| mysql |
| sys |
+-----+
13 rows selected (30.187 seconds)
0: jdbc:drill:zk=local>
```



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

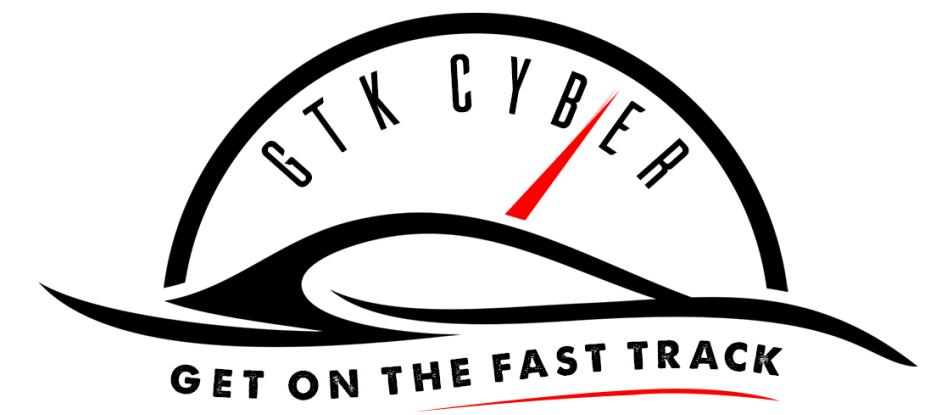


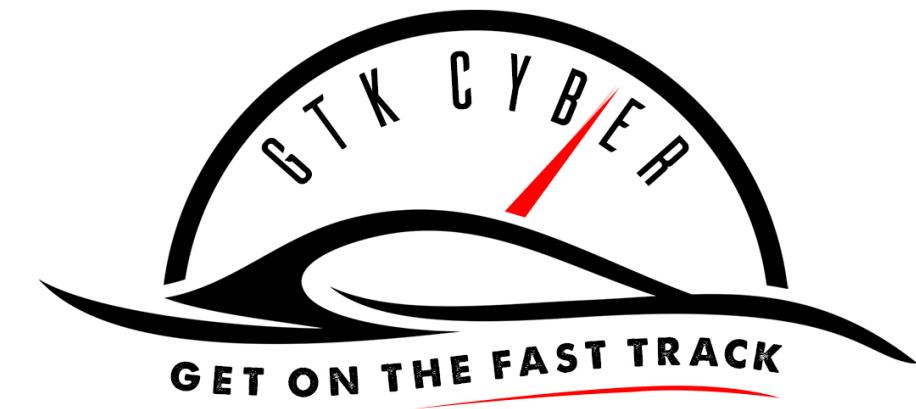
Connecting other Data Sources

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

MySQL: 0.047 seconds

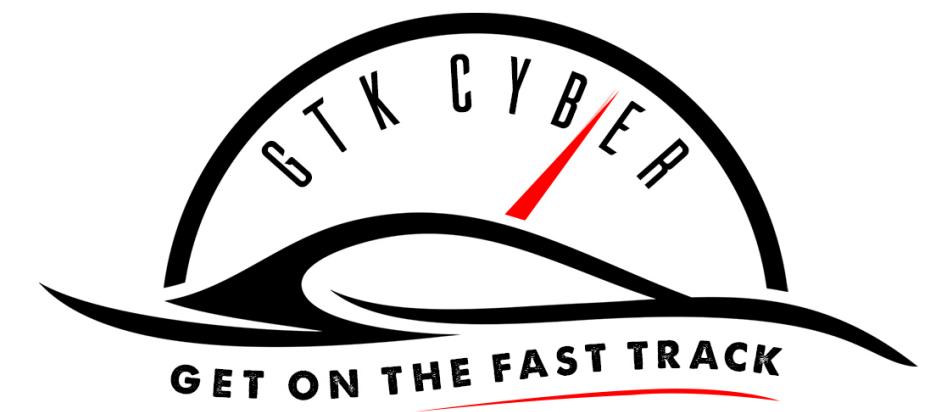
Drill: 0.366 seconds



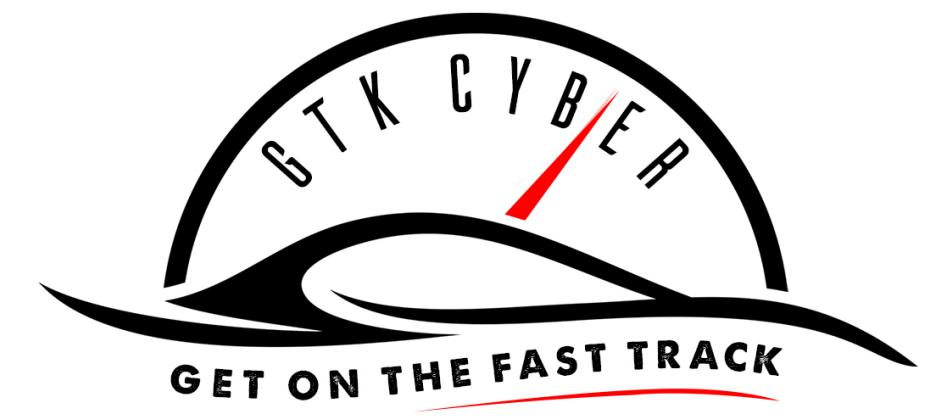


Just like DFS, except you specify a link to the Hadoop namenode.

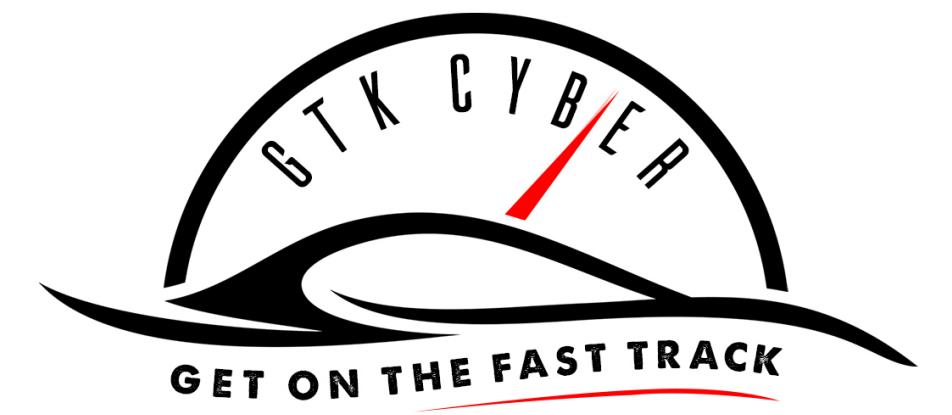
```
{  
    "type": "file",  
    "enabled": true,  
    "connectionhdfs://localhost:54310",  
    "config": null,  
    "workspaces": {  
        "demodata": {  
            "location": "/user/merlinuser/demo",  
            "writable": true,  
            "defaultInputFormat": null  
        }  
 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



```
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
```

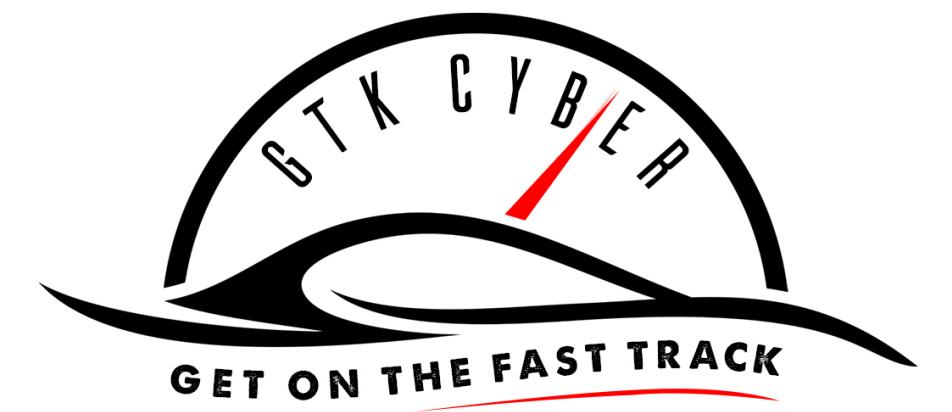


Connecting to Drill

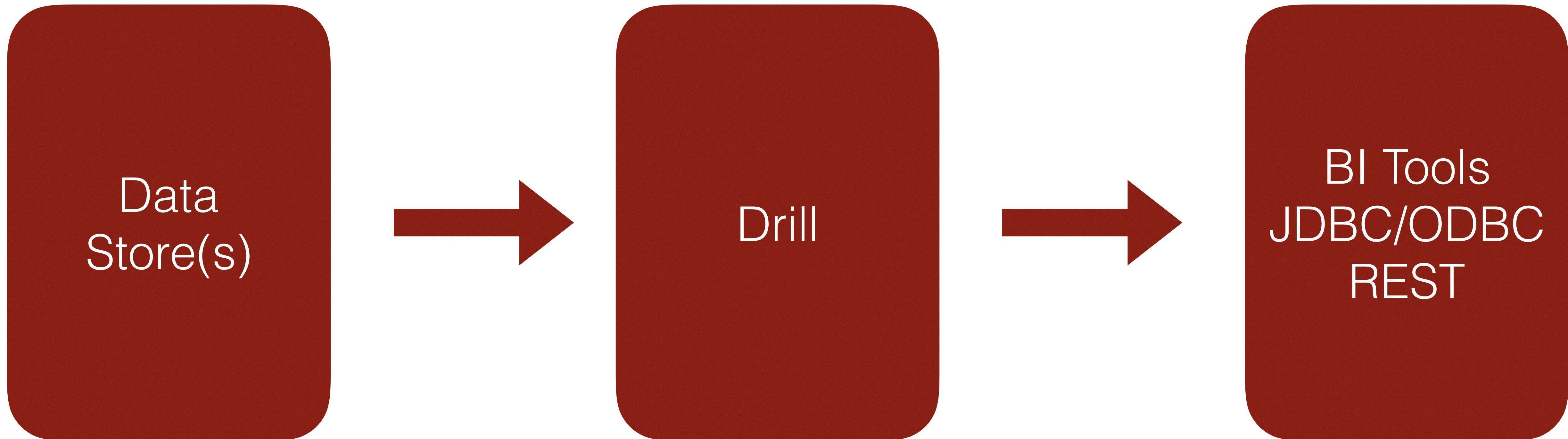


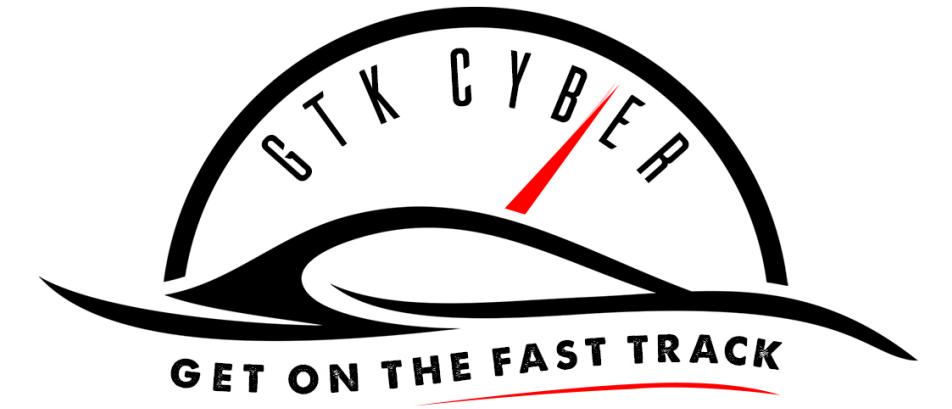
Python





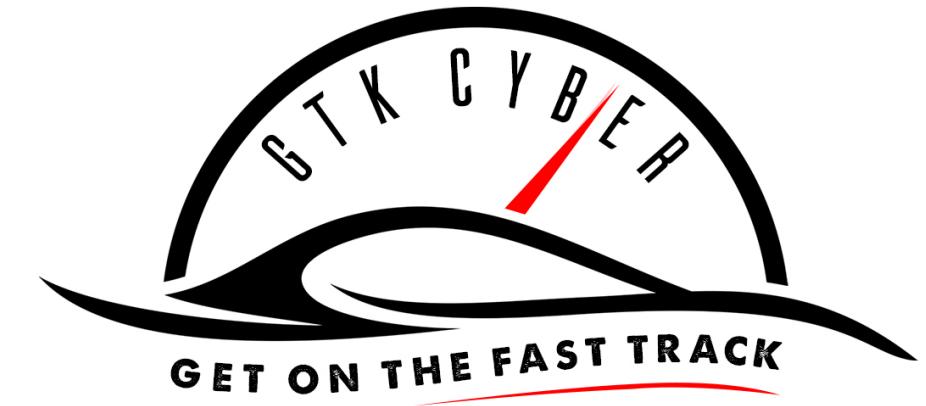
Connecting to Drill





Connecting to Drill

```
pip install pydrill
```



Connecting to Drill

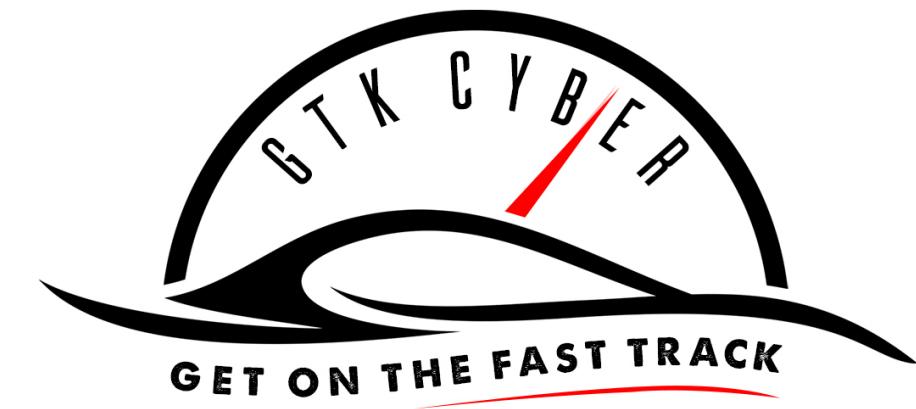
```
from pydrill.client import PyDrill
```



Connecting to Drill

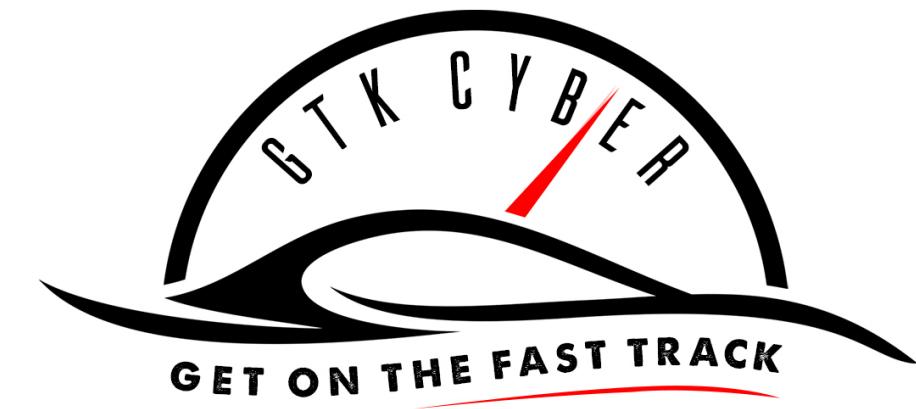
```
drill = PyDrill(host='localhost', port=8047)

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



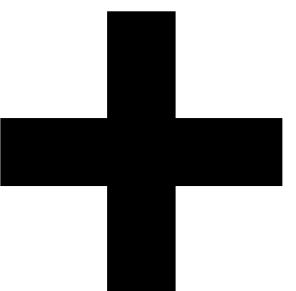
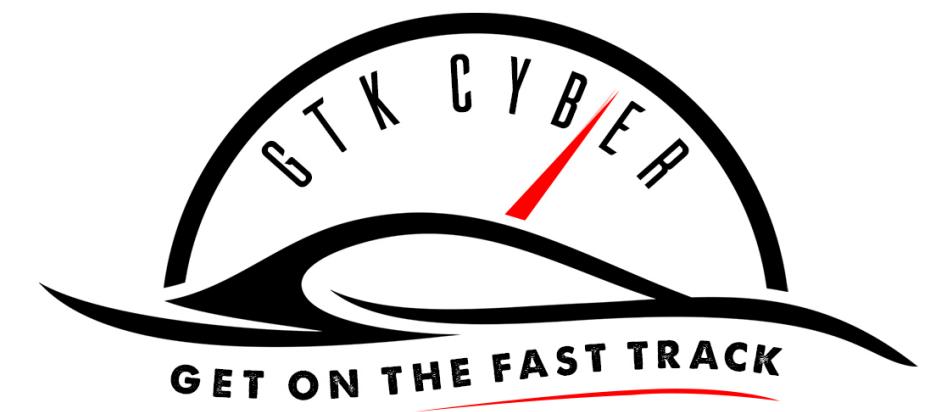
Connecting to Drill

```
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



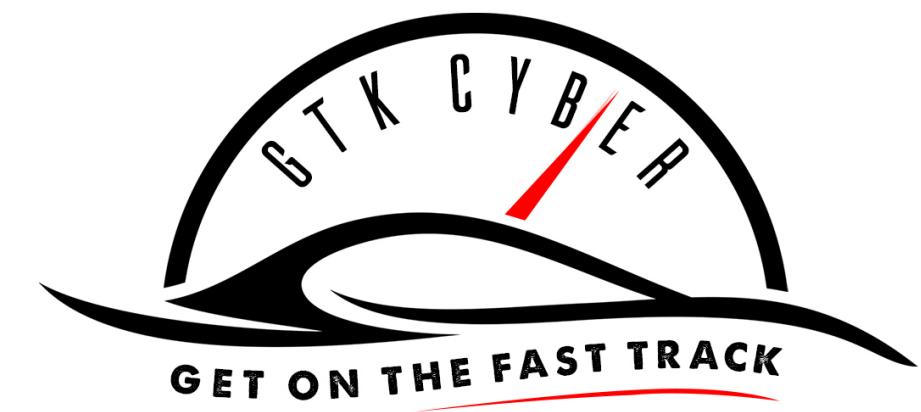
Connecting Drill and Zeppelin

- 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

localhost

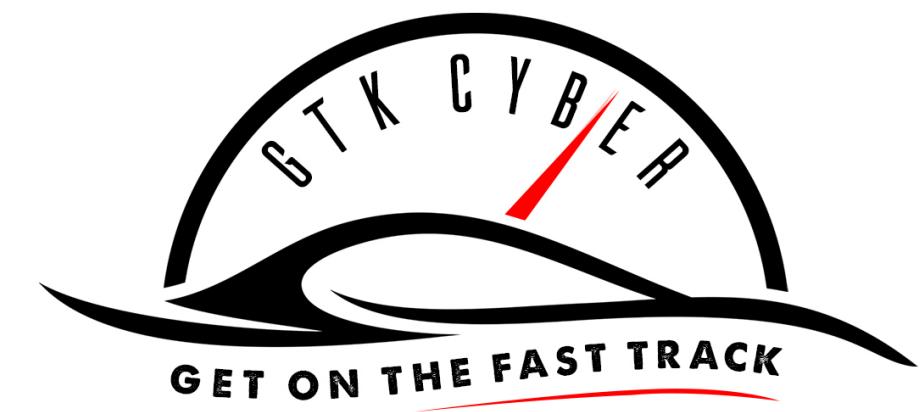
Zeppelin Notebook Job Search your Notes anonymous

Drill Example

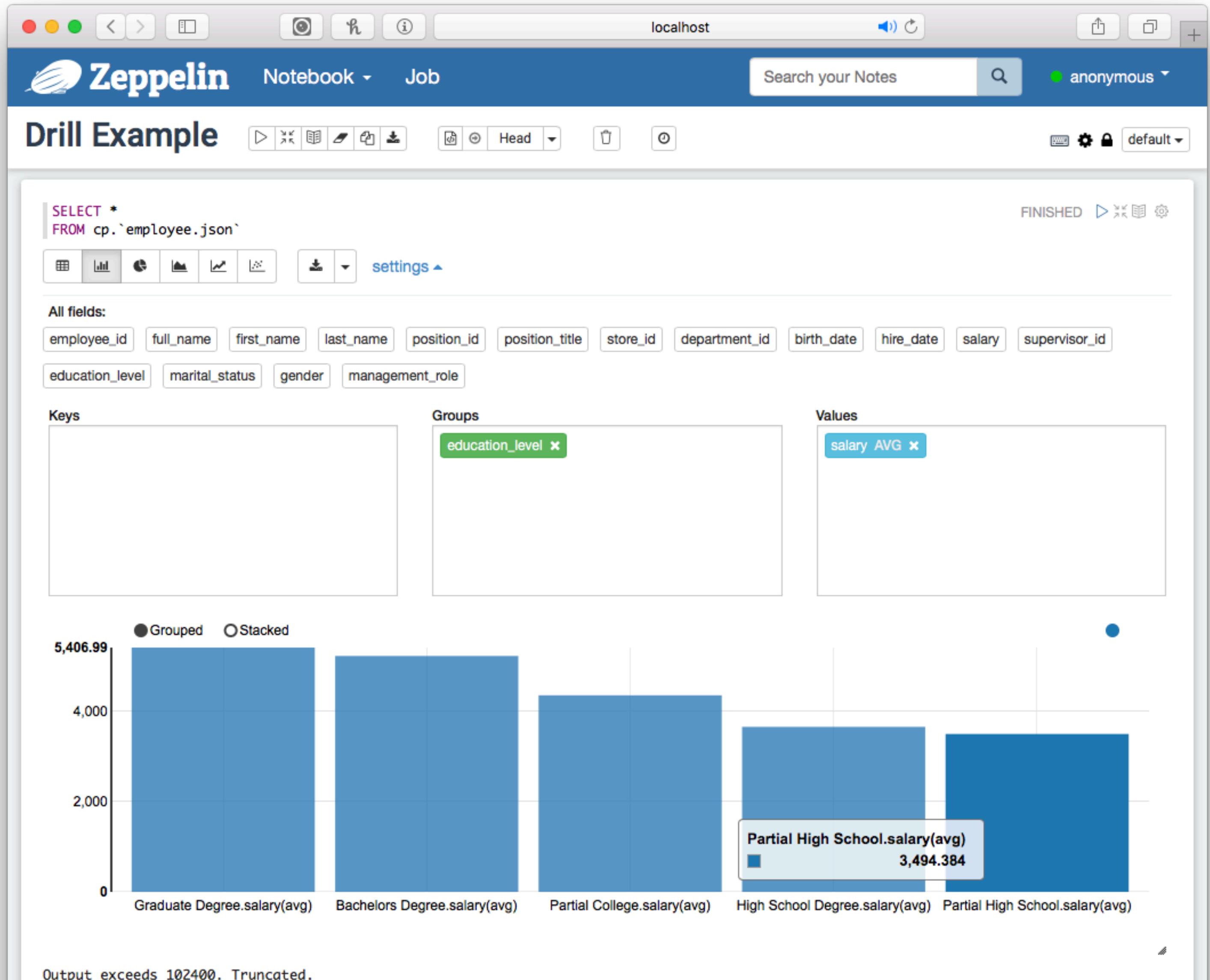
```
SELECT *
FROM cp.`employee.json`
LIMIT 100
```

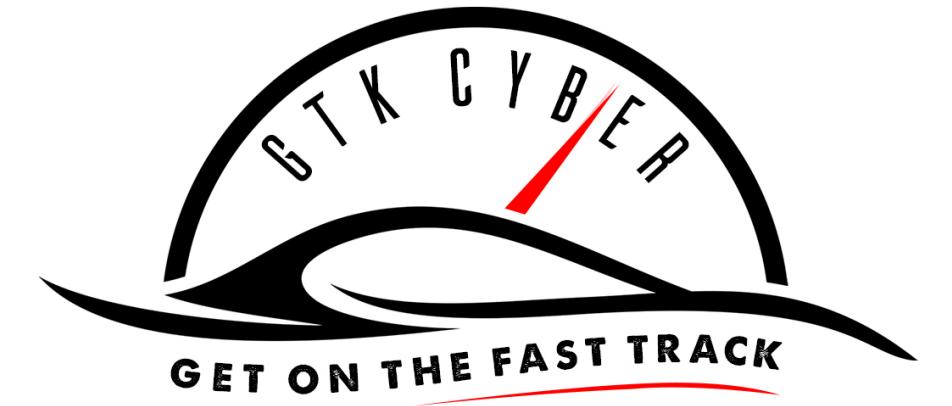
FINISHED

employee_id	full_name	first_name	last_name	position_id	position_title	store_id	department_id	birth_date
1	Sheri Nowmer	Sheri	Nowmer	1	President	0	1	1961-08-2
2	Derrick Whelby	Derrick	Whelby	2	VP Country Manager	0	1	1915-07-0
4	Michael Spence	Michael	Spence	2	VP Country Manager	0	1	1969-06-2
5	Maya Gutierrez	Maya	Gutierrez	2	VP Country Manager	0	1	1951-05-1

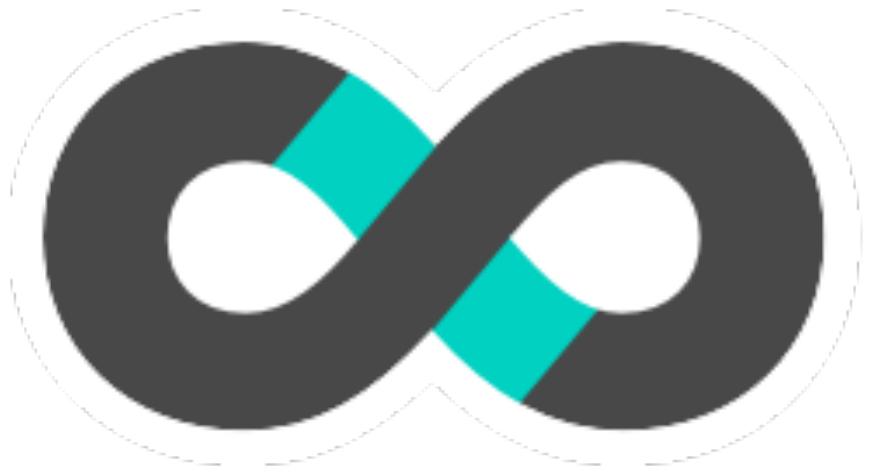


Connecting Drill and Zeppelin

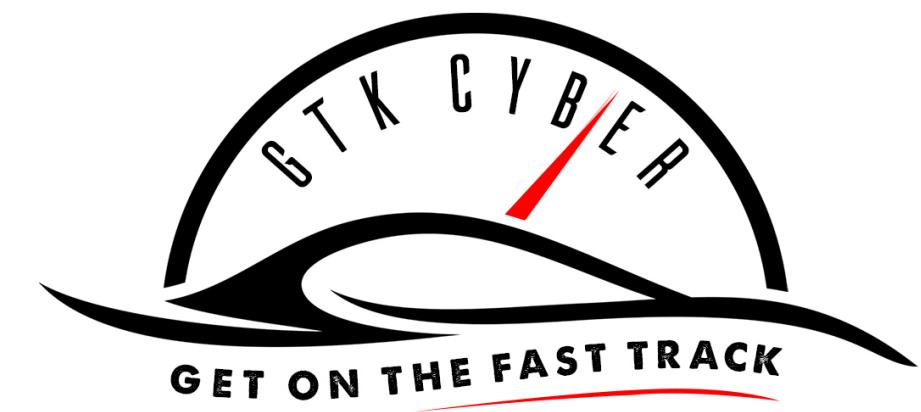




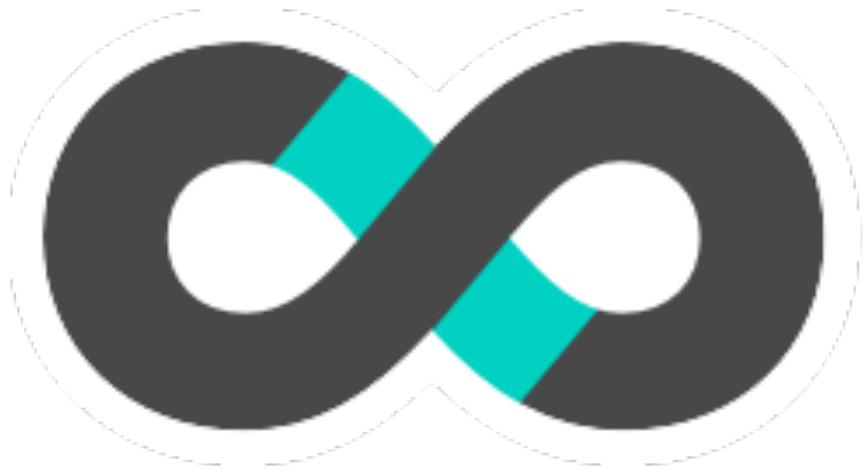
Building Dashboards with Drill and Superset



- 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>

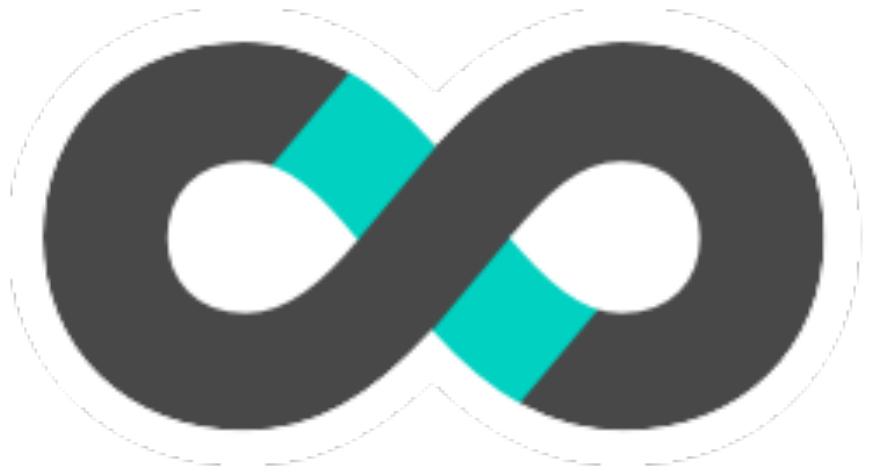
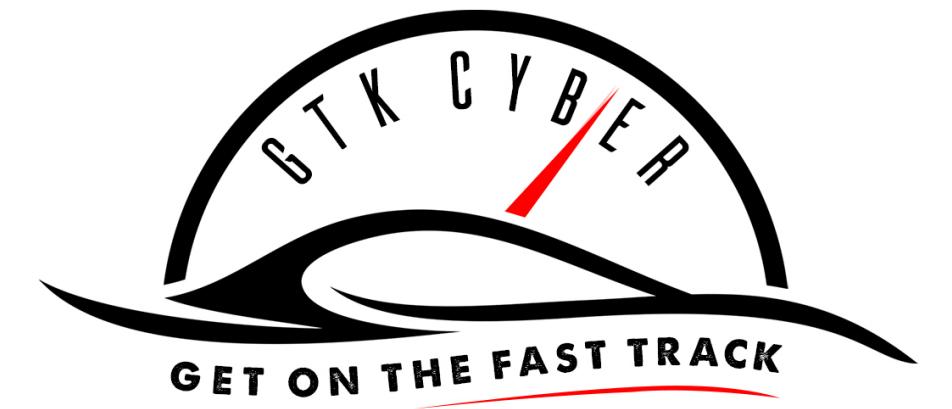


Building Dashboards with Drill and Superset



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

The screenshot shows the Superset web application running in a browser. The URL in the address bar is 'localhost'. The top navigation bar includes links for 'Superset', 'Security', 'Manage', 'Sources' (which is currently selected), 'Slices', 'Dashboards', 'SQL Lab', and user account settings. A search bar is also present. On the left, there's a sidebar with a 'Dashboards' section and a 'Databases' button. The main content area is currently empty, indicating no data sources have been added yet.

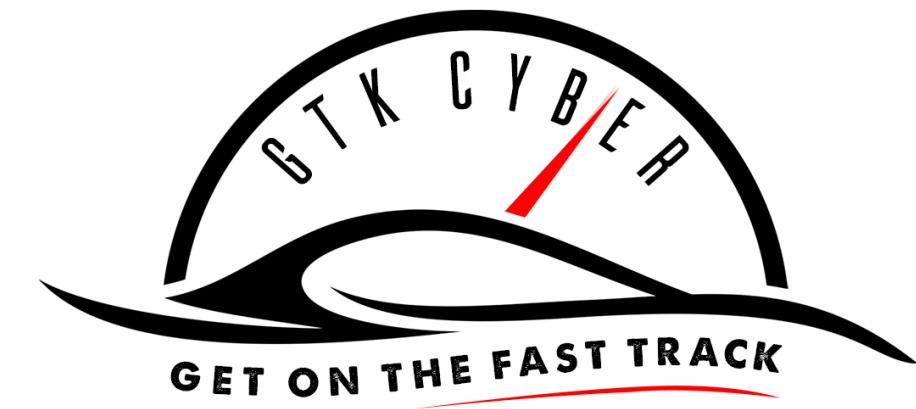


Building Dashboards with Drill and Superset

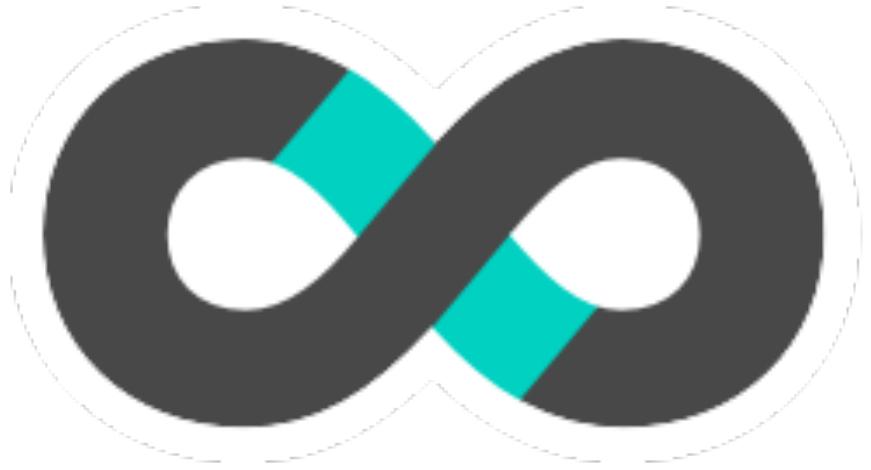
- Use this string to connect to Drill in embedded mode:

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

The screenshot shows the 'Edit Database' page in the Superset interface. The 'Database' dropdown is set to 'Drill'. The 'SQLAlchemy URI' field contains the value 'drill+sadrill://localhost:8047/dfs/drillclass?use_ssl=False'. A note below the field says 'Refer to the [SqlAlchemy docs](#) for more information on how to structure your URI.' A 'Test Connection' button is visible. The URL in the browser bar is 'localhost'.



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`
```

The screenshot shows the Superset SQLLab interface. At the top, there is a code editor containing the following SQL query:

```
1 SELECT Agency, TO_NUMBER(`AnnualSalary`, '¤') As AnnualSalary
2 FROM dfs.drillclass.`baltimore_salaries_2016.csvh`
```

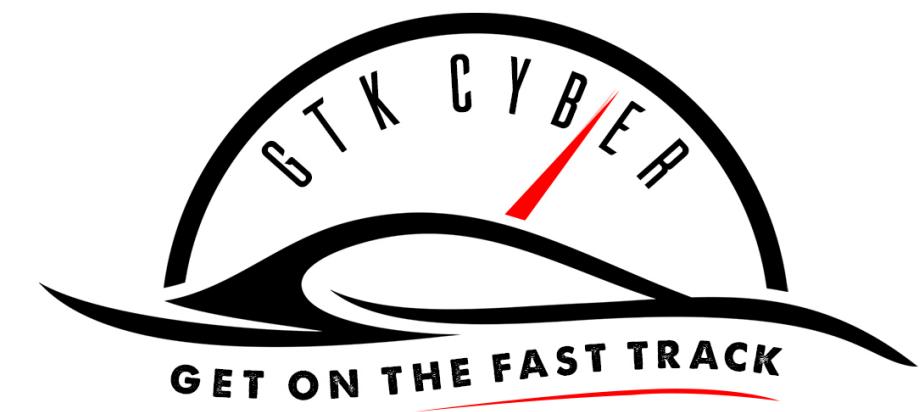
Below the code editor are two buttons: "Run Query" (green) and "Save Query". To the right of the buttons is a green progress bar indicating the query took 00:00:01.19 to run.

Underneath the code editor, there are two tabs: "Results" (selected) and "Query History".

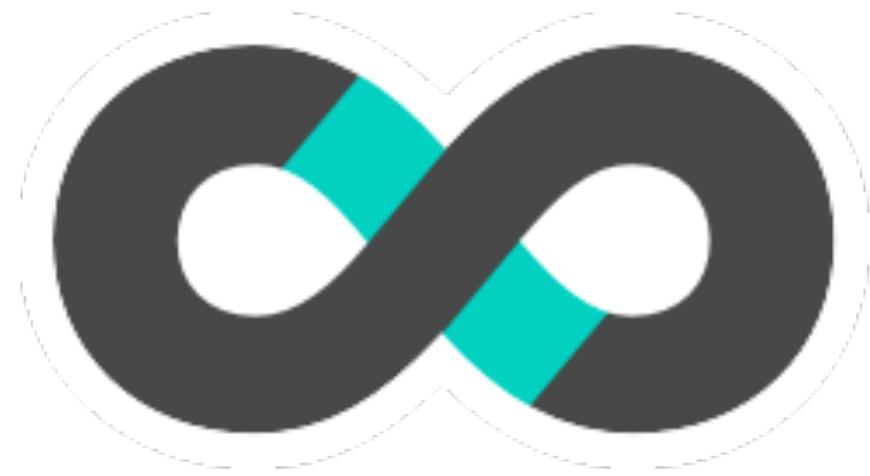
At the bottom, there are two buttons: "Visualize" and ".CSV". To the right of these buttons is a "Search Results" input field.

The main area displays the query results in a table:

Agency	AnnualSalary
OED-Employment Dev (031)	56705
States Attorneys Office (045)	75500



Building Dashboards with Drill and Superset



- Click on Visualize

Visualize

Chart Type: Distribution - Bar Chart

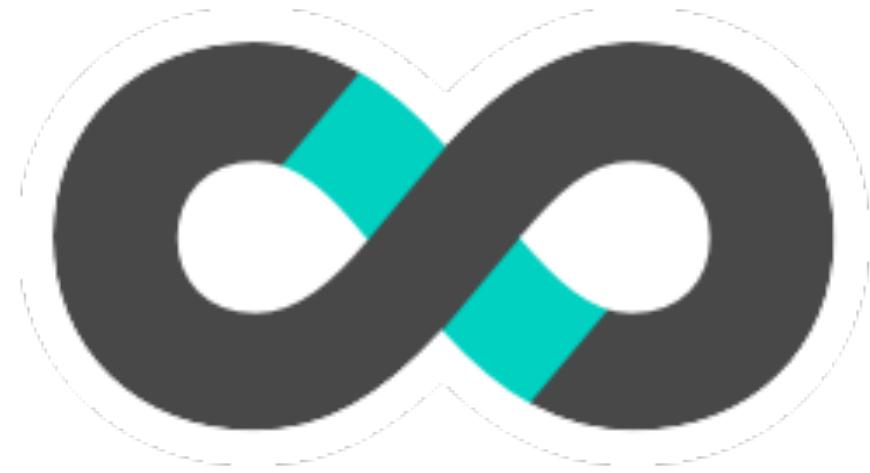
Datasource Name: admin-Drill-Untitled Query 2-Bkx0VHrP5M

column	is_dimension	is_date	agg_func
Agency	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Select...
AnnualSalary	<input checked="" type="checkbox"/>	<input type="checkbox"/>	AVG(x)

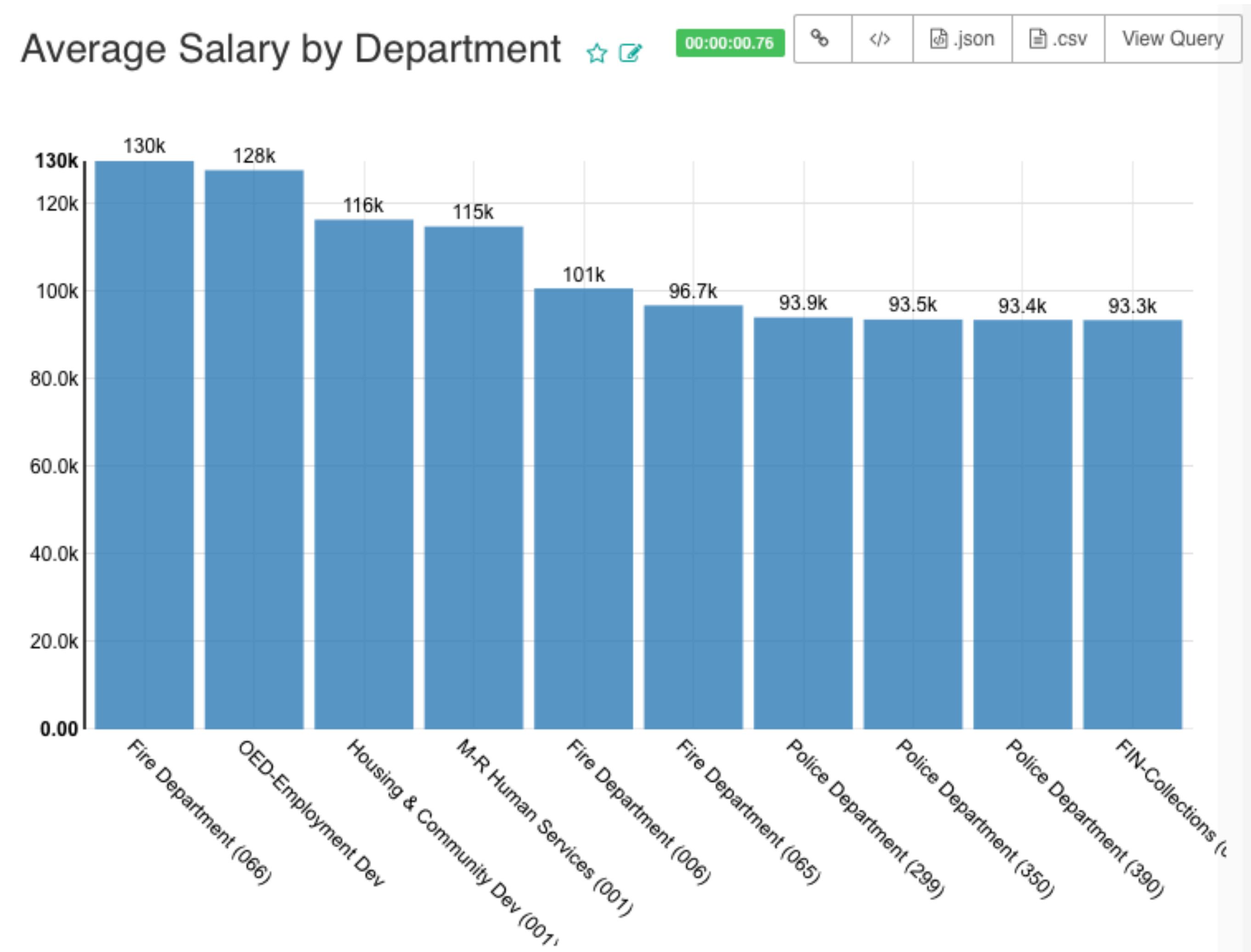
Visualize

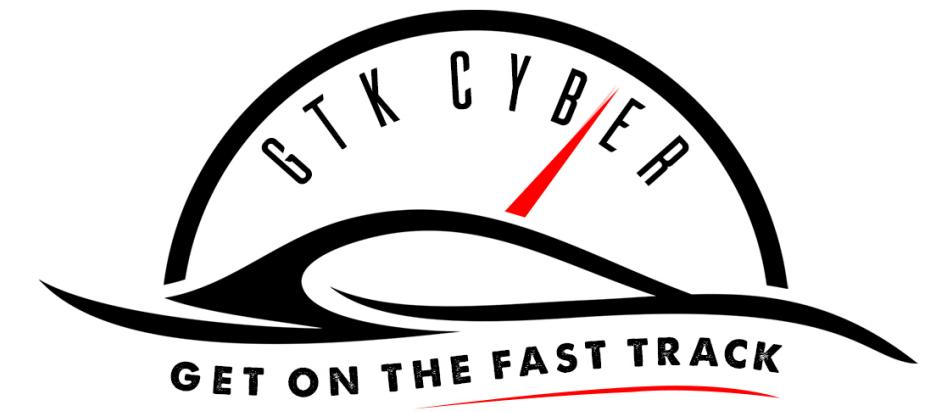


Building Dashboards with Drill and Superset



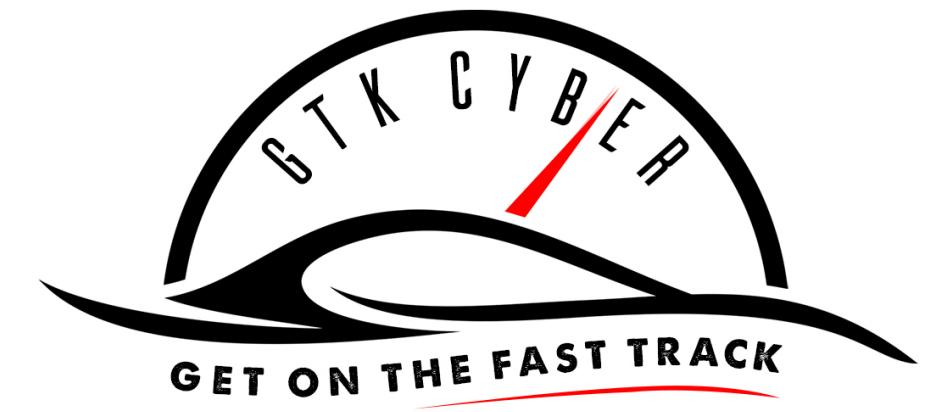
- Voila!



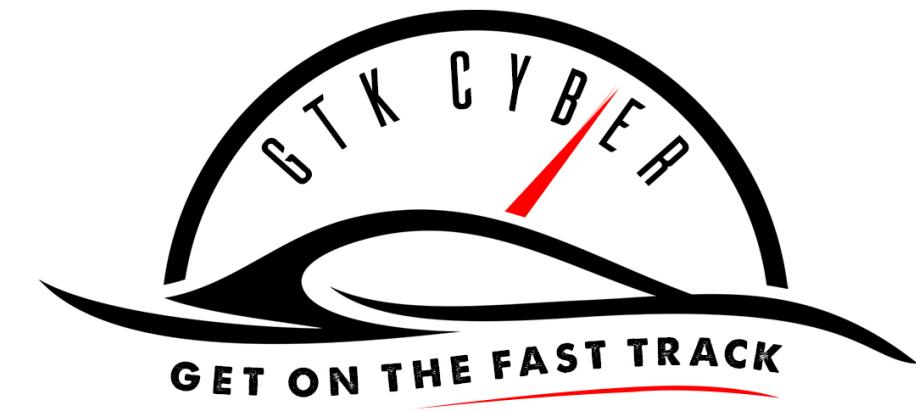


Conclusion

- 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 8: Drill Demonstration Worksheet.