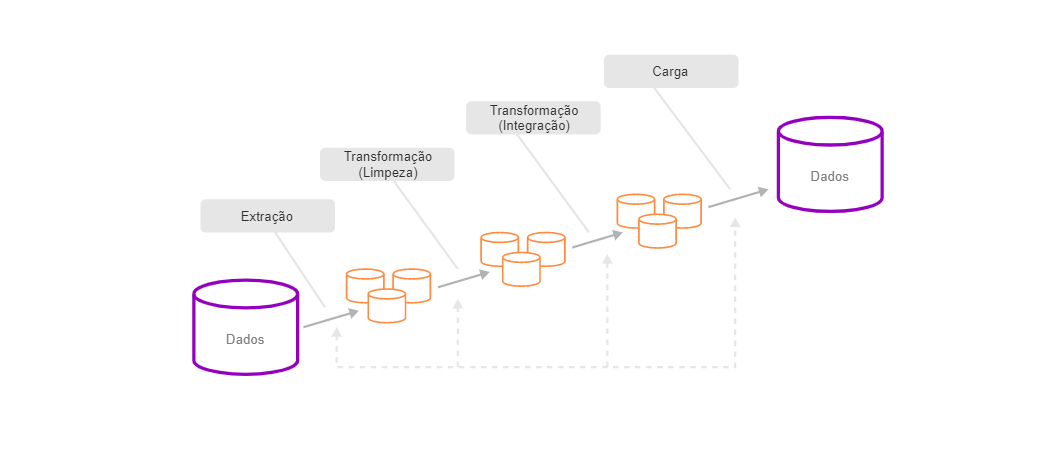
**Aula 3 - Integração dos Dados**

A integração de dados, na esfera analítica (OLAP), consiste em transportar as mais variadas fontes (ERP, CRM, Cubo, Redes Sociais, Logs, Eventos) de seus ambientes de origem e combiná-las, a fim de obter informações que suportem à tomada de decisão, a busca de padrões, a obtenção de *insights*, carregando-as em seu ambiente de destino.



### Tipos de Integração – Ingestão de Dados

O termo ingestão é comumente utilizado em projetos de big data quando o assunto é integrar dados. Isso porque ingerir significa buscar os dados em sua origem e carregá-los em seu destino. Perceba que isso é típico da zona Raw, na qual os dados entram de forma bruta, tal como vieram da fonte. Portanto, embora o termo ingestão seja utilizado de forma indiscriminada para qualquer processo de movimentação dos dados, o correto é afirmarmos que a ingestão pressupõe a carga do dado sem tratamentos ou modificações. De forma simplista, podemos dizer que ingerir é fazer *“copy/paste”* dos dados.

Quando os cliques de um usuário em uma página web vão sendo carregados em um *Data Lake*, é adequado dizer que eles estão sendo ingeridos. Assim como, quando os logs de funcionamento de um sistema são gravados na *Raw Data*, eles também estão sendo ingeridos.

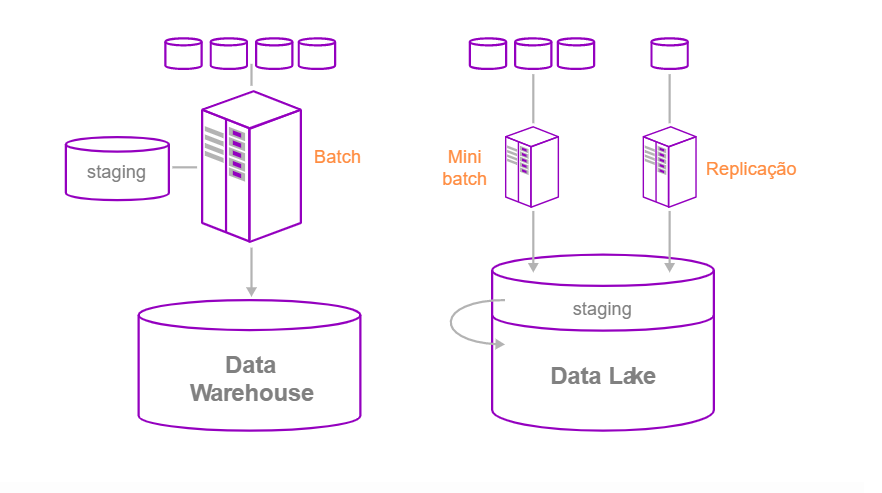
Mas, e quando antes ou mesmo após a ingestão, existe alguma operação que transforma o dado (limpeza, qualificação, enriquecimento, cálculos, conversões), que nome damos a esse processo?

### Tipos de Integração – ETL *versus* ELT

O processo completo, que não somente move o dado, mas o transforma também, é intitulado pelo seu acrônimo, **E**xtract, **T**ransform and **L**oad - ETL.

Ele inclusive aponta para uma sequência saudável que visa não concorrer com a produção. A ideia aqui é retirar os dados da origem, buscando impactar o mínimo possível o sistema transacional, carregá-los em uma área conhecida como *Staging*, área que pode ser chamada de temporária, trabalhar esse dado, e quando ele estiver pronto, carregá-lo na produção.

Contudo, é possível também, fazer uma inversão na ordem desse processo. Pode-se movimentar primeiro o dado, ou seja, ingeri-lo no destino, o que equivale ao **E**xtract e o **L**oad, e já no destino, na base de produção fazer as modificações necessárias no dado, o **T**ransform. Essa sequência modificada, é chamada de ELT. Observe a imagem “ETL x ELT”:



Note que a grande diferença consiste no fato das transformações serem feitas em momentos distintos.

E embora, não se possa estabelecer como regra que o ETL está para o Data Warehouse e o ELT para o Lake, na maioria das vezes é o que ocorre.

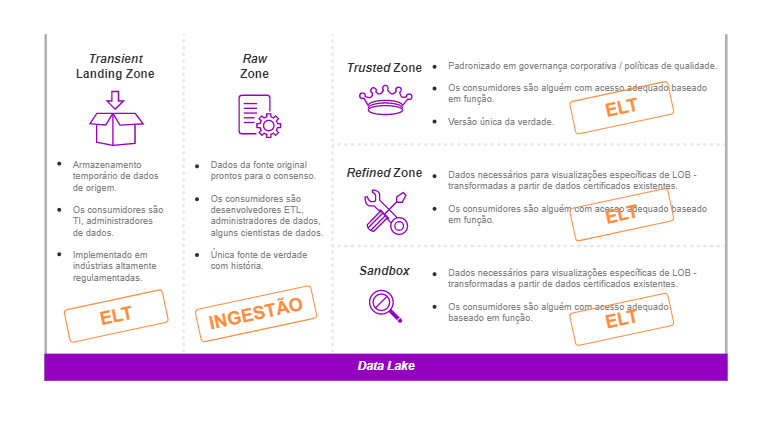
Entenda, o ELT é mais prático. Você move o dado da origem para o destino, e já o modifica para ser consumido. Contudo, esse momento de transformação pode onerar a produção, o que significa que é necessário termos um ambiente de alto desempenho, que suporte as transformações e o consumo das informações ocorrendo em paralelo. Um ambiente de big data necessariamente precisa ser de alta performance, o que combina com o ELT.

Já quando a estrutura física na qual o ambiente foi desenhado é mais tradicional, com um volume médio de dados, estruturados, uma realidade mais comum em sistemas de Business Intelligence (Data Warehouses), o ETL sempre será uma medida mais prudente!

Podemos fechar o nosso raciocínio dizendo que, na maioria das vezes, o ambiente de um DW não suporta o ELT, assim como, seria desperdício de tempo e espaço fazer ETL em um Data Lake.

Conectando o nosso novo entendimento, as zonas do lake aprendidas no primeiro capítulo, já podemos fazer distinção do tipo de carga que ocorre em cada uma delas:

Na *Raw Data*, ocorre somente ingestão, os dados são empilhados, acumulados de forma bruta.

Já nas demais zonas (*Transient, Trusted, Refined e Sandbox*), como os dados sofrem algum tipo de modificação, ocorre ELT.

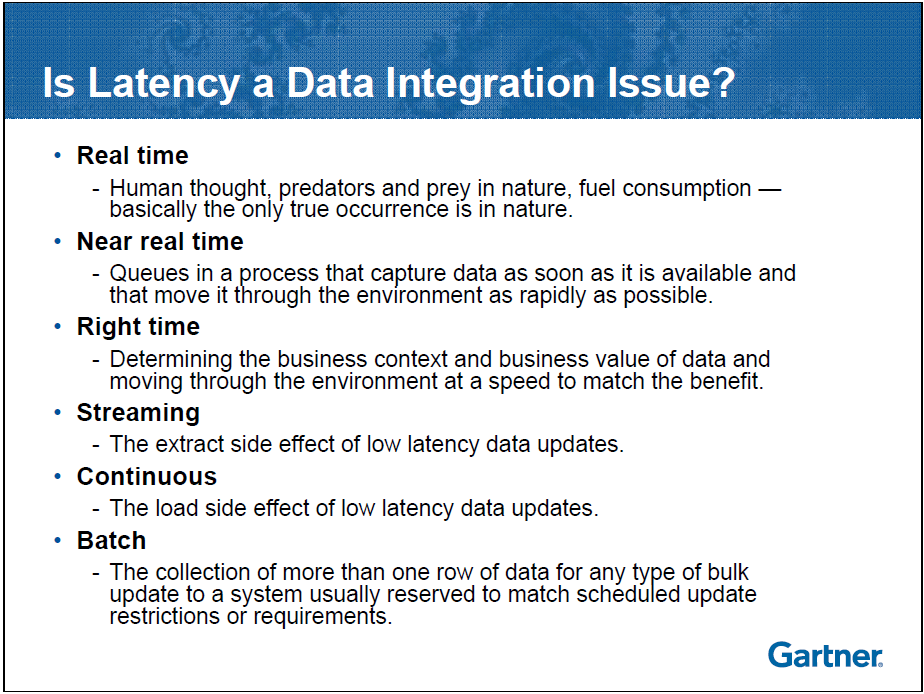
### Latência

Ainda olhando para a figura “ETL x ELT”, podemos notar uma diferença na latência da carga realizada. Perceba que no DW as cargas estão marcadas como batch, já no DL como mini batch e replicação. Aqui também cabe a ressalva em dizer que isso é uma tendência, e não uma regra.

É mais comum em BI, acumular os dados transacionais ao longo do dia, e deixar para fazer cargas maiores (em lote) em períodos de menor acesso ao sistema, tal como, a madrugada. Novamente isso se dá muitas vezes pelos recursos físicos serem mais limitados.

Em big data, com uma infraestrutura preparada para alta velocidade, a carga dos dados pode ocorrer ao longo do dia, não há necessidade de aguardar, podem ser lotes menores de tempo em tempo, ou mesmo uma carga contínua, uma espécie de *streaming* de dados que permite análises e tomadas de decisão muito próximas do tempo real.

A figura “latências” mostra as frequências mais comuns para carga de dados:



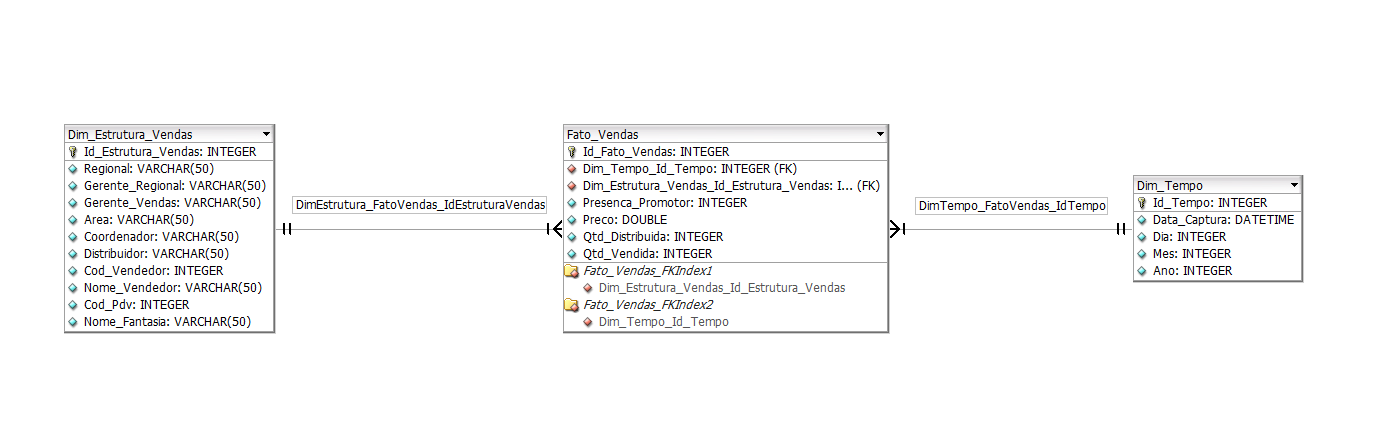
### Ferramentas Tradicionais de Integração

* **Data Stage – IBM**
* **Power Center – Informatica**
* **Integration Services – Microsoft**
* **ODI – Oracle**
* **Pentaho Data Integration**
* **Alteryx *(Data Blending)***
* **Trifacta *(Data Wrangling)***

### Ferramentas dentro do ecossistema Hadoop

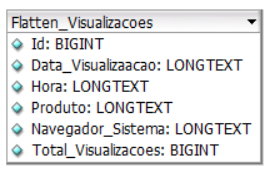
* + **Flume**
  + **Sqoop**
  + **Kafka**
  + **Nifi**

**Case de Trade Marketing – Modelagem de Dados**

* Modelo Dimensional para os Dados Estruturados

*\* Os dados integrados via ETL para esse modelo serão fornecidos.*

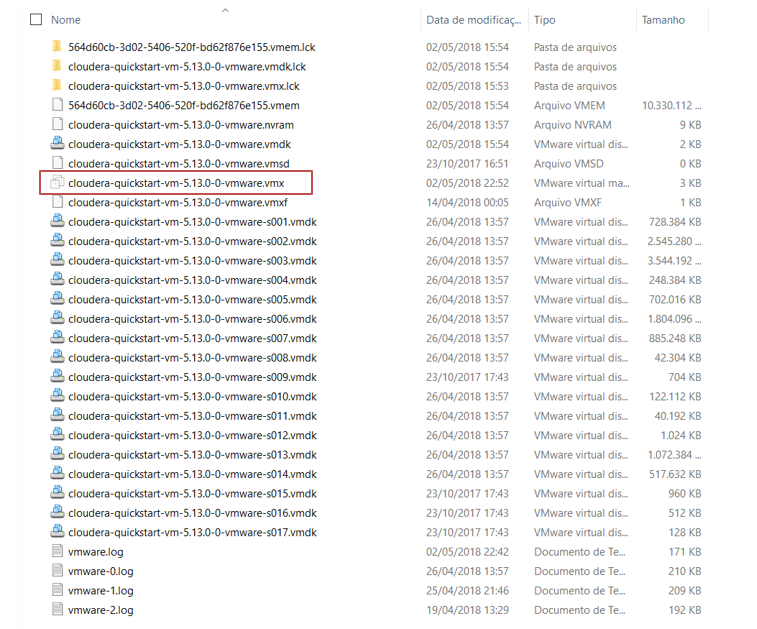
* Modelo Flatten para os Dados Semiestruturados



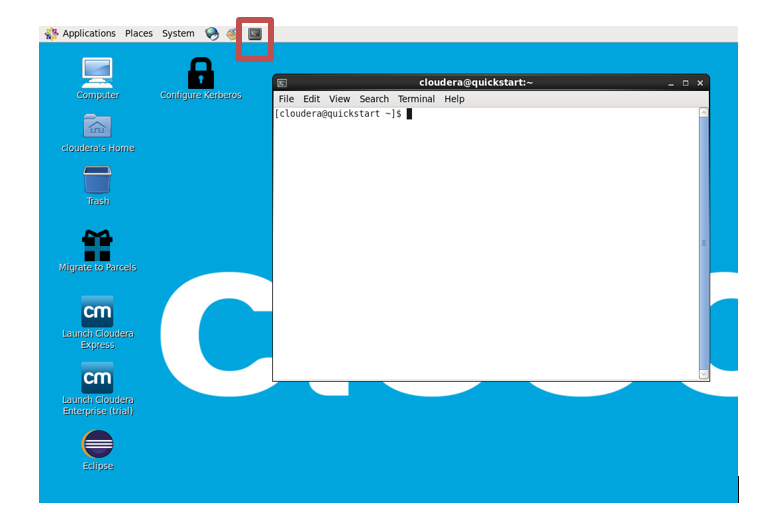
**Hands On 6 - Ingestão e Transformação dos Dados Semiestruturados via Hadoop (Hive e Impala)**

Parte I – Uso do HIVE para estruturação dos dados

1. Inicie a partir do Desktop a VM “Dados”:



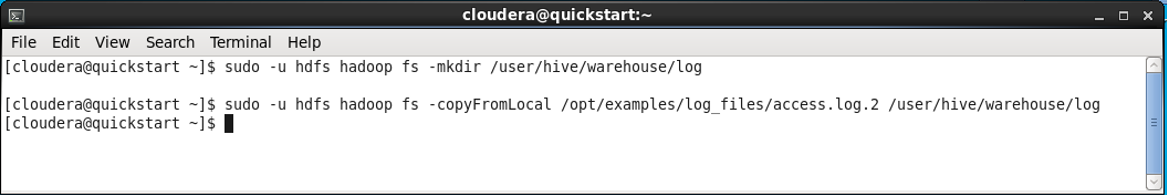
1. Abra o terminal para realizar o “Bulk Upload” dos dados semiestruturados. O arquivo se encontra em: /opt/examples/log\_files/access.log.2



- Digite o seguinte código:

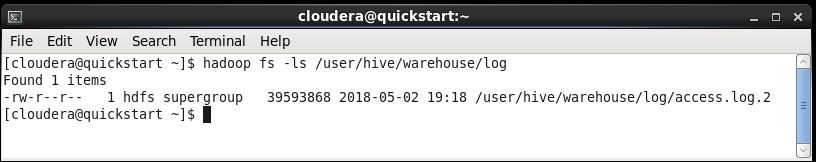
sudo -u hdfs hadoop fs -mkdir /user/hive/warehouse/log

sudo -u hdfs hadoop fs -copyFromLocal /opt/examples/log\_files/access.log.2 /user/hive/warehouse/log

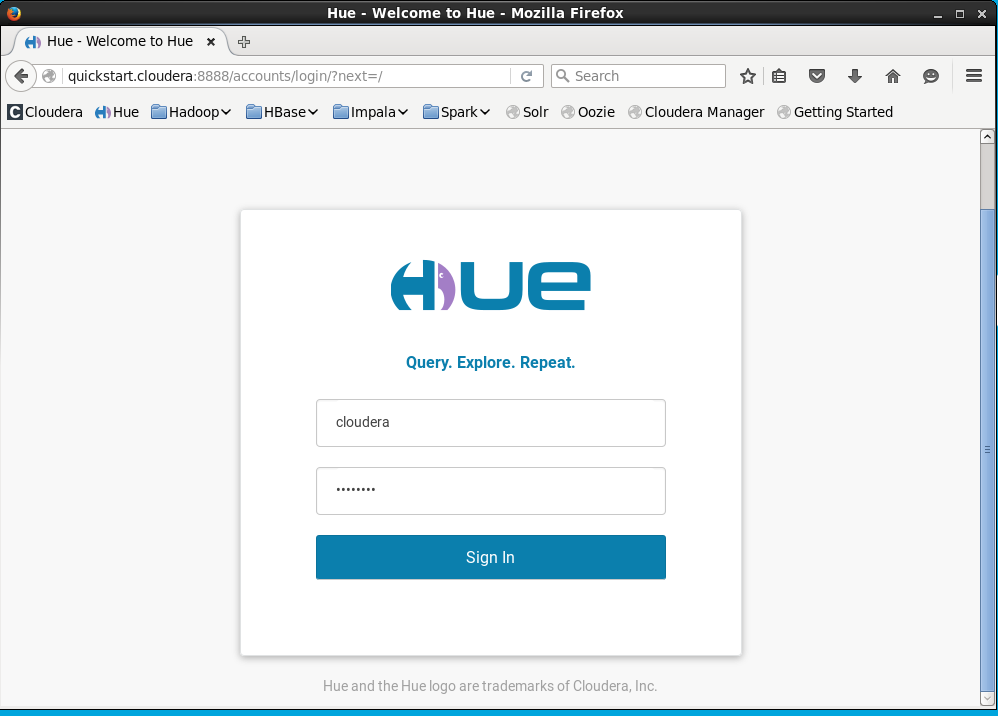


- Para verificar a carga do arquivo no HDFS, digite:

hadoop fs -ls /user/hive/warehouse/log



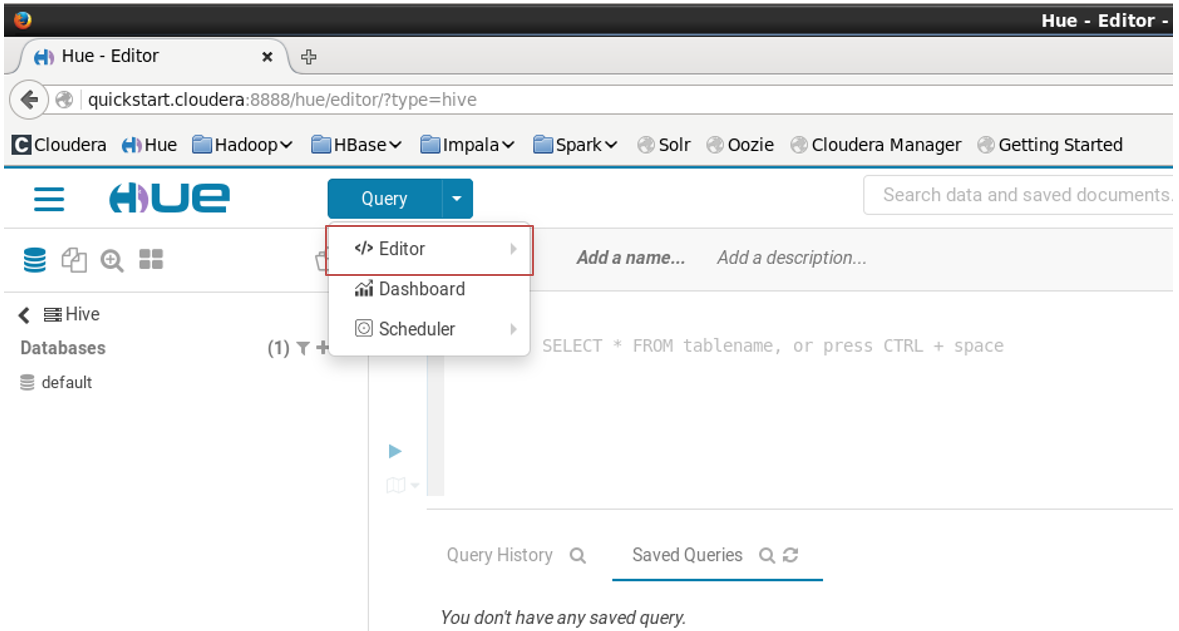
1. Para criar, a partir do arquivo de log, uma tabela no “Hive”, acesse a interface “Hue” via “Firefox”:



Usuário: cloudera

Senha: cloudera

- Selecione o ambiente de “Query Editor” do “Hive”:



1. Crie uma tabela intermediária no Hive, utilizando expressões regulares para organizar os dados semiestruturados em colunas:

CREATE EXTERNAL TABLE log\_intermediario (

ip STRING,

data STRING,

metodo STRING,

url STRING,

http\_versao STRING,

codigo1 STRING,

codigo2 STRING,

traco STRING,

navegador\_sistema STRING)

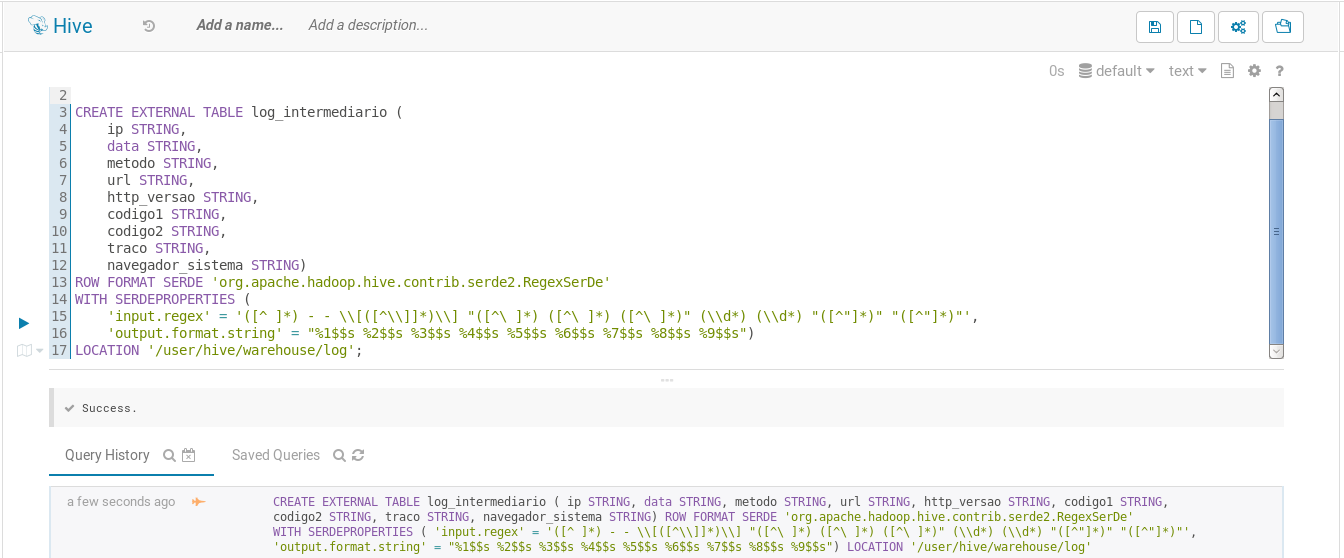
ROW FORMAT SERDE 'org.apache.hadoop.hive.contrib.serde2.RegexSerDe'

WITH SERDEPROPERTIES (

'input.regex' = '([^ ]\*) - - \\[([^\\]]\*)\\] "([^\ ]\*) ([^\ ]\*) ([^\ ]\*)" (\\d\*) (\\d\*) "([^"]\*)" "([^"]\*)"',

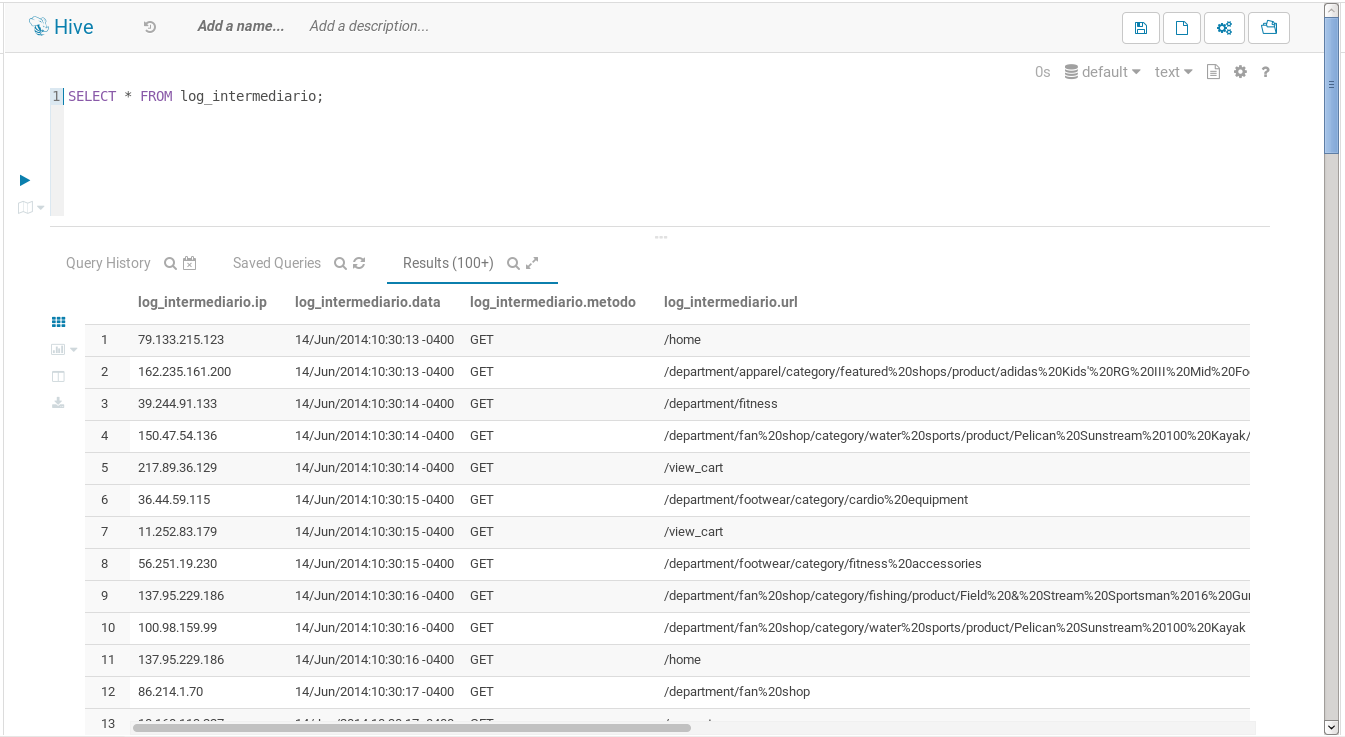
'output.format.string' = "%1$$s %2$$s %3$$s %4$$s %5$$s %6$$s %7$$s %8$$s %9$$s")

LOCATION '/user/hive/warehouse/log';



- Consulte o resultado:

SELECT \* FROM log\_intermediario;



1. Crie uma tabela definitiva no Hive, selecionando as colunas que serão utilizadas para compor a tabela flatten:

CREATE EXTERNAL TABLE log (

ip STRING,

data STRING,

url STRING,

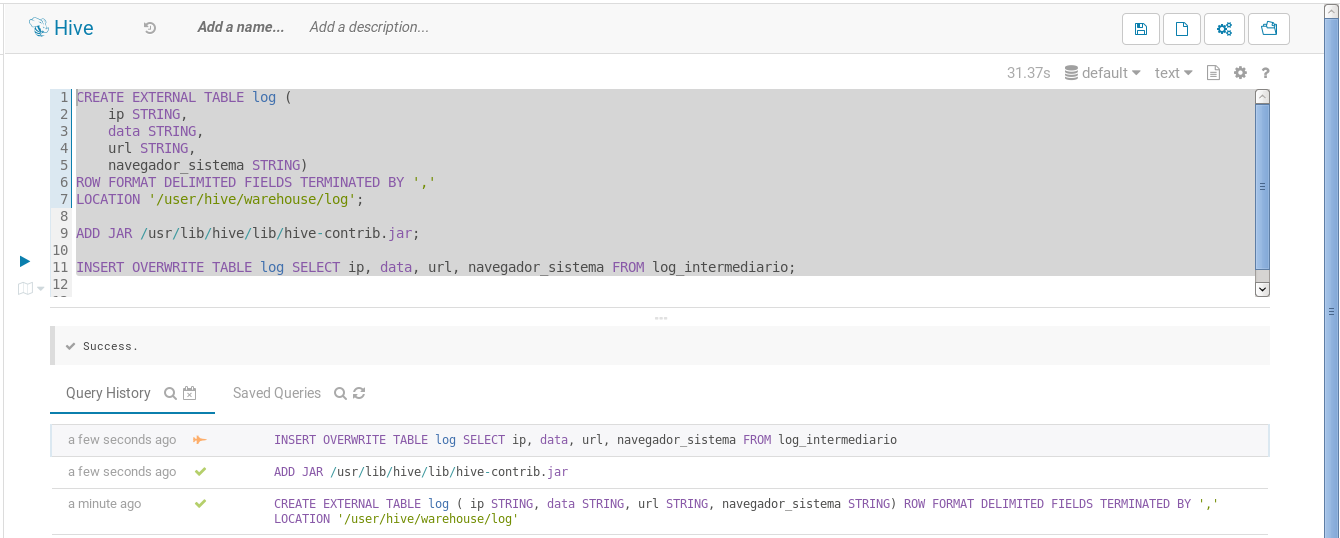
navegador\_sistema STRING)

ROW FORMAT DELIMITED FIELDS TERMINATED BY ','

LOCATION '/user/hive/warehouse/log';

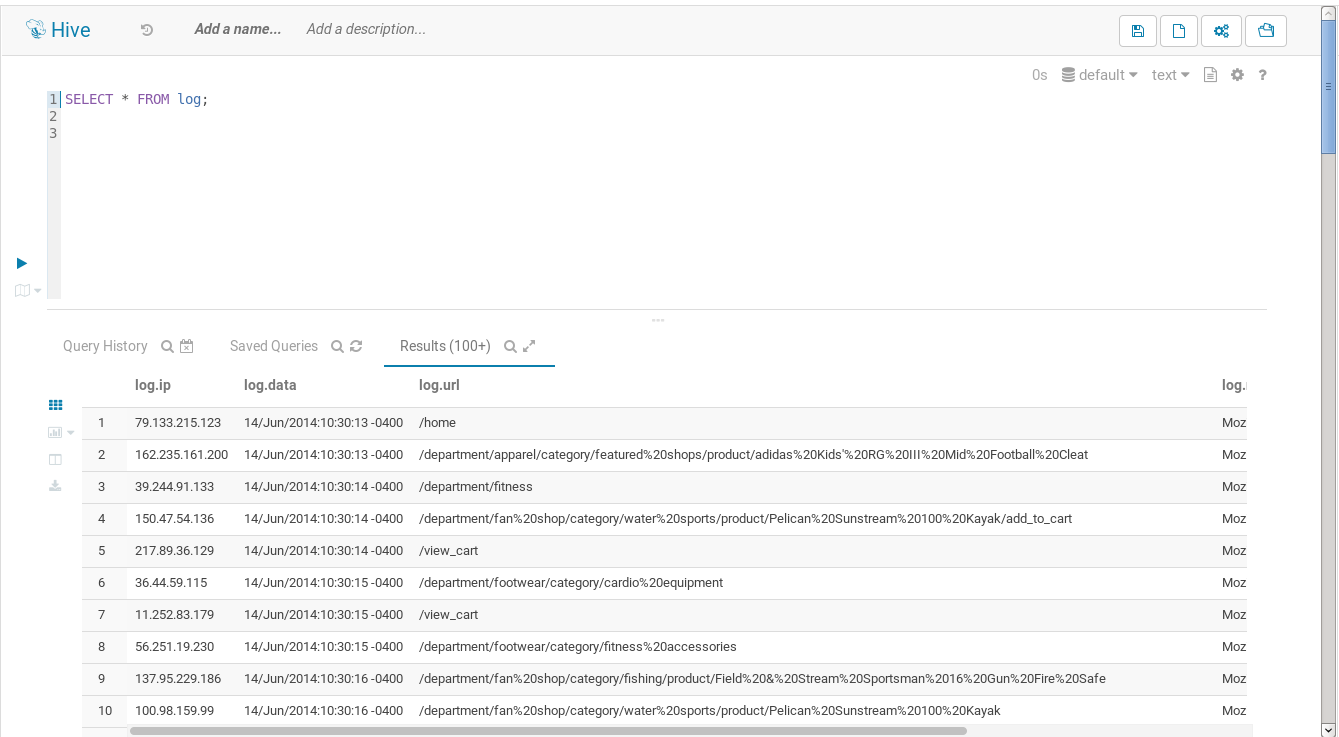
ADD JAR /usr/lib/hive/lib/hive-contrib.jar;

INSERT OVERWRITE TABLE log SELECT ip, data, url, navegador\_sistema FROM log\_intermediario;



- Consulte o resultado:

SELECT \* FROM log;



Parte II – Sumarização dos dados via Impala

1. Altere o “Query Editor” para “Impala”. Crie a tabela que receberá os dados agregados e execute a consulta para preencher essa tabela:

invalidate metadata;

CREATE TABLE flatten\_visualizacoes (id bigint, data\_visualizacao string, hora string, produto string, navegador\_sistema string,

total\_visualizacoes bigint);

insert into table flatten\_visualizacoes

select row\_number() over(order by count(\*) desc) as id,

substring(`data`,1,11) as data\_visualizacao,

substring(`data`,13,8) as hora,

replace(replace(substring(url, instr(url,'product',1) + 8, 100),'%20', ' '),'/add\_to\_cart','') as produto,

substring(navegador\_sistema,14,instr(navegador\_sistema,';',1)-14) as sistema\_operacional,

count(\*) as visualizacoes\_produto

from default.log

where url like '%\/product\/%'

group by substring(`data`,1,11), substring(`data`,13,8), replace(replace(substring(url, instr(url,'product',1) + 8, 100),'%20', ' '),'/add\_to\_cart',

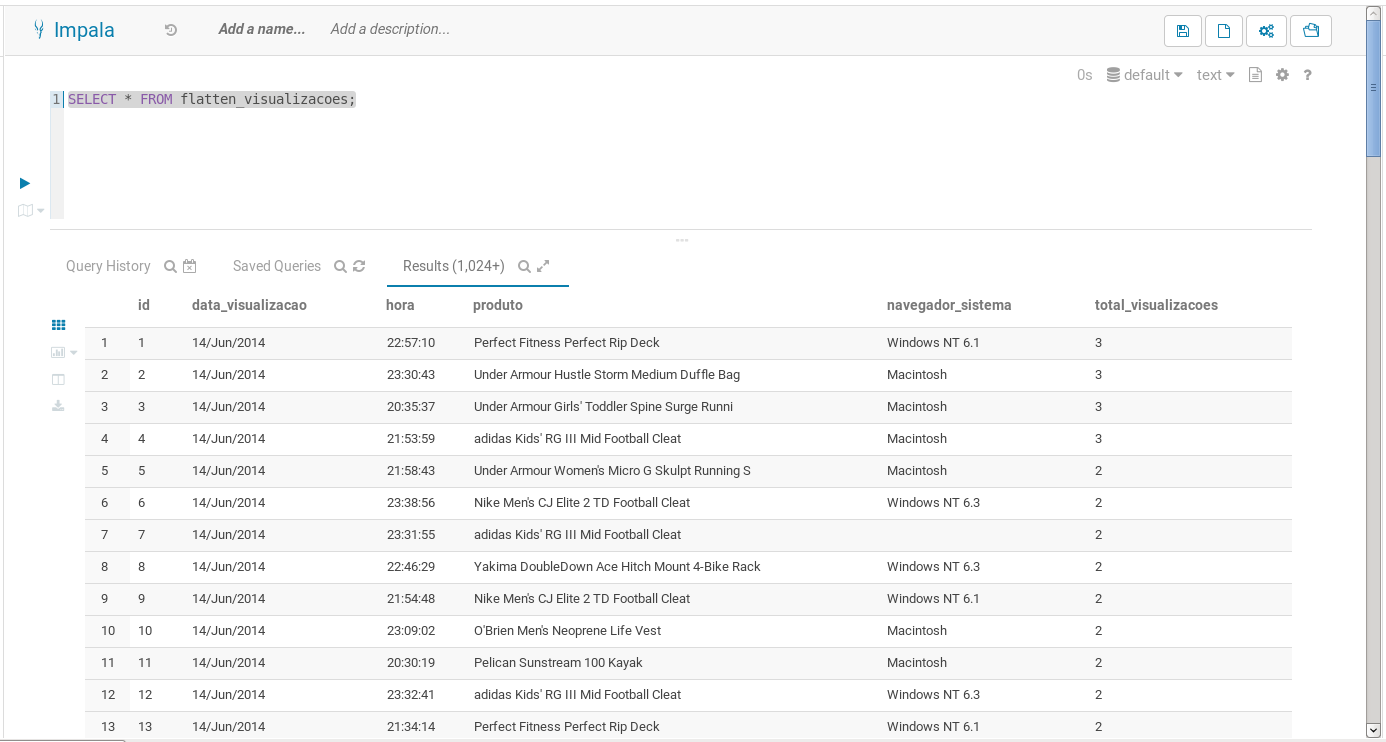
''), substring(navegador\_sistema,14,instr(navegador\_sistema,';',1)-14)

order by count(\*) desc;



- Consulte o resultado:

SELECT \* FROM flatten\_visualizacoes;



1. Para exportar os dados em formato tabular e separado por vírgula, vamos gerar a tabela final dentro do Hive:

CREATE EXTERNAL TABLE flatten\_visualizacoes\_hive (

id bigint,

data\_visualizacao STRING,

hora STRING,

produto STRING,

navegador\_sistema STRING,

total\_visualizacoes bigint)

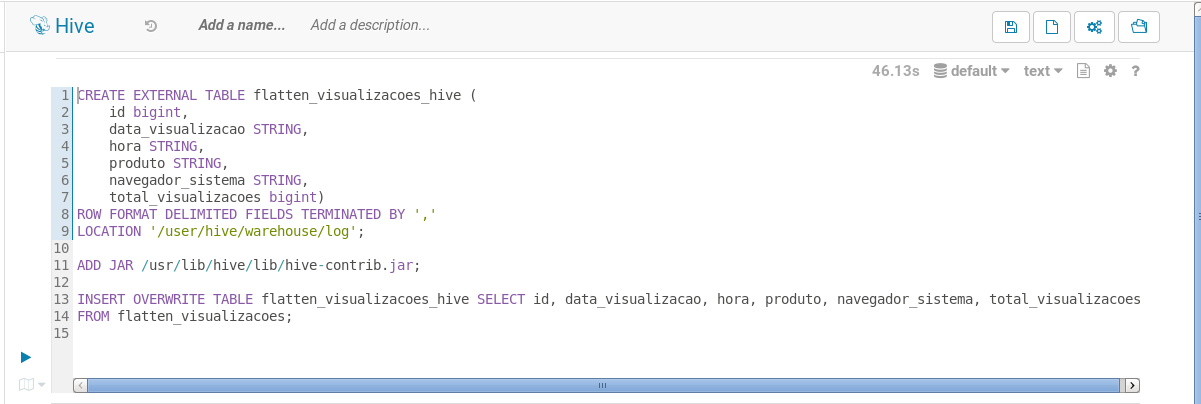
ROW FORMAT DELIMITED FIELDS TERMINATED BY ','

LOCATION '/user/hive/warehouse/log';

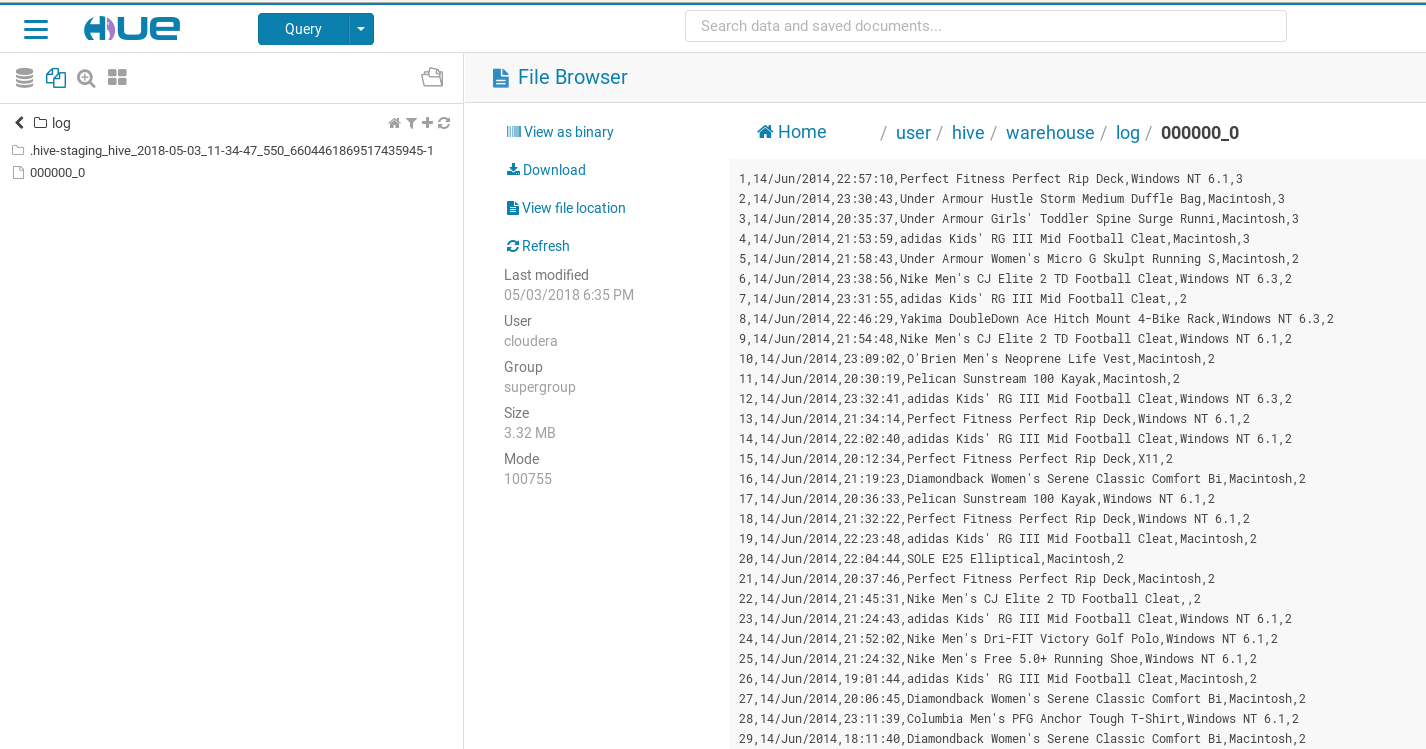
ADD JAR /usr/lib/hive/lib/hive-contrib.jar;

INSERT OVERWRITE TABLE flatten\_visualizacoes\_hive SELECT id, data\_visualizacao, hora, produto, navegador\_sistema, total\_visualizacoes

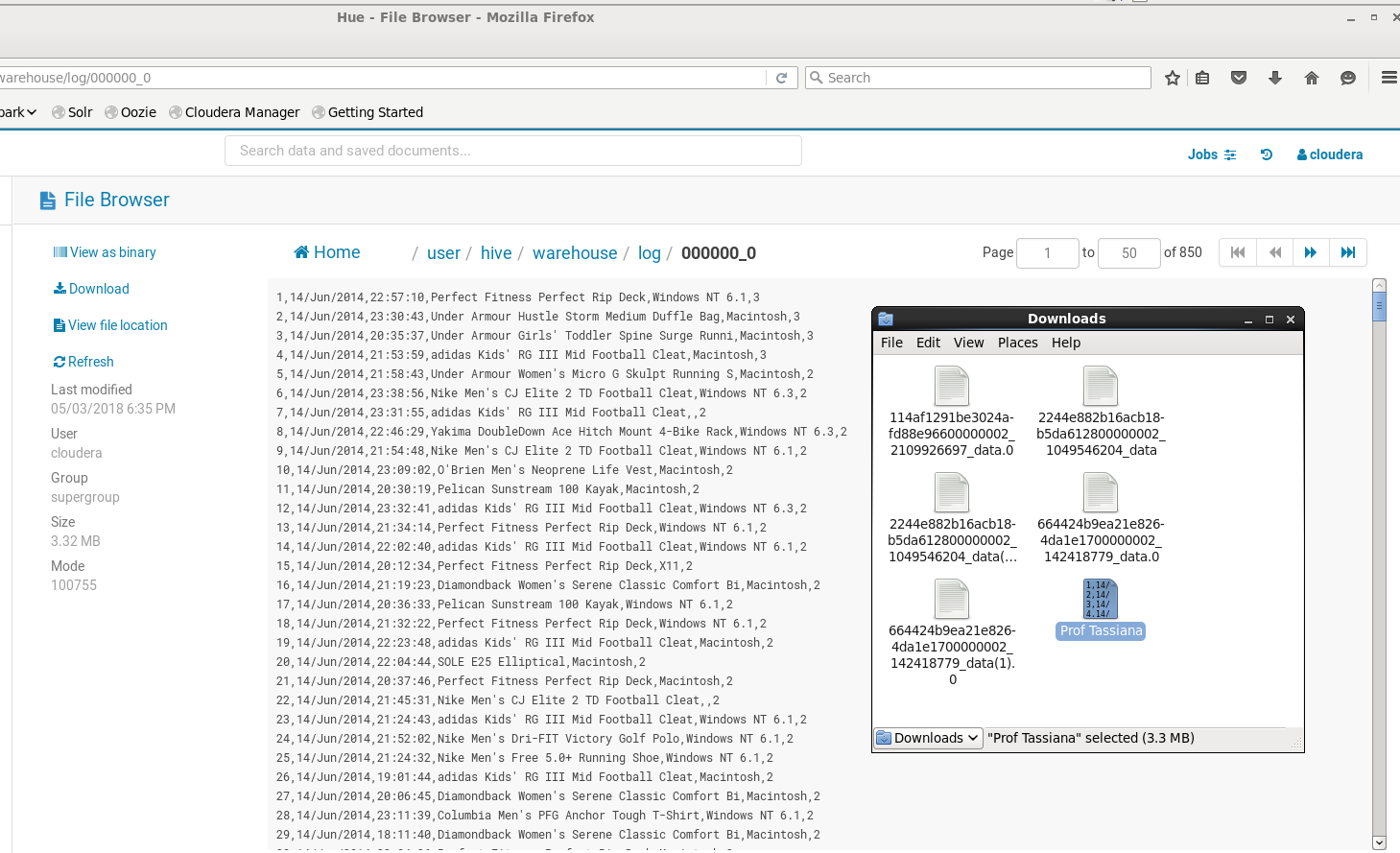
FROM flatten\_visualizacoes;



- Abra o arquivo gerado no HDFS dentro da pasta hive/warehouse/log, e clique em download:



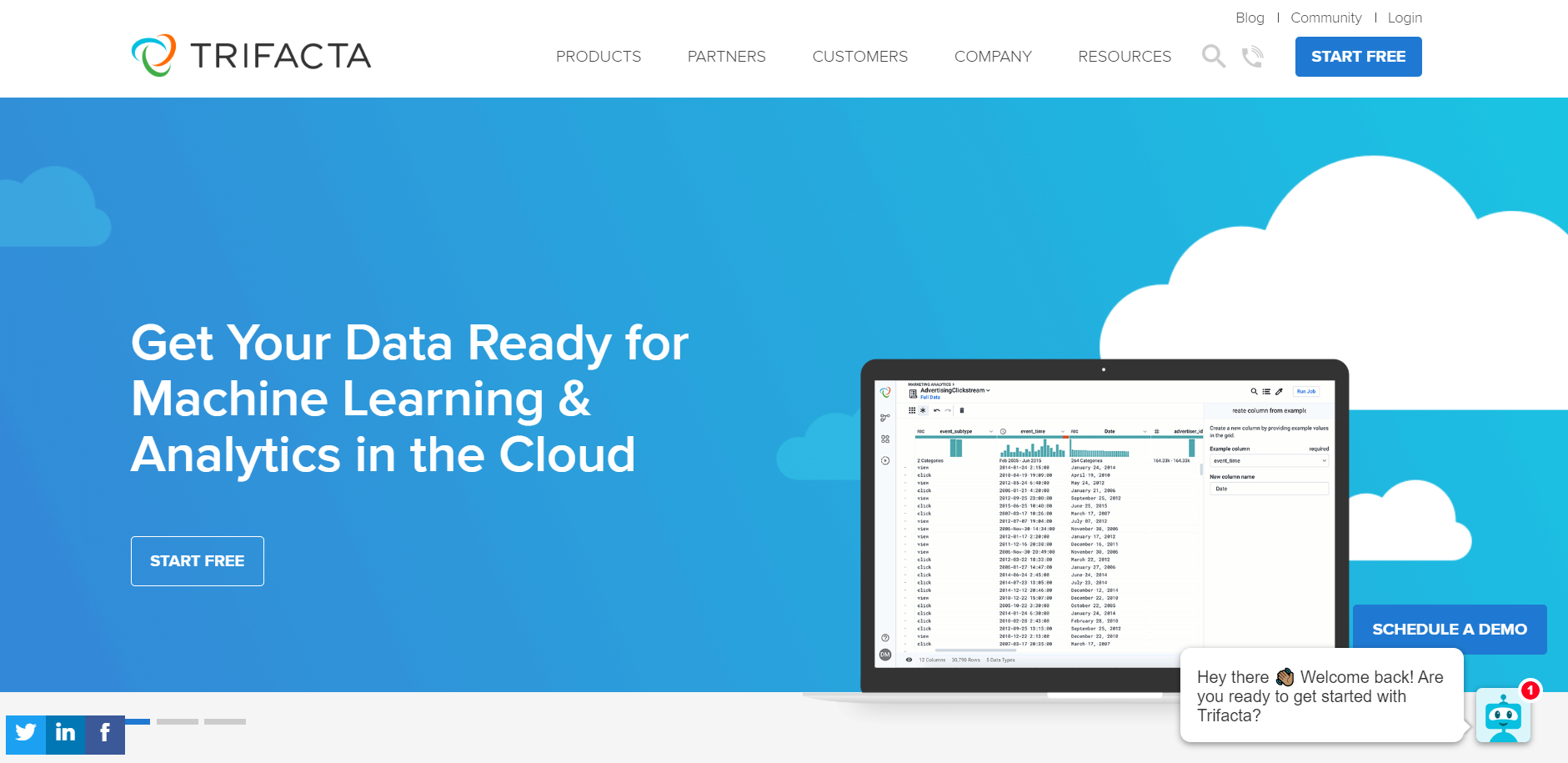
- Pronto! Seu arquivo já pode ser trabalhado dentro da camada de visualização!



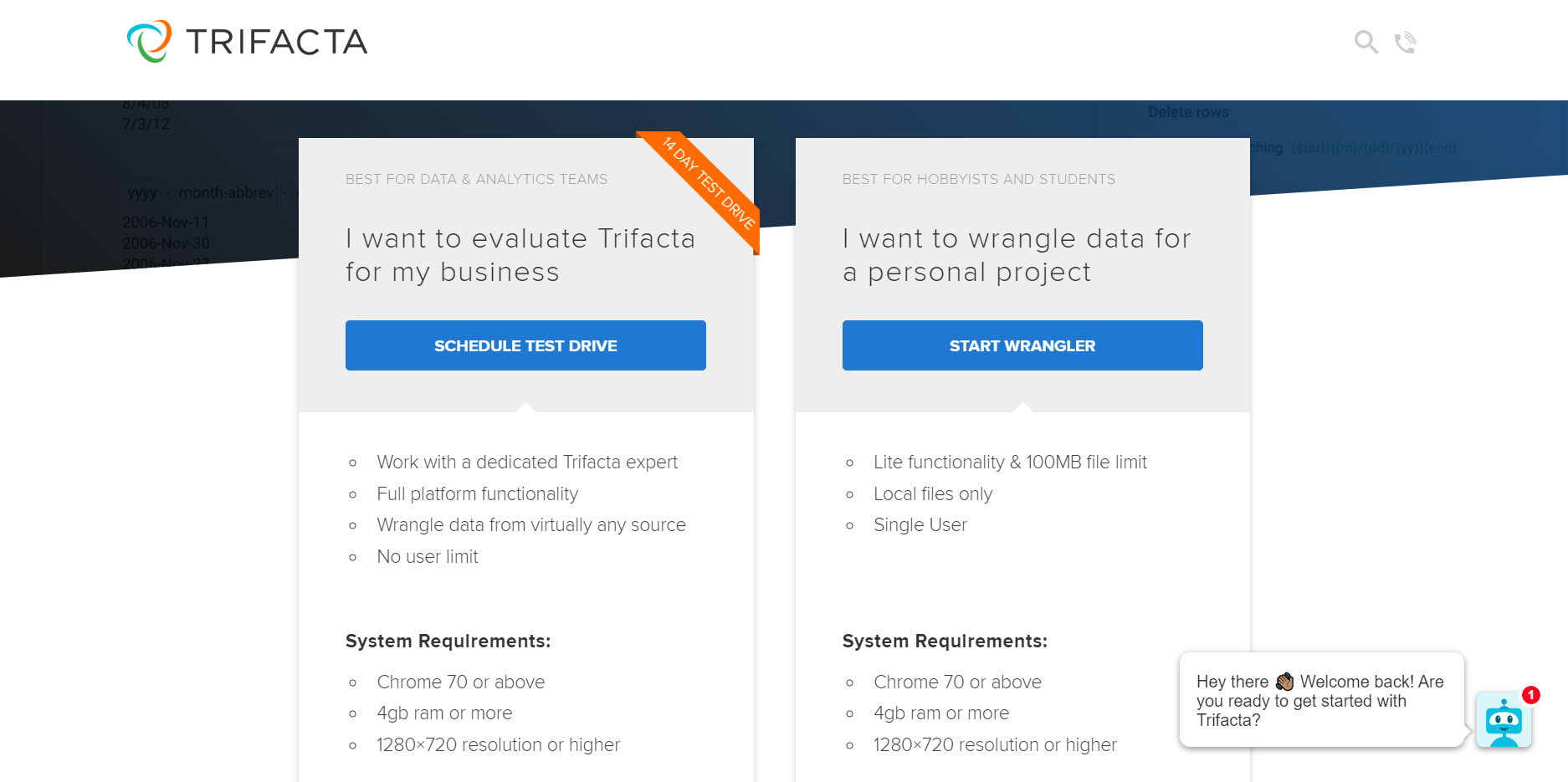
Como evidência, renomei o arquivo com rm1\_rm2\_rm3\_rmx, print essa janela final, tal como na imagem acima, e cole no item 3.2 do documento “Especificação da Solução de Dados.doc”.

**Hands On 6 – Transformação dos Dados Semiestruturados via Trifacta**

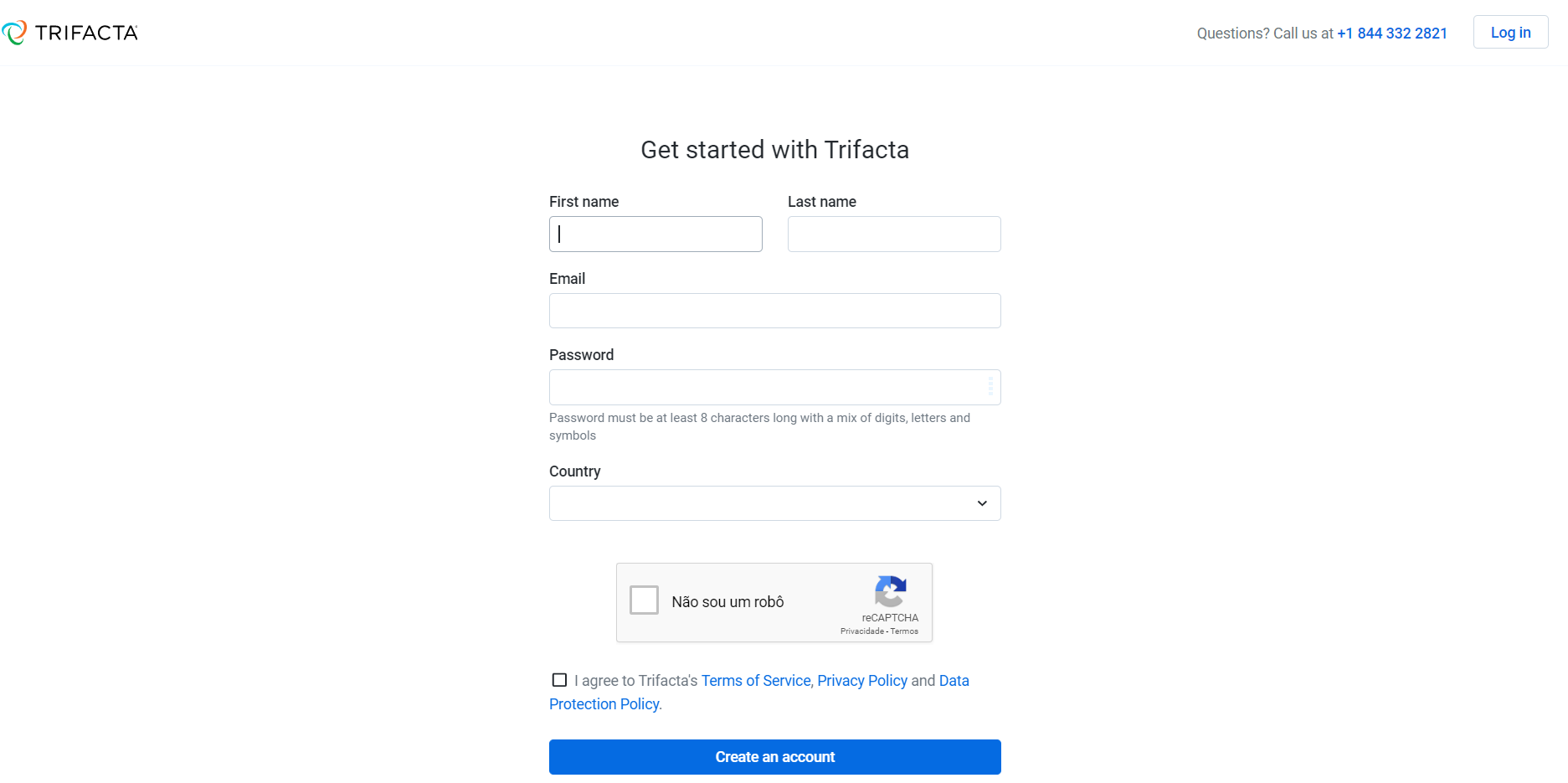
* Acesse <https://www.trifacta.com/> e clique em :



* Escolha a opção à esquerda , que permite a criação de um “personal project” gratuito:



* Crie uma conta e faça login:



Pronto! Agora no ambiente de construção de Jobs da Trifacta, suba o arquivo access.log, crie e execute um Job chamado Flatten\_Visualizações\_rm1\_rm2\_rm3. Como evidência print o resultado do processo e cole no item 3.2 do documento “Especificação da Solução de Dados.doc”.

