

LAB EXTRA

HUE: Hive - Impala

PARTE 1. Ingestar y Consultar Data que es Relacional

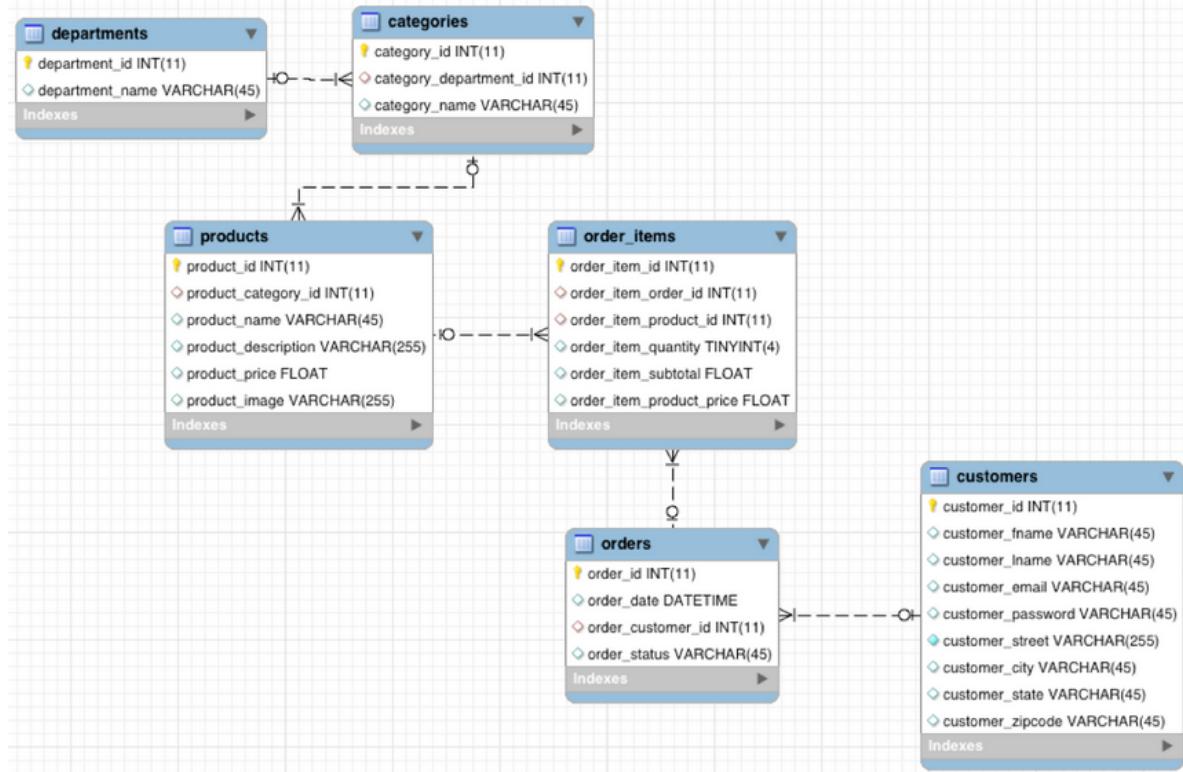
En este escenario, la pregunta comercial de **DataCo** es:

¿Qué productos les gusta comprar a nuestros clientes?

Para responder a esta pregunta, los primero a hacer podría ser mirar los datos de la transacción, lo que debería indicar lo que los clientes realmente compran y les gusta comprar.

Probablemente sea algo que puedes hacer en tu entorno RDBMS habitual, pero un beneficio de la plataforma Cloudera es que puedes hacerlo a mayor escala a un costo menor, en el mismo sistema que también puedes usar para muchos otros tipos de análisis.

Lo que este ejercicio demuestra es cómo hacer exactamente lo mismo que se hace con las bases de datos tradicionales, pero en HDFS.



Para analizar los datos de la transacción necesitamos incorporarlos al Sistema de Archivos Distribuidos de Hadoop (HDFS). Usaremos Apache Sqoop que transfiere fácilmente los datos estructurados de un RDBMS (en este caso MySQL) a HDFS, mientras se preserva la estructura. Eso nos permite consultar los datos, pero no interferir o interrumpir ninguna carga de trabajo regular.

Con algunos parámetros de configuración adicionales, podemos dar un paso más allá y cargar estos datos relacionales directamente en Hive para ser consultado por **Impala** (el motor de consulta analítica de código abierto incluido en Hadoop).

Dado que es posible que queramos aprovechar la potencia del formato de archivo Apache **Avro** para otras cargas de trabajo en el clúster (ya que Avro es un formato de archivo optimizado de Hadoop), tomaremos algunos pasos adicionales para cargar estos datos en Hive usando el formato de archivo Avro, y que así, esté disponible para Impala y otras cargas de trabajo.

1. Para tomar **toda la base de datos retail_db desde mysql y colocarla en Hive (warehouse centralizado)** usaremos **Sqoop**. Debes abrir un terminal e iniciar el job de Sqoop:

```
$ sqoop import-all-tables -m 1 --connect jdbc:mysql://localhost/retail_db --username=retail_dba --password=cloudera --compression-codec=snappy --as-parquetfile --warehouse-dir=/user/hive/warehouse --hive-import
```

Este comando puede tardar un tiempo en completarse, pero está haciendo mucho. Está lanzando jobs de MapReduce para extraer los datos de la base de datos MySQL y escribir los datos en HDFS, distribuidos a través del clúster en formato Apache **Parquet**. También está creando tablas para representar los archivos HDFS en Apache Hive con un esquema coincidente.

Parquet es un formato diseñado para aplicaciones analíticas en Hadoop. En lugar de agrupar sus datos en filas como los formatos de datos típicos, los agrupa en columnas. Esto es ideal para muchas consultas analíticas en las que, en lugar de recuperar datos de registros específicos, está analizando las relaciones entre variables específicas en muchos registros. Parquet está diseñado para optimizar el almacenamiento de datos y la recuperación en estos escenarios.

Una vez que se completa el comando, podemos confirmar que nuestros datos de las 6 tablas de la base relacional se importaron a HDFS en el warehouse Hive:

```
[cloudera@quickstart ~]$ hdfs dfs -ls /user/hive/warehouse/
```

```
[cloudera@quickstart ~]$ hdfs dfs -ls /user/hive/warehouse
Found 6 items
drwxrwxrwx  - cloudera supergroup      0 2021-06-30 08:22 /user/hive/warehouse/categories
drwxrwxrwx  - cloudera supergroup      0 2021-06-30 08:22 /user/hive/warehouse/customers
drwxrwxrwx  - cloudera supergroup      0 2021-06-30 08:23 /user/hive/warehouse/departments
drwxrwxrwx  - cloudera supergroup      0 2021-06-30 08:24 /user/hive/warehouse/order_items
drwxrwxrwx  - cloudera supergroup      0 2021-06-30 08:25 /user/hive/warehouse/orders
drwxrwxrwx  - cloudera supergroup      0 2021-06-30 08:26 /user/hive/warehouse/products
[cloudera@quickstart ~]$
```

Estos comandos mostrarán los directorios y los archivos dentro de ellos que componen nuestras tablas:

```
[cloudera@quickstart ~]$ hdfs dfs -ls /user/hive/warehouse/categories
Found 3 items
drwxr-xr-x  - cloudera supergroup          0 2021-06-30 08:21 /user/hive/warehouse/categories/.metadata
a
drwxr-xr-x  - cloudera supergroup          0 2021-06-30 08:22 /user/hive/warehouse/categories/.signals
-rw-r--r--  1 cloudera supergroup 1957 2021-06-30 08:22 /user/hive/warehouse/categories/c2f295e7
-3182-448a-be8f-5408ad0a79be.parquet
[cloudera@quickstart ~]$ hdfs dfs -ls /user/hive/warehouse/products
Found 3 items
drwxr-xr-x  - cloudera supergroup          0 2021-06-30 08:26 /user/hive/warehouse/products/.metadata
drwxr-xr-x  - cloudera supergroup          0 2021-06-30 08:26 /user/hive/warehouse/products/.signals
-rw-r--r--  1 cloudera supergroup 44856 2021-06-30 08:26 /user/hive/warehouse/products/44952688-a
408-4055-9216-29a0e69d1646.parquet
[cloudera@quickstart ~]$
```

Nota: El número de archivos .parquet que se muestra será igual al número de mapeadores utilizados por Sqoop. En un solo nodo solo verá uno, pero los clústeres más grandes tendrán una mayor cantidad de archivos.

Hive e Impala también le permiten crear tablas definiendo un esquema sobre los archivos existentes con las declaraciones 'CREATE EXTERNAL TABLE', similar a las bases de datos relacionales tradicionales. Pero Sqoop ya creó estas tablas para nosotros, por lo que podemos usarlas y consultarlas.

Vamos a usar la aplicación Impala de Hue para consultar nuestras tablas.

Hue proporciona una interfaz basada en web para muchas de las herramientas en Cloudera. En QuickStart VM, el **nombre de usuario del administrador de Hue es 'cloudera' y la contraseña es 'cloudera'**.

Familiarícese con el entorno siguiendo la guía inicial de Hue:

Some of the apps have a right panel too with additional information to assist you out in your data discovery.

Next

any saved query.

** En <**Query – Editor**> usted puede seleccionar los **entornos de consulta**, trabajaremos con **Impala**:

Para ahorrar tiempo durante las consultas, Impala no sondea constantemente los cambios de metadatos. **Entonces, lo primero que debemos hacer es decirle a Impala que sus metadatos están desactualizados (y que debe actualizarse)**. Entonces deberíamos ver nuestras tablas aparecer, listas para ser consultadas:

invalidate metadata;
show tables;

```
1 invalidate metadata;
2
3 show tables;
```

También puede hacer clic en el icono "Refresh Table List" a la izquierda para ver sus nuevas tablas en el menú lateral.

The screenshot shows the Hue interface in Mozilla Firefox. The title bar says "Hue - Editor - Mozilla Firefox". The address bar shows the URL "quickstart.cloudera:8888/hue/editor?editor=32". The navigation bar includes links for Cloudera, Hue, Hadoop, HBase, Impala, Spark, Solr, Oozie, Cloudera Manager, and Getting Started. The main area has a "HUE" logo and a "Query" dropdown. On the left, there's a sidebar with a "Tables" section showing "default" and several tables: movies, order_items, orders, and products. An orange circle highlights the "refresh" icon next to the "default" section. The main query editor shows the following text:

```

1 invalidate metadata;
2
3 show tables;
4
5

```

Below the editor, the results table shows the following data:

name
1 categories
2 customers
3 departments
4 movies
5 order_items
6 orders
7 products

Ahora que los datos de sus transacciones están disponibles para consultas estructuradas, es hora de abordar la pregunta comercial de DataCo.

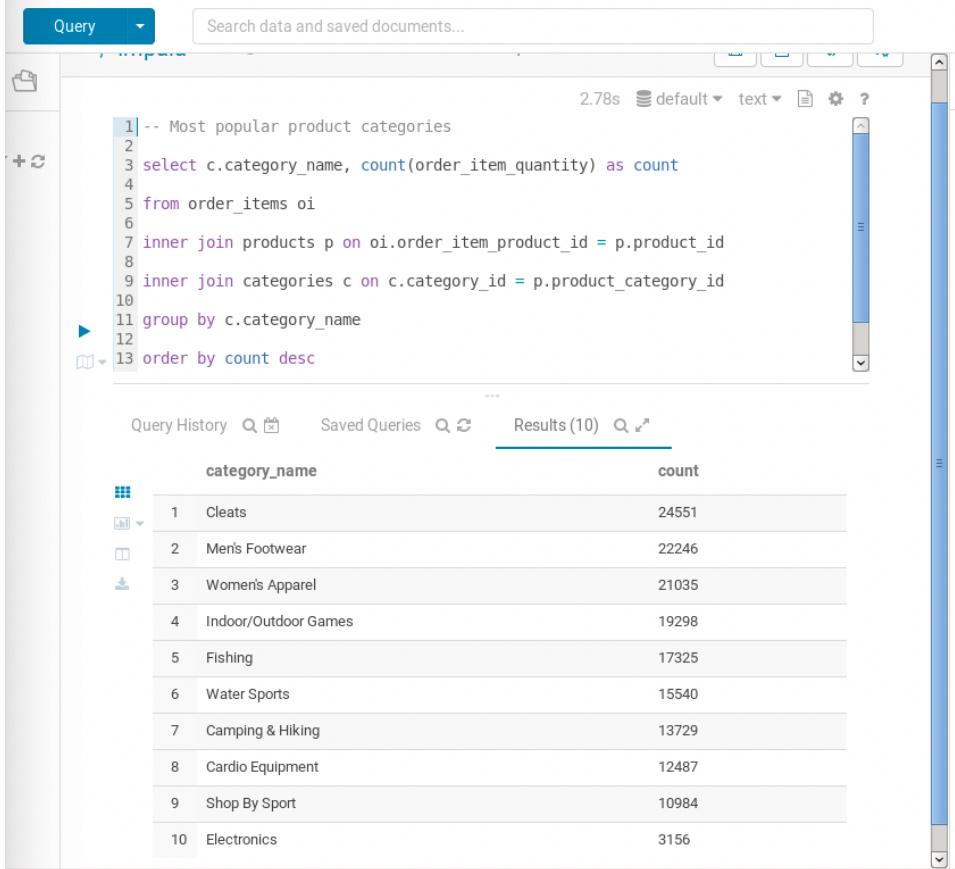
Escriba las siguientes consultas de ejemplo estándar de SQL para calcular los ingresos totales por producto y mostrar los 10 principales productos que generan ingresos:

```

-- Most popular product categories
select c.category_name, count(order_item_quantity) as count
from order_items oi
inner join products p on oi.order_item_product_id = p.product_id
inner join categories c on c.category_id = p.product_category_id
group by c.category_name
order by count desc
limit 10;

```

(1) Describa qué hace esta consulta



The screenshot shows a database query interface. The query is:

```

1 -- Most popular product categories
2
3 select c.category_name, count(order_item_quantity) as count
4
5 from order_items oi
6
7 inner join products p on oi.order_item_product_id = p.product_id
8
9 inner join categories c on c.category_id = p.product_category_id
10
11 group by c.category_name
12
13 order by count desc

```

The results are displayed in a table:

category_name	count
1 Cleats	24551
2 Men's Footwear	22246
3 Women's Apparel	21035
4 Indoor/Outdoor Games	19298
5 Fishing	17325
6 Water Sports	15540
7 Camping & Hiking	13729
8 Cardio Equipment	12487
9 Shop By Sport	10984
10 Electronics	3156

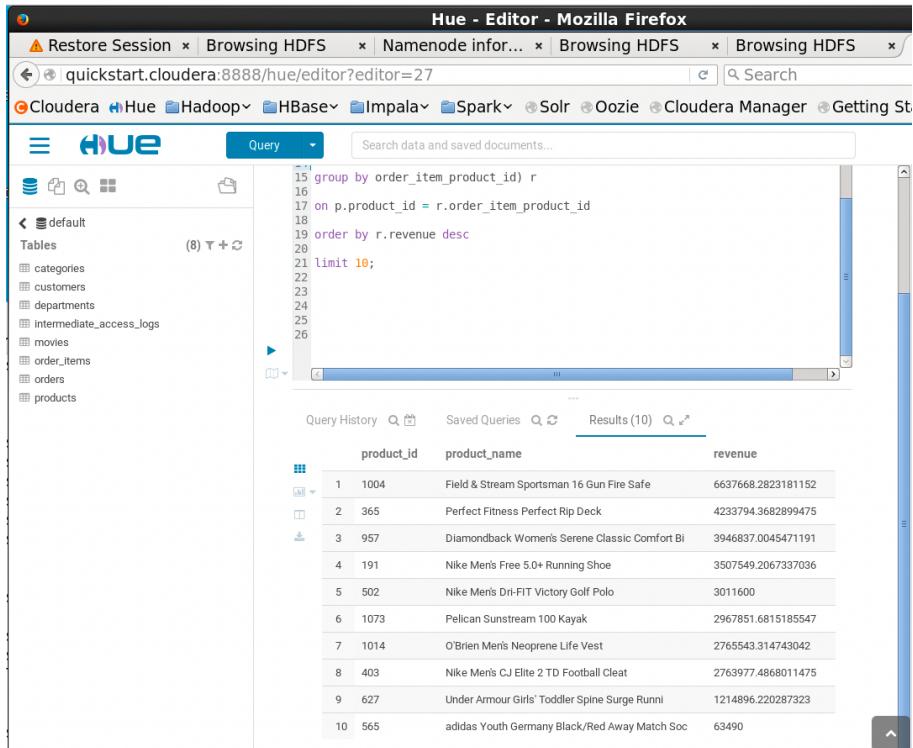
Ahora haga la siguiente consulta:

```

-- top 10 revenue generating products
select p.product_id, p.product_name, r.revenue
from products p inner join
(select oi.order_item_product_id, sum(cast(oi.order_item_subtotal as
float)) as revenue
from order_items oi inner join orders o
on oi.order_item_order_id = o.order_id
where o.order_status <> 'CANCELED'
and o.order_status <> 'SUSPECTED_FRAUD'
group by order_item_product_id) r
on p.product_id = r.order_item_product_id
order by r.revenue desc
limit 10;

```

(2) Describa qué hace esta consulta



The screenshot shows the Hue interface in Mozilla Firefox. The top navigation bar includes tabs for 'Restore Session', 'Browsing HDFS', 'Namenode infor...', 'Browsing HDFS', and 'Browsing HDFS'. Below the tabs, the URL is 'quickstart.cloudera:8888/hue/editor?editor=27'. The main area is divided into two panes. The left pane, titled 'HUE', shows a sidebar with a 'Tables' section containing 'default', 'categories', 'customers', 'departments', 'intermediate_access_logs', 'movies', 'order_items', 'orders', and 'products'. The right pane contains a 'Query' editor with the following SQL code:

```

15 group by order_item_product_id) r
16
17 on p.product_id = r.order_item_product_id
18
19 order by r.revenue desc
20
21 limit 10;
22
23
24
25
26

```

Below the query editor is a 'Results (10)' tab, which displays a table with the following data:

	product_id	product_name	revenue
1	1004	Field & Stream Sportsman 16 Gun Fire Safe	6637668.2823181152
2	365	Perfect Fitness Perfect Rip Deck	4233794.3682899475
3	957	Diamondback Women's Serene Classic Comfort Bi	3946837.0045471191
4	191	Nike Men's Free 5.0+ Running Shoe	3507549.2067337036
5	502	Nike Men's Dri-FIT Victory Golf Polo	3011600
6	1073	Pelican Sunstream 100 Kayak	2967851.6815185547
7	1014	O'Brien Men's Neoprene Life Vest	2765543.314743042
8	403	Nike Men's CJ Elite 2 TD Football Cleat	2763977.4868011475
9	627	Under Armour Girls' Toddler Spine Surge Runn	1214896.220287323
10	565	adidas Youth Germany Black/Red Away Match Soc	63490

Puede notar que le dijimos a Sqoop que importara los datos a Hive, pero usamos Impala para consultar los datos. Esto se debe a que Hive e Impala pueden compartir archivos de datos y los metadatos de la tabla. Hive funciona compilando consultas SQL en trabajos de MapReduce, lo que lo hace muy flexible, mientras que Impala ejecuta consultas por sí mismo y se construye desde cero para ser lo más rápido posible, lo que lo hace mejor para el análisis interactivo.

CONCLUSIÓN

Hemos hecho consultas a tablas con Impala y también puede usar interfaces y herramientas regulares (como SQL) dentro de un entorno Hadoop. La idea aquí es que puede hacer los mismos análisis, pero donde la arquitectura de Hadoop, frente a los sistemas tradicionales, proporciona una escala y flexibilidad mucho mayores.

PARTE 2: Big Data, mostrando valor

En este escenario usted produjo lo que siempre produce: un informe sobre datos estructurados, pero realmente no demostró ningún valor adicional.

Pero usted tiene un as bajo la manga:

Correlacionar datos estructurados con datos no estructurados

Usted se da cuenta de que otra pregunta comercial interesante sería:

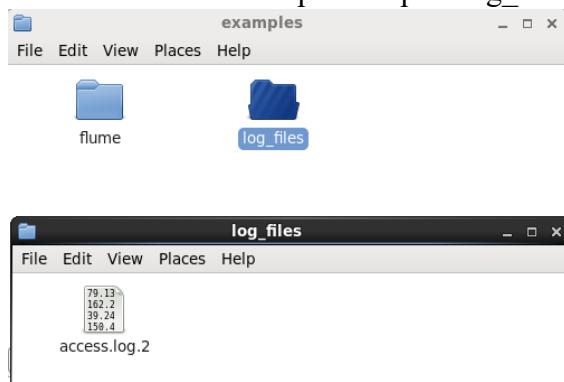
¿son los productos más vistos en el e-commerce también los más vendidos? (o para otros escenarios, los más buscados, los más conversados sobre ...).

Dado que Hadoop puede almacenar datos no estructurados y semiestructurados junto con datos estructurados sin tener que remodelar una base de datos completa, también puede ingerir, almacenar y procesar eventos de registros web. Veamos qué visitantes del sitio han visto realmente más.

Usaremos un conjunto de datos de flujo de clics web que puede cargar en bloque directamente en HDFS.

- Datos de carga masiva

La muestra (aproximadamente 180,000 líneas) de los datos de registro de acceso de un mes se encuentra en /opt/examples/log_files/access.log.2.



Movamos estos datos del sistema de archivos local Centos a HDFS.

```
$ sudo -u hdfs hadoop fs -mkdir /user/hive/warehouse/original_access_logs
$ sudo -u hdfs hadoop fs -copyFromLocal /opt/examples/log_files/access.log.2 /user/hive/warehouse/original_access_logs
```

Verifique que sus datos (logs de la web del e-commerce) estén en HDFS ejecutando el siguiente comando:

```
$ hdfs dfs -ls /user/hive/warehouse/original_access_logs
[cloudera@quickstart ~]$ hdfs dfs -ls /user/hive/warehouse/original_access_logs
Found 1 items
-rw-r--r-- 1 hdfs supergroup 39593868 2021-06-30 08:54 /user/hive/warehouse/original_access_logs/access.log.2
[cloudera@quickstart ~]$
```

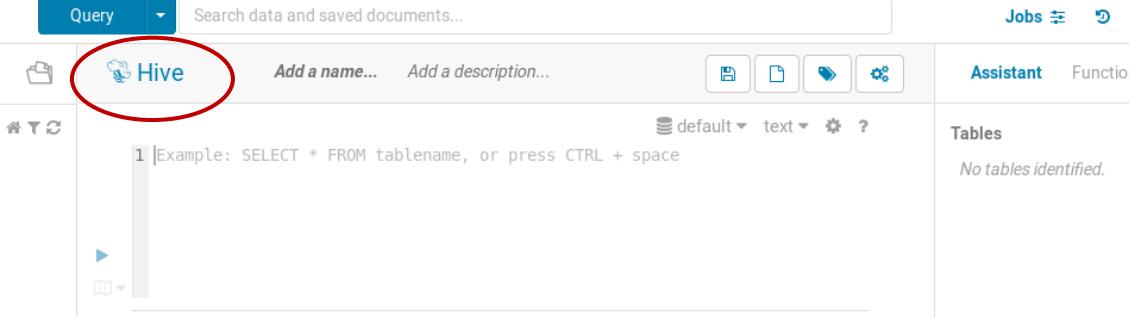
Ahora puede crear una tabla en Hive y consultar los datos a través de Impala. Construirá esta tabla en 2 pasos.

Primero, aprovechará los SerDe (serializers/deserializers) flexibles de Hive para analizar los registros en campos individuales utilizando una expresión regular.

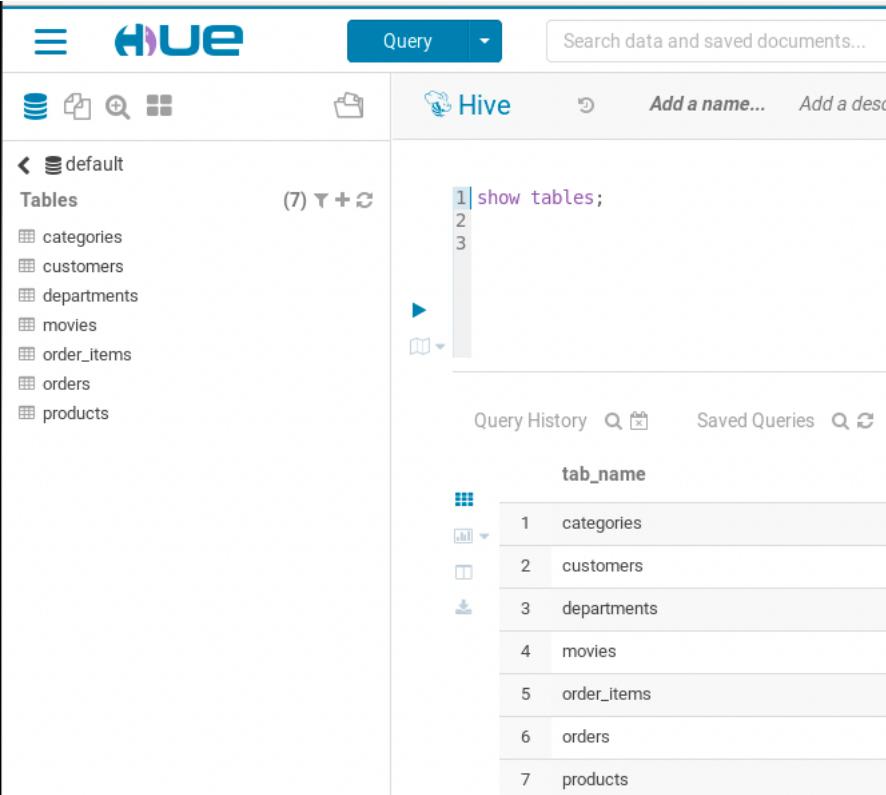
En segundo lugar, transferirá los datos de esta tabla intermedia a una que no requiera ningún SerDe especial. Una vez que los datos están en esta tabla, puede consultarlos mucho más rápido y de manera más interactiva utilizando Impala.

Utilizaremos la aplicación **Hive Query Editor** en Hue para ejecutar las siguientes consultas:

Cambie de Editor: Vaya a <Query - Editor> y seleccione Hive



The screenshot shows the Hue interface with the 'Query' tab selected. A red circle highlights the 'Hive' tab in the top navigation bar. The main query editor area contains the text '1 Example: SELECT * FROM tablename, or press CTRL + space'. The right panel shows a 'Tables' section with the message 'No tables identified.'



The screenshot shows the Hue interface with the 'Query' tab selected. The left sidebar shows a list of tables: categories, customers, departments, movies, order_items, orders, and products. The main query editor area contains the text '1 show tables;'. The right panel shows a results table with the following data:

tab_name
1 categories
2 customers
3 departments
4 movies
5 order_items
6 orders
7 products

```
CREATE EXTERNAL TABLE intermediate_access_logs (
  ip STRING,
  date STRING,
  method STRING,
  url STRING,
  http_version STRING,
  code1 STRING,
  code2 STRING,
  dash STRING,
  user_agent STRING)
ROW FORMAT SERDE 'org.apache.hadoop.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/original_access_logs';
```

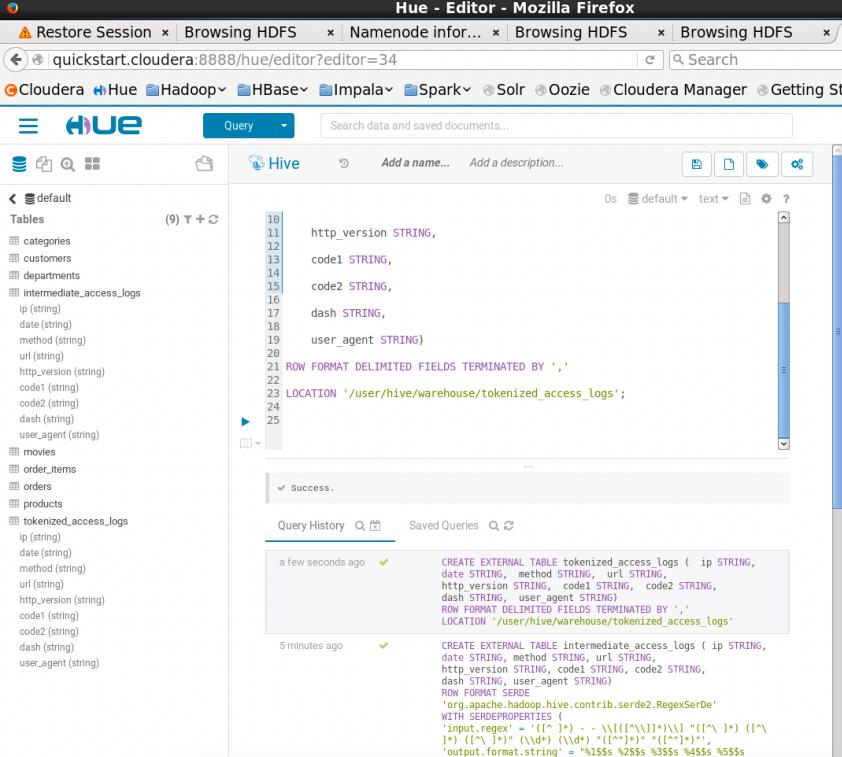
Vemos que ahora aparece la nueva tabla en el warehouse:

The screenshot shows the Hue interface for Hive. On the left, the 'Tables' section shows the 'default' database with several tables listed. In the center, a query editor window is open with the following code:

```
1 CREATE EXTERNAL TABLE intermediate_access_logs (
2   ip STRING,
3   date STRING,
4   method STRING,
5   url STRING,
6   http_version STRING,
7   code1 STRING,
8   code2 STRING,
9   dash STRING,
10  user_agent STRING)
11 ROW FORMAT SERDE 'org.apache.hadoop.hive.contrib.serde2.RegexSerDe'
12 WITH SERDEPROPERTIES (
13   'input.regex' = '([^\ ]*) -- \\[([^\ ]*)\\] "([^\ ]*) ([^\ ]*) ([^\ ]*)" (\d*) (\d*) "([^\ ]*)" "([^\ ]*)"',
14   'output.format.string' = "%1$$s %2$$s %3$$s %4$$s %5$$s %6$$s %7$$s %8$$s %9$$s",
15   LOCATION '/user/hive/warehouse/original_access_logs';
```

Below the code, a message says 'Success.' and shows the query history with the same query listed as 'a minute ago' and '6 minutes ago'.

```
CREATE EXTERNAL TABLE tokenized_access_logs (
  ip STRING,
  date STRING,
  method STRING,
  url STRING,
  http_version STRING,
  code1 STRING,
  code2 STRING,
  dash STRING,
  user_agent STRING)
ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
LOCATION '/user/hive/warehouse/tokenized_access_logs';
```



The screenshot shows the Hue interface in Mozilla Firefox. The left sidebar shows a list of tables in the 'default' database, including 'categories', 'customers', 'departments', 'intermediate_access_logs', 'movies', 'order_items', 'orders', 'products', and 'tokenized_access_logs'. The 'intermediate_access_logs' table is selected. The main pane shows the following Hive SQL code:

```

10 http_version STRING,
11
12 code1 STRING,
13
14 code2 STRING,
15
16 dash STRING,
17
18 user_agent STRING)
19
20
21 ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
22
23 LOCATION '/user/hive/warehouse/tokenized_access_logs';
24
25

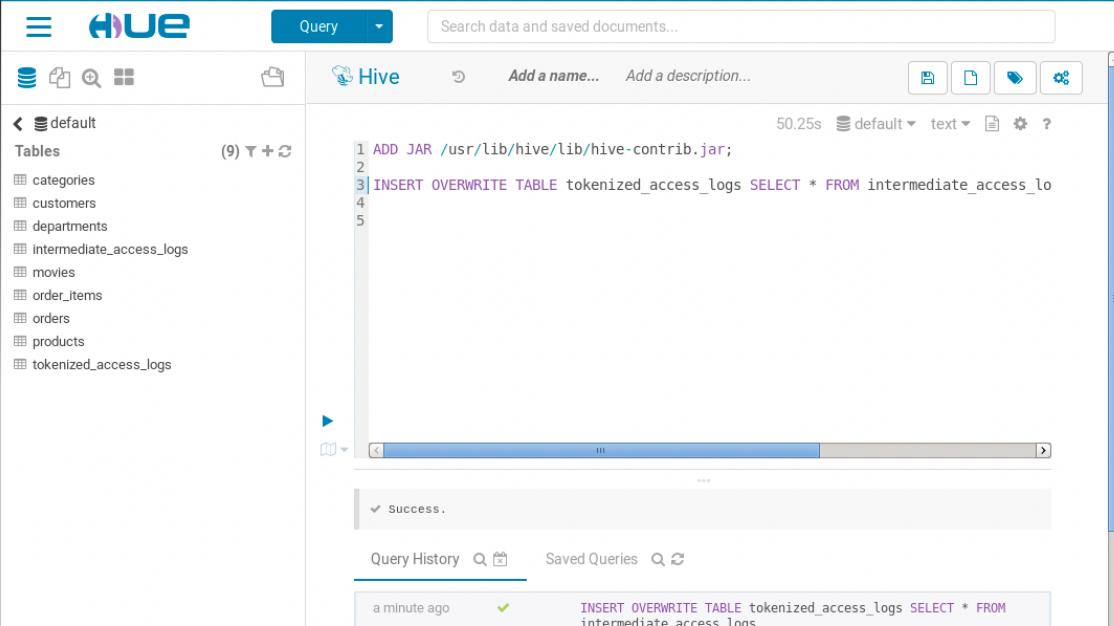
```

Below the code, a success message is displayed: 'Success.'

Ahora ejecutamos:

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

```
INSERT OVERWRITE TABLE tokenized_access_logs SELECT * FROM
intermediate_access_logs;
```



The screenshot shows the Hue interface in Mozilla Firefox. The left sidebar shows a list of tables in the 'default' database, including 'categories', 'customers', 'departments', 'intermediate_access_logs', 'movies', 'order_items', 'orders', 'products', and 'tokenized_access_logs'. The 'intermediate_access_logs' table is selected. The main pane shows the following Hive SQL code:

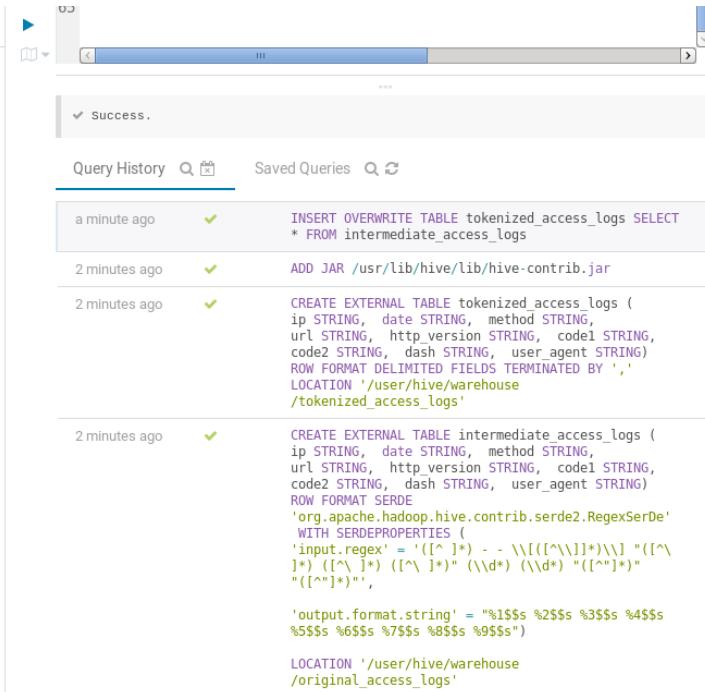
```

1 ADD JAR /usr/lib/hive/lib/hive-contrib.jar;
2
3 INSERT OVERWRITE TABLE tokenized_access_logs SELECT * FROM intermediate_access_lo
4
5

```

Below the code, a success message is displayed: 'Success.'

La consulta final tardará algo de tiempo en ejecutarse. Está utilizando un trabajo MapReduce, al igual que lo hizo la importación Sqoop, para transferir los datos de una tabla a otra en paralelo.



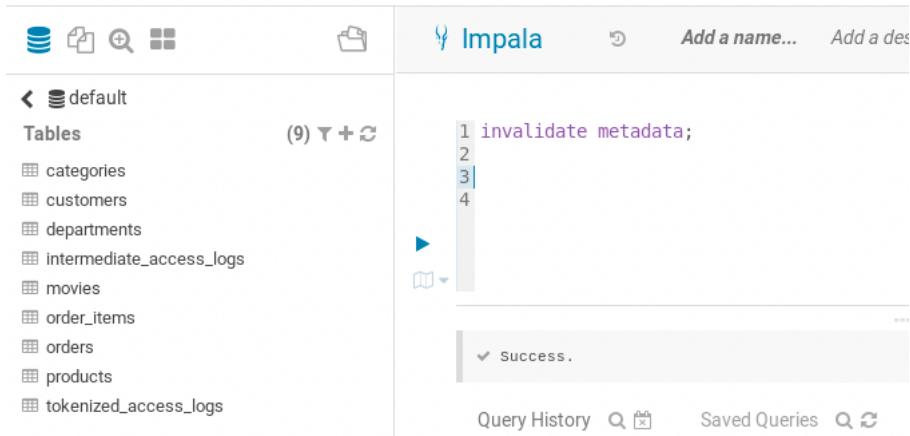
```

a minute ago ✓ INSERT OVERWRITE TABLE tokenized_access_logs SELECT
* FROM intermediate_access_logs
2 minutes ago ✓ ADD JAR /usr/lib/hive/lib/hive-contrib.jar
2 minutes ago ✓ CREATE EXTERNAL TABLE tokenized_access_logs (
ip STRING, date STRING, method STRING,
url STRING, http_version STRING, code1 STRING,
code2 STRING, dash STRING, user_agent STRING)
ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
LOCATION '/user/hive/warehouse
/tokenized_access_logs'
2 minutes ago ✓ CREATE EXTERNAL TABLE intermediate_access_logs (
ip STRING, date STRING, method STRING,
url STRING, http_version STRING, code1 STRING,
code2 STRING, dash STRING, user_agent STRING)
ROW FORMAT SERDE
'org.apache.hadoop.hive.contrib.serde2.RegexSerDe'
WITH SERDEPROPERTIES (
'input.regex' = '([^\s]+) - - \[\[([^\]]*)\]\] "([^\n]+) ([^\n]+) ([^\n]+)" (\d*) (\d*) "([^\n]+)" (\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
/original_access_logs'

```

Nuevamente, debemos decirle a Impala que algunas tablas se han creado a través de una herramienta diferente. Vuelva a la aplicación **Impala Query Editor** e ingrese el siguiente comando:

```
invalidate metadata;
```



```

1 invalidate metadata;
2
3
4

```

Success.

Ahora, si ingresa a 'show tables', consulta o actualiza la lista de tablas en la columna de la izquierda, debería ver las dos nuevas tablas externas en la base de datos predeterminada.

The screenshot shows the Hue interface. On the left, the 'Tables' section lists tables: categories, customers, departments, intermediate_access_logs, movies, order_items, orders, products, and tokenized_access_logs. The main area shows the results of the query 'SHOW TABLES;'. The results table has a header 'name' and 9 rows, each containing a number (1-9) and a table name. The 'Results' tab is selected.

	name
1	categories
2	customers
3	departments
4	intermediate_access_logs
5	movies
6	order_items
7	orders
8	products
9	tokenized_access_logs

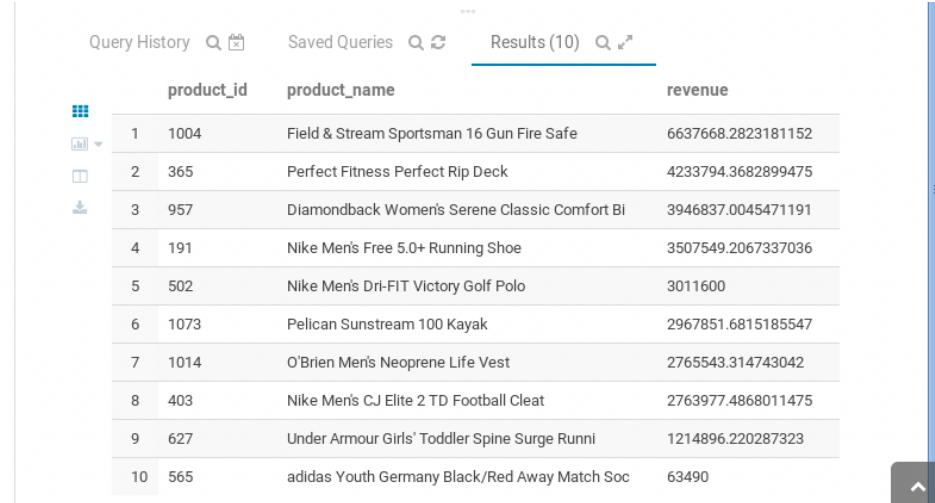
Ejecute la siguiente consulta en el Editor de consultas:

```
select count(*), url from tokenized_access_logs
where url like '%\product\%'
group by url order by count(*) desc;
```

The screenshot shows the Impala query editor. The query 'select count(*), url from tokenized_access_logs where url like '%\product\%' group by url order by count(*) desc;' is executed. The results table has headers 'count(*)' and 'url', with 13 rows of data. The 'Results' tab is selected.

count(*)	url
1926	/department/apparel/category/cleats/product/Perfect%20Fitness%20Perfect%20Rip%20Deck
1793	/department/apparel/category/featured%20shops/product/adidas%20Kids%20RG%20III%20Mid%20Football%20Cleat
1780	/department/golf/category/women's%20apparel/product/Nike%20Men's%20Dri-FIT%20Victory%20Golf%20Polo
1757	/department/apparel/category/men's%20footwear/product/Nike%20Men's%20CJ%20Elite%20%20TD%20Football%20C
1104	/department/fan%20shop/category/water%20sports/product/Pelican%20Sunstream%20100%20Kayak
1084	/department/fan%20shop/category/indoor%20outdoor%20games/product/O'Brien%20Men's%20Neoprene%20Life%20Vest
1059	/department/fan%20shop/category/camping%20&%20hiking/product/Diamondback%20Women's%20Serene%20Classic
1028	/department/fan%20shop/category/fishing/product/Field%20&%20stream%20Sportsman%2016%20Gun%20Fire%20Saf
1004	/department/footwear/category/cardio%20equipment/product/Nike%20Men's%20Free%205.0%20Running%20Shoe
939	/department/footwear/category/fitness%20accessories/product/Under%20Armour%20Hustle%20Storm%20Medium%20
930	/department/golf/category/shop%20by%20sport/product/Columbia%20Men's%20PFG%20Anchor%20Tough%20T-Shirt
896	/department/fitness/category/tennis%20&%20racquet/product/Nike%20Men's%20Comfort%20%20Slide
892	/department/footwear/category/as%20seen%20on%20%20tv!/product/Nike%20Men's%20Free%20TR%205.0%20TB%20

Note el resultado final de la *Parte 1* de la Práctica: productos que generan más ganancia:



	product_id	product_name	revenue
1	1004	Field & Stream Sportsman 16 Gun Fire Safe	6637668.2823181152
2	365	Perfect Fitness Perfect Rip Deck	4233794.3682899475
3	957	Diamondback Women's Serene Classic Comfort Bi	3946837.0045471191
4	191	Nike Men's Free 5.0+ Running Shoe	3507549.2067337036
5	502	Nike Men's Dri-FIT Victory Golf Polo	3011600
6	1073	Pelican Sunstream 100 Kayak	2967851.6815185547
7	1014	O'Brien Men's Neoprene Life Vest	2765543.314743042
8	403	Nike Men's CJ Elite 2 TD Football Cleat	2763977.4868011475
9	627	Under Armour Girls' Toddler Spine Surge Runni	1214896.220287323
10	565	adidas Youth Germany Black/Red Away Match Soc	63490

Al revisar los resultados de esta lista observe si contiene muchos de los productos en la lista más vendida de los pasos que hizo en la Parte 1, **responda**, ¿hay algún producto que no apareció en el resultado anterior (visitas web)? En otras palabras, **¿hay un producto que parece ser visto mucho, pero nunca comprado?** Si en la etapa de análisis usted encuentra estos hallazgos extraños se debe prestar mucha atención.

Suponga que descubre que en esa página de vista del producto, con gran mayoría de visitantes, la ruta de ventas del producto tenía un error tipográfico en el precio del artículo. Una vez que se corrija el error tipográfico y muestre un precio correcto, las ventas de ese producto comenzarán a aumentar rápidamente.

(3) Discuta en su grupo las actividades hechas en esta práctica (por ejemplo, cómo fue usado el archivo de logs de internet y colocado en el warehouse en la tabla tokenized_access_logs) y desde el punto de vista de business analytics deduzca la utilidad de los hallazgos a través de las consultas hechas.

CONCLUSIÓN

Existe el riesgo de pérdida si una organización busca respuestas dentro de datos parciales. La **correlación de dos conjuntos de datos** para la misma pregunta comercial mostró valor, y poder hacerlo dentro de la misma plataforma hizo la vida más fácil para usted y para la organización.

PARTE FINAL:

(4) NOTA: Tienes tu tabla *movies* en Hive? **Usa Impala para generar 3 consultas:**

The screenshot shows the Hue interface. On the left, a sidebar lists tables: default, categories, customers, departments, intermediate_access_logs, movies, order_items, orders, products, and tokenized_access_logs. The 'movies' table is selected, showing its schema: id (int), nombre (string), anio (int), rating (float), and numratings (int). The main area displays a query in the Impala editor:

```

2 select count(*), anio from movies
3
4 where ANIO < 2012
5
6 group by ANIO order by count(*) desc;
7
8
9
10

```

The results table shows the count of movies for each year from 2011 down to 2009:

	count(*)	anio
1	5511	2011
2	5107	2010
3	4451	2009

Explora la herramienta que te permite generar gráficas o visualizaciones de los resultados de las queries ejecutadas, ¿te permite esto hacer análisis exploratorio de tus datos?

The screenshot shows the Hue interface with the same query and results as the previous screenshot. The results are now displayed as a bar chart. The x-axis is labeled 'anio' and the y-axis represents the count of movies. The bars are grouped by year. A red circle highlights the 'X-AXIS' dropdown menu, which is set to 'anio'. The chart has a legend indicating the series is 'count(*)'.

anio	count(*)
2011	5511
2010	5107
2009	4451