Introduction to Impala and Hive

Prof. Hyuk-Yoon Kwon

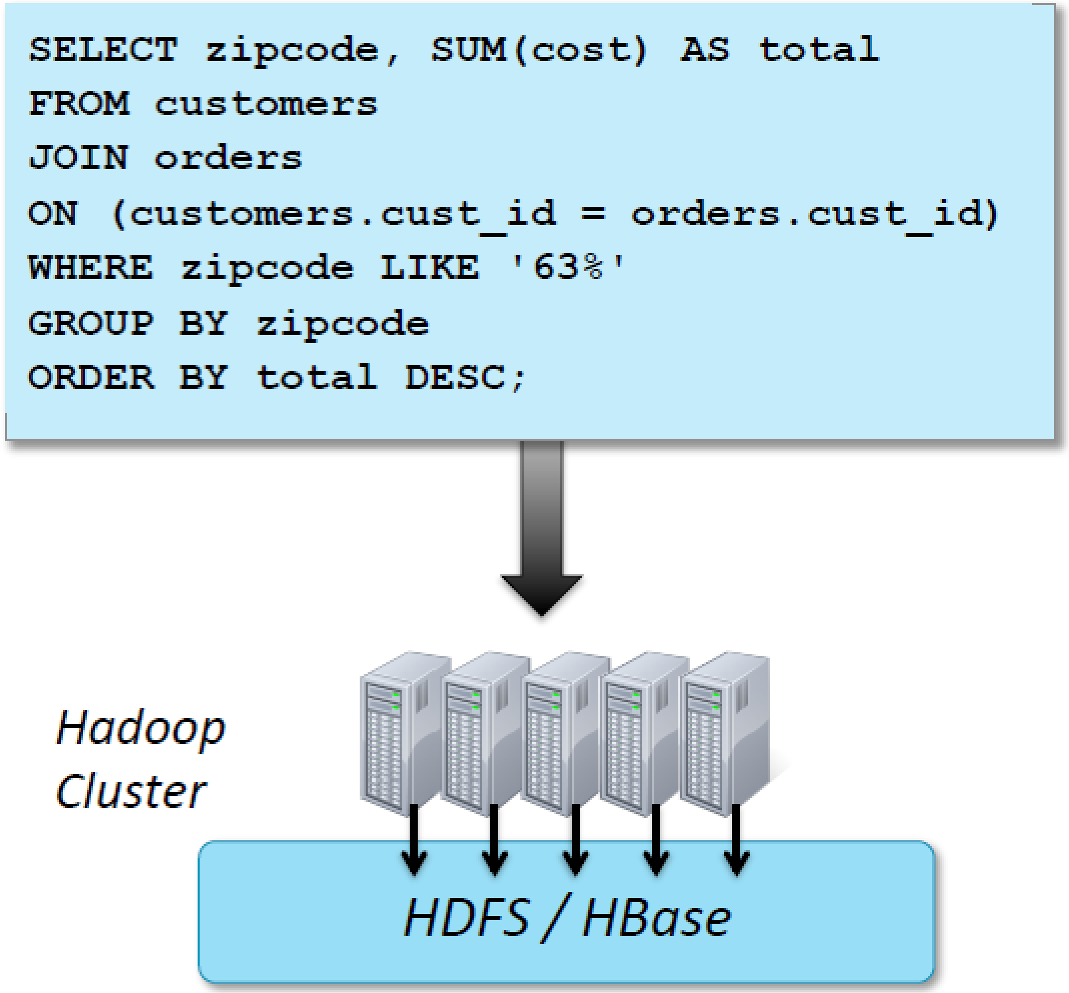
Introduction to Impala and Hive

In this chapter you will learn

- What Hive is
- What Impala is
- How Impala and Hive Compare
- How to query data using Impala and Hive
- How Hive and Impala differ from a relational database
- Ways in which organizations use Hive and Impala

Introduction to Impala and Hive (1)

- Impala and Hive are both tools that provide SQL querying of data stored in HDFS / HBase



```
SELECT zipcode, SUM(cost) AS total
FROM customers
JOIN orders
ON (customers.cust_id = orders.cust_id)
WHERE zipcode LIKE '63%'
GROUP BY zipcode
ORDER BY total DESC;
```

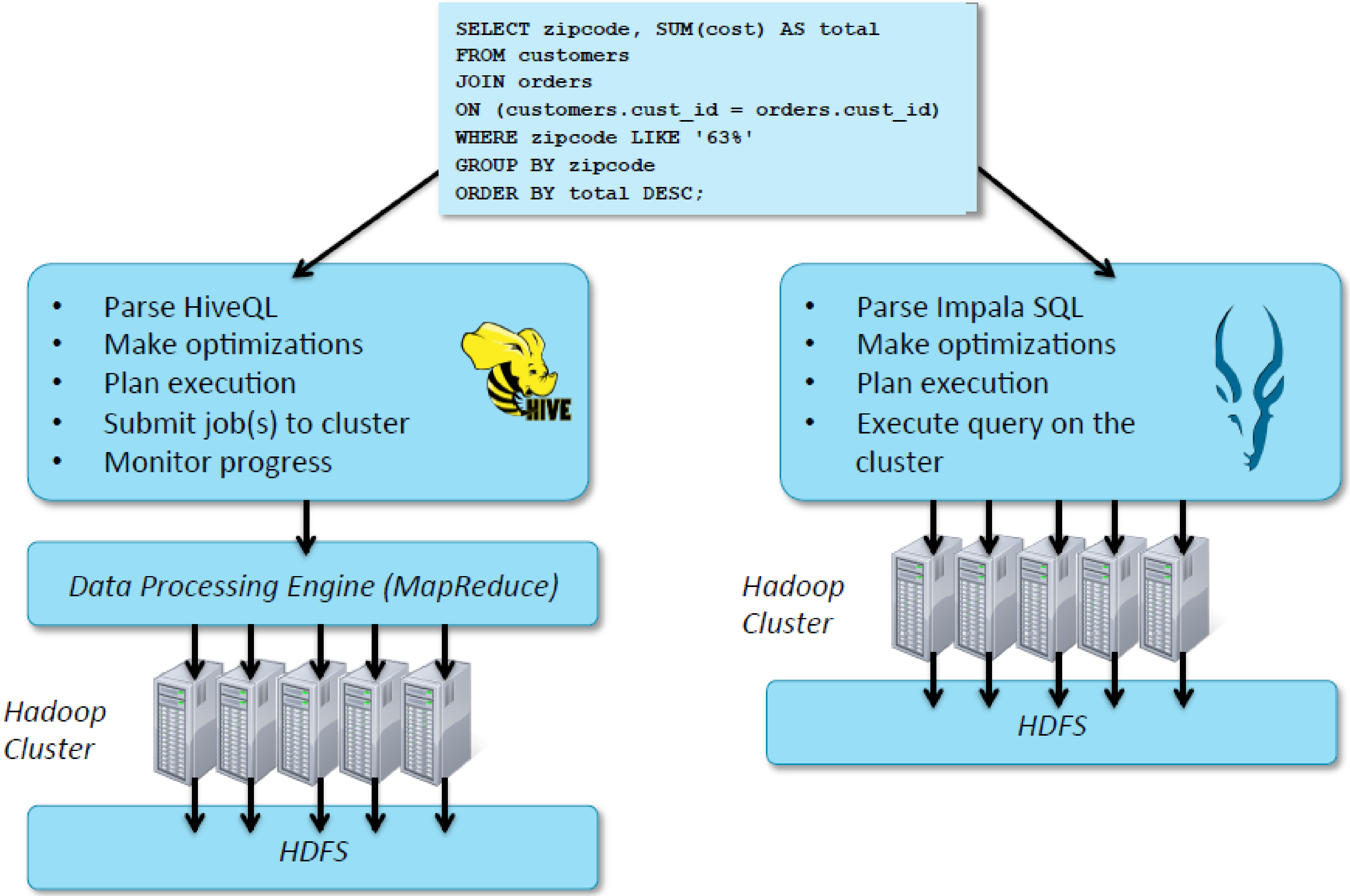
Introduction to Impala and Hive (2)

- **Apache Hive is a high-level abstraction on top of MapReduce**
  - Uses HiveQL
  - Generates MapReduce or Spark* jobs that run on the Hadoop cluster
  - Originally developed at Facebook around 2007
    - Now an open-source Apache project
- **Cloudera Impala is a high-performance dedicated SQL engine**
  - Uses Impala SQL
  - Inspired by Google’s Dremel project
  - Query latency measured in milliseconds
  - Developed at Cloudera in 2012
    - Open-source with an Apache license

What’s the Difference?

- **Hive has more features**
  - E.g. Complex data types (arrays, maps) and full support for windowing analytics
  - Highly extensible
  - Commonly used for batch processing
- **Impala is much faster**
  - Specialized SQL engine offers 5x to 50x better performance
  - Ideal for interactive queries and data analysis
  - More features being added over time

High-Level Overview



```
SELECT zipcode, SUM(cost) AS total
FROM customers
JOIN orders
ON (customers.cust_id = orders.cust_id)
WHERE zipcode LIKE '63%'
GROUP BY zipcode
ORDER BY total DESC;
```

- Parse HiveQL
- Make optimizations
- Plan execution
- Submit job(s) to cluster
- Monitor progress

Data Processing Engine (MapReduce)

Hadoop Cluster

HDFS

- Parse Impala SQL
- Make optimizations
- Plan execution
- Execute query on the cluster

Hadoop Cluster

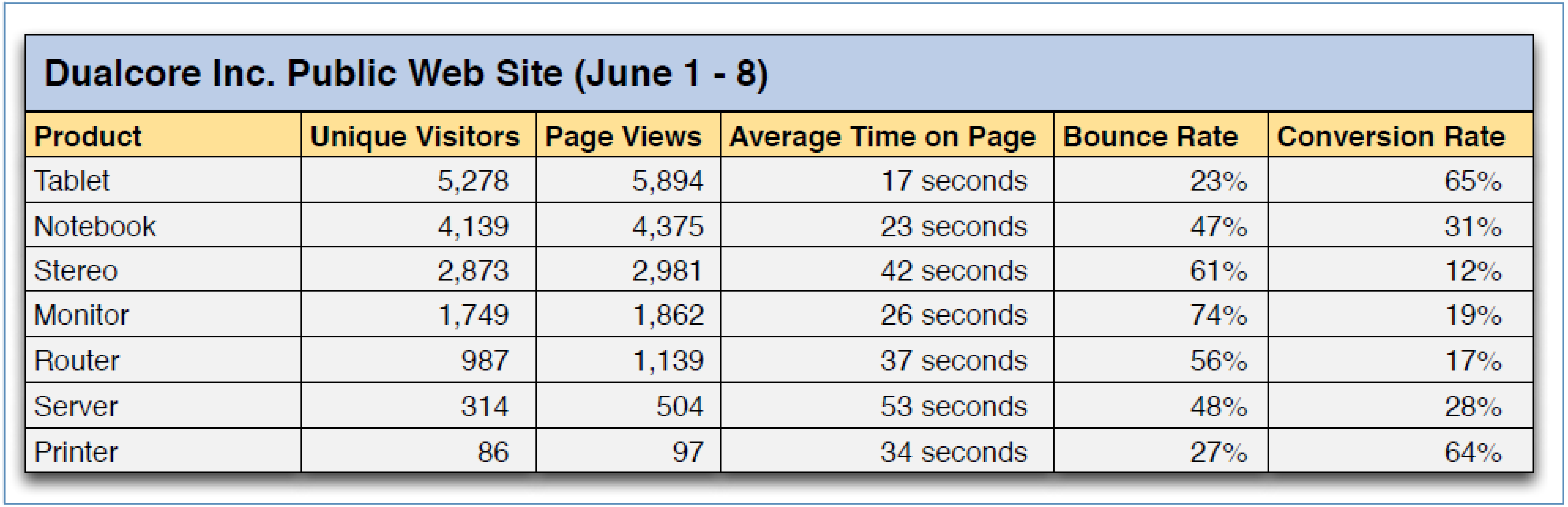
HDFS

Why Use Hive and Impala?

- **Brings large-scale data analysis to a broader audience**
  - No software development experience required
  - Leverage existing knowledge of SQL
- **More productive than writing MapReduce or Spark directly**
  - Five lines of HiveQL/Impala SQL might be equivalent to 200 lines or more of Java
- **Offers interoperability with other systems**
  - Extensible through Java and external scripts
  - Many business intelligence (BI) tools support Hive and/or Impala

Use Case: Log File Analytics

- Server log files are an important source of data
- Hive and Impala allow you to treat a directory of log files like a table
  - Allows SQL-like queries against raw data

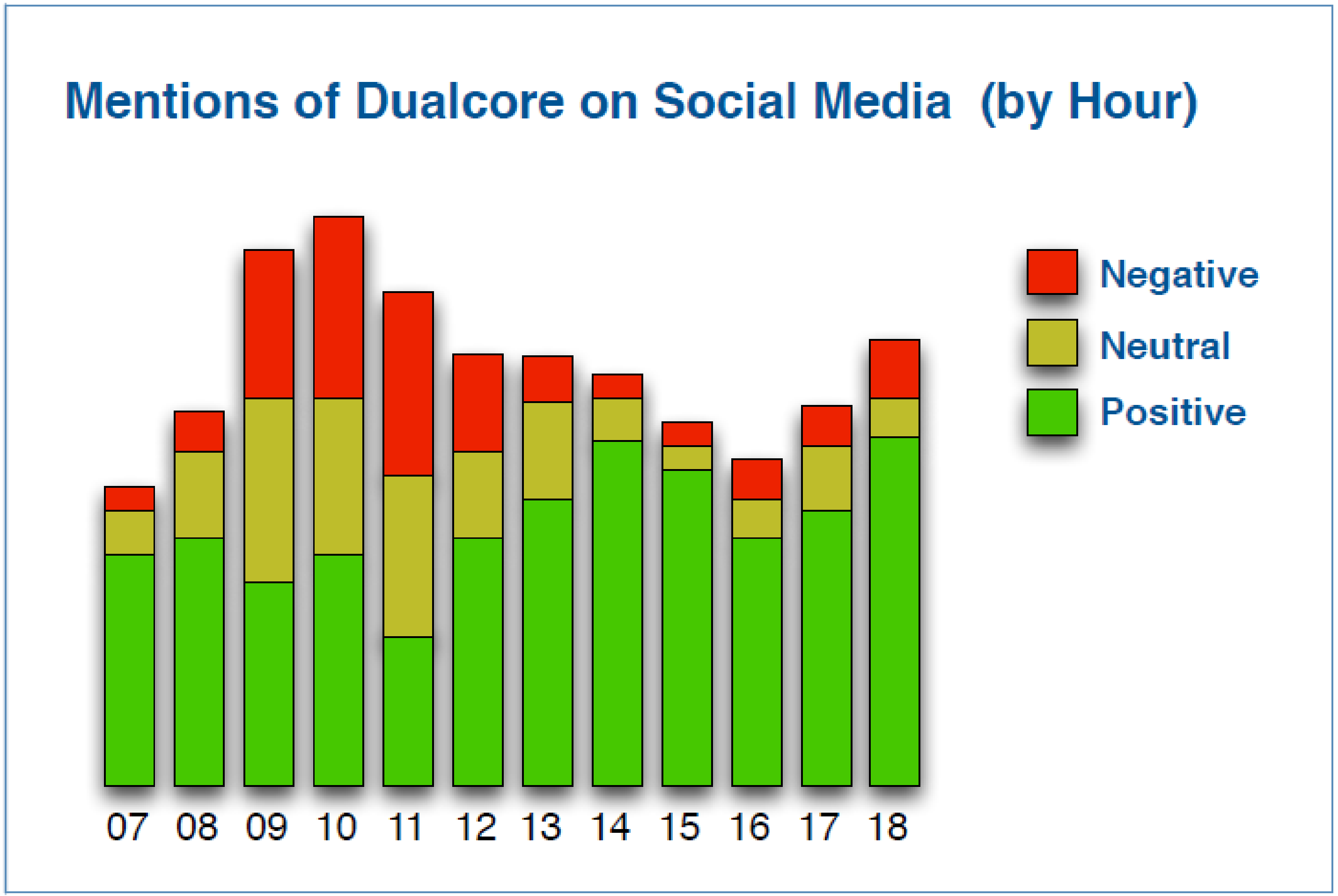


Dualcore Inc. Public Web Site (June 1 - 8)

Product	Unique Visitors	Page Views	Average Time on Page	Bounce Rate	Conversion Rate
Tablet	5,278	5,894	17 seconds	23%	65%
Notebook	4,139	4,375	23 seconds	47%	31%
Stereo	2,873	2,981	42 seconds	61%	12%
Monitor	1,749	1,862	26 seconds	74%	19%
Router	987	1,139	37 seconds	56%	17%
Server	314	504	53 seconds	48%	28%
Printer	86	97	34 seconds	27%	64%

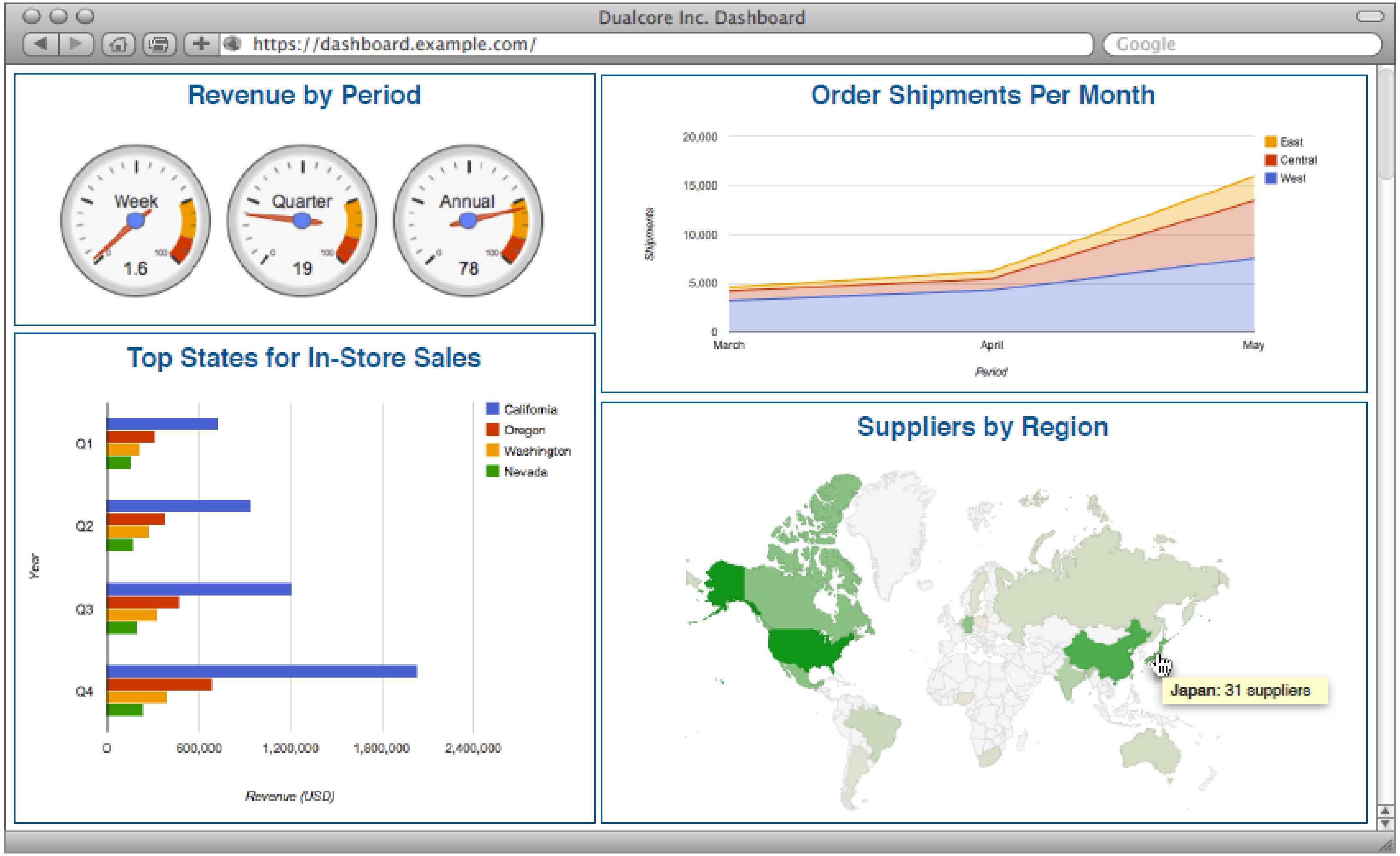
Use Case: Sentiment Analytics

- Many organizations use Hive or Impala to analyze social media coverage



Use Case: Business Intelligence

- Many leading business intelligence tools support Hive and Impala



Interacting with Hive and Impala

- **Hive and Impala offer many interfaces for running queries**
  - Command-line shell
    - Impala: Impala shell
    - Hive: Beeline
  - Hue Web UI
    - Hive Query Editor
    - Impala Query Editor
    - Metastore Manager
  - ODBC / JDBC

Starting the Impala Shell

- **You can execute statements in the Impala shell**
  - This interactive tool is similar to the shell in MySQL
- **Execute the `impala-shell` command to start the shell**
  - Some log messages truncated to better fit the slide

```
$ impala-shell
Connected to localhost.localdomain:21000
Server version: impalad version 2.1.0-cdh5 (…)
Welcome to the Impala shell.
[localhost.localdomain:21000] >
```

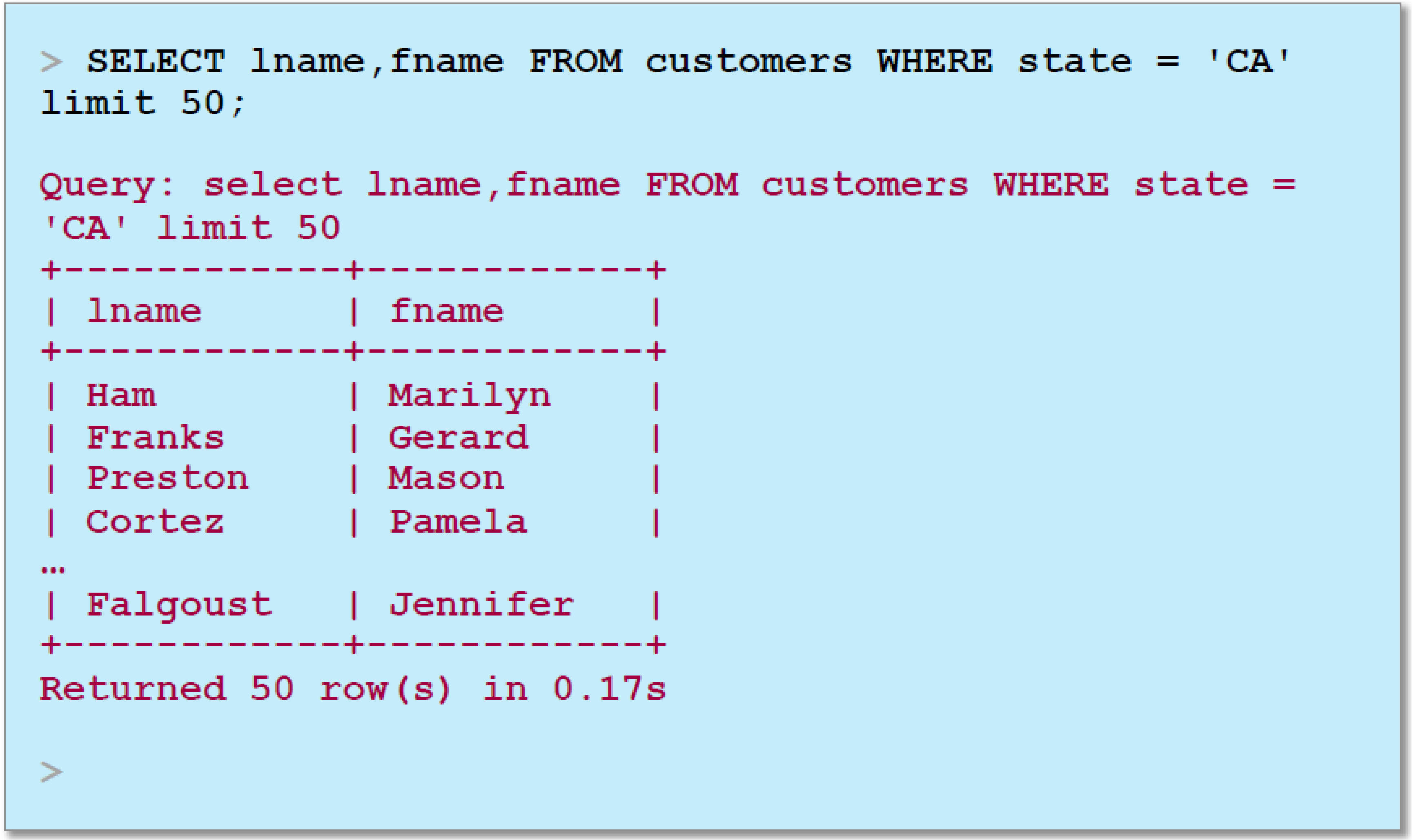
- **Use `-i  hostname:port` option to connect to a different server**

```
$ impala-shell -i myserver.example.com:21000
[myserver.example.com:21000] >
```

Using the Impala Shell

- **Enter semicolon-terminated statements at the prompt**
  - Hit [Enter] to execute a query or command
  - Use the `quit` command to exit the shell
- **Use `impala-shell --help` for a full list of options**

Executing Queries in the Impala Shell



```
> SELECT lname,fname FROM customers WHERE state = 'CA'
limit 50;

Query: select lname,fname FROM customers WHERE state =
'CA' limit 50
+------------+------------+
| lname      | fname      |
+------------+------------+
| Ham        | Marilyn    |
| Franks     | Gerard     |
| Preston    | Mason      |
| Cortez     | Pamela     |
…
| Falgoust   | Jennifer   |
+------------+------------+
Returned 50 row(s) in 0.17s

>
```

Note: shell prompt abbreviated as **>**

Interacting with the Operating System

- Use `shell` to execute system commands from within Impala shell

```
> shell date;
Mon May 20 16:44:35 PDT 2013
```

- No direct support for HDFS commands
  - But could run `hdfs dfs` using `shell`

```
> shell hdfs dfs -mkdir /reports/sales/2013;
```

Running Impala Queries from the Command Line

- You can execute a file containing queries using the -f option

```
$ impala-shell -f myquery.sql
```

- Run queries directly from the command line with the -q option

```
$ impala-shell -q 'SELECT * FROM users'
```

- Use -o to capture output to file
  - Optionally specify delimiter

```
$ impala-shell -f myquery.sql \
    -o results.txt \
    --delimited \
    --output_delimiter=','
```

Practice – Impala Shell

- Setup environments (remember these commands when the server is not working)
  - $DEV1/scripts/training_setup_dev1.sh
  - sudo service zookeeper-server start
  - sudo service hive-server2 start

1. In a terminal window, import the **webpage** table from MySQL directly into the Hive Metastore
   - Hint1: use Sqoop command
   - Hint2: use the option “--hive-import” to check the copied results from Impala-Shell
   - Hint3: use --warehouse-dir= /user/hive/warehouse to specify the stored location

2. Using Hue or the HDFS command line, review the imported data files. The Hive import copies the data to the Hive warehouse location

3. In Impala Shell, execute a SQL that finds the name from the **webpage** table whose name includes starting with “ifruit”
   - Caution: execute “invalidate metadata;” before executing SQL

4. In Impala Shell, modify the previous SQL to find only 5 results

Starting Beeline (Hive's Shell)

- **You can execute HiveQL statements in the Beeline shell**
  - Interactive shell based on the SQLLine utility
  - Similar to the Impala shell
- **Start Beeline by specifying the URL for a Hive2 server**
  - Plus username and password if required

```
$ beeline -u jdbc:hive2://host:10000 \
-n username -p password

0: jdbc:hive2://localhost:10000>
```

Executing Queries in Beeline

- **SQL commands are terminated with semi-colon (;)**
- **Similar to Impala shell**
  - Results formatting is slightly different

```
1: url> SELECT lname,fname FROM customers
. . . > WHERE state = 'CA' LIMIT 50;

+-------------+-------------+
|    lname    |    fname    |
+-------------+-------------+
| Ham         | Marilyn     |
| Franks      | Gerard      |
| Preston     | Mason       |
...
| Falgoust    | Jennifer    |
+-------------+-------------+
50 rows selected (15.829 seconds)

1: url>
```

Using Beeline

- **Execute Beeline commands with ‘!’**
  - No terminator character
- **Some commands**
  - `!connect` *`url`* – connect to a different Hive2 server
  - `!exit` – exit the shell
  - `!help` – show the full list of commands
  - `!verbose` – show added details of queries

```
0: jdbc:hive2://localhost:10000> !exit
```

Executing Hive Queries from the Command Line

- You can also execute a file containing HiveQL code using the `-f` option

```
$ beeline -u ... -f myquery.hql
```

- Or use HiveQL directly from the command line using the `-e` option

```
$ beeline -u ... -e 'SELECT * FROM users'
```

- Use the `--silent` option to suppress informational messages
  - Can also be used with `-e` or `-f` options

```
$ beeline -u ... --silent
```

Practice - setup

If you plan to use Hive rather than Impala for this or subsequent exercises, start the Hive server, which is not started by default, by entering the following two commands in a terminal window:

```
$ sudo service zookeeper-server start
$ sudo service hive-server2 start
```

Practice – Hive's Shell

1. In a terminal window, import two tables **device** and **accountdevice** tables from MySQL directly into the Hive Metastore

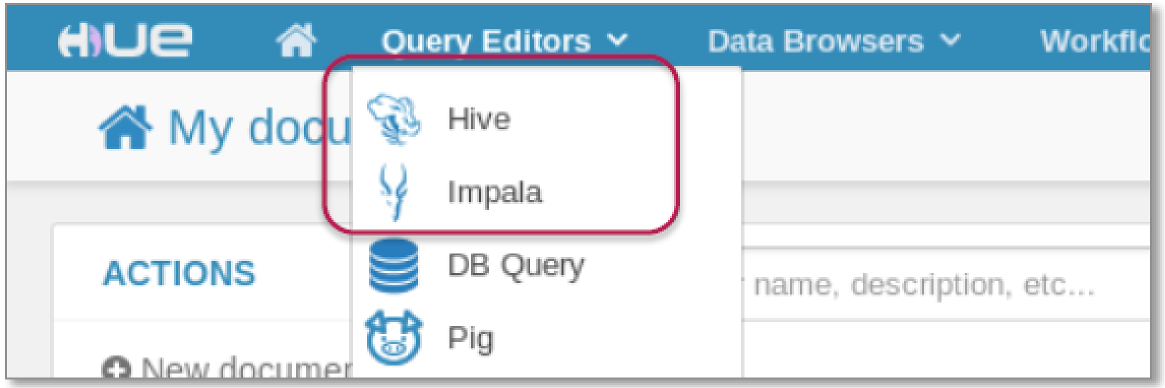
2. In Beeline, execute a SQL that joins two tables device and accountdevice where device_id = 5
   - Hint: Join operations look like this - FROM CUSTOMERS c JOIN ORDERS o ON (c.ID = o.CUSTOMER_ID);

3. Make the SQL with a sql file and execute it from the command line

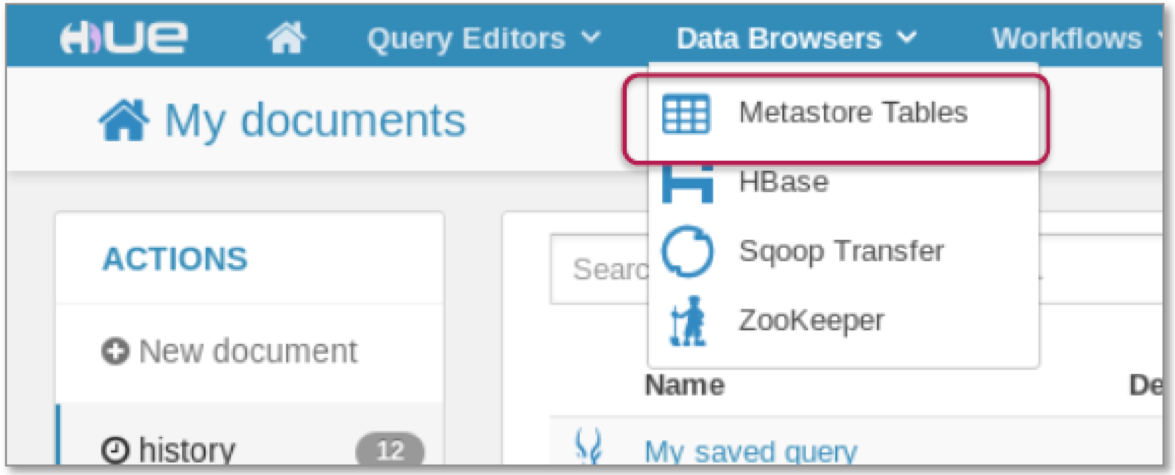
Using Hue with Hive and Impala

You can use Hue to…

Query data with Hive or Impala

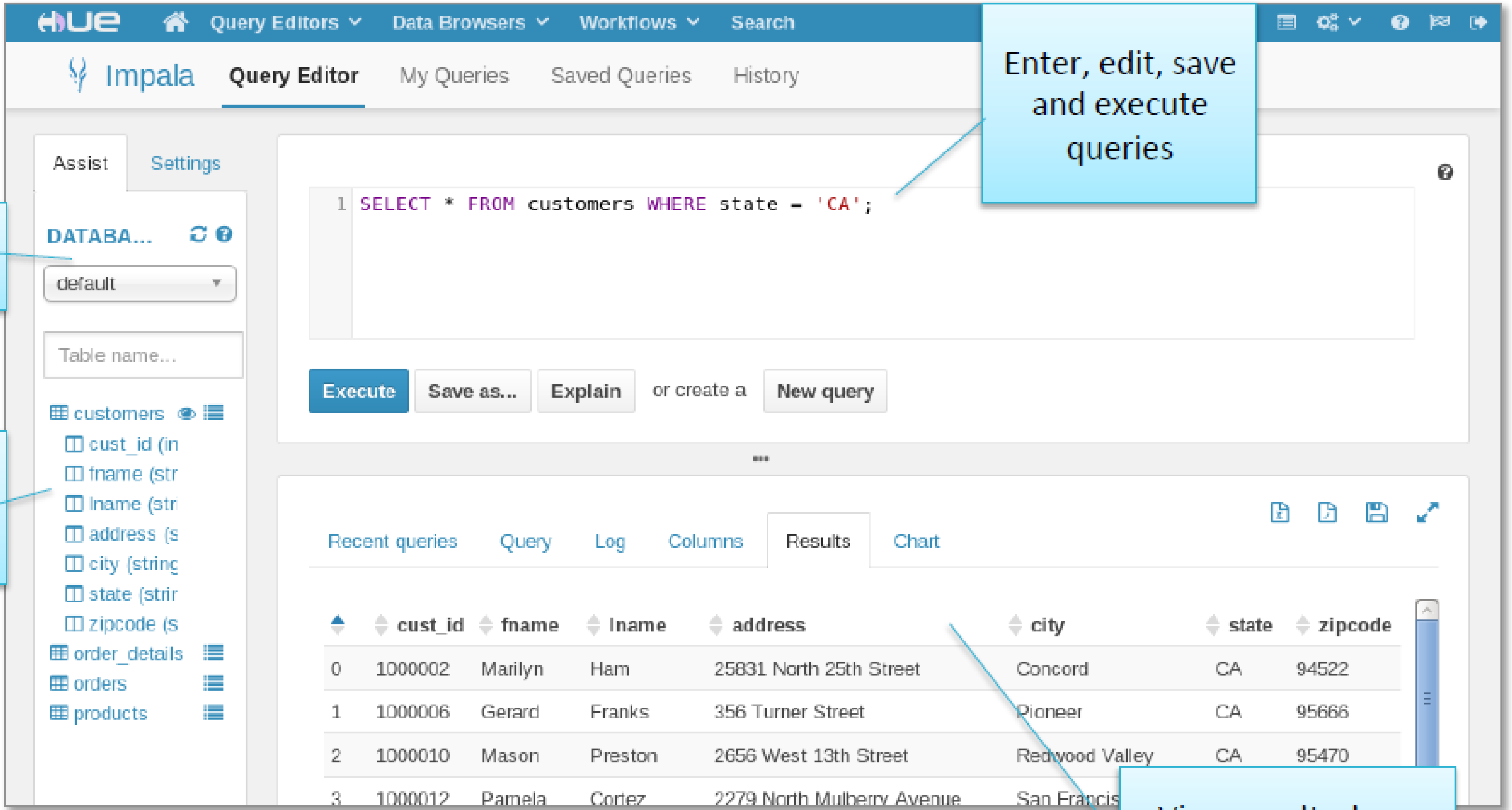


View and manage the Metastore



The Hue Query Editor

- **The Impala and Hive Query editors are nearly identical**



```
SELECT * FROM customers WHERE state = 'CA';
```

	cust_id	fname	lname	address	city	state	zipcode
0	1000002	Marilyn	Ham	25831 North 25th Street	Concord	CA	94522
1	1000006	Gerard	Franks	356 Turner Street	Pioneer	CA	95666
2	1000010	Mason	Preston	2656 West 13th Street	Redwood Valley	CA	95470
3	1000012	Pamela	Cortez	2279 North Mulberry Avenue	San Francis		

Choose a database

Explore schema and sample data

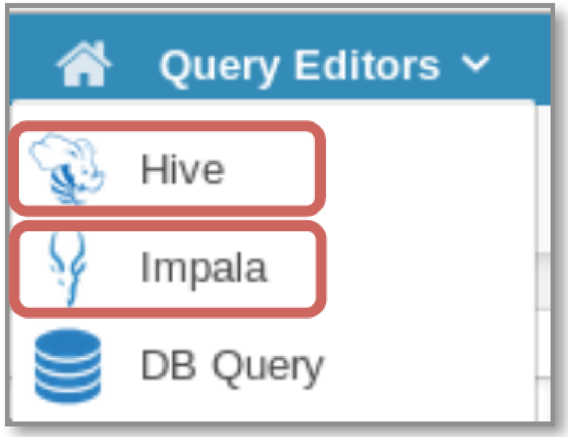
Enter, edit, save and execute queries

View results, logs, reports, etc.

Practice – Hue Query Editors

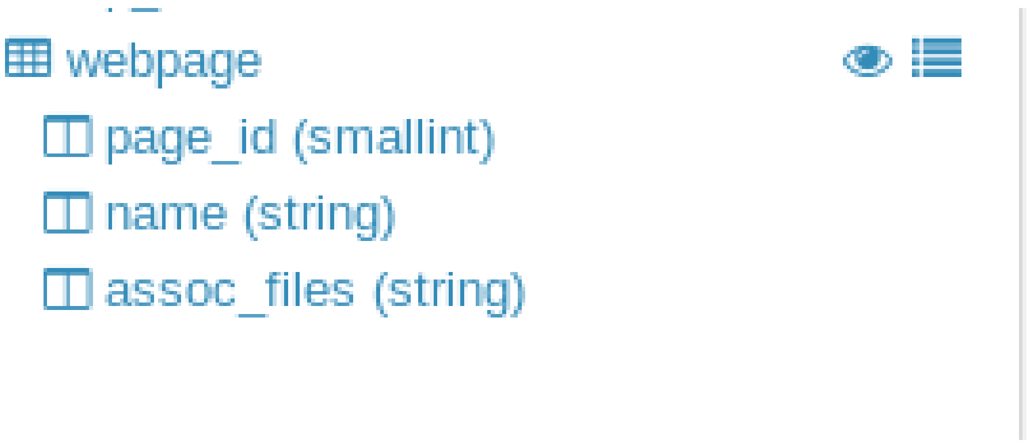
Visit the Hue page in Firefox, as described earlier in the “Using HDFS” exercise.

Open either the Impala query editor or Hive query editor, by selecting the editor of your choice from the **Query Editors** menu.



Practice – Hue Query Editors

1. Create the following table webpage using Hue query editors



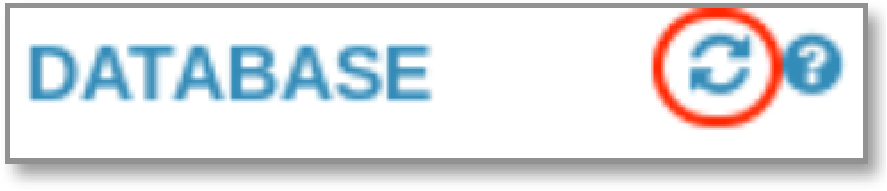
webpage

page_id (smallint)

name (string)

assoc_files (string)

2. To see the table you just created, refresh the table list on the left.

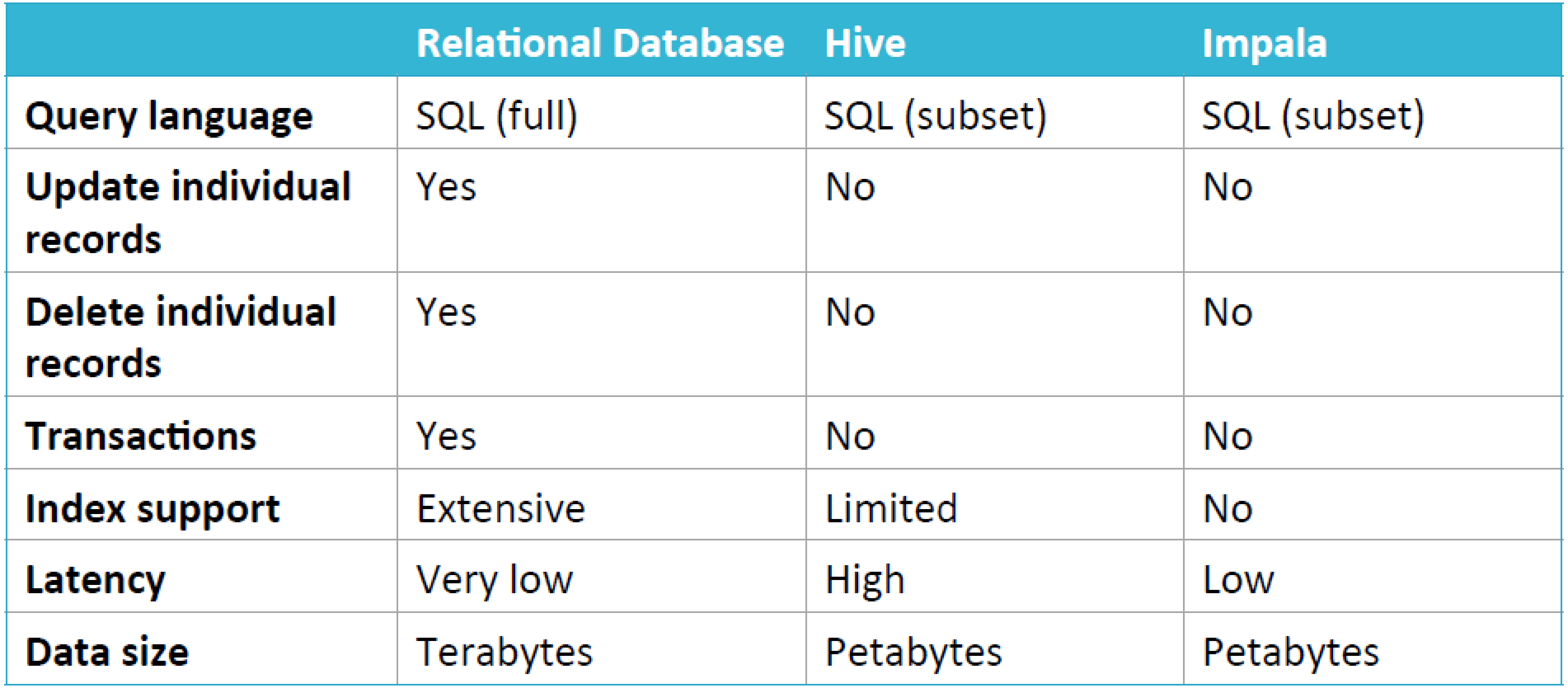


3. Click on the **webpage** table to see the column definitions.

Your Cluster is Not a Database Server

- **Client-server database management systems have many strengths**
  - Very fast response time
  - Support for transactions
  - Allow modification of existing records
  - Can serve thousands of simultaneous clients
- **Your Hadoop cluster is not an RDBMS**
  - Hive generates processing engine jobs (MapReduce) from HiveQL queries
    - Limitations of HDFS and MapReduce still apply
  - Impala is faster but not intended for the throughput speed required for an OLTP database
  - No transaction support

Comparing Hive and Impala to A Relational Database



	Relational Database	Hive	Impala
Query language	SQL (full)	SQL (subset)	SQL (subset)
Update individual records	Yes	No	No
Delete individual records	Yes	No	No
Transactions	Yes	No	No
Index support	Extensive	Limited	No
Latency	Very low	High	Low
Data size	Terabytes	Petabytes	Petabytes

Essential Points

- **Impala and Hive are tools for performing SQL queries on data in HDFS**
- **HiveQL and Impala SQL are very similar to SQL-92**
  - Easy to learn for those with relational database experience
  - However, does *not* replace your RDBMS
- **Hive generates jobs that run on the Hadoop cluster data processing engine**
  - Runs MapReduce jobs on Hadoop based on HiveQL statements
- **Impala execute queries directly on the Hadoop cluster**
  - Uses a very fast specialized SQL engine, not MapReduce