Hive

Jay Urbain, PhD

Hive

- SQL-like query language that generates MapReduce
- Developed at Facebook
- It uses H-SQL, not quite the same as ANSI SQL, ~70% overlap
- Underlying execution much different than relational engine
  - MapReduce
- Batch, not interactive.
  - Latency in results
  - Move to Spark, Impala
- Often used with HBase wide column database
- Create schema “on read”

HBase

- Wide-column, NoSQL database
- Use CREATE TABLE over HDFS
- Once your table is created, query with Hive H-SQL
- Hive libraries are integrated with HBase
- Hive libraries include the H-SQL/HQL language
- Commercial distributions include Hive

Basic Hive usage example:

- CREATE TABLE table col1, col2
- SELECT col1, col2 FROM table

Hive

Hive SQL Datatypes	Hive SQL Semantics
INT	SELECT, LOAD, INSERT from query
TINYINT/SMALLINT/BIGINT	Expressions in WHERE and HAVING
BOOLEAN	GROUP BY, ORDER BY, SORT BY
FLOAT	Sub-queries in FROM clause
DOUBLE	CLUSTER BY, DISTRIBUTE BY
STRING	ROLLUP and CUBE
TIMESTAMP	UNION
BINARY	LEFT, RIGHT, and FULL INNER/OUTER JOIN
ARRAY, MAP, STRUCT, UNION	CROSS JOIN, LEFT SEMI JOIN
DECIMAL	Windowing functions (OVER, RANK, etc.)
CHAR	INTERSECT, EXCEPT, UNION DISTINCT
VARCHAR	Subqueries in WHERE
DATE	Subqueries in HAVING

Hive 0.10
Hive 0.11
Future

Note: JOINs can have substantial overhead

Working with Hive

- Start the Hive service
  - Use admin tools to verify its started
  - Turned on by default with commercial distributions
- Managing the metastore database
  - Stores the metadata for Hive tables in a database
  - Can be embedded, local, or remote
  - Schema tool: offline tool for defining schema

Metadata

MySQL	Hive
USE database;	USE database;
SHOW DATABASES;	SHOW DATABASES;
SHOW TABLES;	SHOW TABLES;
DESCRIBE table;	DESCRIBE (FORMATTED | EXTENDED) table;
CREATE DATABASE db_name;	CREATE DATABASE db_name;
DROP DATABASE db_name;	DROP DATABASE db_name (CASCADE);

Why use Hive?

- You're a analyst, use databases, spreadsheets
- You know SQL
- Want to ask analytical questions, the questions you ask with SQL
  - Not as useful for data preprocessing
  - Use MapReduce, scripting language, Pig
- Gives convenience of SQL, but is *really* MapReduce.
  - JOINs function differently than on RDBMS
  - NoSQL has no need to be *normalized*, very expensive operation on big data

WordCount in Hive

- Pseudo code:
  1. Create HBase table for text input
  2. Load your data into the table
  3. Create a new destination table
  4. Load the table with processed results

WordCount in Hive

```
CREATE TABLE wordcount AS
SELECT word, count(1) AS count
FROM (SELECT EXPLODE(SPLIT(LCASE
   (REGEXP_REPLACE
      (line,'[\\p{Punct},\\p{Cntrl}]','')),' '))
AS word FROM myinput) words
GROUP BY word
ORDER BY count DESC, word ASC;
```

Regex is ugly, best to use something else for preprocessing

Using Hive

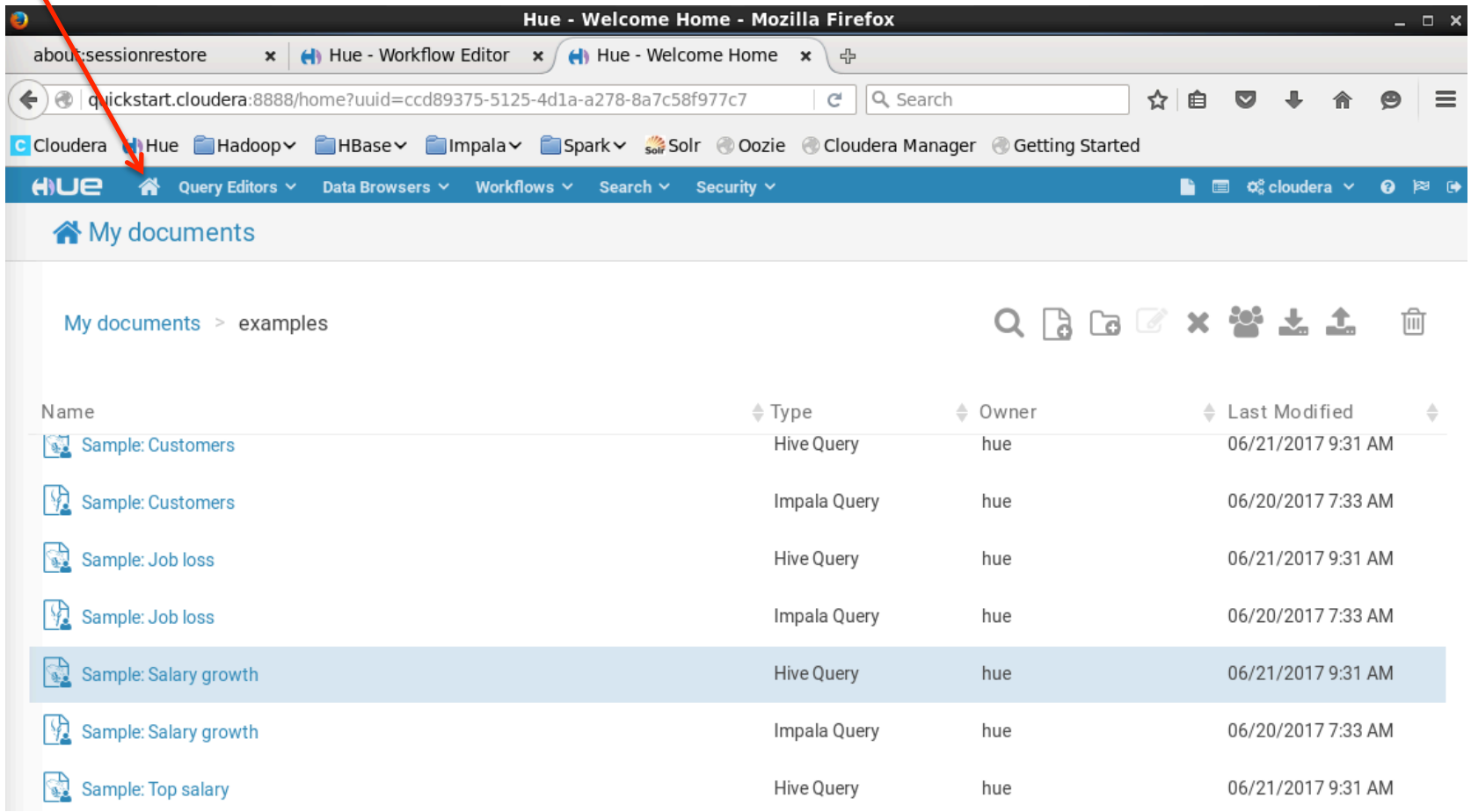
- HBase tables and Hive queries
- Query optimization for Hive
- Partitioning, bucketizing, or sampling (subsets)
- Cost-based optimization (CBO)
- Column based statistics

Reading HQL Query Plans

- Read from bottom to top:

```
[impalad-host:21000] > explain select count(*) from customer_address;
+----------------------------------------------------------+
| Explain String                                           |
+----------------------------------------------------------+
| Estimated Per-Host Requirements: Memory=42.00MB VCores=1 |
|                                                          |
| 03:AGGREGATE [MERGE FINALIZE]                            |
| |  output: sum(count())                                  |
| |                                                        |
| 02:EXCHANGE [PARTITION=UNPARTITIONED]                    |
| |                                                        |
| 01:AGGREGATE                                             |
| | output: count(*)                                       |
| |                                                        |
| 00:SCAN HDFS [default.customer_address]                  |
|    partitions=1/1 size=5.25MB                            |
+----------------------------------------------------------+
```

Hue Home -> examples -> Salary Growth (Hive Query)



Execute Query

Hue - Editor - Mozilla Firefox

about:sessionrestore | Hue - Workflow Editor | Hue - Editor

quickstart.cloudera:8888/notebook/editor?editor=5

Cloudera | Hue | Hadoop | HBase | Impala | Spark | Solr | Oozie | Cloudera Manager | Getting Started

Query Editors | Data Browsers | Workflows | Search | Security | cloudera

Hive | Sample: Salary growth | Salary growth (sorted) from 2007-08

default

Tables (4)

- customers
- sample_07
- sample_08
- web_logs

default

```
1 SELECT s07.description, s07.salary, s08.salary,
2   s08.salary - s07.salary
3 FROM
4   sample_07 s07 JOIN sample_08 s08
5 ON ( s07.code = s08.code)
6 WHERE
7  s07.salary < s08.salary
8 ORDER BY s08.salary-s07.salary DESC
9 LIMIT 1000
```

Query History | Saved Queries

Name	Description	Owner	Last Modified
Sample: Salary growth	Salary growth (sorted) from 2007-08	hue	06/21/2017 9:31 AM
Sample: Customers	Email Survey Opt-Ins, Customers for Shipping ZIP Code, Total Amount per Order	hue	06/21/2017 9:31 AM
Sample: Job loss	Job loss among the top earners 2007-08	hue	06/21/2017 9:31 AM
Sample: Top salary	Top salary 2007 above $100k	hue	06/21/2017 9:31 AM

Hue - Editor - Mozilla Firefox

about:sessionrestore | Hue - Workflow Editor | Hue - Editor

quickstart.cloudera:8888/notebook/editor?editor=50020

Cloudera | Hue | Hadoop | HBase | Impala | Spark | Solr | Oozie | Cloudera Manager | Getting Started

Query Editors | Data Browsers | Workflows | Search | Security | cloudera

Hive | Sample: Salary growth | Salary growth (sorted) from 2007-08

default

Tables (4)

- customers
- sample_07
- sample_08
- web_logs

55.117s default

```
SELECT s07.description, s07.salary, s08.salary,
  s08.salary - s07.salary
FROM
  sample_07 s07 JOIN sample_08 s08
ON ( s07.code = s08.code)
WHERE
 s07.salary < s08.salary
ORDER BY s08.salary-s07.salary DESC
LIMIT 1000
```

Query History | Saved Queries | Results (767)

	s07.description	s07.salary	s08.salary	_c3
1	Dentists, all other specialists	120360	142070	21710
2	Surgeons	191410	206770	15360
3	Oral and maxillofacial surgeons	178440	190420	11980
4	Natural sciences managers	113170	123140	9970
5	Physicians and surgeons, all other	155150	165000	9850
6	Orthodontists	185340	194930	9590
7	Internists, general	167270	176740	9470
8	Political scientists	90050	99320	9270
9	Obstetricians and gynecologists	183600	192780	9180
10	Chief executives	151370	160440	9070

Explain

Hue - Editor - Mozilla Firefox

about:sessionrestore | Hue - Workflow Editor | Hue - Editor

quickstart.cloudera:8888/notebook/editor?editor=50020

Cloudera | Hue | Hadoop | HBase | Impala | Spark | Solr | Oozie | Cloudera Manager | Getting Started

Query Editors | Data Browsers | Workflows | Search | Security | cloudera

Hive | Sample: Salary growth | Salary growth (sorted) from 2007-08

default

Tables (4)

- customers
- sample_07
- sample_08
- web_logs

55.117s default

```
1 SELECT s07.description, s07.salary, s08.salary,
2   s08.salary - s07.salary
3 FROM
4   sample_07 s07 JOIN sample_08 s08
5 ON ( s07.code = s08.code)
6 WHERE
7  s07.salary < s08.salary
8 ORDER BY s08.salary-s07.salary DESC
9 LIMIT 1000
```

Explain
Format
Clear

Saved Queries | Results (767)

	ription	s07.salary	s08.salary	_c3
1	Dentists, all other specialists	120360	142070	21710
2	Surgeons	191410	206770	15360
3	Oral and maxillofacial surgeons	178440	190420	11980
4	Natural sciences managers	113170	123140	9970
5	Physicians and surgeons, all other	155150	165000	9850
6	Orthodontists	185340	194930	9590
7	Internists, general	167270	176740	9470
8	Political scientists	90050	99320	9270
9	Obstetricians and gynecologists	183600	192780	9180
10	Chief executives	151370	160440	9070

Hue - Editor - Mozilla Firefox

about:sessionrestore | Hue - Workflow Editor | Hue - Editor

quickstart.cloudera:8888/notebook/editor?editor=50020

Cloudera | Hue | Hadoop | HBase | Impala | Spark | Solr | Oozie | Cloudera Manager | Getting Started

Query Editors | Data Browsers | Workflows | Search | Security | cloudera

Hive | Sample: Salary growth | Salary growth (sorted) from 2007-08

default

Tables (4)

- customers
- sample_07
- sample_08
- web_logs

default

```
1 SELECT s07.description, s07.salary, s08.salary,
2   s08.salary - s07.salary
3 FROM
4   sample_07 s07 JOIN sample_08 s08
5 ON ( s07.code = s08.code)
6 WHERE
7  s07.salary < s08.salary
8 ORDER BY s08.salary-s07.salary DESC
9 LIMIT 1000
```

Query History | Saved Queries | Results (767) | Explain

```
STAGE DEPENDENCIES:
  Stage-5 is a root stage
  Stage-2 depends on stages: Stage-5
  Stage-0 depends on stages: Stage-2

STAGE PLANS:
  Stage: Stage-5
    Map Reduce Local Work
      Alias -> Map Local Tables:
        s07
          Fetch Operator
            limit: -1
      Alias -> Map Local Operator Tree:
        s07
          TableScan
            alias: s07
            Statistics: Num rows: 225 Data size: 46055 Basic stats: COMPLETE Column stats: NONE
            Filter Operator
              predicate: code is not null (type: boolean)
              Statistics: Num rows: 113 Data size: 23129 Basic stats: COMPLETE Column stats: NONE
              HashTable Sink Operator
                keys:
                  0 code (type: string)
                  1 code (type: string)
```

Review

- All are true about Hive, except ____
  - It is SQL-like query
  - It uses H-SQL
  - It is used in conjunction with HBase
  - It is interactive

- Which of the following data types was supported in Hive 0.11
  - Varchar
  - Binary
  - Float
  - Decimal

- People who are mostly Analysts who know SQL, and who want to ask analytical questions will most likely use Hive over other methods
  - False
  - True

- Some of the query optimization for Hive includes which of the following? All of these answers
  - Partitioning, bucketizing, or sampling
  - Using column-based statistics
  - Cost-base optimization
  - All of the above