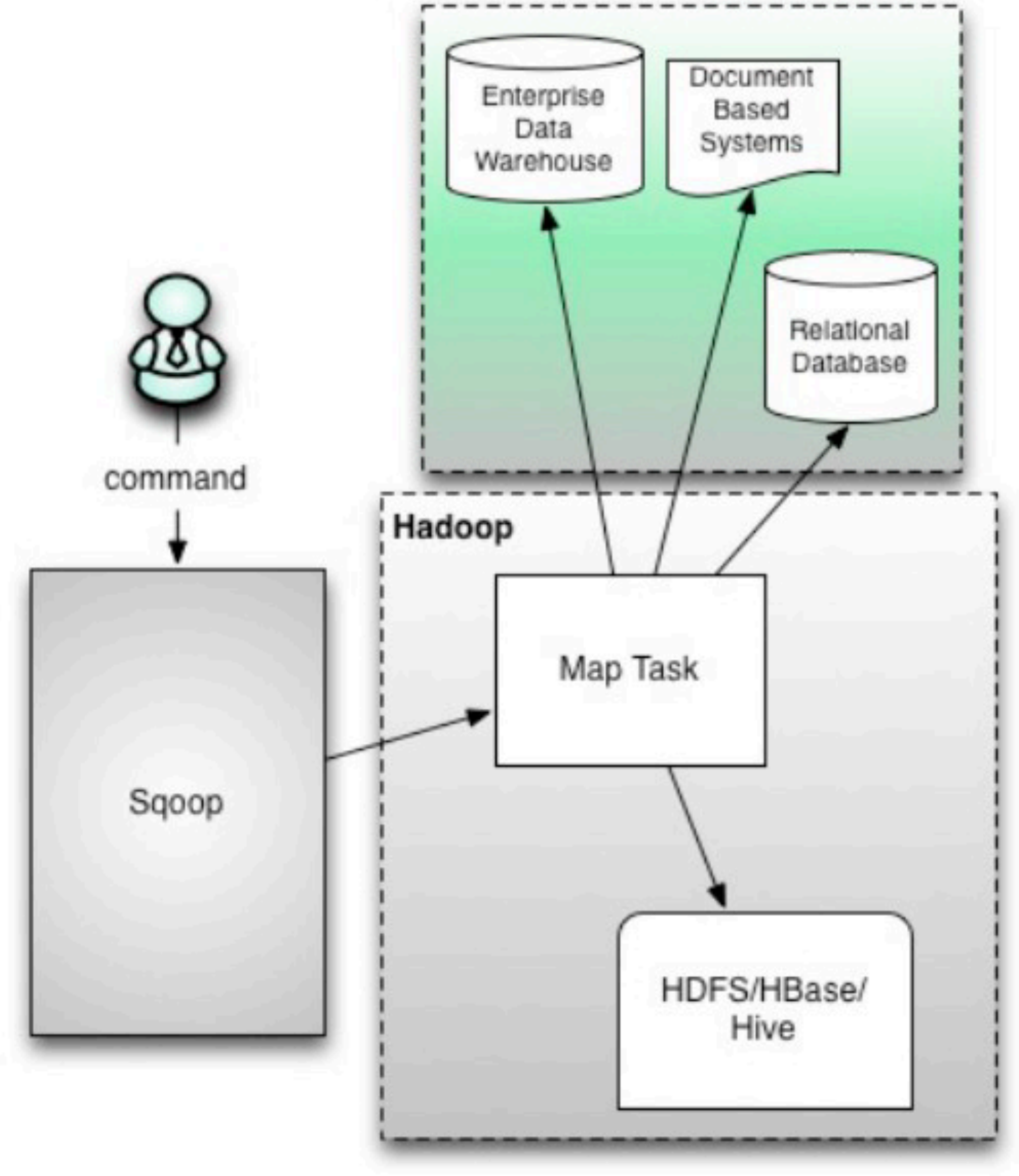
APACHE SQOOP

Introduction

Sqoop (“SQL-to-Hadoop”) is a straightforward command-line tool with the following capabilities:

- Imports individual tables or entire databases to files in HDFS
- Generates Java classes to allow you to interact with your imported data
- Provides the ability to import from SQL databases straight into your Hive data warehouse

Architecture Overview



sqoop.apache.org

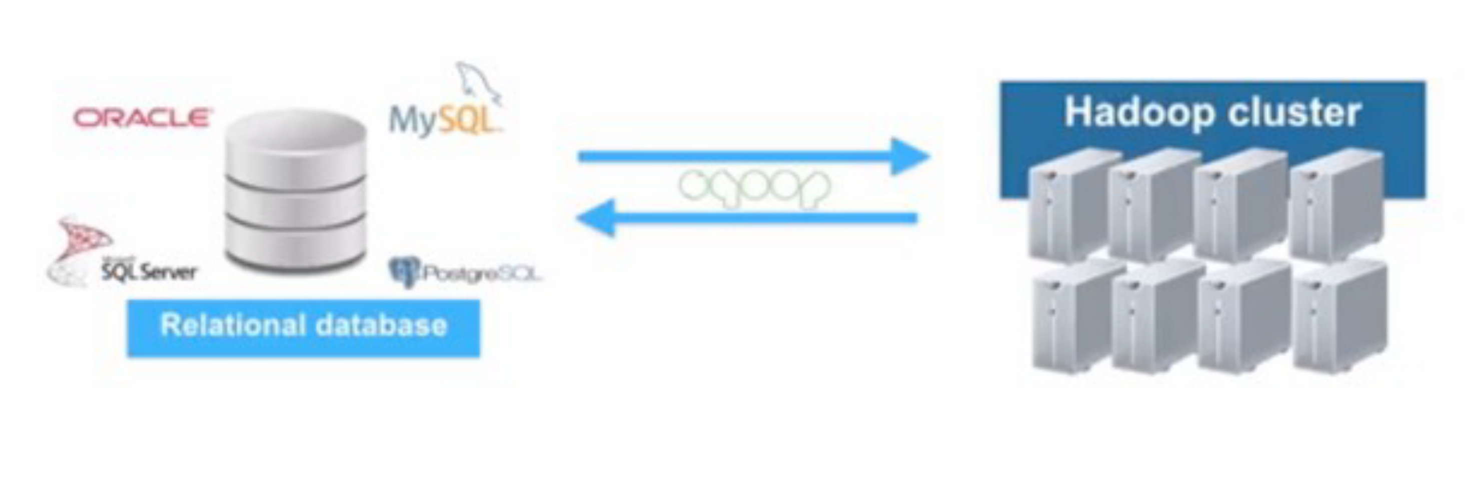
Sqoop Benefit

- Leverages RDBMS metadata to get the column data types
- It is simple to script and uses SQL
- It can be used to handle change data capture by importing daily transactional data to Hadoop
- It uses MapReduce for export and import that enables parallel and efficient data movement

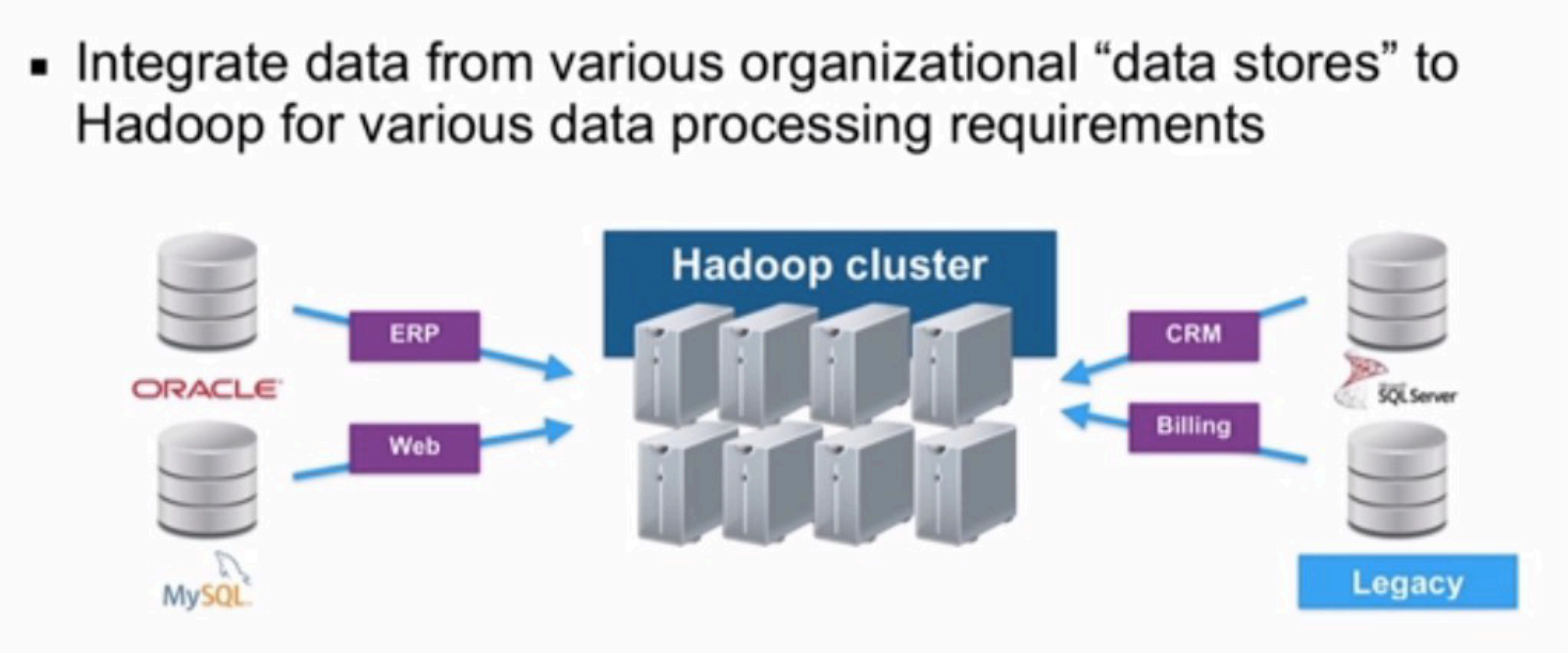
Sqoop Mode

Sqoop import: Data moves from RDBMS to Hadoop

Sqoop export: Data moves from Hadoop to RDBMS



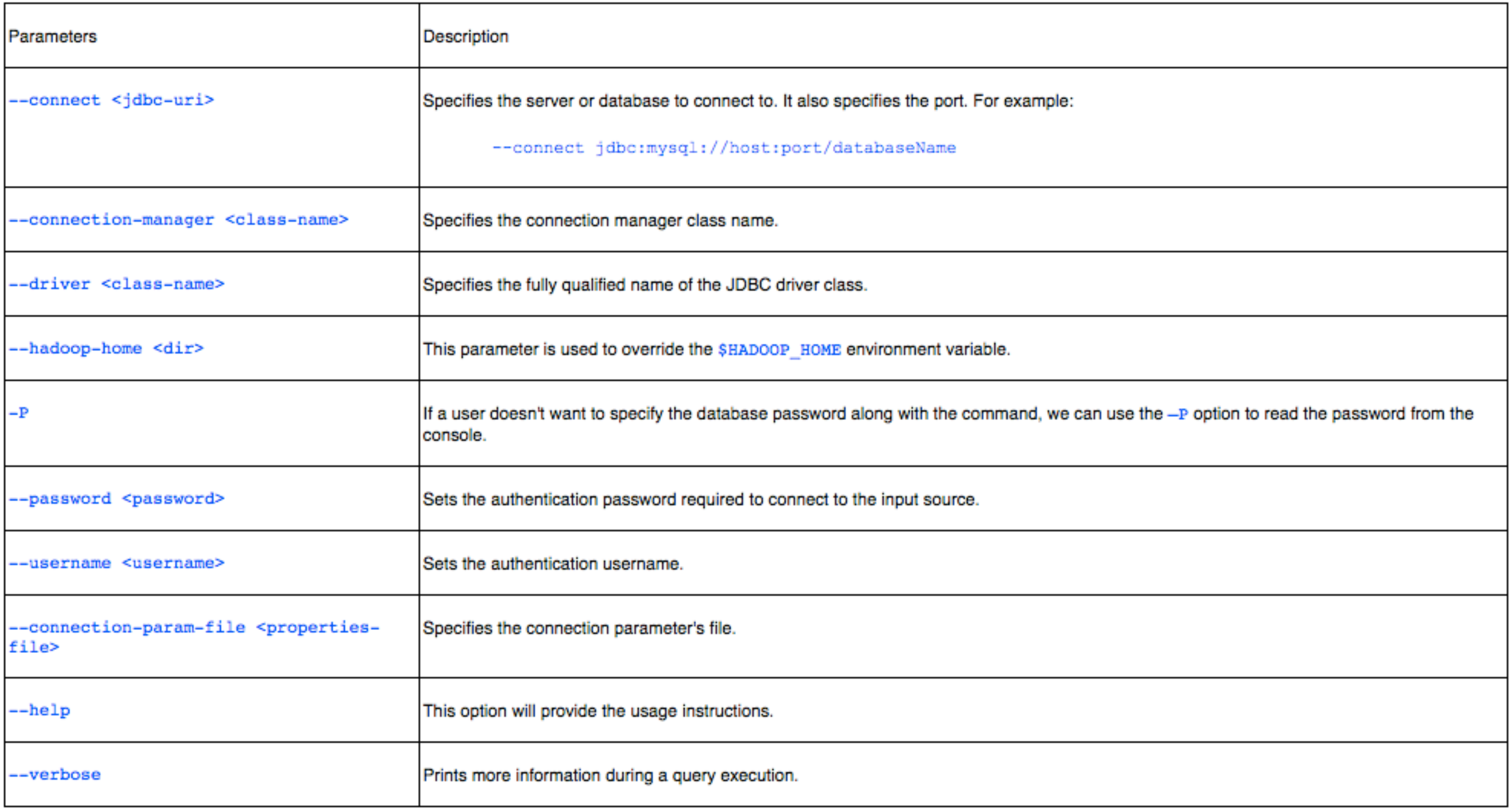
Use Case : Data Consolidation



- Integrate data from various organizational “data stores” to Hadoop for various data processing requirements

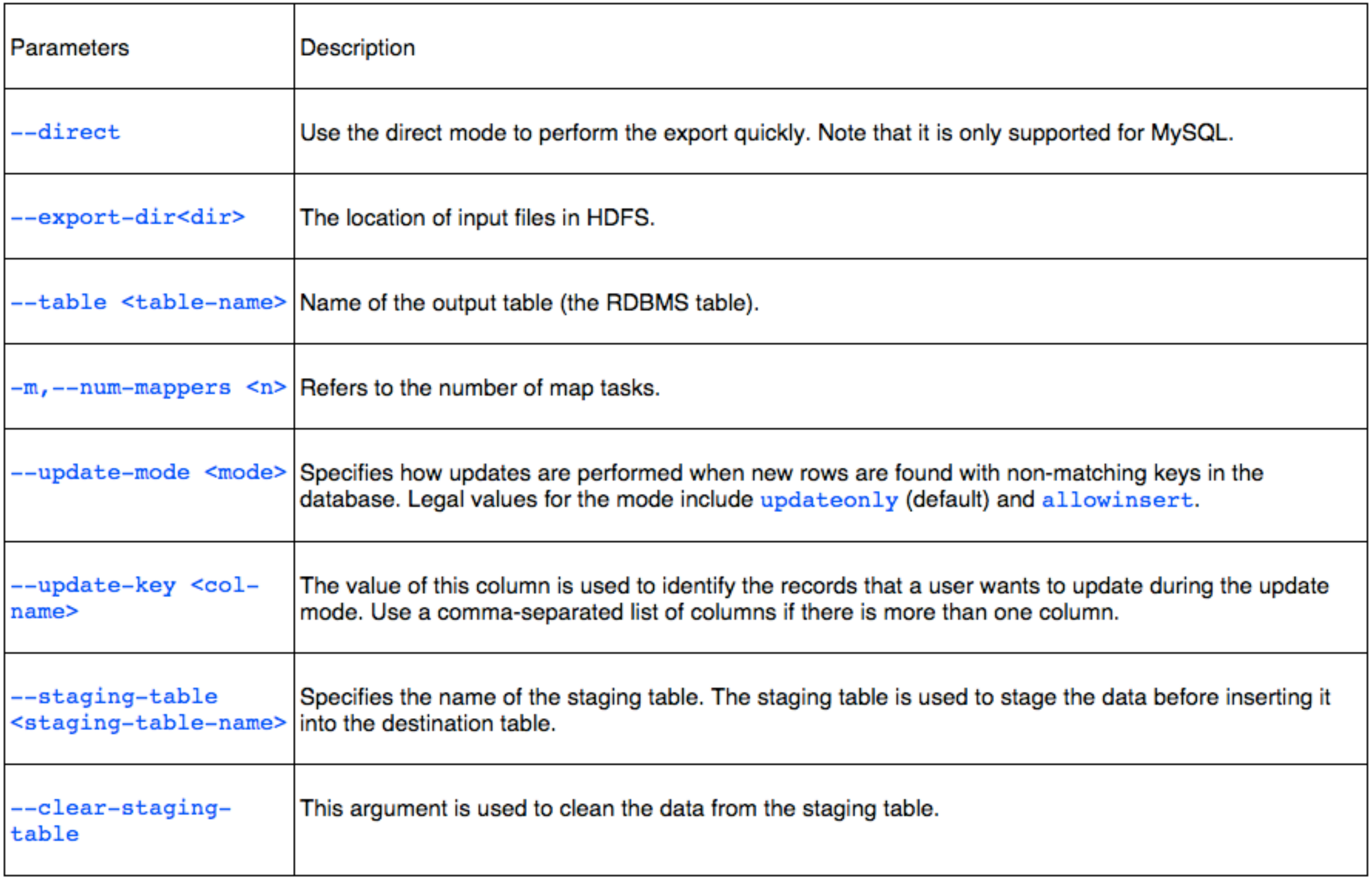
Source: Mastering Apache Sqoop, David Yahalom, 2016

Import Commands



Parameters	Description
`--connect <jdbc-uri>`	Specifies the server or database to connect to. It also specifies the port. For example: `--connect jdbc:mysql://host:port/databaseName`
`--connection-manager <class-name>`	Specifies the connection manager class name.
`--driver <class-name>`	Specifies the fully qualified name of the JDBC driver class.
`--hadoop-home <dir>`	This parameter is used to override the `$HADOOP_HOME` environment variable.
`-P`	If a user doesn't want to specify the database password along with the command, we can use the `–P` option to read the password from the console.
`--password <password>`	Sets the authentication password required to connect to the input source.
`--username <username>`	Sets the authentication username.
`--connection-param-file <properties-file>`	Specifies the connection parameter's file.
`--help`	This option will provide the usage instructions.
`--verbose`	Prints more information during a query execution.

Export Commands



Parameters	Description
`--direct`	Use the direct mode to perform the export quickly. Note that it is only supported for MySQL.
`--export-dir<dir>`	The location of input files in HDFS.
`--table <table-name>`	Name of the output table (the RDBMS table).
`-m,--num-mappers <n>`	Refers to the number of map tasks.
`--update-mode <mode>`	Specifies how updates are performed when new rows are found with non-matching keys in the database. Legal values for the mode include `updateonly` (default) and `allowinsert`.
`--update-key <col-name>`	The value of this column is used to identify the records that a user wants to update during the update mode. Use a comma-separated list of columns if there is more than one column.
`--staging-table <staging-table-name>`	Specifies the name of the staging table. The staging table is used to stage the data before inserting it into the destination table.
`--clear-staging-table`	This argument is used to clean the data from the staging table.

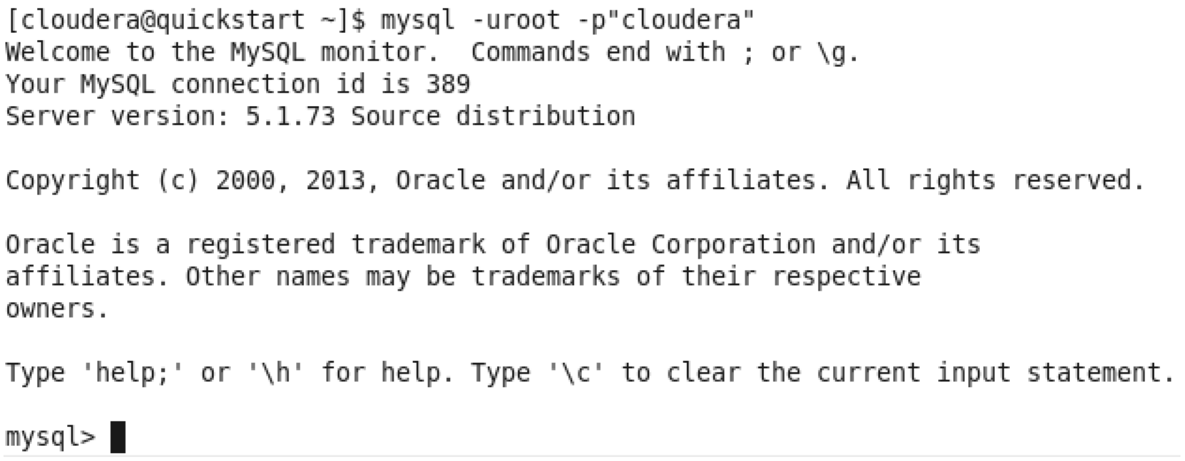
Loading Data from RDBMS to Hadoop

Configuring MySQL On Cloudera.Quickstart

```
$ sudo /usr/bin/mysql_secure_installation
Enter current password for root (enter for none): cloudera
OK, successfully used password, moving on...
Set root password? [Y/n] N
Remove anonymous users? [Y/n] Y
Disallow root login remotely? [Y/n] N
Remove test database and access to it [Y/n] Y
Reload privilege tables now? [Y/n] Y
All done!
```

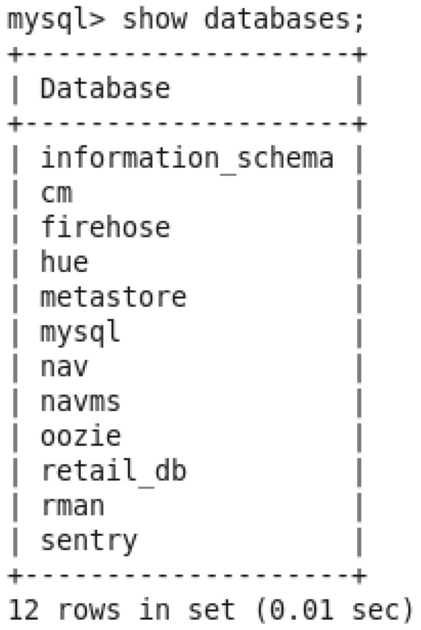
Running MySQL

$ mysql -uroot -p"cloudera"



```
[cloudera@quickstart ~]$ mysql -uroot -p"cloudera"
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 389
Server version: 5.1.73 Source distribution

Copyright (c) 2000, 2013, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> █
```

mysql> show databases;



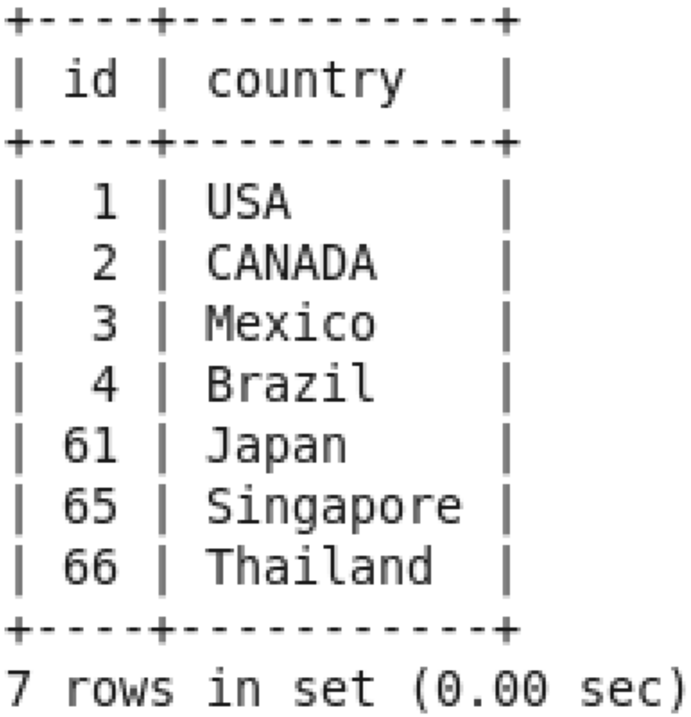
```
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| cm                 |
| firehose           |
| hue                |
| metastore          |
| mysql              |
| nav                |
| navms              |
| oozie              |
| retail_db          |
| rman               |
| sentry             |
+--------------------+
12 rows in set (0.01 sec)
```

Prepare a test database table

```
mysql> CREATE DATABASE test_mysql_db;
mysql> USE test_mysql_db;
mysql> CREATE TABLE country_tbl(id INT NOT NULL, country
VARCHAR(50), PRIMARY KEY (id));
mysql> INSERT INTO country_tbl VALUES(1, 'USA');
mysql> INSERT INTO country_tbl VALUES(2, 'CANADA');
mysql> INSERT INTO country_tbl VALUES(3, 'Mexico');
mysql> INSERT INTO country_tbl VALUES(4, 'Brazil');
mysql> INSERT INTO country_tbl VALUES(61, 'Japan');
mysql> INSERT INTO country_tbl VALUES(65, 'Singapore');
mysql> INSERT INTO country_tbl VALUES(66, 'Thailand');
```

View data in the table

mysql> SELECT * FROM country_tbl;

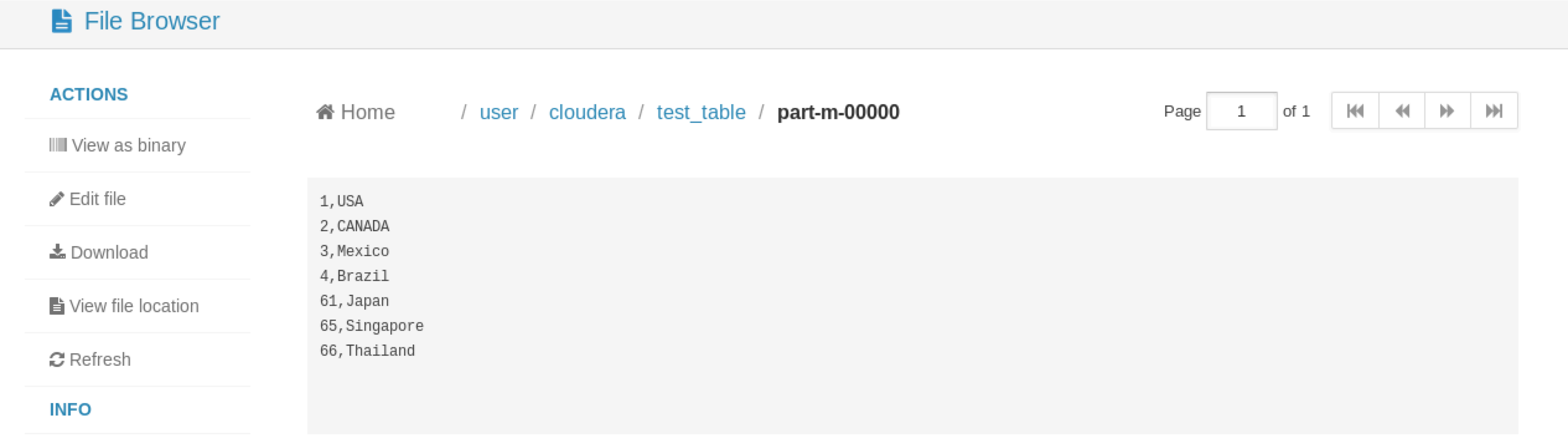


```
+----+-----------+
| id | country   |
+----+-----------+
|  1 | USA       |
|  2 | CANADA    |
|  3 | Mexico    |
|  4 | Brazil    |
| 61 | Japan     |
| 65 | Singapore |
| 66 | Thailand  |
+----+-----------+
7 rows in set (0.00 sec)
```

mysql> exit;

Importing data from MySQL to HDFS

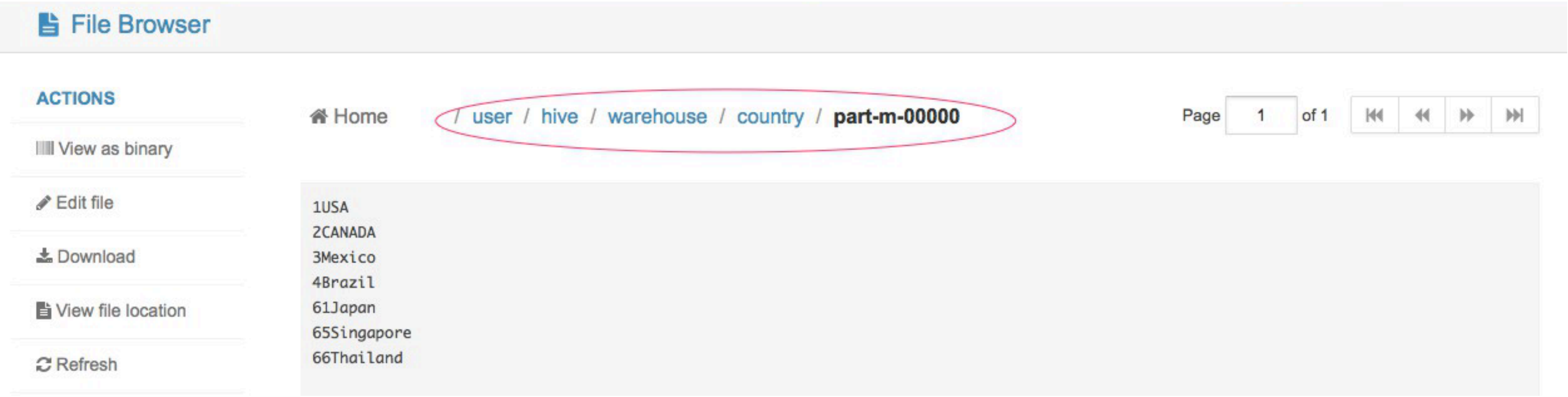
$ sqoop import --connect jdbc:mysql://localhost/test_mysql_db --username root --password cloudera --table country_tbl --target-dir /user/cloudera/test_table -m 1



```
1,USA
2,CANADA
3,Mexico
4,Brazil
61,Japan
65,Singapore
66,Thailand
```

Importing data from MySQL to Hive Table

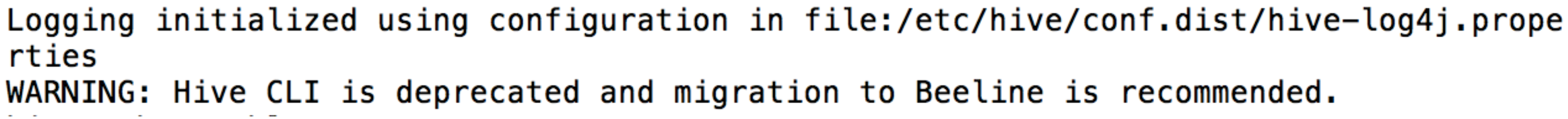
$ sqoop import --connect jdbc:mysql://localhost/test_mysql_db --username root --password cloudera --table country_tbl --hive-import --hive-table country -m 1



```
1USA
2CANADA
3Mexico
4Brazil
61Japan
65Singapore
66Thailand
```

Reviewing data from Hive Table

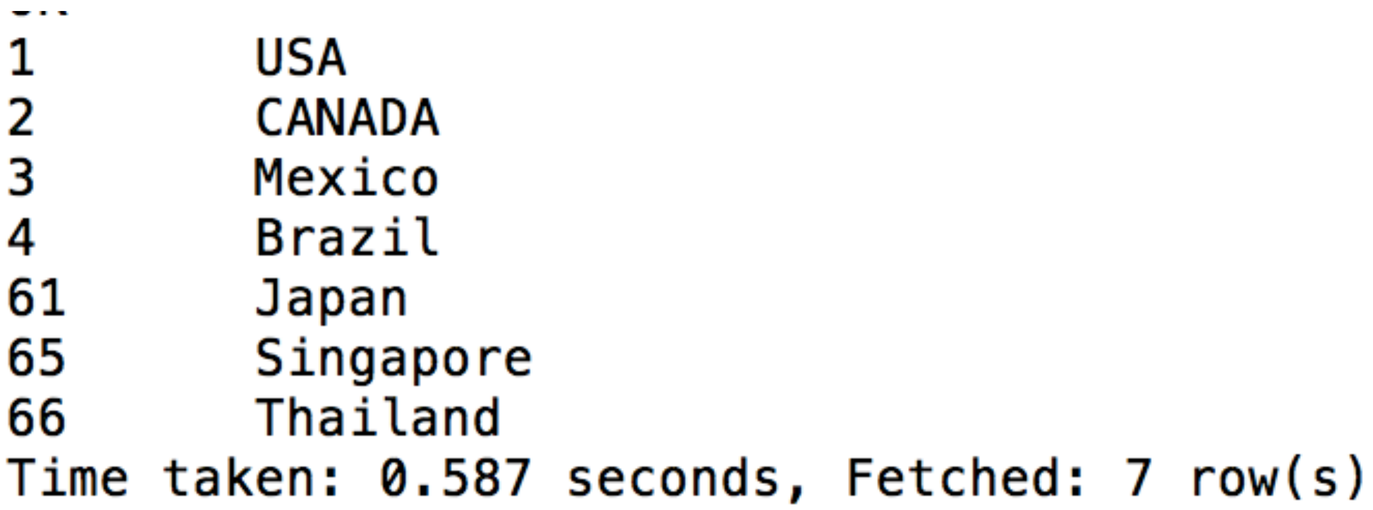
[cloudera@quickstart ~]$ hive



```
Logging initialized using configuration in file:/etc/hive/conf.dist/hive-log4j.prope
rties
WARNING: Hive CLI is deprecated and migration to Beeline is recommended.
```

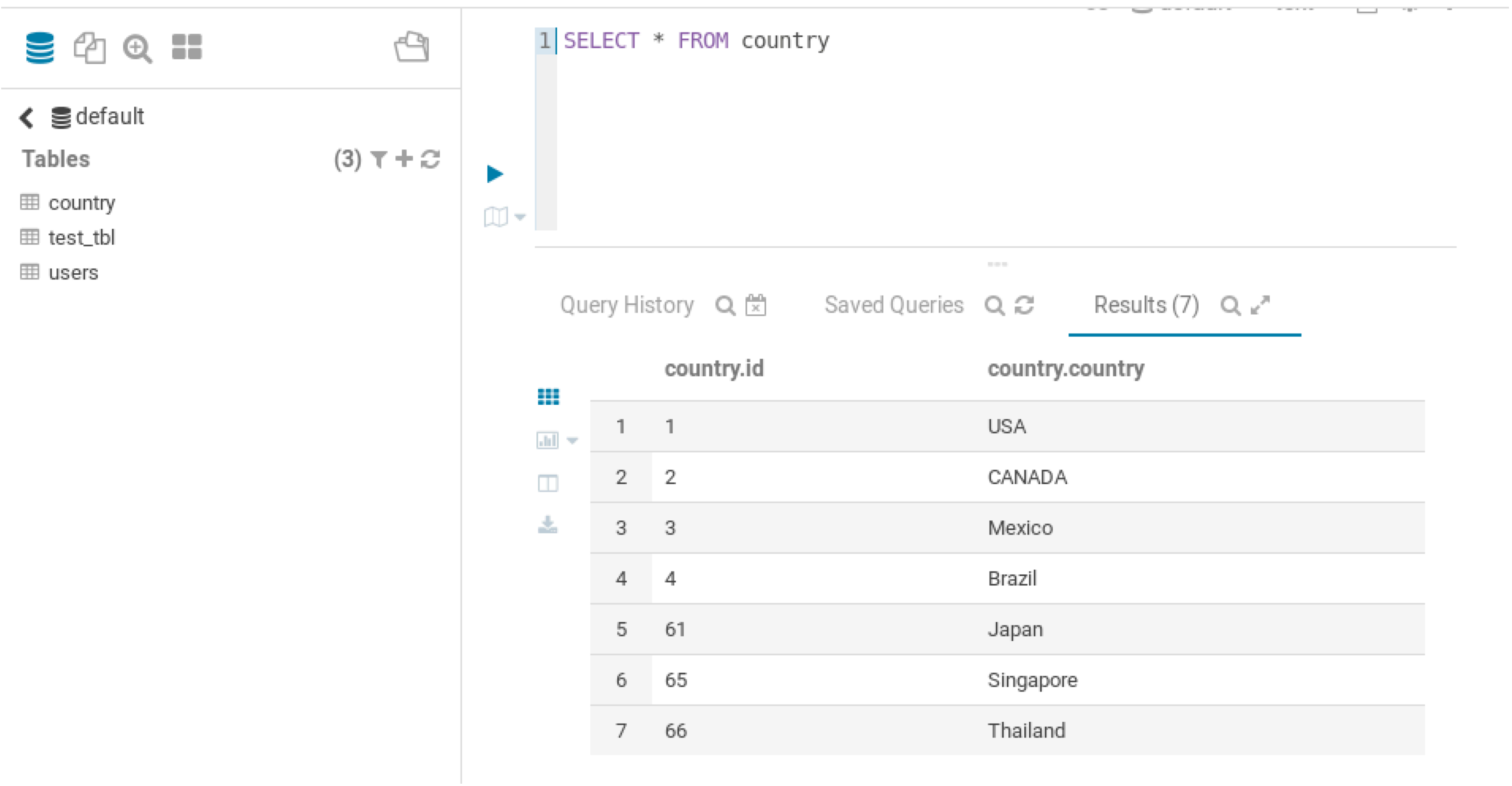
hive> show tables;

hive> select * from country;



```
1        USA
2        CANADA
3        Mexico
4        Brazil
61       Japan
65       Singapore
66       Thailand
Time taken: 0.587 seconds, Fetched: 7 row(s)
```

Running from Hue: Beewax



```
SELECT * FROM country
```

	country.id	country.country
1	1	USA
2	2	CANADA
3	3	Mexico
4	4	Brazil
5	61	Japan
6	65	Singapore
7	66	Thailand

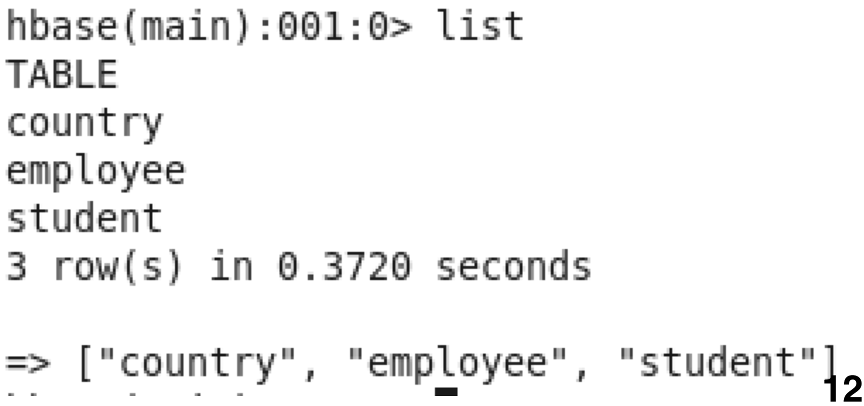
Importing data from MySQL to HBase

$ sqoop import --connect jdbc:mysql://localhost/test_mysql_db --username root --password cloudera --table country_tbl --hbase-table country --column-family hbase_country_cf --hbase-row-key id --hbase-create-table -m 1

Start HBase

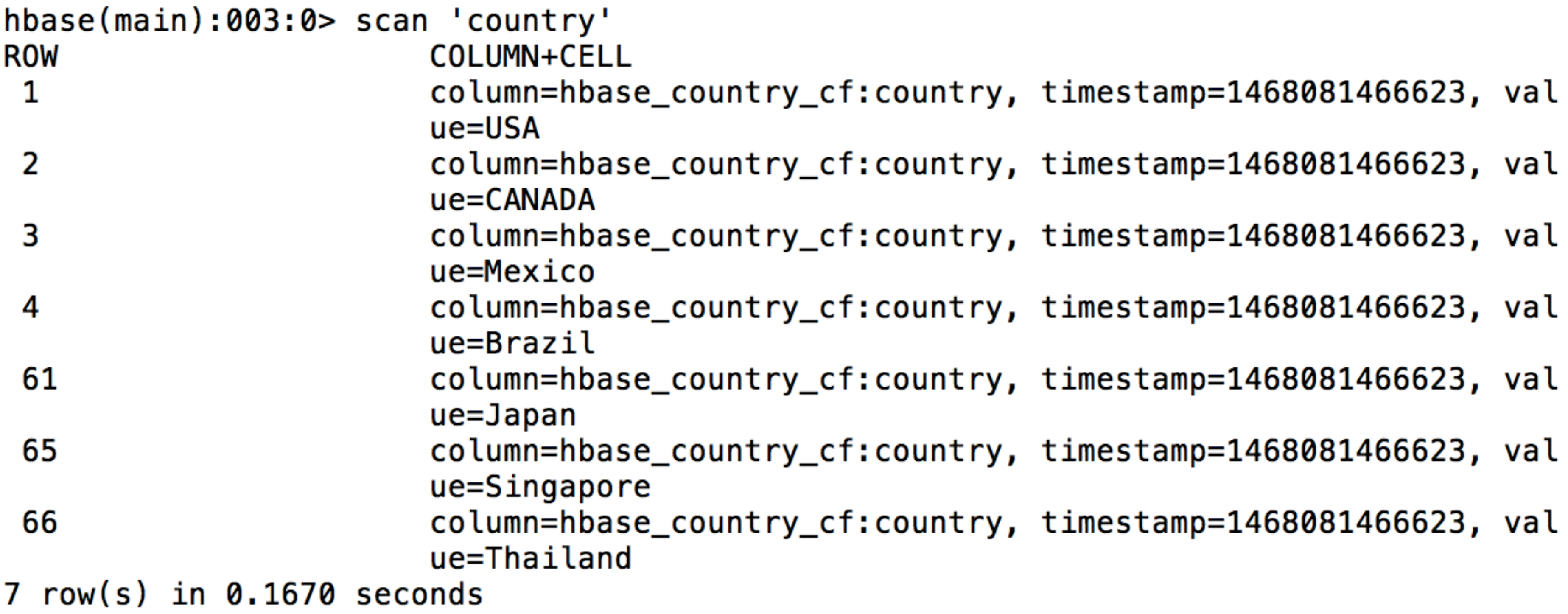
$ hbase shell

hbase(main):001:0> list



```
hbase(main):001:0> list
TABLE
country
employee
student
3 row(s) in 0.3720 seconds

=> ["country", "employee", "student"]
```

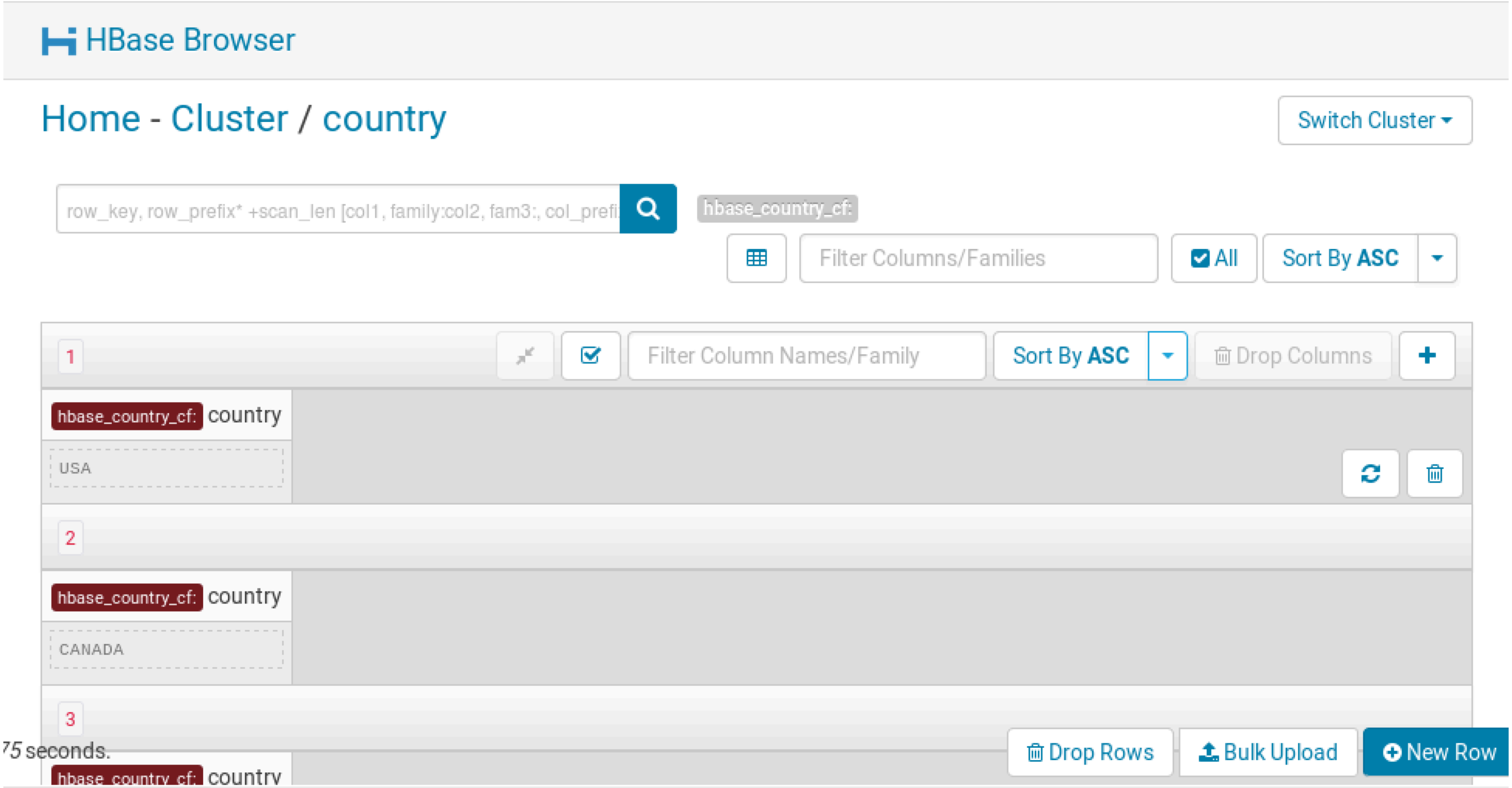
Viewing Hbase data



```
hbase(main):003:0> scan 'country'
ROW                    COLUMN+CELL
 1                     column=hbase_country_cf:country, timestamp=1468081466623, val
                       ue=USA
 2                     column=hbase_country_cf:country, timestamp=1468081466623, val
                       ue=CANADA
 3                     column=hbase_country_cf:country, timestamp=1468081466623, val
                       ue=Mexico
 4                     column=hbase_country_cf:country, timestamp=1468081466623, val
                       ue=Brazil
 61                    column=hbase_country_cf:country, timestamp=1468081466623, val
                       ue=Japan
 65                    column=hbase_country_cf:country, timestamp=1468081466623, val
                       ue=Singapore
 66                    column=hbase_country_cf:country, timestamp=1468081466623, val
                       ue=Thailand
7 row(s) in 0.1670 seconds
```

Viewing data from Hbase browser

CHULALONGKORN BUSINESS SCHOOL
FLAGSHIP FOR LIFE



HBase Browser

Home - Cluster / country

1

hbase_country_cf: country

USA

2

hbase_country_cf: country

CANADA

3

hbase_country_cf: country

75 seconds.