

Aim: Use Sqoop to load data from RDBMS (weblog/transactions data) and analyze it using Hive.

Theory:

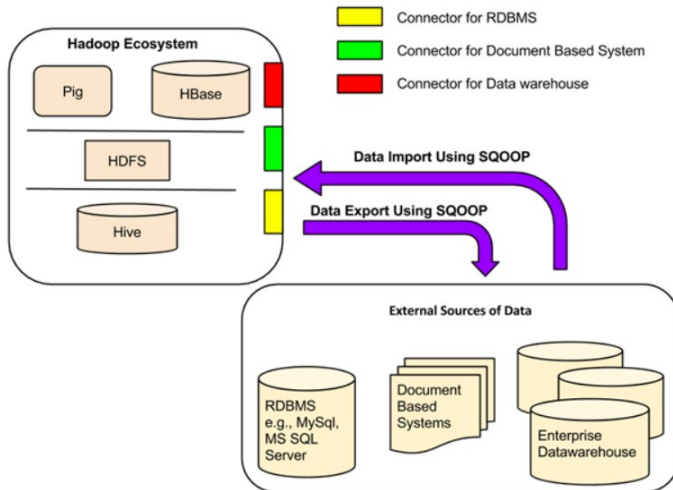
Introduction to Sqoop:

Sqoop (SQL-to-Hadoop) is an open-source data transfer tool that facilitates the efficient movement of data between Apache Hadoop and structured data stores, such as relational databases. It was developed by the Apache Software Foundation and is widely used in the big data ecosystem to import data from relational databases into Hadoop's HDFS (Hadoop Distributed File System) and export data from Hadoop to relational databases.

The primary use case for Sqoop is to bridge the gap between the traditional relational database world and the Hadoop ecosystem, allowing organizations to leverage the power of Hadoop for processing and analysis while still being able to work with their existing data stored in relational databases.

Architecture of Sqoop:

Using its connectors, sqoop facilitates data migration between Hadoop and external storage systems. These connectors enable Sqoop to work with various widely-used relational databases, such as MySQL, PostgreSQL, Oracle, SQL Server, and DB2. Each connector establishes communication with the corresponding DBMS it is associated with. A generic JDBC connector is also available for connecting to any database that adheres to the JDBC standard. Furthermore, Sqoop Big Data offers specialized connectors optimized for PostgreSQL and MySQL, leveraging database-specific APIs for enhanced performance.



Advantages of Sqoop:

1. It entails data transfer from numerous structured sources, like Oracle, Postgres, etc.
2. Due to the parallel data transport, it is quick and efficient.
3. Many procedures can be automated, which increases efficiency.
4. Integration with Kerberos security authentication is feasible.
5. Direct data loading is possible from HBase and Hive.
6. It is a powerful tool with a sizable support network.
7. As a result of its ongoing development and contributions, it is frequently updated.

Issues in Sqoop:

Data load using Scripts

The traditional approach of using scripts to load data is not suitable for bulk data load into Hadoop; this approach is inefficient and very time-consuming.

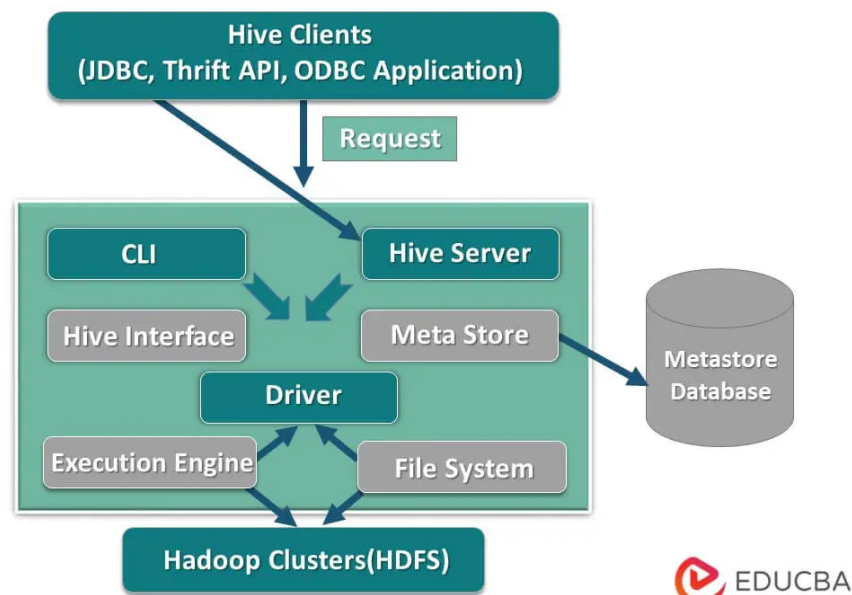
Direct access to external data via Map-Reduce application

Providing direct access to the data residing at external systems (without loading into Hadoop) for map-reduce applications complicates these applications. So, this approach is not feasible.

Introduction to Hive:

Apache Hive is an open-source data warehousing and SQL-like query language tool built on top of the Hadoop ecosystem. It was developed by the Apache Software Foundation to provide a high-level interface for querying and analyzing large datasets stored in Hadoop's distributed storage system, HDFS (Hadoop Distributed File System). Hive is particularly useful for users familiar with SQL as it allows them to work with big data using a familiar query language.

Architecture of Hive:



Hive Clients

They include Thrift application to execute easy hive commands which are available for python, ruby, C++, and drivers. These client application benefits for executing queries on the hive. Hive has three types of client categorization: thrift clients, JDBC, and ODBC clients.

Hive Services

To process all the queries hive has various services. All the functions are easily defined by the user in the hive. Let's see all those services in brief:

Command-line Interface (CLI): User interface for executing Hive commands and queries. Web interfaces (HWI) can also be used for query submission.

Hive Driver: Receives queries from various sources, interacts with ODBC/JDBC drivers, and manages query execution stages.

Compiler: Parses and performs semantic analysis on queries, creating abstract syntax trees and optimizing query execution plans.

Execution Engine: Processes queries by executing tasks, managing task dependencies, and executing DAG stages.

Metastore: Central repository storing metadata about tables, partitions, and HDFS file storage. Manages structured information and acts as a namespace for tables.

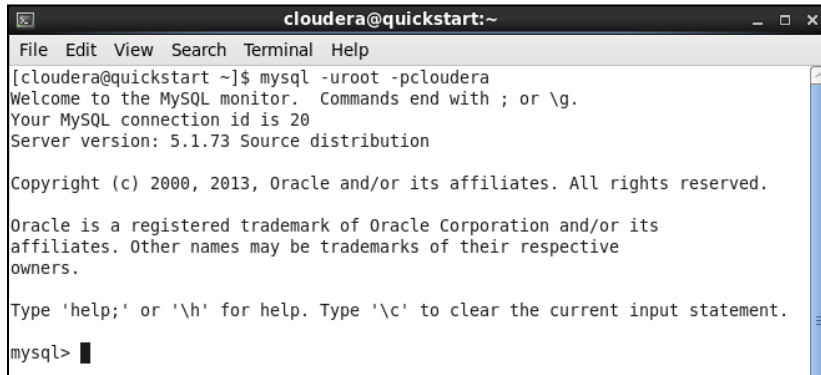
Characteristics of Hive:

1. Databases and tables are built before loading the data.
2. Hive as a data warehouse is built to manage and query only structured data which is residing under tables.
3. At the time of handling structured data, MapReduce lacks optimization and usability functions such as UDFs whereas Hive framework have optimization and usability.
4. Programming in Hadoop deals directly with the files. So, Hive can partition the data with directory structures to improve performance on certain queries.
5. Hive is compatible for the various file formats which are TEXTFILE, SEQUENCEFILE, ORC, RCFILE, etc.
6. Hive uses derby database in single user metadata storage and it uses MYSQL for multiple user Metadata or shared Metadata.

Implementation:

1. Entering into the MySQL command line

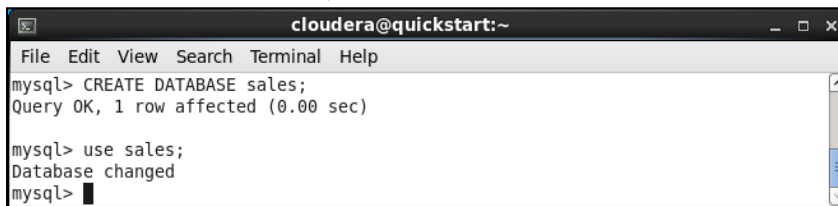
Command: `mysql -uroot -pcloudera`

A terminal window titled 'cloudera@quickstart:~' with a menu bar (File, Edit, View, Search, Terminal, Help). The prompt is '[cloudera@quickstart ~]\$'. The command 'mysql -uroot -pcloudera' has been entered. The output shows the MySQL welcome message: 'Welcome to the MySQL monitor. Commands end with ; or \g. Your MySQL connection id is 20. 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.' The prompt has changed to 'mysql>' with a cursor.

```
cloudera@quickstart:~  
File Edit View Search Terminal Help  
[cloudera@quickstart ~]$ mysql -uroot -pcloudera  
Welcome to the MySQL monitor.  Commands end with ; or \g.  
Your MySQL connection id is 20  
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> █
```

2. Creating a 'sales' database in mysql

Command: `CREATE DATABASE sales;`
`use sales;`

A terminal window titled 'cloudera@quickstart:~' with a menu bar (File, Edit, View, Search, Terminal, Help). The prompt is 'mysql>'. The command 'CREATE DATABASE sales;' has been entered, followed by the output 'Query OK, 1 row affected (0.00 sec)'. The next command 'use sales;' has been entered, followed by the output 'Database changed'. The prompt is now 'mysql>' with a cursor.

```
cloudera@quickstart:~  
File Edit View Search Terminal Help  
mysql> CREATE DATABASE sales;  
Query OK, 1 row affected (0.00 sec)  
  
mysql> use sales;  
Database changed  
mysql> █
```

3. Creating a table in the mysql

Command: CREATE TABLE sales (invoice_id INT(10) NOT NULL PRIMARY KEY, city VARCHAR(20), `product_line` VARCHAR(20), unit_price FLOAT, quantity INT(10), tax FLOAT, total FLOAT, `date` DATETIME, `time` DATETIME, payment VARCHAR(20), cogs FLOAT, rating INT);
desc sales;

```
cloudera@quickstart:~  
File Edit View Search Terminal Help  
mysql> use sales;  
Database changed  
mysql> CREATE TABLE sales(invoice_id INT(10) not null primary key, city VARCHAR(20), product_line VARCHAR(20), unit_price FLOAT, quantity INT(10), tax FLOAT, total FLOAT, date datetime, time datetime, payment VARCHAR(20), cogs FLOAT, rating INT);  
Query OK, 0 rows affected (2.40 sec)  
  
mysql> desc sales;  
+-----+-----+-----+-----+-----+-----+  
| Field      | Type          | Null | Key | Default | Extra |  
+-----+-----+-----+-----+-----+-----+  
| invoice_id | int(10)       | NO   | PRI | NULL    |       |  
| city       | varchar(20)   | YES  |     | NULL    |       |  
| product_line | varchar(20)   | YES  |     | NULL    |       |  
| unit_price | float         | YES  |     | NULL    |       |  
| quantity   | int(10)       | YES  |     | NULL    |       |  
| tax        | float         | YES  |     | NULL    |       |  
| total      | float         | YES  |     | NULL    |       |  
| date       | datetime      | YES  |     | NULL    |       |  
| time       | datetime      | YES  |     | NULL    |       |  
| payment    | varchar(20)   | YES  |     | NULL    |       |  
| cogs       | float         | YES  |     | NULL    |       |  
| rating     | int(11)       | YES  |     | NULL    |       |  
+-----+-----+-----+-----+-----+-----+  
12 rows in set (0.95 sec)
```

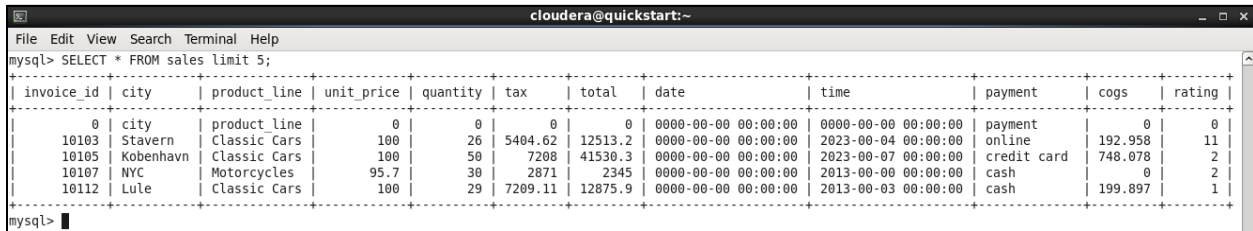
4. Importing the values into the table

Command: LOAD Data Local Infile '/home/cloudera/Desktop/sales.csv' into table sales Fields Terminated By ',' Lines Terminated By '\n' ;

```
cloudera@quickstart:~  
File Edit View Search Terminal Help  
mysql> LOAD Data Local Infile '/home/cloudera/Desktop/sales.csv' into table sales  
Fields Terminated By ',' Lines Terminated By '\n';  
Query OK, 67 rows affected, 27368 warnings (0.22 sec)  
Records: 2824 Deleted: 0 Skipped: 2757 Warnings: 19223  
  
mysql> █
```

5. Checking the values

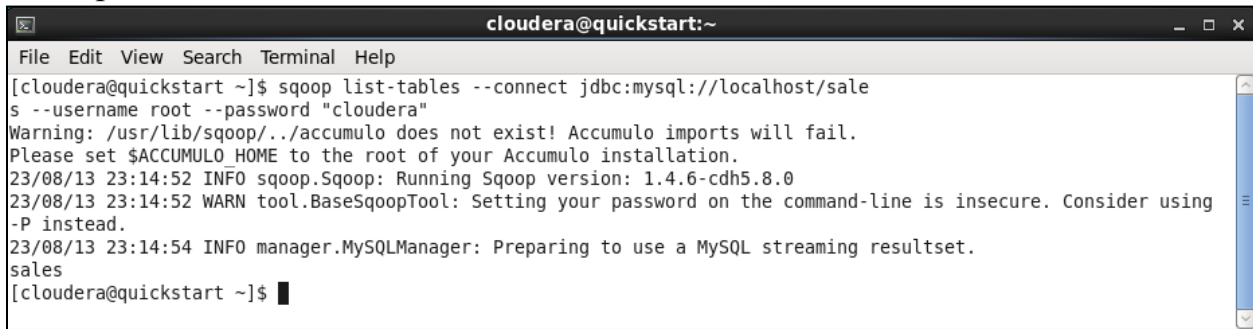
Command: `SELECT * FROM sales;`



invoice_id	city	product_line	unit_price	quantity	tax	total	date	time	payment	cogs	rating
0	city	product_line	0	0	0	0	0000-00-00 00:00:00	0000-00-00 00:00:00	payment	0	0
10103	Stavern	Classic Cars	100	26	5404.62	12513.2	0000-00-00 00:00:00	2023-00-04 00:00:00	online	192.958	11
10105	Kobenhavn	Classic Cars	100	50	7208	41530.3	0000-00-00 00:00:00	2023-00-07 00:00:00	credit card	748.078	2
10107	NYC	Motorcycles	95.7	30	2871	2345	0000-00-00 00:00:00	2013-00-00 00:00:00	cash	0	2
10112	Lule	Classic Cars	100	29	7209.11	12875.9	0000-00-00 00:00:00	2013-00-03 00:00:00	cash	199.897	1

6. Listing the tables present in database sales on MySQL. We use Sqoop to list all the tables present in the MySQL database.

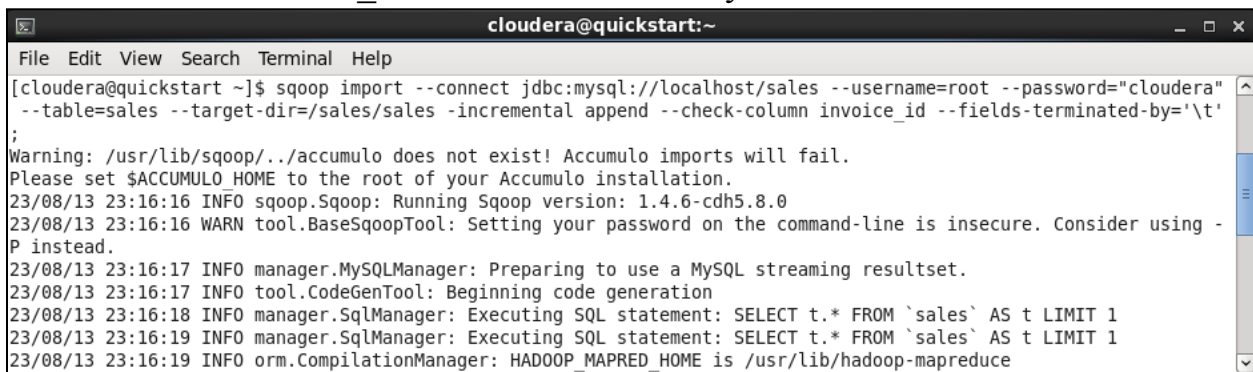
Command: `sqoop list-tables --connect jdbc:mysql://localhost/sales --username root --password "cloudera"`



```
[cloudera@quickstart ~]$ sqoop list-tables --connect jdbc:mysql://localhost/sale
s --username root --password "cloudera"
Warning: /usr/lib/sqoop/./accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
23/08/13 23:14:52 INFO sqoop.Sqoop: Running Sqoop version: 1.4.6-cdh5.8.0
23/08/13 23:14:52 WARN tool.BaseSqoopTool: Setting your password on the command-line is insecure. Consider using
-P instead.
23/08/13 23:14:54 INFO manager.MySQLManager: Preparing to use a MySQL streaming resultset.
sales
[cloudera@quickstart ~]$
```

7. Import tables from rdbms to hdfs using sqoop

Command: `sqoop import --connect jdbc:mysql://localhost/sales --username root --password "cloudera" --table sales --target-dir /sales/sales --incremental append --check-column invoice_id --fields-terminated-by '\t'`



```
[cloudera@quickstart ~]$ sqoop import --connect jdbc:mysql://localhost/sales --username=root --password="cloudera"
--table=sales --target-dir=/sales/sales --incremental append --check-column invoice_id --fields-terminated-by='\t'
;
Warning: /usr/lib/sqoop/./accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
23/08/13 23:16:16 INFO sqoop.Sqoop: Running Sqoop version: 1.4.6-cdh5.8.0
23/08/13 23:16:16 WARN tool.BaseSqoopTool: Setting your password on the command-line is insecure. Consider using -
P instead.
23/08/13 23:16:17 INFO manager.MySQLManager: Preparing to use a MySQL streaming resultset.
23/08/13 23:16:17 INFO tool.CodeGenTool: Beginning code generation
23/08/13 23:16:18 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM `sales` AS t LIMIT 1
23/08/13 23:16:19 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM `sales` AS t LIMIT 1
23/08/13 23:16:19 INFO orm.CompilationManager: HADOOP_MAPRED_HOME is /usr/lib/hadoop-mapreduce
```

8. checking if tables are imported properly

Url: <http://quickstart.cloudera:50070/explorer.html#/sales>

Browse Directory							
/sales							Go!
Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
drwxr-xr-x	cloudera	supergroup	0 B	Sun Aug 13 23:26:54 -0700 2023	0	0 B	sales
drwxr-xr-x	cloudera	supergroup	0 B	Sun Aug 13 23:33:12 -0700 2023	0	0 B	salesvivek

9. Import tables from HDFS to Hive

Command: `sqoop import-all-tables --connect jdbc:mysql://localhost/sales --username=root --password="cloudera" --compression-codec=snappy --as-parquetfile --warehouse-dir=/user/hive/warehouse --hive-import`

```
cloudera@quickstart:~  
File Edit View Search Terminal Help  
[cloudera@quickstart ~]$ sqoop import-all-tables --connect jdbc:mysql://localhost/sales --username=root --password="cloudera" --compression-codec=snappy --as-parquetfile --warehouse-dir=/user/hive/warehouse --hive-import  
Warning: /usr/lib/sqoop/./accumulo does not exist! Accumulo imports will fail.  
Please set $ACCUMULO_HOME to the root of your Accumulo installation.  
23/08/13 23:42:05 INFO sqoop.Sqoop: Running Sqoop version: 1.4.6-cdh5.8.0  
23/08/13 23:48:35 INFO mapreduce.ImportJobBase: Transferred 25.1973 KB in 255.1634 seconds (101.1195 bytes/sec)  
23/08/13 23:48:35 INFO mapreduce.ImportJobBase: Retrieved 67 records.  
[cloudera@quickstart ~]$
```


10. Check if it created in hive

Command: `hadoop fs -ls /user/hive/warehouse`

```
cloudera@quickstart:~  
File Edit View Search Terminal Help  
[cloudera@quickstart ~]$ hadoop fs -ls /user/hive/warehouse  
Found 22 items  
drwxrwxrwx - cloudera supergroup      0 2019-11-05 00:53 /user/hive/warehouse/customers  
drwxrwxrwx - root      supergroup      0 2019-10-25 01:58 /user/hive/warehouse/dept  
drwxrwxrwx - root      supergroup      0 2019-10-25 02:04 /user/hive/warehouse/employee  
drwxrwxrwx - root      supergroup      0 2019-10-25 01:35 /user/hive/warehouse/employee.db  
drwxrwxrwx - root      supergroup      0 2022-09-20 02:17 /user/hive/warehouse/exp6.db  
drwxrwxrwx - root      supergroup      0 2023-08-03 02:08 /user/hive/warehouse/movies.db  
drwxrwxrwx - impala    supergroup      0 2022-09-13 22:36 /user/hive/warehouse/my_db.db  
drwxrwxrwx - root      supergroup      0 2019-10-25 02:00 /user/hive/warehouse/project  
drwxrwxrwx - cloudera  supergroup      0 2019-11-04 03:02 /user/hive/warehouse/project.db  
drwxrwxrwx - root      supergroup      0 2023-08-02 02:30 /user/hive/warehouse/sakshi.db  
drwxrwxrwx - impala    supergroup      0 2023-08-02 02:46 /user/hive/warehouse/sakshi1.db  
drwxrwxrwx - cloudera  supergroup      0 2023-08-13 23:48 /user/hive/warehouse/sales  
drwxrwxrwx - cloudera  supergroup      0 2019-11-05 00:53 /user/hive/warehouse/sample_07  
drwxrwxrwx - cloudera  supergroup      0 2019-11-05 00:53 /user/hive/warehouse/sample_08  
drwxrwxrwx - root      supergroup      0 2023-08-02 02:27 /user/hive/warehouse/std  
drwxrwxrwx - root      supergroup      0 2023-08-02 02:31 /user/hive/warehouse/stdinfo  
drwxrwxrwx - cloudera  supergroup      0 2023-08-08 00:10 /user/hive/warehouse/student  
drwxrwxrwx - cloudera  supergroup      0 2022-09-23 01:17 /user/hive/warehouse/student.db  
drwxrwxrwx - cloudera  supergroup      0 2019-11-03 19:56 /user/hive/warehouse/t1  
drwxrwxrwx - cloudera  supergroup      0 2019-11-03 20:21 /user/hive/warehouse/t2  
drwxrwxrwx - cloudera  supergroup      0 2019-11-03 21:16 /user/hive/warehouse/tab  
drwxrwxrwx - cloudera  supergroup      0 2019-11-05 00:53 /user/hive/warehouse/web_logs  
[cloudera@quickstart ~]$
```

11. Logging into hive

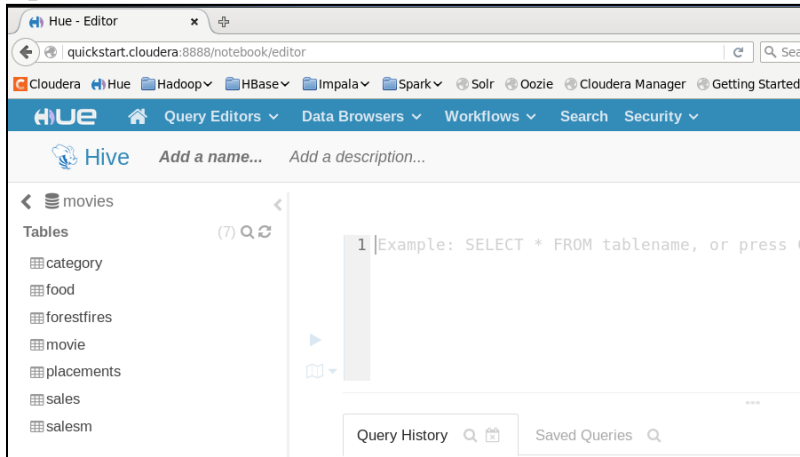
Command: `hive`

Show tables;

```
cloudera@quickstart:~  
File Edit View Search Terminal Help  
[cloudera@quickstart ~]$ hive  
2023-08-13 23:57:18,032 WARN [main] mapreduce.TableMapReduceUtil: The hbase-prefix-tree module j  
ar containing PrefixTreeCodec is not present. Continuing without it.  
  
Logging initialized using configuration in file:/etc/hive/conf.dist/hive-log4j.properties  
WARNING: Hive CLI is deprecated and migration to Beeline is recommended.  
hive> show tables;  
OK  
customers  
dept  
employee  
project  
sales  
sample_07  
sample_08  
std  
stdinfo  
student  
t1  
t2  
tab  
web_logs  
Time taken: 3.222 seconds, Fetched: 14 row(s)  
hive>
```

12.Executing queries

Open the Hue UI in the browser of the cloudera



a. desc sales;

The screenshot shows the Hue Editor interface with the query 'desc sales;' entered in the text input field. The 'Results' tab is selected, displaying the schema of the 'sales' table. The table has four columns: col_name, data_type, and comment. The data is as follows:

	col_name	data_type	comment
1	mid	int	
2	name	varchar(20)	
3	yor	int	
4	sales	int	

Activate Windows

b. Select * from sales;

The screenshot shows the Hue Editor interface with the query 'select * from sales;' entered in the text input field. The 'Results' tab is selected, displaying the data from the 'sales' table. The table has four columns: sales.mid, sales.name, sales.yor, and sales.sales. The data is as follows:

	sales.mid	sales.name	sales.yor	sales.sales
1	1	Team	2015	250000
2	2	Game	2016	200000
3	3	Rock	2016	300000
4	4	Dash	2017	350000

Activate Windows

c. select count(mid), yor from sales GROUP BY yor;



The screenshot shows a SQL query editor with a query and its results. The query is: `1 select count(mid), yor from sales GROUP BY yor;`. The results are displayed in a table with two columns: `_c0` and `yor`. The results are:

	_c0	yor
1	1	2015
2	2	2016
3	1	2017

The interface includes a query history panel, a saved queries panel, and a results panel. The results panel is active, showing the query results. The status bar at the bottom indicates "Activate Windows".

d. SELECT sales.name from sales where sales.sales=200000;



The screenshot shows a SQL query editor with a query and its results. The query is: `1 SELECT sales.name from sales where sales.sales=200000;`. The results are displayed in a table with one column: `sales.name`. The results are:

	sales.name
1	Game

The interface includes a query history panel, a saved queries panel, and a results panel. The results panel is active, showing the query results. The status bar at the bottom indicates "Activate Windows".

e. `SELECT sum(sales.sales) FROM sales;`

The screenshot shows a SQL query execution interface. The query entered is `1 SELECT sum(sales.sales) FROM sales;`. Below the query editor, there are tabs for 'Query History', 'Saved Queries', and 'Results'. The 'Results' tab is active, displaying a table with one column labeled `_c0` and one row with the value `1100000`.

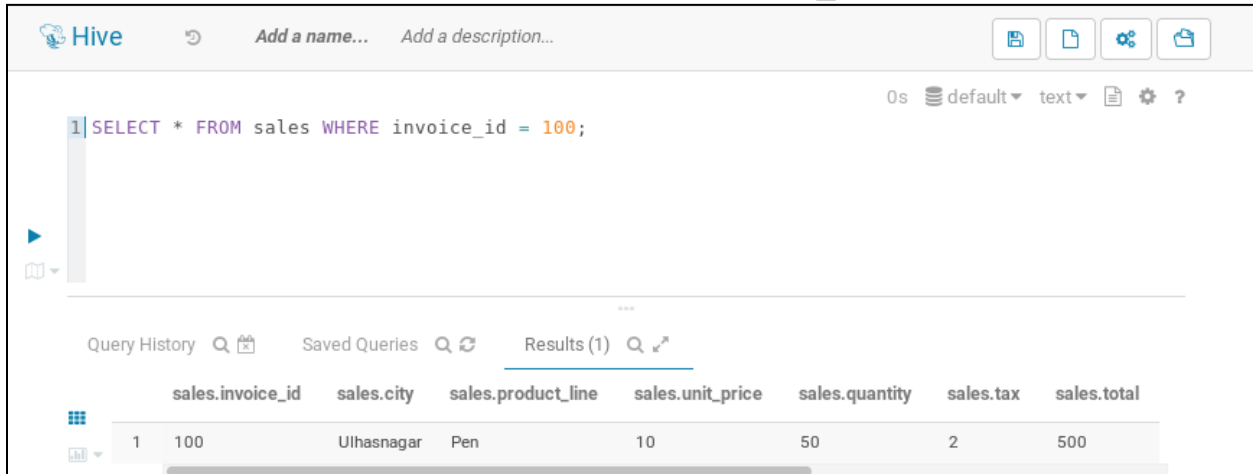
13. Writing back to RDBMS:

S1: `INSERT INTO sales (invoice_id, city, productline, unit_price, quantity, tax, total, date, time, payment, cogs, rating)`
`VALUES (100, 'Ulhasnagar', 'Pen', 10, 50, 2, 500, NULL, NULL, 'online', 3452.123, 8);`

The screenshot shows a Hive query execution interface. The query entered is `1 INSERT INTO sales (invoice_id, city, productline, unit_price, quantity, tax, total, date, time, payment, cogs, rating)`
`2 VALUES (100, 'Ulhasnagar', 'Pen', 10, 50, 2, 500, NULL, NULL, 'online', 3452.123, 8);`
`3`. The query was executed in 11.88s. A success message is displayed: `✓ Success.`. Below the success message, there is a 'Query History' section showing the query was executed 'a few seconds ago'.

S2: Checking whether the tuple is added or not in the hive

Command: `SELECT * FROM sales WHERE invoice_id = 100;`



The screenshot shows the Hive web interface. At the top, there's a header with the Hive logo and options to 'Add a name...' and 'Add a description...'. Below the header, a query is entered in a text area: `1 SELECT * FROM sales WHERE invoice_id = 100;`. To the right of the query area, there are icons for saving, running, and other actions. Below the query area, there's a section for 'Query History', 'Saved Queries', and 'Results (1)'. The 'Results (1)' section displays a table with the following data:

	sales.invoice_id	sales.city	sales.product_line	sales.unit_price	sales.quantity	sales.tax	sales.total
1	100	Ulhasnagar	Pen	10	50	2	500

S3: Checking the data is not in mysql

Command: `SELECT * FROM sales WHERE invoice_id = 100;`

```
mysql> use sales;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> SELECT * FROM sales WHERE invoice_id=100;
Empty set (0.00 sec)
```

```
cloudera@quickstart:~$ sqoop export --connect jdbc:mysql://localhost/sales --
username=root --password="cloudera" --table=sales --hcatalog-table=sales --hcat
tag-database default --m 4
Warning: /usr/lib/sqoop/./accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
23/08/13 23:33:56 INFO sqoop.Sqoop: Running Sqoop version: 1.4.6-cdh5.13.0
23/08/13 23:33:57 WARN tool.BaseSqoopTool: Setting your password on the comman
d-line is insecure. Consider using -P instead.
23/08/13 23:33:57 INFO manager.MySQLManager: Preparing to use a MySQL streaming
resultset.
23/08/13 23:33:57 INFO tool.CodeGenTool: Beginning code generation
23/08/13 23:33:57 ERROR manager.SqlManager: Error executing statement: java.sql
.SQLException: Access denied for user 'root'@'localhost' (using password: YES)
java.sql.SQLException: Access denied for user 'root'@'localhost' (using password
: YES)
    at com.mysql.jdbc.SQLException.createSQLException(SQLException.java:996)
    at com.mysql.jdbc.MySQLIO.checkErrorPacket(MySQLIO.java:3887)
    at com.mysql.jdbc.MySQLIO.checkErrorPacket(MySQLIO.java:3823)
    at com.mysql.jdbc.MySQLIO.checkErrorPacket(MySQLIO.java:870)
    at com.mysql.jdbc.MySQLIO.secureAuth411(MySQLIO.java:4332)
    at com.mysql.jdbc.MySQLIO.doHandshake(MySQLIO.java:1258)
    at com.mysql.jdbc.ConnectionImpl.coreConnect(ConnectionImpl.java:2234)
    at com.mysql.jdbc.ConnectionImpl.connectOneTryOnly(ConnectionImpl.java:2
385)
    at com.mysql.jdbc.ConnectionImpl.createNewIO(ConnectionImpl.java:2064)
    at com.mysql.jdbc.ConnectionImpl.<init>(ConnectionImpl.java:790)
    at com.mysql.jdbc.JDBC4Connection.<init>(JDBC4Connection.java:44)
    at sun.reflect.NativeConstructorAccessorImpl.newInstance0(Native Method)
    at sun.reflect.NativeConstructorAccessorImpl.newInstance(NativeConstruct
orAccessorImpl.java:57)
    at sun.reflect.DelegatingConstructorAccessorImpl.newInstance(DelegatingC
onstructorAccessorImpl.java:45)
    at java.lang.reflect.Constructor.newInstance(Constructor.java:526)
cloudera@quickstart:~$ mysql -u root -p
mysql> SELECT * FROM sales WHERE order_id=100;
ERROR 1054 (42S22): Unknown column 'order_id' in 'where clause'
mysql> SELECT * FROM sales WHERE invoice_id=100;
Empty set (0.00 sec)

mysql> SELECT * FROM sales WHERE invoice_id=10105;
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| invoice_id | city      | product_line | unit_price | quantity | tax    | total    | date              | time              | payment | cogs    | rating |
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 10105      | Kobenhavn | Classic Cars | 100        | 50       | 7200   | 41530.3  | 2023-08-07 00:00:00 | 2023-08-07 00:00:00 | credit card | 748.878 | 2      |
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

cloudera@quickstart:~$
```