Aims

This exercise aims to get you to:

- Install and configure Hive, and Manage data using Hive

Hive Installation and Configuration

1. Download Hive 3.1.3

```
$ wget https://dlcdn.apache.org/hive/hive-3.1.3/apache-hive-3.1.3-bin.tar.gz
```

Then unpack the package:

```
$ tar xvf apache-hive-3.1.3-bin.tar.gz
```

2. Define environment variables for Hive

We need to configure the working directory of Hive, i.e., `HIVE_HOME`.

Open the file `~/.bashrc` and add the following lines at the **end** of this file:

```
export HIVE_HOME=/home/comp9313/apache-hive-3.1.3-bin
export PATH=$HIVE_HOME/bin:$PATH
```

Save the file, and then run the following command to take these configurations into effect:

(Hive 3 only supports Java 8!)

Add the line “export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64” in the files hadoop/etc/hadoop/hadoop-env.sh and ~/.bashrc.

```
$ source ~/.bashrc
```

3. Create /tmp and /user/hive/warehouse and set them for more than one user usage

```
$ hdfs dfs -mkdir /tmp
$ hdfs dfs -mkdir -p /user/hive/warehouse
$ hdfs dfs -chmod g+w /tmp
$ hdfs dfs -chmod g+w /user/hive/warehouse
```

4. Run the schematool command to initialize Hive

```
$ schematool -dbType derby -initSchema
```

Now you have already done the basic configuration of Hive, and it is ready to use. Start Hive shell by the following command (start HDFS first!):

```
$ hive
```

```
comp9313@comp9313-VirtualBox:~$ hive
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/comp9313/hive/lib/log4j-slf4j-impl-2.4.1
.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/comp9313/hadoop/share/hadoop/common/lib/
slf4j-log4j12-1.7.10.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]

Logging initialized using configuration in jar:file:/home/comp9313/hive/lib/hive
-common-2.1.0.jar!/hive-log4j2.properties Async: true
Hive-on-MR is deprecated in Hive 2 and may not be available in the future versio
ns. Consider using a different execution engine (i.e. tez, spark) or using Hive
1.X releases.
hive>
```

Manage Data Using Hive

1. Download the test file “employees.txt” from the course webpage. The file contains only 7 records. Put the file in the home folder.

2. Create a database

```
$ hive> create database employee_data;
$ hive> use employee_data;
```

3. All databases are created under /user/hive/warehouse directory (open a new terminal window).

```
$ hdfs dfs -ls /user/hive/warehouse
```

```
comp9313@comp9313-VirtualBox:~$ hdfs dfs -ls /user/hive/warehouse
Found 2 items
drwxr-xr-x   - comp9313 supergroup          0 2016-09-05 08:08 /user/hive/wareho
use/employee_data.db
```

4. Create the employee table

```
$ hive> CREATE TABLE employees (
  name         STRING,
  salary       FLOAT,
  subordinates ARRAY<STRING>,
  deductions   MAP<STRING, FLOAT>,
  address      STRUCT<street:STRING, city:STRING, state:STRING, zip:INT>
)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY '\001'
COLLECTION ITEMS TERMINATED BY '\002'
MAP KEYS TERMINATED BY '\003'
LINES TERMINATED BY '\n'
STORED AS TEXTFILE;
```

Because '\001', '\002', '\003', and '\n' are by default, and thus you can ignore “ROW FORMAT DELIMITED”. “STORED AS TEXTFILE” is also by default, and can be ignored as well.

5. Show all tables in the current database

```
$ hive> show tables;
```

```
hive> show tables;
OK
employees
Time taken: 0.01 seconds, Fetched: 1 row(s)
```

6. Load data from the local file system into the table

(employees.txt can be download from https://webcms3.cse.unsw.edu.au/COMP9313/23T3/resources/92027)

```
$ hive> LOAD DATA LOCAL INPATH '/home/comp9313/employees.txt' OVERWRITE INTO TABLE employees;
```

```
hive>  LOAD DATA LOCAL INPATH '/home/comp9313/employees.txt' OVERWRITE INTO TABL
E employees;
Loading data to table employee_data.employees
OK
Time taken: 0.564 seconds
```

After loading the data into the table, you can check in HDFS what happened:

```
$ hdfs dfs -ls /user/hive/warehouse/employee_data.db/employees
```

The file employees.txt is copied into this folder corresponding to the table.

7. Check the data in the table

```
$ select * from employees;
```

8. You can do various queries based on the employees table, just as in an RDBMS. For example:

Question 1: show the number of employees and their average salary

Hint: use count() and avg()

Question 2: find the employee who has the highest salary

Hint: use max(), IN clause, and subquery in where clause

9. Usage of explode(). Find all employees who are the subordinates of another person. explode() takes in an array (or a map) as an input and outputs the elements of the array (map) as separate rows.

```
$ hive> SELECT explode(subordinates) FROM employees;
```

```
hive> select explode(subordinates) from employees;
OK
Mary Smith
Todd Jones
Bill King
John Doe
Fred Finance
Stacy Accountant
Time taken: 0.08 seconds, Fetched: 6 row(s)
```

10. Hive partitions. When defining employees, it is not partitioned, and thus you cannot add a partition to it. You can only add a new partition to a table that has already been partitioned!

Create a table employees2, and load the same file into it.

```
$ hive> CREATE TABLE employees2 (
  name         STRING,
  salary       FLOAT,
  subordinates ARRAY<STRING>,
  deductions   MAP<STRING, FLOAT>,
  address      STRUCT<street:STRING, city:STRING, state:STRING, zip:INT>
)PARTITIONED BY (join_year STRING);
$ hive> LOAD DATA LOCAL INPATH '/home/comp9313/employees.txt' OVERWRITE INTO TABLE
employees2 PARTITION (join_year="2015");
```

Now check HDFS again to see what happened:

```
$ hdfs dfs –ls /user/hive/warehouse/employ_data.db/employees2
```

You will see a folder "join_year=2015" created in this folder, corresponding to the partition join_year= "2015".

Add a new partition join_year="2016" to the table.

```
$ hive> ALTER TABLE employees2 ADD PARTITION (join_year='2016') LOCATION
'/user/hive/warehouse/employee_data.db/employees2/join_year=2016';
```

Check HDFS and you will see a new folder created for this partition.

11. Insert a record to partition join_year="2016".

Because Hive does not support literals for complex types (array, map, struct, union), so it is not possible to use them in INSERT INTO...VALUES clauses. You need to create a file to store the new record, and then load it into the partition.

```
$ cp employees.txt employees2016.txt
```

Then use vim or gedit to edit employees2016.txt to add some records, and then load the file into the partition.

12. Query on a partition. Question: find all employees who joined in the year 2016 whose salary is more than 60000.