

```
// Lab Execution steps //
```

1. Data loading in MySQL - Open a terminal window and execute the below commands.

1.1 Copy the data file to MySQL docker container.

```
# docker cp /home/labuser/Project/Dataset/Walmart_Store_sales.csv  
ra_mysql:Walmart_Store_sales.csv
```

```
# docker exec -i -t ra_mysql bash
```

1.2 Launch a mysql shell.

```
# mysql -p  
# password: example
```

```
# SET GLOBAL local_infile=1;  
# quit;
```

1.3 Relaunch the mysql shell with following command.

```
# mysql --local-infile=1 -p
```

```
# CREATE DATABASE if not exists retail;  
# use retail;
```

```
# CREATE TABLE if not exists walmart_sales (Store VARCHAR(255),Date  
Date,Weekly_Sales VARCHAR(255),Holiday_Flag VARCHAR(255),Temperature  
VARCHAR(255),Fuel_Price VARCHAR(255),CPI VARCHAR(255),Unemployment  
VARCHAR(255));
```

```
# show tables;
```

```
# LOAD DATA LOCAL INFILE '/Walmart_Store_sales.csv' INTO TABLE walmart_sales  
FIELDS TERMINATED BY ',' LINES TERMINATED BY '\n' IGNORE 1 ROWS  
(Store,@Date,Weekly_Sales,Holiday_Flag,Temperature,Fuel_Price,CPI,Unemployment)  
SET Date=STR_TO_DATE( @Date, '%d-%m-%Y' );
```

2. Sqoop import job creation for ingesting data - Open a new terminal window and execute the below commands.

2.1 Copy the JAR file for reading data from JSON file by using following command.

```
# docker cp /home/labuser/Project/Code/java-json-schema.jar  
ra_sqoop:/opt/sqoop-1.4.7.bin__hadoop-2.6.0/lib/java-json-schema.jar
```

```
# docker exec -i -t ra_sqoop bash
```

## 2.2 Create a sqoop job data ingestion.

```
# sqoop job --create retail_job -- import --connect  
jdbc:mysql://mysql:3306/retail --username root --password example --table  
walmart_sales --fields-terminated-by "," --hive-import --hive-table raw_sales  
--hive-overwrite --m 1 --incremental append --check-column Date --last-value  
'1900-01-01'
```

## 2.3 Execute the sqoop job.

```
# sqoop job --exec retail_job  
# password: example
```

## 3. Login into the Hive docker container

```
# docker exec -i -t hdp_hive-server bash
```

### 3.1 Verify the data ingested using SQOOP job.

```
# hive  
# show tables;  
# select * from raw_sales;  
# exit;
```

## 4. Incremental load in MySQL table - Execute below commands in new terminal window.

```
# docker exec -i -t ra_mysql bash
```

```
# mysql -p  
# password: example
```

```
# insert into walmart_sales values(45,'2012-11-  
03','760281.43','0','58.85','3.882','192.3088989','8.667');
```

## 5. Data re-loading for analysis.

```
# docker exec -i -t ra_sqoop bash
```

```
# sqoop import --connect jdbc:mysql://mysql:3306/retail --username root --  
password example --table walmart_sales --fields-terminated-by "," --hive-
```

```
import --hive-table raw_sales --hive-overwrite --m 1 --incremental append --
check-column Date --last-value '1900-01-01'
```

## 6. Switch to Hive docker container for Data Analysis.

```
# beeline
# !connect jdbc:hive2://127.0.0.1:10000 scott tiger
# create database if not exists retail;
# use retail;

# create table walmart_sales as
select
  cast(Store as int),
  cast(to_date(from_unixtime(unix_timestamp(`Date`, 'yyyy-MM-dd'))) as date)
as date_of_entry,
  cast(Weekly_Sales as double),
  Holiday_Flag,
  cast(Temperature as float),
  cast(fuel_price as float),
  cast(cpi as double),
  cast(Unemployment as float)
from default.raw_sales;
```

## 7. Refer to Project videos for further execution.