**Load data into HDFS using SQOOP**

**a. Load retail data from foodmart database into MySQL.**

#creating database in mysql and loading data from .sql file located at ~/Desktop/foodmart\_mysql.sql

mysql> CREATE table foodmart;

mysql> use foodmart;

mysql> source ~/foodmart\_mysql.sql

**b. Extract following columns from Promotion table in Food mart DB using SQOOP.**

PromotionID, Promotion Name, Promotion Cost

sqoop import --connect jdbc:mysql://quickstart.cloudera/foodmart --username root --password cloudera --query SELECT promotion\_id,promotion\_name,cost from promotion WHERE $CONDITIONS' --target-dir /user/cloudera/promotion\_tb --m 1 --split-by promotion\_id

**c. Extract following columns from Sales\_Fact\_1997 & Sales\_Fact\_1998 tables in Food mart DB using SQOOP.**

Region\_id, join with store(Region\_id)

ProductID, in sales tb

StoreID, in sales tb

PromotionID, in sales tb

Store Sales, in sales tb

the\_day (day of week), time\_by\_day(time\_id) join sales\_fact\_1998(time\_id)

the\_month, time\_by\_day(time\_id) join sales\_fact\_1998(time\_id)

the\_year time\_by\_day(time\_id) join sales\_fact\_1998(time\_id)

*Sqoop Query*

sqoop import --connect jdbc:mysql://quickstart.cloudera/foodmart --username root --password cloudera --query 'select st.region\_id, sa.product\_id, sa.store\_id, sa.promotion\_id, sa.store\_sales, td.the\_day, td.the\_month, td.the\_year FROM store st INNER JOIN sales\_fact\_1998 sa ON st.store\_id = sa.store\_id INNER JOIN time\_by\_day td ON td.time\_id = sa.time\_id UNION SELECT st.region\_id, sa.product\_id, sa.store\_id, sa.promotion\_id, sa.store\_sales, td.the\_day, td.the\_month, td.the\_year FROM store st inner join sales\_fact\_1997 sa ON st.store\_id = sa.store\_id inner JOIN time\_by\_day td on td.time\_id = sa.time\_id WHERE $CONDITIONS' --target-dir /user/cloudera/sales\_fact\_tb --m 3 --split-by product\_id

***Running Mapreduce JAR***

*Source-dir:* /user/cloudera/sales\_fact\_tb

Dest-dir: /user/cloudera/joinedTablesOutput

Cache table: /user/cloudera/promotion\_tb/part-m-00000

hadoop jar case\_study.jar /user/cloudera/sales\_fact\_tb /user/cloudera/joinedTablesOutput /user/cloudera/promotion\_tb/part-m-00000

**Hive**

**a.)**Creating hive external table

hive> create EXTERNAL table prom\_sales(

> sales\_year int,sales\_month string,region\_id int,prom\_id int,prom\_name string,

> prom\_cost double,WD\_sales double,WE\_sales double)

> row format delimited

> fields terminated by ","

> stored as textfile;

Loading data into table

hive> load data inpath '/user/cloudera/joinedTablesOutput' into table prom\_sales;

**b.)**Create partitioned table

hive> create table parti\_prom\_sales(prom\_id int,prom\_name string,prom\_cost double,

>wd\_sales double,we\_sales double)

>PARTITIONED BY(region\_id int,sales\_year int,sales\_month string);

hive> set hive.exec.max.dynamic.partitions=100000;

hive> hive.exec.max.dynamic.partitions.pernode=100000;

hive> insert overwrite table parti\_prom\_sales PARTITION(region\_id,sales\_year,sales\_month) select prom\_id,prom\_name,prom\_cost,wd\_sales,we\_sales,region\_id,sales\_year,sales\_month from prom\_sales;

**c.Spark**

**(i)** Creating first table in hive

scala> sqlContext.sql("create table sparkdb.sparkFromHive1(region\_id int,prom\_id int,maxPromCost double,sum\_wdsales double,sum\_wesales double)")

Saving query results into table

scala> val query1df = sqlContext.sql("insert into sparkdb.sparkFromHive1 select region\_id,prom\_id,max(prom\_cost) as maxPromCost,sum(wd\_sales) as sum\_wdsales,sum(we\_sales) as sum\_wesales from prom\_sales group by prom\_id,region\_id")

**(ii)**Creating second table in hive

scala> sqlContext.sql("create table sparkdb.sparkFromHive2(region\_id int,prom\_id int,sum\_week double)")

val query2df = sqlContext.sql("insert into sparkdb.sparkFromHive2 select region\_id,prom\_id,max(prom\_cost) as maxPromCost,sum(wd\_sales) + sum(we\_sales) as sum\_week from prom\_sales group by prom\_id,region\_id")