**1) Solution**

val data = sc.parallelize(List(("Rajesh",21,"London"),

("Suresh",28,"California"),

("Sam",26,"Delhi"),

("Rajesh",21,"Gurgaon"),

("Manish",29,"Bengaluru")).toDF("Name","Age","Location")

data.dropDuplicates(Array("Name","Age")).show()

**2)Solution**

LOAD DATA INPATH '//filepath' INTO TABLE test.data\_clickstream;

val hiveContext = new org.apache.spark.sql.hive.HiveContext(sc)

val dataDF= hiveContext.sql("select \* from test.data\_clickstream")

dataDF.write.parquet("/path/to/file")

**3) Solution**

Table Name: data\_final\_clickstream

Columns:sessionid,day,month,year, userid,timestampstream

Table create structure:

CREATE TABLE test.data\_final\_clickstream( userid INT, sessionid VARCHAR,timestampstream timestamp,day INT, month INT) PARTITIONED BY (year INT)

ROW FORMAT DELIMITED

FIELDS TERMINATED BY ‘|’

Stored as parquet;

LOAD DATA INPATH '/path/to/file' INTO TABLE test.data\_final\_clickstream;

**In Spark SCALA**

val sqlContext = new org.apache.spark.sql.hive.HiveContext(sc)

sqlContext.sql("CREATE TABLE test.data\_final\_clickstream( userid INT, sessionid VARCHAR,timestampstream timestamp,day INT, month INT) PARTITIONED BY (year INT)

ROW FORMAT DELIMITED

FIELDS TERMINATED BY ‘|’

Stored as parquet")

sqlContext.sql("LOAD DATA INPATH '/path/to/file' INTO TABLE test.data\_final\_clickstream")