**Test Framework – 1**

**Schema Comparison**

Comare schema of source and target tables.

val testData = spark.sqlContext.sql("Select \* from swamy.employee")

val prodData = spark.sqlContext.sql("Select \* from swamy.employee")

if(testData.schema == prodData.schema)

{

print(“Schema Comparison Passed”)

}

**Row Count Check**

val testCount = testData.count()

val prodCount = testData.count()

if (testCount == prodCount){

print(“Row Count Test Passed”)

}

**Data Comparision**

var resultData = testData.except(prodData)

resultData.unionAll(prodData.except(testData))

if(resultData.count()>0){

print(“There is a mismatch in Data”)

}

**Test Framework - 2**

**Absolute Duplicate Check**

val df = spark.sqlContext.sql("Select \* from swamy.employee")

val cols = df.columns.toSeq

df.groupBy(cols map col: \_\*).count().filter("count>1").show()

**Duplicate Check:**

Create a list with all columns and append to list

val inputColumns = "name,eid,salary"

val cols = inputColumns.split(",").map(\_.trim)

val df2 = df.groupBy(cols map col: \_\*).count().filter("count>1")

**Null Check**

for(x <-cols)

{

If(df.count == df.where(df.col(x).isNull).count())

{

println(x+" is null")

}else{

println(x+" is not null")

}

}

df.where(df.col("friend\_id").isNull)

for(x <- cols)

{

if(df.count == df.filter((col(x)==="") || (col(x)==="null") || (col(x)===null)).count())

{

println(x+" is null")

}else{

println(x+" is not null")

}

}

**Configuration File**

**Test Framework-1**

source {

isDirectHive=true

driverName=as

databaseUrl=as

databaseUserName=as

databasePassword=as

query=select \* from default.categories

isFile=false

fileName=as

fileWithSchema=false

}

target {

isDirectHive=true

driverName=as

databaseUrl=as

databaseUserName=as

databasePassword=as

query=select \* from default.categories

isFile=false

fileName=as

fileWithSchema=false

}

test {

schemaCompare=true

rowCountCompare=true

dataCompare=true

dataCompareFileName=/user/cloudera/testframework/duplicate

}

log {

fileName=/home/cloudera/test.log

}

**Test Framework-2**

input {

isDirectHive=true/false

driverName

databaseUrl=

databaseUserName=

databasePassword=

query=

isFile=

fileName=

fileWithSchema=true/false

numPartitions=

}

test{

absoluteDuplicateCheck=true/false

absoluteDuplicateCheckFileName=fileNamePath

duplicateCheck=true/false

duplicateCheckColumns= list of columns

duplicateCheckFileName=fileNamePath

nullCheck=true/false

}

log {

fileName=/home/cloudera/test.log

}