val health = sqlContext.read.format("com.databricks.spark.csv").option("header", "true").option("inferSchema", "true").load("/hdfs/localport:9000/healthdata/health.csv")  
  
health.registerTempTable("healthData")  
  
  
//Get Values for Perticular condition  
val HA = sqlContext.sql("select \* from healthData where Condition ='Heart Attack or Chest Pain'")  
HA.show()

HA.repartition(1).write.format("com.databricks.spark.csv")  
   .option("header", "true")  
   .save("/hdfs/localport:9000/healthdata/HA.csv")  
  
  
//Count Number of diseases have been reported  
val diseasesReported = sqlContext.sql("select Condition,count(Condition) countVlaues from healthData group by `Condition`")  
  
diseasesReported.repartition(1).write.format("com.databricks.spark.csv")  
   .option("header", "true")  
   .save("/HA.csv")  
  
//number diseases have been reported city wise  //if required you can filter based in state,zipe code  
val city = sqlContext.sql("select City,Condition,count(City,Condition) countVlaues from healthData group by `City`,`Condition`")  
  
city.repartition(1).write.format("com.databricks.spark.csv")  
   .option("header", "true")  
   .save("/hdfs/localport:9000/healthdata/city.csv")  
  
  
  
//empty foot note  
val footnote = sqlContext.sql("select \* from healthData where Footnote != null")  
footnote.repartition(1).write.format("com.databricks.spark.csv")  
   .option("header", "true")  
   .save("/hdfs/localport:9000/healthdata/footnote.csv")