clean_retaildataset.py

```
import xlrd
import csv
import glob
import pandas as pd

#Appends year as a last columns
def Append_Year_To_CSV(inputFileName, outputFileName):
	print "Append_Year_To_CSV Called..."
	sYear = Read_Cell_Specific_Cell(inputFileName)
	with open(inputFileName,'r') as csvinput:
		with open(outputFileName +'.csv', 'w') as csvoutput:
			writer = csv.writer(csvoutput, lineterminator='\n')
			reader = csv.reader(csvinput)
			all = []
			row = next(reader)
			row.append('Year')
			all.append(row)
			for row in reader:
				row.append(sYear)
				all.append(row)
			writer.writerows(all)

#Reading header for Year
def Read_Cell_Specific_Cell(inputFileName):
	fileObj = open(inputFileName,'r')
	reader = csv.reader(fileObj)
	headers = reader.next()
	return headers[2][5:]


def List_All_CSV_Files_From_Folder(filePath, fileType):
	allFiles = glob.glob(filePath + fileType)
	list_ = []
	for file_ in allFiles:
		#Call Append_Year_To_CSV method
		Append_Year_To_CSV(file_, file_)

def Merge_All_CSV_files(inputFilePath, fileType, outputFileName):
	print 'Merging files started..'
	bFlag = False
	csvFiles = glob.glob(inputFilePath + fileType)
	dataFrameList = []
	for fileName in csvFiles:
		print "Merging : " + fileName
		dataFrame = pd.read_csv(fileName, header=None)
		if bFlag == False:
			dataFrameList.append(dataFrame[1:])
		else:
			dataFrameList.append(dataFrame[2:])
	concatDataFrame = pd.concat(dataFrameList, axis=0)
	#concatDataFrame = concatDataFrame.replace(',','')
```

```
      #Spliting the columns for cleaning column data
      list_id = concatDataFrame[0]
      #Replacing ',' present in the list
      list_id = [myid.replace(',', '') for myid in list_id]
      list_Particulars = concatDataFrame[1]
      #Replacing ',' present in the list
      list_Particulars = [myid.replace(',', '') for myid in list_Particulars]
      list_Total= concatDataFrame[14].fillna(0).replace('(NA)',0)
      list_Year = concatDataFrame[15].fillna(0).replace('(NA)',0)
      #Create DataFrame with required columns only
      CleanedDataFrame = pd.DataFrame({'ItemId':list_id, 'Particulars':
list_Particulars, 'Total': list_Total, 'Year': list_Year})
      #Create the csv file using required columns
      #print 'outputFileName : ' + outputFileName
      CleanedDataFrame.to_csv(outputFileName, index=None)

def Process_File():
      print 'Started Processing files..'
      filePath =r'/home/Documents/Project/Data' # use your path
      fileType = '/*.csv.*'
      List_All_CSV_Files_From_Folder(filePath, fileType)
      #Call merging of csv files
      fileType = '/*.csv'
      Merge_All_CSV_files(filePath, fileType, filePath +
'/cleaned/Merged_Data.csv')


Process_File()
```

Step1: Create_Scripts

Create all required tables cloudlab Hive Environment

```
create database if not exists jay_retail_proj;
use jay_retail_proj;
create table if not exists tran_generic(
itemid int,
description string,
total double,
year int)
row format delimited
fields terminated by ',';
create table if not exists tran_2000(
itemid int,
description string,
total double,
year int)
row format delimited
fields terminated by ',';
create table if not exists tran_2015(
itemid int,
description string,
total double,
year int)
row format delimited
fields terminated by ',';
```

```
create table if not exists tran_2005(
itemid int,
description string,
total double,
year int)
row format delimited
fields terminated by ',';create table if not exists tran_2006(
itemid int,
description string,
total double,
year int)
row format delimited
fields terminated by ',';
create table if not exists tran_2013(
itemid int,
description string,
total double,
year int)
row format delimited
fields terminated by ',';
show tables;
```

Step2: Generic_Load_Data

```
use jay_retail_proj;
load data inpath '/user/joy.tat_gmail/retail/Merged_Data.csv' overwrite into table
tran_generic;
```

Step3: Split_Data_to_Tables

This steps loads data from tran_generic table to respective tables.

```
use jay_retail_proj;

INSERT OVERWRITE table tran_2015 select itemid, description,total, year  FROM
tran_generic
where year = 2015;

INSERT OVERWRITE table tran_2000 select itemid, description,total, year  FROM
tran_generic
where year = 2000;

INSERT OVERWRITE table tran_2005 select itemid, description,total, year  FROM
tran_generic
where year = 2005;

INSERT OVERWRITE table tran_2006 select itemid, description,total, year  FROM
tran_generic
where year = 2006;

INSERT OVERWRITE table tran_2013 select itemid, description,total, year  FROM
tran_generic
where year = 2013;
```

Retail project source code by Jayateertha Tatti

Analysis 1:

```
use jay_retail_proj;

SELECT tran_2015.itemid, tran_2015.description,
(((tran_2015.total - tran_2000.total)/ tran_2000.total)*100)/15 as percentage_incr
FROM tran_2015
LEFT JOIN tran_2000 on tran_2015.itemid = tran_2000.itemid
WHERE ((((tran_2015.total - tran_2000.total)/ tran_2000.total)*100)/15) >= 10;
```

Analysis_2:

```
use jay_retail_proj;

SELECT tran_2015.itemid, tran_2015.description,
(((tran_2015.total - tran_2000.total)/ tran_2000.total)*100)/15 as
percentage_decrease
FROM tran_2015
LEFT JOIN tran_2000 ON tran_2000.description = tran_2015.description
WHERE ((((tran_2015.total - tran_2000.total)/ tran_2000.total)*100)/15) <= -5;
```

Analysis 3_1:

```
use jay_retail_proj;

SELECT tran_2005.itemid, tran_2005.description,
(((tran_2005.total - tran_2000.total)/ tran_2000.total)*100)/5 as percentage_incr
FROM tran_2005
LEFT JOIN tran_2000 on tran_2000.itemid = tran_2005.itemid
Where  ((((tran_2005.total - tran_2000.total)/ tran_2000.total)*100)/5) >= 10;
```

Analysis 3_2:

```
use jay_retail_proj;

SELECT tran_2013.itemid, tran_2013.description,
max((((tran_2013.total - tran_2006.total)/ tran_2006.total)*100)/8) as
percentage_descrease
FROM tran_2013
JOIN tran_2006 ON tran_2013.itemid = tran_2006.itemid
Group by tran_2013.itemid, tran_2013.description
HAVING percentage_descrease <=-2;
```