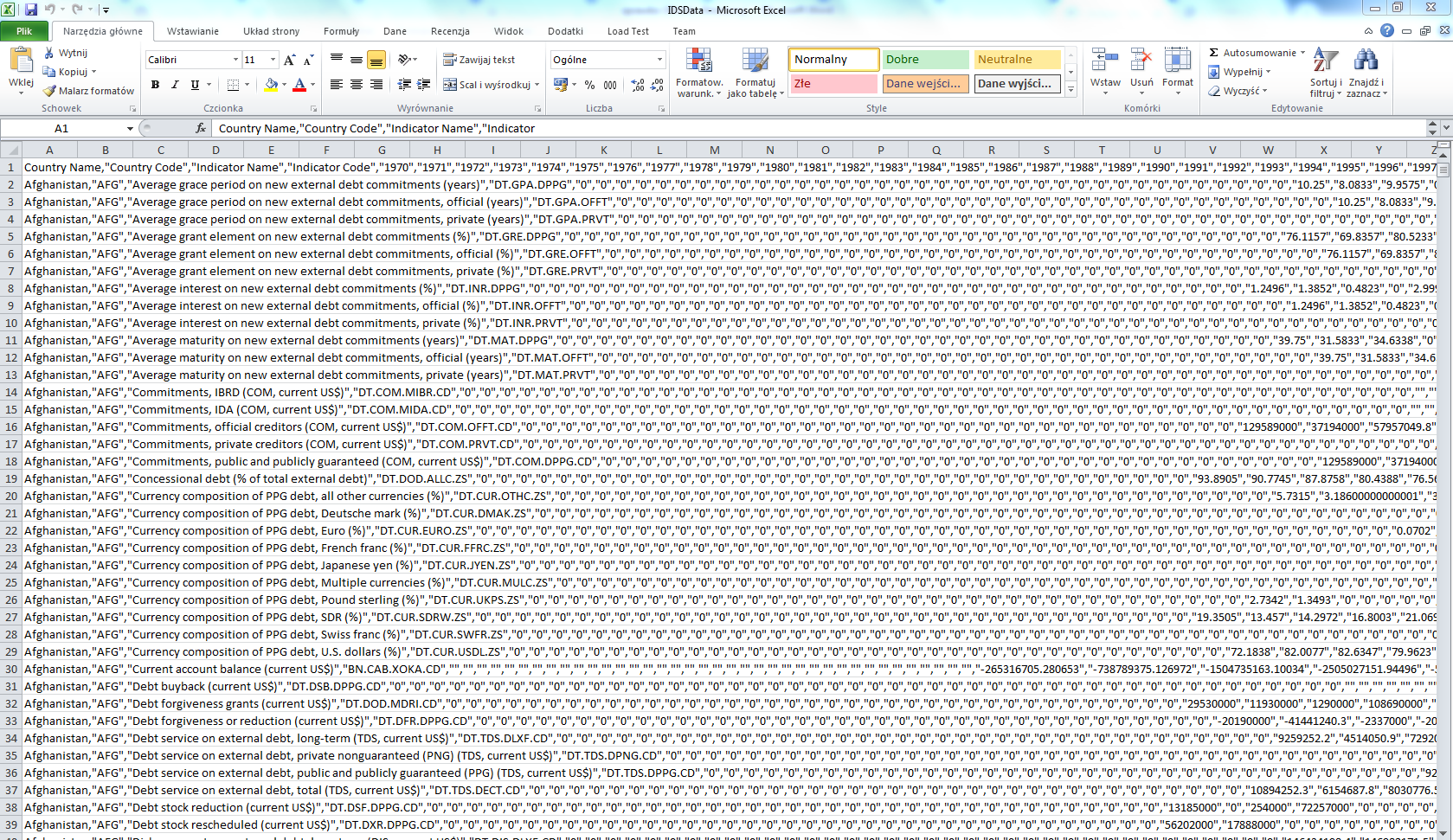
WSTEP

**PRZYGOTOWANIE DO PRZETWARZANIA W HIVE**



Rys. Dane przed prztworzeniem

import csv

old\_file = 'IDSData.csv'

new\_file = 'IDS/data.csv'

i = 0

add\_to\_buff = [

"Average grace period on new external debt commitments (years)",

"Commitments official creditors (COM current US$)",

r"Concessional debt (% of total external debt)",

"Debt stock reduction (current US$)",

"External debt stocks total (DOD current US$)",

"Interest payments on external debt total (INT current US$)",

r"Short-term debt (% of total external debt)",

"Technical cooperation grants (current US$)",

]

content = []

# otwieramy plik z ktorego bierzemy dane

with open(old\_file, 'r', newline='') as csv\_file\_read:

reader = csv.reader(csv\_file\_read)

buff\_row = []

next(reader)

for row in reader:

buff\_row = []

for i in range(0, 2):

buff\_row.append(row[i])

string = row[2].replace(',',' ')

string = string.replace(" "," ")

if string in add\_to\_buff:

buff\_row.append(string)

else:

continue

buff\_row.append(row[3])

for i in range(40, 50):

if row[i]:

buff\_row.append(row[i])

else:

buff\_row.append(0)

content.append(buff\_row)

# dane zapisujemy do nowej csvki

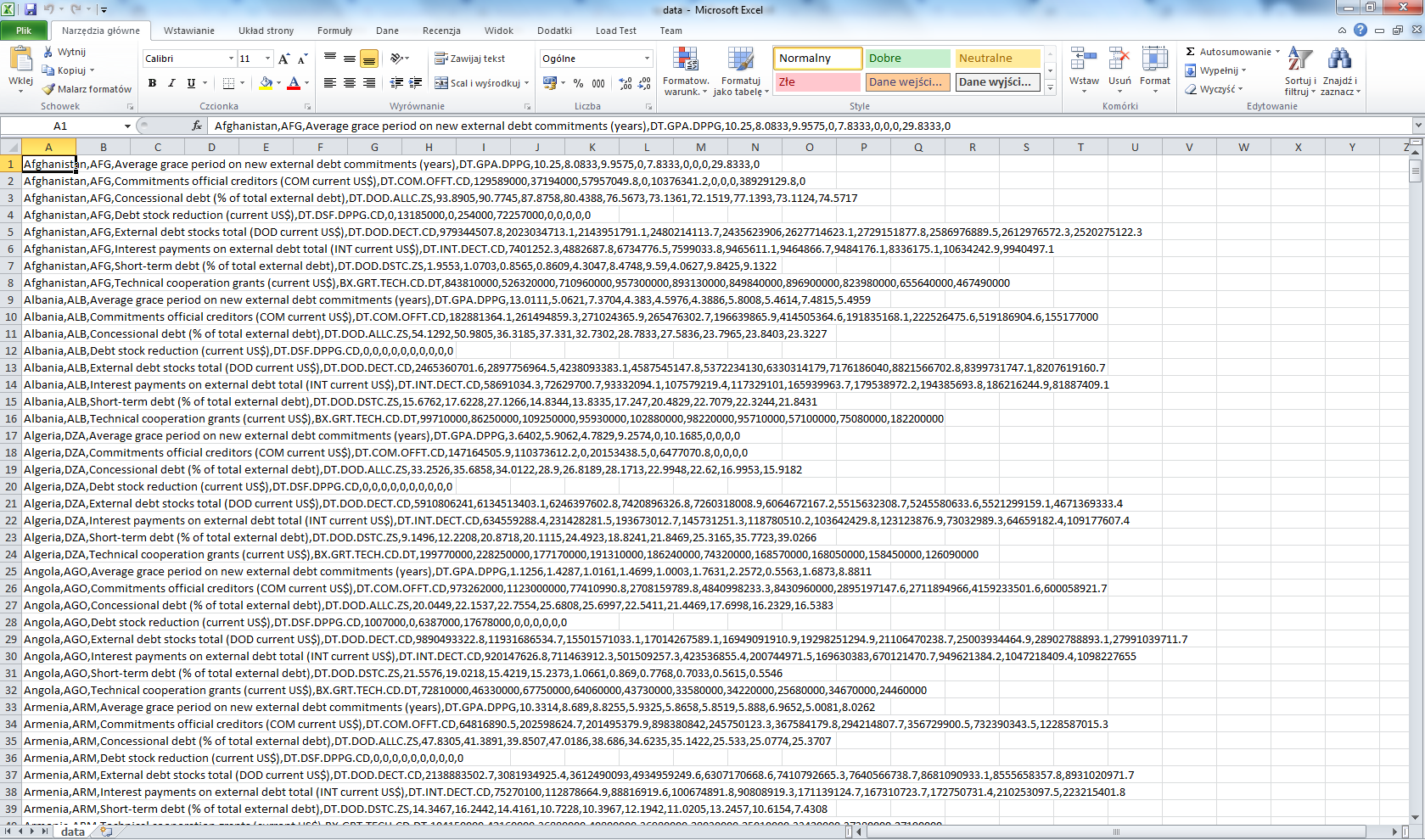
with open(new\_file, 'w', newline='') as csv\_file\_write:

writer = csv.writer(csv\_file\_write)

for row in content:

writer.writerow(row)

Listing Skryptu przetwarzającego dane



Rys. Po przetworzeniu

**Przetwarzanie w hivie**

Stworzyliśmy tabele w

CREATE TABLE ids\_data (CountryName string, CountryCode string, IndicatorName string, IndicatorCode string,

y2006 double, y2007 double, y2008 double, y2009 double, y2010 double, y2011 double,

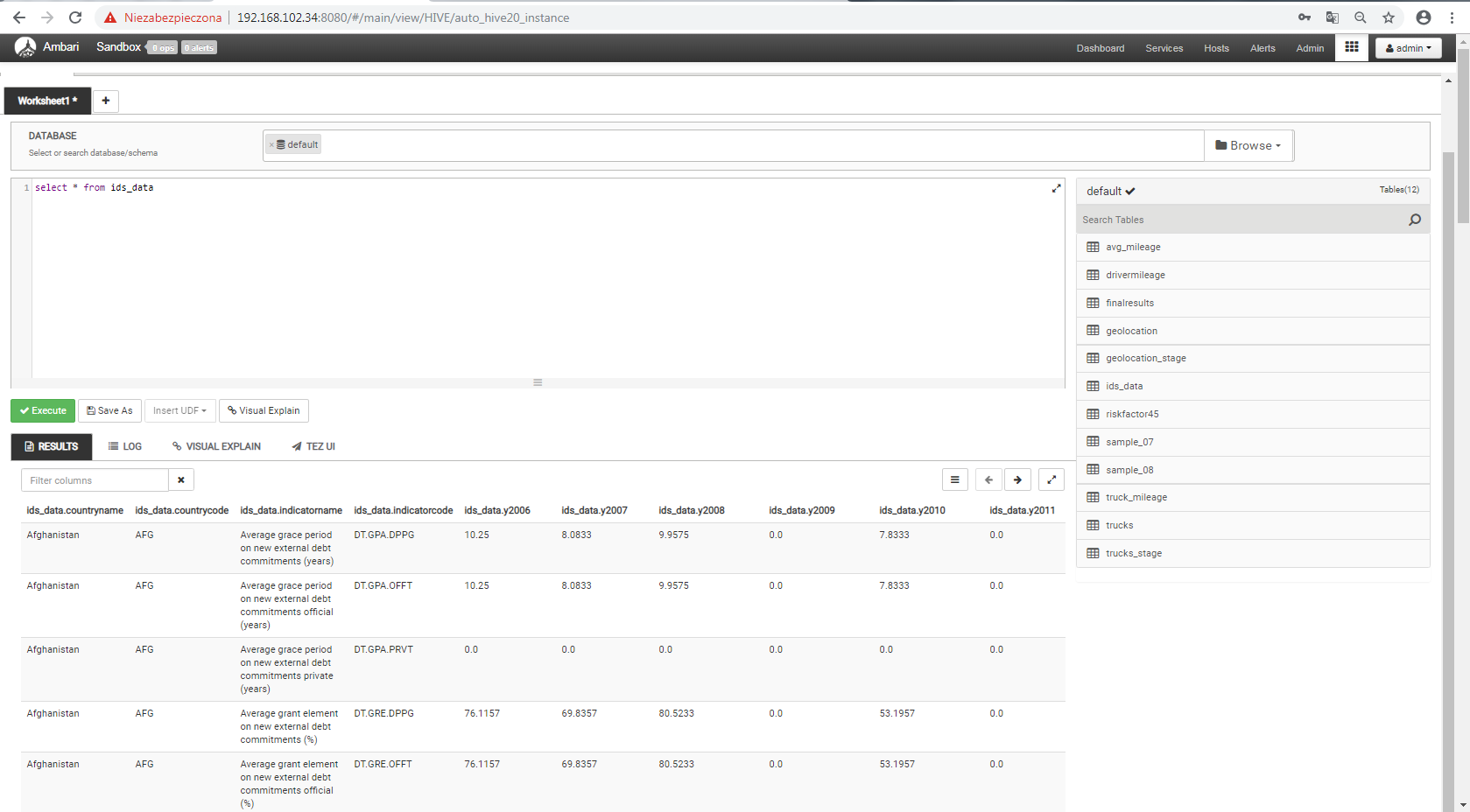
y2012 double, y2013 double, y2014 double, y2015 double)

row format delimited fields terminated by ',';

listing tworzenia tabeli

Uploadowalismy plik przez na stworzony na rysunku do katalogu/apps/hive/warehouse/ids\_data

Ids\_data.csv



Select wykonany po wstwieniu danych do tabeli ids\_data

**Przetwarzanie w Sparku**

Uruchamiamy piga komendą pig -useHCatalog

a = LOAD ‘ids\_data’ using org.apache.hive.hcatalog.pig.HCatLoader();

commitments = FILTER a BY indicatorname == ‘Commitments official creditors (COM current US$)’;

grace\_period = FILTER a BY indicatorname == ‘Average grace period on new external debt commitments (years)’;

concessional\_debt = FILTER a BY indicatorname == ‘Concessional debt (% of total external debt)’;

debt\_stock\_reduciton = FILTER a BY indicatorname == ‘Debt stock reduction (current US$)’;

external\_debt\_total = FILTER a BY indicatorname == ’External debt stocks total (DOD current US$)’;

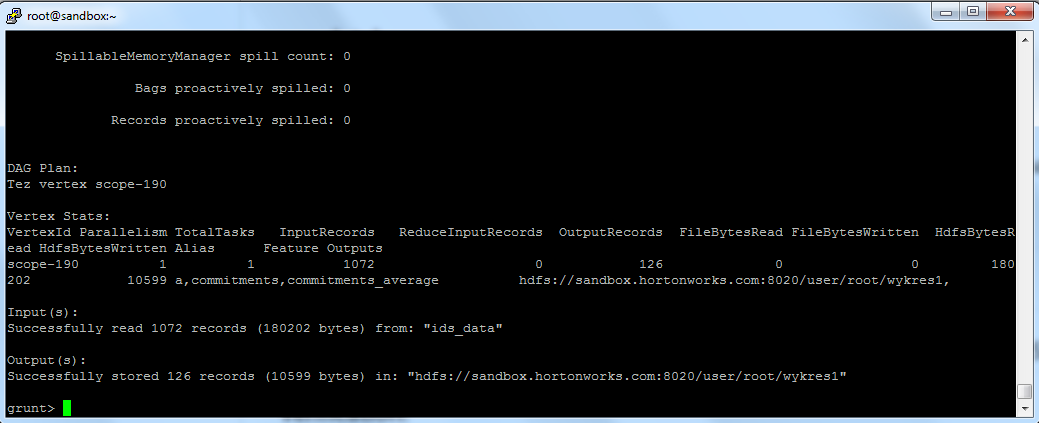
interest\_payments = FILTER a BY indicatorname == ‘Interest payments on external debt total (INT current US$)’;

short\_term\_debt = FILTER a BY indicatorname == ‘Short-term debt (% of total external debt)’;

commitments\_average = FOREACH commitments GENERATE (countryname, countrycode, indicatorname, ((y2006+y2007+y2008+y2009+y2010+y2011+y2012+y2013+y2014+y2015)/10) );

STORE commitments\_average INTO ‘wykres1’

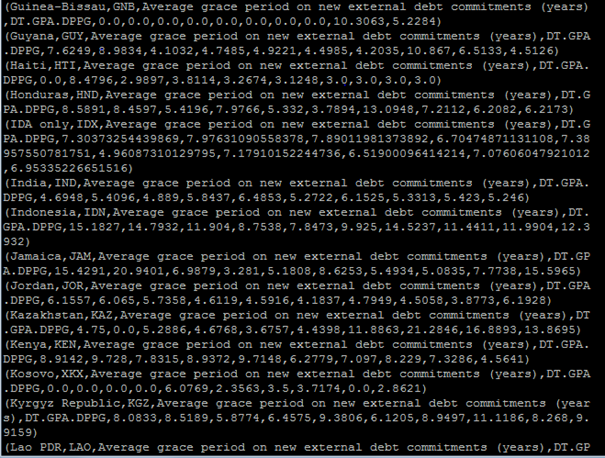
listing skryptu napisanego w pig latin



Rys. Komunikat po zapisaniu zmiennej pig

CREATE TABLE ids\_commitments(CountryName string, CountryCode string, IndicatorName string, srednia\_commitments double)

row format delimited fields terminated by ',');

Rys. Dump wszystkich danych

**PRZETWARZANIE W SPARKU**

Komendy:

%jdbc(hive) show tables

%spark2

val hiveContext = new org.apache.spark.sql.SparkSession.Builder().getOrCreate()

%spark2

val ids\_commitments = hiveContext.sql("select \* from ids\_commitments")

%spark2

ids\_commitments.createOrReplaceTempView("ids\_commitments")

hiveContext.sql("show tables").show()

Listing spark skrypty wszystkie:

