Big Data Hadoop—Real Time Project—Retail

Jayateertha M Tatti

joy.tat@gmail.com

24-Oct-2016

Analyze monthly retail trade report for the US market

- A US-based online retailer wants to launch a new product category and wants to understand the potential growth areas and areas that have stagnated over a period of time.
- It wants to use this information to ensure its product focus is aligned to opportunities that will grow over the next 5–7years.
- The customer has also provided pointers to the data set you can use.

Project goals:

1) Analyze the entire data set and arrive at products that have experienced a consolidated **yearly growth of 10% or more** in sales since 2000.
2) Analyze the entire data set and arrive at products that have experienced a consolidated **yearly drop of 5% or less** since 2000.
3) Arrive at products that have experienced a **growth of 10% or more in sales** from 2000 to 2005, and then subsequently experienced a **drop of at least 2% in sales** from 2006 to 2013.

Data set in xls format from the URL :
http://www.census.gov/retail/index.html

Hadoop Architecture

- Vmware
- Linux Ubuntu 16.04 Lts
- Hadoop 2.7.2

```
admin@poorvi-HP-Pavilion-dv4-Notebook-PC:~$ hadoop version
Hadoop 2.7.2
Subversion https://git-wip-us.apache.org/repos/asf/hadoop.git -r b165c4fe8a74265c792ce23f546c64604acf0e41
Compiled by jenkins on 2016-01-26T00:08Z
Compiled with protoc 2.5.0
From source with checksum d0fda26633fa762bff87ec759ebe689c
This command was run using /home/admin/hadoop/share/hadoop/common/hadoop-common-2.7.2.jar
admin@poorvi-HP-Pavilion-dv4-Notebook-PC:~$ jps
4040 NodeManager
4410 Jps
3579 DataNode
3917 ResourceManager
3759 SecondaryNameNode
admin@poorvi-HP-Pavilion-dv4-Notebook-PC:~$
```

Approach

- Referred lesson 3 to load the dataset in HDFS
- Referred lesson 7 Hive to arrive at the below results detailed in following pages
-

Download dataset

Get the "Retail and food services sales" dataset from "Monthly retail trade report " tab

Refer image below

```
admin@poorvi-HP-Pavilion-dv4-Notebook-PC:~/Documents/SimpliLearn/Python$ ls -l
total 848
drwxrwxr-x 4 admin admin   4096 Oct 18 13:12 Books
drwxrwxr-x 3 admin admin   4096 Oct 14 14:40 Download
-rw-rw-r-- 1 admin admin 421376 Oct 12 21:58 mrtssales92-present.xls
drwxrwxr-x 5 admin admin   4096 Oct 24 12:07 Project
-rw-rw-r-- 1 admin admin   3260 Oct 14 14:33 Read_SheetName_Back.py
-rw-rw-r-- 1 admin admin   3094 Oct 20 15:23 Read_SheetName.py
-rw-rw-r-- 1 admin admin 421376 Oct 12 21:58 TestData.xls
-rw-rw-r-- 1 admin admin   3443 Oct 14 12:47 Test.py
admin@poorvi-HP-Pavilion-dv4-Notebook-PC:~/Documents/SimpliLearn/Python$
```

Conversion xls to csv files

For conversion linux gnumeric was used

Note: I have renamed the mrtssales92-present.xls to Test_Data.xls

ssconvert -S Test_Data.xls Test_Data.csv

ssconvert created one csv per sheet. Hence 0 to 23 csv files were created for each year 1992 to 2015.

Refer the image below

```
admin@poorvi-HP-Pavilion-dv4-Notebook-PC:~/Documents/SimpliLearn/Python/Project/Data$ ls
cleaned               Test_Data.csv.12      Test_Data.csv.16.csv  Test_Data.csv.20      Test_Data.csv.24.csv  Test_Data.csv.6.csv
output                Test_Data.csv.12.csv  Test_Data.csv.17      Test_Data.csv.20.csv  Test_Data.csv.2.csv   Test_Data.csv.7
Test_Data.csv.0       Test_Data.csv.13      Test_Data.csv.17.csv  Test_Data.csv.21      Test_Data.csv.3       Test_Data.csv.7.csv
Test_Data.csv.0.csv   Test_Data.csv.13.csv  Test_Data.csv.18      Test_Data.csv.21.csv  Test_Data.csv.3.csv   Test_Data.csv.8
Test_Data.csv.1       Test_Data.csv.14      Test_Data.csv.18.csv  Test_Data.csv.22      Test_Data.csv.4       Test_Data.csv.8.csv
Test_Data.csv.10      Test_Data.csv.14.csv  Test_Data.csv.19      Test_Data.csv.22.csv  Test_Data.csv.4.csv   Test_Data.csv.9
Test_Data.csv.10.csv  Test_Data.csv.15      Test_Data.csv.19.csv  Test_Data.csv.23      Test_Data.csv.5       Test_Data.csv.9.csv
Test_Data.csv.11      Test_Data.csv.15.csv  Test_Data.csv.1.csv   Test_Data.csv.23.csv  Test_Data.csv.5.csv   Test_Data.xls
Test_Data.csv.11.csv  Test_Data.csv.16      Test_Data.csv.2       Test_Data.csv.24      Test_Data.csv.6       Working
admin@poorvi-HP-Pavilion-dv4-Notebook-PC:~/Documents/SimpliLearn/Python/Project/Data$
```

Cleaning csv files

Using sed command csv files were cleaned.

Listed all the commands for reference

```
sed -i '74,150d' *.csv.*
sed -i '56,58d' *.csv.*
sed -i '28,28d' *.csv.*
sed -i '6,11d' *.csv.*
sed -i '1,4d' *.csv.*
```

Additional step performed:

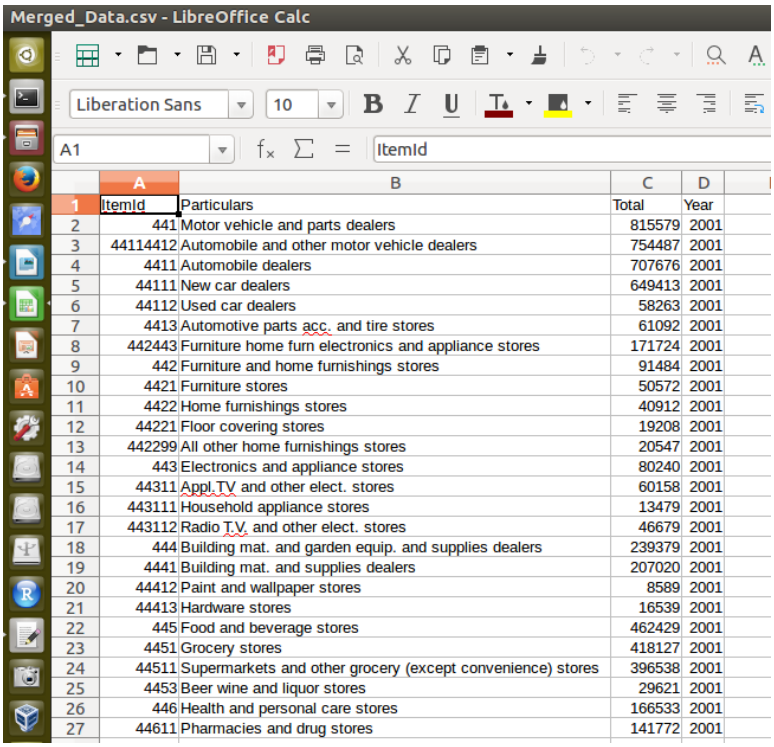
I was experience too many timeouts while loading the data to hive. Hence decided to merge the all the required files into one csv with year as new column.

- Upload the merged data csv file to cloudlab
- Create a generic table tran_generic

- Load the data to generic table
- load respective year data to specific tables to meet the project requirement.

These above steps helped me to speed up the processing. And also provided me an opportunity to enhance my python skills.

Sample Merged Data



	A	B	C	D
1	ItemId	Particulars	Total	Year
2	441	Motor vehicle and parts dealers	815579	2001
3	44114412	Automobile and other motor vehicle dealers	754487	2001
4	4411	Automobile dealers	707676	2001
5	44111	New car dealers	649413	2001
6	44112	Used car dealers	58263	2001
7	4413	Automotive parts acc. and tire stores	61092	2001
8	442443	Furniture home furn electronics and appliance stores	171724	2001
9	442	Furniture and home furnishings stores	91484	2001
10	4421	Furniture stores	50572	2001
11	4422	Home furnishings stores	40912	2001
12	44221	Floor covering stores	19208	2001
13	442299	All other home furnishings stores	20547	2001
14	443	Electronics and appliance stores	80240	2001
15	44311	Appl.TV and other elect. stores	60158	2001
16	443111	Household appliance stores	13479	2001
17	443112	Radio T.V. and other elect. stores	46679	2001
18	444	Building mat. and garden equip. and supplies dealers	239379	2001
19	4441	Building mat. and supplies dealers	207020	2001
20	44412	Paint and wallpaper stores	8589	2001
21	44413	Hardware stores	16539	2001
22	445	Food and beverage stores	462429	2001
23	4451	Grocery stores	418127	2001
24	44511	Supermarkets and other grocery (except convenience) stores	396538	2001
25	4453	Beer wine and liquor stores	29621	2001
26	446	Health and personal care stores	166533	2001
27	44611	Pharmacies and drug stores	141772	2001

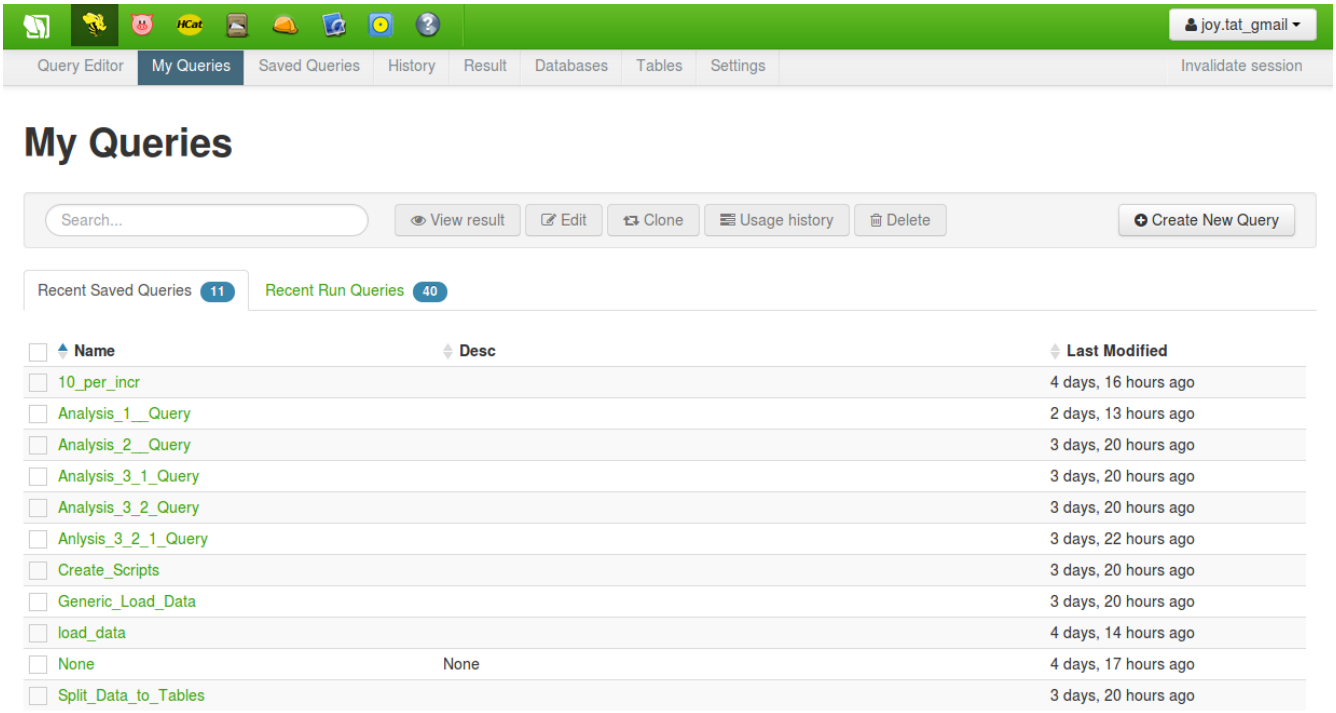
To merge the files Read_SheetName.py was created

```
SimpliLearn/Python/Project/Data$ python ../../Read_SheetName.py
Started Processing files..
Append_Year_To_CSV Called...
Merging files started..
Merged_Data.csv generated
```

Upload this file to couldlab retril_proj folder.

CloudLab steps followed:

Created queries saved them in My Queries section.
Please refer the below image



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;
```

Query Editor : Create_Scripts

```
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 ',';
```

Execute | Save | Save as... | Explain | or create a | New query

Step2: Generic_Load_Data

use jay_retail_proj;

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

Query Editor : 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.

Query Editor : Split_Data_to_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;
```

Analysis 1:

- Analyze the entire data set and arrive at products that have experienced a consolidated yearly growth of 10% or more in sales since 2000
- We have data for period 2000 to 2015
- Input the data in HDFS
- use Hive to arrive at the conclusions
- The total growth % for 15 years will be
    - ((Total_2015-Total_200)/Total_2000)*100 as percentage_decrease
  - and yearly growth will be
    - percentage_decrease / 15
- We want to find yearly growth over 10% or more in sales since year 2000

Query Editor : Analysis_1__Query

```
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;
```

Query Results: Analysis_1__Query

DOWNLOADS

Download as CSV

Download as XLSX

Save

Did you know? If the result contains a large number of columns, click a row to select a

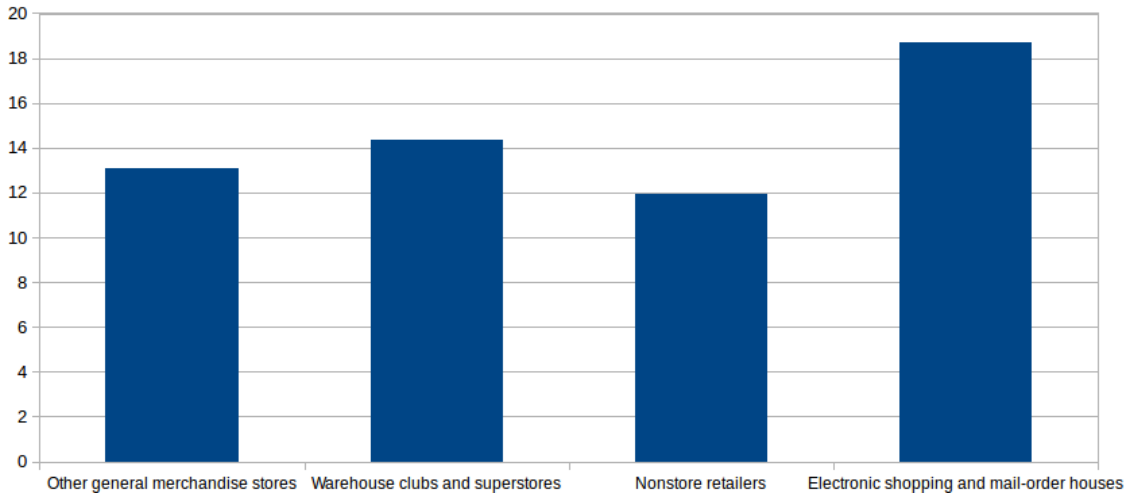
Results | Query | Log | Columns

	tran_2015.itemid	tran_2015.description	percentage_incr
0	4529	Other general merchandise stores	13.0882138885
1	45291	Warehouse clubs and superstores	14.35042331
2	454	Nonstore retailers	11.9239912886
3	4541	Electronic shopping and mail-order houses	18.6865864019

Conclusion Analysis 1:

tran_2015.description	percentage_incr
Other general merchandise stores	13.0882138885
Warehouse clubs and superstores	14.35042331
Nonstore retailers	11.9239912886
Electronic shopping and mail-order houses	18.6865864019

Chart Analysis 1



Analysis 2:

- Analyze the entire data set and arrive at products that have experienced a consolidated yearly drop of 5% or less since 2000
- We have data for period 2000 to 2015
- Input the data in HDFS
- use Hive to arrive at the conclusions
- The total growth % for 15 years will be
    - ((Total_2015-Total_200)/Total_2000)*100 as precentage_decrease
  - and yearly growth will be
    - precentage_decrease / 5
- We want to find yearly drop of 5% or less in sales since year 2000

Query Editor : Analysis_2__Query

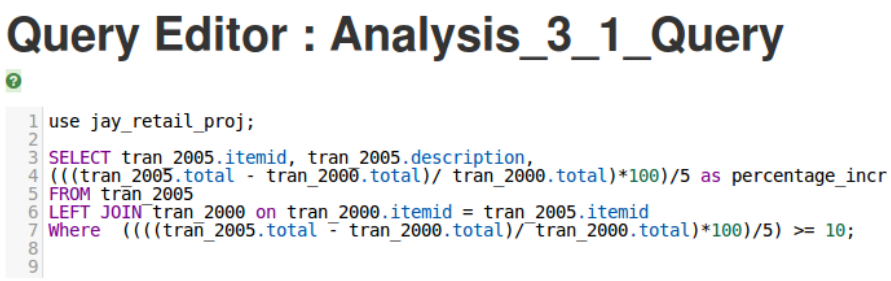
```
use jay_retail_proj;

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;


```

Conclusion Analysis 2:

No products have experienced a consolidated yearly drop of 5% since year 2000

Analysis 3.1:

- Arrive at products that have experienced a growth of 10% or more in sales from 2000 to 2005
- We have data for period 2000 to 2005
- use Hive to arrive at the conclusions
- The total growth % for 5 years will be
    - ((Total_2005-Total_200)/Total_2000)*100 as percentage_incr
  - and yearly growth will be
    - percentage_incr / 5
- We want to find yearly growth of 5% or less in sales since year 2000



Conclusion Analysis 3.1:

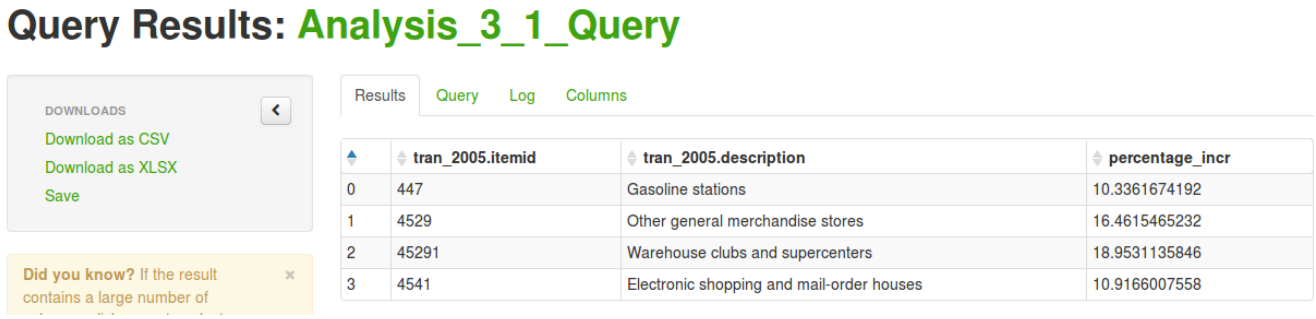
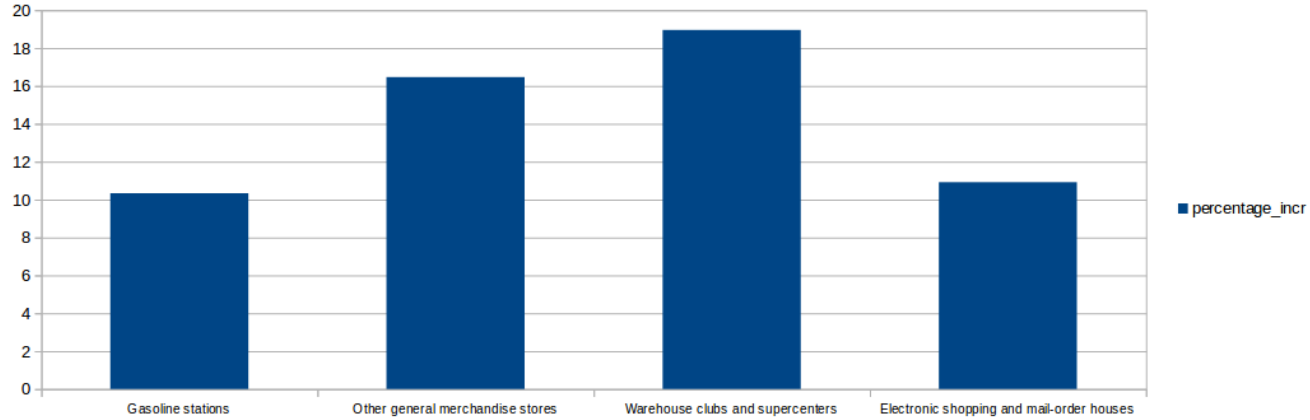


Chart Analysis 3.1:



Analysis 3.2:

- Arrive at products that have experienced a drop of at least 2% in sales from 2006 to 2013
- We have data for period 2006 to 2013
- use Hive to arrive at the conclusions
- The total drop % for 5 years will be
    - ((Total_2013-Total_2006)/Total_2006)*100 as percentage_decrease
  - and yearly drop will be
    - percentage_decrease / 8
- We want to find yearly drop of 2% or less in sales since year 2006 to 2013

Query Editor : Anlysis_3_2_1_Query

```
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;

```

Conclusion Analysis 3.2:

Query Results: Anlysis_3_2_1_Query

DOWNLOADS

Download as CSV

Download as XLSX

Save

Did you know? If the result contains a large number of columns, click a row to select a column to jump to. As you type into the field, a drop-down list displays column names that match the string.

Results | Query | Log | Columns

	tran_2013.itemid	tran_2013.description	percentage_descrease
0	4521	Department stores (excl.L.D)	-2.51935211466
1	4532	Office supplies stationery and gift stores	-2.50877936546
2	44221	Floor covering stores(2)	-3.45566640753
3	45321	Office supplies and stationery stores	-3.37074337371
4	451211	Book stores	-4.03242431382
5	452111	Department stores(excl. discount department stores)	-3.06378195948
6	452112	Discount dept. stores	-2.19030844962

Chart Analysis 3.2: