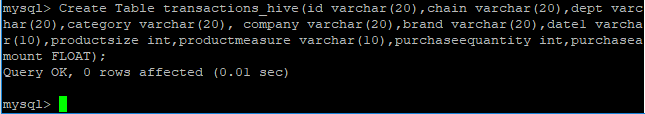
**Project 1**

**Retail Data Analysis with Apache Pig**

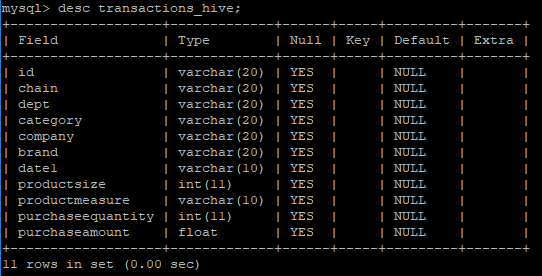
* Retail stores daily generate millions of transactions logs.
* Analyzing these logs would generate beautiful insights and improve business.
* Storing these logs on traditional databases would be costly and scalability will be a big challenge
* Stores like walmart are spread across different locations.
* Daily millions of customers visit these stores and generate billions of logs.
* This billions of logs contribute to huge volume of data.
* Having Huge Volume, High velocity and variety will make this data Big Data.
* Challenges:
* Storage
* Scalability
* Processing
* Sharing
* What is our Goal
  + We will process data and find following.
    - Demand of a given product
    - Trend and seasonality of sales
    - Understand performance of chain
    - Loyal Customer identification
* We have a Transactional data which we will process using pig
* First, we have to load data into mysql

Step 1: Create table in mysql

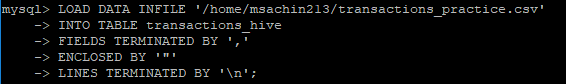
Note we have already loaded data into mysql in previous project so we will use same dataset for this project



Here We have created Table transactions\_hive



Step 2: Load Data into table

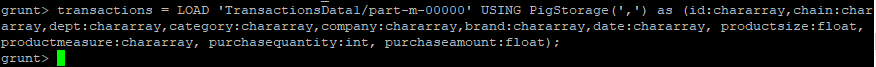


Step 3: Now Use Apache Sqoop for import dataset

sqoop import --connect jdbc:mysql://101.53.130.146/msachin213 --username msachin213 -P --table transactions\_hive -m 1 --target-dir TransactionData1

**This is the command for importing data from mysql to hdfs using sqoop tool**

Step 3: Load Data into Pig



transactions = LOAD 'TransactionsData1/part-m-00000' USING PigStorage(',') as (id:chararray,chain:chararray,dept:chararray,category:chararray,company:chararray,brand:chararray,date:chararray, productsize:float, productmeasure:chararray, purchasequantity:int, purchaseamount:float);

* **Top 10 Customers**
* This is query for getting top 10 customers

custGroup = GROUP transactions BY id; --grouping

custSpendings = FOREACH custGroup GENERATE group, SUM(transactions.purchaseamount) as spendings; --sum operation

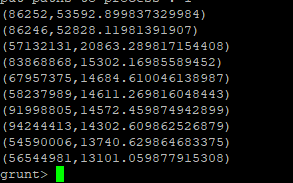
custSpendingsSort = ORDER custSpendings BY spendings desc;

top10Cust = LIMIT custSpendingsSort 10;

DUMP top10Cust;

STORE top10Cust INTO ‘top10Cust’

* Result



* **Chain wise sales**
* This is query for getting Chain wise sales

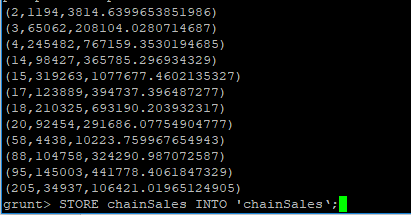
**chainGroup = GROUP transactions BY chain;**

**chainSales = FOREACH chainGroup GENERATE group, SUM(transactions.purchasequantity) as totalQuantity, SUM(transactions.purchaseamount) as totalSales;**

**dump chainSales;**

**STORE chainSales INTO 'chainSales‘;**

* Result



* **Each chain, top 10 customers**
* **Query**

chainGro-upCust = GROUP transactions BY (chain,id);

chainGroupCustSpedings1 = FOREACH chainGroupCust GENERATE group, SUM(transactions.purchaseamount) as spendings;

chainGroupCustSpendings2= FOREACH chainGroupCustSpedings1 generate group.chain as chain,group.id as id, spendings;

chainGroupCustSpendings3= GROUP chainGroupCustSpendings2 BY chain;

chainTop10Cust = FOREACH chainGroupCustSpendings3{ chainGroupCustSpedingsSort = ORDER chainGroupCustSpendings2 BY spendings DESC; top10Cust = LIMIT chainGroupCustSpedingsSort 10; GENERATE top10Cust; }

chainTop10Cust = FOREACH chainTop10Cust GENERATE FLATTEN(top10Cust);

STORE chainTop10Cust INTO ‘chainTop10Cust ’

* **Each customer most bought brand**
* **Query**

CustBrandGroup = GROUP transactions BY (id,brand);

CustBrandQuantity = FOREACH CustBrandGroup GENERATE group, SUM(transactions.purchasequantity) as sales;

CustBrandQuantity = FOREACH CustBrandQuantity GENERATE group.brand as brand, group.id as id, sales;

CustBrandQuantityGroup = GROUP CustBrandQuantity BY brand;

custTop5Brands = FOREACH CustBrandQuantityGroup{ CustBrandQuantityGroupSort = ORDER CustBrandQuantity BY sales DESC; top5Brand = LIMIT CustBrandQuantityGroupSort 5; GENERATE top5Brand;}

custTop5Brands = FOREACH custTop5Brands GENERATE FLATTEN(top5Brand);

STORE custTop5Brands INTO 'custTopFiveBrands';

* **Top 10 brands**
* **Query**

brandGroup = GROUP transactions BY brand;

brandPurchase = FOREACH brandGroup GENERATE group, SUM(transactions.purchaseamount) as purchase;

brandPurchaseSort = ORDER brandPurchase BY purchase desc;

top10Brands = LIMIT brandPurchaseSort 10;

STORE top10Brands INTO 'top10Brands';