Zeppelin

sqlContext.sql("show tables").toDF().show()

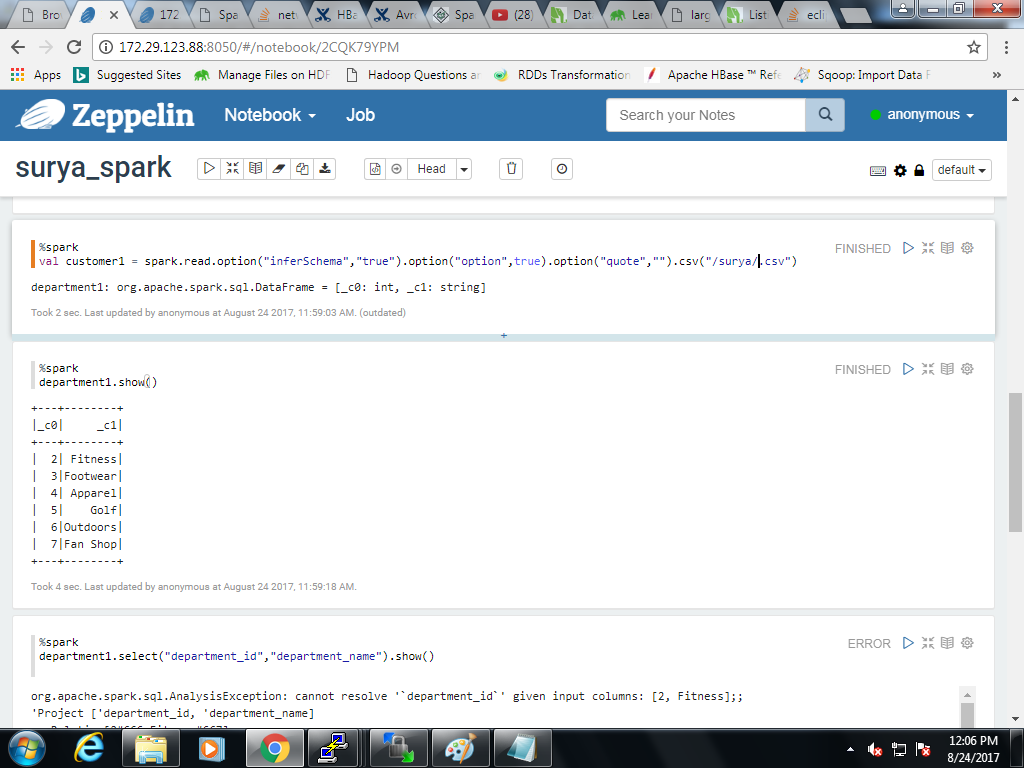
creating dataframe in spark:

https://stackoverflow.com/questions/44901920/create-a-spark-data-frame-from-a-csv-file-which-has-the-headers-in-the-first-row

val department1 = spark.read.option("header","true").option("inferschema","true").csv("/surya/departments.csv")

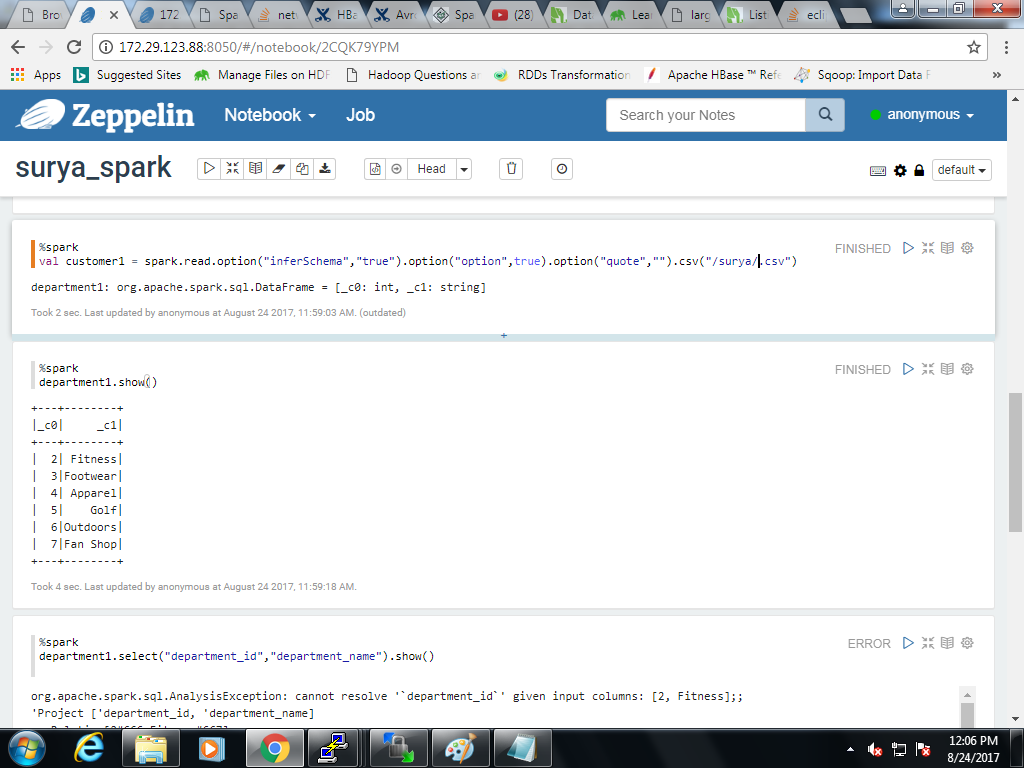
printing the schema in tree format:

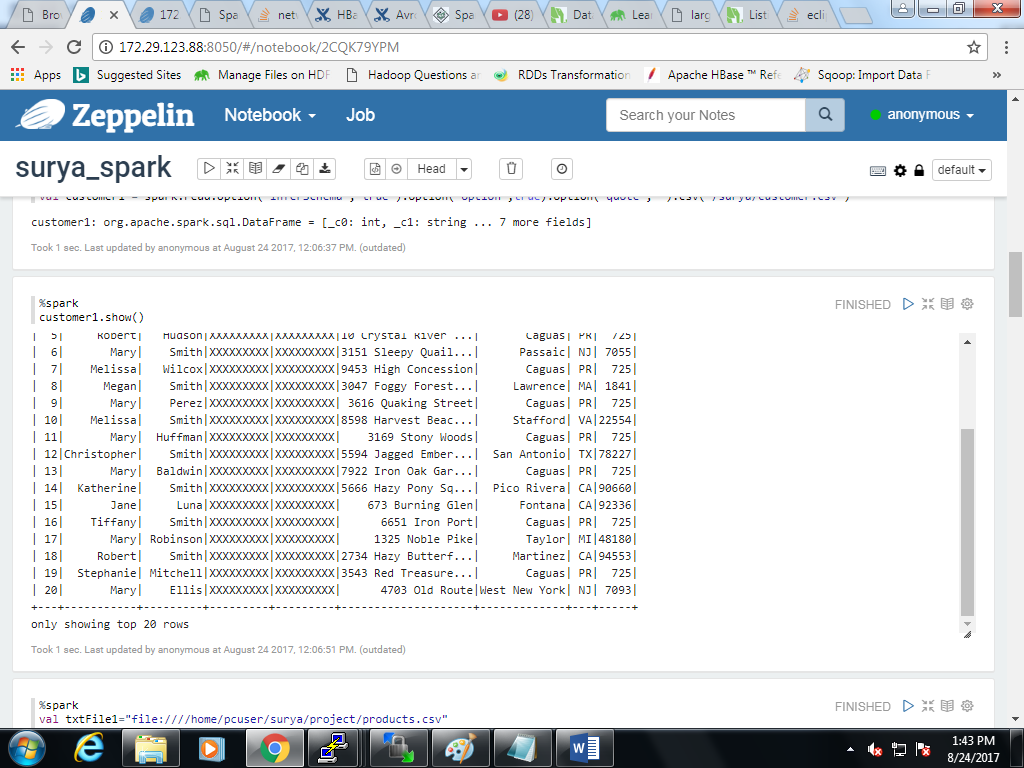
department1.printSchema()



val customer1 = spark.read.option("inferSchema","true").option("option",true).option("quote","").csv("/surya/customer.csv")

customer1.printSchema()





**creating tables in spark and accessing it using sql:**

Customers table:

%spark

val txtFile2="file:///home/pcuser/surya/project/customer.csv"

val txtData=sc.textFile(txtFile2)

txtData.cache()

case class customer100(customer\_id:Integer,

customer\_fname:String,

customer\_lname:String,

customer\_email:String,

customer\_password:String,

customer\_street:String,

customer\_city:String,

customer\_state:String,

customer\_zipcode:String

)

val df= txtData.map {line =>

val s= line.split(",")

val customer\_id=s(0).toInt

val customer\_fname=s(1)

val customer\_lname=s(2)

val customer\_email=s(3)

val customer\_password=s(4)

val customer\_street=s(5)

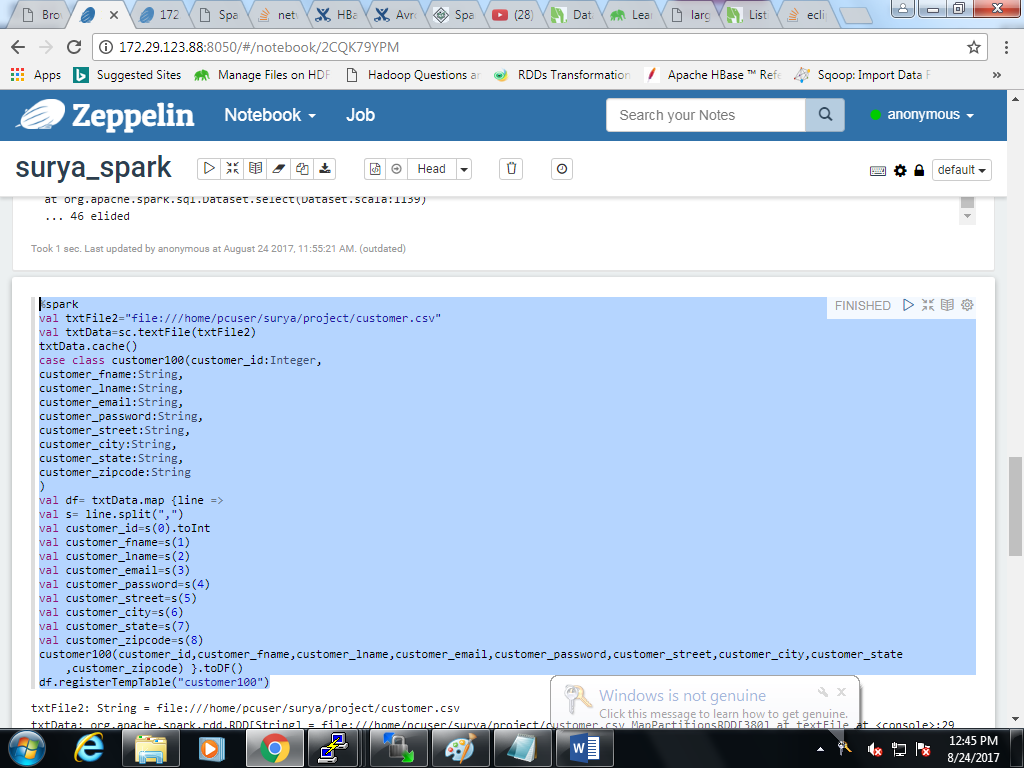
val customer\_city=s(6)

val customer\_state=s(7)

val customer\_zipcode=s(8)

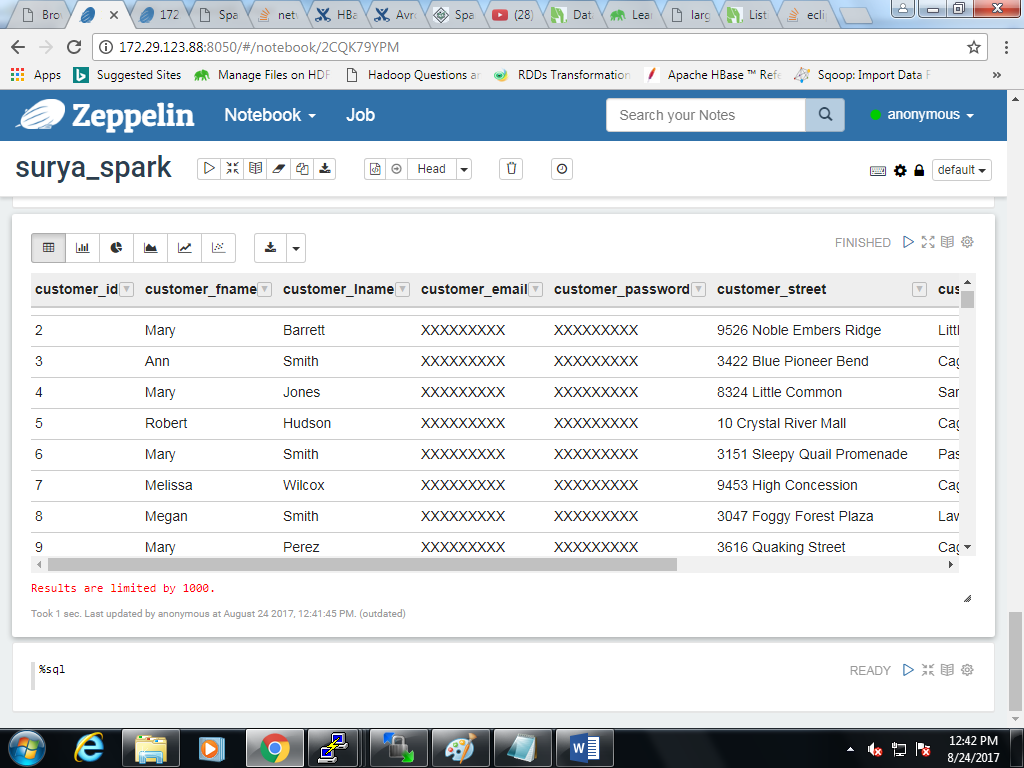
customer100(customer\_id,customer\_fname,customer\_lname,customer\_email,customer\_password,customer\_street,customer\_city,customer\_state,customer\_zipcode) }.toDF()

df.registerTempTable("customer100")



%sql

Select \* from customer100



Departments table:

Val txtFile="file:///home/pcuser/surya/project/departments.csv"

val txtData=sc.textFile(txtFile)

txtData.cache()

case class departments(department\_id:Integer,department\_name:String)

val df= txtData.map {line =>

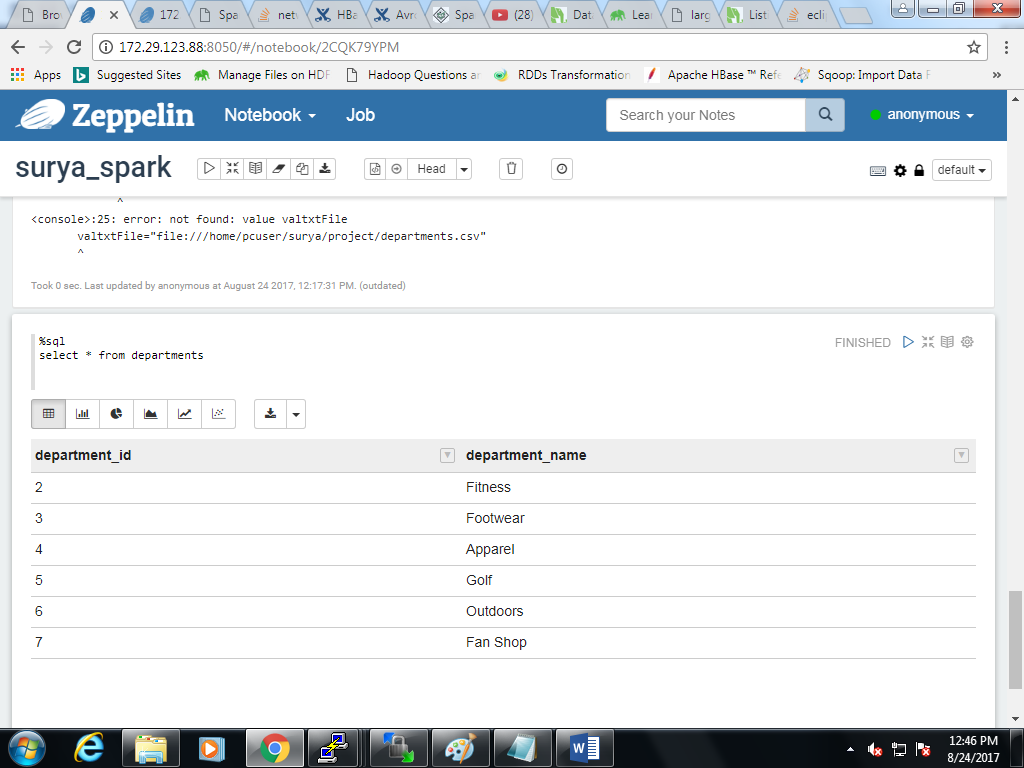
val s= line.split(",")

val id =s(0).toInt

val name=s(1)

departments(id,name) }.toDF()

df.registerTempTable("departments")



Products table:

%spark

val txtFile1="file:////home/pcuser/surya/project/products.csv"

val txtData1=sc.textFile(txtFile1)

txtData1.cache()

case class products(p\_id:Integer,p\_cat\_id:Integer,p\_name:String,p\_description:String,p\_price:String,p\_image:String)

val df=txtData1.map {line =>

val c = line.split(",")

val id = c(0).toInt

val c\_id = c(1).toInt

val p\_name = c(2)

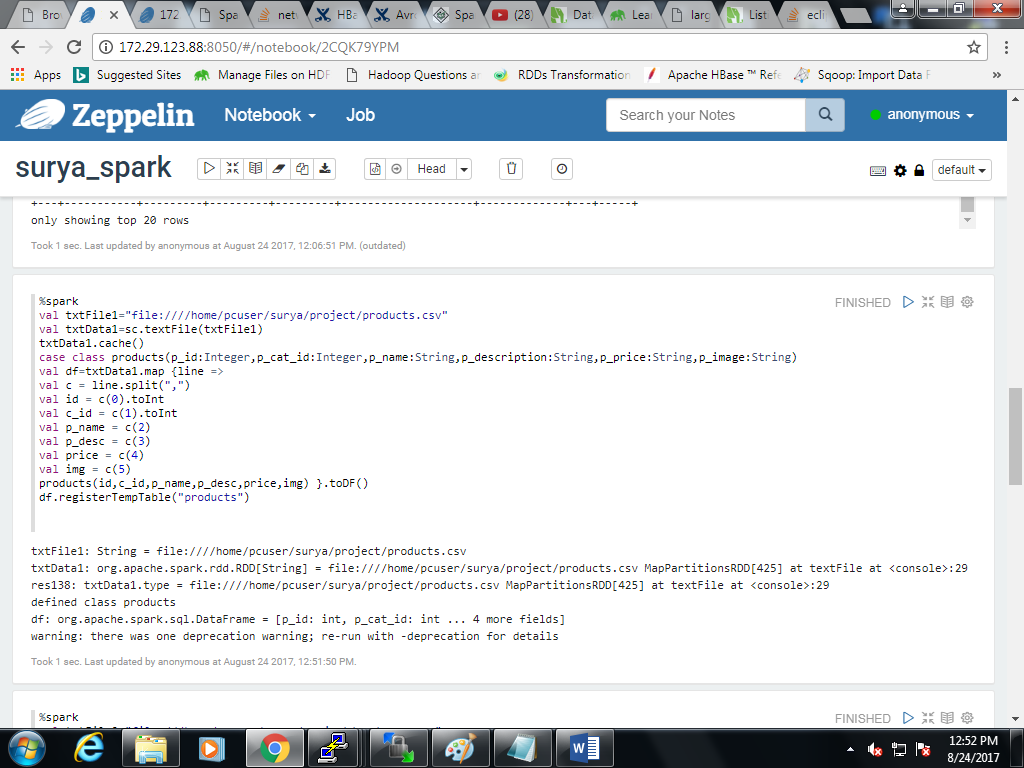
val p\_desc = c(3)

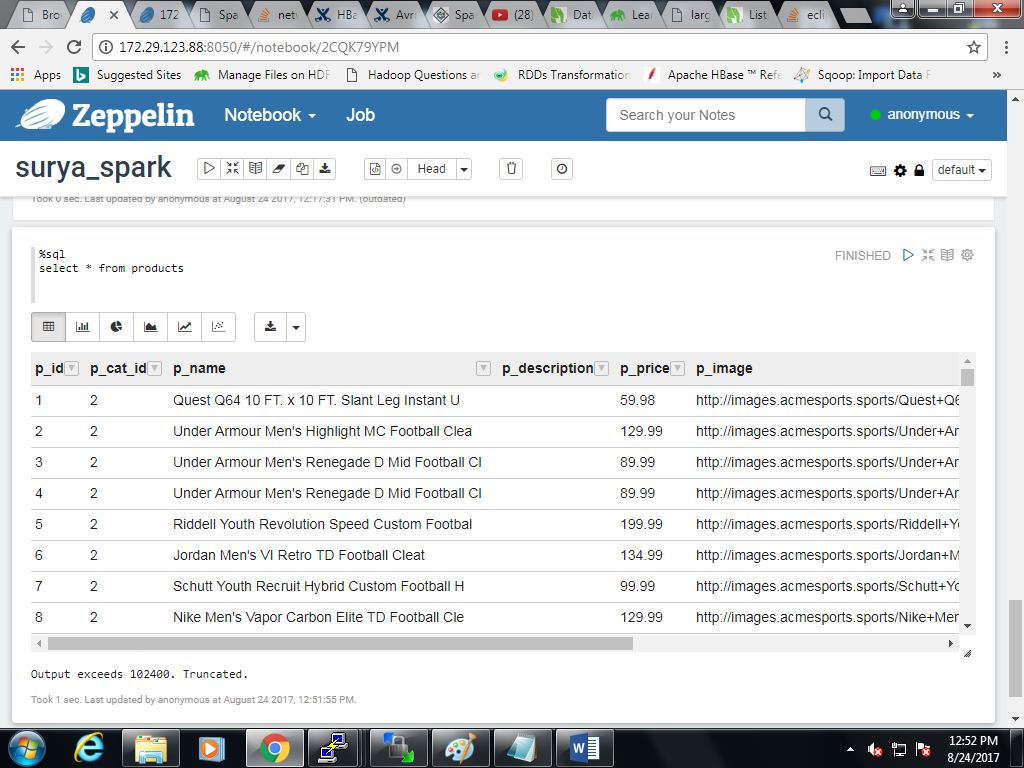
val price = c(4)

val img = c(5)

products(id,c\_id,p\_name,p\_desc,price,img) }.toDF()

df.registerTempTable("products")



Select \* from products

Order\_items table:

%spark

val txtFile3="file:///home/pcuser/surya/project/order\_items.csv"

val txtData=sc.textFile(txtFile3)

txtData.cache()

case class order\_item1 (order\_item\_id:Integer,

order\_item\_order\_id:Integer,

order\_item\_product\_id:Integer,

order\_item\_quantity:Integer,

order\_item\_subtotal:Double,

order\_item\_product\_price:Double

)

val df= txtData.map {line =>

val s= line.split(",")

val order\_item\_id =s(0).toInt

val order\_item\_order\_id=s(1).toInt

val order\_item\_product\_id=s(2).toInt

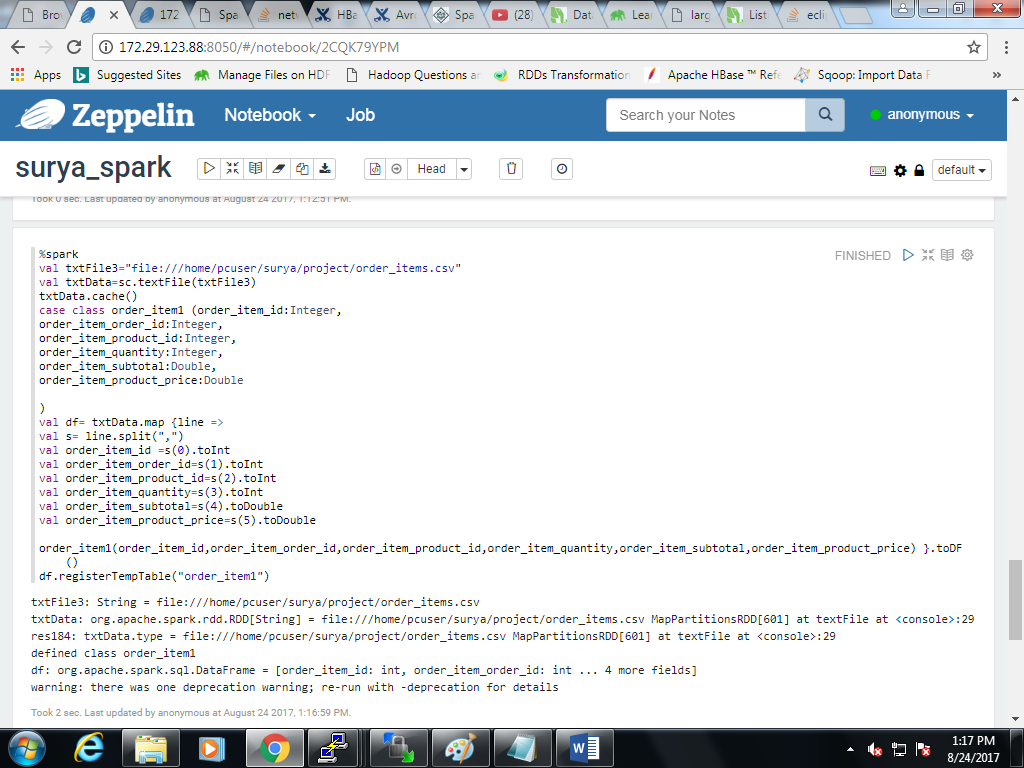
val order\_item\_quantity=s(3).toInt

val order\_item\_subtotal=s(4).toDouble

val order\_item\_product\_price=s(5).toDouble

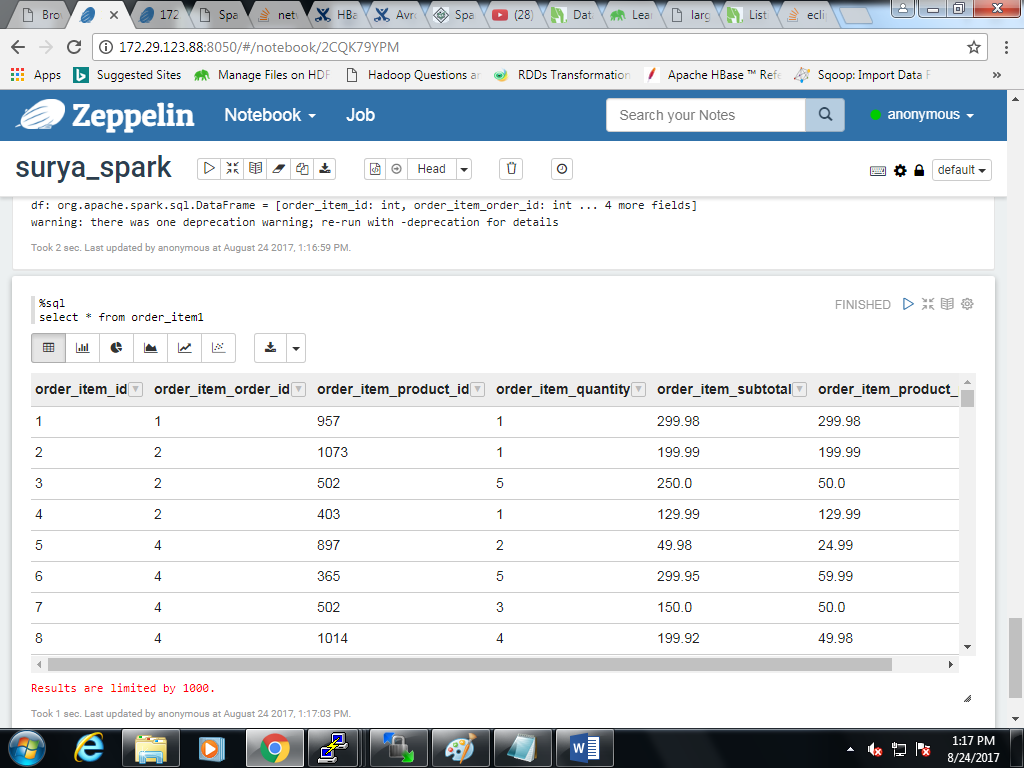
order\_item1(order\_item\_id,order\_item\_order\_id,order\_item\_product\_id,order\_item\_quantity,order\_item\_subtotal,order\_item\_product\_price) }.toDF()

df.registerTempTable("order\_item1")



Sql:

Select \* from order\_item



Orders table:

%spark

val txtFile4="file:///home/pcuser/surya/project/orders.csv"

val txtData4=sc.textFile(txtFile4)

txtData4.cache()

case class orderss(o\_id:Integer,o\_date:String,o\_cat\_id:Integer,o\_status:String)

val df=txtData4.map {line =>

val order= line.split(",")

val o\_id= order(0).toInt;

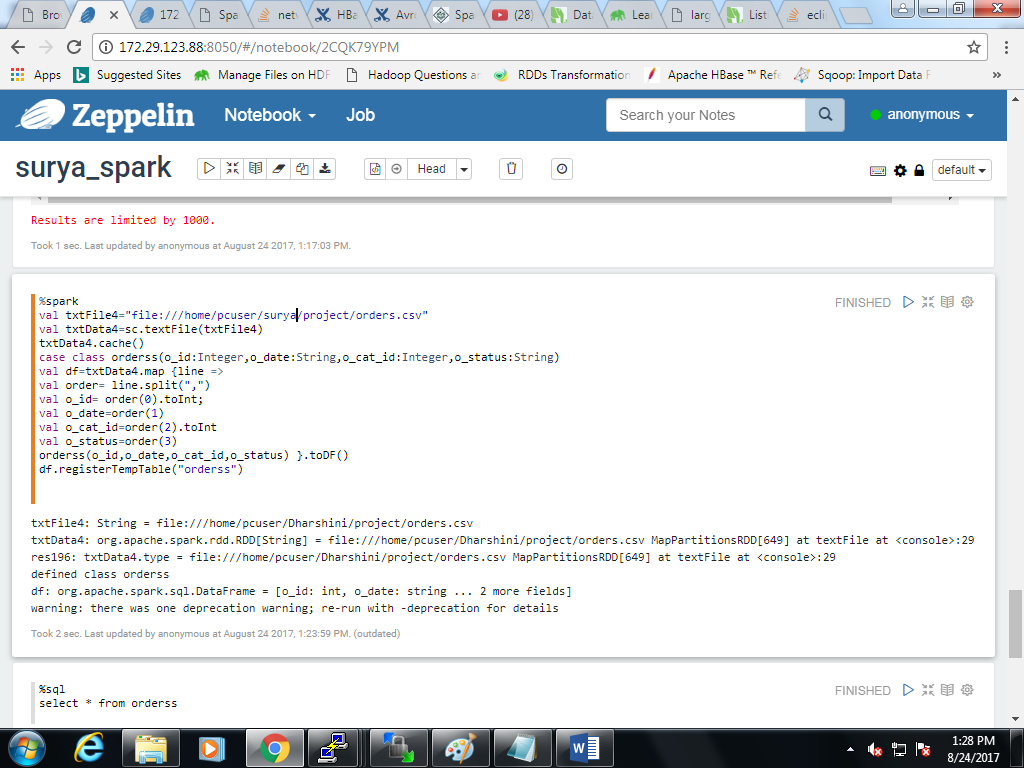
val o\_date=order(1)

val o\_cat\_id=order(2).toInt

val o\_status=order(3)

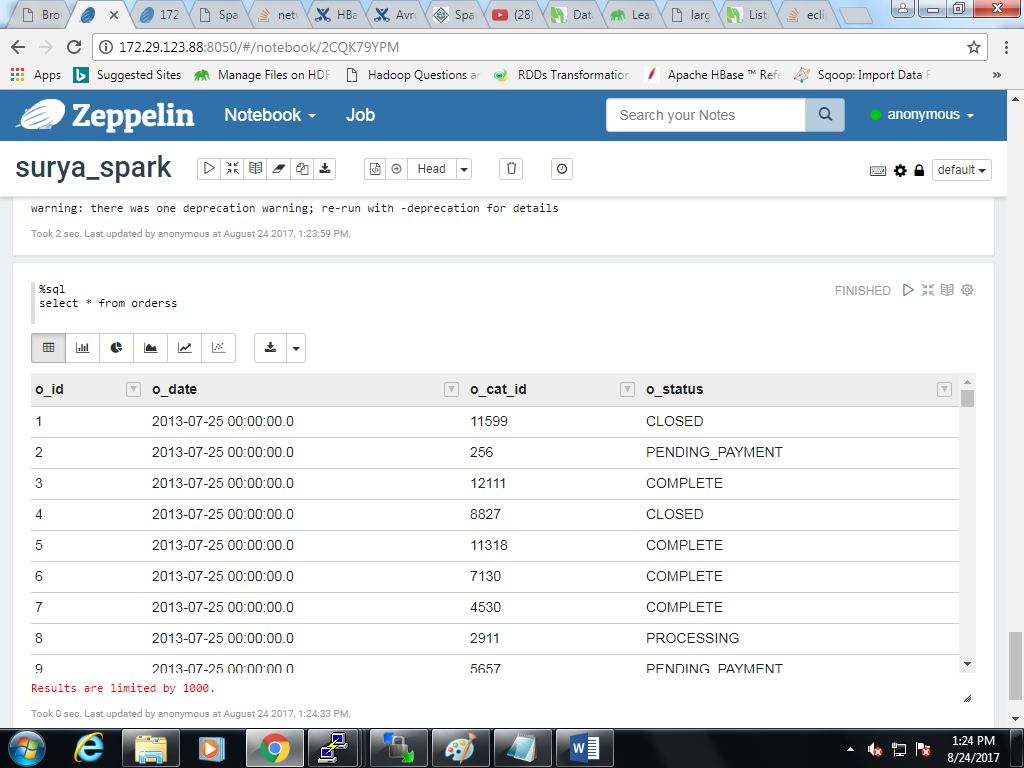
orderss(o\_id,o\_date,o\_cat\_id,o\_status) }.toDF()

df.registerTempTable("orderss")



Sql:

Select \* from orders



Category table:

%spark

val txtFile6="file:///home/pcuser/surya/project/categories.csv"

val txtData6=sc.textFile(txtFile6)

txtData6.cache()

case class category(

category\_id:Integer,

category\_department\_id:Integer,

category\_name:String

)

val df=txtData6.map {line =>

val ca= line.split(",")

val category\_id=ca(0).toInt

val category\_department\_id=ca(1).toInt

val category\_name=ca(2)

category(

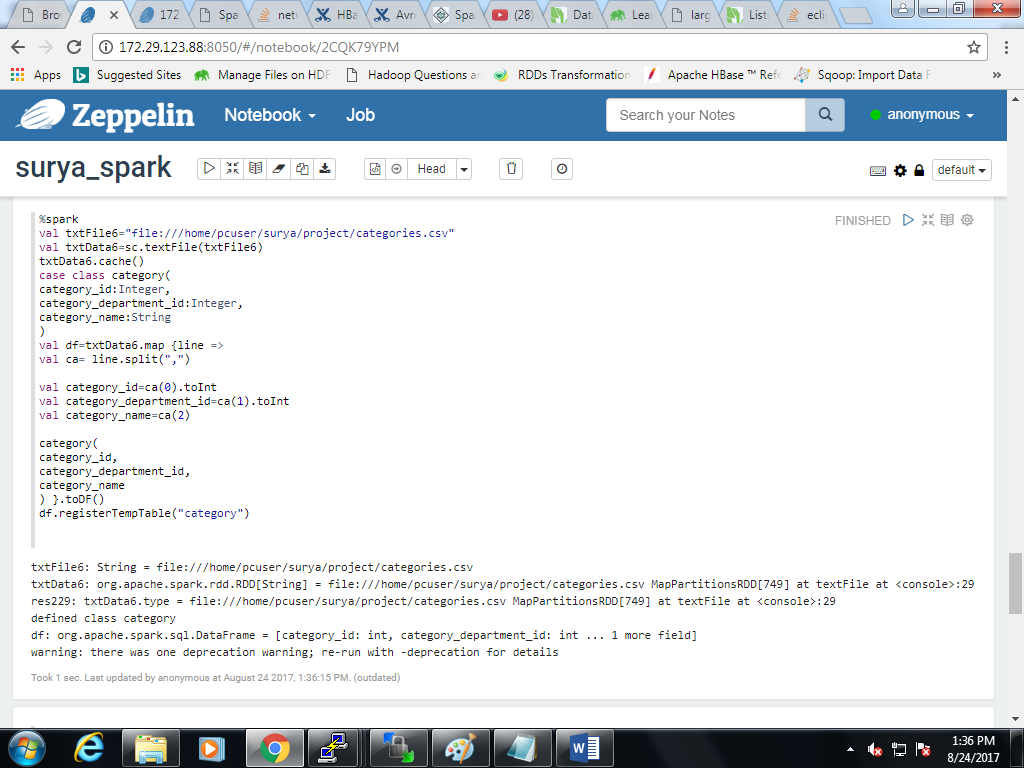
category\_id,

category\_department\_id,

category\_name

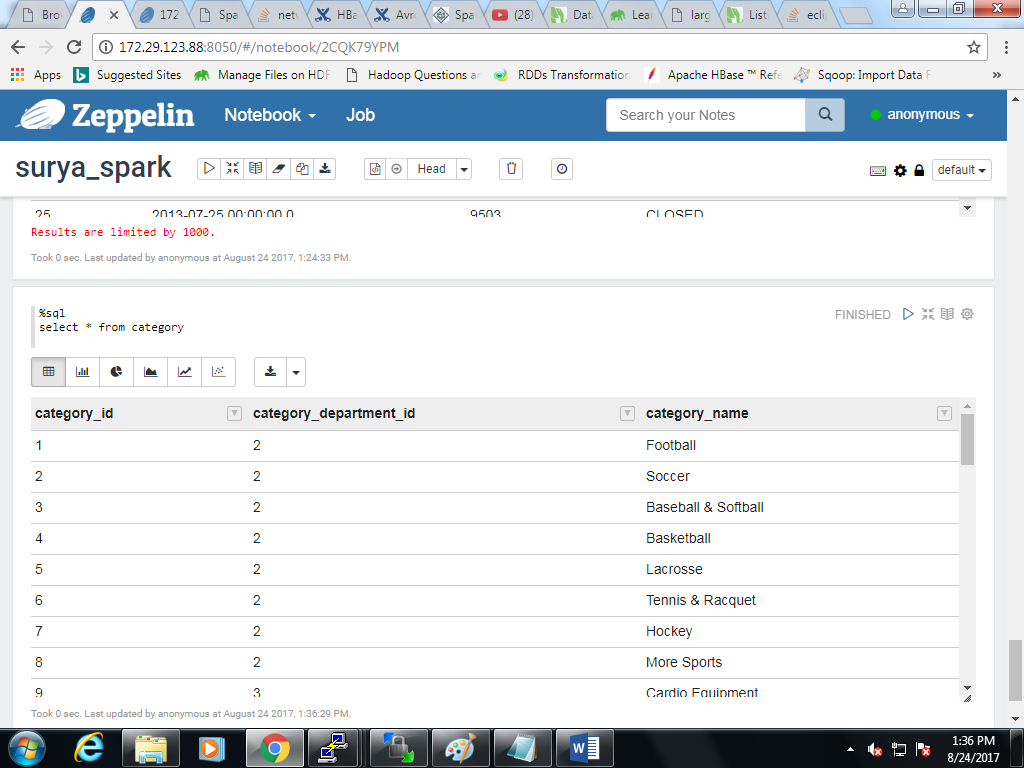
) }.toDF()

df.registerTempTable("category")



Sql:

Select \* from category



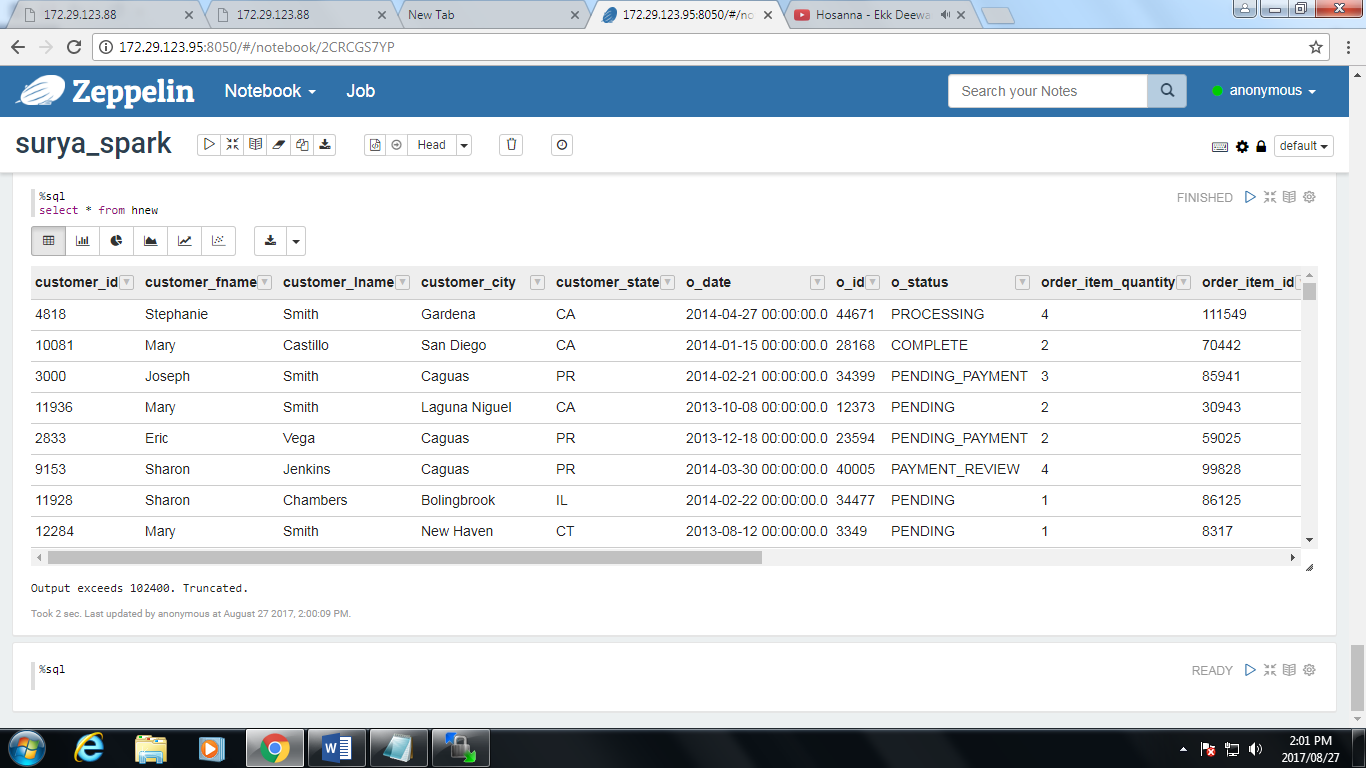
**Joining the tables for queries:**

Create table hnew as (select c.customer\_id,c.customer\_fname,c.customer\_lname,c.customer\_email,c.customer\_password,c.customer\_street,c.customer\_city,c.customer\_state,c.customer\_zipcode,

o.o\_date,o.o\_id,o.o\_status,oi.order\_item\_quantity,oi.order\_item\_id,oi.order\_item\_subtotal,oi.order\_item\_product\_price,p.p\_id,p.p\_name,p.p\_price,p.p\_image,p.p\_description,ca.category\_id,

ca.category\_name,d.department\_id,d.department\_name from order100 o, product100 p, order\_item100 oi,customer100 c,department100 d,category100 ca where (o.order\_id=oi.order\_item\_order\_id)

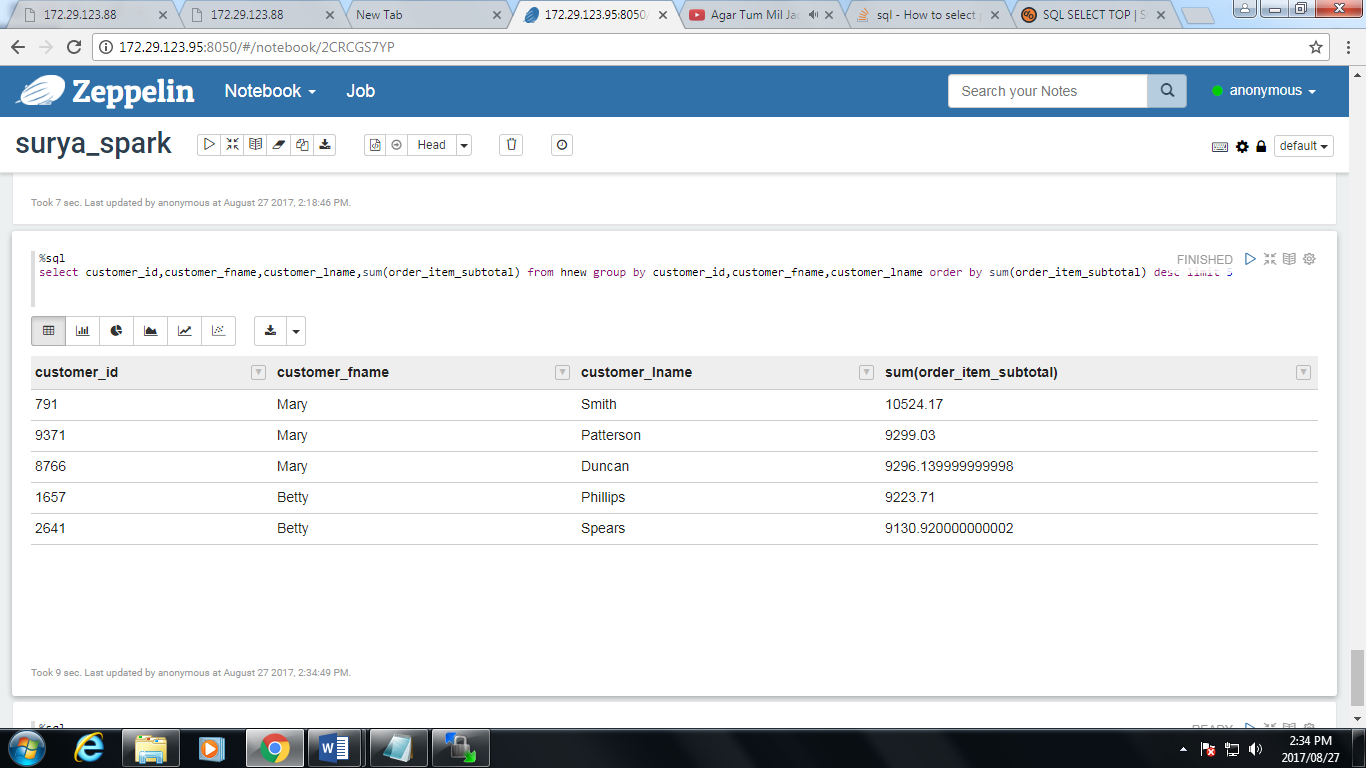
and (oi.order\_item\_product\_id=p.p\_id) and (c.customer\_id=o.o\_cat\_id) and (p.p\_cat\_id=ca.category\_id) and (ca.category\_department\_id=d.department\_id))



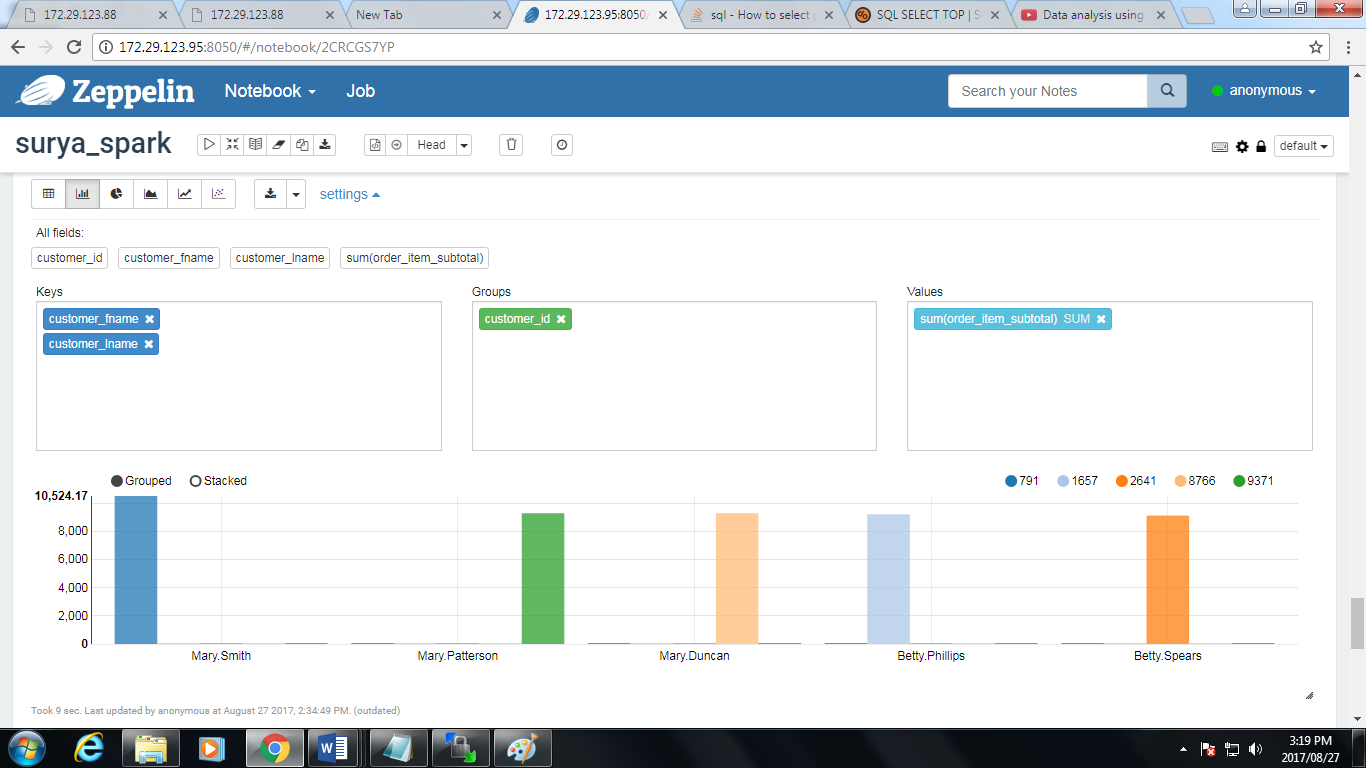
**Top 5 customers:**

%sql

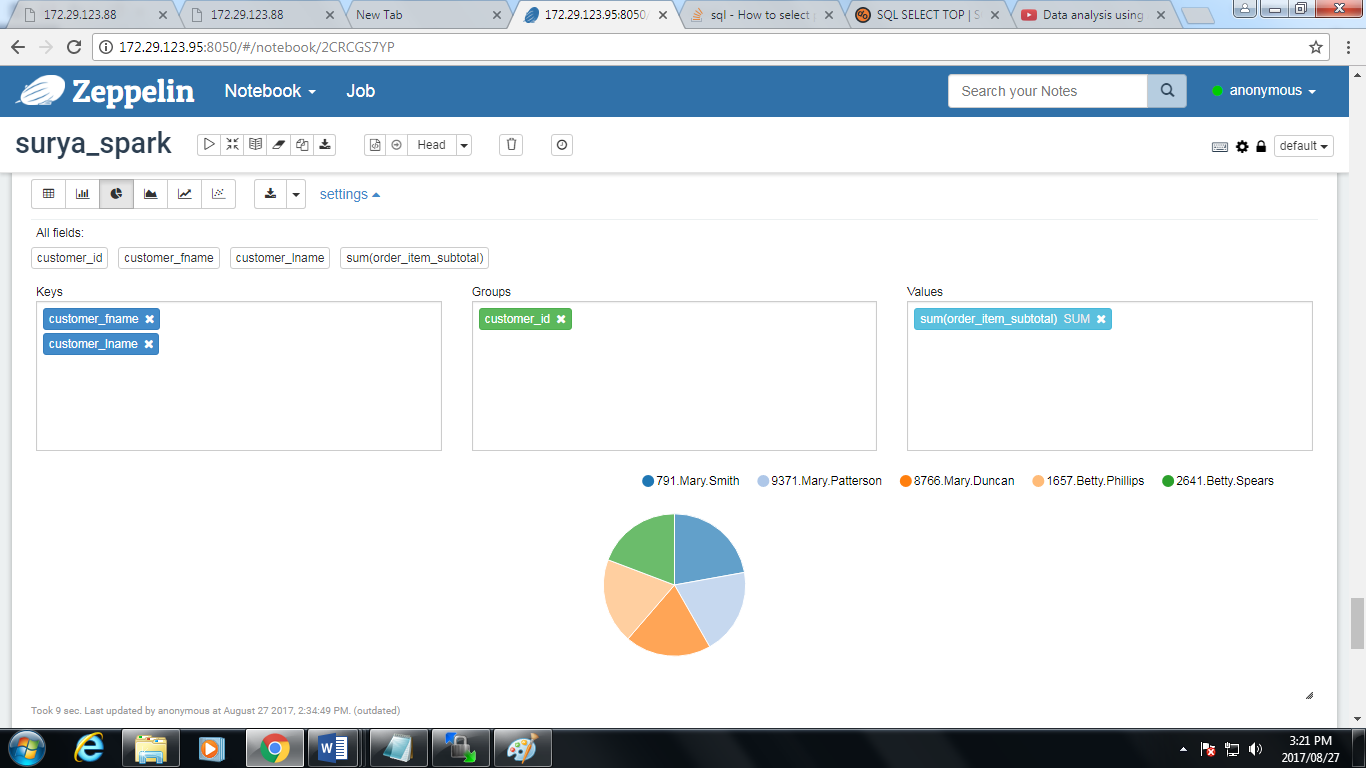
select customer\_id,customer\_fname,customer\_lname,sum(order\_item\_subtotal) from hnew group by customer\_id,customer\_fname,customer\_lname order by sum(order\_item\_subtotal) desc limit 5.

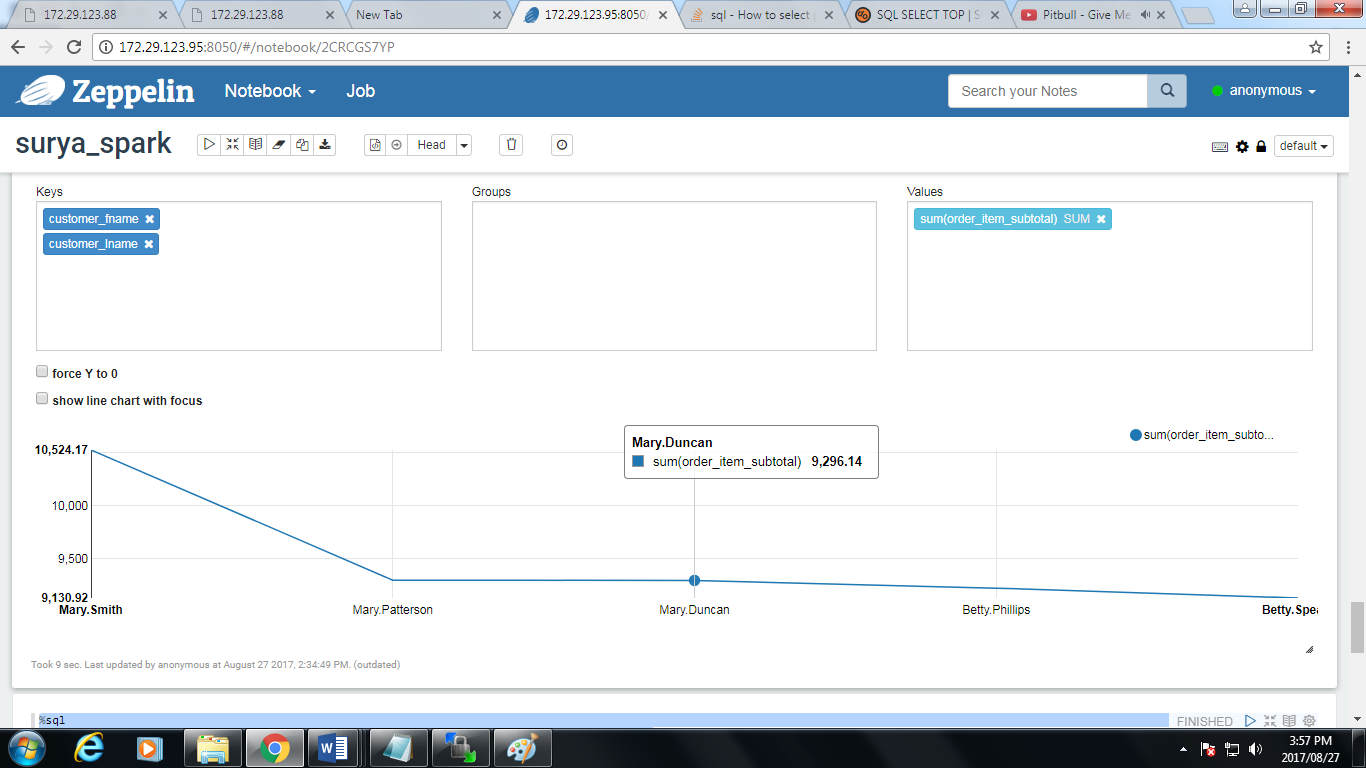
based on order\_item\_subtotal:

Bar graph:



Pie chart:

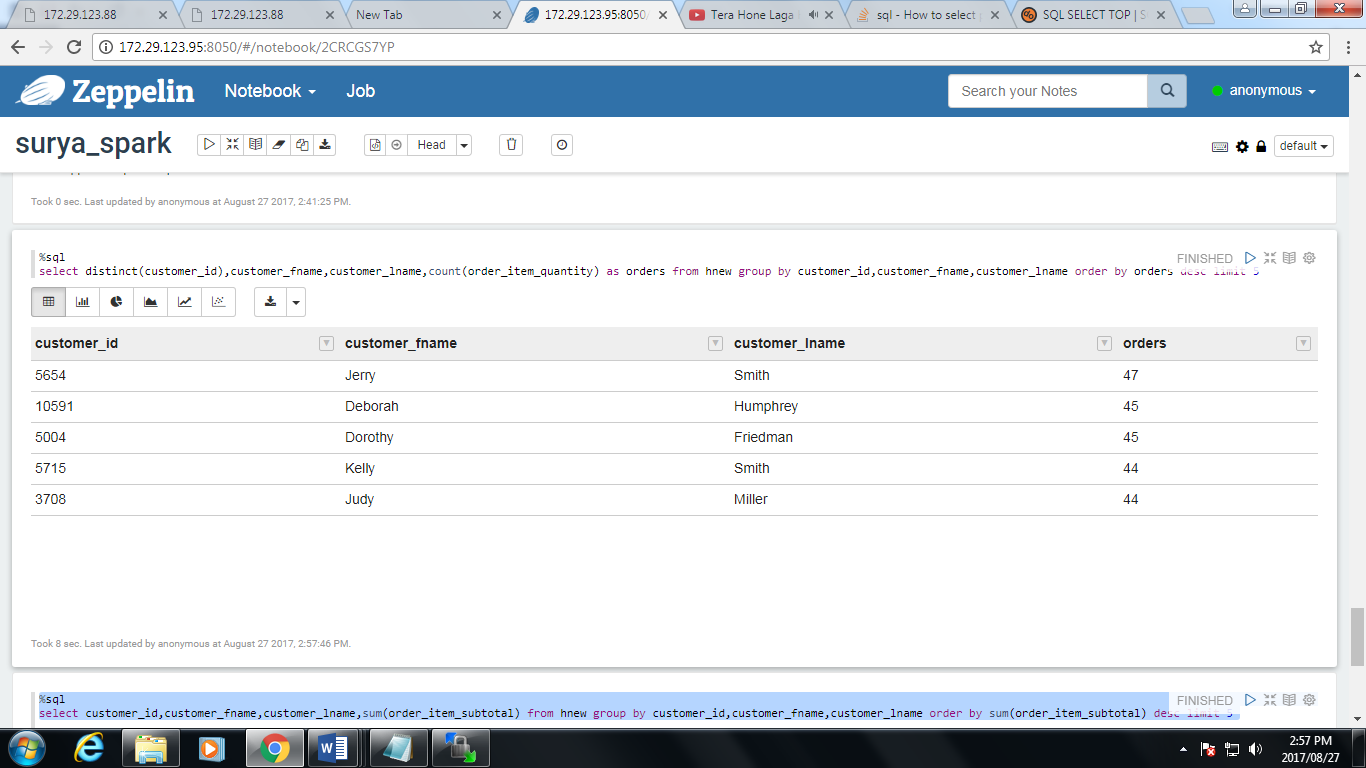




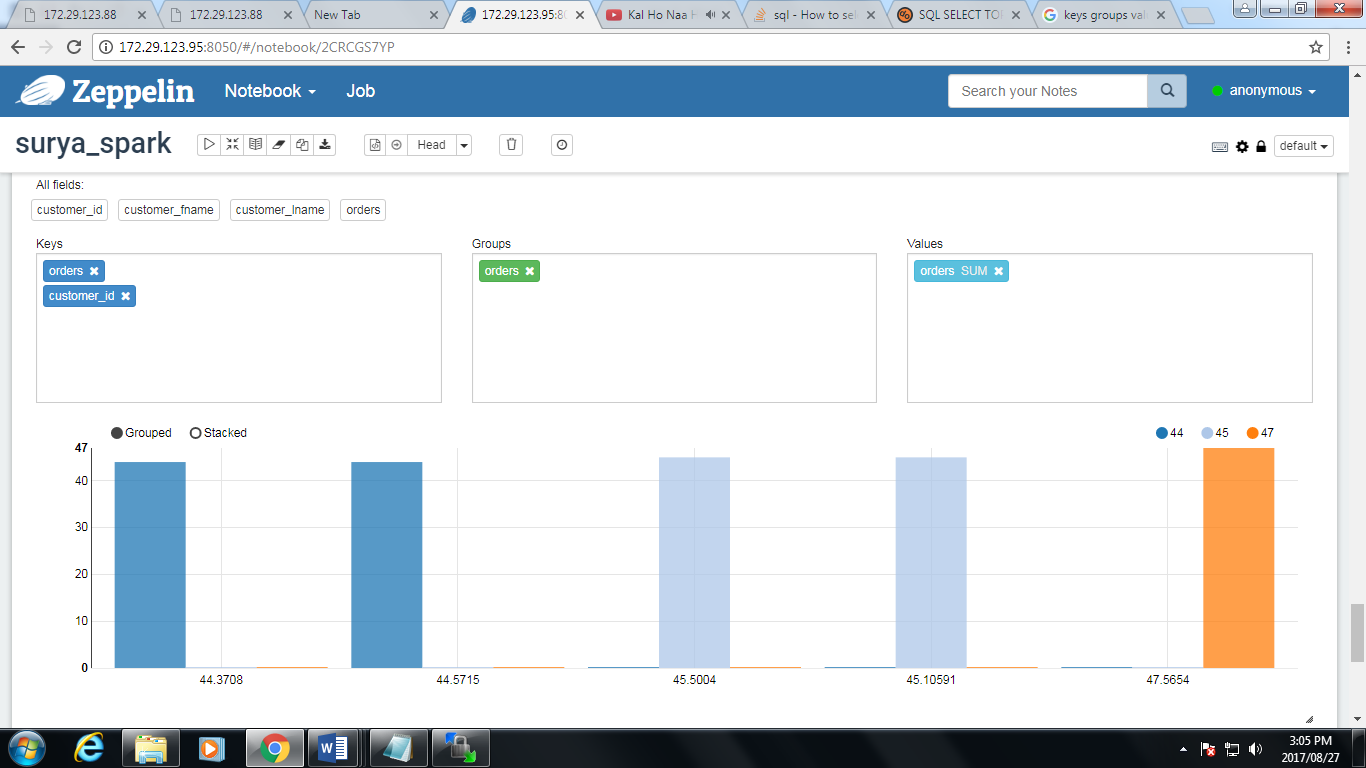
Based on number of orders:

%sql

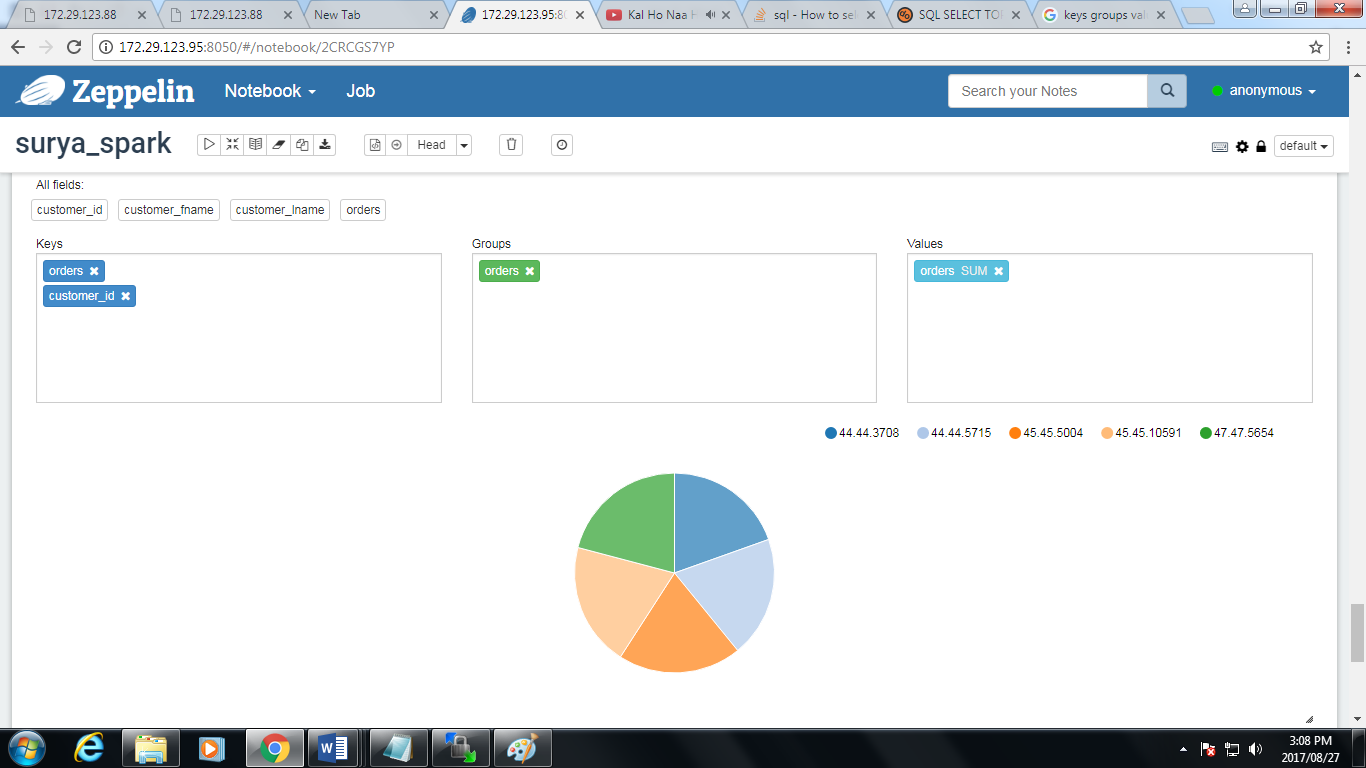
select distinct(customer\_id),customer\_fname,customer\_lname,count(order\_item\_quantity) as orders from hnew group by customer\_id,customer\_fname,customer\_lname order by orders desc limit 5



Bar graph:



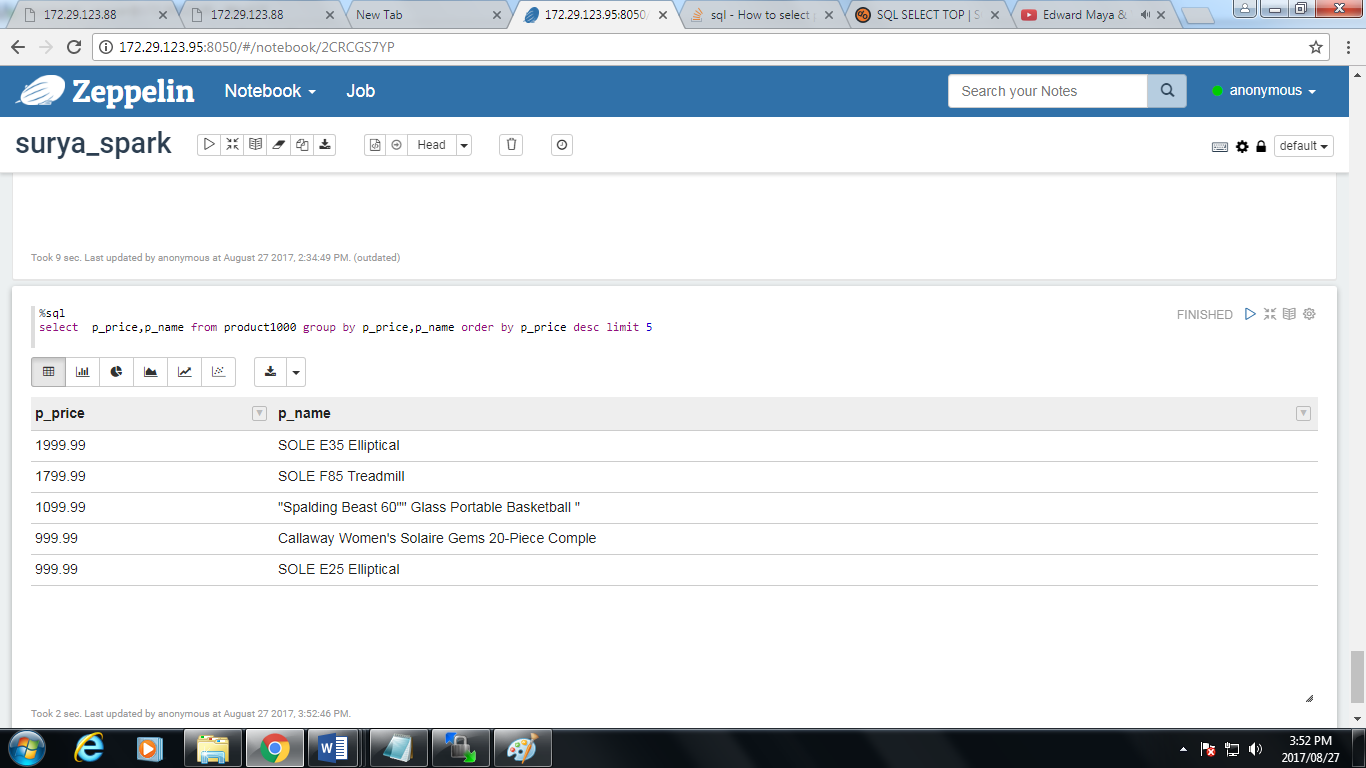
Pie chart:



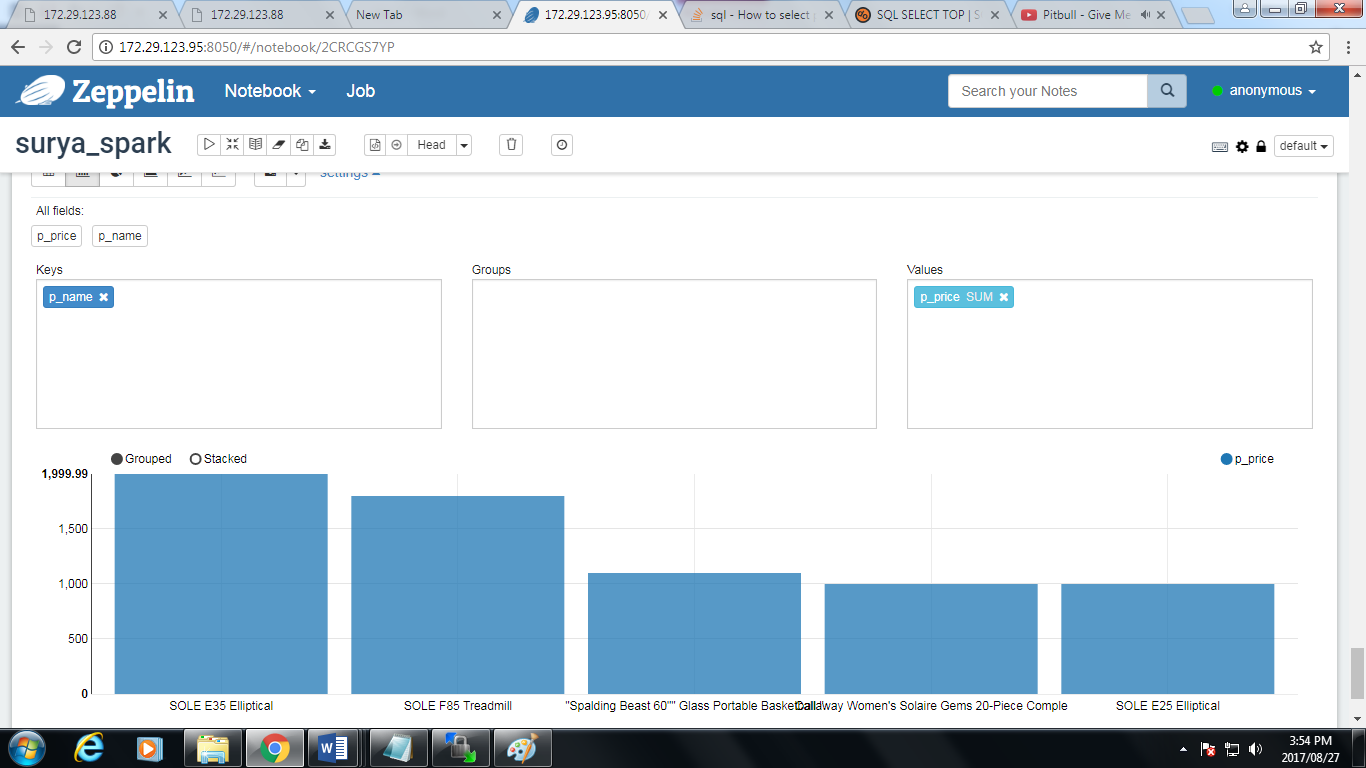
**Top 5 products:**

%sql

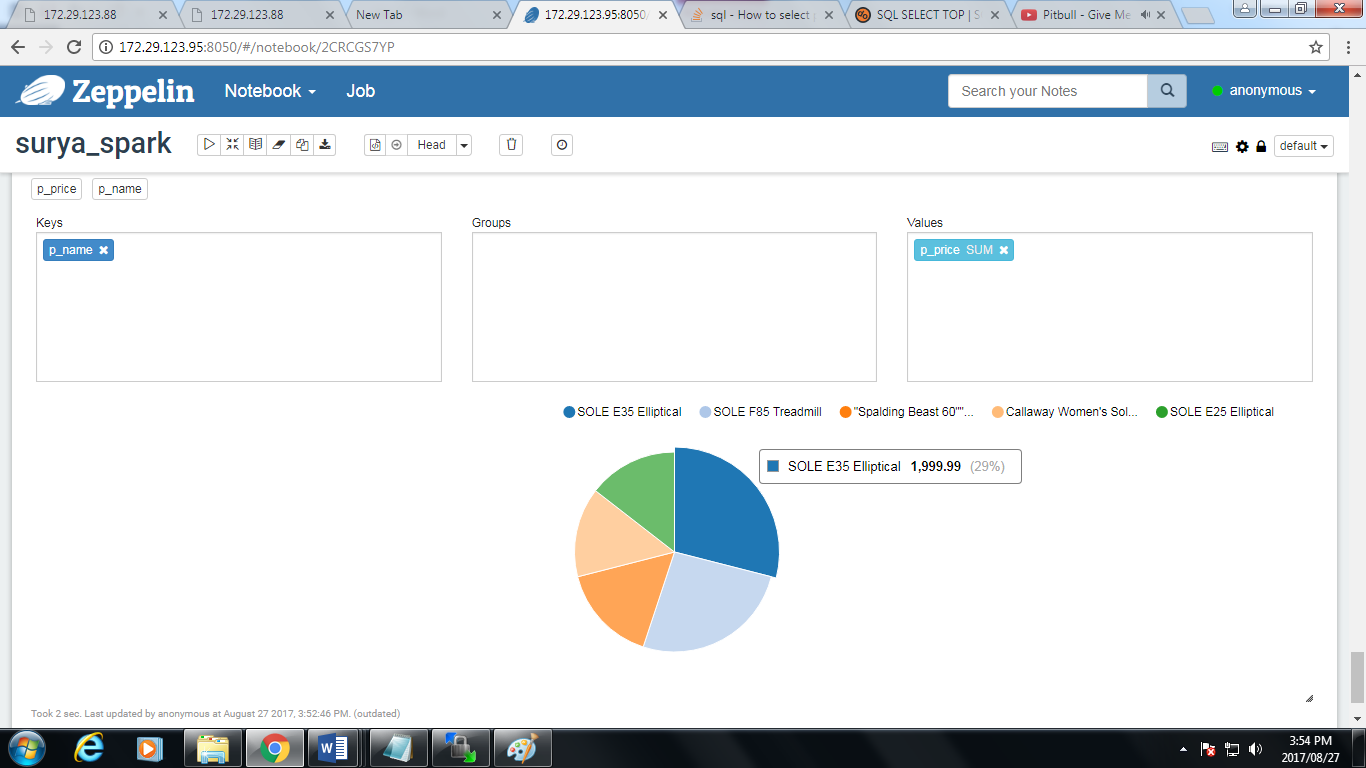
select p\_price,p\_name from product1000 group by p\_price,p\_name order by p\_price desc limit 5

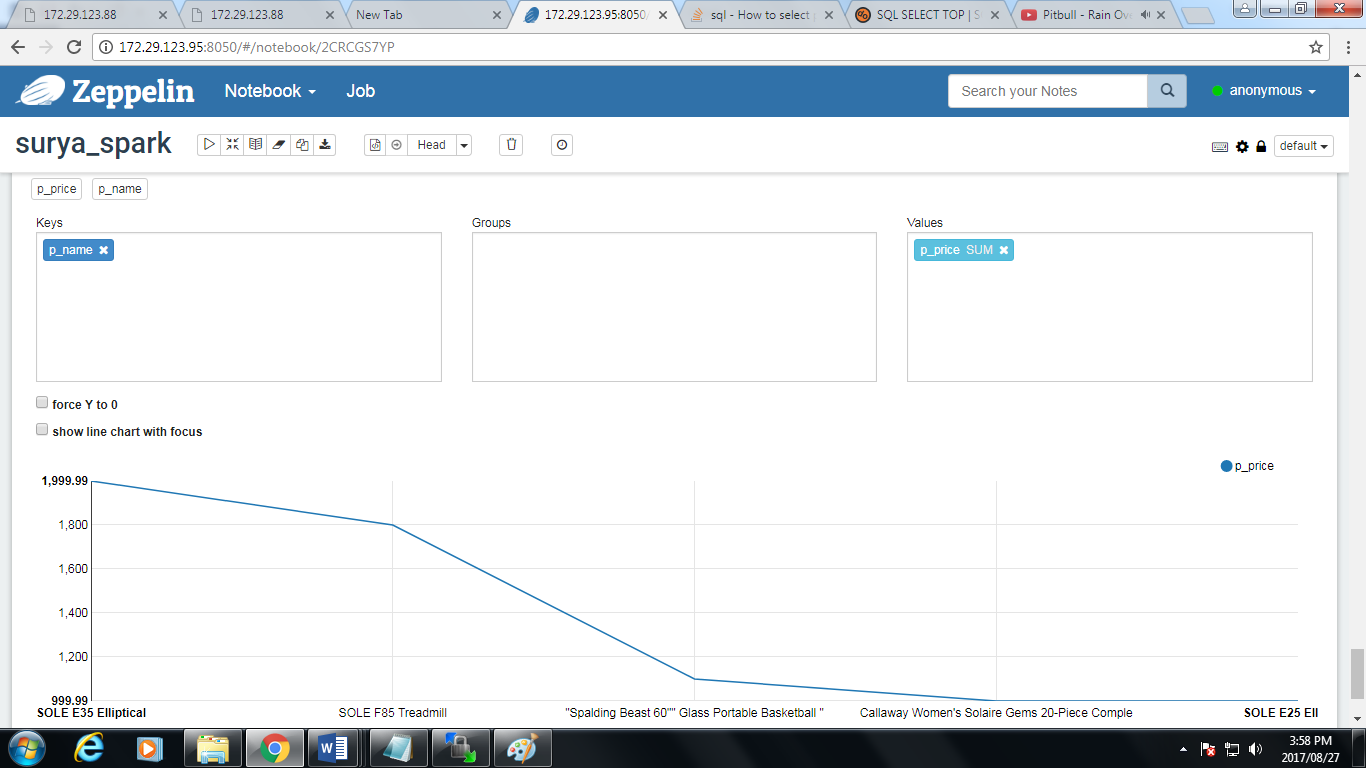


Bar graph:



Pie chart:



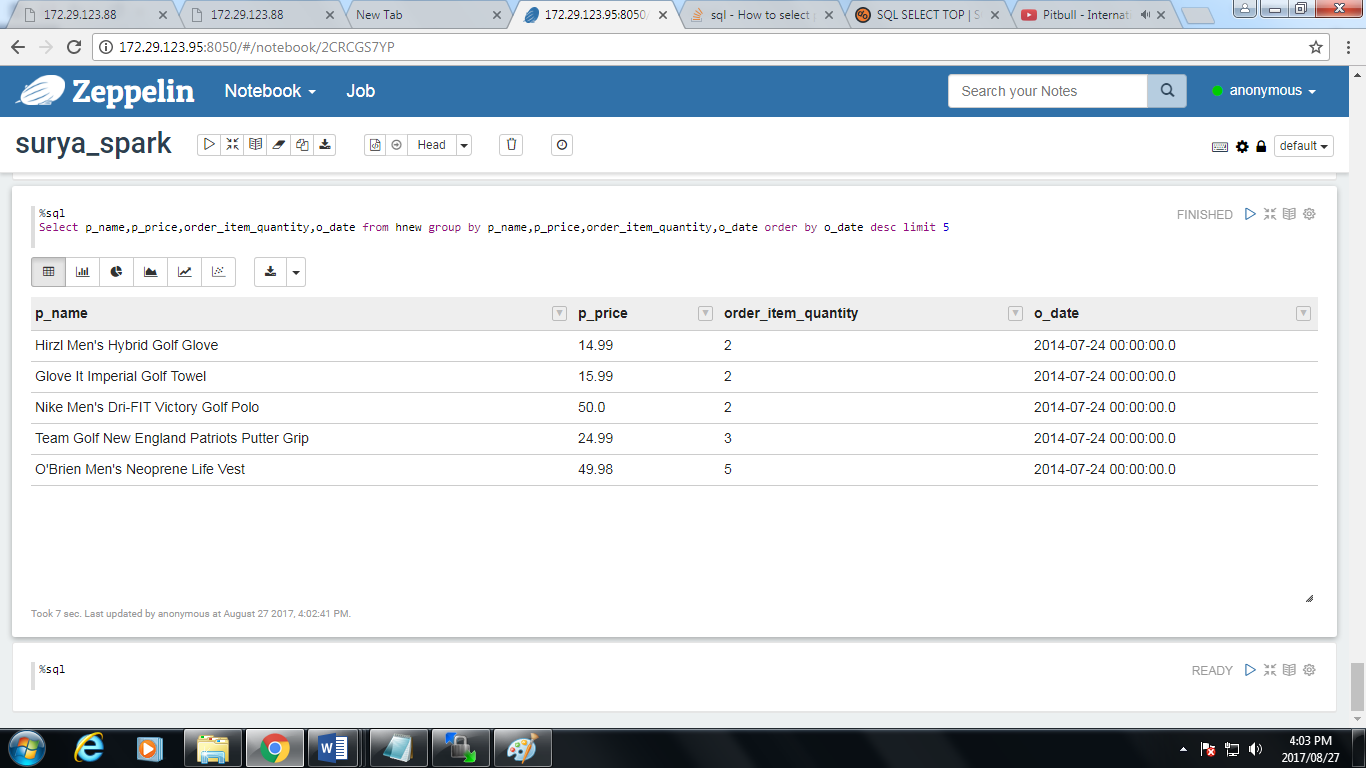


**TOP 5 ORDERS:**

**Based on date:**

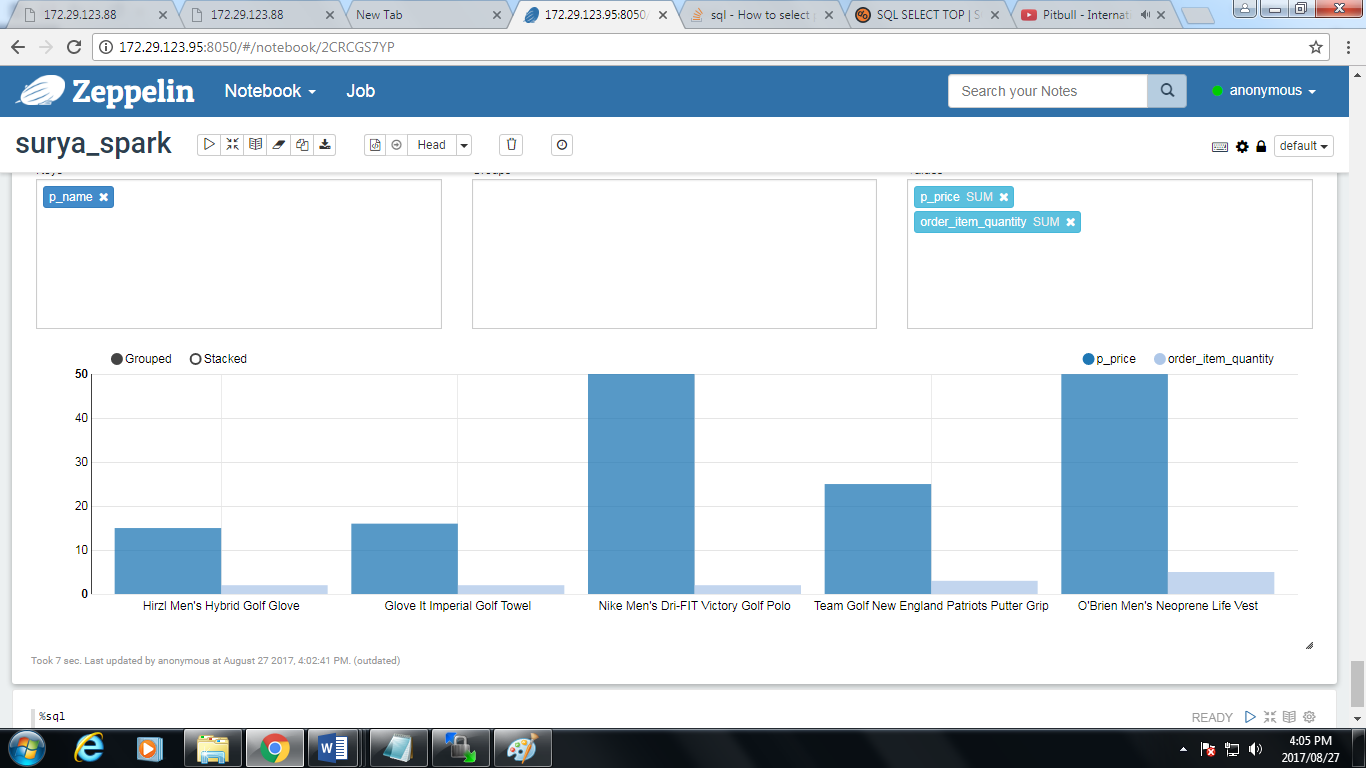
%sql

Select p\_name,p\_price,order\_item\_quantity,o\_date from hnew group by p\_name,p\_price,order\_item\_quantity,o\_date order by o\_date desc limit 5

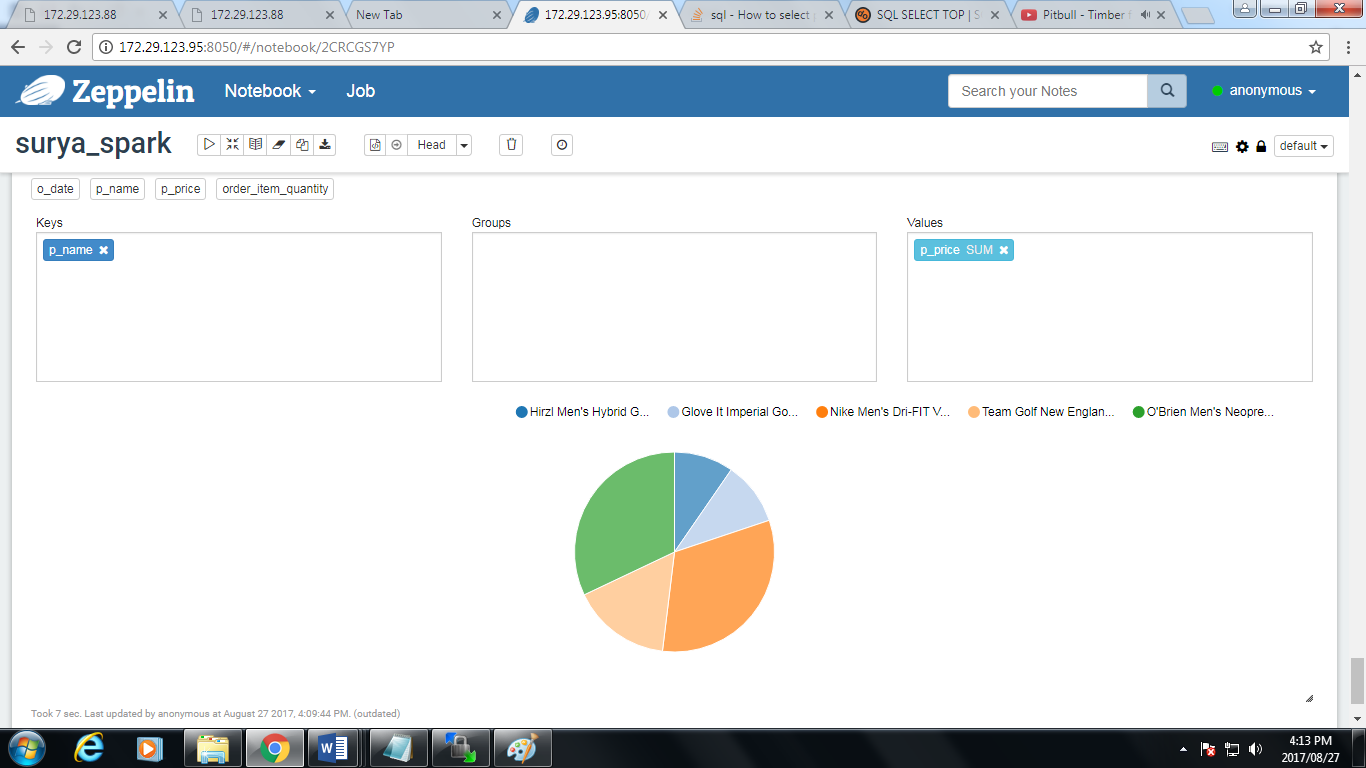


//Latest date from the data.

Bar graph:



Pie chart: based on price:

based on quantity:

