-- for customer.tbl

customer = LOAD '/user/bigdata/customer.tbl' using PigStorage('|')

as

(CUSTOMER\_ID:int,CUSTOMER\_CODE:chararray,COMPANY\_NAME:chararray,CONTACT\_NAME:chararray,CONTACT\_TITLE:chararray,CITY:chararray,REGION:chararray,POSTAL\_CODE:int,COUNTRY:chararray,PHONE:int,FAX:int);

-- for order\_detail.tbl

order\_detail = LOAD '/user/bigdata/order\_detail.tbl' using PigStorage('|')

as

(ORDER\_ID:int,PRODUCT\_ID:int,CUSTOMER\_ID:int,SALESPERSON\_ID:int,UNIT\_PRICE:float,QUANTITY:int,DISCOUNT:float);

-- for product.tbl

product = LOAD '/user/bigdata/product.tbl' using PigStorage('|')

as

(PRODUCT\_ID:int,PRODUCT\_NAME:chararray,UNIT\_PRICE:float,UNITS\_IN\_STOCK:int,UNITS\_ON\_ORDER:int,DISCONTINUED:chararray);

-- for salesperson.tbl

salesperson = LOAD '/user/bigdata/salesperson.tbl' using PigStorage('|')

as

(EMPLOYEE\_ID:int,LASTNAME:chararray,FIRSTNAME:chararray,TITLE:chararray,BIRTHDATE:chararray,HIREDATE:chararray,NOTES:chararray);

-- for order.tbl

orders = LOAD '/user/bigdata/order.tbl' using PigStorage('|')

as

(ORDER\_ID:int,ORDER\_DATE:chararray,SHIP\_VIA:int,SHIP\_CITY:chararray,SHIP\_REGION:chararray,SHIP\_POSTAL\_CODE:chararray,SHIP\_COUNTRY:chararray);

-- (1)

london = FILTER orders BY SHIP\_CITY == ' London ';

londonGroup = Group london All;

londonCount = foreach londonGroup Generate COUNT(london);

dump londonCount;

(33)

-- (2)

orderJoin = join orders by ORDER\_ID, order\_detail by ORDER\_ID;

not1996 = filter orderJoin by not(ORDER\_DATE matches '.\*1996.\*');

getProductID = foreach not1996 generate order\_detail::PRODUCT\_ID;

distinctProduct = distinct getProductID;

productGroup = Group distinctProduct All;

productCount = foreach productGroup Generate COUNT(distinctProduct);

dump productCount;

(77)

-- (3)

orderFields = foreach order\_detail generate ORDER\_ID, UNIT\_PRICE \* QUANTITY as value;

distinctOrder = distinct orderFields;

filterOrderID = FILTER distinctOrder BY (ORDER\_ID > 10269 AND ORDER\_ID < 10280);

orderGroup = group filterOrderID by (ORDER\_ID);

dump orderGroup;

orderSum = foreach orderGroup generate (filterOrderID.ORDER\_ID), SUM(filterOrderID.value) as ordervalue;

dump orderSum;

({(10270),(10270)},1376.0) ({(10271)},48.0) ({(10272),(10272),(10272)},1455.9999694824219) ({(10273),(10273),(10273),(10273),(10273)},2142.399932861328) ({(10274),(10274)},538.5999908447266) ({(10275),(10275)},307.1999969482422) ({(10276),(10276)},420.0) ({(10277),(10277)},1200.8000183105469) ({(10278),(10278),(10278),(10278)},1488.7999877929688) ({(10279)},468.0)

-- (4)

getDetails = foreach order\_detail generate SALESPERSON\_ID, UNIT\_PRICE, QUANTITY, UNIT\_PRICE \* QUANTITY as ordervalue;

groupDetails = group getDetails by SALESPERSON\_ID;

totalSales = foreach groupDetails generate group, SUM(getDetails.ordervalue) as totalOrderValue;

orderTotalSales = order totalSales by totalOrderValue desc;

top3 = limit orderTotalSales 3;

employeeTop3 = join top3 by group left outer, salesperson by EMPLOYEE\_ID;

getTop3 = foreach employeeTop3 generate EMPLOYEE\_ID, FIRSTNAME, LASTNAME;

dump getTop3;

(1, Nancy , Davolio )

(3, Janet , Leverling )

(4, Margaret , Peacock )

-- (5)

odxProduct = join order\_detail by PRODUCT\_ID, product by PRODUCT\_ID;

filterTofuIkura = filter odxProduct BY PRODUCT\_NAME == ' Tofu ' OR PRODUCT\_NAME == ' Ikura ';

distinctTofuIkura = distinct filterTofuIkura;

TofuIkuraGroup = Group distinctTofuIkura All;

totalCount = foreach TofuIkuraGroup Generate COUNT(distinctTofuIkura);

dump totalCount;

(55)