The data set includes the following entities:

|  |  |
| --- | --- |
| **Entity** | **Description** |
| customers\_dim | Customer Data |
| email\_addresses\_dim | Email addresses of customers |
| product\_dim | Product Details |
| customer\_addresses\_dim | Address of each customer |
| order\_lineitems | Line Item for each Order |
| Orders | Details of an Order |
| categories\_dim | Category for each Product |

CREATE TABLE retail\_demo.customers\_dim

(

customer\_id TEXT,

first\_name TEXT,

last\_name TEXT,

gender TEXT

)

CREATE TABLE retail\_demo.email\_addresses\_dim

(

customer\_id TEXT,

email\_address TEXT

)

CREATE TABLE retail\_demo.products\_dim

(

product\_id TEXT,

category\_id TEXT,

price TEXT,

product\_name TEXT

)

CREATE TABLE retail\_demo.categories\_dim

(

category\_id integer NOT NULL,

category\_name character varying(400) NOT NULL

)

CREATE TABLE retail\_demo.customer\_addresses\_dim\_hawq

(

customer\_address\_id TEXT,

customer\_id TEXT,

valid\_from\_timestamp TEXT,

valid\_to\_timestamp TEXT,

house\_number TEXT,

street\_name TEXT,

appt\_suite\_no TEXT,

city TEXT,

state\_code TEXT,

zip\_code TEXT,

zip\_plus\_four TEXT,

country TEXT,

phone\_number TEXT

)

CREATE TABLE retail\_demo.order\_lineitems\_hawq

(

order\_id TEXT,

order\_item\_id TEXT,

product\_id TEXT,

product\_name TEXT,

customer\_id TEXT,

store\_id TEXT,

item\_shipment\_status\_code TEXT,

order\_datetime TEXT,

ship\_datetime TEXT,

item\_return\_datetime TEXT,

item\_refund\_datetime TEXT,

product\_category\_id TEXT,

product\_category\_name TEXT,

payment\_method\_code TEXT,

tax\_amount TEXT,

item\_quantity TEXT,

item\_price TEXT,

discount\_amount TEXT,

coupon\_code TEXT,

coupon\_amount TEXT,

ship\_address\_line1 TEXT,

ship\_address\_line2 TEXT,

ship\_address\_line3 TEXT,

ship\_address\_city TEXT,

ship\_address\_state TEXT,

ship\_address\_postal\_code TEXT,

ship\_address\_country TEXT,

ship\_phone\_number TEXT,

ship\_customer\_name TEXT,

ship\_customer\_email\_address TEXT,

ordering\_session\_id TEXT,

website\_url TEXT

)

CREATE TABLE retail\_demo.orders\_hawq

(

order\_id TEXT,

customer\_id TEXT,

store\_id TEXT,

order\_datetime TEXT,

ship\_completion\_datetime TEXT,

return\_datetime TEXT,

refund\_datetime TEXT,

payment\_method\_code TEXT,

total\_tax\_amount TEXT,

total\_paid\_amount TEXT,

total\_item\_quantity TEXT,

total\_discount\_amount TEXT,

coupon\_code TEXT,

coupon\_amount TEXT,

order\_canceled\_flag TEXT,

has\_returned\_items\_flag TEXT,

has\_refunded\_items\_flag TEXT,

fraud\_code TEXT,

fraud\_resolution\_code TEXT,

billing\_address\_line1 TEXT,

billing\_address\_line2 TEXT,

billing\_address\_line3 TEXT,

billing\_address\_city TEXT,

billing\_address\_state TEXT,

billing\_address\_postal\_code TEXT,

billing\_address\_country TEXT,

billing\_phone\_number TEXT,

customer\_name TEXT,

customer\_email\_address TEXT,

ordering\_session\_id TEXT,

website\_url TEXT

)

CREATE TABLE retail\_demo.date\_dim\_hawq

(

calendar\_day date,

reporting\_year smallint,

reporting\_quarter smallint,

reporting\_month smallint,

reporting\_week smallint,

reporting\_dow smallint

)

CREATE TABLE retail\_demo.payment\_methods\_hawq

(

payment\_method\_id smallint,

payment\_method\_code character varying(20)

)

Find the top ten postal codes by revenue:

select billing\_address\_postal\_code, sum(total\_paid\_amount::float8) as total,

sum(total\_tax\_amount::float8) as tax

from retail\_demo.orders\_hawq

group by billing\_address\_postal\_code

order by total desc limit 10;

billing\_address\_postal\_code | total | tax

-----------------------------+-----------+-----------

48001 | 111868.32 | 6712.0992

15329 | 107958.24 | 6477.4944

42714 | 103244.58 | 6194.6748

41030 | 101365.5 | 6081.93

50223 | 100511.64 | 6030.6984

03106 | 83566.41 | 0

57104 | 77383.63 | 3095.3452

23002 | 73673.66 | 3683.683

25703 | 68282.12 | 4096.9272

26178 | 66836.4 | 4010.184

(10 rows)

Find the most-recent order date for the customer

1.Start the Pig interactive grunt shell by issuing the pig command:

$ pig

grunt>

2.Verify that orders.tsv.gz is available in HDFS:

grunt> fs -ls /retail\_demo/orders;

Found 1 items

-rw-r--r-- 3 gpadmin hadoop 72797064 2013-06-25 10:13 /retail\_demo/orders/orders.tsv.gz

3. Create a relation named orders from orders.tsv.tz:

grunt> orders = LOAD '/retail\_demo/orders/orders.tsv.gz'

USING PigStorage('\t') AS (

order\_id : long,

customer\_id : int,

store\_id : int,

order\_datetime : chararray,

ship\_completion\_datetime : chararray,

return\_datetime : chararray,

refund\_datetime : chararray,

payment\_method\_code : chararray,

total\_tax\_amount :double,

total\_paid\_amount : double,

total\_item\_quantity : int,

total\_discount\_amount : int,

coupon\_code : chararray,

coupon\_amount : int,

order\_canceled\_flag : chararray,

has\_returned\_items\_flag : chararray,

has\_refunded\_items\_flag : chararray,

fraud\_code : chararray,

fraud\_resolution\_code : chararray,

billing\_address\_line1 : chararray,

billing\_address\_line2 : chararray,

billing\_address\_line3 : chararray,

billing\_address\_city : chararray,

billing\_address\_state : chararray,

billing\_address\_postal\_code : int,

billing\_address\_country : chararray,

billing\_phone\_number : chararray,

customer\_name : chararray,

customer\_email\_address :chararray,

ordering\_session\_id : int,

website\_url : chararray

);

Note that Pig can process a compressed file natively.

4. Group records by customer\_id:

grunt> records\_group = GROUP orders BY customer\_id;

The group statement groups the records.

## Finding the most-recent order date for the customer

1. Get the most recent order using the max function:

grunt> recent\_order\_date = FOREACH records\_group GENERATE

flatten(orders.customer\_id) AS customer\_id,

flatten(orders.order\_id) AS order\_id,

MAX(orders.order\_datetime) AS order\_datetime;

The flatten function is used to remove a level of nesting from a relation.

1. Collect distinct records from the recent\_order\_date relation:

grunt> distinct\_order\_date = distinct recent\_order\_date;

1. Join the orders and distinct\_order\_date relations by customer\_id,order\_id,order\_datetime:

grunt> rowOrder = join orders by (customer\_id,order\_id,order\_datetime),

distinct\_order\_date by (customer\_id,order\_id,order\_datetime);

1. Filter the records in the rowOrder relation:

grunt> joined\_recent\_order\_date = filter rowOrder by

orders::customer\_id ==distinct\_order\_date::customer\_id and

orders::order\_id == distinct\_order\_date::order\_id and

orders::order\_datetime == distinct\_order\_date::order\_datetime;

1. Project the required fields from joined\_recent\_order\_date:

grunt> max\_orders\_result = FOREACH joined\_recent\_order\_date GENERATE

orders::customer\_id,

orders::order\_id ,

orders::order\_datetime;

## Find the oldest order date for each customer

1. Get the oldest order date using the min function:

grunt> old\_order\_date = FOREACH records\_group GENERATE

flatten(orders.customer\_id) AS customer\_id,

flatten(orders.order\_id) as order\_id,

MIN(orders.order\_datetime) AS order\_datetime;

1. Collect distinct records from old\_order\_date:

grunt> dist\_old\_order\_date = distinct old\_order\_date;

1. Join orders and dist\_old\_order\_date by customer\_id,order\_id,order\_datetime:

grunt> min\_order\_list = join orders by (customer\_id,order\_id,order\_datetime),

dist\_old\_order\_date by (customer\_id,order\_id,order\_datetime);

1. Filter the records in min\_order\_list:

grunt> joined\_old\_order\_date = filter min\_order\_list by

orders::customer\_id == dist\_old\_order\_date::customer\_id and

orders::order\_id == dist\_old\_order\_date::order\_id and

orders::order\_datetime == dist\_old\_order\_date::order\_datetime;

1. Project the needed fields from joined\_old\_order\_date:

grunt> min\_orders\_result = FOREACH joined\_old\_order\_date GENERATE

orders::customer\_id,

orders::order\_id ,

orders::order\_datetime;

## Join Old Date, Latest Date, and OrderId for every Customer

1. Join max\_orders\_result and min\_orders\_result:

grunt> result = join min\_orders\_result by customer\_id,

max\_orders\_result by customer\_id;

1. Get the first ten records:

grunt> firstten = limit result 10;

The limit function returns no more than 10 records.

1. Use dump to display the records:

grunt> dump firstten;

137 8228753927 2010-10-02 09:26:40 137 6952760836 2010-10-10 23:46:16

274 8228753207 2010-10-02 06:49:05 274 8038062167 2010-10-14 09:17:33

411 8228659208 2010-10-02 02:45:08 411 6326675610 2010-10-11 11:32:28

548 6734479225 2010-10-01 08:31:08 548 6953064348 2010-10-10 19:20:25

1096 6734568190 2010-10-01 21:15:03 1096 8181753531 2010-10-07 04:04:26

1370 6734388086 2010-10-01 02:08:12 1370 7412417661 2010-10-12 23:46:44

1507 8456649021 2010-10-03 04:47:50 1507 7412451029 2010-10-12 07:37:18

1644 7136614975 2010-10-04 09:03:40 1644 8038062935 2010-10-14 17:27:29

2055 7570913900 2010-10-08 23:29:35 2055 4877101631 2010-10-13 21:12:05

2192 7136693581 2010-10-04 19:48:16 2192 8037933831 2010-10-14 12:35:21

1. Save the output to a file using store:

grunt> store firstten into '/user/gpadmin/output2/';

grunt> fs -cat /user/gpadmin/output2/part\*;

137 8228753927 2010-10-02 09:26:40 137 6952760836 2010-10-10 23:46:16

274 8228753207 2010-10-02 06:49:05 274 8038062167 2010-10-14 09:17:33

411 8228659208 2010-10-02 02:45:08 411 6326675610 2010-10-11 11:32:28

548 6734479225 2010-10-01 08:31:08 548 6953064348 2010-10-10 19:20:25

1096 6734568190 2010-10-01 21:15:03 1096 8181753531 2010-10-07 04:04:26

1370 6734388086 2010-10-01 02:08:12 1370 7412417661 2010-10-12 23:46:44

1507 8456649021 2010-10-03 04:47:50 1507 7412451029 2010-10-12 07:37:18

1644 7136614975 2010-10-04 09:03:40 1644 8038062935 2010-10-14 17:27:29

2055 7570913900 2010-10-08 23:29:35 2055 4877101631 2010-10-13 21:12:05

2192 7136693581 2010-10-04 19:48:16 2192 8037933831 2010-10-14 12:35:21