1. Sales Order Shipment by Month and Category Code

Write an SQL statement to display the sum of the extended cost and the sum of the quantity. The results should include data for shipments (transaction type 5) in calendar year 2011. Summarize the result by calendar month and Address Category Code 1. The result should include the grouped columns and the full totals for every combination of grouped columns. Do not use the GROUPING SETS and UNION operators.

Answer  
select sum(extcost) as SumOfExtendedCost,sum(quantity) as SumOfQuantity,calmonth,addrcatcode1 from inventory\_fact,trans\_type\_dim,cust\_vendor\_dim,date\_dim

where inventory\_fact.custvendorkey=cust\_vendor\_dim.custvendorkey

and inventory\_fact.datekey=date\_dim.datekey

and date\_dim.calyear='2011' and trans\_type\_dim.transtypekey=5

group by cube (calmonth,addrcatcode1);

2. Sales Order Shipment by Name, Zip and Quarter  
  
Write an SQL statement to display the sum of the extended cost and the number of inventory transactions. The results should include data for shipments (transaction type 5) in calendar years 2011 and 2012. Summarize the result by calendar quarter, customer zip code, and customer name. The result should include the grouped columns and full set of subtotals for every combination of grouped columns. Do not use the CUBE and UNION operators.

Answer  
select sum(extcost),count(\*),CalQuarter,Zip,Name

from inventory\_fact inv, cust\_vendor\_dim cvd, trans\_type\_dim ttd, date\_dim dd

where inv.CUSTVENDORKEY=cvd.CUSTVENDORKEY

AND inv.datekey=dd.datekey

AND ttd.TRANSTYPEKEY = 5 AND dd.CALYEAR between '2011' and '2012'

group by grouping sets

(

(CalQuarter,Zip,Name),

(CalQuarter,Zip),

(Zip,Name),

(CalQuarter,Name),

(CalQuarter),

(Zip),

(Name),

()

);

3. Display sum of extended cost and number of inventory transactions   
  
Write an SQL statement to display the sum of the extended cost and the number of inventory transactions. The results should include data for shipments (transaction type 5) in calendar years 2011 and 2012. Summarize the result by calendar year, calendar quarter, and customer name. The result should show the grouped columns and the normal set of group by results plus partial subtotals for year and quarter concatenated with customer name. Do not use the GROUPING SETS and UNION operators.  
  
Answer

select sum(extcost) as SumSales,count(\*) as count,CalYear,CalQuarter,Name

from inventory\_fact inv, cust\_vendor\_dim cvd, date\_dim dd

where inv.CUSTVENDORKEY=cvd.CUSTVENDORKEY

AND inv.datekey=dd.datekey

AND TRANSTYPEKEY = 5 AND dd.CALYEAR between '2011' and '2012'

group by Name, rollup (CalYear,CalQuarter)

order by calyear,calquarter,name;

4. Rewrite Query 1 without CUBE, ROLLUP or GROUPING SETS  
  
Rewrite query 1 without the usage of the CUBE ROLLUP, or GROUPING SETS operators. In rewriting the query, you should use NULL as the default value for each column.  
  
Answer

select sum(extcost) as SumOfExtendedCost,sum(quantity) as SumOfQuantity,calmonth,addrcatcode1

from inventory\_fact,cust\_vendor\_dim,date\_dim

where inventory\_fact.custvendorkey=cust\_vendor\_dim.custvendorkey

and inventory\_fact.datekey=date\_dim.datekey

and date\_dim.calyear='2011' and inventory\_fact.transtypekey=5

group by (calmonth,addrcatcode1)

UNION

select sum(extcost) as SumOfExtendedCost,sum(quantity) as SumOfQuantity,calmonth,NULL

from inventory\_fact,cust\_vendor\_dim,date\_dim

where inventory\_fact.custvendorkey=cust\_vendor\_dim.custvendorkey

and inventory\_fact.datekey=date\_dim.datekey

and date\_dim.calyear='2011' and inventory\_fact.transtypekey=5

group by (calmonth)

UNION

select sum(extcost) as SumOfExtendedCost,sum(quantity) as SumOfQuantity,NULL,addrcatcode1

from inventory\_fact,cust\_vendor\_dim,date\_dim

where inventory\_fact.custvendorkey=cust\_vendor\_dim.custvendorkey

and inventory\_fact.datekey=date\_dim.datekey

and date\_dim.calyear='2011' and inventory\_fact.transtypekey=5

group by (addrcatcode1);

5. Sales Order Shipments with Subtotals by Name and Partial Subtotals by Year and Quarter  
  
Write an SQL statement to display the sum of the extended cost and the number of inventory transactions. The results should include data for shipments (transaction type 5) in calendar years 2011 and 2012. Summarize the result by calendar year, calendar quarter, and customer name. The result should include the grouped columns and subtotals for customer name along with partial subtotals for year and quarter. Do not include the normal GROUP BY totals in the result. Do not use the UNION operator.  
  
Answer

select name,calyear,calquarter,sum(extcost)as SumSales, count(\*) as Count

from cust\_vendor\_dim cst,date\_dim dm,inventory\_fact inv

where dm.calyear in (2011,2012)

and inv.datekey=dm.datekey

and inv.custvendorkey=cst.custvendorkey

group by grouping sets(name,

rollup(calyear,calquarter))

order by name,calyear,calquarter;

Analytical Functions

1. Ranking within entire result

Use the RANK function to rank customers in descending order by the sum of extended cost for shipments (transaction type 5). You should use the entire result as a single partition. The result should include the customer name, sum of the extended cost, and rank.

Answer

select Name,sum(extcost), rank() over (order by sum(extcost) desc) rankcost

from inventory\_fact inv, cust\_vendor\_dim cvd

where inv.CUSTVENDORKEY=cvd.CUSTVENDORKEY

AND inv.TRANSTYPEKEY = 5

group by Name;

1. Ranking within a partition

Use the RANK function to rank customers in descending order by the sum of extended cost for shipments (transaction type 5). You should partition the rank values by customer state. The result should include the customer state, customer name, sum of the extended cost, and rank. You should order the result by customer state.  
  
Answer

select state,Name,sum(extcost), rank() over (partition by state order by sum(extcost) desc) rankCost

from inventory\_fact inv, cust\_vendor\_dim cvd

where inv.CUSTVENDORKEY=cvd.CUSTVENDORKEY

AND inv.TRANSTYPEKEY = 5

group by state,name

order by state;

1. Ranking and Dense Ranking within the entire result

Use both RANK and DENSE\_RANK functions to rank customers in descending order by the count of inventory transactions for shipments (transaction type 5). You should use the entire result as a single partition. The result should include the customer name, count of transactions, rank, and dense rank.  
  
Answer

select c.name, count(\*) as count\_transaction,

rank() over(order by count(\*) desc) as rank\_transaction,

dense\_rank() over(order by count(\*) desc) as denserank\_transaction

from cust\_vendor\_dim c, inventory\_fact i

where i.transtypekey=5

and c.custvendorkey=i.custvendorkey

group by c.name;

4 Cumulative extended costs for entire result  
  
Calculate the cumulative sum of extended cost ordered by customer zip code, calendar year, and calendar month for shipments (transaction type 5). The result should include the customer zip code, calendar year, calendar month, sum of the extended cost, and cumulative sum of the extended cost. Note that the cumulative extended cost is the sum of the extended cost in the current row plus the cumulative sum of extended costs in all previous rows.  
  
Answer

select sum(sum(extcost)) over(order by Zip,CALYEAR,calmonth ROWS UNBOUNDED PRECEDING) cumcost, zip, CalYear, Calmonth, sum(extcost)

from inventory\_fact inv, cust\_vendor\_dim cvd, date\_dim dd

where inv.CUSTVENDORKEY=cvd.CUSTVENDORKEY

AND inv.datekey=dd.datekey

AND TRANSTYPEKEY = 5

group by zip, calyear,calmonth;

5 Cumulative extended cost for entire result  
  
Calculate the cumulative sum of extended cost ordered by customer zip code, calendar year, and calendar month for shipments (transaction type 5). Restart the cumulative extended cost after each combination of zip code and calendar year. The result should include the customer zip code, calendar year, calendar month, sum of the extended cost, and cumulative sum of the extended cost. Note that the cumulative extended cost is the sum of the extended cost in the current row plus the cumulative sum of extended costs in all previous rows of the store zip code and years. The value of cumulative extended cost resets in each partition (new value for zip code and year).  
  
Answer

select sum(sum(extcost)) over(order by Zip,CALYEAR,calmonth ROWS UNBOUNDED PRECEDING) cumcost, zip,

CalYear, Calmonth, sum(extcost) as SumExtCost

from inventory\_fact inv, cust\_vendor\_dim cvd, date\_dim dd

where inv.CUSTVENDORKEY=cvd.CUSTVENDORKEY

AND inv.datekey=dd.datekey

AND TRANSTYPEKEY = 5

group by zip, calyear,calmonth;

Materialized view

1. Create materialized view for 2011 Shipments

 The result should contain the sum of the extended cost, the sum of the quantity, and the count of inventory transactions.

 These calculated amounts should be summarized by the customer vendor key and the date key.

 The result should include only sales shipment transactions (transaction type 5) for the year 2011.  
 The materialized view should not contain subtotals that are created by the CUBE and the ROLLUP keywords.

 To make the peer assessment easier, you should name your materialized view “SalesByVendorDateKeyMV2011”.

Answer

CREATE MATERIALIZED VIEW SalesByVendorDateKeyMV2011 as

Select sum(extcost), sum(quantity), count(\*)

from inventory\_fact inv, cust\_vendor\_dim cvd, date\_dim dd

where inv.CUSTVENDORKEY=cvd.CUSTVENDORKEY

AND inv.datekey=dd.datekey

AND inv.TRANSTYPEKEY = 5 AND dd.CALYEAR = '2011'

group by cvd.custvendorkey,dd.datekey;

1. Create materialized view for 2012 Shipments

 The result should contain the sum of the extended cost, the sum of the quantity, and the count of inventory transactions.

 These calculated amounts should be summarized by the customer vendor key and the date key.

 The result should include only sales shipment transactions (transaction type 5) for the year 2012.

 The materialized view should not contain subtotals that are created by the CUBE and the ROLLUP keywords.

 To make the peer assessment easier, you should name your materialized view “SalesByVendorDateKeyMV2012”.  
  
Answer

CREATE MATERIALIZED VIEW SalesByVendorDateKeyMV2011 as

Select sum(extcost), sum(quantity), count(\*)

from inventory\_fact inv, cust\_vendor\_dim cvd, date\_dim dd

where inv.CUSTVENDORKEY=cvd.CUSTVENDORKEY

AND inv.datekey=dd.datekey

AND inv.TRANSTYPEKEY = 5 AND dd.CALYEAR = '2012'

group by cvd.custvendorkey,dd.datekey;