Truett bloxsom

Tsb962

HW2

Q1 -- 1. Find the top 5 products for the year 2013 by looking at their actual sales?

Q1 Query:

select sum(sales), product, EXTRACT(YEAR FROM factdate)

from factcoffee, prodcoffee

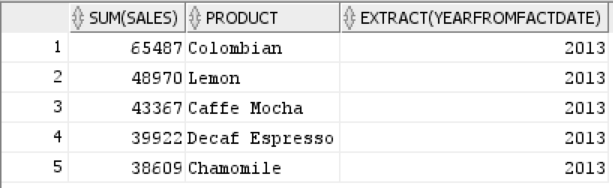
where factcoffee.productid = prodcoffee.productid and EXTRACT(YEAR FROM factdate) = 2013

group by product, EXTRACT(YEAR FROM factdate)

order by sum(sales) desc

FETCH NEXT 5 ROWS ONLY;

Q1 output:



Q2 -- 2. Find all the area codes and associated state names for which the actual product sales was at least 200% more than the budgeted sales and the budgeted sales was at least 100, during the year 2012.

-- So my logic is as follows: I want to sum both sales and budget sales for each areacode and associated state for the year 2012. I am assuming that you want to compare the sum budget sale for each particular areacode with the corresponding sum sales of that areacode. the reason why there is no output is because the sales and budget sales for each areacode are quite similar to each other in almost all area codes which can be shown if you do not run the HAVING line.

Q2 Query:

select areacoffee.areacode, statecoffee.state, sum(sales), sum(budgetsales)

from areacoffee, factcoffee, statecoffee

where areacoffee.stateid = statecoffee.stateid

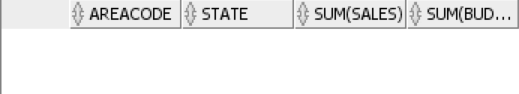
and areacoffee.areacode = factcoffee.areacode

and EXTRACT(YEAR FROM factdate) = 2012

group by areacoffee.areacode, statecoffee.state

having sum(budgetsales) >= 100 and sum(sales) >= 2 \* sum(budgetsales);

Q2 output:



Q3 -- 3. Find all the months during the year 2012 for which the actual product sales are three times greater than the average monthly sales of all products for the year 2012.

-- my logic is as follows: I want to compare each monthly sales with the overall average monthly sales for the year 2012. subquery A gets me the sum product sales for each month, and subquery B gets me the average of all the months. Again, if you run the query without the WHERE clause, you will see that the product sales for each month does not differ that much month to month so thats why there is no output with the WHERE clause. This makes sense since coffee is not a seasonal product and so month to month the sales will not fluctuate that much.

Q3 Query:

select A.allmonths, A.prodsales, B.monthsales

from (select sum(sales) prodsales, extract(month from factdate) allmonths

from factcoffee

where extract(year from factdate) = 2012

group by extract(month from factdate) )A,

(select avg(sum(sales)) monthsales

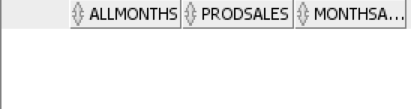
from factcoffee

where extract(year from factdate) = 2012

group by extract(month from factdate) )B

where A.prodsales > 3 \* B.monthsales;

Q3 output:



Q4 -- 4. For the year 2013, find all the area codes for which the annual difference between the actual and budgeted sales is greater than the average of the annual sales difference between the actual and budgeted sales

Q4 Query:

select A.areacode , A.actualsales, A.actualbudgets, B.avgsales, B.avgbudgets

from (select sum(sales) actualsales, sum(budgetsales) actualbudgets, areacode

from factcoffee

where extract(year from factdate) = 2013

group by areacode) A,

(select avg(sum(sales)) avgsales, avg(sum(budgetsales)) avgbudgets

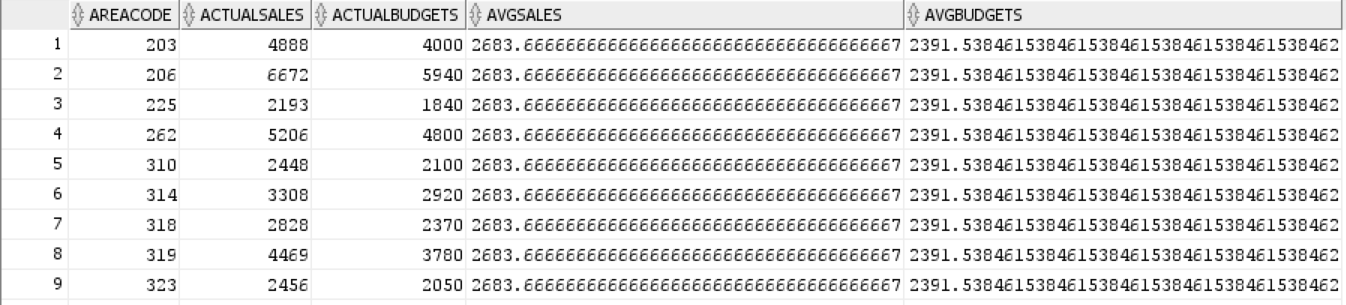
from factcoffee

where extract(year from factdate) = 2013

group by areacode) B

where A.actualsales - A.actualbudgets > (B.avgsales - B.avgbudgets);

Q4 output:



Q5 -- 5. Identify the top markets for each of the product in terms of sales for 2012. Note: Market is a specific column in one of the tables. Do not confuse it with States or StateID. As one example, you are trying to find out Caffe Latte was most popular in which market in 2012, where popularity is measured in terms of actual sales.

Q5 Query:

select \*

from (

with sum\_sales as (

select pc.product, sc.market, sum(sales) as sum\_sales

from factcoffee fc inner join areacoffee ac on fc.areacode = ac.areacode

inner join prodcoffee pc on fc.productid = pc.productid

inner join statecoffee sc on ac.stateid = sc.stateid

where extract(year from fc.factdate) = 2012

group by pc.product, sc.market

order by pc.product

)

select \*

from (

select product, max(sum\_sales) as max\_sales

from sum\_sales

group by product

) x

inner join sum\_sales on sum\_sales.sum\_sales = x.max\_sales

and sum\_sales.product = x.product

) y;

Q5 output:

