Assignment 3 Jushira Thelakkat (jt39634)

# PART A – Coffee Sales

The following questions are based on the Coffee store sales data. Answer the following queries.

* 1. **Extract the total sales for each product for each month. List all months (like January, February, etc) in the columns.**

**Solution:**

SELECT P.productid,

SUM(CASE WHEN extract(month from F.factdate)=01 then F.actsales else 0 end) as Jan,

SUM(CASE WHEN extract(month from F.factdate)=02 then F.actsales else 0 end) as Feb,

SUM(CASE WHEN extract(month from F.factdate)=03 then F.actsales else 0 end) as March,

SUM(CASE WHEN extract(month from F.factdate)=04 then F.actsales else 0 end) as April,

SUM(CASE WHEN extract(month from F.factdate)=05 then F.actsales else 0 end) as May,

SUM(CASE WHEN extract(month from F.factdate)=06 then F.actsales else 0 end) as June,

SUM(CASE WHEN extract(month from F.factdate)=07 then F.actsales else 0 end) as July,

SUM(CASE WHEN extract(month from F.factdate)=08 then F.actsales else 0 end) as Aug,

SUM(CASE WHEN extract(month from F.factdate)=09 then F.actsales else 0 end) as Sept,

SUM(CASE WHEN extract(month from F.factdate)=10 then F.actsales else 0 end) as Oct,

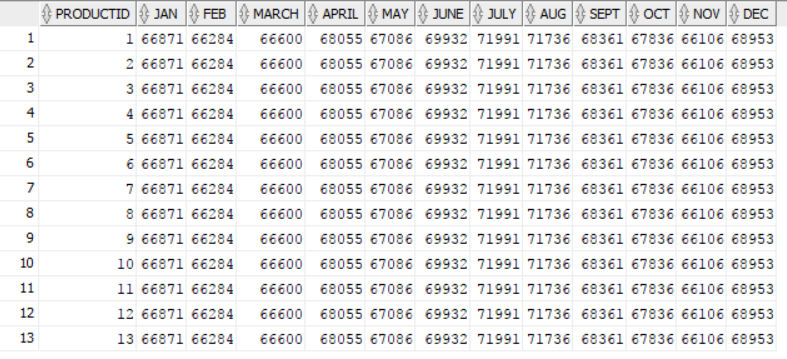
SUM(CASE WHEN extract(month from F.factdate)=11 then F.actsales else 0 end) as Nov,

SUM(CASE WHEN extract(month from F.factdate)=12 then F.actsales else 0 end) as Dec

FROM factcoffee F , PRODCOFFEE P

GROUP BY P.productid

ORDER BY P.productid;



**Insights:** Using CASE WHEN command, I am extracting sales for each of the months and grouping by the product id.

* 1. **In each state, identify the product with greatest sales for the year 2012.**

**Solution:**

CREATE or replace View firstprod AS(

SELECT \* FROM(

SELECT statename, productid,tot\_sale,row\_number() over(partition by statename order by tot\_sale desc) as rankid

FROM(

SELECT statename, productid,sum(actsales) as tot\_sale

FROM factcoffee, areacode, states

WHERE areacode.areaid = factcoffee.areaid and states.stateid = areacode.STATEID and EXTRACT(YEAR FROM factdate) in '2012'

GROUP BY statename, productid))

WHERE rankid = 1);

CREATE or replace View Secprod AS(

SELECT \* FROM(

SELECT statename, productid,tot\_sale,row\_number() over(partition by statename order by tot\_sale desc) as rankid

FROM(

SELECT statename, productid,sum(actsales) as tot\_sale

FROM factcoffee, areacode, states

WHERE areacode.areaid = factcoffee.areaid and states.stateid = areacode.STATEID and EXTRACT(YEAR FROM factdate) in '2013'

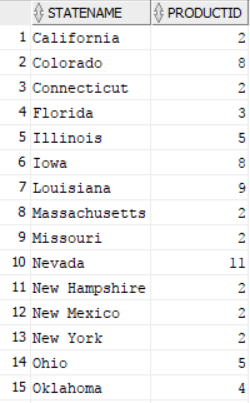
GROUP BY statename, productid))

WHERE rankid = 1);

* + 1. **Identify the states where the best selling product remained the same in 2013 (compared to best selling product in 2012)**

SELECT firstprod.statename, firstprod.productid

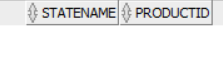
FROM firstprod JOIN Secprod ON firstprod.statename = secprod.statename and firstprod.productid = secprod.productid;



* + 1. **Identify the states where the bestselling product has changed.**

SELECT firstprod.statename, firstprod.productid

FROM firstprod JOIN Secprod ON firstprod.statename = secprod.statename and firstprod.productid <> secprod.productid;



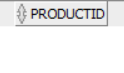
**Insights:** We get an empty list when we try to find the states where the best selling product has changed.

* + 1. **Identify the products that were best in 2012 but not in 2013.**

SELECT firstprod.productid

FROM firstprod

WHERE firstprod.productid NOT IN (SELECT secprod.productid FROM secprod where firstprod.statename = secprod.statename);



**Insights:** We again get an empty set when we try to find the products that were best in 2012 but not in 2013.

* 1. **Identify the top two best selling products that are common to both 2012 and 2013.**

**Solution:**

CREATE or replace View firstprod AS(

SELECT \* FROM(

SELECT productid, sum(actsales)

FROM factcoffee

WHERE EXTRACT(YEAR FROM factdate) in '2012'

GROUP BY productid

ORDER BY sum(actsales) desc)

WHERE rownum <= 2);

CREATE or replace View secprod AS(

SELECT \* FROM(

SELECT productid, sum(actsales)

FROM factcoffee

WHERE EXTRACT(YEAR FROM factdate) in '2013'

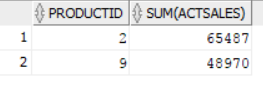
GROUP BY productid

ORDER BY sum(actsales) desc)

WHERE rownum <= 2);

select \* from firstprod;

select \* from secprod;



**Insights:** We get the same two products as best selling in both 2012 and 2013

* 1. **What fraction of the top selling states contributes to at least 50% of the total sales? Do they also contribute to 50% of the profit share as well? (Please note that you won’t likely get exact 50% when you do your analysis)**

**Solution:**

WITH Q1 as(

SELECT sum(actsales) as tot\_sales

FROM factcoffee),

Q2 as(

SELECT statename,

sum(actsales) as act\_sales

FROM factcoffee, areacode, states

WHERE areacode.areaid = factcoffee.areaid and states.stateid = areacode.STATEID

group by statename)

select \* from (

select statename, sum(act\_sales) over (order by act\_sales desc) as cumulative\_Sales,

sum(act\_sales) over (order by act\_sales desc)/tot\_sales as cumulative\_rank

from Q2 ,Q1)

where ROWNUM <= (SELECT min(case when cumulative\_rank>0.5 then ROWNUM end)from(

select statename, sum(act\_sales) over (order by act\_sales desc) as cumulative\_Sales,

sum(act\_sales) over (order by act\_sales desc)/tot\_sales as cumulative\_rank

from Q2 ,Q1))

;

-- Profit

WITH Q1 as(

SELECT sum(actprofit) as tot\_profit

FROM factcoffee),

Q2 as(

SELECT statename,

sum(actprofit) as act\_profit

FROM factcoffee, areacode, states

WHERE areacode.areaid = factcoffee.areaid and states.stateid = areacode.STATEID

group by statename)

select \* from (

select statename, sum(act\_profit) over (order by act\_profit desc) as cumulative\_profit,

sum(act\_profit) over (order by act\_profit desc)/tot\_profit as cumulative\_rank

from Q2 ,Q1)

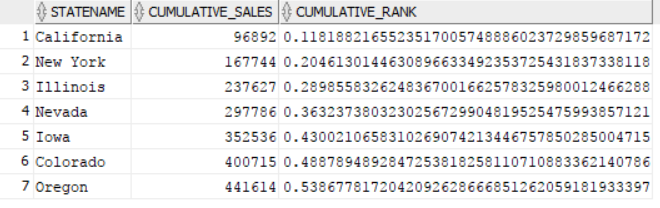
where ROWNUM <= (SELECT min(case when cumulative\_rank>0.5 then ROWNUM end)from(

select statename, sum(act\_profit) over (order by act\_profit desc) as cumulative\_profit,

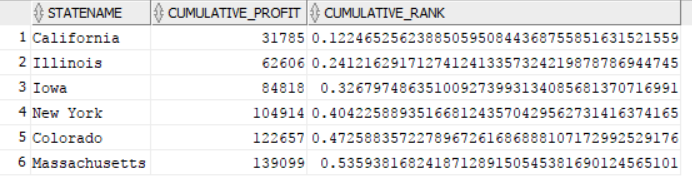
sum(act\_profit) over (order by act\_profit desc)/tot\_profit as cumulative\_rank

from Q2 ,Q1))

;



**Insights:** Theseseven statesaccount for 50% of total sales.



**Insights:** Thesesix statesaccount for 50% of total profits.

* 1. **If you have to discontinue some product, which one would you suggest and why? Formulate your questions.**

**Solution:**

WITH Q1 AS (

Select productid, Extract(Year from factdate) as Years, sum(actprofit) as TotProfits

FROM factcoffee

GROUP BY productid, Extract(Year from factdate)),

Q2 AS (SELECT \* FROM Q1

PIVOT (

SUm(TotProfits)

FOR Years IN (2012 as FYear,2013 as SYear)

))

select \* from(

SELECT productid, Fyear, SYear, ROUND(100\*(Syear-Fyear)/abs(Fyear),2) as PerInc

FROM Q2

WHERE Fyear <> 0

ORDER BY PerInc )

where rownum <=5;

--

WITH Q1 AS (

Select productid, Extract(Year from factdate) as Years, sum(actsales) as Totsales

FROM factcoffee

GROUP BY productid, Extract(Year from factdate)),

Q2 AS (SELECT \* FROM Q1

PIVOT (

SUm(Totsales)

FOR Years IN (2012 as FYear,2013 as SYear)

))

select \* from(

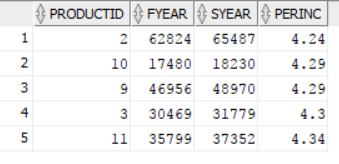
SELECT productid, Fyear, SYear, ROUND(100\*(Syear-Fyear)/abs(Fyear),2) as PerInc

FROM Q2

WHERE Fyear <> 0

ORDER BY PerInc )

where rownum <=5;



**Insights:** I would discontinue products which have lease percentage increase in sales from 2012 to 2013. The five products with least increase in sales are shown in the table above.

* 1. **The overall sales per month could be seasonal. That is, you will find sales in some months greater than the others and this may be consistent in both 2012 and 2013. Identify if there are seasonal trends. Plot month vs. sales for each year.**
     1. **Are there trends for any particular product?**
     2. **Are there trends in any particular state for any product?**

**Solution:**

SELECT P.productid, Extract(Year from F.factdate) as Years,

SUM(CASE WHEN extract(month from factdate)=01 then actsales else 0 end) as Jan,

SUM(CASE WHEN extract(month from factdate)=02 then actsales else 0 end) as Feb,

SUM(CASE WHEN extract(month from factdate)=03 then actsales else 0 end) as March,

SUM(CASE WHEN extract(month from factdate)=04 then actsales else 0 end) as April,

SUM(CASE WHEN extract(month from factdate)=05 then actsales else 0 end) as May,

SUM(CASE WHEN extract(month from factdate)=06 then actsales else 0 end) as June,

SUM(CASE WHEN extract(month from factdate)=07 then actsales else 0 end) as July,

SUM(CASE WHEN extract(month from factdate)=08 then actsales else 0 end) as Aug,

SUM(CASE WHEN extract(month from factdate)=09 then actsales else 0 end) as Sept,

SUM(CASE WHEN extract(month from factdate)=10 then actsales else 0 end) as Oct,

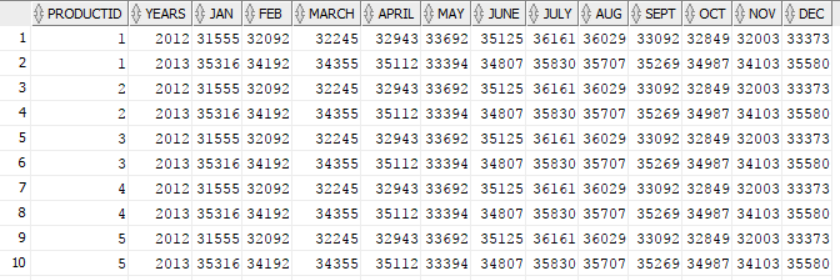
SUM(CASE WHEN extract(month from factdate)=11 then actsales else 0 end) as Nov,

SUM(CASE WHEN extract(month from factdate)=12 then actsales else 0 end) as Dec

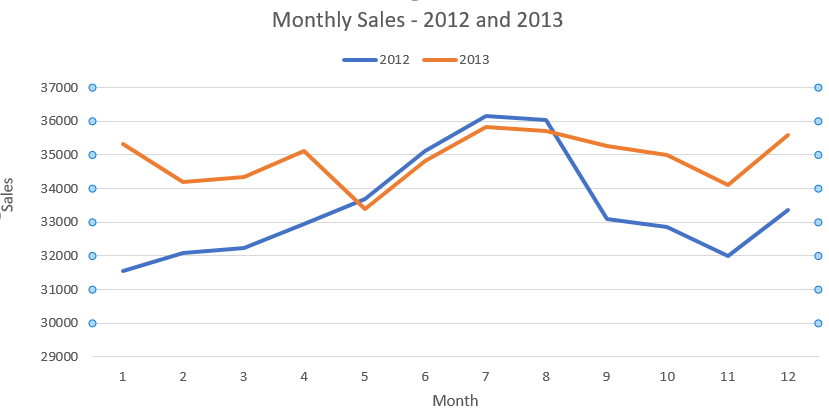
FROM factcoffee F, prodcoffee P

GROUP BY P.productid, Extract(Year from F.factdate)

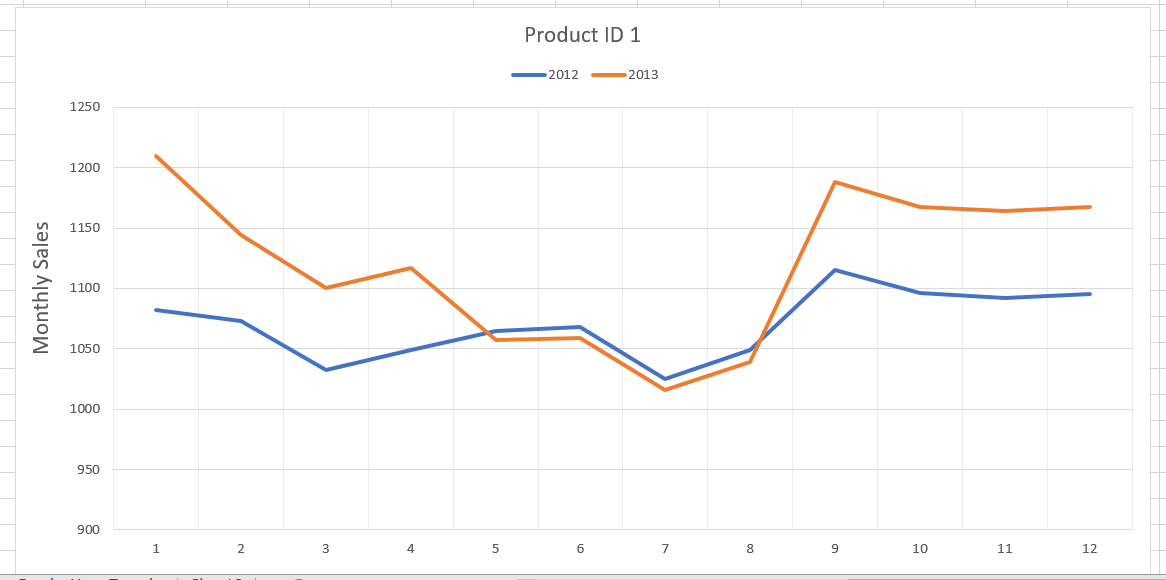
ORDER BY P.productid;



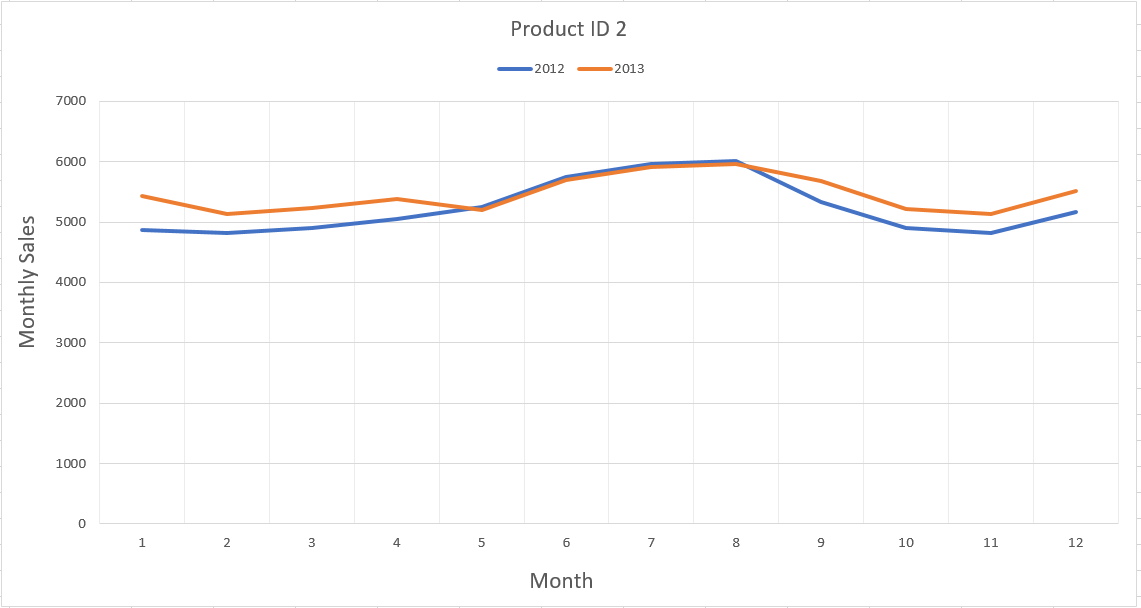
|  |  |  |
| --- | --- | --- |
| **MONTH** | **2012** | **2013** |
| 1 | 31555 | 35316 |
| 2 | 32092 | 34192 |
| 3 | 32245 | 34355 |
| 4 | 32943 | 35112 |
| 5 | 33692 | 33394 |
| 6 | 35125 | 34807 |
| 7 | 36161 | 35830 |
| 8 | 36029 | 35707 |
| 9 | 33092 | 35269 |
| 10 | 32849 | 34987 |
| 11 | 32003 | 34103 |
| 12 | 33373 | 35580 |
|  |  |  |



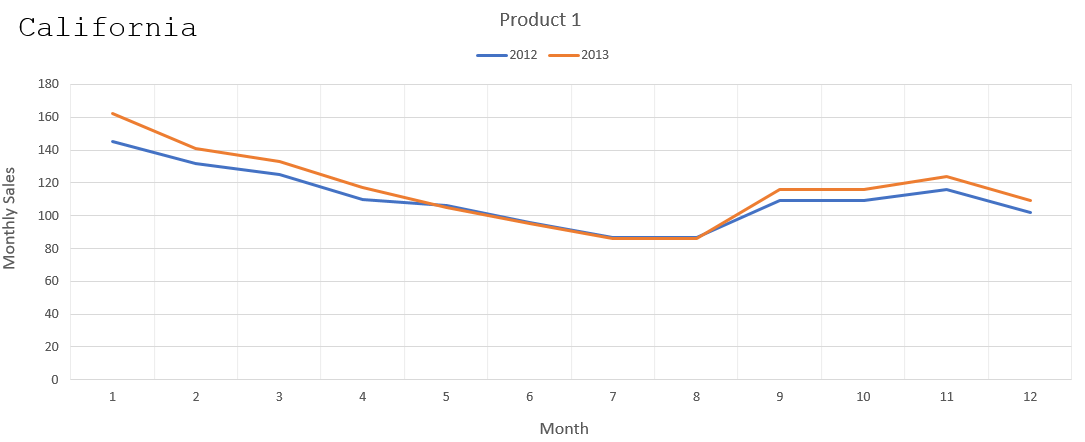
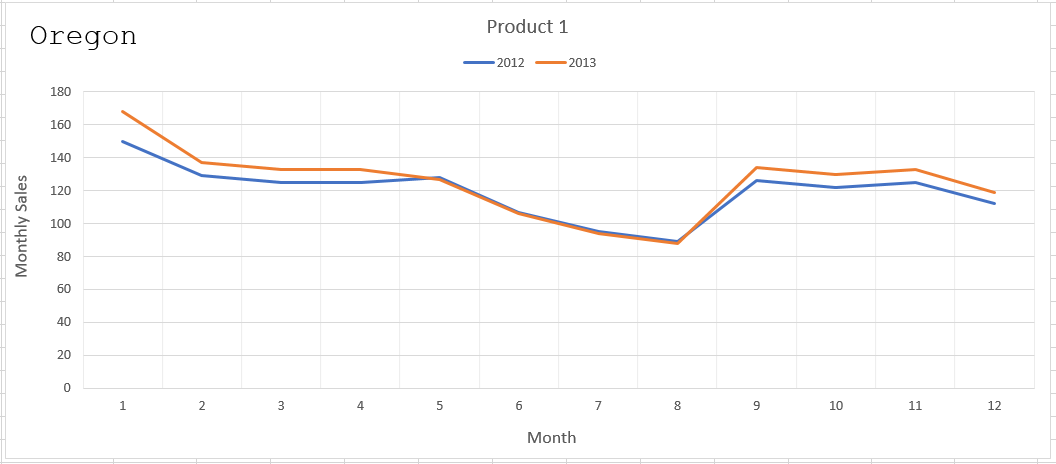
**Insights:** Looking at monthly sales for 2012 and 2013, we see that sales follow similar trends from May to August and then there is a drop in sales of 2012 while sales of 2013 maintain to be at a higher value.

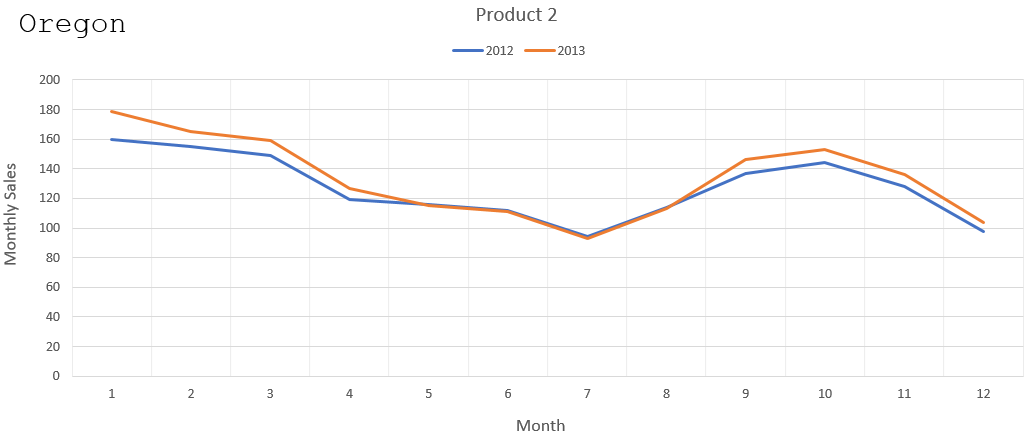


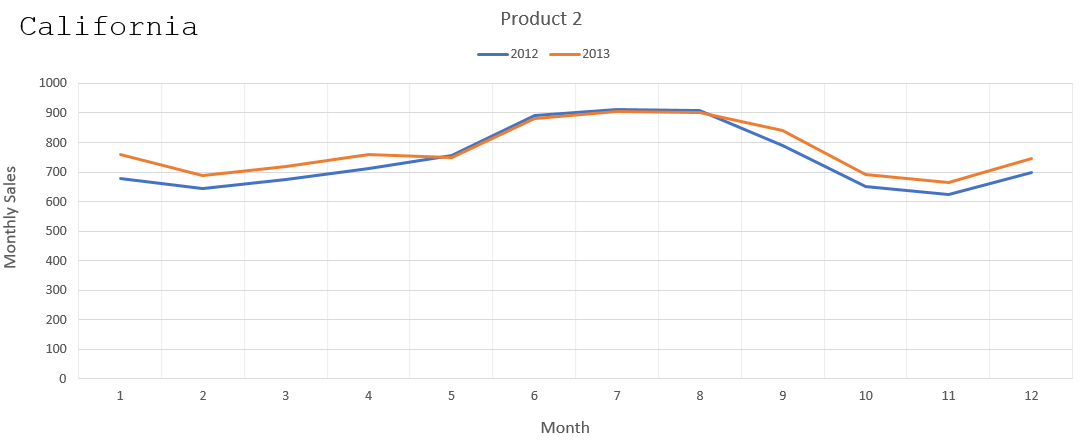
**Insights:** For Product 1, sales show same pattern for months May to August and there is an increase in sales from mid August to mid September for the year 2013 and then the sales again follow the same trend from October for both the years.



**Insights:** For product 2, the sales mostly show similar trends throughout both the years of 2012 and 2013.







**Insights:** Choosing two states for further analysis, Oregon and Washington, we see that within a state, for the same product, sales pretty much follow similar trends for both 2012 and 2013.

* 1. **Insert a new column into Factcoffee table called Quarter. Now depending on the month, update the quarter number as Q1, Q2, Q3, or Q4 for each row.** 
     1. **Now find the total sales for years 2012 and 2013 for each quarter. Display quarter in columns.**

**Solution:**

ALTER TABLE Factcoffee

ADD Quarter Char(2);

UPDATE Factcoffee

SET Quarter = 'Q1'

WHERE to\_char(factdate, 'Mon') in ('Jan' , 'Feb','Mar');

UPDATE Factcoffee

SET Quarter = 'Q2'

WHERE to\_char(factdate, 'Mon') in ('Apr' , 'May','Jun');

UPDATE Factcoffee

SET Quarter = 'Q3'

WHERE to\_char(factdate, 'Mon') in ('Jul' , 'Aug','Sep');

UPDATE Factcoffee

SET Quarter = 'Q4'

WHERE to\_char(factdate, 'Mon') in ('Oct' , 'Nov','Dec');

SELECT \* FROM(

SELECT Extract(Year from factdate) as Year, Quarter, sum(actsales) as sum\_sales

FROM factcoffee

GROUP BY Extract(Year from factdate),Quarter

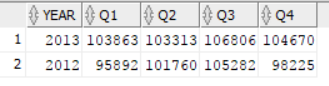
)

PIVOT (

sum(sum\_sales)

FOR Quarter IN ('Q1' as Q1, 'Q2' as Q2, 'Q3' as Q3, 'Q4' as Q4)

);



* + 1. **Which quarter has the greatest sales and profits?**

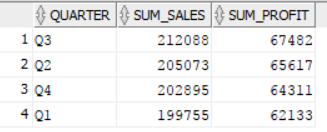
**Solution:**

SELECT Quarter, sum(actsales) as sum\_sales, sum(actprofit) as sum\_profit

FROM factcoffee

GROUP BY Quarter

ORDER BY sum\_sales desc;



* 1. **CREATE a TABLE that captures for each state, product, and quarter combination, the following measures - the total sales, total profits, percentage margin, total marketing expenses, and rank order of sales for each quarter. You may use many different queries to INSERT or UPDATE using a single query or union of many different queries.**

**Solution:**

WITH Q1 as(

SELECT statename, productid, Quarter, sum(actsales) as sum\_sales, sum(actprofit) as sum\_profit,

round(100\*sum(actprofit)/sum(actsales),2) as per\_margin, sum(actmarkcost) as mark\_cost

FROM areacode, factcoffee, states

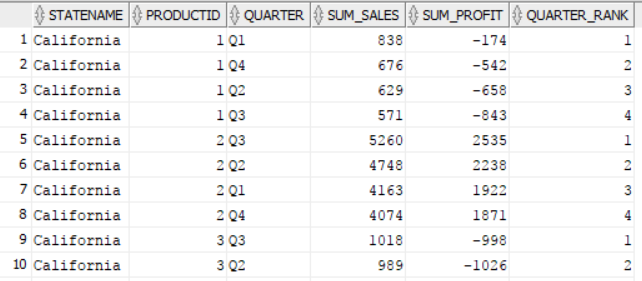
WHERE areacode.areaid = factcoffee.areaid and states.stateid = areacode.STATEID

GROUP BY statename, productid, Quarter)

SELECT statename, productid, Quarter,sum\_sales, sum\_profit,row\_number() over(partition by statename,productid order by sum\_sales desc) as Quarter\_rank

FROM Q1

ORDER BY statename;



PART B: Office Product

1. **Rank managers based on the sales generated.**

**Solution:**

SELECT

m.regmanager,

SUM(d.ordsales) totsales,

ROW\_NUMBER() OVER(

ORDER BY SUM(d.ordsales) DESC

) AS rankid

FROM

managers m,

customers c,

orderdet d

WHERE

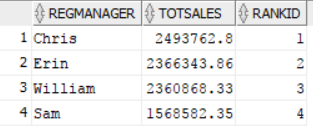
m.regid = c.custreg

AND

c.custid = d.custid

GROUP BY

m.regmanager;



**Insights:** Grouping the total sales (sum of sales) by manger names lets me rank them based on their sales.

Chris stands first.

1. **Find the products that had the worst average shipping times.**

**Solution:**

SELECT

p.prodname,

round(

AVG(d.ordshipdate - d.orddate),

2

) avgshiptime,

ROW\_NUMBER() OVER(

ORDER BY(AVG(d.ordshipdate - d.orddate) ) DESC

) AS rankid

FROM

products p,

orderdet d

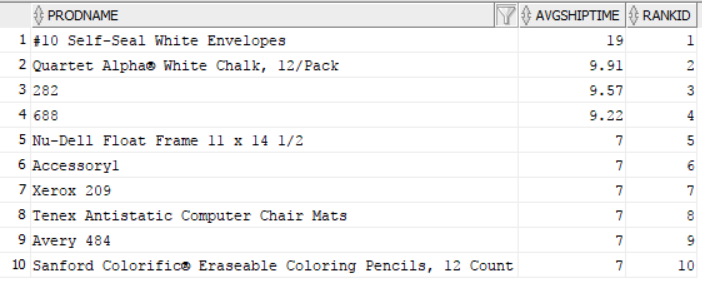
WHERE

p.prodid = d.prodid

GROUP BY

p.prodname

FETCH FIRST 10 ROWS ONLY;



**Insights:** Here I have assumed the average shipping time to be the time between when an order is placed and when the order is shipped. Finding the average and ordering in descending order gives me the highest average shipping times for each of the products.

1. **What fraction of the revenues is generated from the top 10% of the customers?**

**Solution:**

WITH q1 AS (

SELECT

c.custid,

SUM(d.ordsales) sumsales,

ROW\_NUMBER() OVER(

ORDER BY SUM(d.ordsales) DESC

) rowsales

FROM

customers c,

orderdet d

WHERE

c.custid = d.custid

GROUP BY

c.custid

),q2 AS (

SELECT

COUNT(\*) AS totcustomers

FROM

q1

),q3 AS (

SELECT

SUM(q1.sumsales) AS numerator

FROM

q1,

q2

WHERE

ROWNUM <= 0.1 \* totcustomers

),q4 AS (

SELECT

SUM(sumsales) denominator

FROM

q1

) SELECT

q3.numerator,

q4.denominator,

round(

100 \* ( (q3.numerator) / (q4.denominator) ),

2

) fracrevenue

FROM

q3,

q4;



**Insights:**

Here the numerator is the sum of sales from the top 10% of customers contributing to the overall sales. Denominator is the sum of total sales of all the customers. Dividing them gives us the fraction of revenues generated from the top 10% customers.

1. **Are these 10% of the customers also the leaders in the number of orders placed?**

**Solution:**

-- Customers in top 10% of sales that are not in top 10% of orders

WITH q1 AS (

SELECT

c.custid AS custid,

SUM(ordsales) sumsales,

ROW\_NUMBER() OVER(

ORDER BY SUM(ordsales) DESC

) rowsales

FROM

customers c,

orderdet d

WHERE

c.custid = d.custid

GROUP BY

c.custid

),q2 AS (

SELECT

COUNT(\*) AS totcustomers

FROM

q1

),q3 AS (

SELECT

custid,

sumsales,

rowsales

FROM

q1,

q2

WHERE

ROWNUM <= 0.1 \* totcustomers

),q4 AS (

SELECT

c.custid AS custid,

SUM(d.ordqty) sumordqty,

ROW\_NUMBER() OVER(

ORDER BY SUM(ordqty) DESC

) rowqty

FROM

customers c,

orderdet d

WHERE

c.custid = d.custid

GROUP BY

c.custid

),q5 AS (

SELECT

custid,

sumordqty,

rowqty

FROM

q4,

q2

WHERE

ROWNUM <= 0.1 \* totcustomers

) SELECT

\*

FROM

q3

WHERE

q3.custid NOT IN (

SELECT

custid

FROM

q5

WHERE

q3.custid = q5.custid

);



**Insights:** Here we are trying to see if the customers that contribute to top 10% of sales are there in top 10% of orders as well. For this we use a series of queries using WITH.

Using just Q1 and Q2 , we see that there are a total of 269 customers contributing to the top 10% of sales.

Writing similar queries for both sales and orders, we use NOT IN command in our WHERE statement to see what customers are not a part of the top 10%. We get a total of 133 customers that are not in the top 10% of orders. If we run the same using IN command in the WHERE statement, we get 136 as expected. So we can say around half the customers in top 10% of sales are the leaders in orders placed too.

1. **For each city and product combination, list the total sales and rank order in each city by total sales.**

**Solution:**

SELECT

\*

FROM

(

SELECT

c.custcity,

p.prodname,

ROW\_NUMBER() OVER(PARTITION BY

c.custcity

ORDER BY

SUM(d.ordsales)

DESC

) AS rankid,

SUM(d.ordsales) totsales

FROM

products p,

orderdet d,

customers c

WHERE

d.prodid = p.prodid

AND

d.custid = c.custid

GROUP BY

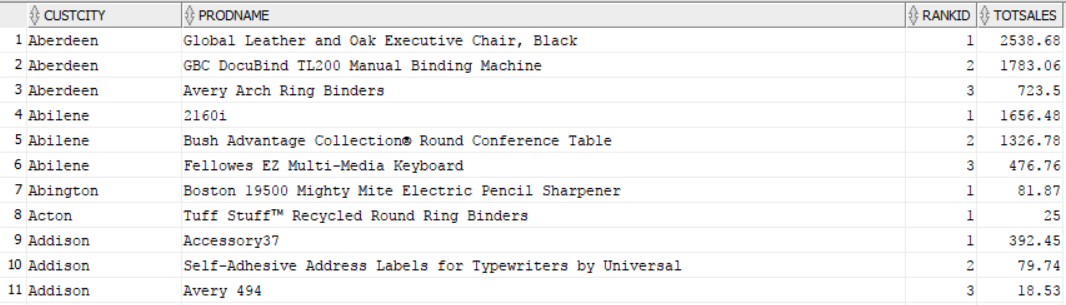
p.prodname,

c.custcity

)

WHERE

rankid <= 3;



**Insights:** Here we just select city and prodname and rank it according to total sales of each of the combination using PARTITION command in our SELECT statement.

1. **Which are the top 5 customers for each of the years?**
   1. **Who are common customers across all years?**
   2. **Are there some customers in any year that are distinct?**

**Solution:**

SELECT

\*

FROM

(

SELECT

EXTRACT(YEAR FROM d.orddate) year,

ROW\_NUMBER() OVER(PARTITION BY

EXTRACT(YEAR FROM d.orddate)

ORDER BY

SUM(d.ordsales)

DESC

) AS rankid,

c.custname customer,

SUM(d.ordsales) totsales

FROM

orderdet d,

customers c

WHERE

d.custid = c.custid

GROUP BY

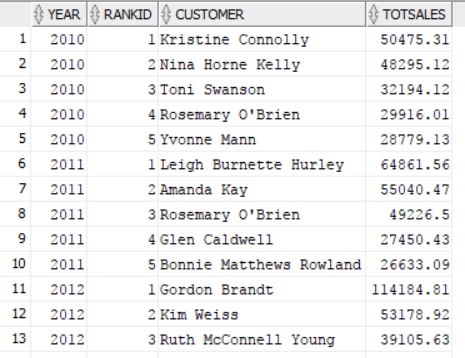
EXTRACT(YEAR FROM d.orddate),

c.custname

)

WHERE

rankid <= 5;



**Insights:** These are the top five customers in each of the years from 2010 to 2013. We got this by just grouping by year and customer name and ordering by tot sales.

--Trial 2

--2010

SELECT

c.custname customer,

SUM(d.ordsales) totsales

FROM

orderdet d,

customers c

WHERE

d.custid = c.custid

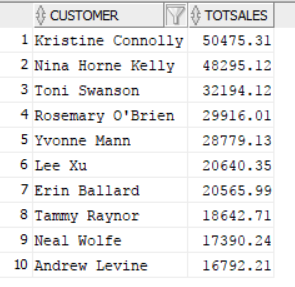
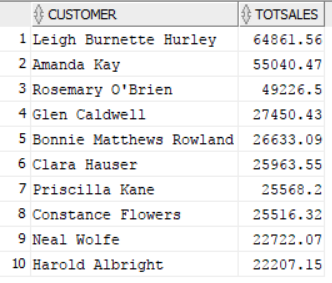
AND

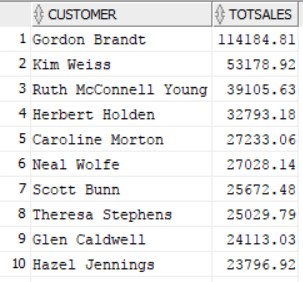
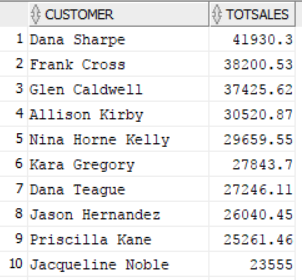
EXTRACT(YEAR FROM d.orddate) = 2010

GROUP BY

c.custname

ORDER BY SUM(d.ordsales) DESC;

**Insights:** 2010 (1124 rows), 2011(1212 rows), 2012(1222 rows), 2013(1584 rows)

There sure are customers that are distinct, especially in 2013 (largest number of customers)

And there are customers that are common for all years too.

If we just see the top five, Rosemary is common in 2010, 2011. Glen is common in 2011 and 2013.

--Trial 3

with q1 as (

SELECT \* FROM

(

SELECT extract(Year from OrdDate) Year, CustName Customer, SUM(OrdSales) TotSales,

ROW\_number() OVER(PARTITION BY extract(Year from OrdDate) Order By sum(OrdSales) DESC) AS RANKID

FROM OrderDet, Customers

WHERE Orderdet.custid = customers.custid

GROUP BY extract(Year from OrdDate), CustName

)

WHERE RANKID <= 5)

select customer,cust\_present\_in\_years from (

select customer,count(distinct year) as cust\_present\_in\_years

from q1

group by customer

order by cust\_present\_in\_years desc)

where cust\_present\_in\_years=4

;

-- or even check using intersect

with q1 as (

SELECT \* FROM

(

SELECT extract(Year from OrdDate) Year, CustName Customer, SUM(OrdSales) TotSales,

ROW\_number() OVER(PARTITION BY extract(Year from OrdDate) Order By sum(OrdSales) DESC) AS RANKID

FROM OrderDet, Customers

WHERE Orderdet.custid = customers.custid

GROUP BY extract(Year from OrdDate), CustName

)

WHERE RANKID <= 5)

select customer from q1 where year = 2010

intersect

select customer from q1 where year = 2011

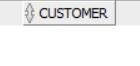
intersect

select customer from q1 where year = 2012

intersect

select customer from q1 where year = 2013

;



**Insights:** Using INTERSECT, if we check common customers for all years, we see that the result is an empty set. So even though there are common customers between 2 years, there is no common customer between all four years.

1. **Find the number of orders in each subcategory in states Michigan and Washington. List Washington and Michigan in different columns.**

**Solution:**

SELECT

\*

FROM

(

SELECT

p.prodsubcat,

c.custstate,

COUNT(d.orderid) numorders

FROM

products p,

orderdet d,

customers c

WHERE

p.prodid = d.prodid

AND

c.custid = d.custid

AND

c.custstate IN (

'Michigan','Washington'

)

GROUP BY

p.prodsubcat,

c.custstate

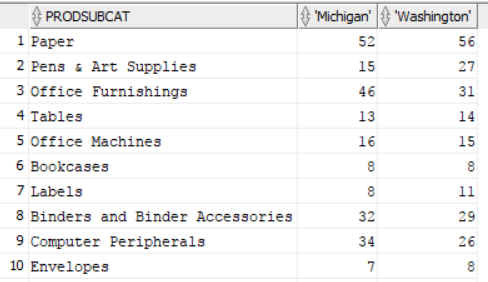
)

PIVOT ( SUM ( numorders )

FOR custstate

IN ( 'Michigan','Washington' )

);



**Insights:** The first ten rows of total number of orders in Michigan and Washington for each of the subcategories.

1. **Find total orders in each quarter.**

**Solution:**

ALTER TABLE orderdet ADD (

quarter VARCHAR2(30)

);

UPDATE orderdet

SET

quarter = 'Q1'

WHERE

EXTRACT(MONTH FROM orddate) IN (

1,2,3

);

UPDATE orderdet

SET

quarter = 'Q2'

WHERE

EXTRACT(MONTH FROM orddate) IN (

4,5,6

);

UPDATE orderdet

SET

quarter = 'Q3'

WHERE

EXTRACT(MONTH FROM orddate) IN (

7,8,9

);

UPDATE orderdet

SET

quarter = 'Q4'

WHERE

EXTRACT(MONTH FROM orddate) IN (

10,11,12

);

SELECT

\*

FROM

orderdet;

SELECT

quarter,

COUNT(orderid) totorders

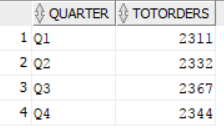
FROM

orderdet

GROUP BY

quarter

ORDER BY quarter;



**Insights:** Here, I added a new column called quarter by using ALTER table command.

Using UPDATE and SET, I assigned quarter 1,2,3,4 to each of the rows by comparing with the months extracted.

And then just found total orders for each quarter.

1. **For each quarter and customer segment, find the total sales. Display data for quarters in column.**

**Solution:**

**SELECT**

**\***

FROM

(

SELECT

custseg,

quarter,

SUM(ordsales) totsales

FROM

customers

INNER JOIN orderdet ON orderdet.custid = customers.custid

GROUP BY

custseg,

quarter

ORDER BY totsales DESC

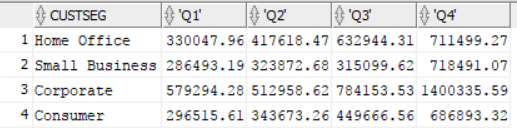
)

PIVOT ( SUM ( totsales )

FOR quarter

IN ( 'Q1','Q2','Q3','Q4' )

);



**Insights:** Now that we have Quarter in the table, we can just use that and customer segment to group by. We find the total sales for each combination.

1. **If you are creating a data warehouse for Office Product, what would be your dimensions and measures? Develop star schema for the data warehouse.**

**Solution:**

**Insights:** The OrderDet table is my fact table. Customers (Including Managers) , Products, Orders and Mangers are my Dimension tables. \* referring to the primary ids.

**Products**

**\***ProdID

ProdName

ProdCat

ProdSubCat

ProdCont

ProdUnitPrice

ProdMargin

**OrderDet- FACT TABLE**

\*OrderID

\*CustID  
\*ProdID

OrdPriority

OrdDiscount

OrdShipMode

OrdDate

OrdShipDate

OrdShipCost

OrdQty

OrdSales

**Customers**

\*CustID

\*CustName

\*CustState

CustCity

CustReg

CustZip

CustSeg

Region

RegManager

**`**

**Orders**

\*OrderID

Status