**PE 09 - SQL EXTENSIONS #2**

1. **Show the percentage of earnings by quarter for each year.**

SELECT year , quarter, QuarterTotal, TotalEarnings,

FORMAT(100\*QuarterTotal/TotalEarnings, 2) AS Percentage

FROM

(SELECT d.year, SUM(f.earnings) TotalEarnings

FROM PE\_Earnings.earningsfact f JOIN PE\_Earnings.datedim d USING(dateKey) GROUP BY d.year) AS a

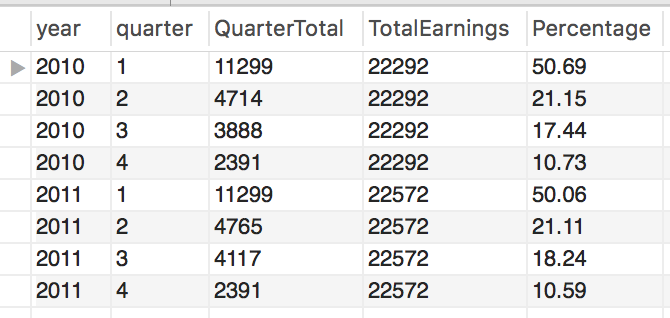
join

(SELECT d.year,d.quarter, SUM(f.earnings) QuarterTotal

FROM PE\_Earnings.earningsfact f JOIN PE\_Earnings.datedim d USING(dateKey) GROUP BY d.year,d.quarter) AS b

USING(year)

ORDER BY 1,2;



In order to get the percentage of earnings by quarter for each year, we are using nested select statements to get the total earnings by year and total earnings for each quarter. QuarterTotal is the subtotal for each quarter TotalEarnings is the subtotal for each year. and Then we are doing a join on both the select statements and grouping by year and quarter, this way we can calculate the percentage by dividing the QuarterTotal by TotalEarnings in the outer query and order the results by year and quarter.

1. **Rank the total earnings by month in 2011 but only show the top 5 months without using the RANK function or Limit.**

SELECT a.year , a.quarter, a.month, a.MonthTotal, count(b.MonthTotal) AS ‘Rank’

FROM

(SELECT d.year,d.quarter, d.month, SUM(f.earnings) MonthTotal

FROM PE\_Earnings.earningsfact f JOIN PE\_Earnings.datedim d USING(dateKey) GROUP BY d.year,d.quarter, d.month) AS a

JOIN

(SELECT d.year,d.quarter,d.month, SUM(f.earnings) MonthTotal

FROM PE\_Earnings.earningsfact f JOIN PE\_Earnings.datedim d USING(dateKey) GROUP BY d.year,d.quarter, d.month) AS b

USING(year)

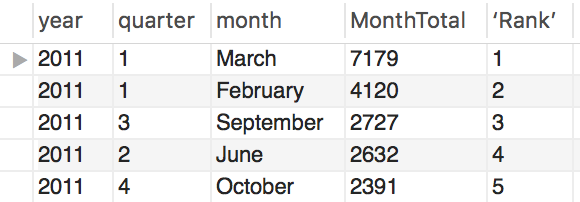
WHERE a.year = '2011'

AND a.MonthTotal <= b.MonthTotal

GROUP BY 3

HAVING count(b.MonthTotal) < 6

ORDER BY 5;

****

We can rank the total earnings by month in 2011 by using nested select statements to get the total earnings by month. The total earnings by month can be ranked by doing a self join along with two conditions - year should be 2011 and left select statement should be less than or equal to right select statement (This condition will help in omitting

irrelevant records) and group by month. Self joining is matching each record from the left side select statement with all the records of the right side select statement. Having command was used to get the top 5 ranks by selecting only records that are less than rank 6. Finally the results are ordered by rank.

1. **Show the earnings by quarter comparing quarters side by side.**

SELECT a.quarter q1, b.quarter q2, a.QuarterTotal amt1, b.QuarterTotal amt2

FROM

(SELECT d.quarter, SUM(f.earnings) QuarterTotal

FROM PE\_Earnings.earningsfact f JOIN PE\_Earnings.datedim d USING(dateKey) GROUP BY d.quarter) AS a

JOIN

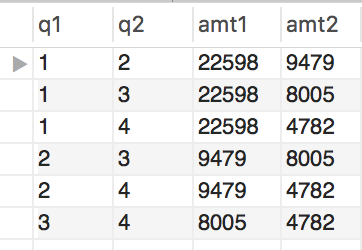
(SELECT d.quarter, SUM(f.earnings) QuarterTotal

FROM PE\_Earnings.earningsfact f JOIN PE\_Earnings.datedim d USING(dateKey) GROUP BY d.quarter) AS b

ON a.quarter < b.quarter

GROUP BY q1,q2

ORDER BY q1, q2;

****

The earnings by quarter can be compared side by side by using nested select statements to get the Quarter Total. Then we are doing a self join of the two nested select statements on a condition that left select should be less than right select(to avoid duplication of data) and group by and order by quarter.