psql -h 129.105.248.26 -U nkp030 -d postgres;

\c saleco\_dw;

\dt;

\d dwcustomer;

\d dwdaysalesfact;

\d dwproduct;

\d dwregion;

\d dwtime;

\d dwvendor;

1. List the total sales by region and customer. Your output should be sorted by region name and customer code. (**6 pts**)

SELECT reg\_name as region, c.cus\_code as customer, SUM(sale\_units \* sale\_price) as total\_sales

FROM dwdaysalesfact s

LEFT JOIN dwcustomer c

ON s.cus\_code = c.cus\_code

LEFT JOIN dwregion r

ON c.reg\_id = r.reg\_id

GROUP BY 1, 2

ORDER BY 1, 2;

1. Repeat #1 but produce the output using ROLLUP with region name and customer code. (**2 pts**)

SELECT reg\_name as region, c.cus\_code as customer, SUM(sale\_units \* sale\_price) as total\_sales

FROM dwdaysalesfact s

LEFT JOIN dwcustomer c

ON s.cus\_code = c.cus\_code

LEFT JOIN dwregion r

ON c.reg\_id = r.reg\_id

GROUP BY ROLLUP(1, 2)

ORDER BY 1, 2;

1. Repeat #1 but product the output using CUBE with region name and customer code. (**2 pts**)

SELECT reg\_name as region, c.cus\_code as customer, SUM(sale\_units \* sale\_price) as total\_sales

FROM dwdaysalesfact s

LEFT JOIN dwcustomer c

ON s.cus\_code = c.cus\_code

LEFT JOIN dwregion r

ON c.reg\_id = r.reg\_id

GROUP BY CUBE(1, 2)

ORDER BY 1, 2;

1. a) Explain the additional information/intelligence gained when using ROLLUP or CUBE (**5 pts**)  b) Use the output from questions 1, 2 and 3 to explain what the data reveals. (**5 pts**)

When using the rollup function on top of the group by function, you get a subtotal for the columns that are included in the group by. For example in question 2, we get a subtotal for each part of the group by in hierarchical order. This shows up in the answer as a subtotal for region & customer, region, and total. On the other hand, the cube function gives a subtotal for every combination of the fields in the group by, not maintain hierarchical order. This is seen in question 3 where we get a sub total for region & customer, region, customer, and total.

The additional value from both of these is the opportunity to see a variety of subtotals without the need for another query.

1. List the total sales by customer code, month, and product code; sort by customer code and month. (**5 pts**)

SELECT c.cus\_code, tm\_month, p\_code, SUM(sale\_units \* sale\_price) as total\_sales

FROM dwdaysalesfact s

LEFT JOIN dwcustomer c

ON s.cus\_code = c.cus\_code

LEFT JOIN dwtime t

ON s.tm\_id = t.tm\_id

GROUP BY 1, 2, 3

ORDER BY 1, 2;

1. Show all purchases (total sales) in September to show which customer bought the most product in September. Show customer code, customer name and total sales; sort all output by total sales with the highest sales on top.  (**5 pts**).

SELECT c.cus\_code, cus\_fname, cus\_lname, SUM(sale\_units \* sale\_price) as total\_sales

FROM dwdaysalesfact s

LEFT JOIN dwcustomer c

ON s.cus\_code = c.cus\_code

LEFT JOIN dwtime t

ON s.tm\_id = t.tm\_id

WHERE tm\_month = 9

GROUP BY 1, 2, 3

ORDER BY 4 desc;

1. List the total sales by month and product category. Your output should be sorted by month and product category. (**8 pts**)

SELECT tm\_month, p\_category, SUM(sale\_units \* sale\_price) as total\_sales

FROM dwdaysalesfact s

LEFT JOIN dwtime t

ON s.tm\_id = t.tm\_id

LEFT JOIN dwproduct p

ON s.p\_code = p.p\_code

GROUP BY 1, 2

ORDER BY 1, 2;

1. List the number of product sales (number of rows) and total sales by month. Your output should be sorted by month and should show one row per month. (**8 pts**)

SELECT tm\_month, COUNT(\*) as product\_sales, SUM(sale\_units \* sale\_price) as total\_sales

FROM dwdaysalesfact s

LEFT JOIN dwtime t

ON s.tm\_id = t.tm\_id

GROUP BY 1

ORDER BY 1;

1. Show product category, product code, product description and units sold (sum). Which product is the best seller based on units sold? a) Show units sold for September (**3 pts**), b) Show units sold for October (**3 pts**).

SELECT tm\_month, p\_category, p.p\_code, p\_descript, sum(sale\_units) as units\_sold

FROM dwdaysalesfact s

LEFT JOIN dwtime t

ON s.tm\_id = t.tm\_id

LEFT JOIN dwproduct p

ON s.p\_code = p.p\_code

WHERE p.p\_code IN

(SELECT p\_code FROM dwdaysalesfact GROUP BY 1 ORDER BY sum(sale\_units) desc LIMIT 1)

GROUP BY 1, 2, 3, 4;

1. List the number of product sales (number of rows) and total sales by month, product category, and product. Your output should be sorted by month, product category and product. **(8 pts**)

SELECT tm\_month, p\_category, p.p\_code, p\_descript, COUNT(\*) as product\_sales, SUM(sale\_units \* sale\_price) as total\_sales

FROM dwdaysalesfact s

LEFT JOIN dwtime t

ON s.tm\_id = t.tm\_id

LEFT JOIN dwproduct p

ON s.p\_code = p.p\_code

GROUP BY 1, 2, 3, 4

ORDER BY 1, 2, 3, 4;