Kagen Quiballo | MSDS 420 | Assignment 4 | 10/29/2021

Note: Total sales = sale\_units \* sale\_price

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

SELECT R.reg\_name, S.cus\_code,

SUM(S.sale\_units \* S.sale\_price) AS total\_sales

FROM dwdaysalesfact AS S

JOIN dwcustomer AS C

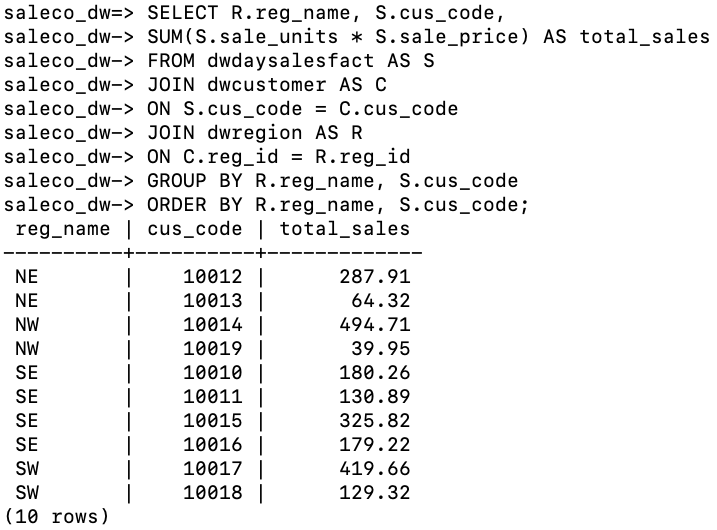
ON S.cus\_code = C.cus\_code

JOIN dwregion AS R

ON C.reg\_id = R.reg\_id

GROUP BY R.reg\_name, S.cus\_code

ORDER BY R.reg\_name, S.cus\_code;



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

SELECT coalesce (R.reg\_name::text,'All Regions') AS reg\_id,

coalesce (S.cus\_code::text, 'All Customers') AS cus\_code,

SUM(S.sale\_units \* S.sale\_price) AS total\_sales

FROM dwdaysalesfact AS S

JOIN dwcustomer AS C

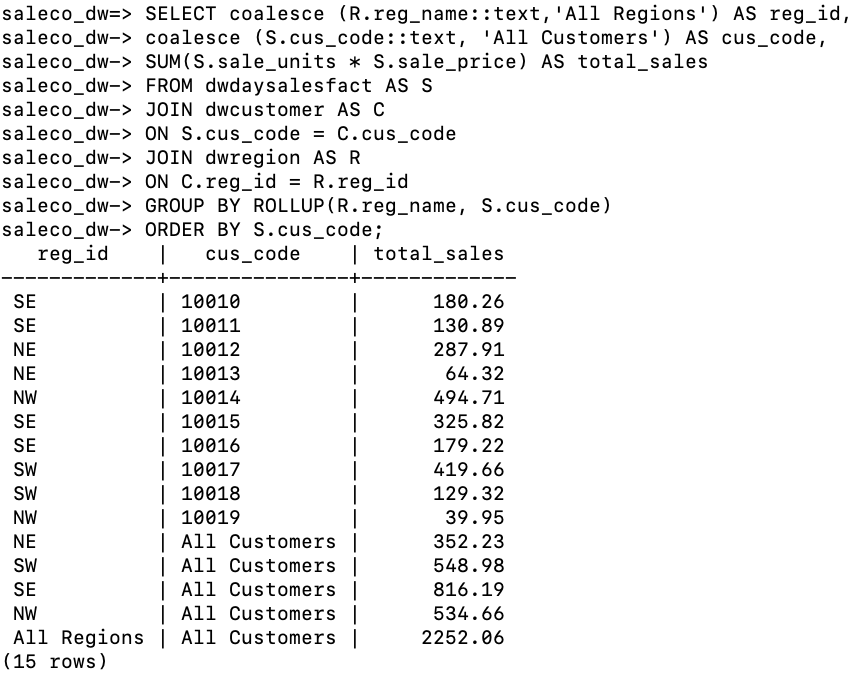
ON S.cus\_code = C.cus\_code

JOIN dwregion AS R

ON C.reg\_id = R.reg\_id

GROUP BY **ROLLUP**(R.reg\_name, S.cus\_code)

ORDER BY S.cus\_code;



**#2 continued**

SELECT coalesce (R.reg\_name::text,'All Regions') AS reg\_id,

coalesce (S.cus\_code::text, 'All Customers') AS cus\_code,

SUM(S.sale\_units \* S.sale\_price) AS total\_sales

FROM dwdaysalesfact AS S

JOIN dwcustomer AS C

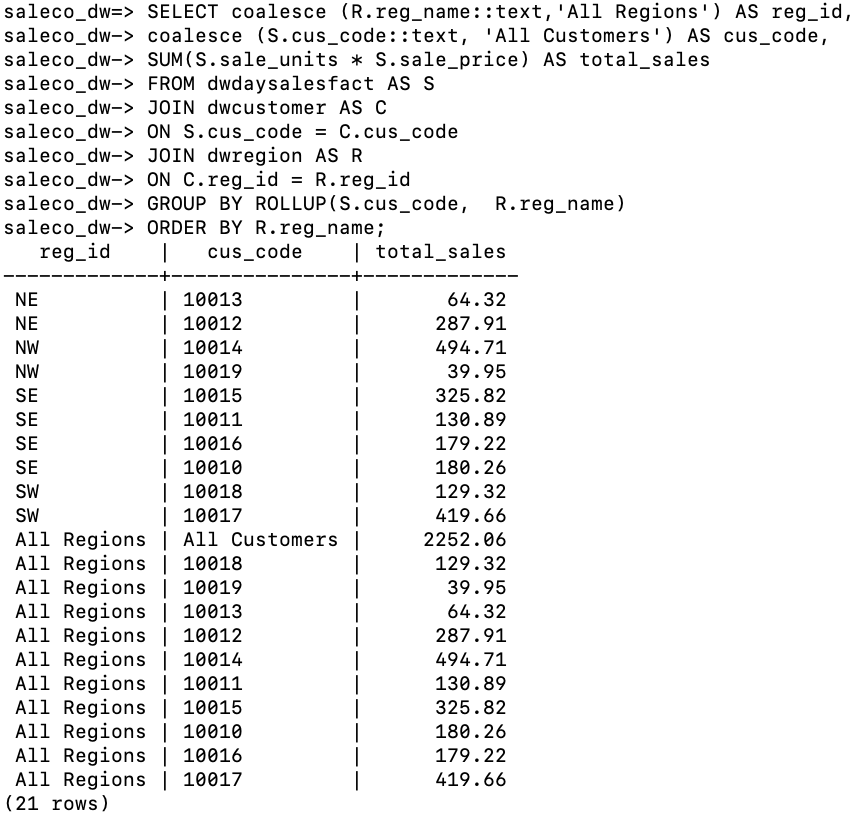
ON S.cus\_code = C.cus\_code

JOIN dwregion AS R

ON C.reg\_id = R.reg\_id

GROUP BY **ROLLUP**(S.cus\_code, R.reg\_name)

ORDER BY R.reg\_name;



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

SELECT coalesce (R.reg\_name::text,'All Regions') AS reg\_id,

coalesce (S.cus\_code::text, 'All Customers') AS cus\_code,

SUM(S.sale\_units \* S.sale\_price) AS total\_sales

FROM dwdaysalesfact AS S

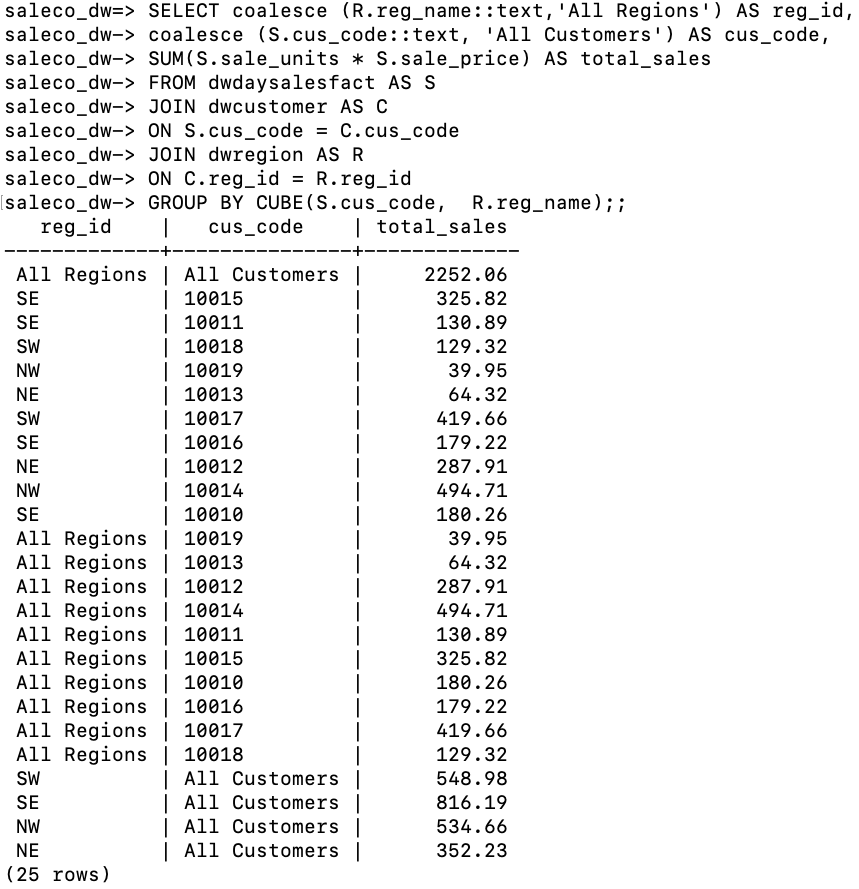
JOIN dwcustomer AS C

ON S.cus\_code = C.cus\_code

JOIN dwregion AS R

ON C.reg\_id = R.reg\_id

GROUP BY **CUBE**(S.cus\_code, R.reg\_name);



4. Explain the additional information/intelligence gained when using ROLLUP or CUBE. Discuss the output from the first three queries in your answer. (**10 pts**)

*ROLLUP can either add the subtotals for* ***either*** *the regional variable or customer variable whereas CUBE gives you the subtotals for* ***both*** *regional and customer data.*

*Q1 has a simple output of total sales by customer and region but no subtotals because we used a group by statement. Q2 can either add the additional regional or customer subtotals for total sales. We can see both individually, but using the ROLLUP function only allows us to see subtotals for one of these dimensions (as well as grand total). Q3 shows us the subtotals for both regional and customer data. We see these subtotals across both these dimensions because we used cube (as well as grand total).*

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

SELECT S.cus\_code, S.p\_code, T.tm\_month,

SUM(S.sale\_units \* S.sale\_price) AS total\_sales

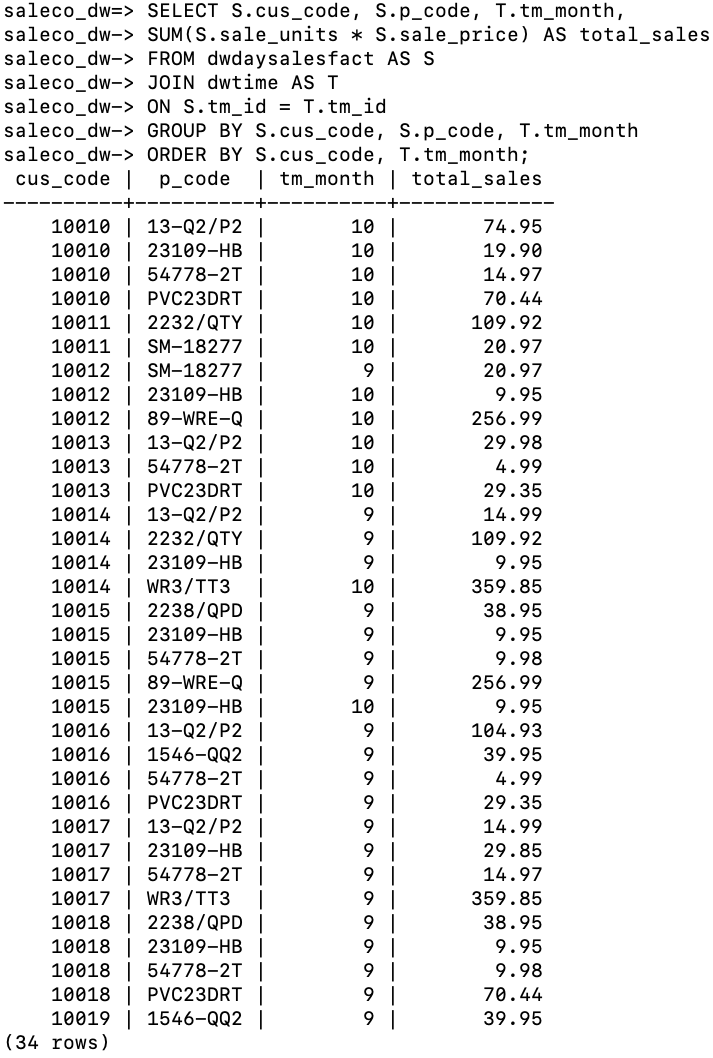
FROM dwdaysalesfact AS S

JOIN dwtime AS T

ON S.tm\_id = T.tm\_id

GROUP BY S.cus\_code, S.p\_code, T.tm\_month

ORDER BY S.cus\_code, T.tm\_month;



6. 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, C.cus\_fname, C.cus\_lname, T.tm\_month,

SUM(S.sale\_units \* S.sale\_price) AS total\_sales

FROM dwdaysalesfact AS S

JOIN dwtime AS T

ON S.tm\_id = T.tm\_id

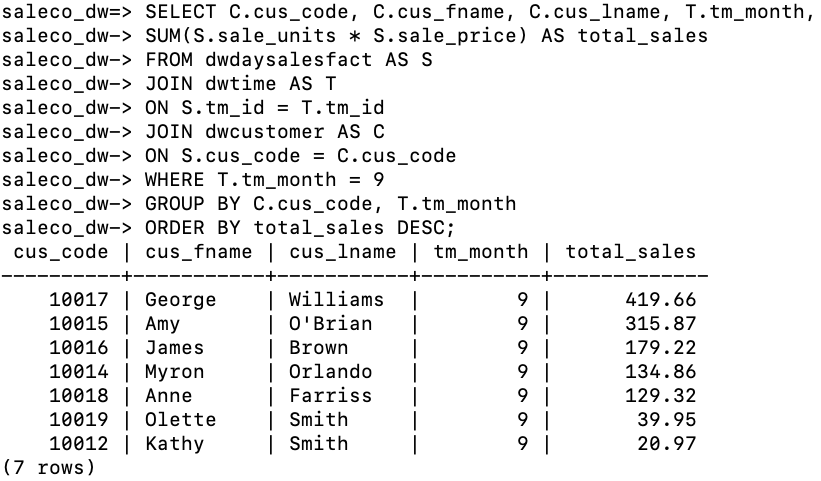
JOIN dwcustomer AS C

ON S.cus\_code = C.cus\_code

WHERE T.tm\_month = 9

GROUP BY C.cus\_code, T.tm\_month

ORDER BY total\_sales DESC;



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

SELECT T.tm\_month, P.p\_category,

SUM(S.sale\_units \* S.sale\_price) AS total\_sales

FROM dwdaysalesfact AS S

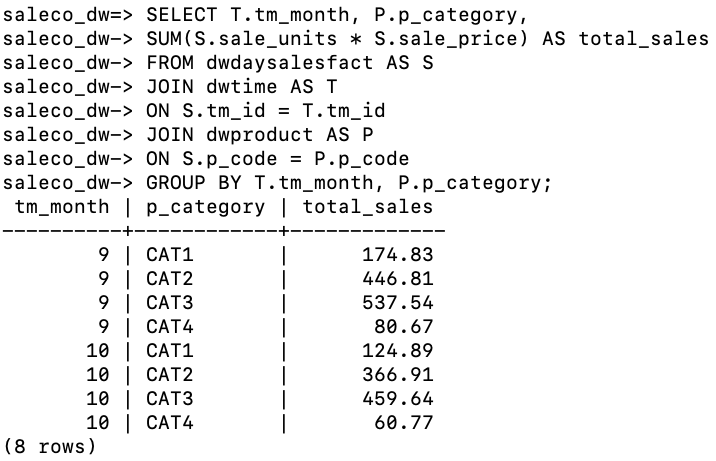
JOIN dwtime AS T

ON S.tm\_id = T.tm\_id

JOIN dwproduct AS P

ON S.p\_code = P.p\_code

GROUP BY T.tm\_month, P.p\_category;



8. 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 T.tm\_month,

SUM(S.sale\_units \* S.sale\_price) AS total\_sales,

COUNT(S.sale\_units) AS product\_sales

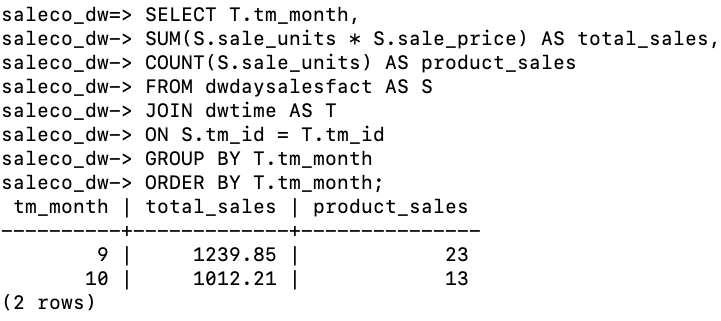
FROM dwdaysalesfact AS S

JOIN dwtime AS T

ON S.tm\_id = T.tm\_id

GROUP BY T.tm\_month

ORDER BY T.tm\_month;



9. 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 T.tm\_month, P.p\_category, P.p\_code, p.p\_descript,

SUM(S.sale\_units) AS units\_sold

FROM dwdaysalesfact AS S

JOIN dwtime AS T

ON S.tm\_id = T.tm\_id

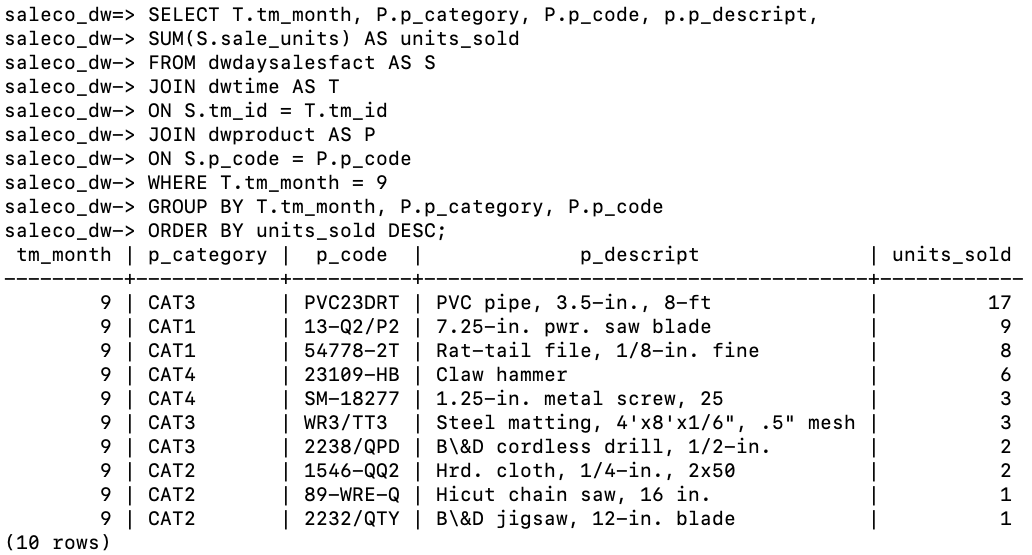
JOIN dwproduct AS P

ON S.p\_code = P.p\_code

WHERE **T.tm\_month = 9**

GROUP BY T.tm\_month, P.p\_category, P.p\_code

ORDER BY units\_sold DESC;



The “PVC pipe, 3.5-in., 8-ft” was the best seller of September (17 units)

**#9 continued**

SELECT T.tm\_month, P.p\_category, P.p\_code, p.p\_descript,

SUM(S.sale\_units) AS units\_sold

FROM dwdaysalesfact AS S

JOIN dwtime AS T

ON S.tm\_id = T.tm\_id

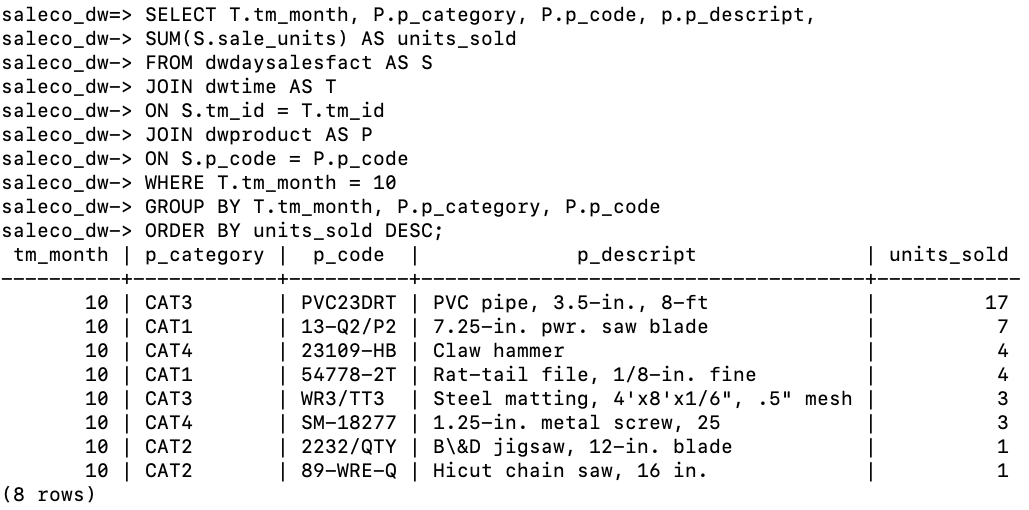
JOIN dwproduct AS P

ON S.p\_code = P.p\_code

WHERE **T.tm\_month = 10**

GROUP BY T.tm\_month, P.p\_category, P.p\_code

ORDER BY units\_sold DESC;



The “PVC pipe, 3.5-in., 8-ft” was the best seller of October as well. (17 units)

10. 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 T.tm\_month, P.p\_category, P.p\_code, p.p\_descript,

SUM(S.sale\_units) AS units\_sold,

SUM(S.sale\_units \* S.sale\_price) AS total\_sales

FROM dwdaysalesfact AS S

JOIN dwtime AS T

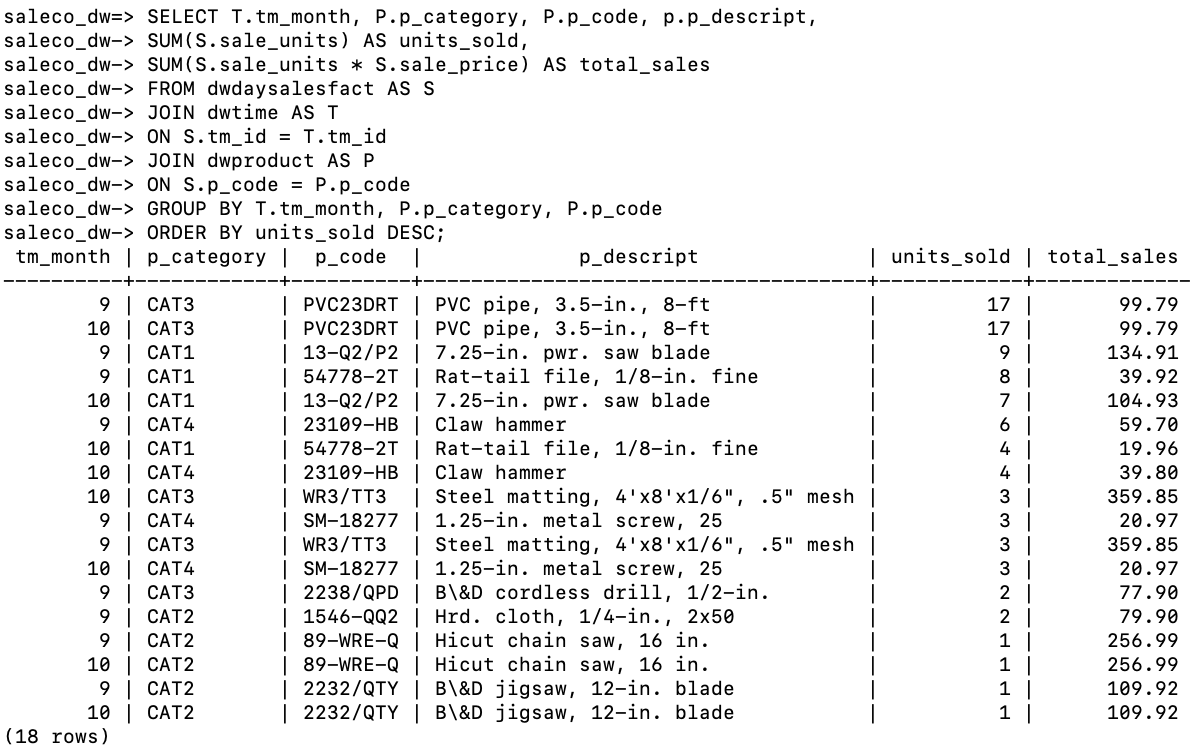
ON S.tm\_id = T.tm\_id

JOIN dwproduct AS P

ON S.p\_code = P.p\_code

GROUP BY T.tm\_month, P.p\_category, P.p\_code

ORDER BY units\_sold DESC;



List of relations

Schema | Name | Type | Owner

--------+----------------+-------+--------

public | dwcustomer | table | ajb254

public | dwdaysalesfact | table | ajb254

public | dwproduct | table | ajb254

public | dwregion | table | ajb254

public | dwtime | table | ajb254

public | dwvendor | table | ajb254

Table "public.dwcustomer"

Column | Type | Collation | Nullable | Default | Storage | Stats target | Description

-------------+-----------------------+-----------+----------+---------+----------+--------------+-------------

cus\_code | integer | | not null | | plain | |

cus\_lname | character varying(15) | | | | extended | |

cus\_fname | character varying(15) | | | | extended | |

cus\_initial | character(1) | | | | extended | |

cus\_state | character(2) | | | | extended | |

reg\_id | integer | | | | plain | |

Table "public.dwdaysalesfact"

Column | Type | Collation | Nullable | Default | Storage | Stats target | Description

------------+-----------------------+-----------+----------+---------+----------+--------------+-------------

tm\_id | integer | | not null | | plain | |

cus\_code | integer | | not null | | plain | |

p\_code | character varying(10) | | not null | | extended | |

sale\_units | integer | | | | plain | |

sale\_price | numeric(10,2) | | | | main | |

Table "public.dwproduct"

Column | Type | Collation | Nullable | Default | Storage | Stats target | Description

------------+-----------------------+-----------+----------+---------+----------+--------------+-------------

p\_code | character varying(10) | | not null | | extended | |

p\_descript | character varying(35) | | | | extended | |

p\_category | character varying(5) | | | | extended | |

v\_code | integer | | | | plain | |

Table "public.dwregion"

Column | Type | Collation | Nullable | Default | Storage | Stats target | Description

----------+-----------------------+-----------+----------+---------+----------+--------------+-------------

reg\_id | integer | | not null | | plain | |

reg\_name | character varying(10) | | | | extended | |

Table "public.dwtime"

Column | Type | Collation | Nullable | Default | Storage | Stats target | Description

----------+---------+-----------+----------+---------+---------+--------------+-------------

tm\_id | integer | | not null | | plain | |

tm\_year | integer | | | | plain | |

tm\_month | integer | | | | plain | |

tm\_day | integer | | | | plain | |

tm\_qtr | integer | | | | plain | |

Table "public.dwvendor"

Column | Type | Collation | Nullable | Default | Storage | Stats target | Description

------------+-----------------------+-----------+----------+---------+----------+--------------+-------------

v\_code | integer | | not null | | plain | |

v\_name | character varying(35) | | | | extended | |

v\_areacode | character(3) | | | | extended | |

v\_state | character(2) | | | | extended | |