key clauses: inner join, outer join, with, group by, having, select case, row_number(), partition

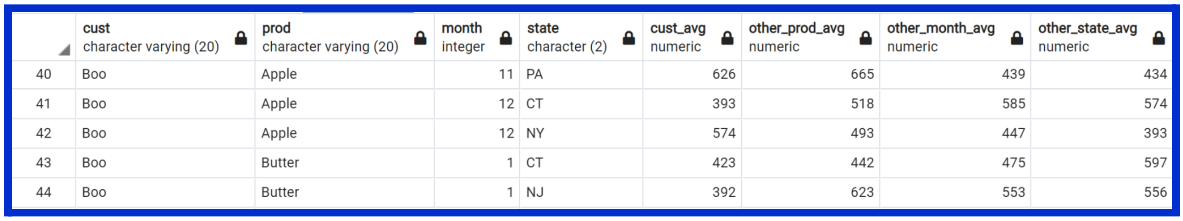
- For customer, product and state, show the average sales **before and after each quarter** (e.g., for Q2, show average sales of Q1 and Q3. For "before" Q1 and "after" Q4, **display <NULL>.** The "YEAR" attribute is not considered for this query – for example, both Q1 of 2017 and Q1 of 2018 are considered Q1 regardless of the year.

	cust character varying (20)	prod character varying (20)	state character (2)	quarter integer	before_avg numeric	after_avg numeric
1	Boo	Apple	CT	1	[null]	528
2	Boo	Apple	CT	2	539	683
3	Boo	Apple	CT	3	528	454
4	Boo	Apple	CT	4	683	[null]
5	Boo	Apple	NJ	1	[null]	514

```
with q as (
        SELECT cust, prod, state, quant,
                CASE
                WHEN 1<=month and month<=3  THEN 1
                WHEN 4<=month and month<=6  THEN 2
                WHEN 7<=month and month<=9  THEN 3
                WHEN 10<=month and month<=12  THEN 4
                END as quarter
        from sales
        order by cust, prod, state, quarter),
avg as (
        select cust, prod, state, quarter, round(avg(quant)) quarter_sum
        from q
        group by cust, prod, state, quarter
        order by cust, prod, state, quarter),
before as(
        select distinct q.cust, q.prod, q.state, q.quarter, a.quarter_sum before_avg
        from q left outer join avg a on q.cust=a.cust and q.prod=a.prod and q.state=a.state and
q.quarter-1=a.quarter
        order by cust, prod, state, quarter)
        select distinct b.cust, b.prod, b.state, b.quarter, b.before_avg, a.quarter_sum after_avg
        from before b left outer join avg a on b.cust=a.cust and b.prod=a.prod and b.state=a.state
and b.quarter+1=a.quarter
        order by cust, prod, state, quarter
```

- For each customer, product, month and state combination, compute
  (1) the customer's **average sale** of this product for the **given month and state**,
  (2) the customer's average sale for the given month and state, **but for all other products**

(3) the customer's average sale for the given product and state, **but for all other months** and
(4) the average sale of the product and the month **but for all other states**.



	cust character varying (20)	prod character varying (20)	month integer	state character (2)	cust_avg numeric	other_prod_avg numeric	other_month_avg numeric	other_state_avg numeric
40	Boo	Apple	11	PA	626	665	439	434
41	Boo	Apple	12	CT	393	518	585	574
42	Boo	Apple	12	NY	574	493	447	393
43	Boo	Butter	1	CT	423	442	475	597
44	Boo	Butter	1	NJ	392	623	553	556

```
with base as (
        select cust, prod, month, state, round(avg(quant)) avg_q
        from sales
        group by cust, prod, month, state
        order by cust, prod, month, state),
other_prod as(
        select  s1.cust, s1.prod, s1.month, s1.state, s1.avg_q cust_avg, round(avg(s2.quant))
other_prod_avg
        from base s1 inner join sales s2 on
        s1.cust = s2.cust and s1.prod != s2.prod and s1.month =s2.month and s1.state = s2.state
        group by s1.cust, s1.prod, s1.month, s1.state, s1.avg_q
        order by s1.cust, s1.prod, s1.month, s1.state),
other_month as(
        select  s1.cust, s1.prod, s1.month, s1.state, s1.cust_avg, s1.other_prod_avg,
round(avg(s2.quant)) other_month_avg
        from other_prod s1 inner join sales s2 on
         s1.cust = s2.cust and s1.prod = s2.prod and s1.month !=s2.month and s1.state = s2.state
        group by s1.cust, s1.prod, s1.month, s1.state, s1.cust_avg, s1.other_prod_avg
        order by s1.cust, s1.prod, s1.month, s1.state)
select  s1.cust, s1.prod, s1.month, s1.state, s1.cust_avg, s1.other_prod_avg, s1.other_month_avg,
round (avg(s2.quant)) other_state_avg
from other_month s1 inner join sales s2 on s1.cust = s2.cust and s1.prod = s2.prod and s1.month
=s2.month and s1.state != s2.state
group by s1.cust, s1.prod, s1.month, s1.state, s1.cust_avg, s1.other_prod_avg,
s1.other_month_avg
order by s1.cust, s1.prod, s1.month, s1.state, s1.cust_avg, s1.other_prod_avg,
s1.other_month_avg
```

- For each customer, execute **minimum** and **maximum sales** quantities along with the **corresponding products**, **dates** (i.e., dates of those maximum and minimum sales quantities), **states** and, **average sales quantity**.

	cust character varying (20)	min_q integer	min_prod character varying (20)	min_date date	min_state character (2)	max_q integer	max_prod character varying (20)	max_date date	max_state character (2)	avg_q numeric
1	Boo	1	Grapes	2020-12-18	NJ	1000	Dates	2020-02-24	NY	488
2	Mia	1	Ice	2019-08-11	PA	999	Cherry	2019-08-19	NJ	501
3	Dan	1	Dates	2020-08-13	PA	1000	Apple	2018-01-21	NJ	508
4	Dan	1	Cherry	2017-02-18	NY	1000	Apple	2018-01-21	NJ	508
5	Emily	2	Butter	2016-03-18	NJ	999	Cherry	2020-04-07	PA	472

```
with aggs as
        (select cust, min(quant) min_q, max(quant) max_q, avg(quant) avg_q
        from sales
        group by cust),
min_details as  --inclusive result, expanding the result as move
        (select a.cust cust, min_q, prod min_prod, date min_date, state min_state --min_q, ... is
the only thing, so no need to qualifying rest
        from aggs a inner join sales s
        on a.cust= s.cust and a.min_q= s.quant
        order by cust, prod)
select m.cust, m.min_q, m.min_prod, m.min_date, m.min_state, aggs.max_q, prod max_prod,
date max_date, state max_state, aggs.avg_q
from sales s inner join aggs on aggs.cust= s.cust and aggs.max_q= s.quant
inner join min_details m on m.cust = s.cust
```

- For each combination of customer and product, output the **maximum sales** quantities for October (regardless of the year) and **minimum sales** quantities for November and December (again, regardless of the year) in 3 separate columns. Like the first report, display the **corresponding dates** (i.e., dates of those maximum and minimum sales quantities). Furthermore, for October (MAX), include **only the sales that occurred after 2017** (that is, not to include sales that occurred in 2017 or earlier); for November (MIN) and December (MIN), include all sales.

	cust character varying (20)	prod character varying (20)	oct_max integer	date date	nov_min integer	date date	dec_min integer	date date
1	Dan	Fish	986	2020-10-12	63	2017-11-20	32	2018-12-12
2	Claire	Ham	949	2019-10-03	37	2019-11-25	23	2017-12-10
3	Chae	Dates	829	2017-10-20	134	2018-11-14	25	2019-12-05
4	Chae	Dates	829	2017-10-20	134	2018-11-14	25	2016-01-07
5	Claire	Fish	913	2020-10-12	31	2016-11-14	433	2018-12-15

```
with oct as (
        with aggs as (
        select cust, prod, max(quant) oct_max
        from sales
        where month =10 and year>=2017
        group by cust, prod
        order by cust, prod)
select aggs.cust, aggs.prod, oct_max, date
from sales inner join aggs on sales.cust = aggs.cust and sales.prod = aggs.prod and sales.quant =
oct_max),
nov as(
        with aggs as (
        select cust, prod, min(quant) nov_min
        from sales
        where month =11
        group by cust, prod
        order by cust, prod)
select aggs.cust, aggs.prod, nov_min, date
from sales inner join aggs on sales.cust = aggs.cust and sales.prod = aggs.prod and sales.quant =
nov_min),
dec as(
        with aggs as (
        select cust, prod, min(quant) dec_min
        from sales
        where month =12
        group by cust, prod
        order by cust, prod)
select aggs.cust, aggs.prod, dec_min, date
```

from sales inner join aggs on sales.cust = aggs.cust and sales.prod = aggs.prod and sales.quant = dec_min)
select oct.cust, oct.prod, oct.oct_max, oct.date, nov.nov_min, nov.date, dec.dec_min, dec.date
from oct inner join nov on oct.cust= nov.cust and oct.prod= nov.prod
inner join dec on oct.cust= dec.cust and oct.prod= dec.prod

- For each of the 12 months (regardless of the year), find the **most "popular"** and **least "popular"** products (those products with most and least total sales quantities) and the **corresponding total sales** quantities (i.e., SUMs).

	month integer	most_popular_product character varying (20)	most_popular_total_q bigint	least_popular_product character varying (20)	least_pop_total_q bigint
1	1	Eggs	48495	Apple	35692
2	2	Butter	47401	Eggs	33915
3	3	Eggs	59259	Jellies	29293
4	4	Apple	45852	Cherry	34941
5	5	Jellies	53615	Apple	33556

with base as
(select month, prod, sum(quant) total_q
from sales
group by month, prod
order by month),
mppq as
(select month, max(total_q) max_t
from base
group by month
order by month),
lppq as
(select month, min(total_q) min_t
from base
group by month
order by month),
mpp as
(select m.month, s.prod mos_prod, m.max_t mos_quant
from mppq m, base s
where s.total_q = m.max_t)
select l.month, mpp.mos_prod most_popular_product, mpp.mos_quant most_popular_total_q ,
s.prod least_popular_product, l.min_t least_pop_total_q
from lppq l natural join base s natural join mpp
where s.total_q = l.min_t

- For each product, find the **“most favorable”** month (when most amount of the product was sold) and the **“least favorable”** month (when the least amount of the product was sold).

	prod character varying (20)	most_fav_month integer	least_fav_mo integer
1	Apple	8	5
2	Butter	8	9
3	Cherry	3	5
4	Dates	3	1
5	Eggs	3	2

```
with base as
        (select prod, month, sum(quant) total
        from sales
        group by prod, month
        order by prod, month),
ml as
        (select prod, max(total) max, min(total) min
        from base
        group by prod),
mfm as
        (select base.prod prod, base.month most_fav_month
        from ml, base
        where ml.max = base.total)
select mfm.prod, mfm.most_fav_month, base.month least_fav_mo
from mfm natural join base natural join ml
where ml.min = base.total
order by prod
```

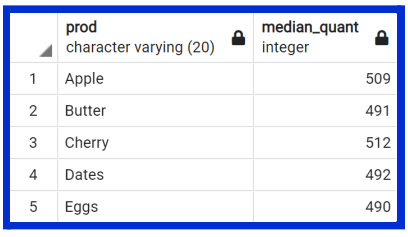
- For the years 2016, 2017, 2018, 2019 and 2020, show, for each product and customer combination, the **average sales quantities for the 4 states**, ‘CT’, ‘NY’, ‘NJ’ and ‘PA’ (in four separate columns). Also compute the **average for the “whole” year** (again ignoring the YEAR component, meaning simply compute AVG) along with **the total quantities** (SUM) and the **counts** (COUNT).

	year integer	prod character varying (20)	cust character varying (20)	round numeric	round numeric	round numeric	round numeric	round numeric	total bigint	count bigint
1	2016	Apple	Boo	357	340	340	470	461	9213	20
2	2016	Apple	Chae	526	507	507	534	485	7280	15
3	2016	Apple	Claire	506	460	460	927	564	11848	21
4	2016	Apple	Dan	484	437	437	314	383	10732	28
5	2016	Apple	Emily	519	467	467	433	476	12852	27

```
with base as
(select year, prod, cust, avg(quant) average, sum(quant) total, count(quant) count
```

```
from sales
group by year, prod, cust
order by year, prod, cust),
ct as
        (select year, prod, cust, state, avg(quant) ct_avg
        from sales
        group by year, prod, cust, state
         having state = 'CT'
        order by year, prod, cust),
ny as
        (select year, prod, cust, state, avg(quant) ny_avg
        from sales
        group by year, prod, cust, state
         having state = 'NY'
        order by year, prod, cust),
nj as
        (select year, prod, cust, state, avg(quant) nj_avg
        from sales
        group by year, prod, cust, state
         having state = 'NY'
        order by year, prod, cust),
pa as
        (select year, prod, cust, state, avg(quant) pa_avg
        from sales
        group by year, prod, cust, state
         having state = 'PA'
        order by year, prod, cust)
select base.year, base.prod, base.cust, round(ct.ct_avg), round(ny.ny_avg), round(nj.nj_avg),
round(pa.pa_avg), round(base.average), base.total, base.count
from base inner join ct on base.prod = ct.prod and base.year = ct.year and base.cust = ct.cust
inner join ny on base.prod = ny.prod and base.year = ny.year and base.cust = ny.cust
inner join nj on base.prod = nj.prod and base.year = nj.year and base.cust = nj.cust
inner join pa on base.prod = pa.prod and base.year = pa.year and base.cust = pa.cust
```

- For each product, find the **median sales** quantity (assume an odd number of sales for simplicity of presentation). without using built-in median() method



	prod character varying (20)	median_quant integer
1	Apple	509
2	Butter	491
3	Cherry	512
4	Dates	492
5	Eggs	490

```
with num as (
        select prod, quant, row_number() over (partition by prod order by  sales.quant) as num
        from sales
        order by prod, quant, num),
counts as (
        select prod, count(prod)/2 med
        from sales
        group by prod
        order by prod)
select num.prod, num.quant median_quant
from num, counts
where num.num = counts.med and num.prod = counts.prod
order by prod
```

- For customer and product, find the **month by which time, 75% of the sales quantities have been purchased**.

	cust character varying (20)	prod character varying (20)	min integer
1	Boo	Apple	9
2	Boo	Butter	9
3	Boo	Cherry	10
4	Boo	Dates	10
5	Boo	Eggs	9

```
with sum as (
        select cust, prod, month, sum(quant) q
        from sales
        group by cust, prod, month
        order by cust, prod, month),
cum_sum as(
        select cust, prod, month,  sum(q) OVER (partition by cust, prod ORDER BY cust, prod,
month) AS cum_sum
        from sum
        order by cust, prod, month),
total as (
        select cust, prod,  sum(quant) total
        from sales
        group by cust, prod
        order by cust, prod),
over as (
        select c.cust cust, c.prod prod, month
        from cum_sum c inner join total t on c.prod=t.prod and c.cust=t.cust
        where c.cum_sum >= t.total *0.75)
select cust, prod, min(month)
```

```
from over
group by cust, prod
order by cust, prod
```