**569. Median Employee Salary**

WITH cte AS

(select \*, ROW\_NUMBER() OVER(PARTITION BY company ORDER BY salary) AS rnk,

count(id) OVER(PARTITION by company) AS n

from Employee

)

select id, company, rnk, salary

from cte

where (n%2=0 AND (rnk = n/2 OR rnk = (n/2)+1))OR (n%2 !=0 AND rnk = n/2)

**571. Find Median Given Frequency of Numbers**

with recursive cte as ( select number, frequency, 1 as cnt

from Numbers

union

select number, frequency, cnt+1 from

cte where cnt <frequency

),

cte1 as (select number, ROW\_NUMBER() OVER(order by number) as row\_id,

count(\*) over()as total

from cte)

select sum(number) from(

select number from cte1

where ((total % 2 != 0) AND (row\_id = (total/2)+1)) OR

((total % 2=0) AND ((row\_id = total/2) OR (row\_id = (total/2)+1)))) sub

[**579. Find Cumulative Salary of an Employee**](https://leetcode.com/problems/find-cumulative-salary-of-an-employee/)

select sub.id,sub.salary from

(select id,sum(salary) as salary from montly\_sal

where month <=2 group by id

union

select id, salary from montly\_sal where month = 1

union

select id,sum(salary) as salary from montly\_sal

where month <=3

group by id) sub

order by sub.id

# **615 - Average Salary: Departments VS Company**

select dept.month, dept\_id,

case when dept\_avg> comp\_avg then 'higher'

when dept\_avg < comp\_avg then 'lower'

else 'same'

end as comparison

from

(select e.dept\_id as dept\_id, avg(s.amount) as dept\_avg, Extract(month from s.pay\_date) as month

from salary s

join emp e

on s.emp\_id = e.emp\_id

group by e.dept\_id, month) dept join

(select avg(s.amount) as comp\_avg, Extract(month from s.pay\_date) as month

from salary s

group by month) comp\_avg

on comp\_avg.month = dept.month

order by dept.month

# **Students Report By Geography**

select America, Asia,Europe from

(select row\_number() over(order by name) as amid, name as America from student where continent = 'America') s1

left join

(select row\_number() over(order by name) as asiaid,name as Asia from student where continent = 'Asia') s2

on s1.amid = s2.asiaid

left join

(select row\_number() over(order by name) as euid,name as Europe from student where continent = 'Europe') s3

on s3.euid = s2.asiaid

# 1127 - User Purchase Platform

with cte as (select spend\_date, user\_id, max(platform) as platform,

sum(amount) as total

from spending

group by spend\_date, user\_id

having count(distinct platform) = 1

union all

select spend\_date, user\_id, 'both' as platform,sum(amount) as total from spending

group by spend\_date, user\_id

having count(distinct platform) = 2)

select spend\_date, platform,

count(user\_id), total from cte

group by spend\_date,platform , total

# **1159. Market Analysis II**

select u.user\_id as seller\_id,

(case when u.fav\_brand = b.item\_brand then 'yes' else 'no' end)

as fav\_brand

from

(select seller\_id, a.item\_id, r, item\_brand

from (select \*, row\_number() over (partition by seller\_id order by order\_date) as r

from orders) a

join items i

on a.item\_id = i.item\_id where r=2) b

join users u

on u.user\_id = b.seller\_id

# Tournament Winners

select g,p from

(select p.group\_id g, p.player\_id p,sub1.max\_score as s,

row\_number() over(partition by p.group\_id order by sub1.max\_score desc) as rnk from

(select player\_id, max(score) as max\_score from

(select m1.first\_player player\_id,m1.first\_score+m2.first\_score as score from Matches m1

join Matches m2

on m1.first\_player = m2.second\_player

union

select first\_player as player\_id, sum(first\_score) as score from matches

group by first\_player

union

select second\_player as player\_id, sum(second\_score) as score from matches

group by second\_player) sub

group by player\_id order by player\_id) sub1

join players p

on p.player\_id = sub1.player\_id) sub2

where rnk=1

# Get the Second Most Recent Activity

Select usename,activity,statedate,enddate from

(Select username ,activity, startDate , endDate , rank() over(partition by username order by enddate desc) rnk, count(activity) over(partition by username) as cnt from useractivity) sub

Where rnk = 2 or count = 1

# 1384 - Total Sales Amount by Year

select prod\_name,report\_year,no\_period\*avg\_sales from

(select avg\_sales, prod\_name, '2019' as report\_year, case when period\_start >='2019-01-01' and period\_end <='2019-12-31' then

(period\_end - period\_start) +1

when period\_start >='2019-01-01' and period\_end <='2020-12-31' then

period\_end - '2019-12-31'

when period\_start >='2018-01-01' and period\_end <='2020-12-31' then

period\_end - '2019-01-01'

else 0 end as no\_period

from sales

union

select avg\_sales,prod\_name, '2018' as report\_year,

case when (period\_start BETWEEN '2018-01-01' and '2018-12-31') and

period\_end <= '2018-12-31' then

(period\_end - period\_start) +1

when (period\_start BETWEEN '2018-01-01' and '2018-12-31') and period\_end <='2020-12-31' then

('2018-12-31' - period\_start)+1

else 0 end

from sales

union

select avg\_sales,prod\_name, '2020' as report\_year,

case when (period\_start BETWEEN '2018-01-01' and '2018-12-31') and

(period\_end BETWEEN '2020-01-01' and '2020-12-31') then

(period\_end - '2020-01-01')+1

when (period\_start BETWEEN '2019-01-01' and '2019-12-31') and

(period\_end BETWEEN '2020-01-01' and '2020-12-31') then

(period\_end - '2020-01-01')+1

else 0 end

from sales) sub

where no\_period !=0

# Find the Quiet Students in All Exams

select distinct student\_id from exam

where student\_id not in

(select student\_id from(

select exam\_id, student\_id, score, rank() over(partition by exam\_id order by score desc) as rnk,

count(score) over(partition by exam\_id) as cnt

from exam) sub

where rnk = 1 or rnk = cnt)

# 1479 - Sales by Day of the Week

select distinct category,

sum(case when day\_=1 then total else 0 end) as Monday,

sum(case when day\_=2 then total else 0 end) as Tudsday,

sum(case when day\_=3 then total else 0 end) as Wednesday,

sum(case when day\_=4 then total else 0 end) as Thursday,

sum(case when day\_=5 then total else 0 end) as Friday,

sum(case when day\_=6 then total else 0 end) as Saturday,

sum(case when day\_=0 then total else 0 end) as sunday from(

select distinct items1.category as category,

extract(dow from o.order\_date) as day\_,sum(o.qty) over(partition by extract(dow from o.order\_date)) as total from

orders1 o join items1

on items1.item\_id = o.item\_id) sub

group by category

[1767. Find the Subtasks That Did Not Execute](https://leetcode.com/problems/find-the-subtasks-that-did-not-execute/)

with recursive cte as (select task\_id, 1 as sub\_task, subtask\_count from task

union select task\_id,sub\_task+1,subtask\_count from cte where sub\_task < subtask\_count),

cte1 as(select \*, row\_number() over(partition by task\_id) rk from cte )

SELECT task\_id, sub\_task from cte1

where (task\_id, sub\_task) not in (SELECT task\_id, subtask\_id from Executed)

order by task\_id

# 1917 - Leetcodify Friends Recommendations

with cte as (SELECT a.user\_id user\_id,

b.user\_id recommended\_id

FROM listens a

INNER JOIN listens b ON a.user\_id != b.user\_id AND a.song\_id = b.song\_id

GROUP BY a.user\_id,b.user\_id HAVING Count(DISTINCT a.song\_id) >= 3)

select \* from cte

where (user\_id,recommended\_id) NOT IN (select user1\_id,user2\_id from frindship) AND

(user\_id,recommended\_id) NOT IN (select user2\_id,user1\_id from frindship)

# 1919 - Leetcodify Similar Friends

with cte as (select a.user\_id as user\_id,

b.user\_id as recommended\_id

from listens a join listens b

on a.user\_id != b.user\_id and a.song\_id = b.song\_id

group by a.user\_id, b.user\_id having count(b.song\_id)>=3)

select \* from cte

where (user\_id, recommended\_id) IN (select user1\_id,user2\_id from frindship)

# 1972 - First and Last Call On the Same Day

select distinct call\_id from(

select \*, row\_number() over(partition by call\_time ) as row\_num,

count(call\_id) over(partition by call\_time) as cnt

from calls) sub

where row\_num = 1 or row\_num=cnt

union

select distinct receive\_id from (

select \*, row\_number() over(partition by call\_time ) as row\_num,

count(call\_id) over(partition by call\_time) as cnt

from calls) sub1

where row\_num=1 and row\_num=cnt

[2004. The Number of Seniors and Juniors to Join the Company](https://leetcode.com/problems/the-number-of-seniors-and-juniors-to-join-the-company/)

select title, count(emp\_id) from

(select \* from

(SELECT emp\_id,title,

SUM(salary) OVER(ORDER BY salary,emp\_id) AS salary\_accu

FROM emp1

WHERE title='s') sub where salary\_accu <=70000

union

select \* from(

SELECT emp\_id,title,

SUM(salary) OVER(ORDER BY salary,emp\_id) AS salary\_accu

FROM emp1

WHERE title='j') sub where salary\_accu+(select max(salary) from emp1 where title='s' group by title)<=70000) sub2

GROUP BY title

# 2474. Customers With Strictly Increasing Purchases

select customer\_id, sum(cnt) from(

select customer\_id, yr,total,lead(total, 1) over(partition by customer\_id order by yr),

case when lead(total, 1) over(partition by customer\_id order by yr) - total > 0 then 1 else 0 end as cnt

from

(

select customer\_id, sum(price) as total, extract(year from order\_date) as yr from orders2

group by customer\_id, extract(year from order\_date) order by customer\_id,yr) sub) sub2

group by customer\_id order by sum(cnt) desc Limit 1

# 2362. Generate the Invoice

with cte as(

select p.invoice\_id, sum(p.quantity\*p1.price) from purchases p

left join products p1

on p.product\_id = p1.product\_id

group by p.invoice\_id order by sum(p.quantity\*p1.price) desc,p.invoice\_id asc limit 1)

select p.product\_id ,p.quantity, p.quantity \* prod.price

from purchases p left join products prod

on p.product\_id = prod.product\_id

where invoice\_id in (select invoice\_id from cte)

[570. Managers with at Least 5 Direct Re](https://leetcode.com/problems/managers-with-at-least-5-direct-reports/)

select manager, max(id) as count\_ from

(select count(e.id) as id,m.name as manager from staff e

join staff m

on e.manager\_id = m.id

group by manager) sub

group by manager order by count\_ desc LIMIT 1

# 1336 - Number of Transactions per Visit

select t as transaction\_count, count(user\_id) from(

select count(t.\*) as t,t.user\_id,t.transaction\_date from transactions t

left join visits v

on v.user\_id = t.user\_id and v.visit\_date = t.transaction\_date

group by t.user\_id, t.transaction\_date) sub

group by t

union

select transaction\_count,count(user\_id) from(

select 0 as transaction\_count, user\_id, visit\_date from visits v

where NOT EXISTS

(SELECT user\_id,transaction\_date from transactions t

where t.user\_id = v.user\_id and t.transaction\_date = v.visit\_date))sub1

group by transaction\_count

order by transaction\_count

# 1369 - Get the Second Most Recent Activity

select \* from

(SELECT \*,ROW\_number() over(partition by username order by enddate desc) rnk,

count(activity) over(partition by username ) cnt from useractivity) sub

where rnk=2 or cnt=1

# 1635 - Hopper Company Queries I

with recursive cte\_month as (select 1 as month union

select month+1 from cte\_month where month<=11),

cte\_accepted\_rides as (

select

distinct( extract(month from b.requested\_at)) as month,

count(ride\_id) over(partition by extract(month from b.requested\_at) ) as accepted\_rides

from AcceptedRides a

join Rides b using(ride\_id)

where extract(year from b.requested\_at)=2020)

select cte\_month.month, count(d.driver\_id) as active\_drivers,

COALESCE(cte\_accepted\_rides.accepted\_rides,0) as accepted\_rides

from cte\_month

left join cte\_accepted\_rides

using(month)

left join drivers d

on extract(month from d.join\_date)<=cte\_month.month

group by cte\_month.month,cte\_accepted\_rides.accepted\_rides order by cte\_month.month asc

# 1645 - Hopper Company Queries II

with recursive cte\_month as (select 1 as month union

select month+1 from cte\_month where month<=11),

cte\_accepted\_rides as (

select

distinct( extract(month from b.requested\_at)) as month,

count(ride\_id) over(partition by extract(month from b.requested\_at) ) as accepted\_rides

from AcceptedRides a

join Rides b using(ride\_id)

where extract(year from b.requested\_at)=2020)

select cte\_month.month, round(count(d.driver\_id),2) as active\_drivers,

round(round(COALESCE(cte\_accepted\_rides.accepted\_rides,0),2)/round(count(d.driver\_id),2)\*100,2)

from cte\_month

left join cte\_accepted\_rides

using(month)

left join drivers d

on extract(month from d.join\_date)<=cte\_month.month

group by cte\_month.month,cte\_accepted\_rides.accepted\_rides order by cte\_month.month asc

# 1651 - Hopper Company Queries III

with recursive cte\_month as (select 1 as month union

select month+1 from cte\_month where month<=11),

cte\_accepted\_rides as (

select

distinct( extract(month from b.requested\_at)) as month,

sum(ride\_distance) over(partition by extract(month from b.requested\_at)) as total\_ride,

sum(ride\_duration) over(partition by extract(month from b.requested\_at)) as total\_duration

from AcceptedRides a

join Rides b using(ride\_id)

where extract(year from b.requested\_at)=2020)

select cte\_month.month, round(AVG(COALESCE(cte\_accepted\_rides.total\_ride,0))

OVER(ORDER BY cte\_month.month ROWS BETWEEN 0 PRECEDING AND 2 FOLLOWING),2) as avg\_distance,

round(AVG(COALESCE(cte\_accepted\_rides.total\_duration,0))

OVER(ORDER BY cte\_month.month ROWS BETWEEN 0 PRECEDING AND 2 FOLLOWING),2) as avg\_duration

from cte\_month

left join cte\_accepted\_rides

using(month) left join drivers d

on extract(month from d.join\_date)<=cte\_month.month

group by cte\_month.month,cte\_accepted\_rides.total\_ride,cte\_accepted\_rides.total\_duration

order by cte\_month.month asc

[2252. Dynamic Pivoting of a Table](https://leetcode.com/problems/dynamic-pivoting-of-a-table/)

select product\_id, sum(case when store='LC' then price end) as LC\_Store,

sum(case when store='Nozama' then price end) as Nozama,

sum(case when store='Shop' then price end) as Shop,

sum(case when store='Souq' then price end) as Souq

from products1

group by product\_id

order by product\_id

# 2253. Dynamic Unpivoting of a Table

select product\_id,lc\_store as price, 'LC' as store

from cte\_pivot where lc\_store IS NOT NULL

union

select product\_id,Nozama as price, 'Nozama' as store

from cte\_pivot where Nozama IS NOT NULL

union

select product\_id,Shop as price, 'Shop' as store

from cte\_pivot where Shop IS NOT NULL

union

select product\_id,Souq as price, 'Souq' as store

from cte\_pivot where Souq IS NOT NULL

# 1225. Report Contiguous Dates

select 'succeeded' as period\_state, min(sucess\_date) as start\_date,

max(sucess\_date) as end\_date

from

(select sucess\_date, row\_number() over(order by sucess\_date) as row\_num

from Success

where sucess\_date between '2019-01-01' and '2019-12-31') t

group by extract(day from sucess\_date) - row\_num

union

select 'failed' as period\_state, min(fail\_date) as start\_date, max(fail\_date) as end\_date

from

(select fail\_date, row\_number() over(order by fail\_date) as row\_num

from failed

where fail\_date between '2019-01-01' and '2019-12-31') t

group by extract(day from fail\_date) - row\_num

# 1097 - Game Play Analysis V

select a.event\_date as event\_date,round(count(a.player\_id),2) as installs,

round(round(count(b.player\_id),2)/round(count(a.player\_id),2),2) as day1\_retention

from (select player\_id, min(event\_date) as event\_date from activity

group by player\_id) a

left join Activity b

on a.player\_id = b.player\_id AND a.event\_date+1 = b.event\_date

group by a.event\_date

**Medium –**

[585. Investments in 2016](https://leetcode.com/problems/investments-in-2016/)

select sum(tiv\_2016) as tiv\_2016 from insurance

where concat(lat,lon) not in(

select concat(lat,lon)

from insurance group by lat,lon

having count(1) >1)

# 602. Friend Requests II Who Has the Most Friends

select id, sum(cnt) as num from

(select count(\*) as cnt, request\_id as id from requested

group by request\_id

union

select count(\*) as cnt1, accept\_id from requested

group by accept\_id) sub

group by id order by sum(cnt) desc

limit 1

[614. Second Degree Follower](https://leetcode.com/problems/second-degree-follower/)

select followee as follower,count(\*) as num from follow

where followee IN(Select follower from follow)

group by followee

[1077. Project Employees III](https://leetcode.com/problems/project-employees-iii/)

select project\_id, employee\_id FROM

(select p.project\_id as project\_id, e.employee\_id as employee\_id,

e.experience,

rank() over(partition by p.project\_id

order by e.experience) as rnk,

count(\*) over(partition by p.project\_id

) as cnt

from project p

join empl e

on p.employee\_id = e.employee\_id)sub

where rnk = 2

# Game Play Analysis III

select a1.player\_id, a1.event\_date,

sum(a2.games\_played)

from activity a1

join activity a2

on a2.event\_date<=a1.event\_date

AND a1.player\_id=a2.player\_id

group by a1.player\_id, a1.event\_date

order by a1.player\_id, a1.event\_date

select player\_id, event\_date,

sum(games\_played)

OVER(partition by player\_id order by event\_date

ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW)

from activity

# **550. Game Play Analysis IV**

with cte as (select player\_id,

count(games\_played) as total

from activity group by player\_id)

select round(round(sub.cnt,2)/round(cte.total,2),2) as Fraction from

(select player\_id, event\_date - lag(event\_date)

Over(partition by player\_id order by event\_date) as cnt

from activity) sub

join cte

on sub.player\_id = cte.player\_id

where cnt=1

# 1112 - Highest Grade For Each Student

select student\_id, min(course\_id), grades from

(select student\_id, course\_id, grades,

rank() over(partition by student\_id order by grades desc)as rnk

from enrollments) sub

where rnk =1

group by student\_id, grades

# **1126. Active Businesses**

with cte as (

select event\_type, sum(occur)/count(b\_id) as average from events

group by event\_type)

select b\_id from events

left join cte

on events.event\_type = cte.event\_type

where events.occur> cte.average

group by b\_id

having count(b\_id) >1

# 1132 - Reported Posts II

select round(avg(remove\_count),2) as average\_daily\_percent from(

select a.action\_date, round(round(count(r.post\_id),2)/count(a.post\_id) \* 100,2) as remove\_count

from actions a

left join remove r

on a.post\_id = r.post\_id

where a.extra = 'spam'

group by a.action\_date) sub

# 1164 - Product Price at a Given Date

select min(prod\_id) as id, max(new\_price) as price from products2

where change\_date <= ('2019-08-16')

group by prod\_id

union

select prod\_id as id, 10 as price from products2

where prod\_id not in (select prod\_id from products2 where change\_date <= ('2019-08-16'))

# **Immediate Food Delivery II Problem**

select round(sum(case when order\_date = pref\_date then 1 else 0 end),2)/count(delivery\_id) \* 100 as status

from delivery

where order\_date in (select min(order\_date) from delivery group by customer\_id)

# 1205 - Monthly Transactions II

select extract(month from t.trans\_date), t.country, count(t.state),

sum(case when t.state = 'approved' then t.amount end) as approved\_amount,

sum(case when t.state = 'declined' then t.amount end) as charge\_back,

count(c.trans\_id)

from transact t

left join chargeback c

on c.trans\_id = t.id AND extract(month from t.trans\_date) = extract(month from c.trans\_date)

group by extract(month from t.trans\_date), t.country

# 1270. All People Report to the Given Manager

select employee\_id from(

select employee\_id from employee where manager\_id in

(select employee\_id from employee where manager\_id IN (select employee\_id from employee where manager\_id = 1)) sub

where employee\_id !=1

# 1285 - Find the Start and End Number of Continuous Ranges

select start\_id, min(end\_id) from

(select id as start\_id from logs where id not in (select id+1 from logs)) s1,

(select id as end\_id from logs where id not in (select id-1 from logs)) s2

where start\_id <= end\_id

group by start\_id

order by start\_id

# 1308 - Running Total for Different Genders

select gender,day,

sum(scores) over(partition by gender order by day ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW)

from scores

select s.gender,s.day, (select sum(s1.scores) from scores s1 where s1.gender = s.gender AND s1.day <= s.day)

from scores s

group by s.gender,s.day

order by s.gender,s.day

# 1355 - Activity Participants

select activity from(

select activity, count(\*) from friends

group by activity

having count(\*) NOT IN

(select activity, count(\*) as total from Friends group by activity order by count(\*) asc Limit 1

union

select activity, count(\*) as total from Friends group by activity order by count(\*) desc Limit 1))

[1468. Calculate Salaries](https://leetcode.com/problems/calculate-salaries/)

with cte as(

select s.company\_id, max(salary) as total from salaries

group by s.company\_id)

select case when cte.total< 1000 then s.salary

when cte.total IN between 1000 and 10000 then salary - salary\*0.24

when cte.total > 10000 then salary - salary\*0.49

company\_id,emp\_id, emp\_name from cte

join salaries s

on cte.company\_id = s.company\_id

[1532. The Most Recent Three Orders](https://leetcode.com/problems/the-most-recent-three-orders/)

select customer\_id, order\_id,order\_date,customer\_name from

(select customer\_id, order\_id,order\_date, rank() over(partition by customer\_id order by date desc) rnk

from orders) sub

join customers

on customers.id = orders.customer\_id

where rnk<=3

# 1613 - Find the Missing IDs

with recursive cte as(select 1 as id union

select id+1 from cte where id <(select max(customer\_id) from customer))

select \* from cte

where id NOT IN (select customer\_id from customer)

[1699. Number of Calls Between Two P](https://leetcode.com/problems/number-of-calls-between-two-persons/)

select least(from\_id,to\_id) as person1,

greatest(from\_id,to\_id) as person2, sum(duration)

from caller

group by person1,person2

order by person1

[1709. Biggest Window Between Visits](https://leetcode.com/problems/biggest-window-between-visits/)

select id, max(case when lead\_ IS NOT NULL then lead\_ - visit\_date

else '2021-01-01' - visit\_date end) as biggest\_window from(

SELECT id, visit\_date, lead(visit\_date) over(partition by id order by visit\_date) as lead\_ from uservist) sub

group by id

# 1715 - Count Apples and Oranges

select sum(ifnull(box.apple\_count, 0) + ifnull(chest.apple\_count, 0)) as apple\_count,

sum(ifnull(box.orange\_count, 0) + ifnull(chest.orange\_count, 0)) as orange\_count

from Boxes as box

left join Chests as chest

on box.chest\_id = chest.chest\_id;

# 1783 - Grand Slam Titles

select sub.player\_id, p.player\_name, count(sub.player\_id) from

(select wimbledon as player\_id from champinships

union

select fr\_open from champinshoips

union

select usopen from championships

union

select auopen from championships) sub

inner join players p

on p.player\_id = sub.player\_id

group by sub.player\_id

[2142. The Number of Passengers in Each Bus I](https://leetcode.com/problems/the-number-of-passengers-in-each-bus-i/)

select b.bus\_id, count(p.arrival\_time)

from buses b

left join passengers p

on b.arrival\_time < p.arrival\_time

group by b.bus\_id

[2159. Order Two Columns Independe](https://leetcode.com/problems/order-two-columns-independently/)

with cte1 as (select first\_col, rank() over(first\_col order by asc) as rnk from table),

cte2 as (select second\_col, rank() over(second\_col order by desc) as rnk from table)

select first\_col, second\_col from cte1

join cte2

using(rnk)

[2175. The Change in Global Rankings](https://leetcode.com/problems/the-change-in-global-rankings/)

with cte1 as (select team\_id, name, points, rank() over(order by points desc) as

rnk1 from teams point)sub

with cte2 as (select team\_id, name, points, rank() over(order by points-points\_change desc) as

rnk2 from teams point

join points\_change using(team\_id)

select team\_id,name, rnn1-rnk2 from cte1 join cte2 using(team\_id)

[2228. Users With Two Purchases Within Seven Days](https://leetcode.com/problems/users-with-two-purchases-within-seven-days/)

select user\_id from

(select user\_id, purchase\_date,lag(purchase\_date) over(partition by user\_id order purchase\_date) as lag\_

from puchase) sub

where purchase\_date - lag\_ <=7

[2308. Arrange Table by Gender](https://leetcode.com/problems/arrange-table-by-gender/)

select id, gender from

(select id, gender, row\_number() over (partition by gender order by id) as rank\_,

case when gender = 'female' then 1

when gender = 'other' then 2

else 3 end as tags

from genders) sub

order by rank\_,tags

[2314. The First Day of the Maximum R](https://leetcode.com/problems/the-first-day-of-the-maximum-recorded-degree-in-each-city/)

select \* from(

select city\_id, day, degree, rank() over(partition by city\_id order by degree desc, day asc) rank from weather) sub

where rank = 1