1. Consecutive numbers

Write an sql query to find all numbers that appear at least three times consecutively.

Ans)

create table consecutive\_nums (id int,num int);

insert into consecutive\_nums values

(1,34),

(2,45),

(3,45),

(4,56),

(5,45),

(6,47),

(7,47),

(8,47),

(9,47),

(10,48),

(11,48),

(12,48);

select sub1.num,count(\*) from (select sub.num num,sub.lead\_val from (select id,num,lead(num,1,0) over (order by id) as lead\_val

from consecutive\_nums) sub

where sub.num=sub.lead\_val) sub1

group by sub1.num

having count(\*) >=2;

----------------

1. Trips and users

<https://medium.com/jen-li-chen-in-data-science/leetcode-sql-dec67244ed80>

-- customers who never order

create table Trips (id int,client\_id int,driver\_id int,city\_id int,status varchar(50),request\_at date);

----

insert into Trips values

(1,1,10,1,'completed','2013-10-01'),

(2,2,11,1,'cancelled\_by\_driver','2013-10-01'),

(3,3,12,6,'completed','2013-10-01'),

(4,4,13,6,'cancelled\_by\_client','2013-10-01'),

(5,1,10,1,'completed','2013-10-02'),

(6,2,11,6,'completed','2013-10-02'),

(7,3,12,6,'completed','2013-10-02'),

(8,2,12,12,'completed','2013-10-03'),

(9,3,10,12,'completed','2013-10-03'),

(10,4,13,12,'cancelled\_by\_driver','2013-10-03');

----

create table users (id int,banned varchar(50),role varchar(50));

------

insert into users values

(1,'No','client'),

(2,'Yes','client'),

(3,'No','client'),

(4,'No','client'),

(10,'No','Driver'),

(11,'No','Driver'),

(12,'No','Driver'),

(13,'No','Driver');

----

select sum(case when a.status!='completed' then 1 else 0 end)/count(\*) as cancellation\_rate, a.request\_at

from

(select t.id id,client\_id,request\_at ,status

from trips t inner join users u

on t.client\_id=u.id where u.banned='No') a

inner join

(select t.id id,driver\_id,request\_at ,status

from trips t inner join users u

on t.driver\_id=u.id where u.banned='No') b

on a.id=b.id

group by a.request\_at;

-----------

1. [Human Traffic of Stadium](https://leetcode.com/problems/human-traffic-of-stadium/)

<https://leetcode.com/problems/human-traffic-of-stadium/description/>

create table stadium (id int,visit\_date date,people int);

insert into stadium values

(1,'2017-01-01',10),

(2,'2017-01-02',109),

(3,'2017-01-03',150),

(4,'2017-01-04',990),

(5,'2017-01-05',14),

(6,'2017-01-06',1455),

(7,'2017-01-07',199),

(8,'2017-01-08',188),

(9,'2017-01-09',188),

(10,'2017-01-10',10),

(11,'2017-01-11',101),

(12,'2017-01-12',101),

(13,'2017-01-13',90);

with q1 as (

select \*,

count(\*) over( order by id range between current row and 2 following ) following\_cnt,

count(\*) over( order by id range between 2 preceding and current row ) preceding\_cnt,

count(\*) over( order by id range between 1 preceding and 1 following ) current\_cnt

from stadium

where people > 99

)

select id, visit\_date, people

from q1

where following\_cnt = 3 or preceding\_cnt = 3 or current\_cnt = 3

order by visit\_date;

1. Tree node

<https://leetcode.com/problems/tree-node/description/>

create table tree (id int,p\_id int);

insert into tree values

(1,null),

(2,1),

(3,1),

(4,2),

(5,2),

(6,3);

select id,

case when p\_id is null then 'Root'

when id not in (select p\_id from tree where p\_id is not null) then 'Leaf'

else 'Inner'

end as type from tree;

1. Exchange seats

<https://leetcode.com/problems/exchange-seats/description/>

create table seat (id int,student varchar(100));

insert into seat values

(1,'A'),

(2,'B'),

(3,'C'),

(4,'D'),

(5,'E'),

(6,'F'),

(7,'G');

with s as

(select id,student,

case when (lead(id,1) over (order by id)) is null then id

when (mod(lead(id,1,0) over (order by id),2)) ==0 then lead(id,1,0) over (order by id)

else lag(id,1,0) over (order by id) end as swapped\_seat

from seat)

select s1.id,s.student

from seat s1 inner join s

on s1.id=s.swapped\_seat order by s1.id;

1. Sales analysis III

<https://leetcode.com/problems/sales-analysis-iii/description/>

create table product (product\_id int,product\_name varchar(100),unit\_price int);

create table sales (seller\_id int,product\_id int,buyer\_id int,sale\_sate date,quantity int,price int);

insert into product values

(1,'S8',1000),

(2,'G4',800),

(3,'iphone',1400);

insert into sales values

(1,1,1,'2019-01-21',2,2000),

(1,2,2,'2019-02-17',1,800),

(2,2,3,'2019-06-02',1,800),

(3,3,4,'2019-05-13',2,2800);

select product\_name

from sales inner join product

on sales.product\_id=product.product\_id

group by product\_name

having (min(sale\_sate) >=to\_date('2019-01-01','YYYY-MM-DD') and max(sale\_sate <=to\_date('2019-03-31','YYYY-MM-DD')));