***Below are the Leetcode SQL 50 problems with answers***

[**1757. Recyclable and Low Fat Products**](https://leetcode.com/problems/recyclable-and-low-fat-products/)

select product\_id

from products

where low\_fats = "Y" and recyclable = "Y"

[**584. Find Customer Referee**](https://leetcode.com/problems/find-customer-referee/)

# Write your MySQL query statement below

select name

from customer

where referee\_id is null or referee\_id <> 2

[**595. Big Countries**](https://leetcode.com/problems/big-countries/)

# Write your MySQL query statement below

select name, population, area

from world

where area >= 3000000 or population >= 25000000

[**1148. Article Views I**](https://leetcode.com/problems/article-views-i/)

# Write your MySQL query statement below

select distinct author\_id as id

from views

where (author\_id = viewer\_id)

order by id

[**1683. Invalid Tweets**](https://leetcode.com/problems/invalid-tweets/)

# Write your MySQL query statement below

select tweet\_id

from tweets

where length(content)>15

[**1378. Replace Employee ID With The Unique Identifier**](https://leetcode.com/problems/replace-employee-id-with-the-unique-identifier/)

select b.unique\_id, a.name

from employees a left join employeeUNI b on b.id = a.id

[**1068. Product Sales Analysis I**](https://leetcode.com/problems/product-sales-analysis-i/)

# Write your MySQL query statement below

select product\_name, year, price

from sales

left join product on sales.product\_id = product.product\_id

[**1581. Customer Who Visited but Did Not Make Any Transactions**](https://leetcode.com/problems/customer-who-visited-but-did-not-make-any-transactions/)

select V.customer\_id, count(V.visit\_id) AS count\_no\_trans

from Visits V

left join Transactions T on V.visit\_id = T.visit\_id

where T.transaction\_id IS NULL

group by V.customer\_id;

[**197. Rising Temperature**](https://leetcode.com/problems/rising-temperature/)

select w1.id

from weather w1 , weather w2

where datediff(w1.recordDate, w2.recordDate) = 1

and w1.temperature > w2.temperature

[**1661. Average Time of Process per Machine**](https://leetcode.com/problems/average-time-of-process-per-machine/)

# Write your MySQL query statement below

select a1.machine\_id,round(avg(a2.timestamp - a1.timestamp),3) as processing\_time

from activity a1

join activity a2 on a1.machine\_id = a2.machine\_id

and a1.process\_id = a2.process\_id

and a1.timestamp < a2.timestamp

group by a1.machine\_id

[**577. Employee Bonus**](https://leetcode.com/problems/employee-bonus/)

# Write your MySQL query statement below

select a.name, b.bonus

from employee a

left join bonus b on a.empid = b.empid

where b.bonus <1000 or b.bonus is null

[**1280. Students and Examinations**](https://leetcode.com/problems/students-and-examinations/)

# Write your MySQL query statement below

select s.student\_id, s.student\_name, sub.subject\_name,

count(e.subject\_name) as attended\_exams

from students s

cross join subjects sub

left join examinations e on e.student\_id = s.student\_id and sub.subject\_name = e.subject\_name

group by s.student\_id, s.student\_name, sub.subject\_name

order by s.student\_id

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

select e1.name

from employee e1

join employee e2 on e1.id = e2.managerid

group by e2.managerid

having count(e2.managerid) >= 5

[**1934. Confirmation Rate**](https://leetcode.com/problems/confirmation-rate/)

# Write your MySQL query statement below

select s.user\_id, ifnull(round((count(case when c.action = 'confirmed' then 1 end)/count(c.action)),2),0.00) as confirmation\_rate

from signups s

left join confirmations c on s.user\_id = c.user\_id

group by s.user\_id

[**620. Not Boring Movies**](https://leetcode.com/problems/not-boring-movies/)

# Write your MySQL query statement below

select id, movie, description, rating

from cinema

where description != "boring" and id%2 != 0

order by rating desc;

[**1251. Average Selling Price**](https://leetcode.com/problems/average-selling-price/)

# Write your MySQL query statement below

select p.product\_id, ifnull(round(sum(p.price \* u.units)/sum(u.units),2),0) as average\_price

from prices p

left join unitssold u on p.product\_id = u.product\_id

and u.purchase\_date between p.start\_date and p.end\_date

group by product\_id

[**1075. Project Employees I**](https://leetcode.com/problems/project-employees-i/)

# Write your MySQL query statement below

select project\_id, round(sum(e.experience\_years)/count(p.employee\_id),2) as average\_years

from project p

inner join employee e on e.employee\_id = p.employee\_id

group by project\_id

[**1633. Percentage of Users Attended a Contest**](https://leetcode.com/problems/percentage-of-users-attended-a-contest/)

# Write your MySQL query statement below

select contest\_id , round(count(distinct user\_id)\*100/(select count(user\_id) from users),2) as percentage

from register r

group by r.contest\_id

order by percentage DESC, contest\_id

[**1211. Queries Quality and Percentage**](https://leetcode.com/problems/queries-quality-and-percentage/)

# Write your MySQL query statement below

select query\_name,

round(avg(q.rating/q.position),2) as quality,

round(avg(CASE WHEN rating < 3 THEN 1 ELSE 0 END)\*100,2) as poor\_query\_percentage

from queries q

group by query\_name

[**1193. Monthly Transactions I**](https://leetcode.com/problems/monthly-transactions-i/)

select country,

date\_format(trans\_date, '%Y-%m') as month,

count(\*) as trans\_count,

sum(case when state ='approved' then 1 else 0 end) as approved\_count,

sum(case when state = 'approved' then amount else 0 end) as approved\_total\_amount,

sum(case when state = 'approved' then amount else amount end) as trans\_total\_amount

-- sum(case when )

from transactions t

group by month, country

[**1174. Immediate Food Delivery II**](https://leetcode.com/problems/immediate-food-delivery-ii/)

with customers\_first\_order as

(select \*,row\_number() over(partition by customer\_id order by order\_date) as first\_order from delivery)

select

round(sum(if(order\_date=customer\_pref\_delivery\_date,1,0))\*100/count(first\_order),2) as immediate\_percentage

from customers\_first\_order

where customers\_first\_order.first\_order < 2

[**550. Game Play Analysis IV**](https://leetcode.com/problems/game-play-analysis-iv/)

with players\_first\_login as

(

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

),

players\_logged\_in as

(select

f1.player\_id

from players\_first\_login f1

join activity a1 on a1.player\_id = f1.player\_id

where a1.event\_date= date\_add(f1.event\_date, interval 1 day)

)

select

round(count(distinct p1.player\_id) / (select count(player\_id) from players\_first\_login),2) as fraction

from players\_logged\_in p1

[**2356. Number of Unique Subjects Taught by Each Teacher**](https://leetcode.com/problems/number-of-unique-subjects-taught-by-each-teacher/)

# Write your MySQL query statement below

select teacher\_id,count(distinct subject\_id) as cnt

from teacher

group by teacher\_id

[**1141. User Activity for the Past 30 Days I**](https://leetcode.com/problems/user-activity-for-the-past-30-days-i/)

with cte as

(

select count(distinct user\_id) as active\_users, activity\_date as day

from activity

where activity\_date between '2019-06-28' AND '2019-07-27'

group by activity\_date

)

Select \* from cte c

-- select count(distinct a1.user\_id) as active\_users, a1.activity\_date as day

-- from activity a1

-- join activity a2 on a1.user\_id = a2.user\_id

-- where a1.activity\_date = a2.activity\_date

-- group by a1.activity\_date

[**1070. Product Sales Analysis III**](https://leetcode.com/problems/product-sales-analysis-iii/)

select product\_id, year as first\_year, quantity, price

from sales

where (product\_id, year) in (select product\_id, min(year) from sales group by product\_id)

[**596. Classes More Than 5 Students**](https://leetcode.com/problems/classes-more-than-5-students/)

with count\_of\_students as (

select class, count(student) as head\_count from courses

group by class

)

select c1.class from count\_of\_students c1

where c1.head\_count >= '5'

[**1729. Find Followers Count**](https://leetcode.com/problems/find-followers-count/)

# Write your MySQL query statement below

select distinct user\_id,count(distinct follower\_id) as followers\_count

from followers

group by user\_id order by user\_id

[619. Biggest Single Number](https://leetcode.com/problems/biggest-single-number/)

with max\_num as (

select num, count(num) as count\_num from mynumbers group by num

)

select max(num) as num from max\_num

where count\_num = 1

[**1045. Customers Who Bought All Products**](https://leetcode.com/problems/customers-who-bought-all-products/)

select customer\_id

from customer

group by customer\_id

having count(distinct product\_key) = (select count(product\_key) as counts from product)

[**1731. The Number of Employees Which Report to Each Employee**](https://leetcode.com/problems/the-number-of-employees-which-report-to-each-employee/)

select e1.employee\_id, e1.name,

count(e2.reports\_to) as reports\_count,

round(avg(e2.age)) as average\_age

from employees e1

join employees e2

on e1.employee\_id = e2.reports\_to

group by e1.employee\_id order by e1.employee\_id