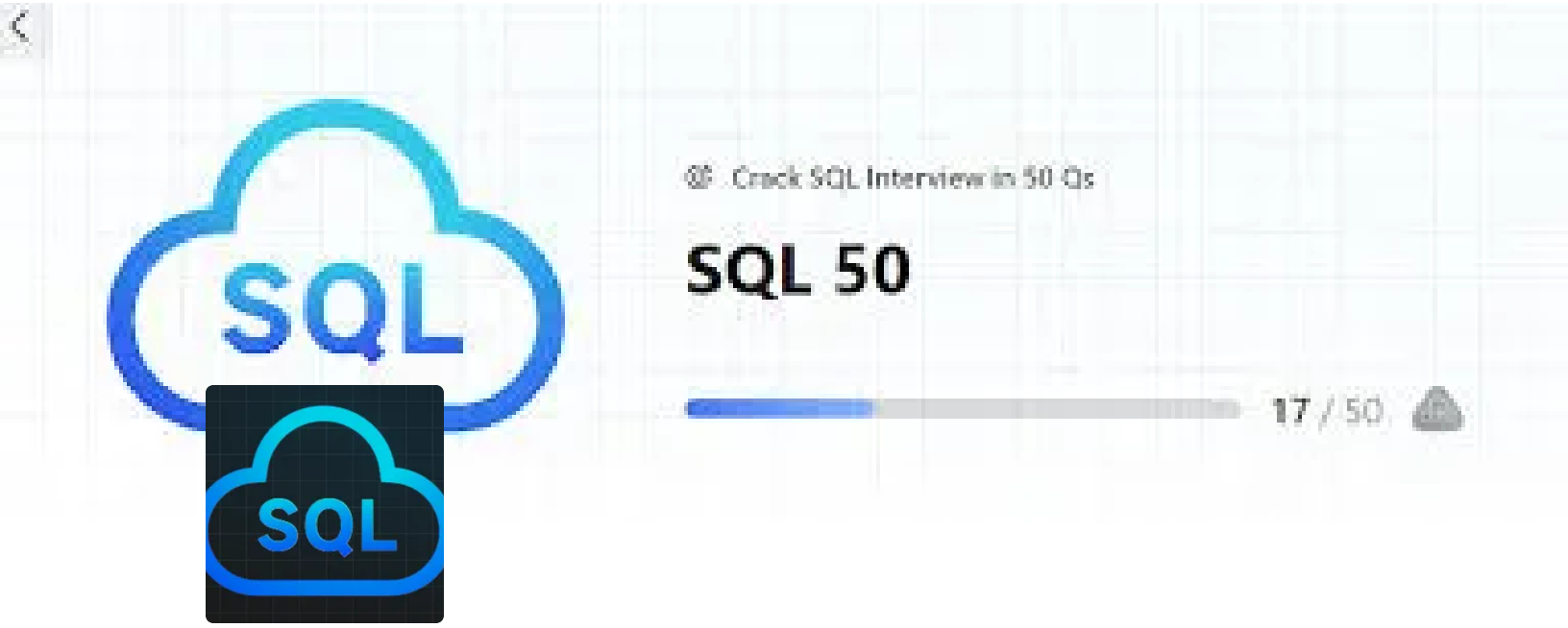

SQL 50 Study Plan

LeetCode SQL 50 - Study Plan - LeetCode

- 1757. Recyclable and Low Fat Products

```
select product_id from Products where low_fats='Y' and recyclable ='Y';
```

- 584. Find Customer Referee

```
select name from Customer where id not in ( select id from Customer where
referee_id =2 );
```

- 595. Big Countries

```
select name, population, area from World where area >= 3000000 or population
>= 25000000;
```

- 1148. Article Views I

```
select distinct author_id as id from Views where viewer_id = author_id order
by id asc;
```

- 1683. Invalid Tweets

```
select tweet_id from Tweets where len(content) > 15;
```

- 1378. Replace Employee ID With The Unique Identifier

```
select unique_id, name from EmployeeUNI eu right join Employees e on e.id =
eu.id;
```

- 1068. Product Sales Analysis I

```
select product_name, year, price from Sales s left join Product p on
p.product_id = s.product_id;
```

- 1581. Customer Who Visited but Did Not Make Any Transactions

```
select customer_id, count(*) 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

```
select w1.id from Weather w1 join Weather w2 on w1.recordDate = dateadd(day,
1, w2.recordDate) where w1.temperature > w2.temperature;
```

- 1661. Average Time of Process per Machine

```
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.activity_type =
'start' and A2.activity_type = 'end' group by A1.machine_id;
```

- 577. Employee Bonus

```
select e.name, b.bonus from Employee e left join Bonus b on e.empId = b.empId
where b.bonus < 1000 or b.bonus is null;
```

- 1280-Students-and-Examinations

```
select s.student_id, s.student_name, j.subject_name, count(e.subject_name) as
attended_exams from Students s cross join Subjects j left join Examinations e
on s.student_id = e.student_id and j.subject_name = e.subject_name group by
s.student_id, s.student_name, j.subject_name order by s.student_id,
j.subject_name;
```

- 570. Managers with at Least 5 Direct Reports

```
select e1.name from Employee e1 join Employee e2 on e1.id = e2.managerId
group by e1.id, e1.name having count(e2.id) >= 5;
```

- 1934. Confirmation Rate

```
select s.user_id, ROUND( ISNULL( CAST( SUM( CASE When c.action = 'confirmed'
THEN 1 ELSE 0 END )as float ) / NULLIF(count(c.user_id),0) ,0 ) ,2) 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

```
select * from Cinema where description != 'boring' and id % 2 = 1 order by
rating desc;
```

- 1251. Average Selling Price

```
select p.product_id, round(isnull( cast( sum(p.price * u.units) as float) /
nullif(sum(u.units),0) ,0) ,2) 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 p.product_id;
```

- 1075. Project Employees I

```
select p.project_id, round(avg(e.experience_years * 1.0) ,2) as average_years
from Project p join Employee e on p.employee_id = e.employee_id group by
p.project_id
```

- 1633. Percentage of Users Attended a Contest

```
select contest_id, round((count(distinct user_id) * 100.0) / (select count(*)
from Users), 2) as percentage from Register group by contest_id order by
percentage desc, contest_id asc;
```

- 1211. Queries Quality and Percentage

```
select query_name, round(avg(rating*1.0 / position), 2) as quality,
round(sum(case when rating < 3 then 1 else 0 end)* 100.0 / count(query_name)
,2) as poor_query_percentage from Queries where query_name is not null group
by query_name;
```

- 1193. Monthly Transactions I

```
select format(t.trans_date, 'yyyy-MM') as month, t.country, count(*) as
trans_count, sum(case when t.state = 'approved' then 1 else 0 end) as
approved_count, sum(amount) as trans_total_amount, sum(case when t.state =
'approved' then t.amount else 0 end) as approved_total_amount from
Transactions t group by format(t.trans_date, 'yyyy-MM'), t.country;
```

- 1174. Immediate Food Delivery II

```
SELECT ROUND( SUM(CASE WHEN d.order_date = d.customer_pref_delivery_date THEN
1 ELSE 0 END) * 100.0 / COUNT(*), 2 ) AS immediate_percentage FROM Delivery d
WHERE d.order_date = ( SELECT MIN(order_date) FROM Delivery WHERE customer_id
= d.customer_id );
```

- 550. Game Play Analysis IV

```
select round( cast( (select count(distinct a1.player_id) from Activity a1
where exists ( select 1 from Activity a2 where a2.player_id = a1.player_id
and dateadd(day, 1, (select min(event_date) from Activity a3 where
a3.player_id = a1.player_id)) = a2.event_date ) ) as FLOAT ) / (select
count(distinct player_id) from Activity), 2 ) as fraction;
```

- 550. Game Play Analysis IV ⇒ Using CTE

```
with first_login as ( select player_id, min(event_date) as first_login_date
from activity group by player_id ) select round( cast(count(distinct case
when a.event_date = dateadd(day, 1, f.first_login_date) then a.player_id end)
as float) / count(distinct f.player_id), 2 ) as fraction from first_login f
left join activity a on f.player_id = a.player_id and a.event_date =
dateadd(day, 1, f.first_login_date);
```

- 2356. Number of Unique Subjects Taught by Each Teacher

```
select teacher_id, count(distinct subject_id) as cnt from teacher group by
teacher_id;
```

- 1141. User Activity for the Past 30 Days I

```
select activity_date as day, count(distinct user_id) as active_users from
activity where activity_date between dateadd(day, -29, '2019-07-27') and
'2019-07-27' group by activity_date order by activity_date;
```

- 1070. Product Sales Analysis III

```
select s.product_id, s.year as first_year, s.quantity, s.price from Sales s
where year = (select min(year) from sales as sl where sl.product_id =
s.product_id);
```

- 596. Classes More Than 5 Students

```
select class from Courses group by class having count(student)>= 5;
```

- 1729. Find Followers Count

```
select user_id, count(follower_id) as followers_count from Followers group by
user_id order by user_id;
```

- 619. Biggest Single Number

```
select max(num) as num from (select num from mynumbers group by num having
count(num) = 1) as single_numbers; or => with numbercount as ( select num,
count(num) as count from mynumbers group by num ) select max(num) as num from
numbercount where count = 1;
```

- 1045. Customers Who Bought All Products

```
select customer_id from customer group by customer_id having count(distinct
product_key) = (select count(*) from product);
```

- 1731. The Number of Employees Which Report to Each Employee

```
select e.employee_id, e.name, count(m.employee_id) as reports_count,
round(avg(m.age * 1.0), 0) as average_age from Employees e join Employees m
on e.employee_id = m.reports_to group by e.employee_id, e.name order by
e.employee_id, e.name;
```

- 1789. Primary Department for Each Employee

```
with emp as( select e.employee_id, e.department_id from Employee e where
primary_flag = 'Y' ) select employee_id, department_id from Employee where
primary_flag = 'Y' or employee_id not in(select employee_id from emp);
```

- 610. Triangle Judgement

```
select x, y, z, case when x + y > z and x + z > y and y + z > x then 'Yes'
else 'No' end as triangle from Triangle;
```

- 180. Consecutive Numbers

```
select distinct num as ConsecutiveNums from( select num, lead(num) over
(order by id) as nex, lag(num) over (order by id) as prev from Logs ) tabl
where num = nex and num = prev;
```

- 1164. Product Price at a Given Date

- 1204. Last Person to Fit in the Bus

```
with sun_Weight as( select * , sum(weight) over (order by turn) as sunWeight
from Queue ) select top 1 person_name from sun_Weight where sunWeight <= 1000
order by turn desc;
```

- 1907. Count Salary Categories

```
select 'Low Salary' as category, count(*) as accounts_count from Accounts
where income < 20000 union all select 'Average Salary' as category, count(*)
as accounts_count from Accounts where income between 20000 and 50000 union
all select 'High Salary' as category, count(*) as accounts_count from
Accounts where income > 50000;
```

- 1978. Employees Whose Manager Left the Company

```
select employee_id from employees where salary < 30000 and manager_id not in
(select employee_id from employees) order by employee_id;
```

- 626. Exchange Seats

```
select id, case when id % 2 = 1 and id + 1 <= (select max(id) from seat)
then (select student from seat where seat.id = s.id + 1) when id % 2 = 0 then
(select student from seat where seat.id = s.id - 1) else student end as
student from seat as s order by id;
```

- List

- 1667. Fix Names in a Table

```
select user_id, upper(substring(name, 1, 1)) + lower(substring(name, 2,
len(name)-1)) as name from users order by user_id;
```

- 1527. Patients With a Condition

```
select * from Patients where conditions like 'DIAB1%' or conditions like '%
DIAB1' or conditions like '% DIAB1%'
```

- 196. Delete Duplicate Emails

```
delete from Person where id not in( select min(id) from Person group by email
);
```

- 176. Second Highest Salary

```
with rangeSalary as( select salary, dense_rank() over (order by salary desc)
as ranke from Employee ) select max(salary) as SecondHighestSalary from
rangeSalary where ranke = 2;
```

- 1484. Group Sold Products By The Date

```
with uniproduct as( select distinct sell_date, product from Activities )
select sell_date, count(product) as num_sold, string_agg(product, ',') within
group (order by product) as products from uniproduct group by sell_date order
by sell_date;
```