Following notebook contains solution for the 30 days of pandas problem which was on leetcode Using SQL(MySQL and PgSQL). https://leetcode.com/studyplan/30-days-of-pandas/

595. Big Countries

Easy Topics Companies SQL Schema Pandas Schema Table: World

+-------------+---------+ | Column Name | Type | +-------------+---------+ | name | varchar | | continent | varchar | | area | int | | population | int | | gdp | bigint | +-------------+---------+ name is the primary key (column with unique values) for this table. Each row of this table gives information about the name of a country, the continent to which it belongs, its area, the population, and its GDP value.

A country is big if:

it has an area of at least three million (i.e., 3000000 km2), or it has a population of at least twenty-five million (i.e., 25000000). Write a solution to find the name, population, and area of the big countries.

Return the result table in any order.

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

1757. Recyclable and Low Fat Products

Easy Topics Companies SQL Schema Pandas Schema Table: Products

+-------------+---------+ | Column Name | Type | +-------------+---------+ | product_id | int | | low_fats | enum | | recyclable | enum | +-------------+---------+ product_id is the primary key (column with unique values) for this table. low_fats is an ENUM (category) of type ('Y', 'N') where 'Y' means this product is low fat and 'N' means it is not. recyclable is an ENUM (category) of types ('Y', 'N') where 'Y' means this product is recyclable and 'N' means it is not.

Write a solution to find the ids of products that are both low fat and recyclable.

Return the result table in any order.

```
# Write your PostgreSQL query statement below
select product_id from products where low_fats ='Y' and recyclable ='Y';
```

183. Customers Who Never Order

Easy Topics Companies SQL Schema Pandas Schema Table: Customers

+-------------+---------+ | Column Name | Type | +-------------+---------+ | id | int | | name | varchar | +-------------+---------+ id is the primary key (column with unique values) for this table. Each row of this table indicates the ID and name of a customer.

Table: Orders

+-------------+------+ | Column Name | Type | +-------------+------+ | id | int | | customerId | int | +-------------+------+ id is the primary key (column with unique values) for this table. customerId is a foreign key (reference columns) of the ID from the Customers table. Each row of this table indicates the ID of an order and the ID of the customer who ordered it.

Write a solution to find all customers who never order anything.

Return the result table in any order.

```
SELECT Customers.name as Customers FROM Customers LEFT JOIN
Orders ON Customers.id=Orders.customerId
WHERE Orders.CustomerId is NULL
```

1148. Article Views I

Easy Topics Companies SQL Schema Pandas Schema Table: Views

+---------------+---------+ | Column Name | Type | +---------------+---------+ | article_id | int | | author_id | int | | viewer_id | int | | view_date | date | +---------------+---------+ There is no primary key (column with unique values) for this table, the table may have duplicate rows. Each row of this table indicates that some viewer viewed an article (written by some author) on some date. Note that equal author_id and viewer_id indicate the same person.

Write a solution to find all the authors that viewed at least one of their own articles.

Return the result table sorted by id in ascending order.

```
# 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

Easy Topics Companies SQL Schema Pandas Schema Table: Tweets

+----------------+---------+ | Column Name | Type | +----------------+---------+ | tweet_id | int | | content | varchar | +----------------+---------+ tweet_id is the primary key (column with unique values) for this table. This table contains all the tweets in a social media app.

Write a solution to find the IDs of the invalid tweets. The tweet is invalid if the number of characters used in the content of the tweet is strictly greater than 15.

Return the result table in any order.

```
# Write your MySQL query statement below
select tweet_id from Tweets where length(content) > 15;
```

1873. Calculate Special Bonus

Easy Topics Companies SQL Schema Pandas Schema Table: Employees

+-------------+---------+ | Column Name | Type | +-------------+---------+ | employee_id | int | | name | varchar | | salary | int | +-------------+---------+ employee_id is the primary key (column with unique values) for this table. Each row of this table indicates the employee ID, employee name, and salary.

Write a solution to calculate the bonus of each employee. The bonus of an employee is 100% of their salary if the ID of the employee is an odd number and the employee's name does not start with the character 'M'. The bonus of an employee is 0 otherwise.

Return the result table ordered by employee_id.

```
select employee_id, if(employee_id%2!=0 and name not like 'M%',salary,0) as bonus
from Employees
order by employee_id
```

1667. Fix Names in a Table

Easy Topics Companies SQL Schema Pandas Schema Table: Users

+----------------+---------+ | Column Name | Type | +----------------+---------+ | user_id | int | | name | varchar | +----------------+---------+ user_id is the primary key (column with unique values) for this table. This table contains the ID and the name of the user. The name consists of only lowercase and uppercase characters.

Write a solution to fix the names so that only the first character is uppercase and the rest are lowercase.

Return the result table ordered by user_id.

```
# Write your MySQL query statement below
SELECT
    user_id,
    CONCAT(UPPER(SUBSTRING(name, 1, 1)), LOWER(SUBSTRING(name, 2))) as name
FROM users
ORDER BY user_id;
```

1517. Find Users With Valid E-Mails

Easy Topics Companies SQL Schema Pandas Schema Table: Users

+---------------+---------+ | Column Name | Type | +---------------+---------+ | user_id | int | | name | varchar | | mail | varchar | +---------------+---------+ user_id is the primary key (column with unique values) for this table. This table contains information of the users signed up in a website. Some e-mails are invalid.

Write a solution to find the users who have valid emails.

A valid e-mail has a prefix name and a domain where:

The prefix name is a string that may contain letters (upper or lower case), digits, underscore '_', period '.', and/or dash '-'. The prefix name must start with a letter. The domain is '@leetcode.com'. Return the result table in any order.

```
SELECT *
FROM Users
WHERE mail REGEXP '^[A-Za-z][A-Za-z0-9_\.\-]*@leetcode\\.com$';
```

1527. Patients With a Condition

Easy Topics Companies SQL Schema Pandas Schema Table: Patients

+--------------+---------+ | Column Name | Type | +--------------+---------+ | patient_id | int | | patient_name | varchar | | conditions | varchar | +--------------+---------+ patient_id is the primary key (column with unique values) for this table. 'conditions' contains 0 or more code separated by spaces. This table contains information of the patients in the hospital.

Write a solution to find the patient_id, patient_name, and conditions of the patients who have Type I Diabetes. Type I Diabetes always starts with DIAB1 prefix.

Return the result table in any order.

```
SELECT patient_id, patient_name, conditions
FROM Patients
WHERE conditions LIKE 'DIAB1%' OR conditions LIKE '% DIAB1%';
```

177. Nth Highest Salary

Medium Topics Companies SQL Schema Pandas Schema Table: Employee

+-------------+------+ | Column Name | Type | +-------------+------+ | id | int | | salary | int | +-------------+------+ id is the primary key (column with unique values) for this table. Each row of this table contains information about the salary of an employee.

Write a solution to find the nth highest salary from the Employee table. If there is no nth highest salary, return null.

```
CREATE FUNCTION getNthHighestSalary(N INT) RETURNS INT
BEGIN
set N=N-1;
  RETURN (
      # Write your MySQL query statement below.
select distinct salary from employee order by salary desc limit N,1
  );
END
```

176. Second Highest Salary

Medium Topics Companies SQL Schema Pandas Schema Table: Employee

+-------------+------+ | Column Name | Type | +-------------+------+ | id | int | | salary | int | +-------------+------+ id is the primary key (column with unique values) for this table. Each row of this table contains information about the salary of an employee.

Write a solution to find the second highest salary from the Employee table. If there is no second highest salary, return null (return None in Pandas).

```
SELECT  MAX(SALARY) AS SecondHighestSalary FROM EMPLOYEE WHERE SALARY <(SELECT MAX(SALARY) FROM EMPLOYEE);
```

184. Department Highest Salary

Medium Topics Companies SQL Schema Pandas Schema Table: Employee

+--------------+---------+ | Column Name | Type | +--------------+---------+ | id | int | | name | varchar | | salary | int | | departmentId | int | +--------------+---------+ id is the primary key (column with unique values) for this table. departmentId is a foreign key (reference columns) of the ID from the Department table. Each row of this table indicates the ID, name, and salary of an employee. It also contains the ID of their department.

Table: Department

+-------------+---------+ | Column Name | Type | +-------------+---------+ | id | int | | name | varchar | +-------------+---------+ id is the primary key (column with unique values) for this table. It is guaranteed that department name is not NULL. Each row of this table indicates the ID of a department and its name.

Write a solution to find employees who have the highest salary in each of the departments.

Return the result table in any order.

```
select department.name as department ,employee.name as employee ,employee.salary as salary from department ,employee
where department.id=employee.departmentid and (employee.departmentid ,salary)in
 (select departmentid ,max(salary)from employee group by departmentid);
```

178. Rank Scores

Medium Topics Companies SQL Schema Pandas Schema Table: Scores

+-------------+---------+ | Column Name | Type | +-------------+---------+ | id | int | | score | decimal | +-------------+---------+ id is the primary key (column with unique values) for this table. Each row of this table contains the score of a game. Score is a floating point value with two decimal places.

Write a solution to find the rank of the scores. The ranking should be calculated according to the following rules:

The scores should be ranked from the highest to the lowest. If there is a tie between two scores, both should have the same ranking. After a tie, the next ranking number should be the next consecutive integer value. In other words, there should be no holes between ranks. Return the result table ordered by score in descending order.

```
select score ,dense_rank () over (order by score desc) as rank from scores;
```

196. Delete Duplicate Emails

Easy Topics Companies SQL Schema Pandas Schema Table: Person

+-------------+---------+ | Column Name | Type | +-------------+---------+ | id | int | | email | varchar | +-------------+---------+ id is the primary key (column with unique values) for this table. Each row of this table contains an email. The emails will not contain uppercase letters.

Write a solution to delete all duplicate emails, keeping only one unique email with the smallest id.

For SQL users, please note that you are supposed to write a DELETE statement and not a SELECT one.

For Pandas users, please note that you are supposed to modify Person in place.

After running your script, the answer shown is the Person table. The driver will first compile and run your piece of code and then show the Person table. The final order of the Person table does not matter.

```
delete A from person A,person B where A.id>B.id and A.email=B.email;
```

1795. Rearrange Products Table

Easy Topics Companies SQL Schema Pandas Schema Table: Products

+-------------+---------+ | Column Name | Type | +-------------+---------+ | product_id | int | | store1 | int | | store2 | int | | store3 | int | +-------------+---------+ product_id is the primary key (column with unique values) for this table. Each row in this table indicates the product's price in 3 different stores: store1, store2, and store3. If the product is not available in a store, the price will be null in that store's column.

Write a solution to rearrange the Products table so that each row has (product_id, store, price). If a product is not available in a store, do not include a row with that product_id and store combination in the result table.

Return the result table in any order.

```
select product_id,'store1' as store ,store1 as price from products where store1 is not null
union
select product_id ,'store2' as store,store2 as price from products where store2 is not null
union
select product_id ,'store3' as store,store3 as price from products where store3 is not null
```

1907. Count Salary Categories

Medium Topics Companies SQL Schema Pandas Schema Table: Accounts

+-------------+------+ | Column Name | Type | +-------------+------+ | account_id | int | | income | int | +-------------+------+ account_id is the primary key (column with unique values) for this table. Each row contains information about the monthly income for one bank account.

Write a solution to calculate the number of bank accounts for each salary category. The salary categories are:

"Low Salary": All the salaries strictly less than $20000.''AverageSalary'': Allthesalariesintheinclusiverange[$20000, $50000].''HighSalary'': Allthesalariesstrictlygreaterthan$50000. The result table must contain all three categories. If there are no accounts in a category, return 0.

Return the result table in any order.

```
# Write your MySQL query statement below
SELECT 'Low Salary' AS category, COUNT(*) AS accounts_count
FROM Accounts
WHERE income < 20000
UNION
SELECT 'Average Salary' AS category, COUNT(*) AS accounts_count
FROM Accounts
WHERE income BETWEEN 20000 AND 50000
UNION
SELECT 'High Salary' AS category, COUNT(*) AS accounts_count
FROM Accounts
WHERE income > 50000;
```

1741. Find Total Time Spent by Each Employee

Easy Topics Companies SQL Schema Pandas Schema Table: Employees

+-------------+------+ | Column Name | Type | +-------------+------+ | emp_id | int | | event_day | date | | in_time | int | | out_time | int | +-------------+------+ (emp_id, event_day, in_time) is the primary key (combinations of columns with unique values) of this table. The table shows the employees' entries and exits in an office. event_day is the day at which this event happened, in_time is the minute at which the employee entered the office, and out_time is the minute at which they left the office. in_time and out_time are between 1 and 1440. It is guaranteed that no two events on the same day intersect in time, and in_time < out_time.

Write a solution to calculate the total time in minutes spent by each employee on each day at the office. Note that within one day, an employee can enter and leave more than once. The time spent in the office for a single entry is out_time - in_time.

Return the result table in any order.

```
# Write your MySQL query statement below
select event_day as day,emp_id,(sum(out_time-in_time))as total_time from employees group by event_day,emp_id;
```

511. Game Play Analysis I

Easy Topics Companies SQL Schema Pandas Schema Table: Activity

+--------------+---------+ | Column Name | Type | +--------------+---------+ | player_id | int | | device_id | int | | event_date | date | | games_played | int | +--------------+---------+ (player_id, event_date) is the primary key (combination of columns with unique values) of this table. This table shows the activity of players of some games. Each row is a record of a player who logged in and played a number of games (possibly 0) before logging out on someday using some device.

Write a solution to find the first login date for each player.

Return the result table in any order.

```
# Write your MySQL query statement below
SELECT player_id,
MIN(event_date) AS first_login
FROM activity
GROUP BY player_id;
```

2356. Number of Unique Subjects Taught by Each Teacher

Easy Topics Companies SQL Schema Pandas Schema Table: Teacher

+-------------+------+ | Column Name | Type | +-------------+------+ | teacher_id | int | | subject_id | int | | dept_id | int | +-------------+------+ (subject_id, dept_id) is the primary key (combinations of columns with unique values) of this table. Each row in this table indicates that the teacher with teacher_id teaches the subject subject_id in the department dept_id.

Write a solution to calculate the number of unique subjects each teacher teaches in the university.

Return the result table in any order.

```
# Write your MySQL query statement below
select teacher_id ,count(distinct subject_id)as cnt from teacher group by teacher_id;
```

596. Classes More Than 5 Students

Easy Topics Companies SQL Schema Pandas Schema Table: Courses

+-------------+---------+ | Column Name | Type | +-------------+---------+ | student | varchar | | class | varchar | +-------------+---------+ (student, class) is the primary key (combination of columns with unique values) for this table. Each row of this table indicates the name of a student and the class in which they are enrolled.

Write a solution to find all the classes that have at least five students.

Return the result table in any order.

```
# Write your MySQL query statement below
select class from courses group by class having count(student)>4;
```

586. Customer Placing the Largest Number of Orders

Easy Topics Companies Hint SQL Schema Pandas Schema Table: Orders

+-----------------+----------+ | Column Name | Type | +-----------------+----------+ | order_number | int | | customer_number | int | +-----------------+----------+ order_number is the primary key (column with unique values) for this table. This table contains information about the order ID and the customer ID.

Write a solution to find the customer_number for the customer who has placed the largest number of orders.

The test cases are generated so that exactly one customer will have placed more orders than any other customer.

```
-- Write your PostgreSQL query statement below
select customer_number from orders group by customer_number order by count(customer_number) desc limit 1;
```

1484. Group Sold Products By The Date

Easy Topics Companies SQL Schema Pandas Schema Table Activities:

+-------------+---------+ | Column Name | Type | +-------------+---------+ | sell_date | date | | product | varchar | +-------------+---------+ There is no primary key (column with unique values) for this table. It may contain duplicates. Each row of this table contains the product name and the date it was sold in a market.

Write a solution to find for each date the number of different products sold and their names.

The sold products names for each date should be sorted lexicographically.

Return the result table ordered by sell_date.

```
# Write your MySQL query statement below
select sell_date,
  count(distinct product)as num_sold,
  group_concat(distinct product order by product) as products
 from activities
 group by sell_date
 order by sell_date;
```

1693. Daily Leads and Partners

Easy Topics Companies SQL Schema Pandas Schema Table: DailySales

+-------------+---------+ | Column Name | Type | +-------------+---------+ | date_id | date | | make_name | varchar | | lead_id | int | | partner_id | int | +-------------+---------+ There is no primary key (column with unique values) for this table. It may contain duplicates. This table contains the date and the name of the product sold and the IDs of the lead and partner it was sold to. The name consists of only lowercase English letters.

For each date_id and make_name, find the number of distinct lead_id's and distinct partner_id's.

Return the result table in any order.

```
# Write your MySQL query statement below
select date_id,make_name ,
  count(distinct lead_id )as unique_leads,
  count(distinct partner_id)as unique_partners
from dailysales
group by date_id,make_name ;
```

1050. Actors and Directors Who Cooperated At Least Three Times

Easy Topics Companies SQL Schema Pandas Schema Table: ActorDirector

+-------------+---------+ | Column Name | Type | +-------------+---------+ | actor_id | int | | director_id | int | | timestamp | int | +-------------+---------+ timestamp is the primary key (column with unique values) for this table.

Write a solution to find all the pairs (actor_id, director_id) where the actor has cooperated with the director at least three times.

Return the result table in any order.

```
# Write your MySQL query statement below
select actor_id,director_id
from ActorDirector
group by actor_id,director_id
Having count(timestamp)>=3;
```

1378. Replace Employee ID With The Unique Identifier

Easy Topics Companies SQL Schema Pandas Schema Table: Employees

+---------------+---------+ | Column Name | Type | +---------------+---------+ | id | int | | name | varchar | +---------------+---------+ id is the primary key (column with unique values) for this table. Each row of this table contains the id and the name of an employee in a company.

Table: EmployeeUNI

+---------------+---------+ | Column Name | Type | +---------------+---------+ | id | int | | unique_id | int | +---------------+---------+ (id, unique_id) is the primary key (combination of columns with unique values) for this table. Each row of this table contains the id and the corresponding unique id of an employee in the company.

Write a solution to show the unique ID of each user, If a user does not have a unique ID replace just show null.

Return the result table in any order.

```
# Write your MySQL query statement below
select employeeuni.unique_id,employees.name from employeeuni
right join employees on employeeuni.id=employees.id;
```

1280. Students and Examinations

Easy Topics Companies SQL Schema Pandas Schema Table: Students

+---------------+---------+ | Column Name | Type | +---------------+---------+ | student_id | int | | student_name | varchar | +---------------+---------+ student_id is the primary key (column with unique values) for this table. Each row of this table contains the ID and the name of one student in the school.

Table: Subjects

+--------------+---------+ | Column Name | Type | +--------------+---------+ | subject_name | varchar | +--------------+---------+ subject_name is the primary key (column with unique values) for this table. Each row of this table contains the name of one subject in the school.

Table: Examinations

+--------------+---------+ | Column Name | Type | +--------------+---------+ | student_id | int | | subject_name | varchar | +--------------+---------+ There is no primary key (column with unique values) for this table. It may contain duplicates. Each student from the Students table takes every course from the Subjects table. Each row of this table indicates that a student with ID student_id attended the exam of subject_name.

Write a solution to find the number of times each student attended each exam.

Return the result table ordered by student_id and subject_name.

```
# Write your MySQL query statement below
select students.student_id,students.student_name ,subjects.subject_name, count(examinations.subject_name)as attended_exams
from students
join subjects
left join examinations on examinations.subject_name=subjects.subject_name and examinations.student_id=students.student_id
group by students.student_id ,subjects.subject_name
order by students.student_id,subjects.subject_name;
```

570. Managers with at Least 5 Direct Reports

Medium Topics Companies Hint SQL Schema Pandas Schema Table: Employee

+-------------+---------+ | Column Name | Type | +-------------+---------+ | id | int | | name | varchar | | department | varchar | | managerId | int | +-------------+---------+ id is the primary key (column with unique values) for this table. Each row of this table indicates the name of an employee, their department, and the id of their manager. If managerId is null, then the employee does not have a manager. No employee will be the manager of themself.

Write a solution to find managers with at least five direct reports.

Return the result table in any order.

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

607. Sales Person

Easy Topics Companies Hint SQL Schema Pandas Schema Table: SalesPerson

+-----------------+---------+ | Column Name | Type | +-----------------+---------+ | sales_id | int | | name | varchar | | salary | int | | commission_rate | int | | hire_date | date | +-----------------+---------+ sales_id is the primary key (column with unique values) for this table. Each row of this table indicates the name and the ID of a salesperson alongside their salary, commission rate, and hire date.

Table: Company

+-------------+---------+ | Column Name | Type | +-------------+---------+ | com_id | int | | name | varchar | | city | varchar | +-------------+---------+ com_id is the primary key (column with unique values) for this table. Each row of this table indicates the name and the ID of a company and the city in which the company is located.

Table: Orders

+-------------+------+ | Column Name | Type | +-------------+------+ | order_id | int | | order_date | date | | com_id | int | | sales_id | int | | amount | int | +-------------+------+ order_id is the primary key (column with unique values) for this table. com_id is a foreign key (reference column) to com_id from the Company table. sales_id is a foreign key (reference column) to sales_id from the SalesPerson table. Each row of this table contains information about one order. This includes the ID of the company, the ID of the salesperson, the date of the order, and the amount paid.

Write a solution to find the names of all the salespersons who did not have any orders related to the company with the name "RED".

Return the result table in any order

```
# Write your MySQL query statement below
select s.name from salesperson s
where s.sales_id not in (
    select o.sales_id
    from orders o
    join company c
    on c.com_id=o.com_id
    where c.name="red"
)
```

Double-click (or enter) to edit