584. Find Customer Referee

Easy 34 15 Favorite Share

SQL Schema ›

Given a table `customer` holding customers information and the referee.

```
+------+------+-----------+
| id   | name | referee_id|
+------+------+-----------+
|    1 | Will |      NULL |
|    2 | Jane |      NULL |
|    3 | Alex |         2 |
|    4 | Bill |      NULL |
|    5 | Zack |         1 |
|    6 | Mark |         2 |
+------+------+-----------+
```

Write a query to return the list of customers **NOT** referred by the person with id '2'.

For the sample data above, the result is:

```
+------+
| name |
+------+
| Will |
| Jane |
| Bill |
| Zack |
+------+
```

Accepted 9,314 | Submissions 14,229

```sql
SELECT name
FROM customer
WHERE referee_id <> 2 OR referee_id IS NULL
```

585. Investments in 2016

Medium 39 31 Favorite Share

SQL Schema ›

Write a query to print the sum of all total investment values in 2016 (**TIV_2016**), to a scale of 2 decimal places, for all policy holders who meet the following criteria:

1. Have the same **TIV_2015** value as one or more other policyholders.
2. Are not located in the same city as any other policyholder (i.e.: the (latitude, longitude) attribute pairs must be unique).

Input Format:
The ***insurance*** table is described as follows:

Column Name	Type
PID	INTEGER(11)
TIV_2015	NUMERIC(15,2)
TIV_2016	NUMERIC(15,2)
LAT	NUMERIC(5,2)
LON	NUMERIC(5,2)

where **PID** is the policyholder's policy ID, **TIV_2015** is the total investment value in 2015, **TIV_2016** is the total investment value in 2016, **LAT** is the latitude of the policy holder's city, and **LON** is the longitude of the policy holder's city.

Sample Input

PID	TIV_2015	TIV_2016	LAT	LON
1	10	5	10	10
2	20	20	20	20
3	10	30	20	20
4	10	40	40	40

Sample Output

TIV_2016
45.00

Explanation

The first record in the table, like the last record, meets both of the two criteria.
The **TIV_2015** value '10' is as the same as the third and forth record, and its location unique.

The second record does not meet any of the two criteria. Its **TIV_2015** is not like any other policyholders.

And its location is the same with the third record, which makes the third record fail, too.

So, the result is the sum of **TIV_2016** of the first and last record, which is 45.

```
SELECT name
FROM customer
WHERE referee_id <> 2 OR referee_id IS NULL
```

586. Customer Placing the Largest Number of Orders

Easy 52 3 Favorite Share

SQL Schema ›

Query the **customer_number** from the ***orders*** table for the customer who has placed the largest number of orders.

It is guaranteed that exactly one customer will have placed more orders than any other customer.

The ***orders*** table is defined as follows:

```
| Column            | Type      |
|-------------------|-----------|
| order_number (PK) | int       |
| customer_number   | int       |
| order_date        | date      |
| required_date     | date      |
| shipped_date      | date      |
| status            | char(15)  |
| comment           | char(200) |
```

Sample Input

```
| order_number | customer_number | order_date |
required_date | shipped_date | status | comment |
|--------------|-----------------|------------|--
-------------|--------------|--------|---------|
| 1            | 1               | 2017-04-09 |
2017-04-13    | 2017-04-12   | Closed |         |
| 2            | 2               | 2017-04-15 |
2017-04-20    | 2017-04-18   | Closed |         |
| 3            | 3               | 2017-04-16 |
2017-04-25    | 2017-04-20   | Closed |         |
| 4            | 3               | 2017-04-18 |
2017-04-28    | 2017-04-25   | Closed |         |
```

Sample Output

```
| customer_number |
|-----------------|
| 3               |
```

Explanation

```
The customer with number '3' has two orders,
which is greater than either customer '1' or '2'
because each of them  only has one order.
So the result is customer_number '3'.
```

Follow up: *What if more than one customer have the largest number of orders, can you find all the customer_number in this case?*

Accepted 10,299 | Submissions 16,024

```mysql
# Write your MySQL query statement below
select customer_number from orders
group by customer_number
order by count(*) desc limit 1;
```

596. Classes More Than 5 Students

Easy 132 345 Favorite Share

SQL Schema ›

There is a table `courses` with columns: **student** and **class**

Please list out all classes which have more than or equal to 5 students.

For example, the table:

```
+---------+------------+
| student | class      |
+---------+------------+
| A       | Math       |
| B       | English    |
| C       | Math       |
| D       | Biology    |
| E       | Math       |
| F       | Computer   |
| G       | Math       |
| H       | Math       |
| I       | Math       |
+---------+------------+
```

Should output:

```
+---------+
| class   |
+---------+
| Math    |
+---------+
```

Note:
The students should not be counted duplicate in each course.

Accepted 33,285 | Submissions 97,098

```mysql
# Write your MySQL query statement below
select class from courses group by class having count(distinct student) >= 5;
```

601. Human Traffic of Stadium

Hard 65 146 Favorite Share

SQL Schema ›

X city built a new stadium, each day many people visit it and the stats are saved as these columns: **id**, **date**, **people**

Please write a query to display the records which have 3 or more consecutive rows and the amount of people more than 100(inclusive).

For example, the table `stadium`:

```
+------+------------+-----------+
| id   | date       | people    |
+------+------------+-----------+
| 1    | 2017-01-01 | 10        |
| 2    | 2017-01-02 | 109       |
| 3    | 2017-01-03 | 150       |
| 4    | 2017-01-04 | 99        |
| 5    | 2017-01-05 | 145       |
| 6    | 2017-01-06 | 1455      |
| 7    | 2017-01-07 | 199       |
| 8    | 2017-01-08 | 188       |
+------+------------+-----------+
```

For the sample data above, the output is:

```
+------+------------+-----------+
| id   | date       | people    |
+------+------------+-----------+
| 5    | 2017-01-05 | 145       |
| 6    | 2017-01-06 | 1455      |
| 7    | 2017-01-07 | 199       |
| 8    | 2017-01-08 | 188       |
+------+------------+-----------+
```

Note:
Each day only have one row record, and the dates are increasing with id increasing.

Accepted 10,631 | Submissions 30,404

```sql
SELECT s1.* FROM stadium AS s1, stadium AS s2, stadium as s3
    WHERE
    ((s1.id + 1 = s2.id
    AND s1.id + 2 = s3.id)
    OR
    (s1.id - 1 = s2.id
    AND s1.id + 1 = s3.id)
    OR
    (s1.id - 2 = s2.id
    AND s1.id - 1 = s3.id)
    )
    AND s1.people>=100
    AND s2.people>=100
    AND s3.people>=100

    GROUP BY s1.id
```

602. Friend Requests II: Who Has the Most Friends

Medium 64 26 Favorite Share

SQL Schema ›

In social network like Facebook or Twitter, people send friend requests and accept others' requests as well.

Table `request_accepted` holds the data of friend acceptance, while **requester_id** and **accepter_id** both are the id of a person.

```
| requester_id | accepter_id | accept_date|
|--------------|-------------|------------|
| 1            | 2           | 2016_06-03 |
| 1            | 3           | 2016-06-08 |
| 2            | 3           | 2016-06-08 |
| 3            | 4           | 2016-06-09 |
```

Write a query to find the the people who has most friends and the most friends number. For the sample data above, the result is:

```
| id | num |
|----|-----|
| 3  | 3   |
```

Note:

- It is guaranteed there is only 1 people having the most friends.
- The friend request could only been accepted once, which mean there is no multiple records with the same **requester_id** and **accepter_id** value.

Explanation:
The person with id '3' is a friend of people '1', '2' and '4', so he has 3 friends in total, which is the most number than any others.

Follow-up:
In the real world, multiple people could have the same most number of friends, can you find all these people in this case?

Accepted 7,880 | Submissions 18,203

```
select id1 as id, count(id2) as num
from
(select requester_id as id1, accepter_id as id2
from request_accepted
union
select accepter_id as id1, requester_id as id2
from request_accepted) tmp1
group by id1
order by num desc limit 1
```

607. Sales Person

Easy 49 7 Favorite Share

SQL Schema ›

Description

Given three tables: `salesperson`, `company`, `orders`.
Output all the **names** in the table `salesperson`, who didn't have sales to company 'RED'.

Example
Input

Table: `salesperson`

```
+----------+------+--------+-----------------+---
--------+
| sales_id | name | salary | commission_rate |
hire_date |
+----------+------+--------+-----------------+---
--------+
|   1      | John | 100000 |     6           |
4/1/2006  |
|   2      | Amy  | 120000 |     5           |
5/1/2010  |
|   3      | Mark | 65000  |     12          |
12/25/2008|
|   4      | Pam  | 25000  |     25          |
1/1/2005  |
|   5      | Alex | 50000  |     10          |
2/3/2007  |
+----------+------+--------+-----------------+---
--------+
```

The table `salesperson` holds the salesperson information. Every salesperson has a **sales_id** and a **name**.
Table: `company`

```
+---------+--------+------------+
| com_id  |  name  |    city    |
+---------+--------+------------+
|   1     |  RED   |   Boston   |
|   2     | ORANGE |   New York |
|   3     | YELLOW |   Boston   |
|   4     | GREEN  |   Austin   |
+---------+--------+------------+
```

The table `company` holds the company information. Every company has a **com_id** and a **name**.
Table: `orders`

```
+----------+----------+---------+----------+-----
---+
| order_id |  date    | com_id  | sales_id |
amount |
+----------+----------+---------+----------+-----
---+
| 1        | 1/1/2014 |    3    |    4     |
100000 |
| 2        | 2/1/2014 |    4    |    5     | 5000
|
| 3        | 3/1/2014 |    1    |    1     |
50000  |
| 4        | 4/1/2014 |    1    |    4     |
25000  |
+----------+----------+---------+----------+-----
---+
```

The table `orders` holds the sales record information, salesperson and customer company are represented by **sales_id** and **com_id**.
output

```
+------+
| name |
+------+
| Amy  |
| Mark |
| Alex |
+------+
```

Explanation

According to order '3' and '4' in table `orders`, it is easy to tell only salesperson 'John' and 'Alex' have sales to company 'RED',
so we need to output all the other **names** in table `salesperson`.

Accepted 7,995 | Submissions 14,891

```
# Write your MySQL query statement below
select salesperson.name
from orders o join company c on (o.com_id = c.com_id and c.name = 'RED')
right join salesperson on salesperson.sales_id = o.sales_id
where o.sales_id is null
```

612. Shortest Distance in a Plane

Medium 38 6 Favorite Share

SQL Schema ›

Table `point_2d` holds the coordinates (x,y) of some unique points (more than two) in a plane.

Write a query to find the shortest distance between these points rounded to 2 decimals.

```
| x  | y  |
|----|----|
| -1 | -1 |
| 0  | 0  |
| -1 | -2 |
```

The shortest distance is 1.00 from point (-1,-1) to (-1,2). So the output should be:

```
| shortest |
|----------|
| 1.00     |
```

Note: The longest distance among all the points are less than 10000.

Accepted 5,313 | Submissions 10,155

```
select round(sqrt(min(pow(a.x-b.x,2)+pow(a.y-b.y,2))),2) shortest
from point_2d a, point_2d b
where (a.x,a.y)!=(b.x,b.y)
```

613. Shortest Distance in a Line

Easy 71 10 Favorite Share

SQL Schema ›

Table `point` holds the x coordinate of some points on x-axis in a plane, which are all integers.

Write a query to find the shortest distance between two points in these points.

```
| x   |
|-----|
| -1  |
| 0   |
| 2   |
```

The shortest distance is '1' obviously, which is from point '-1' to '0'. So the output is as below:

```
| shortest|
|---------|
| 1       |
```

Note: Every point is unique, which means there is no duplicates in table `point`.

Follow-up: What if all these points have an id and are arranged from the left most to the right most of x axis?

Accepted 10,264 | Submissions 14,298

```
# Write your MySQL query statement below
SELECT MIN(ABS(P1.x - P2.x)) AS shortest FROM point AS P1
JOIN point AS P2 ON P1.x <> P2.x
```

614. Second Degree Follower

Medium 20 186 Favorite Share

SQL Schema ›

In facebook, there is a `follow` table with two columns: **followee**, **follower**.

Please write a sql query to get the amount of each follower's follower if he/she has one.

For example:

```
+-------------+------------+
| followee    | follower   |
+-------------+------------+
|     A       |     B      |
|     B       |     C      |
|     B       |     D      |
|     D       |     E      |
+-------------+------------+
```

should output:

```
+-------------+------------+
| follower    | num        |
+-------------+------------+
|     B       |  2         |
|     D       |  1         |
+-------------+------------+
```

Explaination:
Both B and D exist in the follower list, when as a followee, B's follower is C and D, and D's follower is E. A does not exist in follower list.

Note:
Followee would not follow himself/herself in all cases.
Please display the result in follower's alphabet order.

Accepted 4,699 | Submissions 20,611

```sql
select distinct follower, num
from follow,
(select followee, count(distinct follower) as num from follow
group by followee) as t
where follower = t.followee
order by follower;
```

```sql
select f1.follower, count(distinct f2.follower) as num
from follow f1
join follow f2 on f1.follower = f2.followee
group by f1.follower
order by f1.follower;
```

618. Students Report By Geography

Hard 12 40 Favorite Share

SQL Schema ›

A U.S graduate school has students from Asia, Europe and America. The students' location information are stored in table `student` as below.

```
| name   | continent |
|--------|-----------|
| Jack   | America   |
| Pascal | Europe    |
| Xi     | Asia      |
| Jane   | America   |
```

Pivot the continent column in this table so that each name is sorted alphabetically and displayed underneath its corresponding continent. The output headers should be America, Asia and Europe respectively. It is guaranteed that the student number from America is no less than either Asia or Europe.

For the sample input, the output is:

For the sample input, the output is:

```
| America | Asia | Europe |
|---------|------|--------|
| Jack    | Xi   | Pascal |
| Jane    |      |        |
```

Follow-up: If it is unknown which continent has the most students, can you write a query to generate the student report?

Accepted 1,889 | Submissions 4,560

```mysql
# Write your MySQL query statement below
SELECT MAX(America) AS America, MAX(Asia) as Asia, MAX(Europe) AS Europe
FROM (
    SELECT
        CASE WHEN continent = 'America' THEN @r1 := @r1 + 1
             WHEN continent = 'Asia'    THEN @r2 := @r2 + 1
             WHEN continent = 'Europe'  THEN @r3 := @r3 + 1 END id,
        CASE WHEN continent = 'America' THEN name ELSE NULL END America,
        CASE WHEN continent = 'Asia'    THEN name ELSE NULL END Asia,
        CASE WHEN continent = 'Europe'  THEN name ELSE NULL END Europe
    FROM student, (SELECT @r1 := 0, @r2 := 0, @r3 := 0) AS ids
    ORDER BY name
) AS tempTable
GROUP BY id;
```

619. Biggest Single Number

Easy 28 27 Favorite Share

SQL Schema ›

Table `number` contains many numbers in column **num** including duplicated ones.
Can you write a SQL query to find the biggest number, which only appears once.

```
+---+
|num|
+---+
| 8 |
| 8 |
| 3 |
| 3 |
| 1 |
| 4 |
| 5 |
| 6 |
```

For the sample data above, your query should return the following result:

```
+---+
|num|
+---+
| 6 |
```

Note:
If there is no such number, just output **null**.

Accepted 8,178 | Submissions 21,610

```sql
select(
  select num
  from number
  group by num
  having count(*) = 1
  order by num desc limit 1
) as num;
```

620. Not Boring Movies

Easy 163 179 Favorite Share

SQL Schema ›

X city opened a new cinema, many people would like to go to this cinema. The cinema also gives out a poster indicating the movies' ratings and descriptions.
Please write a SQL query to output movies with an odd numbered ID and a description that is not 'boring'. Order the result by rating.

For example, table `cinema`:

```
+---------+-----------+--------------+-----------
+
|   id    | movie     |  description |  rating
|
+---------+-----------+--------------+-----------
+
|   1     | War       |   great 3D   |   8.9
|
|   2     | Science   |   fiction    |   8.5
|
|   3     | irish     |   boring     |   6.2
|
|   4     | Ice song  |   Fantacy    |   8.6
|
|   5     | House card|   Interesting|   9.1
|
+---------+-----------+--------------+-----------
+
```

For the example above, the output should be:

```
+---------+-----------+-------------+-----------
+
|   id    | movie     |  description |  rating
|
+---------+-----------+-------------+-----------
+
|   5     | House card|   Interesting|   9.1
|
|   1     | War       |   great 3D   |   8.9
|
+---------+-----------+-------------+-----------
+
```

Accepted 47,501 | Submissions 78,166

```sql
# Write your MySQL query statement below
SELECT *
FROM cinema
WHERE (id % 2 = 1) AND (description <> 'boring')
ORDER BY rating DESC
```

627. Swap Salary

Easy 254 176 Favorite Share

SQL Schema ›

Given a table `salary`, such as the one below, that has m=male and f=female values. Swap all f and m values (i.e., change all f values to m and vice versa) with a single update query and no intermediate temp table.

For example:

```
| id | name | sex | salary |
|----|------|-----|--------|
| 1  | A    | m   | 2500   |
| 2  | B    | f   | 1500   |
| 3  | C    | m   | 5500   |
| 4  | D    | f   | 500    |
```

For the sample data above, the output is:

```
+------+------------+-----------+
| id   | date       | people    |
+------+------------+-----------+
| 5    | 2017-01-05 | 145       |
| 6    | 2017-01-06 | 1455      |
| 7    | 2017-01-07 | 199       |
| 8    | 2017-01-08 | 188       |
+------+------------+-----------+
```

Note:
Each day only have one row record, and the dates are increasing with id increasing.

Accepted 10,631 | Submissions 30,404

```
update salary set sex = CHAR(ASCII('f') ^ ASCII('m') ^ ASCII(sex));
```

```
update salary set sex= CHAR(ASCII('f') + ASCII('m') - ASCII(sex));
```