LeetCode SQL 50

Subqueries

Created by: Bhawna Gupta

1978. Employees Whose Manager Left the Company

Input:

Employees table:

employee_id	name	manager_id	salary
3	Mila	9	60301
12	Antonella	null	31000
13	Emery	null	67084
1	Kalel	11	21241
9	Mikaela	null	50937
11	Joziah	6	28485

Output:

employee_id
11

```sql
Select
  e.employee_id
From
  Employees e
  Left Join Employees m On e.manager_id = m.employee_id
Where
  e.salary < 30000
  And e.manager_id Is Not Null
  And m.employee_id Is Null
Order By
  e.employee_id;
```

626. Exchange Seats

Input:
Seat table:

id	student
1	Abbot
2	Doris
3	Emerson
4	Green
5	Jeames

Output:

id	student
1	Doris
2	Abbot
3	Green
4	Emerson
5	Jeames

```sql
Select
  Case
  when id % 2 = 0 then id - 1
  when id % 2 = 1 and id < (select count(*) from seat) then id + 1
  else id end as id,
  student
From Seat
Order By id;
```

1341. Movie Rating

Input:

Movies table:

movie_id	title
1	Avengers
2	Frozen 2
3	Joker

Users table:

user_id	name
1	Daniel
2	Monica
3	Maria
4	James

MovieRating table:

movie_id	user_id	rating	created_at
1	1	3	2020-01-12
1	2	4	2020-02-11
1	3	2	2020-02-12
1	4	1	2020-01-01
2	1	5	2020-02-17
2	2	2	2020-02-01
2	3	2	2020-03-01
3	1	3	2020-02-22
3	2	4	2020-02-25

Output:

results
Daniel
Frozen 2

```
(Select u.name As results From Users u
Left Join MovieRating r On u.user_id=r.user_id
Group By u.user_id
Order By count(r.rating) DESC, u.name ASC
Limit 1)

Union All

(Select m.title As results From Movies m Join
MovieRating r On m.movie_id=r.movie_id
Where r.created_at LIKE "2020-02-%%"
Group By r.movie_id
Order By Avg(r.rating) DESC, m.title ASC
Limit 1);
```

1321. Restaurant Growth

Input:
Customer table:

customer_id	name	visited_on	amount
1	Jhon	2019-01-01	100
2	Daniel	2019-01-02	110
3	Jade	2019-01-03	120
4	Khaled	2019-01-04	130
5	Winston	2019-01-05	110
6	Elvis	2019-01-06	140
7	Anna	2019-01-07	150
8	Maria	2019-01-08	80
9	Jaze	2019-01-09	110
1	Jhon	2019-01-10	130
3	Jade	2019-01-10	150

Output:

visited_on	amount	average_amount
2019-01-07	860	122.86
2019-01-08	840	120
2019-01-09	840	120
2019-01-10	1000	142.86

```sql
WITH dates AS (
  SELECT DISTINCT visited_on
  FROM customer)
SELECT c1.visited_on,
  sum(c2.amount) as amount,
  round(sum(c2.amount) / 7, 2) as average_amount
FROM dates c1
  JOIN customer c2 ON
  datediff(c1.visited_on, c2.visited_on)
  between 0 and 6
WHERE datediff(c1.visited_on,
    (SELECT min(visited_on) FROM customer)) >= 6
GROUP BY c1.visited_on
ORDER BY c1.visited_on;
```

602. Friend Requests II: Who Has the Most Friends

Input:

RequestAccepted table:

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

Output:

id	num
3	3

```
With CTE as
(Select requester_id As id1
  From RequestAccepted
  Union All
  Select accepter_id As id1
  From RequestAccepted)
Select id1 As id, count(id1) As num
From CTE
Group By id1
Order by num DESC
Limit 1;
```

585. Investments in 2016

Input:
Insurance table:

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

Output:

tiv_2016
45.00

```sql
Select
  round(sum(tiv_2016), 2) As tiv_2016
From Insurance
Where tiv_2015 IN
    (Select tiv_2015 from Insurance
    group by tiv_2015 having count(*)> 1)
  And (lat, lon) In
  (Select lat, lon from Insurance
    group by lat, lon
    having count(*)= 1)
```

185. Department Top Three Salaries

Input:

Employee table:

id	name	salary	departmentId
1	Joe	85000	1
2	Henry	80000	2
3	Sam	60000	2
4	Max	90000	1
5	Janet	69000	1
6	Randy	85000	1
7	Will	70000	1

Department table:

id	name
1	IT
2	Sales

Output:

Department	Employee	Salary
IT	Max	90000
IT	Joe	85000
IT	Randy	85000
IT	Will	70000
Sales	Henry	80000
Sales	Sam	60000

```
Select
  r.Department, r.Employee, r.Salary
From (Select
      d.name As Department,
      e.name As Employee,
      e.salary,
      DENSE_Rank() OVER (PARTITION BY d.name ORDER BY e.salary DESC) AS rnk
    From Employee e
      Join Department d On e.departmentId = d.id
  ) As r
Where rnk <= 3;
```