LeetCode SQL 50

Basic Joins

Created by: Bhawna Gupta

1378. Replace Employee ID With The Unique Identifier

Input:

Employees table:

id	name
1	Alice
7	Bob
11	Meir
90	Winston
3	Jonathan

EmployeeUNI table:

id	unique_id
3	1
11	2
90	3

Output:

unique_id	name
null	Alice
null	Bob
2	Meir
3	Winston
1	Jonathan

```sql
Select
  unique_id,
  name
From
  Employees
  Left Join EmployeeUNI
  On Employees.id = EmployeeUNI.id;
```

Created by: Bhawna Gupta

1068. Product Sales Analysis I

Input:

Sales table:

sale_id	product_id	year	quantity	price
1	100	2008	10	5000
2	100	2009	12	5000
7	200	2011	15	9000

Product table:

product_id	product_name
100	Nokia
200	Apple
300	Samsung

Output:

product_name	year	price
Nokia	2008	5000
Nokia	2009	5000
Apple	2011	9000

```
Select
  product_name,
  year,
  price
From
  Sales
  Join Product
  On Sales.product_id = Product.product_id;
```

Created by: Bhawna Gupta

1581. Customer Who Visited but Did Not Make Any Transactions

Input:

Visits

visit_id	customer_id
1	23
2	9
4	30
5	54
6	96
7	54
8	54

Transactions

transaction_id	visit_id	amount
2	5	310
3	5	300
9	5	200
12	1	910
13	2	970

Output:

customer_id	count_no_trans
54	2
30	1
96	1

```sql
Select
  customer_id,
  Count(customer_id) As count_no_trans
From
  Visits
  Left Join Transactions
  On Visits.visit_id = Transactions.visit_id
Where
```

Created by: Bhawna Gupta

197. Rising Temperature

Input:

Weather table:

id	recordDate	temperature
1	2015-01-01	10
2	2015-01-02	25
3	2015-01-03	20
4	2015-01-04	30

Output:

id
2
4

```
Select
  W1.id
From
  Weather W1,
  Weather W2
Where
  Datediff(W1.recordDate, W2.recordDate)=1
  AND W1.temperature > W2.temperature;
```

Created by: Bhawna Gupta

1661. Average Time of Process per Machine

Input:

Activity table:

machine_id	process_id	activity_type	timestamp
0	0	start	0.712
0	0	end	1.520
0	1	start	3.140
0	1	end	4.120
1	0	start	0.550
1	0	end	1.550
1	1	start	0.430
1	1	end	1.420
2	0	start	4.100
2	0	end	4.512
2	1	start	2.500
2	1	end	5.000

Output:

machine_id	processing_time
0	0.894
1	0.995
2	1.456

```sql
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
Where
  A1.activity_type = 'start'
  And A2.activity_type = 'end'
Group By A1.machine_id;
```

Created by: Bhawna Gupta

577. Employee Bonus

Input:

Employee table:

empId	name	supervisor	salary
3	Brad	null	4000
1	John	3	1000
2	Dan	3	2000
4	Thomas	3	4000

Bonus table:

empId	bonus
2	500
4	2000

Output:

name	bonus
Brad	null
John	null
Dan	500

```sql
Select
  name,
  bonus
From
  Employee e
  Left Join Bonus b On e.empId = b.empId
Where
  b.bonus < 1000
  Or b.bonus Is Null;
```

Created by: Bhawna Gupta

1280. Students and Examinations

Input:

Students table:

student_id	student_name
1	Alice
2	Bob
13	John
6	Alex

Subjects table:

subject_name
Math
Physics
Programming

Examinations table:

student_id	subject_name
1	Math
1	Physics
1	Programming
2	Programming
1	Physics
1	Math
13	Math
13	Programming
13	Physics
2	Math
1	Math

```
Select i.student_id, i.student_name, s.subject_name,
Count(e.subject_name) As attended_exams
From Students i Cross Join Subjects s
Left Join Examinations e
On i.student_id=e.student_id
And s.subject_name=e.subject_name
Group by i.student_id, s.subject_name
Order by i.student_id, s.subject_name;
```

Output:

student_id	student_name	subject_name	attended_exams
1	Alice	Math	3
1	Alice	Physics	2
1	Alice	Programming	1
2	Bob	Math	1
2	Bob	Physics	0
2	Bob	Programming	1
6	Alex	Math	0
6	Alex	Physics	0
6	Alex	Programming	0
13	John	Math	1
13	John	Physics	1
13	John	Programming	1

Created by: Bhawna Gupta

570. Managers with at Least 5 Direct Reports

Input:

Employee table:

id	name	department	managerId
101	John	A	null
102	Dan	A	101
103	James	A	101
104	Amy	A	101
105	Anne	A	101
106	Ron	B	101

Output:

name
John

```sql
Select
  m.name As name
from
  Employee e
  Left Join Employee m On e.managerId = m.id
group by
  m.id
having
  count(m.id)>= 5;
```

Created by: Bhawna Gupta

1934. Confirmation Rate

Input:

Signups table:

user_id	time_stamp
3	2020-03-21 10:16:13
7	2020-01-04 13:57:59
2	2020-07-29 23:09:44
6	2020-12-09 10:39:37

Confirmations table:

user_id	time_stamp	action
3	2021-01-06 03:30:46	timeout
3	2021-07-14 14:00:00	timeout
7	2021-06-12 11:57:29	confirmed
7	2021-06-13 12:58:28	confirmed
7	2021-06-14 13:59:27	confirmed
2	2021-01-22 00:00:00	confirmed
2	2021-02-28 23:59:59	timeout

Output:

user_id	confirmation_rate
6	0.00
3	0.00
7	1.00
2	0.50

```sql
Select
  s.user_id,
  Round(Avg(if(c.action = 'confirmed', 1, 0)), 2)
  As confirmation_rate
From
  Signups s
  Left Join Confirmations c On s.user_id = c.user_id
Group by
  s.user_id;
```

Created by: Bhawna Gupta