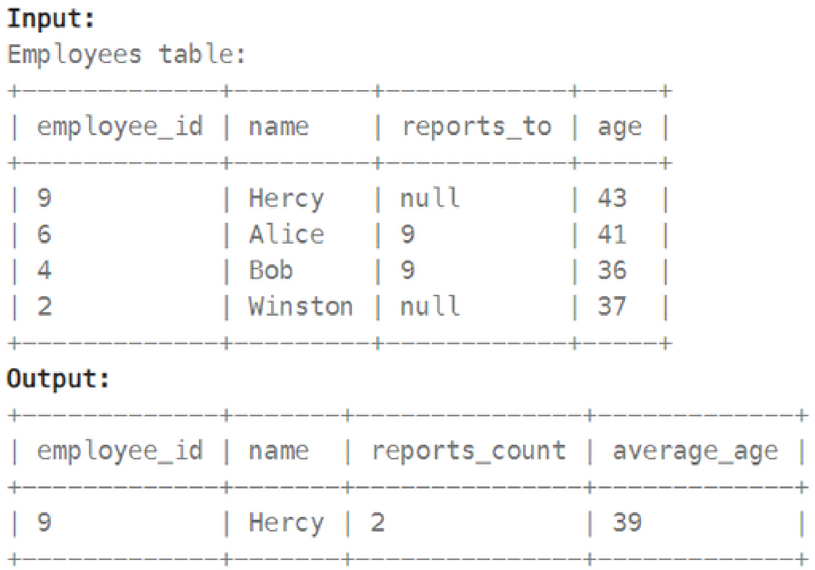
LeetCode SQL 50

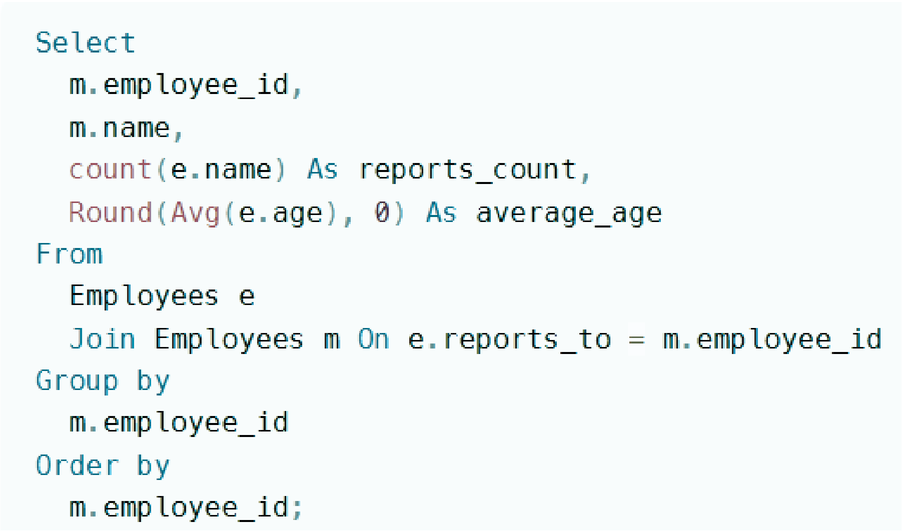
Advanced Select and Joins

Created by: Bhawna Gupta

1731. The Number of Employees Which Report to Each Employee



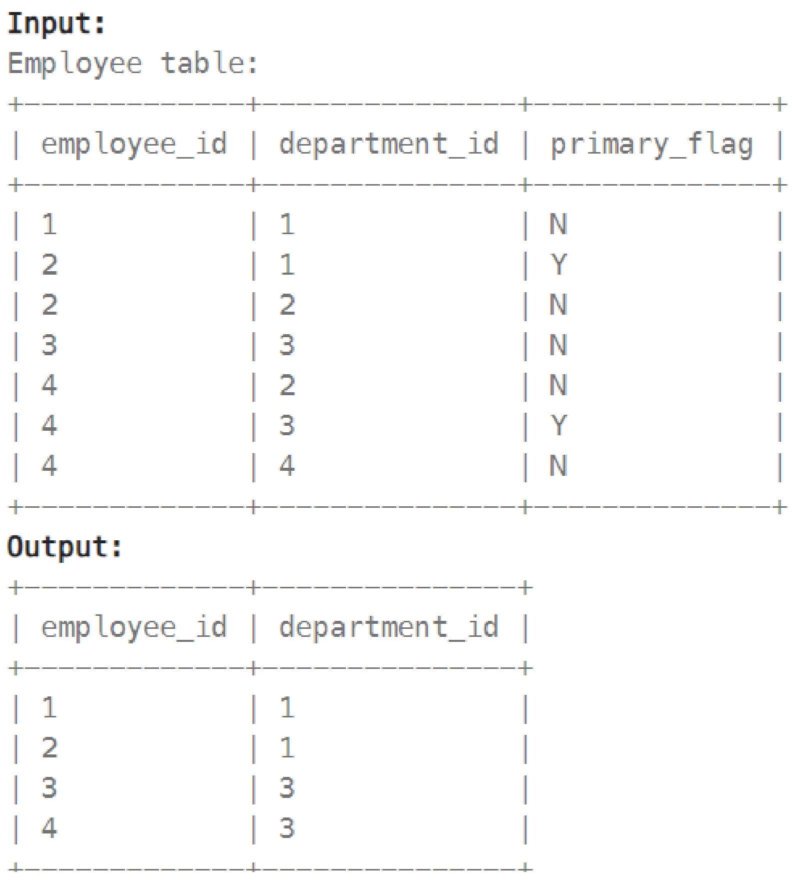
```
Input:
Employees table:
+-------------+---------+-------------+-----+
| employee_id | name    | reports_to  | age |
+-------------+---------+-------------+-----+
| 9           | Hercy   | null        | 43  |
| 6           | Alice   | 9           | 41  |
| 4           | Bob     | 9           | 36  |
| 2           | Winston | null        | 37  |
+-------------+---------+-------------+-----+
Output:
+-------------+-------+---------------+-------------+
| employee_id | name  | reports_count | average_age |
+-------------+-------+---------------+-------------+
| 9           | Hercy | 2             | 39          |
+-------------+-------+---------------+-------------+
```



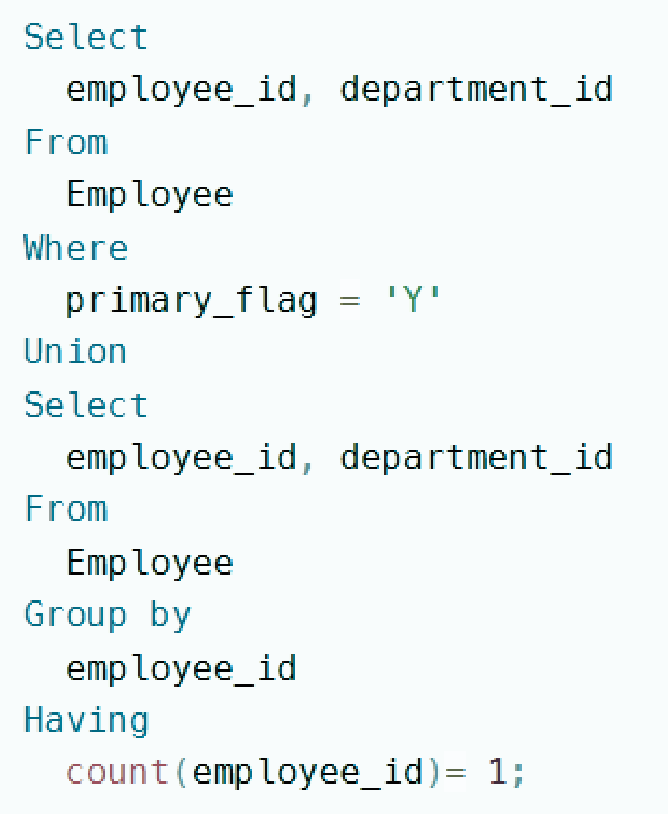
```sql
Select
  m.employee_id,
  m.name,
  count(e.name) As reports_count,
  Round(Avg(e.age), 0) As average_age
From
  Employees e
  Join Employees m On e.reports_to = m.employee_id
Group by
  m.employee_id
Order by
  m.employee_id;
```

Created by: Bhawna Gupta

1789. Primary Department for Each Employee



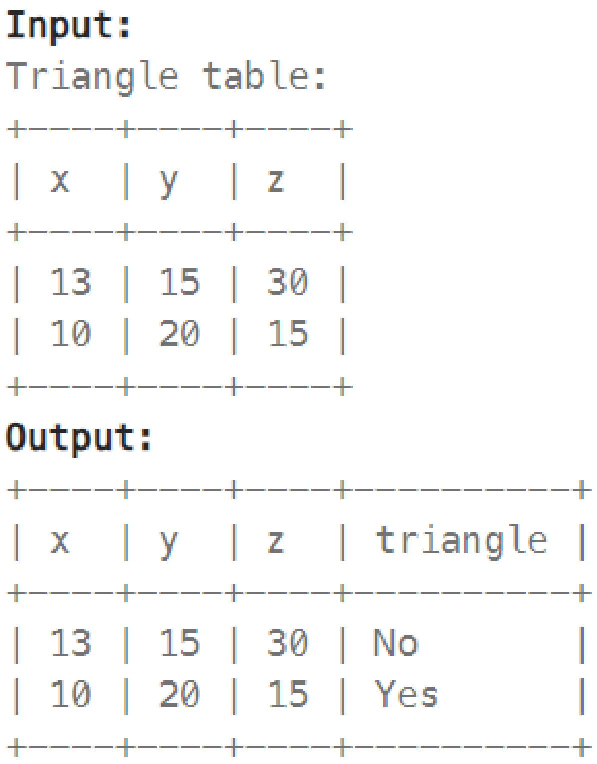
```
Input:
Employee table:
+-------------+---------------+--------------+
| employee_id | department_id | primary_flag |
+-------------+---------------+--------------+
| 1           | 1             | N            |
| 2           | 1             | Y            |
| 2           | 2             | N            |
| 3           | 3             | N            |
| 4           | 2             | N            |
| 4           | 3             | Y            |
| 4           | 4             | N            |
+-------------+---------------+--------------+
Output:
+-------------+---------------+
| employee_id | department_id |
+-------------+---------------+
| 1           | 1             |
| 2           | 1             |
| 3           | 3             |
| 4           | 3             |
+-------------+---------------+
```



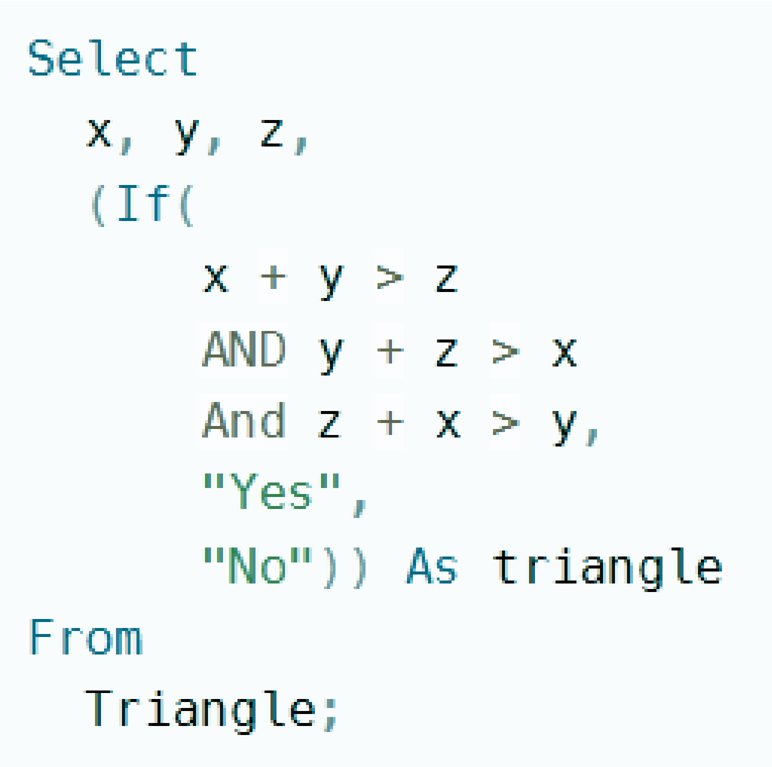
```
Select
  employee_id, department_id
From
  Employee
Where
  primary_flag = 'Y'
Union
Select
  employee_id, department_id
From
  Employee
Group by
  employee_id
Having
  count(employee_id)= 1;
```

Created by: Bhawna Gupta

610. Triangle Judgement



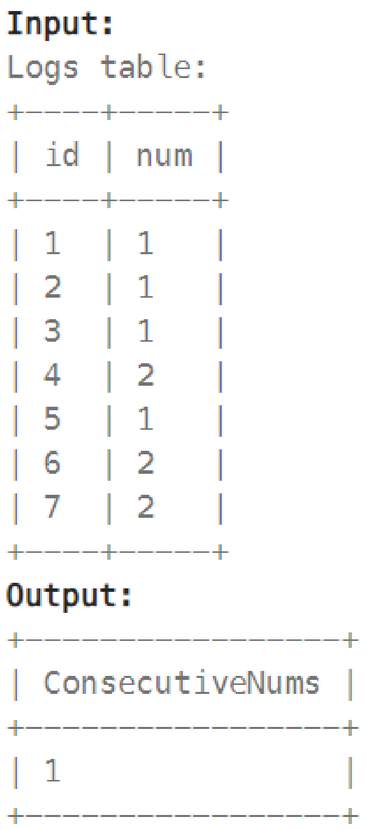
```
Input:
Triangle table:
+----+----+----+
| x  | y  | z  |
+----+----+----+
| 13 | 15 | 30 |
| 10 | 20 | 15 |
+----+----+----+
Output:
+----+----+----+----------+
| x  | y  | z  | triangle |
+----+----+----+----------+
| 13 | 15 | 30 | No       |
| 10 | 20 | 15 | Yes      |
+----+----+----+----------+
```



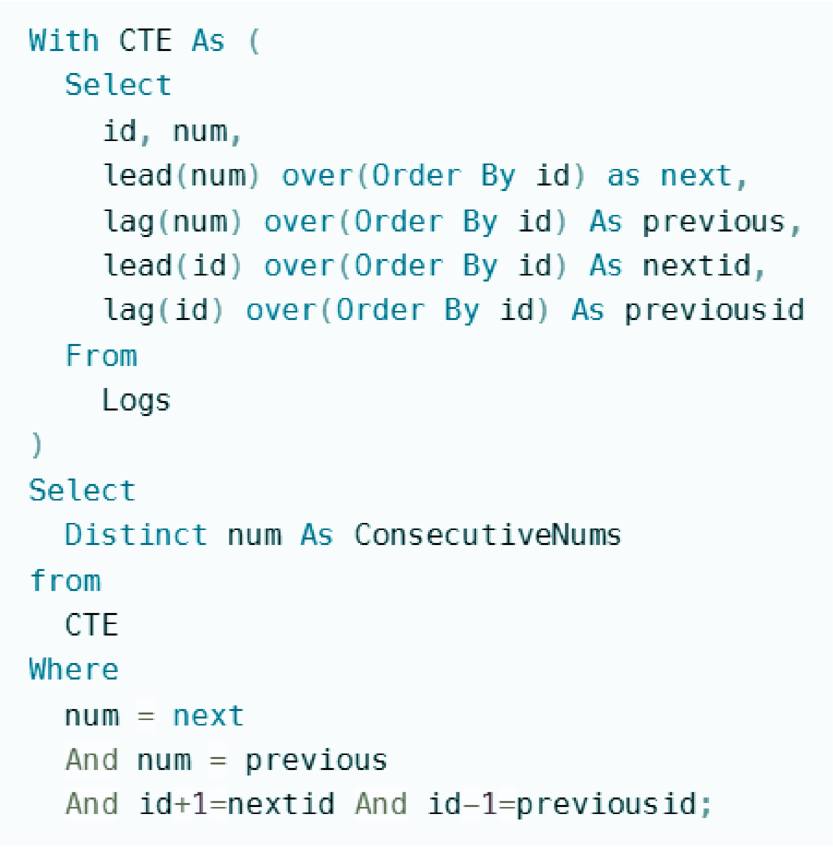
```sql
Select
  x, y, z,
  (If(
      x + y > z
      AND y + z > x
      And z + x > y,
      "Yes",
      "No")) As triangle
From
  Triangle;
```

Created by: Bhawna Gupta

180. Consecutive Numbers



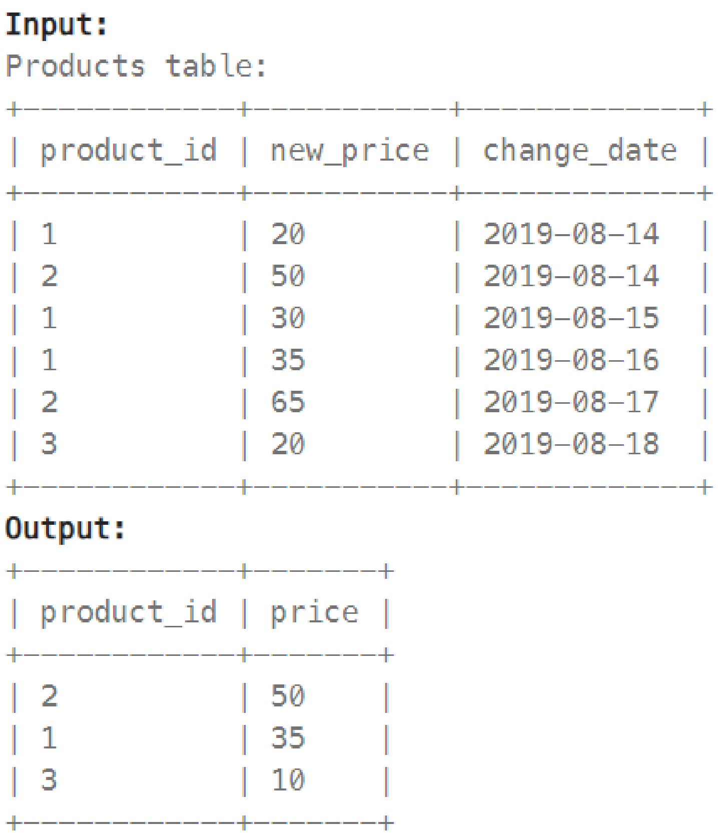
```
Input:
Logs table:
+----+-----+
| id | num |
+----+-----+
| 1  | 1   |
| 2  | 1   |
| 3  | 1   |
| 4  | 2   |
| 5  | 1   |
| 6  | 2   |
| 7  | 2   |
+----+-----+
Output:
+-----------------+
| ConsecutiveNums |
+-----------------+
| 1               |
+-----------------+
```



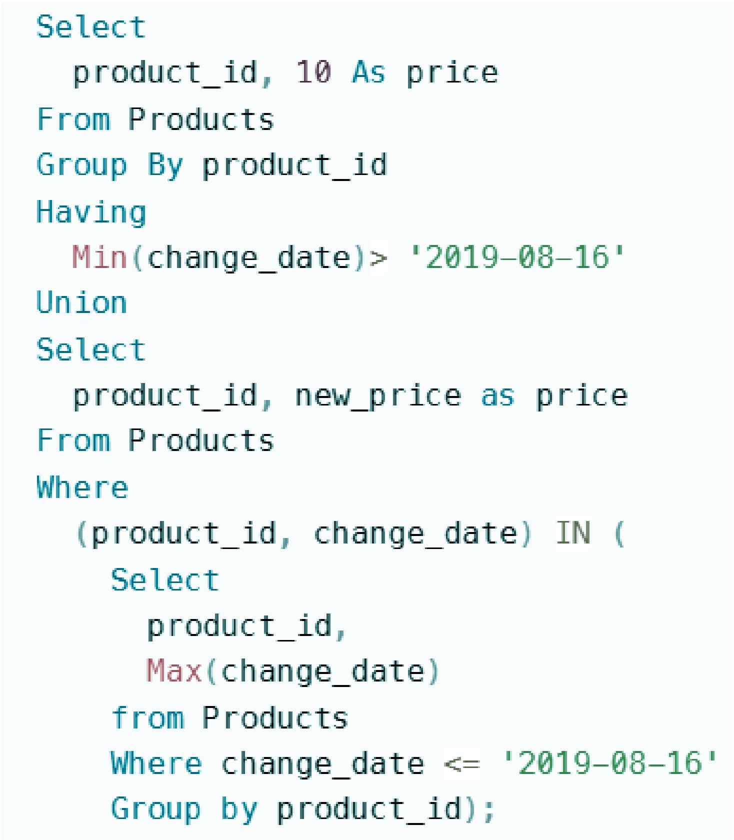
```sql
With CTE As (
  Select
    id, num,
    lead(num) over(Order By id) as next,
    lag(num) over(Order By id) As previous,
    lead(id) over(Order By id) As nextid,
    lag(id) over(Order By id) As previousid
  From
    Logs
)
Select
  Distinct num As ConsecutiveNums
from
  CTE
Where
  num = next
  And num = previous
  And id+1=nextid And id-1=previousid;
```

Created by: Bhawna Gupta

1164. Product Price at a Given Date



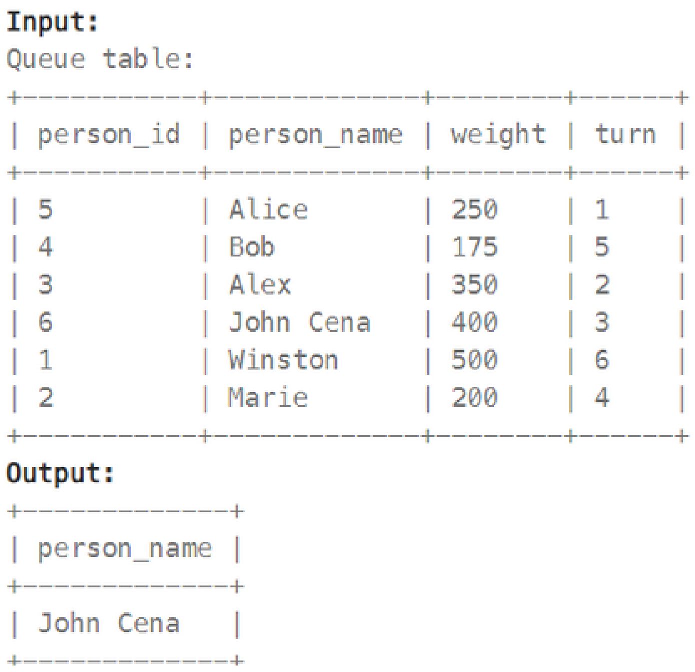
```
Input:
Products table:
+------------+-----------+-------------+
| product_id | new_price | change_date |
+------------+-----------+-------------+
| 1          | 20        | 2019-08-14  |
| 2          | 50        | 2019-08-14  |
| 1          | 30        | 2019-08-15  |
| 1          | 35        | 2019-08-16  |
| 2          | 65        | 2019-08-17  |
| 3          | 20        | 2019-08-18  |
+------------+-----------+-------------+
Output:
+------------+-------+
| product_id | price |
+------------+-------+
| 2          | 50    |
| 1          | 35    |
| 3          | 10    |
+------------+-------+
```



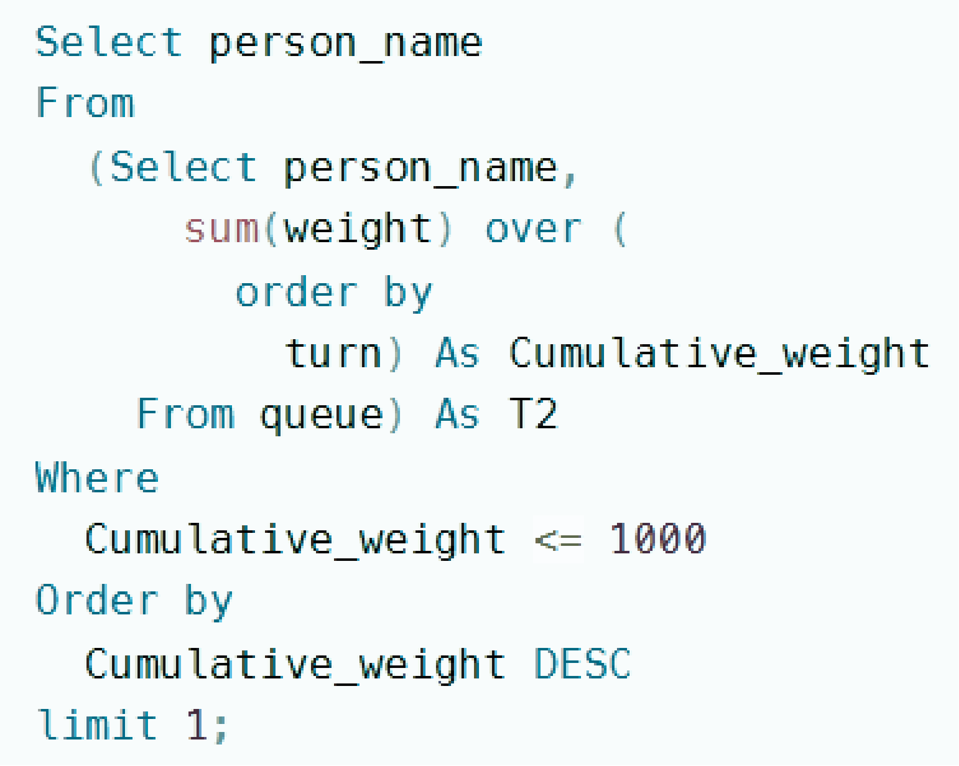
```
Select
  product_id, 10 As price
From Products
Group By product_id
Having
  Min(change_date)> '2019-08-16'
Union
Select
  product_id, new_price as price
From Products
Where
  (product_id, change_date) IN (
    Select
      product_id,
      Max(change_date)
    from Products
    Where change_date <= '2019-08-16'
    Group by product_id);
```

Created by: Bhawna Gupta

1204. Last Person to Fit in the Bus



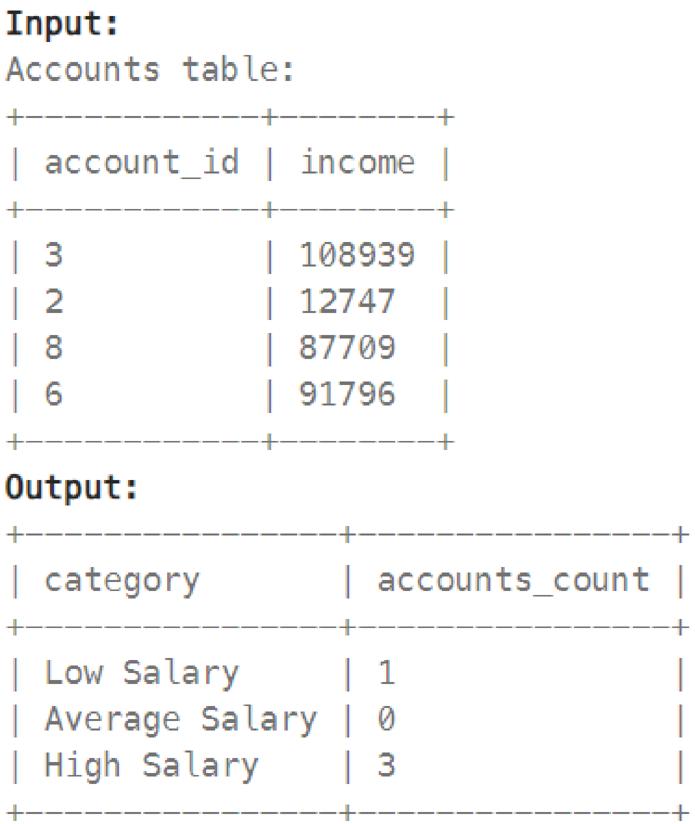
```
Input:
Queue table:
+-----------+-------------+--------+------+
| person_id | person_name | weight | turn |
+-----------+-------------+--------+------+
| 5         | Alice       | 250    | 1    |
| 4         | Bob         | 175    | 5    |
| 3         | Alex        | 350    | 2    |
| 6         | John Cena   | 400    | 3    |
| 1         | Winston     | 500    | 6    |
| 2         | Marie       | 200    | 4    |
+-----------+-------------+--------+------+
Output:
+-------------+
| person_name |
+-------------+
| John Cena   |
+-------------+
```



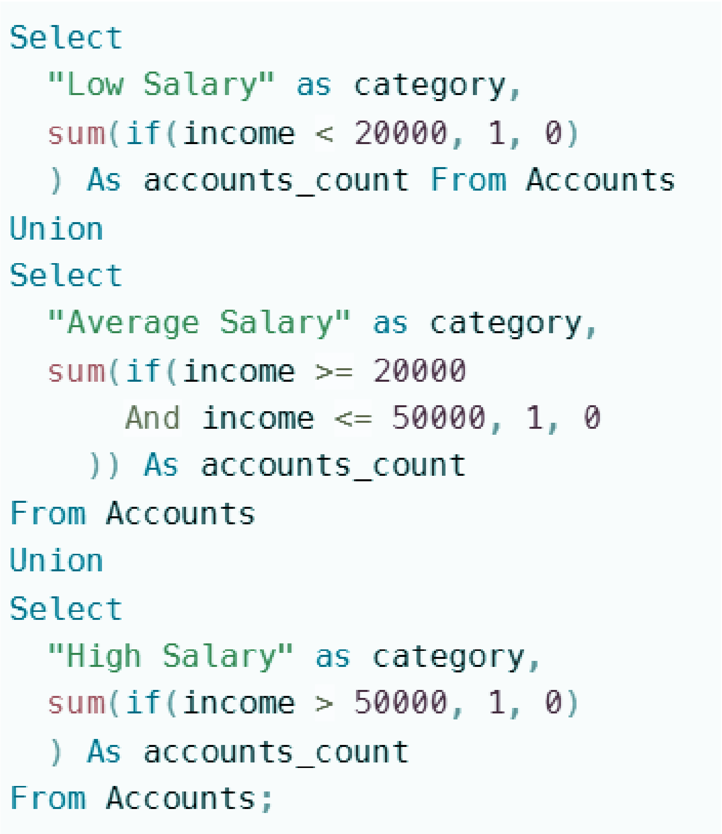
```sql
Select person_name
From
  (Select person_name,
      sum(weight) over (
        order by
          turn) As Cumulative_weight
    From queue) As T2
Where
  Cumulative_weight <= 1000
Order by
  Cumulative_weight DESC
limit 1;
```

Created by: Bhawna Gupta

1907. Count Salary Categories



```
Input:
Accounts table:
+------------+--------+
| account_id | income |
+------------+--------+
| 3          | 108939 |
| 2          | 12747  |
| 8          | 87709  |
| 6          | 91796  |
+------------+--------+
Output:
+----------------+----------------+
| category       | accounts_count |
+----------------+----------------+
| Low Salary     | 1              |
| Average Salary | 0              |
| High Salary    | 3              |
+----------------+----------------+
```



```
Select
  "Low Salary" as category,
  sum(if(income < 20000, 1, 0)
  ) As accounts_count From Accounts
Union
Select
  "Average Salary" as category,
  sum(if(income >= 20000
      And income <= 50000, 1, 0
    )) As accounts_count
From Accounts
Union
Select
  "High Salary" as category,
  sum(if(income > 50000, 1, 0)
  ) As accounts_count
From Accounts;
```

Created by: Bhawna Gupta