**175. Combine Two Tables**

SELECT Person.firstName, Person.lastName, Address.city, Address.state from Person

LEFT JOIN Address ON Person.personID=Address.personId;

**176. Second Highest Salary**

# Write your MySQL query statement below

SELECT

CASE

WHEN (SELECT COUNT(\*) FROM Employee) = 1 THEN NULL

WHEN (SELECT COUNT(DISTINCT salary) FROM Employee) = 1 THEN NULL

ELSE sub.salary

END AS SecondHighestSalary from

(SELECT DISTINCT salary from Employee ORDER BY Salary DESC LIMIT 2) sub

ORDER BY Salary ASC LIMIT 1

;

**177. Nth Highest Salary**

select distinct Salary from Employee order by Salary desc limit N, 1

**178. RANK SCORES**

**# Write your MySQL query statement below**

SELECT score, DENSE\_RANK() OVER (ORDER BY score DESC) as `rank`

from Scores;

**180. Consecutive Numbers**

WITH sub as

(SELECT num as num1, LEAD(num,1) over() num2, LEAD(num,2) over() num3 from logs)

SELECT distinct num1 as ConsecutiveNums from sub where num1=num2 AND num1=num3;

**181. Employees Earning More Than Their Managers**

# Write your MySQL query statement below

SELECT e1.name as Employee from Employee e1

JOIN Employee e2 on e1.managerId=e2.id

WHERE e1.salary>e2.salary;

**182. Duplicate Emails**

SELECT email from Person

GROUP BY Person.email

HAVING COUNT(PERSON.email)>1;

**183. Customers Who Never Order**

SELECT name as Customers

from Customers

where id not in (

select customerId

from Orders

);

**262.Trips and users:**

SELECT request\_at AS Day, ROUND(COUNT(CASE WHEN status LIKE 'cancelled%' THEN 1 ELSE NULL END)/COUNT(status),2) "Cancellation Rate"

FROM Trips

JOIN users u1 ON Trips.client\_id = u1.Users\_id AND u1.banned = 'No'

JOIN users u2 ON Trips.driver\_id = u2.Users\_id AND u2.banned = 'No'

where request\_at BETWEEN '2013-10-01' AND '2013-10-03'

GROUP BY request\_at

**511.GAME PLAY ANALYSIS 1:**

**# Write your MySQL query statement below**

SELECT DISTINCT player\_id,min(event\_date) as first\_login FROM Activity

GROUP BY player\_id;

**570. Managers with 5 reports:**

select name from Employee where id in (select managerId from Employee group by managerId having count(\*)>=5);

**EXCHANGE SEATS:**

SELECT id,

CASE WHEN id%2!=0 AND id<(SELECT count(\*) from seat) THEN LEAD(student,1) OVER(order by id)

WHEN id=(SELECT count(\*) from seat) and id%2!=0 THEN student

ELSE LAG(student,1) OVER(order by id)

END as student

FROM Seat s;

**627. Swap Salary**

# Write your MySQL query statement below

update Salary

SET

sex=CASE

WHEN sex='f' THEN 'm'

ELSE 'f'

END;

**1050. Actors and Directors Who Cooperated At Least Three Times:**

# Write your MySQL query statement below

select actor\_id, director\_id from ActorDirector group by actor\_id, director\_id having count(timestamp)>=3;

**1070. Product Sales Analysis III:**

SELECT product\_id, year AS first\_year, quantity, price

FROM Sales

WHERE (product\_id, year) IN (

SELECT product\_id, MIN(year) as year

FROM Sales

GROUP BY product\_id) ;

**1251. Average Selling Price:**

**/\* Write your PL/SQL query statement below \*/**

SELECT

p.product\_id,

ROUND(SUM(u.units \* p.price)/SUM(u.units), 2) AS average\_price

FROM

Prices p

INNER JOIN UnitsSold u ON p.product\_id = u.product\_id

WHERE

u.purchase\_date >= p.start\_date AND u.purchase\_date <= p.end\_date

GROUP BY

p.product\_id;

**1174. Immediate Food Delivery II**

with cte as(

select \*,RANK() over( PARTITION BY customer\_id order by order\_date) R from delivery

)

SELECT

ROUND(COUNT(CASE WHEN order\_date = customer\_pref\_delivery\_date THEN delivery\_id END)/

(SELECT COUNT(customer\_id)),4)\*100 immediate\_percentage

FROM cte WHERE R=1;