Name: Gaurav Shivaji Tanpure

PRN: 123B2B329

College: PCCOE

Domain: SQL

WEEK 6 TASK

Leetcode Account Link: https://leetcode.com/u/gaurav_tanpure_07/

Problem Statement 1:

https://leetcode.com/problems/nth-highest-salary/

```
CREATE FUNCTION getNthHighestSalary(@N INT) RETURNS INT AS
BEGIN
RETURN (
select distinct salary
from
(
select salary, dense_rank() over (order by Salary desc) as Rank
from Employee
) p
where p.Rank = @N
);
END
```

Problem Statement 2:

https://leetcode.com/problems/rank-scores/

```
SELECT score
    ,DENSE_RANK() OVER (
        ORDER BY score DESC
        ) AS [rank]
FROM Scores;
```

Problem Statement 3:

https://leetcode.com/problems/consecutive-numbers/

```
declare @times int = 3;

select distinct num ConsecutiveNums
from (
    select id, num,
        id - row_number()over(partition by num order by num ) + 1 as rw
    from Logs
) c
group by num, rw
having count(rw) >= @times
```

Problem Statement 4:

https://leetcode.com/problems/department-top-three-salaries/

```
SELECT Department, Employee, Salary
FROM(
    SELECT D.name AS Department, E.name AS Employee, E.salary AS Salary,
    DENSE_RANK() OVER (PARTITION BY departmentId ORDER BY salary DESC) AS
d_rank
    FROM Employee E INNER JOIN Department D ON E.departmentId = D.id
) T
WHERE d_rank <= 3;
```

Problem Statement 5:

https://leetcode.com/problems/trips-and-users/

```
SELECT
    request_at AS Day,
    ROUND(
        CAST(SUM(CASE WHEN STATUS != 'completed' THEN 1 ELSE 0 END) AS FLOAT)
/ COUNT(*),
        2
    ) AS [Cancellation Rate]
FROM
    trips t
JOIN
    users u1
    ON t.client_id = u1.users_id
```

```
    AND u1.banned = 'No'
JOIN
    users u2
    ON t.driver_id = u2.users_id
    AND u2.banned = 'No'
WHERE
    request_at BETWEEN '2013-10-01' AND '2013-10-03'
GROUP BY
    request_at;
```

Problem Statement 6:

https://leetcode.com/problems/game-play-analysis-iv/

```
declare @max int = (select count(distinct player_id) from Activity)

;with firstLogin as(
select
    a1.player_id [player_id],
    min(a1.event_date) [event_date]
from
    Activity a1
group by
    a1.player_id
),

atLeastTwice as(
select
    count(distinct a2.player_id) [p]
from
    Activity a2
    inner join firstLogin f on f.player_id = a2.player_id
        and a2.event_date = dateadd(day, 1, f.event_date)
)


select
    round(atl.p * 1.0 / @max,2) [fraction]
from atLeastTwice atl
```

Problem Statement 7:

https://leetcode.com/problems/managers-with-at-least-5-direct-reports/

```
SELECT name
FROM Employee e
INNER JOIN (
    SELECT managerId
    FROM Employee
    GROUP BY managerId
    HAVING COUNT(*) >=5 ) t
ON e.id=t.managerId
```

Problem Statement 8:

https://leetcode.com/problems/employee-bonus/

```
select
    emp.name,
    b.bonus
from
    Employee emp
    left join Bonus b on b.empId = emp.empId
where isnull(b.bonus, 0) < 1000
```

Problem Statement 9:

https://leetcode.com/problems/human-traffic-of-stadium/

```
with cte as
(
SELECT id, visit_date, people,
row_number() over(order by id) as rn
FROM stadium
where people >= 100
), cte2 as
(
SELECT id, visit_date, people, id-rn as diff
from cte
)
select id, visit_date, people
from cte2
WHERE diff in (select diff from cte2
```

```
group by diff
having count(diff) >= 3)
```

Problem Statement 10:

https://leetcode.com/problems/tree-node/

```
select distinct T1.id, (
    case
    when T1.p_id is null then 'Root'
    when T2.id is not null then 'Inner'
    else 'Leaf'
    end
) as type
from Tree T1
left join Tree T2
on T1.id = T2.p_id
```