MySQL Questions

Combine Two Tables

Write an SQL query to report the first name, last name, city, and state of each person in the Person table. If the address of a personId is not present in the Address table, report null instead.

```
Select a.firstName, a.lastName,b.city,b.state from Person a LEFT JOIN Address b ON
a.personID=b.personID;
```

Second Highest Salary

Write an SQL query to report the second highest salary from the Employee table. If there is no second highest salary, the query should report null

IFNULL (Expression1, Expression2) => It returns expression1 when the expression1 is not null. Otherwise, it will return expression2.

The LIMIT clause is used to specify the number of records to return

The OFFSET argument is used to identify the starting point to return rows from a result set. Basically, it exclude the first set of records. Only used with ORDER BY.

```
Select IFNULL((SELECT distinct salary
from Employee order by Salary desc LIMIT 1 offset 1 )
,null) as SecondHighestSalary;
```

Nth Highest Salary

Write an SQL query to report the nth highest salary from the Employee table. If there is no nth highest salary, the query should report null.

```
CREATE FUNCTION getNthHighestSalary(N INT) RETURNS INT
BEGIN
DECLARE M INT;
SET M=N-1;
  RETURN (
    # Write your MySQL query statement below.
    SELECT DISTINCT Salary FROM Employee ORDER BY Salary DESC LIMIT 1 OFFSET M
  );
END
```

```
CREATE FUNCTION getNthHighestSalary(N INT) RETURNS INT
BEGIN
  RETURN (
    # Write your MySQL query statement below.
    SELECT DISTINCT Salary
    FROM Employee e1
    WHERE N-1 = (SELECT COUNT(DISTINCT Salary) FROM Employee e2 WHERE e1.Salary <
e2.Salary)
  );
END
```

Rank Scores

Write an SQL query to rank the scores. The ranking should be calculated according to the following rules:

The scores should be ranked from the highest to the lowest. If there is a tie between two scores, both should have the same ranking. After a tie, the next ranking number should be the next consecutive integer value. In other words, there should be no holes between ranks. Return the result table ordered by score in descending order.

The GROUP BY statement groups rows that have the same values into summary rows, like "find the number of customers in each country".

```
# Write your MySQL query statement below
Select S.score,COUNT(S2.score) as "rank"
FROM Scores S,(Select distinct score from Scores) S2
WHERE S.score<=S2.score GROUP BY S.id ORDER BY S.score DESC;
```

```
select Score, (select count(distinct Score) from Scores where Score >= s.Score)
as 'Rank'
from Scores as s
order by Score desc;
```

Consecutive Numbers

Write an SQL query to find all numbers that appear at least three times consecutively.

Return the result table in any order.

The query result format is in the following example.

```
SELECT DISTINCT l1.num AS 'ConsecutiveNums'
FROM Logs l1, Logs l2, Logs l3
```

```
WHERE l1.num = l2.num AND l2.num = l3.num AND l1.id = l2.id + 1 AND l2.id = l3.id
+ 1
```
