DESCRIPTION

Table: Logs

+-------------+---------+

| Column Name | Type |

+-------------+---------+

| id | int |

| num | varchar |

+-------------+---------+

In SQL, id is the primary key for this table.

id is an autoincrement column starting from 1.

Find all numbers that appear at least three times consecutively.

Return the result table in **any order**.

The result format is in the following example.

Example 1:

Input:

Logs table:

+----+-----+

| id | num |

+----+-----+

| 1 | 1 |

| 2 | 1 |

| 3 | 1 |

| 4 | 2 |

| 5 | 1 |

| 6 | 2 |

| 7 | 2 |

+----+-----+

Output:

```
+-----------------+
| ConsecutiveNums |
+-----------------+
| 1           |
+-----------------+
```

Explanation: 1 is the only number that appears consecutively for at least three times.

SOLUTION

MySQL:

- Self-join Logs three times as L1, L2, and L3
- Select l1.num using DISTINCT to remove duplicates
- Add the following conditions in the WHERE clause:
- L1.id = l2.id -1 (i.e. 1 is 2 – 1), and L2.id = L3.id – 1 (i.e. 2 is 3-1)
- L1, L2 and L3 have the same num

```
SELECT DISTINCT l1.num ConsecutiveNums
FROM Logs l1, Logs l2, Logs l3
WHERE L1.id = l2.id - 1 AND l2.id = l3.id -1 AND l1.num = L2.num AND L2.num = L3.num;
```

PostgreSQL:

- Similar approach as using JOIN three times, and id = id – 1 as keys to get consecutive rows
- The WHERE clause enforce conditions such as L1, L2 and L3 have the same num

```
SELECT DISTINCT l1.num ConsecutiveNums
FROM Logs l1
JOIN logs l2
ON l1.id = l2.id - 1
JOIN logs l3
ON l2.id = l3.id - 1
WHERE l1.num = l2.num AND l2.num = l3.num
```