3262. Find Overlapping Shifts Premium

Medium Topics

SQL Schema > Pandas Schema >

Table: `EmployeeShifts`

```
+------------------+---------+
| Column Name      | Type    |
+------------------+---------+
| employee_id      | int     |
| start_time       | time    |
| end_time         | time    |
+------------------+---------+
(employee_id, start_time) is the unique key for this table.
This table contains information about the shifts worked by employees, including the start and end times on a specific date.
```

Write a solution to count the number of **overlapping shifts** for each employee. Two shifts are considered overlapping if one shift's `end_time` is **later than** another shift's `start_time`.

Return the result table ordered by `employee_id` *in* ***ascending*** *order.*

The query result format is in the following example.

Example:

Input:

`EmployeeShifts` table:

```
+-------------+------------+----------+
| employee_id | start_time | end_time |
+-------------+------------+----------+
| 1           | 08:00:00   | 12:00:00 |
| 1           | 11:00:00   | 15:00:00 |
| 1           | 14:00:00   | 18:00:00 |
| 2           | 09:00:00   | 17:00:00 |
| 2           | 16:00:00   | 20:00:00 |
| 3           | 10:00:00   | 12:00:00 |
| 3           | 13:00:00   | 15:00:00 |
| 3           | 16:00:00   | 18:00:00 |
| 4           | 08:00:00   | 10:00:00 |
| 4           | 09:00:00   | 11:00:00 |
+-------------+------------+----------+
```

Output:

```
+-------------+--------------------+
| employee_id | overlapping_shifts |
+-------------+--------------------+
| 1           | 2                  |
| 2           | 1                  |
| 4           | 1                  |
+-------------+--------------------+
```

Explanation:

- Employee 1 has 3 shifts:
  - 08:00:00 to 12:00:00
  - 11:00:00 to 15:00:00
  - 14:00:00 to 18:00:00

  The first shift overlaps with the second, and the second overlaps with the third, resulting in 2 overlapping shifts.
- Employee 2 has 2 shifts:
  - 09:00:00 to 17:00:00
  - 16:00:00 to 20:00:00

  These shifts overlap with each other, resulting in 1 overlapping shift.
- Employee 3 has 3 shifts:
  - 10:00:00 to 12:00:00
  - 13:00:00 to 15:00:00
  - 16:00:00 to 18:00:00

  None of these shifts overlap, so Employee 3 is not included in the output.
- Employee 4 has 2 shifts:
  - 08:00:00 to 10:00:00
  - 09:00:00 to 11:00:00

  These shifts overlap with each other, resulting in 1 overlapping shift.

The output shows the employee_id and the count of overlapping shifts for each employee who has at least one overlapping shift, ordered by employee_id in ascending order.

Seen this question in a real interview before? 1/5

Yes No

Accepted 627 | Submissions 1K | Acceptance Rate 61.1%

Topics

Database

Discussion (6)