Table: Employees

```
+-------------+------+
| Column Name | Type |
+-------------+------+
| emp_id      | int  |
| event_day   | date |
| in_time     | int  |
| out_time    | int  |
+-------------+------+
(emp_id, event_day, in_time) is the primary key of this table.
The table shows the employees&#39; entries and exits in an office.
event_day is the day at which this event happened, in_time is the minute at which the employ
in_time and out_time are between 1 and 1440.
It is guaranteed that no two events on the same day intersect in time, and in_time &lt; out_
```

Write an SQL query to calculate the total time **in minutes** spent by each employee on each day at the office. Note that within one day, an employee can enter and leave more than once. The time spent in the office for a single entry is `out_time - in_time`.

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

The query result format is in the following example.

Example 1:**

```
Input:
Employees table:
+--------+------------+---------+----------+
| emp_id | event_day  | in_time | out_time |
+--------+------------+---------+----------+
| 1      | 2020-11-28 | 4       | 32       |
| 1      | 2020-11-28 | 55      | 200      |
| 1      | 2020-12-03 | 1       | 42       |
| 2      | 2020-11-28 | 3       | 33       |
| 2      | 2020-12-09 | 47      | 74       |
+--------+------------+---------+----------+
Output:
+------------+--------+------------+
| day        | emp_id | total_time |
+------------+--------+------------+
| 2020-11-28 | 1      | 173        |
| 2020-11-28 | 2      | 30         |
| 2020-12-03 | 1      | 41         |
```

```
| 2020-12-09 | 2      | 27         |
+------------+--------+------------+
Explanation:
Employee 1 has three events: two on day 2020-11-28 with a total of (32 - 4) + (200 - 55) = 1
Employee 2 has two events: one on day 2020-11-28 with a total of (33 - 3) = 30, and one on d
```