Table: `Transactions`

```
+-------------+------+
| Column Name | Type |
+-------------+------+
| account_id  | int  |
| day         | date |
| type        | ENUM |
| amount      | int  |
+-------------+------+
(account_id, day) is the primary key for this table.
Each row contains information about one transaction, including the transaction type, the day
type is an ENUM of the type (&#39;Deposit&#39;,&#39;Withdraw&#39;)
```

Write an SQL query to report the balance of each user after each transaction. You may assume that the balance of each account before any transaction is `0` and that the balance will never be below `0` at any moment.

Return the result table **in ascending order** by `account_id`, then by `day` in case of a tie.

The query result format is in the following example.

Example 1:**

```
Input:
Transactions table:
+------------+------------+----------+--------+
| account_id | day        | type     | amount |
+------------+------------+----------+--------+
| 1          | 2021-11-07 | Deposit  | 2000   |
| 1          | 2021-11-09 | Withdraw | 1000   |
| 1          | 2021-11-11 | Deposit  | 3000   |
| 2          | 2021-12-07 | Deposit  | 7000   |
| 2          | 2021-12-12 | Withdraw | 7000   |
+------------+------------+----------+--------+
Output:
+------------+------------+---------+
| account_id | day        | balance |
+------------+------------+---------+
| 1          | 2021-11-07 | 2000    |
| 1          | 2021-11-09 | 1000    |
| 1          | 2021-11-11 | 4000    |
| 2          | 2021-12-07 | 7000    |
| 2          | 2021-12-12 | 0       |
+------------+------------+---------+
Explanation:
Account 1:
- Initial balance is 0.
- 2021-11-07 --&gt; deposit 2000. Balance is 0 + 2000 = 2000.
- 2021-11-09 --&gt; withdraw 1000. Balance is 2000 - 1000 = 1000.
- 2021-11-11 --&gt; deposit 3000. Balance is 1000 + 3000 = 4000.
Account 2:
- Initial balance is 0.
- 2021-12-07 --&gt; deposit 7000. Balance is 0 + 7000 = 7000.
- 2021-12-12 --&gt; withdraw 7000. Balance is 7000 - 7000 = 0.
```