Medium

Table: Delivery

```
+-----------------------------+---------+
| Column Name                 | Type    |
+-----------------------------+---------+
| delivery_id                 | int     |
| customer_id                 | int     |
| order_date                  | date    |
| customer_pref_delivery_date | date    |
+-----------------------------+---------+
```

delivery_id is the column of unique values of this table.
The table holds information about food delivery to customers that make orders at some date and specify a preferred delivery date (on the same order date or after it).

If the customer's preferred delivery date is the same as the order date, then the order is called **immediate;** otherwise, it is called **scheduled**.

The **first order** of a customer is the order with the earliest order date that the customer made. It is guaranteed that a customer has precisely one first order.

Write a solution to find the percentage of immediate orders in the first orders of all customers, **rounded to 2 decimal places**.

The result format is in the following example.

Example 1:

Input:
Delivery table:

```
+-------------+-------------+------------+-----------------------------+
| delivery_id | customer_id | order_date | customer_pref_delivery_date |
+-------------+-------------+------------+-----------------------------+
| 1           | 1           | 2019-08-01 | 2019-08-02                  |
| 2           | 2           | 2019-08-02 | 2019-08-02                  |
| 3           | 1           | 2019-08-11 | 2019-08-12                  |
| 4           | 3           | 2019-08-24 | 2019-08-24                  |
| 5           | 3           | 2019-08-21 | 2019-08-22                  |
| 6           | 2           | 2019-08-11 | 2019-08-13                  |
| 7           | 4           | 2019-08-09 | 2019-08-09                  |
+-------------+-------------+------------+-----------------------------+
```

Output:

```
+----------------------+
| immediate_percentage |
+----------------------+
| 50.00                |
+----------------------+
```

Explanation:
The customer id 1 has a first order with delivery id 1 and it is scheduled.
The customer id 2 has a first order with delivery id 2 and it is immediate.
The customer id 3 has a first order with delivery id 5 and it is scheduled.
The customer id 4 has a first order with delivery id 7 and it is immediate.
Hence, half the customers have immediate first orders.

```
# Write your MySQL query statement below
```

```
#m1
-- SELECT
--     ROUND(100.0 * SUM(IF(Status = 'immediate', 1, 0)) / COUNT(*),2) AS immediate_percentage
-- FROM (
--     SELECT customer_id,order_date,
--         CASE
--             WHEN order_date = customer_pref_delivery_date THEN 'immediate'
--             ELSE 'scheduled'
--         END AS Status
--     FROM Delivery
--     WHERE (customer_id, order_date) IN (SELECT customer_id, MIN(order_date)
--                                 FROM Delivery
--                                 GROUP BY customer_id
--     )
-- ) AS FirstOrders;


#m2
-- WITH FirstOrders AS (
--     SELECT
--         customer_id,
--         order_date,
--         CASE
--             WHEN order_date = customer_pref_delivery_date THEN 'immediate'
--             ELSE 'scheduled'
--         END AS Status
--     FROM Delivery
--     WHERE (customer_id, order_date) IN (
--         SELECT customer_id, MIN(order_date)
--         FROM Delivery
--         GROUP BY customer_id
--     )
-- )
-- SELECT
--     ROUND(100.0 * SUM(IF(Status = 'immediate', 1, 0)) / COUNT(*), 2) AS immediate_percentage
-- FROM FirstOrders;


#m3
-- SELECT
--     ROUND(100.0 * SUM(IF(d.order_date = d.customer_pref_delivery_date, 1, 0)) / COUNT(*), 2) AS
immediate_percentage
-- FROM
--     Delivery d
-- JOIN
--     (SELECT customer_id, MIN(order_date) AS first_order_date
--      FROM Delivery
--      GROUP BY customer_id) first_orders
-- ON d.customer_id = first_orders.customer_id
-- AND d.order_date = first_orders.first_order_date;


#m4
WITH RankedOrders AS (
    SELECT
        customer_id,
        order_date,
        customer_pref_delivery_date,
        ROW_NUMBER() OVER (PARTITION BY customer_id ORDER BY order_date) AS rn
    FROM Delivery
```

```
)
SELECT
    ROUND(100.0 * SUM(IF(order_date = customer_pref_delivery_date, 1, 0)) / COUNT(*), 2) AS
immediate_percentage
FROM
    RankedOrders
WHERE
    rn = 1;
```