Problem 1173: Immediate Food Delivery I

Problem Information

Difficulty: Easy
Acceptance Rate: 0.00%
Paid Only: No

Problem Description

Table:

Delivery

+------------------------------+---------+ | Column Name | Type | +------------------------------+---------+ | delivery_id | int | | customer_id | int | | order_date | date | | customer_pref_delivery_date | date | +------------------------------+---------+ delivery_id is the primary key (column with 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

.

Write a solution to find the percentage of immediate orders in the table,

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 | 5 | 2019-08-02 | 2019-08-02 | | 3 | 1 | 2019-08-11 | 2019-08-11 | | 4 | 3 | 2019-08-24 | 2019-08-26 | | 5 | 4 | 2019-08-21 | 2019-08-22 | | 6 | 2 | 2019-08-11 | 2019-08-13 | +-------------+-------------+------------+----------------------------+

Output:

+----------------------+ | immediate_percentage | +----------------------+ | 33.33 | +----------------------+

Explanation:

The orders with delivery id 2 and 3 are immediate while the others are scheduled.

Code Snippets

MySQL:

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

MS SQL Server:

```
/* Write your T-SQL query statement below */
```

PostgreSQL:

```
-- Write your PostgreSQL query statement below
```

Oracle:

```
/* Write your PL/SQL query statement below */
```

Pandas:

```
import pandas as pd

def food_delivery(delivery: pd.DataFrame) -> pd.DataFrame:
```

Solutions

MySQL Solution:

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

MS SQL Server Solution:

```
/* Write your T-SQL query statement below */
```

PostgreSQL Solution:

```
-- Write your PostgreSQL query statement below
```

Oracle Solution:

```
/* Write your PL/SQL query statement below */
```

Pandas Solution:

```
import pandas as pd

def food_delivery(delivery: pd.DataFrame) -> pd.DataFrame:
```