

```

mysql> CREATE DATABASE ecommerce_analysis;
Query OK, 1 row affected (0.10 sec)

mysql> USE ecommerce_analysis;
Database changed

mysql> CREATE TABLE orders (
->     order_id VARCHAR(20) PRIMARY KEY,
->     product_id VARCHAR(20),
->     category VARCHAR(50),
->     supplier VARCHAR(50),
->     customer_region VARCHAR(50),
->     marketing_channel VARCHAR(50),
->     order_value DECIMAL(10,2),
->     order_date DATE
-> );
Query OK, 0 rows affected (0.13 sec)

mysql>
mysql> CREATE TABLE returns (
->     return_id VARCHAR(20) PRIMARY KEY,
->     order_id VARCHAR(20),
->     return_date DATE,
->     return_reason VARCHAR(100),
->     FOREIGN KEY (order_id) REFERENCES orders(order_id)
-> );
Query OK, 0 rows affected (0.07 sec)

mysql> SELECT
->     o.order_id,
->     o.product_id,
->     o.category,
->     o.supplier,
->     o.customer_region,
->     o.marketing_channel,
->     o.order_value,
->     o.order_date,
->     r.return_id,
->     r.return_date,
->     r.return_reason,
->     CASE WHEN r.return_id IS NOT NULL THEN 1 ELSE 0 END AS is_returned
-> FROM orders o

```

app.py ×

app.py

```
1  import streamlit as st
2  import pandas as pd
3  import plotly.express as px
4
5  # --- Load datasets ---
6
7  orders = pd.read_csv("C:/Users/Veena/Downloads/orders.csv")
8
9  returns = pd.read_csv("C:/Users/Veena/Downloads/returns.csv")
10
11
12  # --- Clean column names ---
13  orders.columns = orders.columns.str.strip().str.lower()
14  returns.columns = returns.columns.str.strip().str.lower()
15
16  # --- Convert join columns to string ---
17  orders['id'] = orders['id'].astype(str)
18  returns['order_id'] = returns['order_id'].astype(str)
19
20  # --- Merge datasets and create return flag ---
21  data = pd.merge(orders, returns, left_on="id", right_on="order_id", how="left")
22  data['is_returned'] = data['return_id'].notnull().astype(int)
23
```

Num Lock ON

app.py

app.py

```
16 # --- Convert join columns to string ---
17 orders['id'] = orders['id'].astype(str)
18 returns['order_id'] = returns['order_id'].astype(str)
19
20 # --- Merge datasets and create return flag ---
21 data = pd.merge(orders, returns, left_on="id", right_on="order_id", how="left")
22 data['is_returned'] = data['return_id'].notnull().astype(int)
23
24 # --- Convert dates to datetime ---
25 data['order_date'] = pd.to_datetime(data['order_date'], errors='coerce')
26 data['return_date'] = pd.to_datetime(data['return_date'], errors='coerce')
27
28 # --- Sidebar filters ---
29 st.sidebar.header("Filters")
30 min_date = data['order_date'].min()
31 max_date = data['order_date'].max()
32
33 date_range = st.sidebar.date_input("Select Order Date Range", [min_date, max_date])
34 ship_modes = st.sidebar.multiselect(
35     "Select Shipping Mode", options=data['ship_mode'].unique(), default=data['ship_mode'].unique()
36 )
37
38 # --- Apply filters ---
39 filtered_data = data[
    (data['order_date'] >= min_date) && (data['order_date'] <= max_date) &&
    (data['ship_mode'].isin(ship_modes)) &&
    (data['is_returned'] == 0)]
```

Num Lock ON

Filters

Select Order Date Range

2014/01/03 – 2017/12/30

Select Shipping Mode

Standard ×

Second ×

First ×

Same Day ×



Orders & Returns Dashboard

Total Orders

5009

Total Returns

0

Return Rate (%)

0.00%

Return Rate by Shipping Mode

Return Rate by Shipping Mode

Filters

Select Order Date Range

2014/01/03 – 2017/12/30

Select Shipping Mode

Standard

Second

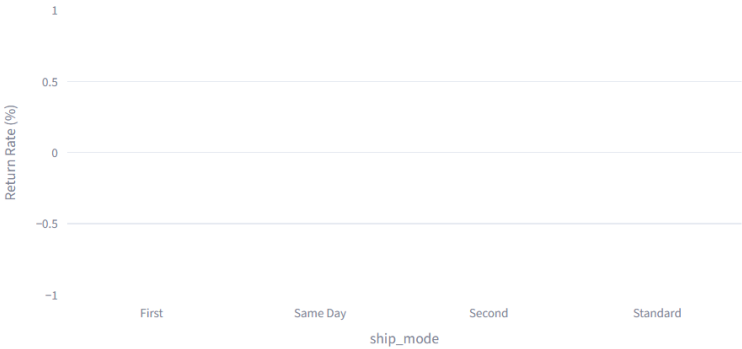
First

Same Day



Return Rate by Shipping Mode

Return Rate by Shipping Mode



Monthly Return Rate

Monthly Return Rate (%)

Num Lock ON

Deploy

Filters

Select Order Date Range

2014/01/03 – 2017/12/30

Select Shipping Mode

Standard

×

Second

×

First

×

Same Day

×



Monthly Return Rate (%)



Returns by Reason

No returns in the selected filters.

Num Lock ON