cms

July 27, 2025

1 ITP Mini Project

2 Courier Management System

For a seamless eCommerce shopping experience, it is essential to deliver the product promptly to the customer. And that's where a professional courier service plays a vital role. 'FastTrack' courier company stores the relevant data of its clients and parcels in the form of dictionary. Create a dictionary for storing shipment information in key-value pairs. Shipment id is used as a key and list of other attributes like sender, receiver, start date, Delivery Date, Sender_location, Receiver_location, Delivery status, Shipping cost is associated with shipment id. Use the data shown in the table below.

Shipment_id Sender Receiver Start_date Delivery_date Sender_location Receiver_location Delivery_status Shipping_cost

Q1. Create a Dictionary of lists to store the information of shipments given in the table. The key should be the shipment id and the value with it's details.

[31]:
```
# simple data for shipment
import pandas as pd

shipment={
    "Shipment_id":[101, 102, 103, 104, 105, 106],
    "Sender":[1, 4, 2, 1, 3, 5],
    "Receiver":[3, 1, 3, 5, 4, 2],
    "Start_date":["14-03-2020", "18-06-2020", "01-12-2020", "23-06-2020",␣
 ↪"29-08-2020", "28-08-2020"],
    "Delivery_date":["25-03-2020", "09-07-2020", None, "25-06-2020",␣
 ↪"10-09-2020", None],
    "Sender_location":["Area1", "Area2", "Area5", "Area1", "Area5", "Area3"],
    "Reciever_location":["Area6", "Area4", "Area1", "Area4", "Area4", "Area1"],
    "Delivery_status":["Delivered", "Delivered", "In-Transit", "Delivered",␣
 ↪"Delivered", "In-Transit"],
    "Shipping_cost":[198, 275, 200, 314, 275, 270]
}
```

[32]:
```
df = pd.DataFrame(shipment)
print(df)
```

```
   Shipment_id  Sender  Receiver  Start_date Delivery_date Sender_location  \
0          101       1         3  14-03-2020    25-03-2020            Area1
1          102       4         1  18-06-2020    09-07-2020            Area2
2          103       2         3  01-12-2020          None            Area5
3          104       1         5  23-06-2020    25-06-2020            Area1
4          105       3         4  29-08-2020    10-09-2020            Area5
5          106       5         2  28-08-2020          None            Area3

  Reciever_location Delivery_status  Shipping_cost
0             Area6       Delivered            198
1             Area4       Delivered            275
2             Area1      In-Transit            200
3             Area4       Delivered            314
4             Area4       Delivered            275
5             Area1      In-Transit            270
```

Q2. Create a Dictionary to store the information of clients given in the table

[33]:
```
client={
    "Client_id":[1, 2, 3, 4, 5],
    "Client_name":["Phillip", "Omega lll", "Ramya", "Romesh", "John"]
}

clients=pd.DataFrame(client)
print(clients)
```

```
   Client_id Client_name
0          1     Phillip
1          2   Omega lll
2          3       Ramya
3          4      Romesh
4          5        John
```

Q3. Write a code to replace client's id with their respective name in shipment dictionary using a loop and dictionary comprehension

[34]:
```
# Create a mapping from client id to client name
client_id_to_name = dict(zip(client['Client_id'], client['Client_name']))

# Replace Sender and Receiver ids with names in shipment dictionary
shipment_with_names = shipment.copy()
shipment_with_names['Sender'] = [client_id_to_name[sid] for sid in␣
 ↪shipment['Sender']]
shipment_with_names['Receiver'] = [client_id_to_name[rid] for rid in␣
 ↪shipment['Receiver']]

df=pd.DataFrame(shipment_with_names)
print(df)
```

```
   Shipment_id     Sender   Receiver  Start_date Delivery_date  \
0          101    Phillip      Ramya  14-03-2020    25-03-2020
1          102     Romesh    Phillip  18-06-2020    09-07-2020
2          103  Omega lll      Ramya  01-12-2020          None
3          104    Phillip       John  23-06-2020    25-06-2020
4          105      Ramya     Romesh  29-08-2020    10-09-2020
5          106       John  Omega lll  28-08-2020          None

  Sender_location Reciever_location Delivery_status  Shipping_cost
0           Area1              Area6       Delivered            198
1           Area2              Area4       Delivered            275
2           Area5              Area1      In-Transit            200
3           Area1              Area4       Delivered            314
4           Area5              Area4       Delivered            275
5           Area3              Area1      In-Transit            270
```

Q5. Print all shipment details that are sent by Phillip

[35]:
```
# Filter and print all shipment details where Sender is 'Phillip'
phillip_shipments = df[df['Sender'] == 'Phillip']
print(phillip_shipments)
```

```
   Shipment_id   Sender Receiver  Start_date Delivery_date Sender_location  \
0          101  Phillip    Ramya  14-03-2020    25-03-2020           Area1
3          104  Phillip     John  23-06-2020    25-06-2020           Area1

  Reciever_location Delivery_status  Shipping_cost
0             Area6       Delivered            198
3             Area4       Delivered            314
```

Q5. Print all shipment details that are received by Ramya

[36]:
```
# Filter and print all shipment details where sender is "Ramya"
ramys_shipments = df[df['Receiver'] == "Ramya"]
print(ramys_shipments)
```

```
   Shipment_id     Sender Receiver  Start_date Delivery_date Sender_location  \
0          101    Phillip    Ramya  14-03-2020    25-03-2020           Area1
2          103  Omega lll    Ramya  01-12-2020          None           Area5

  Reciever_location Delivery_status  Shipping_cost
0             Area6       Delivered            198
2             Area1      In-Transit            200
```

Q6. Print all shipments which are in 'In-Transit' status

[37]:
```
# Filter and print all shipment details where Delivery_status is "In-Transit"
in_transit_shipments = df[df['Delivery_status'] == "In-Transit"]
print(in_transit_shipments)
```

```
   Shipment_id     Sender   Receiver  Start_date Delivery_date  \
2          103  Omega lll      Ramya  01-12-2020          None
5          106       John  Omega lll  28-08-2020          None

  Sender_location Reciever_location Delivery_status  Shipping_cost
2           Area5             Area1      In-Transit            200
5           Area3             Area1      In-Transit            270
```

Q7. Print all shipments which are delivered within 7 days of courier Start date

[38]:
```
df['Start_date'] = pd.to_datetime(df['Start_date'], dayfirst=True)
df['Delivery_date'] = pd.to_datetime(df['Delivery_date'], dayfirst=True)

delivered_within_7 = df[
    (df['Delivery_status'] == 'Delivered') &
    ((df['Delivery_date'] - df['Start_date']).dt.days <= 7)
]
print(delivered_within_7)
```

```
   Shipment_id   Sender Receiver Start_date Delivery_date Sender_location  \
3          104  Phillip     John 2020-06-23    2020-06-25           Area1

  Reciever_location Delivery_status  Shipping_cost
3             Area4       Delivered            314
```

Q8. Print all shipments that are delivered after 15 days of the courier start date or have not yet been delivered.

[39]:
```
delivered_after_15_days = df[
    (df['Delivery_status'] == 'Delivered') &
    ((df['Delivery_date'] - df['Start_date']).dt.days > 15) |
    (df['Delivery_status'] != 'Delivered')
]
print(delivered_after_15_days)
```

```
   Shipment_id     Sender   Receiver Start_date Delivery_date Sender_location  \
1          102     Romesh    Phillip 2020-06-18    2020-07-09           Area2
2          103  Omega lll      Ramya 2020-12-01           NaT           Area5
5          106       John  Omega lll 2020-08-28           NaT           Area3

  Reciever_location Delivery_status  Shipping_cost
1             Area4       Delivered            275
2             Area1      In-Transit            200
5             Area1      In-Transit            270
```

Q9. Print the shipment details whose shipment cost in the range of 200 to 300

[40]:
```
shipment_cost_range = df[(df['Shipping_cost'] >= 200) & (df['Shipping_cost'] <=␣
↪300)]
print(shipment_cost_range)
```

```
   Shipment_id     Sender   Receiver Start_date Delivery_date Sender_location  \
1          102     Romesh    Phillip 2020-06-18    2020-07-09           Area2
2          103  Omega lll      Ramya 2020-12-01           NaT           Area5
4          105      Ramya     Romesh 2020-08-29    2020-09-10           Area5
5          106       John  Omega lll 2020-08-28           NaT           Area3

  Reciever_location Delivery_status  Shipping_cost
1             Area4       Delivered            275
2             Area1      In-Transit            200
4             Area4       Delivered            275
5             Area1      In-Transit            270
```

Q10. Print the name of all senders whose delivery status is 'in-transit'.

[41]:
```
# Get unique sender names where delivery status is 'In-Transit'
in_transit_senders = df[df['Delivery_status'] == 'In-Transit']['Sender'].
 ↪unique()
print(in_transit_senders)
```

```
['Omega lll' 'John']
```

Q11. Print the shipment ID that has the highest shipment cost

[43]:
```
highest_cost_shipment = df.loc[df['Shipping_cost'].idxmax(), 'Shipment_id']
print(highest_cost_shipment)
```

```
104
```

Q12. Print all the shipment details that are delivered by 'John' and received by 'Omega III'

[46]:
```
# Filter and print all shipments where sender is "John" and receiver is "Omega␣
 ↪lll"
john_to_omega_shipments = df[(df['Sender'] == 'John') & (df['Receiver'] ==␣
 ↪'Omega lll')]
print(john_to_omega_shipments)
```

```
   Shipment_id Sender   Receiver Start_date Delivery_date Sender_location  \
5          106   John  Omega lll 2020-08-28           NaT           Area3

  Reciever_location Delivery_status  Shipping_cost
5             Area1      In-Transit            270
```