In [1]:

```
import pandas as pd
```

In [2]:

```
df_Jan = pd.read_csv("monthly_revenue_plan - Jan_2024.csv")
df_Feb = pd.read_csv("monthly_revenue_plan - Feb_2024.csv")
df_Mar = pd.read_csv("monthly_revenue_plan - Mar_2024.csv")
df_April = pd.read_csv("monthly_revenue_plan - Apr_2024.csv")
df_May = pd.read_csv("monthly_revenue_plan - May_2024.csv")
df_June = pd.read_csv("monthly_revenue_plan - Jun_2024.csv")
df_July = pd.read_csv("monthly_revenue_plan - Jul_2024.csv")
df_Aug = pd.read_csv("monthly_revenue_plan - Aug_2024.csv")
df_Sept = pd.read_csv("monthly_revenue_plan - Sep_2024.csv")
df_Oct = pd.read_csv("monthly_revenue_plan - Oct_2024.csv")
df_Nov = pd.read_csv("monthly_revenue_plan - Nov_2024.csv")
df_Dec = pd.read_csv("monthly_revenue_plan - Dec_2024.csv")
```

In [4]:

```
df_Feb.head(4)
```

Out[4]:

	date	city_code	plans	plan_revenue_crores
0	01-Feb-24	400001	p2	7.86
1	01-Feb-24	110001	p2	5.43
2	01-Feb-24	700001	p2	5.36
3	01-Feb-24	560001	p2	5.47

In [3]:

```
concat_df = pd.concat([df_Jan,df_Feb,df_Mar,df_April,
                       df_May,df_June,df_July,df_Aug,
                       df_Sept,df_Oct,df_Nov,df_Dec],
                       ignore_index=True,axis=0)
concat_df
```

Out[3]:

	date	city_code	plans	plan_revenue_crores
0	01-Jan-24	400001	p1	6.26
1	01-Jan-24	110001	p1	4.58
2	01-Jan-24	700001	p1	4.29
3	01-Jan-24	560001	p1	4.26
4	01-Jan-24	600001	p1	3.60
...	...	...	...	...
175	01-Dec-24	800008	p3	1.33
176	01-Dec-24	641001	p3	0.54
177	01-Dec-24	160017	p3	0.90
178	01-Dec-24	122001	p3	0.47
179	01-Dec-24	492001	p3	5.66

180 rows × 4 columns

In [4]:

```
concat_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 180 entries, 0 to 179
Data columns (total 4 columns):
 #   Column              Non-Null Count  Dtype  
---  ------              --------------  -----  
 0   date                180 non-null    object 
 1   city_code           180 non-null    int64  
 2   plans               180 non-null    object 
 3   plan_revenue_crores  180 non-null    float64
dtypes: float64(1), int64(1), object(2)
memory usage: 5.8+ KB
```

In [6]:

```
concat_df["date"] = pd.to_datetime(concat_df["date"])
```

```
C:\Users\lenovo\AppData\Local\Temp\ipykernel_18016\1958413171.py:1: UserWarning:
Could not infer format, so each element will be parsed individually, falling back
to `dateutil`. To ensure parsing is consistent and as-expected, please specify a
format.
  concat_df["date"] = pd.to_datetime(concat_df["date"])
```

In [9]:

```
concat_df["plan_revenue_crores"] = concat_df["plan_revenue_crores"].astype(float
```

In [10]:

```
concat_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 180 entries, 0 to 179
Data columns (total 4 columns):
 #   Column              Non-Null Count  Dtype         
---  ------              --------------  -----         
 0   date                180 non-null    datetime64[ns]
 1   city_code           180 non-null    int64         
 2   plans               180 non-null    object        
 3   plan_revenue_crores  180 non-null    float64       
dtypes: datetime64[ns](1), float64(1), int64(1), object(1)
memory usage: 5.8+ KB
```

In [4]:

```
concat_df.to_csv("Revenue_data.csv",index=False)
```

In [1]:

```
import pandas as pd
import mysql.connector

# Connect to MySQL
db = mysql.connector.connect(
    host="127.0.0.1",
    user="root",
    password="riddhizala",
    database="gdb0041"
)
cursor = db.cursor()

# Read CSV file
df = pd.read_csv("Revenue_data.csv")

# Insert rows into MySQL
for index, row in df.iterrows():
    sql = """INSERT INTO revenue (date, city_code,
             plans, plan_revenue_crores)
             VALUES (%s, %s, %s, %s)"""
    values = (row['date'], row['city_code'], row['plans'],
              row['plan_revenue_crores'])
    cursor.execute(sql, values)

db.commit()
cursor.close()
db.close()
print("✅ Data inserted successfully!")
```

```
✅ Data inserted successfully!
```

```
In [ ]:
```