

# Python Programming

## Lab:- 27(Pandas Pivot Table)

Student Id:- AF0417098

Student Name:- Ankush

---

**Panda** is a powerful Python library used for data manipulation and analysis. It provides two main data structures:

- **Series:** A one-dimensional labeled array (similar to a list or column in a table).
- **DataFrame:** A two-dimensional labeled data structure, similar to a table or Excel spreadsheet.

A **Pandas pivot table** is a powerful tool used to summarize and aggregate data in a DataFrame. It allows you to reshape the data by specifying which columns should be grouped, aggregated, or displayed. Pivot tables are very similar to Excel pivot tables and provide great flexibility when working with large datasets.

### Key features of a pandas pivot table:

1. **Aggregation:** It allows you to aggregate data (like sum, mean, count, etc.) based on one or more keys.
2. **Grouping:** You can group data by multiple columns (e.g., region, manager, item) and calculate summary statistics.
3. **Multi-level Indexing:** Pivot tables can create multi-level rows and columns, allowing more detailed summaries.

### Parameters:

- **data:** The DataFrame to be summarized.
- **values:** The column(s) to aggregate (e.g., Sale\_amt).
- **index:** The columns to group by in the rows (e.g., Region, Manager).
- **columns:** The columns to group by in the columns.
- **aggfunc:** The aggregation function to apply (e.g., sum, mean, count).

Here are some key feature and components of the Pandas library:-

**DataFrame:** The DataFrame is one of the central data structures in Pandas. It is a two-dimensional table with rows and columns, similar to a spreadsheet or a SQL table. Each column in a DataFrame can have a different data type, making it suitable for heterogeneous and structured data.

**Series:** A Series is a one-dimensional array-like object in Pandas. It can be thought of as a single column of data within a DataFrame, with an associated index. Series are used for representing and working with one-dimensional data.

**Data Input and Output:** Pandas provides functions to read data from various file formats, such as CSV, Excel, SQL databases, and more. It also allows you to write data to these formats.

**Data Cleaning:** Pandas includes functions for handling missing data, removing duplicates, and transforming data to make it suitable for analysis.

**Data Transformation:** You can perform various data transformation operations, such as filtering, merging, and reshaping data. This is essential for data preprocessing and analysis.

**Grouping and Aggregation:** Pandas allows you to group data by one or more columns and perform aggregation operations on the groups. This is valuable for summarizing and analyzing data.

**Indexing and Selection:** Pandas provides powerful indexing capabilities, including label-based and integer-based indexing. This allows you to select and access specific parts of the data easily.

## Assignment Questions:-

Ques1:- Region-wise, Manager-wise and Salesman wise - To Find Total Sales Amount.

Program:-

```
lab27.py > ...
1 import pandas as pd
2 sales_data = pd.read_csv('salesdata.csv')
3 # Region-wise, Manager-wise, and Salesman-wise - To Find Total Sales Amount
4 total_sales_by_region_manager_salesman = sales_data.groupby(['Region', 'Manager', 'SalesMan'])['Sale_amt'].sum()
5   .reset_index()
6
7 # Display the result
8 print(total_sales_by_region_manager_salesman)
```

Output:-

	Region	Manager	SalesMan	Sale_amt
0	Central	Douglas	John	124016.0
1	Central	Hermann	Luis	206373.0
2	Central	Hermann	Shelli	33698.0
3	Central	Hermann	Sigal	125037.5
4	Central	Marth	Steven	14000.0
5	Central	Martha	Steven	185690.0
6	Central	Timothy	David	140955.0
7	East	Douglas	Karen	48204.0
8	East	Martha	Alexander	236703.0
9	East	Martha	Diana	36100.0
10	West	Douglas	Michael	66836.0
11	West	Timothy	Stephen	88063.0

**Ques 2. Item-Wise - To find total Units.**

**Program:-**

```
lab27.py > ...
1 import pandas as pd
2 sales_data = pd.read_csv('salesdata.csv')
3 # # Region-wise, Manager-wise, and Salesman-wise - To Find Total Sales Amount
4 # total_sales_by_region_manager_salesman = sales_data.groupby(['Region', 'Manager', 'SalesMan'])['Sale_amt'].
  sum().reset_index()
5
6 # # Display the result
7 # print(total_sales_by_region_manager_salesman)
8
9 # Item-wise - To find total Units
10 total_units_by_item = sales_data.groupby('Item')['Units'].sum().reset_index()
11
12 # Display the result
13 print(total_units_by_item)
14
```

```
14
13 total_units_by_item
12
```

**Output:-**

```
sktop/python programming/lab27.py"
   Item  Units
0  Cell Phone  278
1      Desk    10
2  Home Theater  722
3  Television  716
4  Video Games  395
25 C:\Users\Rezi Kumar\Desktop\python programming>
```

**Ques 3. Region-Wise, Item Wise - To find Sales Amt .**

**Program:-**

```
lab27.py > ...
1 import pandas as pd
2 sales_data = pd.read_csv('salesdata.csv')
3
4 # Region-wise, Item-wise - To find Sales Amount
5 sales_amount_by_region_item = sales_data.groupby(['Region', 'Item'])['Sale_amt'].sum().reset_index()
6
7 # Display the result
8 print(sales_amount_by_region_item)
9
```

**Output:-**

```
sktop/python programming/lab27.py"
   Region    Item  Sale_amt
0  Central  Cell Phone    6075.0
1  Central    Desk        875.0
2  Central  Home Theater  212000.0
3  Central  Television  596604.0
4  Central  Video Games   14215.5
5    East    Cell Phone   39375.0
6    East  Home Theater  117000.0
7    East  Television  155740.0
8    East  Video Games    8892.0
9    West    Cell Phone  17100.0
10   West    Desk        375.0
11   West  Home Theater   32000.0
12   West  Television  105424.0
PS C:\Users\Raj Kumar\Desktop\python programming> |
```

#### Ques 4. Manager-Wise - To Find Mean Sale Amt.

Program:-

```
C:\Users\Raj Kumar\Desktop\python programming\lab21.py
1 import pandas as pd
2 sales_data = pd.read_csv('salesdata.csv')
3
4 # Region-wise, Item-wise - To find Sales Amount
5 # sales_amount_by_region_item = sales_data.groupby(['Region', 'Item'])['Sale_amt'].sum().reset_index()
6
7 # Display the result
8 # print(sales_amount_by_region_item)
9
10 # Manager-wise - To Find Mean Sales Amount
11 mean_sales_by_manager = sales_data.groupby('Manager')['Sale_amt'].mean().reset_index()
12
13 # Display the result
14 print(mean_sales_by_manager)
```

```
14 print(mean_sales_by_manager)
13 # Display the result
12
```

Output:-

```
C:\Users\Raj Kumar\Desktop\python programming\lab27.py"
Manager      Sale_amt
0 Douglas  29882.000000
1 Hermann  30425.708333
2 Marth    14000.000000
3 Martha   35268.692308
4 Timothy  25446.444444
25 C:\Users\Raj Kumar\Desktop\python programming\
```