

# Data Transfer Between Files, Databases and DataFrames

Based on CBSE Curriculum

Class -11



## Chapter- 12

By-

Neha Tyagi

PGT CS

KV 5 Jaipur II Shift

Jaipur Region

# Introduction

- In the last chapter we have learnt about python pandas library in which we have gone through dataframes and series.
- In this chapter we will see that how to change data into .CSV file and how to import data in the program from a .CSV file. (.CSV file, is a file of Comma Seperated Values from which data can be taken to a dataframe and vice-versa).
- In this chapter we will learn that how to connect a database table with python by SQL commands.

# Data transfer between DataFrames and .CSV file

- CSV format is a kind of tabular data separated by comma and is stored in the form of plaintext.

Roll No	Name	Marks
101	Ramesh	77.5
102	Harish	45.6

Tabular Data

After conversion to CSV Format

```
Roll No., Name, Marks  
101, Ramesh, 77.5  
102, Harish, 45.6
```

## In CSV format-

- Each row of the table is stored in one row.
- The field-values of a row are stored together with comma after every field value.

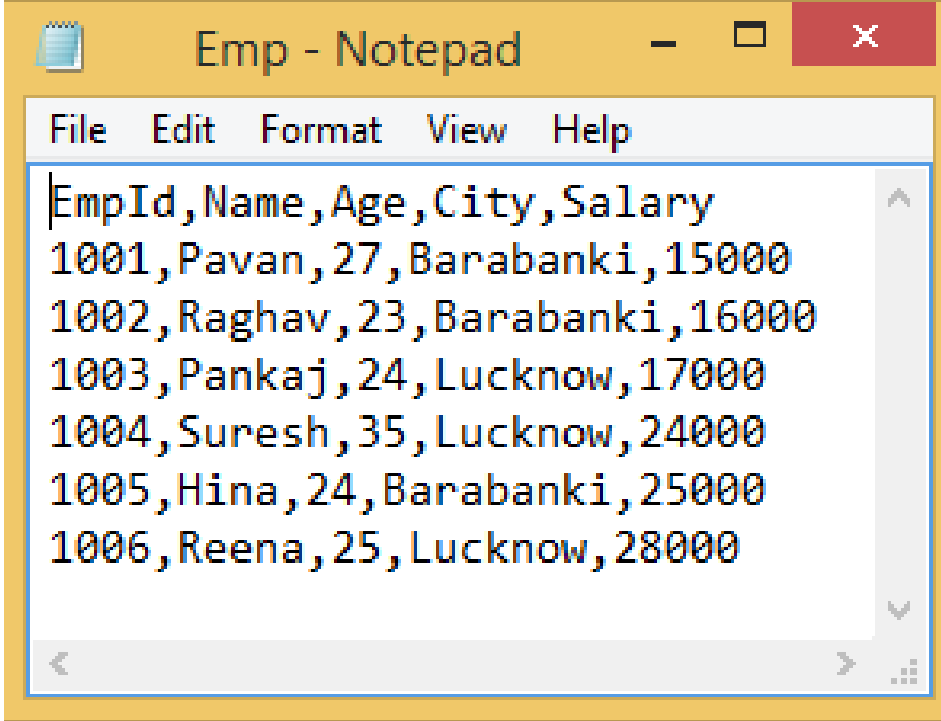
## Advantages of CSV format-

- A simple, compact and ubiquitous format for data storage.
- A common format for data interchange.
- It can be opened in popular spreadsheet packages like MS-EXCEL etc.
- Nearly all spreadsheets and databases support import/export to csv format.

# Loading Data from CSV to DataFrame

EmpId	Name	Age	City	Salary
1001	Pavan	27	Barabanki	15000
1002	Raghav	23	Barabanki	16000
1003	Pankaj	24	Lucknow	17000
1004	Suresh	35	Lucknow	24000
1005	Hina	24	Barabanki	25000
1006	Reena	25	Lucknow	28000

Emp.csv file , in tabular format



```
EmpId,Name,Age,City,Salary
1001,Pavan,27,Barabanki,15000
1002,Raghav,23,Barabanki,16000
1003,Pankaj,24,Lucknow,17000
1004,Suresh,35,Lucknow,24000
1005,Hina,24,Barabanki,25000
1006,Reena,25,Lucknow,28000
```

Emp.csv file , in Notepad format

# Reading from a CSV file to DataFrame

```
import pandas as pd
```

```
<DF>=pd.read_csv(<FilePath>)
```

- Assume the file path as c:\data\emp.csv then following type of file will be opened-

```
>>> import pandas as pd
>>> empdata=pd.read_csv("C:\\data\\emp.csv")
>>> empdata
```

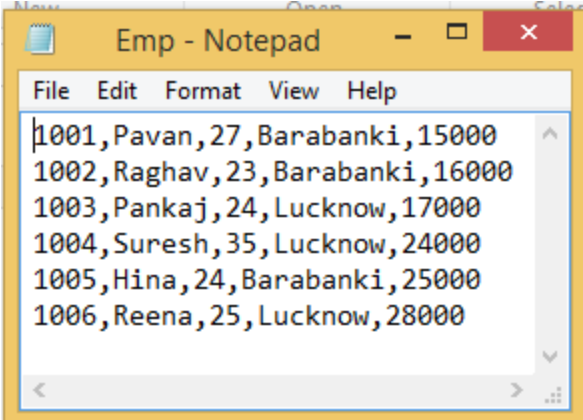
	EmpId	Name	Age	City	Salary
0	1001	Pavan	27	Barabanki	15000
1	1002	Raghav	23	Barabanki	16000
2	1003	Pankaj	24	Lucknow	17000
3	1004	Suresh	35	Lucknow	24000
4	1005	Hina	24	Barabanki	25000
5	1006	Reena	25	Lucknow	28000

# Reading from a CSV file to DataFrame

- If a file does not have top row i.e. Headings then it is possible to provide headings in python.

```
>>> empdata=pd.read_csv("C:\\data\\emp.csv", names=['RollNo.', 'EmpName', 'Age', 'City', 'Salary'])
>>> empdata
```

	RollNo.	EmpName	Age	City	Salary
0	1001	Pavan	27	Barabanki	15000
1	1002	Raghav	23	Barabanki	16000
2	1003	Pankaj	24	Lucknow	17000
3	1004	Suresh	35	Lucknow	24000
4	1005	Hina	24	Barabanki	25000
5	1006	Reena	25	Lucknow	28000



```
>>> empdata1=pd.read_csv("C:\\data\\emp.csv", header=None)
>>> empdata1
```

	0	1	2	3	4
0	1001	Pavan	27	Barabanki	15000
1	1002	Raghav	23	Barabanki	16000
2	1003	Pankaj	24	Lucknow	17000
3	1004	Suresh	35	Lucknow	24000
4	1005	Hina	24	Barabanki	25000
5	1006	Reena	25	Lucknow	28000

Headings does nt come from header=None

It takes data by skipping one row from skiprows = 1

# Reading selected lines from CSV file

```
>>> empdata=pd.read_csv("C:\\data\\emp.csv",\
                        names=['RollNo.', 'EmpName', 'Age', 'City', 'Salary'], nrows=3)
>>> empdata
```

	RollNo.	EmpName	Age	City	Salary
0	1001	Pavan	27	Barabanki	15000
1	1002	Raghav	23	Barabanki	16000
2	1003	Pankaj	24	Lucknow	17000

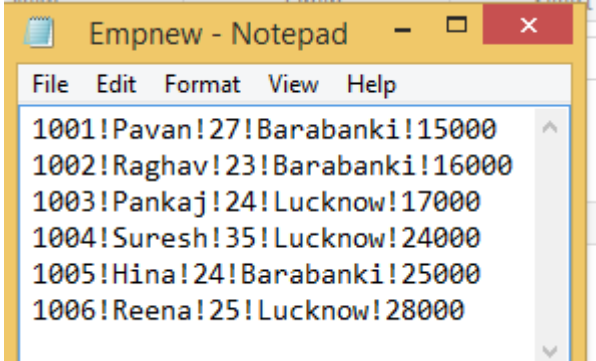
Use of nrows= <n>

## Reading from CSV file when separator is other than comma

```
>>> empdata=pd.read_csv("C:\\data\\empnew.csv",\
                        sep='!',\
                        names=['RollNo.', 'EmpName', 'Age', 'City', 'Salary'])
>>> empdata
```

	RollNo.	EmpName	Age	City	Salary
0	1001	Pavan	27	Barabanki	15000
1	1002	Raghav	23	Barabanki	16000
2	1003	Pankaj	24	Lucknow	17000
3	1004	Suresh	35	Lucknow	24000
4	1005	Hina	24	Barabanki	25000
5	1006	Reena	25	Lucknow	28000

Use of sep= <char>



# Writing from a DataFrame to CSV file

```
import pandas as pd  
<DF>.to_csv(<FilePath>)
```

or

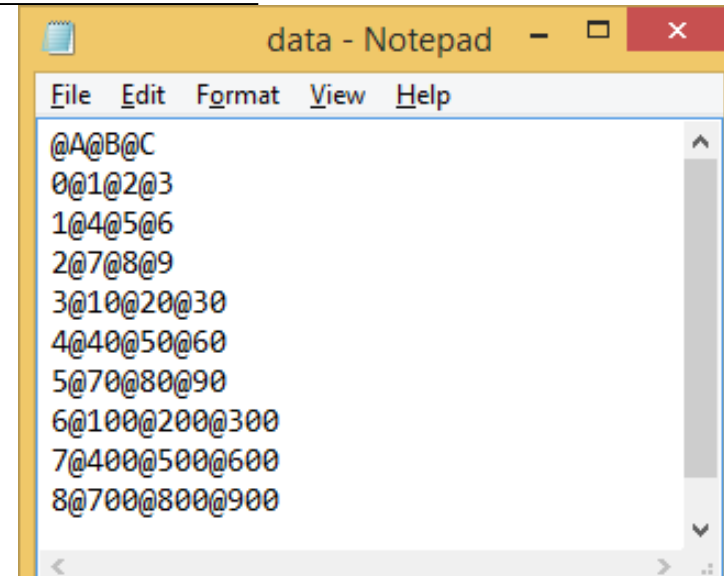
```
<DF>.to_csv(<FilePath>, sep=<char>)
```

- Suppose our file path is c:\data\data.csv then -

```
>>> df5  
   A    B    C  
0   1    2    3  
1   4    5    6  
2   7    8    9  
3  10   20   30  
4  40   50   60  
5  70   80   90  
6 100  200  300  
7 400  500  600  
8 700  800  900
```

If there os **NaN** values then it is stored in file as **empty string**.

Here @ is used as seperator



```
data - Notepad  
File Edit Format View Help  
@A@B@C  
0@1@2@3  
1@4@5@6  
2@7@8@9  
3@10@20@30  
4@40@50@60  
5@70@80@90  
6@100@200@300  
7@400@500@600  
8@700@800@900
```

```
>>> df5.to_csv("C:\\data\\data.csv", sep='@')
```



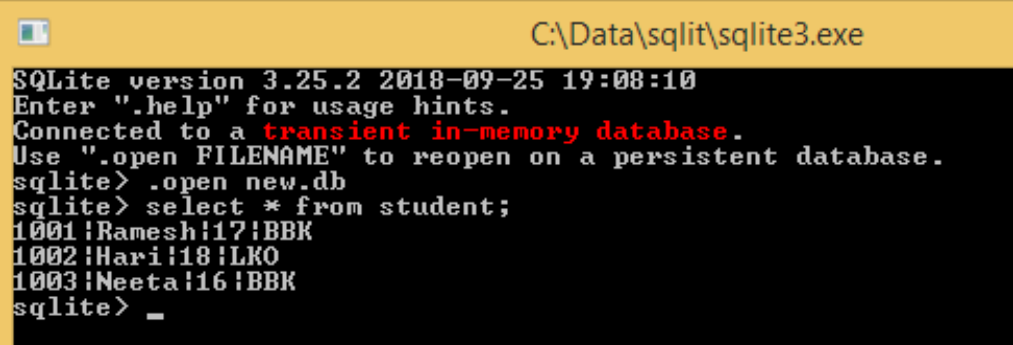
# Data transfer between DataFrames and SQL Database

- In this chapter we will learn that how to transfer data from SQL table with the help of sqlite3 library of python. sqlite3 comes inbuilt with python. sqlite3 library deals with SQLite databases.
- Use [www.sqlite.org/download.html](http://www.sqlite.org/download.html) to download Sqlite3.
- We work on `sqlite>` prompt on Sqlite3. It supports all commands of SQL which are supported by mysql.

# Data transfer between DataFrames and SQL Database

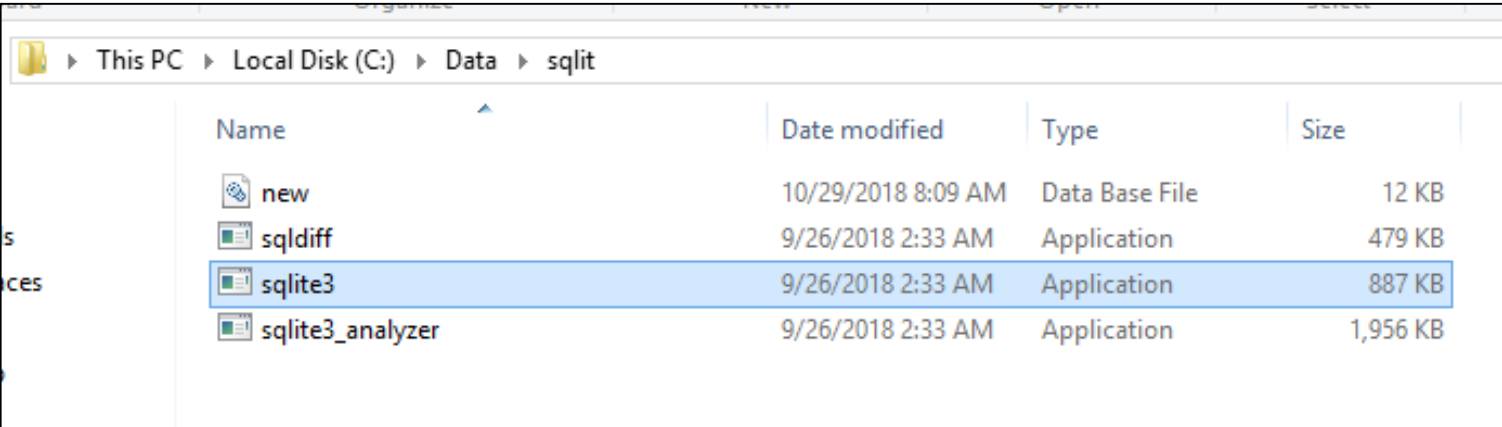
```
>>> import pandas as pd
>>> import sqlite3 as sq
>>> conn=sq.connect("c:\\data\\sqlit\\new.db")
>>> df=pd.read_sql("Select * from Student;",conn)
>>> df
```

```
   roll  name  age  city
0  1001  Ramesh   17  BBK
1  1002    Hari   18  LKO
2  1003   Neeta   16  BBK
```



```
C:\Data\sqlit\sqlite3.exe
SQLite version 3.25.2 2018-09-25 19:08:10
Enter ".help" for usage hints.
Connected to a transient in-memory database.
Use ".open FILENAME" to reopen on a persistent database.
sqlite> .open new.db
sqlite> select * from student;
1001|Ramesh|17|BBK
1002|Hari|18|LKO
1003|Neeta|16|BBK
sqlite> _
```

Creation of table in sqlite3 is shown here.



Name	Date modified	Type	Size
new	10/29/2018 8:09 AM	Data Base File	12 KB
sqldiff	9/26/2018 2:33 AM	Application	479 KB
sqlite3	9/26/2018 2:33 AM	Application	887 KB
sqlite3_analyzer	9/26/2018 2:33 AM	Application	1,956 KB

Here sqlite3 is installed.

# Data transfer between DataFrames and SQL Database

```
>>> empdata
   RollNo.  EmpName  Age  City  Salary
0      1001    Pavan   27  Barabanki  15000
1      1002   Raghav   23  Barabanki  16000
2      1003   Pankaj   24   Lucknow  17000
3      1004   Suresh   35   Lucknow  24000
4      1005     Hina   24  Barabanki  25000
5      1006    Reena   25   Lucknow  28000
>>> import sqlite3 as sq
>>> conn=sq.connect("c:\\data\\sqlit\\new.db")
>>> empdata.to_sql("Employee",conn)
```

```
sqlite>
sqlite> Select * from Employee;
0|1001|Pavan|27|Barabanki|15000
1|1002|Raghav|23|Barabanki|16000
2|1003|Pankaj|24|Lucknow|17000
3|1004|Suresh|35|Lucknow|24000
4|1005|Hina|24|Barabanki|25000
5|1006|Reena|25|Lucknow|28000
sqlite>
```

Data has transferred to database from DataFrame.

# Plotting from DataFrames

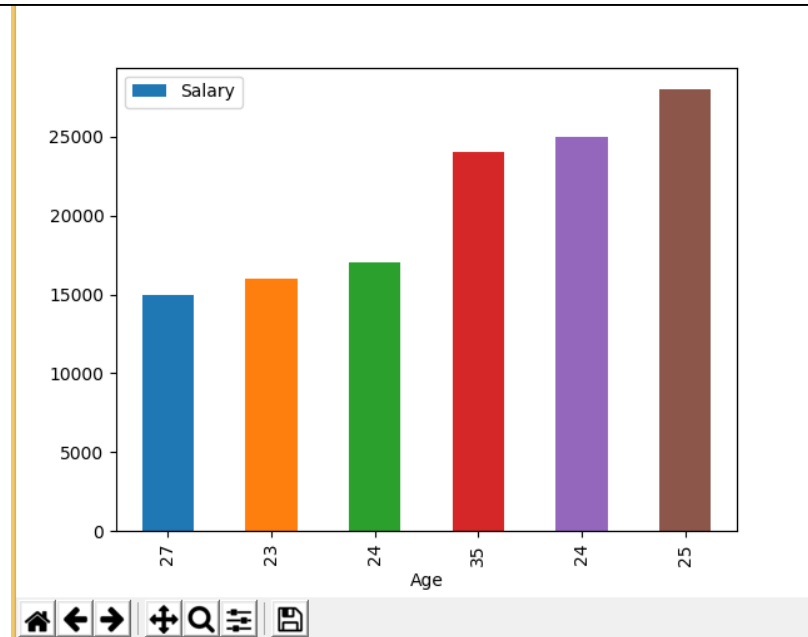
```
C:\Windows\system32\cmd.exe

C:\Users\KUBBKServer\AppData\Local\Programs\Python\Python36\Scripts>pip install
matplotlib
Collecting matplotlib
  Downloading https://files.pythonhosted.org/packages/f1/f2/0e89288925f87716499f
1984c3d7076f4eef2c93db7e30df4844b6e30978/matplotlib-3.0.1-cp36-cp36m-win_amd64.w
hl (8.9MB)
  100% |#####| 8.9MB 262kB/s
Requirement already satisfied: python-dateutil>=2.1 in c:\users\kubbkserver\appd
ata\local\programs\python\python36\lib\site-packages (from matplotlib) (2.7.3)
Collecting kiwisolver>=1.0.1 (from matplotlib)
  Downloading https://files.pythonhosted.org/packages/44/72/16630c3392eba03788ad
87949390516bbc488e8e118047a3b824631d21a6/kiwisolver-1.0.1-cp36-none-win_amd64.wh
l (57kB)
  100% |#####| 61kB 145kB/s
Collecting pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 (from matplotlib)
  Downloading https://files.pythonhosted.org/packages/2b/4a/f06b45ab9690d4c37641
ec776f7ad691974f4cf6943a73267475b05cbfca/pyparsing-2.2.2-py2.py3-none-any.whl (5
7kB)
  100% |#####| 61kB 231kB/s
Collecting cycler>=0.10 (from matplotlib)
  Downloading https://files.pythonhosted.org/packages/f7/d2/e07d3ebb2bd7af696440
ce7e754c59dd546ffe1bbe732c8ab68b9c834e61/cycler-0.10.0-py2.py3-none-any.whl
Requirement already satisfied: numpy>=1.10.0 in c:\users\kubbkserver\appdata\loc
al\programs\python\python36\lib\site-packages (from matplotlib) (1.15.2)
```

Use pip install for installing matplotlib module.

# Plotting from DataFrames

```
>>> import matplotlib.pyplot as plt
>>> empdata
   index  RollNo.  EmpName  Age    City  Salary
0      0     1001    Pavan   27  Barabanki  15000
1      1     1002   Raghav   23  Barabanki  16000
2      2     1003   Pankaj   24   Lucknow  17000
3      3     1004   Suresh   35   Lucknow  24000
4      4     1005    Hina   24  Barabanki  25000
5      5     1006   Reena   25   Lucknow  28000
>>> empdata.plot(kind='bar',x='Age',y='Salary')
<matplotlib.axes._subplots.AxesSubplot object at 0x0000007DE1EAEA58>
>>> plt.show()
```



# Thank you

Please follow us on our blog

[www.pythontrends.wordpress.com](http://www.pythontrends.wordpress.com)