

Python pandas

Part A : Data Handling

Ques = 1) Create a panda's series from a dictionary of values and a ndarray.

Apply all the attributes of series on the above created Series.

Solution:

```
import pandas as pd
import numpy as np
data_dict = {'a': 10, 'b': 20, 'c': 30}
series_dict = pd.Series(data_dict)
data_array = np.array([100, 200, 300])
series_array = pd.Series(data_array)
print("Series from dictionary:")
print(series_dict)
print("\nSeries from ndarray:")
print(series_array)
print("\nAttributes of series_dict:")
print("Index:", series_dict.index)
print("Values:", series_dict.values)
print("Datatype:", series_dict.dtype)
print("Shape:", series_dict.shape)
print("Size:", series_dict.size)
```

Ques = 2) Create a series that stores the area of some states in km².

a) Write code to find out the biggest and smallest three areas from the given series.

b) Write code to find out the areas that are more than 50000 km².

Solution:

```
import pandas as pd

area_series = pd.Series({'Delhi': 1484, 'Rajasthan': 342239, 'Uttar Pradesh': 243286,
'Maharashtra': 307713, 'Gujarat': 196024, 'Haryana': 44212})

print("Top 3 biggest areas:")

print(area_series.sort_values(ascending=False).head(3))

print("\nTop 3 smallest areas:")

print(area_series.sort_values().head(3))

print("\nAreas more than 50000 km²:")

print(area_series[area_series > 50000])
```

Ques = 3) Write a program to create a Series object with 6 random integers and having indexes as: ['p', 'q', 'r', 'n', 't', 'v']. Also write program to calculate cubes of the Series values.

Solution:

```
import pandas as pd
import numpy as np

data = np.random.randint(1, 100, 6)
index = ['p', 'q', 'r', 'n', 't', 'v']
series = pd.Series(data, index=index)

print("Original Series:")

print(series)

print("\nCubes of Series values:")

print(series ** 3)
```

Ques = 4) Consider the following dataframe: CORONA and answer the questions given below:

| ID | State | Cases |
|-----|---------|-------|
| 100 | Delhi | 3000 |
| 110 | Mumbai | 4000 |
| 120 | Chennai | 5000 |
| 130 | Surat | 4500 |

Create the above-given dictionary with the given indexes.

- (a) Write code to add a new column “Recovery” using the series method to store the number of patients recovered in every state.**
- (b) To add a new column “Deaths” using the assign() method to store the number of deaths in every state.**
- (c) To add a new row to store details of another state using loc (assume values).**
- (d) To add a new column “Percentage” using the insert() method to store the percentage of recovery in every state (assume values). The column should be added as the fourth column in the dataframe.**
- (e) To delete the column “Percentage” using del command.**
- (f) To delete the column “Deaths” using pop() method.**
- (g) To insert a new row of values using iloc[] at the 1st position.**
- (h) To delete Cases and State temporarily from the dataframe.**
- (i) Create a dataframe from two series – Name and Grade, Name and Marks of five students.**
 - (a) Display the first three records from student dataframe.**
 - (b) Display the last two records from student dataframe.**
- (j) Create a dataframe of dictionary consisting of Name, Sub1, Sub2, Sub3, Sub4, Sub5 of five students.**
 - (a) Display the dataframe.**
 - (b) Display the first 5 rows and bottom 3 rows of student dataframe.**
- (k) Create two dataframes of salary of five employees and do the following:**
 - (a) Display both the dataframes.**
 - (b) Add 5000 as bonus in both dataframes and display them.**

(l) Create a dataframe using list [10, 11, 12, 13, 14] [23, 34, 45, 32, 65] [55, 60, 65, 70, 75] and do the following:

(a) Display the dataframe.

(m) Create a dataframe of [23, 25], [34], [43, 44, 45, 46] and do the following:

(a) Display the dataframe. Notice that the missing value is represented by NaN.

(b) Replace the missing value with 0.

(c) Replace the missing value with -1, -2, -3, -4 for columns 0, 1, 2, 3.

(d) Replace the missing value by copying the value from the above cell.

Solution:

(j) DataFrame from dictionary

```
student_data = { 'Name': ['A', 'B', 'C', 'D', 'E'], 'Sub1': [90, 80, 85, 75, 88], 'Sub2': [70, 60, 85, 90, 95], 'Sub3': [80, 85, 78, 92, 88]}
```

```
import pandas as pd
```

```
import numpy as np
```

Initial DataFrame

```
data = {'State': ['Delhi', 'Mumbai', 'Chennai', 'Surat'], 'Cases': [3000, 4000, 5000, 4500]}
```

```
corona = pd.DataFrame(data, index=[100, 110, 120, 130])
```

(a) Add Recovery column

```
recovery = pd.Series([2800, 3900, 4700, 4200], index=[100, 110, 120, 130])
```

```
corona['Recovery'] = recovery
```

(b) Add Deaths column using assign()

```
corona = corona.assign(Deaths=[100, 80, 150, 100])
```

(c) Add new row using loc

```
corona.loc[140] = ['Bangalore', 6000, 5800, 200]
```

(d) Add Percentage column using insert()

```
percentage = [93.33, 97.5, 94, 93.33, 96.67]
```

```
corona.insert(3, 'Percentage', percentage)
```

(e) Delete Percentage using del

```
del corona['Percentage']
```

(f) Delete Deaths using pop()

```
corona.pop('Deaths')
```

(g) Insert new row at 1st position using iloc

```
new_row = pd.DataFrame({'State': ['Pune'], 'Cases': [3500], 'Recovery': [3300]}, index=[105])
```

```
corona = pd.concat([corona.iloc[:1], new_row, corona.iloc[1:]])
```

(h) Delete Cases and State temporarily

```
print(corona.drop(['Cases', 'State'], axis=1))
```

(i) Dataframe from series: Name + Grade, Name + Marks

```
names = pd.Series(['A', 'B', 'C', 'D', 'E'])
```

```
grades = pd.Series(['A+', 'B', 'A', 'C', 'B+'])
```

```
marks = pd.Series([90, 80, 85, 70, 88])
```

```
students1 = pd.DataFrame({'Name': names, 'Grade': grades})
```

```
students2 = pd.DataFrame({'Name': names, 'Marks': marks})
```

(i-a) First 3 records

```
print(students1.head(3))
```

(i-b) Last 2 records

```
print(students2.tail(2))
```

```
], 'Sub4': [85, 89, 90, 86, 87], 'Sub5': [70, 75, 80, 85, 90] }
```

```
student_df = pd.DataFrame(student_data)
```

(j-a) Full dataframe

```
print(student_df)
```

(j-b) First 5 and last 3 rows

```
print(student_df.head(5))
```

```
print(student_df.tail(3))
```

(k) Employee salary dataframes

```
df1 = pd.DataFrame({'Employee': ['A', 'B', 'C', 'D', 'E'], 'Salary': [20000, 25000, 22000, 24000, 23000]})
```

```
df2 = pd.DataFrame({'Employee': ['A', 'B', 'C', 'D', 'E'], 'Salary': [21000, 26000, 22500, 24500, 23500]})
```

(k-a) Display

```
print(df1)
```

```
print(df2)
```

(k-b) Add 5000 bonus

```
df1['Bonus'] = df1['Salary'] + 5000
```

```
df2['Bonus'] = df2['Salary'] + 5000
```

(l) Create dataframe from lists

```
df3 = pd.DataFrame({ 0: [10, 11, 12, 13, 14], 1: [23, 34, 45, 32, 65], 2: [55, 60, 65, 70, 75] })
```

```
print(df3)
```

(m) DataFrame with missing values

```
df4 = pd.DataFrame([ [23, 25, np.nan, np.nan], [34, np.nan, np.nan, np.nan], [43, 44, 45, 46] ])
```

```
print(df4)
```

(m-b) Replace missing with 0

```
df4.fillna(0, inplace=True)
```

```
print(df4)
```

(m-c) Replace with -1, -2, -3, -4

```
df4.iloc[0, 2] = -1
```

```
df4.iloc[1, 1] = -2
```

```
df4.iloc[1, 2] = -3
```

```
df4.iloc[1, 3] = -4
```

```
print(df4)
```

(m-d)

```
df4.fillna(method='ffill', inplace=True)
```

```
print(df4)
```

Part C : Data Management

Ques =i) Hospital table data:

| PNo | Name | Age | Department | Dateofadm | Charges | Sex |
|------------|-------------|------------|-------------------|------------------|----------------|------------|
| 1 | Arpit | 62 | Surgery | 2008-01-12 | 300 | M |
| 2 | Zarina | 22 | ENT | 2007-12-12 | 250 | F |
| 3 | Kareem | 32 | Orthopaedic | 2008-02-19 | 200 | M |
| 4 | Arun | 12 | Surgery | 2008-01-11 | 300 | M |
| 5 | Zubin | 30 | ENT | 2008-01-12 | 250 | M |
| 6 | Ketaki | 16 | ENT | 2008-02-24 | 250 | F |
| 7 | Ankita | 29 | Cardiology | 2008-02-20 | 800 | F |
| 8 | Zareen | 45 | Gynaecology | 2008-02-22 | 300 | F |
| 9 | Kush | 19 | Cardiology | 2008-01-13 | 800 | M |
| 10 | Shilpa | 23 | Nuclear Medicine | 2008-01-20 | 400 | F |

Ques =1) Write the command to create HOSPITAL table.

Answer: CREATE TABLE HOSPITAL (PNo INT, Name VARCHAR(30), Age INT, Department VARCHAR(30), Dateofadm DATE, Charges INT, Sex CHAR(1));

Ques =2) Write a command to describe the structure of the above table.

Answer: DESC HOSPITAL;

Ques =3) To select all the information of patients of cardiology department.

Answer: SELECT * FROM HOSPITAL WHERE Department = 'Cardiology';

Quest =4) To list the names of female patients who are in ENT department.

Answer: SELECT Name FROM HOSPITAL WHERE Department = 'ENT' AND Sex = 'F';

Ques =5) To list names of all patients with their date of admission in ascending order.

Answer: SELECT Name, Dateofadm FROM HOSPITAL ORDER BY Dateofadm ASC;

Ques =6) To display Patient's Name, Charges, Age of only female patients.

Answer: SELECT Name, Charges, Age FROM HOSPITAL WHERE Sex = 'F';

Ques =7) To count the number of patients with Age < 30

Answer: SELECT COUNT(*) FROM HOSPITAL WHERE Age < 30;

Ques =8) Display the department wise total charges.

Answer: SELECT Department, SUM(Charges) FROM HOSPITAL GROUP BY Department;

Ques =9) Display the department wise total charges whose maximum charges more than equal to 300.

Answer: SELECT Department, SUM(Charges) FROM HOSPITAL GROUP BY Department HAVING MAX(Charges) >= 300;

Ques =10) Add one column in the table with the name Address of type char(20).

Answer: ALTER TABLE HOSPITAL ADD Address CHAR(20);

Ques =11) Modify the column Address as char(25).

Answer : ALTER TABLE HOSPITAL MODIFY Address CHAR(25);

Ques =12) Change the name of the column Address to Home_address.

Answer: ALTER TABLE HOSPITAL RENAME COLUMN Address TO Home_address;

Ques =13) Drop the column Home_Address.

Answer: ALTER TABLE HOSPITAL DROP COLUMN Home_address;

Ques =ii) Write the SQL commands for delete (a) to (f) on the basis of the table STUDENT.

TABLE : STUDENT

| No. | Name | Age | Department | Dateofadm | Fees | Sex |
|-----|---------|-----|------------|------------|------|-----|
| 1 | Pankaj | 24 | Computer | 1997-01-10 | 120 | M |
| 2 | Shalini | 21 | History | 1998-03-24 | 200 | F |
| 3 | Sanjay | 22 | Hindi | 1996-12-12 | 300 | M |

| | | | | | | |
|----------|---------|----|----------|------------|-----|---|
| 4 | Sudha | 25 | History | 1997-07-01 | 400 | F |
| 5 | Rakesh | 22 | Hindi | 1997-09-05 | 250 | M |
| 6 | Shakeel | 30 | History | 1998-06-27 | 300 | M |
| 7 | Surya | 34 | Computer | 1997-02-25 | 210 | M |
| 8 | Shikha | 23 | Hindi | 1997-07-31 | 200 | F |

Ques =a) To show all information about the students of History department.

Answer: SELECT * FROM STUDENT WHERE Department = 'History';

Ques =b) To list the names of female students who are in History department.

Answer: SELECT Name FROM STUDENT WHERE Department = 'History' AND Sex = 'F';

Ques =c) To list names of all students who are in Hindi department.

Answer: SELECT Name FROM STUDENT WHERE Department = 'Hindi';

Ques =d) To display Student's Name, Fees, age of male students in ascending order.

Answer: SELECT Name, Fees, Age FROM STUDENT WHERE Sex = 'M' ORDER BY Name ASC;

Ques =e) To count the number of students with age > 23.

Answer: SELECT COUNT(*) FROM STUDENT WHERE Age > 23;

Ques =f) To insert a new row in the STUDENT table with the following data: 'Zaheer', 36, 'Computer', '1997-03-12', 230, 'M'

Answer: INSERT INTO STUDENT (Name, Age, Department, Dateofadm, Fees, Sex) VALUES ('Zaheer', 36, 'Computer', '1997-03-12', 230, 'M');

Ques =iii) Write SQL for (a) to (f) on the basis of the table TEACHER.

TABLE : TEACHER

| No. | Name | Age | Department | Dateofadm | Fees | Sex |
|------------|-------------|------------|-------------------|------------------|-------------|------------|
| 1 | Jugal | 34 | Computer | 1997-01-10 | 12000 | M |
| 2 | Sharmila | 31 | History | 1998-03-24 | 20000 | F |
| 3 | Sandeep | 32 | Maths | 1996-12-12 | 30000 | M |

| | | | | | | |
|----------|----------|----|----------|------------|-------|---|
| 4 | Sangeeta | 35 | History | 1999-07-01 | 40000 | F |
| 5 | Rakesh | 42 | Maths | 1997-09-05 | 25000 | M |
| 6 | Shyam | 50 | History | 1998-06-27 | 30000 | M |
| 7 | Shiv Om | 44 | Computer | 1997-02-25 | 21000 | M |
| 8 | Shalakha | 33 | Maths | 1997-07-31 | 20000 | F |

(a) To show all information about the teachers of History department.

Answer: SELECT * FROM TEACHER WHERE Department = 'History';

(b) To list the name of female teachers who are in Maths department.

Answer: SELECT Name FROM TEACHER WHERE Department = 'Maths' AND Sex = 'F';

(c) To list names of all teachers along with their date of joining in ascending order.

Answer: SELECT Name, Dateofjoin FROM TEACHER ORDER BY Dateofjoin ASC;

(d) To display teacher's Name, Salary, age for male teachers only.

Answer: SELECT Name, Salary, Age FROM TEACHER WHERE Sex = 'M';

(e) To count the number of teachers with age > 23.

Answer: SELECT COUNT(*) FROM TEACHER WHERE Age > 23;

(f) To insert a new row in the TEACHER table with the following data:

9, 'Raja', 26, 'Computer', '1995-05-13', 23000, 'M'

Answer: INSERT INTO TEACHER VALUES (9, 'Raja', 26, 'Computer', '1995-05-13', 23000, 'M');

Ques =iv) Write SQL commands for (a) to (d) on the basis of tables CLUB and COACHES:

TABLE : CLUB

| COACH_ID | COACH_NAME | AGE | SPORTS | DATEOFAPP | PAY | SEX |
|----------|------------|-----|--------|------------|------|-----|
| 1 | Kukreja | 35 | Karate | 2002-03-27 | 1000 | M |
| 2 | Ravina | 34 | Karate | 2004-01-20 | 1200 | F |
| 3 | Karan | 34 | Squash | 2004-02-19 | 2000 | M |

| | | | | | | |
|-----------|---------|----|------------|------------|------|---|
| 4 | Tarun | 33 | Basketball | 2004-01-01 | 1500 | M |
| 5 | Zubin | 36 | Swimming | 2004-01-12 | 750 | M |
| 6 | Ketaki | 36 | Swimming | 2004-02-24 | 800 | F |
| 7 | Ankita | 39 | Squash | 2004-02-20 | 2200 | F |
| 8 | Zareen | 37 | Karate | 2004-02-22 | 1100 | F |
| 9 | Kush | 41 | Swimming | 2004-01-13 | 900 | M |
| 10 | Shailya | 37 | Basketball | 2004-02-19 | 1700 | M |

a) To show all information about the swimming coaches in the club.

Answer: SELECT * FROM CLUB WHERE SPORTS = 'SWIMMING';

(b) To list names of all coaches with their date of appointment (DATEOFAPP) in descending order.

Answer: SELECT COACH_NAME, DATEOFAPP FROM CLUB ORDER BY DATEOFAPP DESC;

(c) To display a report, showing coach name, pay, age and bonus (15% of pay) for all the coaches.

Answer: SELECT COACH_NAME, PAY, AGE, PAY * 0.15 AS BONUS FROM CLUB;

(d) To insert a new row in the CLUB table with the following data: 11, 'RAJIV', 40, 'Hockey', '2000-05-27', 2006, 'M'

Answer: INSERT INTO CLUB VALUES (11, 'RAJIV', 40, 'Hockey', '2000-05-27', 2006, 'M');

Ques =v) Write SQL for (a) to (d) on the basis of the table GRADUATE.

TABLE : GRADUATE

| SNO | NAME | STIPEND | SUBJECT | AVERAGE | DIV1 |
|-----|---------|---------|-------------|---------|------|
| 1 | KARAN | 400 | PHYSICS | 68 | 1 |
| 2 | DIVAKAR | 450 | COMPUTER SC | 68 | 1 |
| 3 | DIVYA | 300 | CHEMISTRY | 62 | 2 |
| 4 | ARUN | 350 | PHYSICS | 63 | 1 |
| 5 | SABINA | 500 | MATHEMATICS | 70 | 1 |
| 6 | JOHN | 400 | CHEMISTRY | 55 | 2 |
| 7 | ROBERT | 250 | PHYSICS | 64 | 1 |
| 8 | RUBINA | 450 | MATHEMATICS | 68 | 1 |
| 9 | VIKAS | 500 | COMPUTER SC | 62 | 1 |

| | | | | | |
|----|-------|-----|-------------|----|---|
| 10 | MOHAN | 300 | MATHEMATICS | 57 | 2 |
|----|-------|-----|-------------|----|---|

(a) List the names of those students who have obtained DIVI = 1 sorted by NAME.

Answer: SELECT NAME FROM GRADUATE WHERE DIVI = 1 ORDER BY NAME;

(b) Display the report, listing NAME, STIPEND, SUBJECT and amount of stipend received in a year assuming that the STIPEND is paid every month.

Answer: SELECT NAME, STIPEND, SUBJECT, STIPEND * 12 AS YEARLY_STIPEND FROM GRADUATE;

(c) To count the number of students who are either PHYSICS or COMPUTER SC graduates.

Answer: SELECT COUNT(*) FROM GRADUATE WHERE SUBJECT = 'PHYSICS' OR SUBJECT = 'COMPUTER SC';

(d) To insert a new row in the GRADUATE table:

11, 'KAJOL', 300, 'COMPUTER SC', 75, 1

Answer: INSERT INTO GRADUATE VALUES (11, 'KAJOL', 300, 'COMPUTER SC', 75, 1);

Ques =vi) Write SQL commands for (a) to (g) on the basis of the table SPORTS.

TABLE : SPORTS

| StudentNo | Class | Name | Game1 | Grade1 | Game2 | Grade2 |
|-----------|-------|---------|------------|--------|-----------|--------|
| 10 | 7 | Sameer | Cricket | B | Swimming | A |
| 11 | 8 | Sujit | Tennis | A | Skating | C |
| 12 | 7 | Kamal | Swimming | B | Football | B |
| 13 | 7 | Veena | Tennis | C | Tennis | A |
| 14 | 9 | Archana | Basketball | A | Cricket | A |
| 15 | 10 | Arpit | Cricket | A | Athletics | C |

(a) Display the names of the students who have grade 'C' in either Game1 or Game2 or both.

Answer: SELECT Name FROM SPORTS WHERE Grade1 = 'C' OR Grade2 = 'C';

(b) Display the number of students getting grade 'A' in Cricket.

Answer: SELECT COUNT(*) FROM SPORTS WHERE (Game1 = 'Cricket' AND Grade1 = 'A') OR (Game2 = 'Cricket' AND Grade2 = 'A');

(c) Display the names of the students who have same game for both Game1 and Game2.

Answer: SELECT Name FROM SPORTS WHERE Game1 = Game2;

(d) Display the games taken up by the students, whose names start with 'A'.

Answer: SELECT Game1, Game2 FROM SPORTS WHERE Name LIKE 'A%';

(e) Add a new column named 'Marks'.

Answer: ALTER TABLE SPORTS ADD Marks INT;

(f) Assign a value 200 for Marks for all those who are getting grade 'B' or grade 'A' in both Game1 and Game2.

Answer: UPDATE SPORTS SET Marks = 200 WHERE (Grade1 IN ('A', 'B') AND Grade2 IN ('A', 'B'));

(g) Arrange the whole table in the alphabetical order of Name.

Answer: SELECT * FROM SPORTS ORDER BY Name;

Ques =vii) Given the following lab relations : Write SQL commands for questions (a) to (f).

TABLE : LAB

| No | ItemName | CostPerItem | Quantity | DateOfPurchase | Warranty | Operational |
|----|----------|-------------|----------|----------------|----------|-------------|
| 1 | Computer | 60000 | 9 | 2006-05-21 | 2 | 7 |
| 2 | Printer | 15000 | 3 | 2007-05-21 | 4 | 2 |
| 3 | Scanner | 18000 | 1 | 2008-08-29 | 3 | 1 |
| 4 | Camera | 21000 | 2 | 2006-06-13 | 1 | 2 |
| 5 | Hub | 8000 | 1 | 2009-04-13 | 2 | 1 |
| 6 | UPS | 5000 | 5 | | 1 | 4 |
| 7 | Plotter | 25000 | 2 | | 2 | 2 |

(a) To select the ItemName purchased after 2007-10-31.

Answer: SELECT ItemName FROM LAB WHERE DateofPurchase > '2007-10-31';

(b) To list the ItemName, which are within the Warranty period till present date (i.e. current date).

Answer: SELECT ItemName FROM LAB WHERE DATE_ADD(DateofPurchase, INTERVAL Waranty YEAR) > CURRENT_DATE;

(c) To list the ItemName in ascending order of the date of purchase where quantity is more than 3.

Answer: SELECT ItemName FROM LAB WHERE Quantity > 3 ORDER BY DateofPurchase ASC;

(d) To display ItemName, CostPerItem, and Quantity whose Warranty is over.

Answer: SELECT ItemName, CostPerItem, Quantity FROM LAB WHERE DATE_ADD(DateofPurchase, INTERVAL Waranty YEAR) < CURRENT_DATE;

(e) To count the number of items whose cost is more than 10000.

Answer: SELECT COUNT(*) FROM LAB WHERE CostPerItem > 10000;

(f) To insert a new record in the Lab table with the following data:

8, 'VCR', 10000, 2, '2010-02-02', 1, 2

Answer: INSERT INTO LAB VALUES (8, 'VCR', 10000, 2, '2010-02-02', 1, 2);