Practical List

**Data Handling Using Pandas**

1. Create a panda’s series from the given dictionary of values and a ndarray
2. Create a series from the given list of number and perform the following operations

Numbers =[12,45,67,45,78,89,90,56,90,56,67,34,76,90,67,45,24]

1. Print all the elements that are above the 65
2. Display all the elements whose percentile is more than 75
3. Give and increment of 10 to each element of this series
4. Find out the sum of all the element of this series
5. Display all the element which is divisible by 3
6. Create a Data Frame quarterly sales where each row contains the item category, item name, and expenditure. Group the rows by the category and print the total expenditure per category.
7. Create a data frame for examination result and display row labels, column labels data types of

each column and the dimensions

1. Filter out rows based on different criteria such as duplicate rows.
2. create a data frame from the following data of student and then perform the following operation

Student =[

{‘admno’:1102,’name’:’nikunj’ ‘class’:’12A’,’stream’:’sci’,’fees’:1234.45},

{‘admno’:1103,’name’:’mannat’ ‘class’:’11B’,’stream’:’comm’,’fees’:1834.45},

{‘admno’:1104,’name’:’unnati’ ‘class’:’12C’,’stream’:’Human’,’fees’:2334.45},

{‘admno’:1105,’name’:’Arushi’ ‘class’:’12A’,’stream’:’sci’,’fees’: 2344.45},

{‘admno’:1106,’name’:’pushkar’ ‘class’:’12A’,’stream’:’human’,’fees’:1500.25},

{‘admno’:1107,’name’:’nikunj’ ‘class’:’12A’,’stream’:’sci’,’fees’:1874.45},

{‘admno’:1108,’name’:’vishank’ ‘class’:’11A’,’stream’:’comm’,’fees’:1200.45},

]

1. Add a new column ‘grade’ where the grade of all the student is ‘A’
2. Add a new row in the above dataframe with some suitable values
3. Rename the column name ‘class’ into ‘standard’
4. Display only those rows where the fees of student is more than 1500
5. Give an increment of 500 in fees to all records
6. Delete all those records where stream=’Human’
7. Display all the column name of the above dataframe
8. Find out total number of student in each stream
9. Find out total fees of each stream
10. Find out class wise strength

6. create a data frame from the following data of student and then perform the following operation

Student =[

{‘admno’:1102,’name’:’nikunj’ ‘class’:’12A’,’stream’:’sci’,’fees’:1234.45},

{‘admno’:1103,’name’:’mannat’ ‘class’:’11B’,’stream’:’comm’,’fees’:1834.45},

{‘admno’:1104,’name’:’unnati’ ‘class’:’12C’,’stream’:’Human’,’fees’:2334.45},

{‘admno’:1105,’name’:’Arushi’ ‘class’:’12A’,’stream’:’sci’,’fees’: 2344.45},

{‘admno’:1106,’name’:’pushkar’ ‘class’:’12A’,’stream’:’human’,’fees’:1500.25},

{‘admno’:1107,’name’:’nikunj’ ‘class’:’12A’,’stream’:’sci’,’fees’:1874.45},

{‘admno’:1108,’name’:’vishank’ ‘class’:’11A’,’stream’:’comm’,’fees’:1200.45},

]

exporting data between into CSV file named “student.csv” and into excel file “student.xlsx”

7. import data from a CSV file ‘passport.csv’ into a dataframe and then display all the records

**Data Visualization Using MatPlotLib**

1. Given the school result data, analyses the performance of the students on different parameters, e.g subject

wise or class wise.

2. For the Data frames created above, analyze, and plot appropriate charts with title and legend.

3. Take data of your interest from an open source (e.g. data.gov.in), aggregate and summarize it. Then plot it

using different plotting functions of the Matplotlib library.

**Data Management Using MySQL**

1. Create a Database ‘DAVSCHOOL’
2. Create a student table with the following attributes where the student id is the primary key.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S.No.** | **Field Name** | **Data Type** | **Size** | **Attribute** |
| 1 | ID | Char | 10 | Primary Key |
| 2 | Name | Char | 30 | NOT NULL |
| 3 | Stream | Char | 20 |  |
| 4 | Class | Char | 15 |  |
| 5 | Marks | Float | 6.2 |  |
| 6 | Gender | Char | 1 |  |
| 7 | Grade | Char | 2 |  |

1. Insert the details of a 10 student in the above table using INSERT Command. Sample records of students are given below

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Id | Name | Stream | Class | Marks | Gender | Grade | gameID |
| 101 | Rakesh | Science | 12A | 98.45 | M | A | 1 |
| 102 | Pushkar | Human | 12C | 87.67 | M | B | 2 |
| 102 | Arushi | Human | 12A | 99.67 | F | A | 2 |
| 103 | Nikunj | Comm | 11B | 78.45 | M | C | 4 |
| 104 | Utkarsh | Science | 11A | 89.34 | M | B | 2 |
| 105 | Zareena | Comm | 11C | 67.56 | F | C | 4 |
| 106 | Khushboo | Human | 12B | 78.23 | F | B | 3 |
| 107 | Sehaj | Science | 12A | 99.89 | F | A | 2 |
| 108 | Karan | Human | 11B | 34.56 | M | C | 1 |
| 109 | Varun | Human | 11C | 56.56 | M | C | 3 |
| 110 | Ritik roshan | Comm. | 12B | 68.67 | M | B | 1 |

1. Add following column in the above table STUDENT with the following specification

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| s.no | Field name | Data type | Size | Constraint |
| 1 | Email | Char | 60 | Not null |
| 2 | Website | Char | 60 |  |

1. Display all the records of student table
2. Display student id, name and stream of the students from the above table of student
3. Create a report of student fees containing id,name,fees and anuual fees from the above table student ( assume fees displayed in the above table is monthly fees )
4. Display student name, stream and gameID of all the students

**Where clause**

1. Use the select command to get the details of the students with marks more than 80.
2. Display the records of all those female students
3. Display all the records of all those students whose marks is more than 80.00 and they are from humanities stream
4. Display the records of humanities and commerce stream students
5. Display only the records of all those students who are not from science stream

**Like clause**

1. Display the records of all those students whose name start with alphabet ‘R’
2. Display name and stream of all those female student whose name does not end with ‘na’.
3. Display id,name, stream of all students whose name contains ‘ar’ and whose grade is ‘A’

**Between clause**

1. Display the records of all those students whose fees is between 1200 and 1500
2. Display the records of all those male students whose fees is between 1500 and 2000

**Order by clause**

1. Display all the records of student table in ascending order ( according to student name )
2. Display the names of all the female student in descending order according to their marks.
3. Display id, name, stream , fees of all the male student in ascending order ( order by fees )

**Group By clause**

1. Display stream wise student strength.
2. Find out stream wise , total fees, minimum fees, maximum fees, and average fees
3. Display stream wise strength of student table where number of student in each stream is more than 2

**Update/delete Command**

1. Increase the fees 10% of all those students whose marks is less than 50.00
2. Remove all the records of those students whose grade is ‘C’
3. Create a table **‘Games’** with the following specification

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| s.no | Field name | Data type | Size | Specification |
| 1 | Gameid | Number | 5 |  |
| 2 | Coach name | Char | 30 |  |
| 3 | Gamename | Char | 30 |  |
| 4 | Fees | Float | 7,2 |  |

1. Insert the following records in the above table

|  |  |  |  |
| --- | --- | --- | --- |
| gameID | Coach Name | GameName | Fees |
| 1 | Ravi verma | Cricket | 2500 |
| 2 | Deepak Singh | Sketing | 1200 |
| 3 | Vikas Sharma | Boxing | 1800 |
| 4 | Sakshi juneja | Kho-kho | 1200 |

**Results from multiple table**

1. Display student name, game name, coach name and fees of the game from the above tables (student and games )
2. Display student name, coach name and fees from the above tables where game fees is more than 1200
3. Find out game wise student’s strength.