

Assignment #1

1 Explain how to dump specific table in to file and dump only table structure providing example.

-> Sqlite dump command,

- In spite dump command is used to dump the data base, table's schema and tables data based on our requirements.

- By using sqlite dump command we can save the structure and data of the tables format to the disk.

-> Syntax of sqlite dump command :

- Following is the syntax
↳ `sqlite > . dump.`

-> sqlite dump only table schema (structure) :

- To dump only schema of specific table sqlite offers "schema" command and we can redirect out put of his command to external command.

ex :- open product master.db

- table

- output product schema.sql

- schema products

- quit

Q.2 How to import csv File in to a table and export csv File in to a table ? give appropriate example.

→ import a csv File in to a table :

1. In the first scenario, you want to import data from csv File in to a table does not exist in the sqlite database.

↳ First, the sqlite 3 tool creates the sqlite 3 tool uses the first row of the csv File as the name of the table.

↳ second, import data from the second row of the csv File the table.

↳ we will import a csv File named city, csv.

↳ - mode csv

- import 6 ; | sqlite | city . csv cities

↳ schema cities

CREATE TABLE cities (

"name" TEXT,

"population" TEXT

);

L> view the data of the cities table

```
SELECT NAME population FROM cities
```

(2) table is already available in the database and you just need to import the data.

L> Following commands import the city - with out header

- CSV File in to the cities table.

- mode csv

- import c; | sqlite | city - no - header - csv cities.

→ Export a CSV File From table

L> Turn on header

L> set output mode to csv

L> send the output to a CSV File

L> write a table query for Export table

L> ex - header on

- mode csv

- output Employee - csv

- select - From cities

- quit.

Assignment :- 2

Q.1 How to set pythonpath? explain concepts of namespace, scope and packages in python.

Set pythonpath

write a command

set pythonpath = c:\python20\lib;

→ concepts of namespace :

what is namespace

- A namespace is a system that has a unique name for each and every object in python. An object might be a variable, method.

→ types of namespaces :

- when python interpreter runs `soles -l` without any user-defined modules, methods, classes, and some functions like `print()`, `id()` are always present. These are built in namespace. When a user creates a module, a global gets created after the creation. Of local functions creates the local namespace.

-> The lifetime of namespace :

- A lifetime of a namespace depends upon the scope of objects. If the scope of an object ends the lifetime of that namespace comes to an end. Hence it is not possible to access the namespace.

-> scope :

- Although there are various unique namespaces defined, we may not be able to access all of them every path of the comes in to play.

-> three nested scopes :

1. scope of the current function which has local names.
2. scope of the module which has global names.
3. outermost scope which has built in names.

-> packages in python :

- we organize a large number of files in different folders and subfolders based on some criteria so that we can find and manage them easily in the same way a package in python takes the concept of the modular approach to next logical level. As you know a module Functions.

→ Extracting Excel File :

- To extract data from an Excel file using pandas dataframe you can use the read.

Snippet :

```
import pandas as pd
```

```
df = pd.read_excel (filename e.xisr)
```

→ Writing Excel File :

- To write data to an Excel file using pandas dataframe, you can use the to_excel function.


```
ex: greet.py
```

```
def Say Hello (name) :  
    print ("Hello ", name)
```

Q.2 Explain extract and write commands for csv and excel files using Dataframe

→ Extracting csv File

- To extract data from csv file using pandas Dataframe you can use the read_csv function.

Syntax

```
- import pandas as pd
```

```
df = pd.read_csv (Filename.csv).
```

→ Writing csv File

- To write data to a csv file using pandas Dataframe you can use the to_csv() function

Syntax

```
import pandas as pd
```

```
df.to_csv ('Filename.csv', index  
           False)
```


Assignment 2-3

Q.2 Explain dataframe Function with example head, tail, loc, iloc, value_counts, mumpy().

1. Head()

- The head() Function return the First Few rows of a DataFrame. By default it returns the first 5 rows.

- ex : import pandas as pd

```
data = {'Name': ['John', 'Anna', 'Peter', 'Linda',  
                'Tom', 'Jerry'], 'Age': [28, 24, 35, 32,  
                40, 38]}
```

```
df = pd.DataFrame(data)
```

```
print(df.head())
```

2. tail()

- The tail() function returns the last Few rows of a DataFrame. By default it returns the last 5 rows

- ex : import pandas as pd

```
data = {'Name': ['John', 'Anna', 'Peter', 'Linda',  
                'Tom', 'Jerry'], 'Age': [28, 24, 35,  
                32, 40, 38]}
```

```
df = pd.DataFrame(data)
```

Marks:


```
print(df, tail())
```

3. ix[]

- The ix[] Function is used to access a group of rows and columns by a label or a boolean array

- import pandas as pd

```
data = {'Name': ['John', 'Anna', 'Peter', 'Linda',  
               'Tom', 'Jerry'], 'Age': [28, 24, 35, 32,  
               40, 38]}
```

```
df = pd.DataFrame(data)
```

```
print(df.ix[df['Age'] > 30])
```

4. iloc[]

- The iloc[] Function is use to access a group of rows and columns by position.

- import pandas as pd

```
data = {'Name': ['John', 'Anna', 'Peter', 'Linda',  
               'Tom', 'Jerry'], 'Age': [28, 24, 35,  
               32, 40, 38]}
```

```
df = pd.DataFrame(data)
```

```
print(df.iloc[1,3])
```


5. values :

- The `values` attribute returns a numpy representation of the data frame.

- import pandas as pd

```
data = {'Name': ('John', 'Anna', 'Peter', 'Linda',  
               'Tom', 'Jerry'), 'Age': (28, 24, 35, 32, 40,  
               38)}
```

```
df = pd.DataFrame(data)
```

```
print(df.values)
```

Q2. What is central tendency measures?
Describe mean, median, mode, variance, deviation.

L> central tendency measures

- Central tendency measure are statistical tools used to describe the central or typical value of a dataset. These measures help to summarize the data and provide a sense of the data's distribution. There are three main measures of central tendency: mean, median, and mode.

1. Mean :

L> The mean also known as the arithmetic mean is the average value of a dataset. It is calculated by

by summing up all the values and
by the values

$$\text{Formula: } \mu = (\sum x) / n$$

2. median :

↳ The median is the middle value of a dataset when it is sorted in ascending or descending order if the data set has an even of values.

↳ ex :

sorted dataset = 70, 75, 80, 85, 90

median = 80

3. mode :

↳ The mode is the most frequently occurring value in a dataset. A dataset : A dataset can have multiple modes if there are multiple values that appear with the same frequency.

→ ex : dataset of colour = red, blue, green, red, blue

mode = blue

4. Variance

↳ The variance is a measure of how spread out the values in a dataset are from the mean. It is calculated by taking the average of the squared mean.

↳ Formula $\sigma^2 = \frac{\sum (x - \mu)^2}{n}$

5. Standard Deviation

↳ The standard deviation is the square root of the variance.

↳ Formula $\sigma = \sqrt{\sigma^2}$

Assignment 24

Q.1 what is difference between scatter line, histogram and bar charts?

1. Scatter chart :

↳ A scatter chart, also known as a scatter plot or x-y plot is used to visualize the relationship between two continuous variables. Each point on the chart represents a single observation with the x-axis representing the other variable.

↳ ex: Relationship between height and weight.

2. Line chart :

↳ A line chart is used to show the trend of a single variable over time or across categories. Each point on the chart represents a single observation and the points are connected to show the trend.

↳ ex: stock prices over time.

3. Histograms :

↳ A histogram is a type of bar chart that shows the distribution of a single continuous variable. The x-axis represents the variable.

↳ ex: Distribution of exam scores.

4. Bar charts :

↳ A bar chart is used to compare the values of a categorical variable across different groups. Each bar represents a single group and the height of the bar of the variable.

↳ ex: sales by region

Q.2 Explain select insert update delete using execute() method to interact with table

1. Create table :

↳ import sqlite3

```
conn = sqlite3.connect('database.db')
cur = conn.cursor()
```

```
cur.execute("""create table species
studendid Integer primary key
intoIncedment, name text
Age Integer
marks Real""").
```

```
print('table created successfully')
conn.commit()
conn.close()
```

Marks:

2. Insert a record

L> The Insert statement is used to add new data to a sqlite table you can use the execute method to execute an Insert query add new data to the table.

```
L> ex = import sqlite3
      conn = sqlite3.connect('database.db')
      cur = conn.cursor()
      cur.execute("Insert Into species (genus, species)
                  values ('Citron', 'spiens')")
```

```
conn.commit()
conn.close()
```

3. The select statement is used to retrieve data from a sqlite table you can use the execute() method to execute a select query and data frame.

```
L> ex = import sqlite3
      import pandas as pd
```

```
conn = sqlite3.connect('database.db')
cur = conn.cursor()
cur.execute('select * from')
```

```
df = pd.DataFrame(cur.fetchall())
```



```
print(df, head())
```

```
conn.close()
```

4. update :

↳ The update statement is used to modify existing data in a SQLite table. you can use the execute method to execute an update query data in table.

↳ ex : import sqlite3

```
conn = sqlite3.connect(database.db')  
cur = conn.cursor()
```

```
cur.execute('''' update species genus)
```

```
conn.commit()
```

```
conn.close()
```

5. delete :

↳ The delete statement is used to delete data from a SQLite table. you can use the execute method delete query and delete data the table.

↳ ex : import sqlite3

```
conn = sqlite3.connect
```

```
cur = conn.cursor()
```


cin.execute ("Delete From species
where species = ")

conn.commit()

conn.close()