

DBMS - SQL - PYTHON

Project report submitted in partial fulfillment of the requirement for the degree of
Bachelor of Technology

in

Computer Science and Engineering

By

Vidit Jindal (181413)

Under the supervision of

Mr. Praveen Modi

Assistant Professor

Jaypee University of Information Technology, Solan

to



Department of Computer Science & Engineering and Information Technology

Jaypee University of Information Technology Waknaghat, Solan-173234,

Himachal Pradesh

Candidate's Declaration

I hereby declare that the work presented in this report entitled “ **DBMS - SQL - PYTHON**” in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science and Engineering** submitted in the department of Computer Science & Engineering and Information Technology, Jaypee University of Information Technology, Waknaghat is an authentic record of my own work carried out over a period from February 2022 to July 2022 (tentative) under the supervision of **Mr. Praveen Modi**, Computer Science and Engineering/Information Technology, Assistant Professor at Jaypee University of Information Technology, Solan.

The matter embodied in the report has not been submitted for the award of any other degree or diploma.

Vidit Jindal

181413

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

(Supervisor Signature)

Mr. Praveen Modi

Assistant Professor

Computer Science and Engineering & Information Technology

Dated: 28-05-2022

Esther Grace

Esther Grace

Programmer

Human Resource GenC Next

28-05-2022

ACKNOWLEDGEMENT

I would like to thank and express my gratitude to the project supervisor **Mr. Praveen Modi** for his constant support and guidance. This project would not have been possible without his help. I would always like to give a big thanks to my mentors in Cognizant who helped me in each and every step of this wonderful learning experience. This project taught me many new things and each concept was very interesting. I would also like to express my thanks to the lab assistant for contacting me and helping me in finishing the project within the stipulated time period.

Lastly , I would like to thank my friends and family for their support and love.

Vidit Jindal

181413

LIST OF ABBREVIATIONS

- DBMS Database Management System
- SQL Structured Query Language
- MySQL My Structured Query Language

LIST OF FIGURES

Figure Number - Name	Page No
Figure 1- Different types of database	9
Figure 2- SQL Commands	11
Figure 3- SQL Practice Session	11
Figure 4- SQL Command Output	12
Figure 5 - AWS Console	35
Figure 6 - Welcome to AWS	36
Figure 7 - EC2 Dashboard	36
Figure 8 - HTML and CSS Code	37
Figure 9 - Overall Performance Score	38
Figure 10 - Assessment Scores	38
Figure 11 - Hands on Progress	39

LIST OF TABLES

Table Number - Name	Page No
Table 1- Strings	19
Table 2- Series Object	29
Table 3- Series Object datatypes	30
Table 4- Dictionary Series Object	30
Table 5 - Methods on Series Object	31

TABLE OF CONTENT

1) Chapter 1 - Introduction

1.1 Introduction to Database	9
1.2 Introduction to SQL	9
1.3 Introduction to Python	12
1.4 History of Python	13
1.5 Python Features	13

2) Chapter 2 - Literature Survey

2.1 DBMS	15
2.2 SQL	15
2.3 Python	16
2.4 AWS	16

3) Chapter 3 - System Development and Courses Enrolled

3.1 Python for Beginners : Lean Python programming(Python).....	17
3.2 Python for Programmers.....	23
3.3 Python Enablement : Data Analysis with Pandas and Python.....	25
3.4 AWS Console.....	33
3.4.1 Jumpbox	33
3.4.2 Azure	34
3.4.3 Amazon Web Services	34
3.4.4 Freshdesk	35
3.5 HTML and CSS	37

4) Chapter 4 - Performance Analysis
4.1 RAG38

5) Chapter 5 - Conclusion
5.1 Conclusion40

References

PROBLEM STATEMENT

- It is very difficult to manage the data of large stores, companies, organisations, schools, colleges etc.
- For this purpose Database Management System (DBMS) is the key to this problem. It helps us to maintain a systematic, orderly flow of data in order to keep our data safe and secure .

It was during the year 1970 that the relational database model was developed by Edgar Codd. Many of the database models we use today are relational based. It was considered the standardized database model from then.

The relational model was still in use by many people in the market. Later during the same decade (1980's), IBM developed the Structured Query Language (SQL) as a part of R project. It was declared as a standard language for the queries by ISO and ANSI. The Transaction Management Systems for processing transactions was also developed by James Gray for which he was felicitated the Turing Award.

OBJECTIVE

- To provide help to the organizations by customizing the product for managing the employees.
- Keeping a track of all the access provided to each employee or customer and observing their activities.

CHAPTER - 1

INTRODUCTION

1.1 INTRODUCTION TO DATABASE

Database : Database is a collection of inter-related data which helps in efficient retrieval, insertion and deletion of data from database and organizes the data in the form of tables, views, schemas, reports etc. For Example, university database organizes the data about students, faculty, and admin staff etc. which helps in efficient retrieval, insertion and deletion of data from it.



Fig 1 : Different types of database

Some of the very famous different types of database are MySQL, Oracle, IBM, MongoDB etc.

1.2 INTRODUCTION TO SQL

DDL : DDL is short name of Data Definition Language, which deals with database schemas and descriptions, of how the data should reside in the database.

CREATE: to create a database and its objects like (table, index, views, store procedure, function, and triggers)

ALTER: alters the structure of the existing database

DROP: delete objects from the database

TRUNCATE: remove all records from a table, including all spaces allocated for the records are removed

COMMENT: add comments to the data dictionary

RENAME: rename an object

DML : DML is short name of Data Manipulation Language which deals with data manipulation and includes most common SQL statements such SELECT, INSERT, UPDATE, DELETE, etc., and it is used to store, modify, retrieve, delete and update data in a database.

SELECT: retrieve data from a database

INSERT: insert data into a table

UPDATE: updates existing data within a table

DELETE: Delete all records from a database table

MERGE: UPSERT operation (insert or update)

CALL: call a PL/SQL or Java subprogram

EXPLAIN PLAN: interpretation of the data access path

LOCK TABLE: concurrency Control

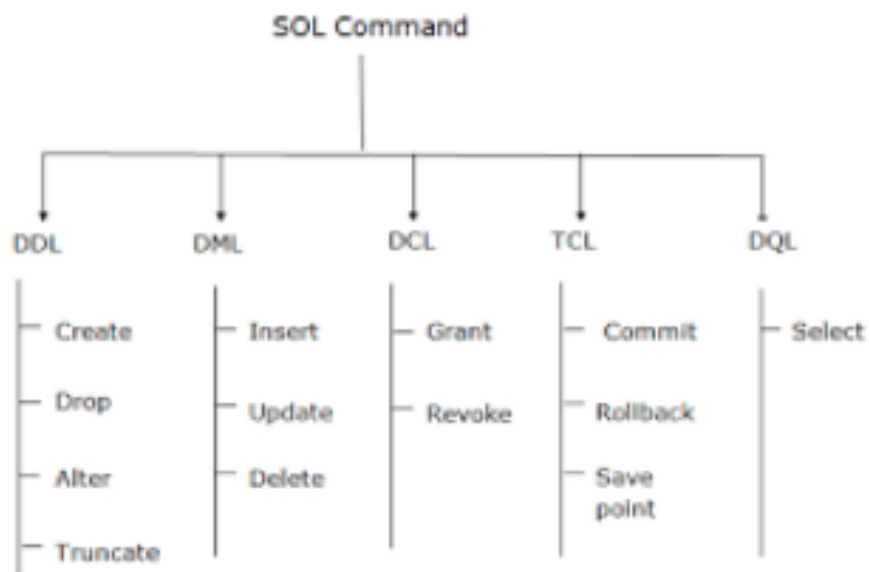


Fig 2 : SQL Commands

Some of the screenshots from our practice sessions are as follows :

```
1 CREATE TABLE shop (  
2     article INT UNSIGNED DEFAULT '0000' NOT NULL,  
3     dealer CHAR(20)      DEFAULT ''      NOT NULL,  
4     price  DECIMAL(16,2) DEFAULT '0.00' NOT NULL,  
5     PRIMARY KEY(article, dealer));  
6 INSERT INTO shop VALUES  
7     (1, 'A', 3.45), (1, 'B', 3.99), (2, 'A', 10.99), (3, 'B', 1.45),  
8     (3, 'C', 1.69), (3, 'D', 1.25), (4, 'D', 19.95);
```

Fig 3 : SQL practice session

```
1 SELECT * FROM shop ORDER BY article;  
2 +-----+-----+-----+  
3 | article | dealer | price |  
4 +-----+-----+-----+  
5 |        1 | A      | 3.45 |  
6 |        1 | B      | 3.99 |  
7 |        2 | A      | 10.99 |  
8 |        3 | B      | 1.45 |  
9 |        3 | C      | 1.69 |  
10 |        3 | D      | 1.25 |  
11 |        4 | D      | 19.95 |  
12 +-----+-----+-----+
```

Fig 4 : SQL Command output

1.3 INTRODUCTION TO PYTHON

Python is a widely used general-purpose, high level programming language. It was initially designed by Guido van Rossum in 1991 and developed by Python Software Foundation. It was mainly developed with emphasis on code readability, and its syntax allows programmers to express concepts in fewer lines of code. Python is a programming language that lets you work quickly and integrate systems more efficiently. Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

- **Python is Interpreted** – Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
- **Python is Interactive** – You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
- **Python is Object-Oriented** – Python supports Object-Oriented style or technique of programming that encapsulates code within objects.
- **Python is a Beginner's Language** – Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

1.4 HISTORY OF PYTHON

Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands. Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, SmallTalk, and Unix shell and

other scripting languages. Python is copyrighted. Like Perl, Python source code is now available under the GNU General Public License (GPL). Python is now maintained by a core development team at the institute, although Guido van Rossum still holds a vital role in directing its progress.

1.5 PYTHON FEATURES

Python's features include –

- **Easy-to-learn** – Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.
- **Easy-to-read** – Python code is more clearly defined and visible to the eyes.
- **Easy-to-maintain** – Python's source code is fairly easy-to-maintain.
- **A broad standard library** – Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.
- **Interactive Mode** – Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
- **Portable** – Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
- **Extendable** – You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
- **Databases** – Python provides interfaces to all major commercial databases.
- **GUI Programming** – Python supports GUI applications that can be created and ported to many system calls, libraries and window systems, such as Windows MFC, Macintosh, and the X Window system of Unix.
- **Scalable** – Python provides a better structure and support for large programs than shell scripting. Apart from the above-mentioned features, Python has a big list of good features, few are listed below –

- It supports functional and structured programming methods as well as OOP.
- It can be used as a scripting language or can be compiled to byte-code for building large applications.
- It provides very high-level dynamic data types and supports dynamic type checking.
- IT supports automatic garbage collection.
- It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java

CHAPTER 2

LITERATURE SURVEY

Database Management System (DBMS): The software which is used to manage database is called Database Management System (DBMS). For Example, MySQL, Oracle etc. are popular commercial DBMS used in different applications. DBMS allows users the following tasks:

Data Definition: It helps in creation, modification and removal of definitions that define the organization of data in database.

Data Updation: It helps in insertion, modification and deletion of the actual data in the database.

Data Retrieval: It helps in retrieval of data from the database which can be used by applications for various purposes.

User Administration: It helps in registering and monitoring users, enforcing data security, monitoring performance, maintaining data integrity, dealing with concurrency control and recovering information corrupted by unexpected failure.

A database management system (DBMS) allows a person to organize, store, and retrieve data from a computer. It is a way of communicating with a computer's "stored memory." In the very early years of computers, "punch cards" were used for input, output, and data storage. Punch cards offered a fast way to enter data and retrieve it. Herman Hollerith is given credit for adapting the punch cards used for weaving looms to act as the memory for a mechanical tabulating machine, in 1890. Much later, databases came along.

Databases (or DBs) have played a very important part in the recent evolution of computers. The first computer programs were developed in the early 1950s and focused almost completely on coding languages and algorithms. At the time, computers were basically giant calculators and data (names, phone numbers) was considered the leftovers of processing information. Computers were just starting to become commercially available, and when business people started using them for real-world purposes, this leftover data suddenly became important.

CHAPTER 3

SYSTEM DEVELOPMENT

AND

COURSES ENROLLED

3.1 Python for Beginners : Learn Python Programming (Python 3)

Course Overview

We'll learn

- How to install Python and prepare our computer for writing python programs - Python Installation and configuration
- Python 2 v/s Python 3
- Variables and Strings
- Functions and Printing
- Numbers and Math
- Performing mathematical calculations with python
- Booleans and conditionals
- How to perform simple tests and make decisions in our program
- Lists
- How to loop through a series of items and perform the same action on each one
- Dictionaries
- Tuples
- How to store read only data
- Files
- How to store and retrieve data from files

- Modules and the standard python library

INSTALLING PYTHON ON MAC

- Go to python.org and choose the section of downloads .
- Go to latest python version available
- Click on Install
- If IDLE is available now o your local desktop then Python has been installed .

PYTHON 2 V/S PYTHON3

Python 3 was released in 2008. Python 2.7 fills the gap between Python 2 and Python 3 . Most of the code for Python 3 will work on Python 2.7. Use latest version of python available. Python 2.7 is compatible with all Python 2 code. Primary reason to choose Python 2 over Python 3 is that if project requires some third party software that is not compatible or just doesn't exist for Python 3.

VARIABLES AND STRINGS

Variables are storage locations that have a name or variables are name value pairs. We can assign values to a variable and recall those values by the variable name. To assign a value to a variable we use = sign.

Syntax - Variable_name = value

example - value of apple is assigned to a variable

```
fruit = "apple"
```

Variable names are case sensitive. Fruit and fruit are two different variables.

A string is used to represent text. Ex: apple, orange, ABC. Strings are surrounded by “ ” or ‘ ’ in Python. Each character in string is assigned an index. To access the character at a given index,

append an open bracket, a number and a closed bracket to a string where the number is the index number.

Example :

String	Index
a	0
p	1
p	2
l	3
e	4

Table 1 : Strings

FUNCTIONS AND PRINTING

A function is a section of reusable code that performs an action. A function has a name and is called or executed by that name. Functions can accept arguments and return data.

There are many built-in-functions in Python.

Example :

1. print() function

When a value is provided as an argument to the print function, it displays that value to the screen. We can supply values directly to the print statement or pass in the variables.

Ex: fruit = "apple"

```
print(fruit)
= apple
```

```
Ex: print("orange")
    = orange
```

2. len() function

It returns the length of the string. It returns the number of characters in a string.

```
Ex: fruit_len = len(fruit)
    print(fruit_len)
    = 5
```

```
Ex: fruit="apple"
    print(len(fruit))
    = 5
```

String Methods

Everything in Python is an object. Every object has a type. As "apple" is an object of type "str".

Objects also have functions but they are not called functions. They are called methods. So, methods are functions that runs against an object.

To call a method on an object, follow the object with a . then the method name and (). We can include parameters in the ().

1. **lower() string method** : It returns a copy of the string and all lowercase letters.

```
Ex: fruit = "Apple"
    print(fruit.lower())
    = apple
```

2. **upper() string method** : It returns a copy of the string and all uppercase letters.

```
Ex: fruit = "Apple"
    print(fruit.ipper())
    = APPLE
```

String Concatenation : To concatenate or combine 2 or more strings, we use the plus sign. It looks like adding 2 strings together. We concatenate multiple strings by using '+' string . '+' sign does not add any new characters.

```
Ex: print( 'I' + ' love' + ' Python' )  
= I love Python
```

To concatenate a string with a number, first convert the no. to a string with built in string function. The str functions convert nonstrings into strings .

```
Ex: version = 3  
print ( " I love Python " + str(version) + ".")  
= I love Python 3.
```

Repeating strings using repeating operator :

* is the repetition operator.

Syntax - 'string' * no. of times string needs to be printed

```
Ex : print ('abc'*2)  
abcabc
```

Formatting Strings :

Instead of concatenating strings to produce the format, we can call the format method on a string created placeholders, also known as format fields by using curly braces in the string and passing values for those fields to the format method. By default, the first pair of curly braces will be replaced by the first value pass to the format and so on.

```
Ex: print( 'I {} python.'.format('love'))
```

= I love python.

{ } will be replaced by string that is passed into the format command. In next statement there are 3 placeholders. Each place holder is replaced by the corresponding string. When we pass multiple objects to a function or method we separate them using comma.

Formatting strings using variables :

Here we are not explicitly passing a string to the format command. First placeholder gets replaced by the first variable and so on. Now we can write strings and numbers with the format method.

```
Ex: first = 'I'  
    second = ' love'  
    third = ' python'  
    print(' { } { } { } '.format(first,second,third))  
    = I love python.
```

We can supply a format specification. Format specifications are contained within the curly braces and begin with a colon to create a field with a minimum character with supplier number following the colon the format field of open curly brace.

NUMBER AND MATH

We can include numbers directly in our source code. Python supports integers and floating point numbers . Floating numbers contains a decimal point whereas integers doesnot.

Some numeric operations that can be performed by python are:

- Addition
- Subtraction
- Multiplication
- Division
- Exponential
- Modulo

3.2 PYTHON FOR PROGRAMMERS

Dictionaries

It is a type which can hold values just like lists, tuples. They store key value pairs. In dictionaries, instead of indexing we use keys. Keys can be anything strings, numbers. So, it's basically taken from english dictionary, we find the word and corresponding to it is definition.

Syntax:

```
dictionary_name = {"Key1" : value1 , ..... , "key n" : value2}
```

To access a particular element in a dictionary

Syntax :

```
print(dictionary_name["key"])
```

In dictionaries, keys have to be unique, key's can't be duplicate. You can have duplicate values. If a key is repeated, it will show only 1 time and the one which was updated at last. Keys have to be Immutable. A list is not immutable.

To add a new item in dictionary

Syntax:

```
dictionary_name["key"] = value
```

To delete a item in dictionary

Syntax:

```
del(dictionary_name["key"])
```

Converting dictionary to list

```
variable_name = list(dict_name.keys())
```

CLASSES AND VARIABLES

Declaring a class

Syntax:

```
class class_name:
```

Declaring an object

Syntax:

```
object_name = class_name()
```

Whenever we make a class, we need to have atleast 1 line of code inside it.

Instance Variable

Instance Variable are those which are specific to some class

Init Method :

Init method can be used to setup something in our class.

A function inside a class is called method.

Defining a init method

Syntax:

```
def __init__(self):
```

Inheritance :

By this we can make a class based on existing class, but with some additional features.

Syntax :

```
class class_name(superclass_name):
```

To create an object of this class, we can do

Syntax:

```
object_name = classname()
```

Files Reading and Writing

Creating a file

Syntax:

```
open("file_name to be created" , access_mode)
```

Example:

```
open("fun.txt","w")
```

If there is no such file then it will create it and if there is such file then it will overwrite it.

Writing in a file

Example :

```
file = open("fun.txt","w")
```

```
file.write("Hello my name is Vidit.\n This is my new file!")
```

Reading in a file

```
file = open("fun.txt", "r")  
print(file.read())
```

The content of the file will be printed.

3.3 PYTHON ENABLEMENT : DATA ANALYSIS WITH PANDAS AND PYTHON

3.3.1 PANDAS

Pandas is a library built on the top of Python Programming language. A library is a bundle of features and functionalities. Pandas focus on data analysis. It's a tool that can be used for analysing, filtering, manipulating, aggregating, merging, pivoting and cleaning data.

Pandas is like excel for Python. It can work on data sets of millions of rows.

- To work with pandas, install python and pandas. We can't run pandas without python.
- Python itself is dependent on other libraries.
- Anaconda distribution bundles python, pandas and data analysis libraries.
- Use terminal to update libraries and install additional ones.
- A distribution is just a big bundle of Python pandas and bunch of different libraries.
- Ananconda distribution is by continuous analytics.
- Install - Update (via terminal)

Importing pandas library

```
import pandas
```

To shorten the name of pandas, we make an alias by giving it a shorter name

Example :

```
import pandas as pd
```

We need to import libraries, if we need to use their features.

Comments

A line that is ignored by python interpreter. used for leaving description, information etc.

To create a comment, we use # symbol.

Example:

```
2+2 # 1+1  
= 4
```

Lists

A list is a mutable data structure that holds an ordered collection of elements. So, we can have list of numbers, list of strings, list of booleans.

Syntax:

```
list_name = []
```

Example:

```
numbers = [1,2,3,4]
```

Declaring an empty list

Syntax:

```
[]
```

Length of the list

Syntax:

```
len()
```

Example:

```
len(numbers)
= 4
```

Adding element at the end of a list

Syntax:

```
list_name.append('value')
```

Example:

```
numbers.append(5)
```

Return value of this method is none object. It doesn't create a new list. It modifies the original list.

Deleting the last element of the list

Syntax:

```
list_name.pop()
```

Example:

```
numbers.pop()
= 5
```

Removing a particular value from the list

Syntax:

```
list_name.remove('value')
```

Example:

```
numbers.remove('1')
numbers
= [2,3,4]
```

If there are multiple numbers of the same value then it will remove the first occurrence.

In operator

In operator is used to check whether the object on Left hand side belongs in the object of right hand side.

Example:

```
“cat” in “catterpillar”  
= True
```

Not in operator

It is the converse of in operator.

Example:

```
“Pluto” not in “Plutonus”  
= False
```

Index Positions and Slicing

This concept applies to lists and strings. In python indexing starts from 0.

Example:

```
spiderman = “Spiderman”  
spiderman[0]  
= S
```

3.3.2 **SERIES**

Series combines best of both list and dictionary. It is able to store single collection of data in order with a unique identification of each element to refer to it.

Creating a series object from a python list

To create a series object, we need to provide a data source which will be used to populate the series value. This data source can be a list as well. Series is not a function.

Example:

```
ice_cream = ["Chocolate", "Vanilla", "Strawberry", "Rum Raisin"]  
pd.Series(ice_cream)
```

0	Chocolate
1	Vanilla
2	Strawberry
3	Rum Raisin

Table 2 : Series object

Example :

```
reg = [True, False, False]
```

0	True
1	False
2	False

```
pd.Series(reg)
```

dtype : bool

Table 3 : Series object datatypes

Creating a series object from a python dictionary

We're going to pass dictionary as a data source. Index of a pandas series need not to have standard numeric index. It can have but it's not compulsory.

Example:

```
sushi = {  
    "Salmon" : "Orange",  
    "Tuna" : "Red",  
    "Eel" : "Brown"  
}
```

```
pd.Series(sushi)
```

Salmon	Orange
Tuna	Red
Eel	Brown

dtype : object

Table 4 : Dictionary Series Object

Keys of the dictionary are treated as index of the series.

Applying methods on series object:

```
prices = pd.Series([2.99, 4.45, 1.36])
```

prices

0	2.99
1	4.45
2	1.36

Table 5 : Methods on Series Object

1. **sum()** : It takes all the values and add them together. It doesnot take any input.

Example:

```
prices.sum()
= 8.8
```

2. **product()** : It gives product of all values.

Example:

```
prices.product()
= 18.09548000000
```

3. **mean()** : It gives average .

Example:

```
prices.mean()
= 2.93333
```

These are those methods which require parameters.

Attributes : Attributes on Pandas series object. An attribute is a characterstick that an object can tell about itself. Method is an action or behaviour or command. Attribute is a property or information or detail. Methods require () at the end.

Some of the attributes are as follows :

Example:

```
adjectives = pd.Series(["small" , "handsome" , "charming" , "brilliant" , "humble" ])
adjectives

= 0
```

- 1
- 2
- 3
- 4

1. **size** : It gives the size of series object.

```
Example : adjectives.size  
= 5
```

2. **is_unique** : Whether or not these series elements are unique.

```
Example : adjectives.is_unique  
= True
```

All elements are unique

3. **values** : This provides an n-d array. it comes from numpy, this is dependency of panda i.e pandas need numpy to sum.

Example:

```
adjectives.values  
= array([ 'small' , 'handsome' , 'charming' , 'brilliant' , 'humble' , 'smart' ] , dtype = object )
```

4. **index** : Range index is an object . It is keeping track of index . There can be different types of indexes as well.

Example: adjectives.index

```
= RangeIndex(start = 0 , stop = 6 , step = 1)
```

3.4 AWS CONSOLE

It is a browser based GUI which can be used for controlling features provided by amazon web services. The users can use the drag/ drop for the service links required. Users can also view resources and applications that are sharing common tags. They can use tag editors to view and make quick changes to all the resources and applications sharing common tags. It supports almost all the operating systems.

3.4.1 Jumpbox

It's a network security technology that allows you to access and manage devices in a distinct security zone. It keeps track of the service that connects the two security zones and allows for controlled access between them.

3.4.2 Azure

Microsoft Azure is another name for it. It offers cloud computing services that are administered by Microsoft data centres. Software as a service (SaaS), platform as a service (PaaS), and infrastructure as a service (IaaS) are all available (IaaS). It is compatible with a wide range of programming languages, tools, and frameworks.

3.4.3 Amazon Web Services

They are a subsidiary of Amazon and provide various cloud computing services. Through AWS server farms, these cloud computing internet services provide dispensed computing processing power and software programme equipment. One of these services is Amazon Elastic Compute Cloud (EC2), which allows consumers to rent virtual machines.

They have at their disposal a digital cluster of computer systems, to be had all of the time, thru the Internet. AWS's digital computer systems emulate maximum of the attributes of an actual computer, along with hardware significant processing units (CPUs) and photographs processing units (GPUs) for processing; local/RAM memory.

AWS offerings are added to clients through a community of AWS server farms positioned throughout the world. Fees are primarily based totally on a mixture of utilization (referred to as a "Pay-as-you-go" model), hardware, running system, software program, or networking functions selected with the aid of using the subscriber required availability, redundancy, safety, and provider options. Subscribers pay for a digital AWS computer, a devoted bodily computer, or clusters of either. Amazon presents pick out quantities of safety for subscribers (e.g. bodily safety of the statistics centers) even as different factors of safety are the duty of the subscriber (e.g. account management, vulnerability scanning, patching). AWS operates from many worldwide geographical areas along with 6 in North America.

Amazon markets AWS to subscribers as a manner of acquiring large-scale computing ability extra speedy and cost effectively than constructing a real bodily server farm. All offerings are billed primarily based totally on utilization, however every provider measures utilization in various ways. As of 2021 Q4, AWS has 33% marketplace proportion for cloud infrastructure even as the following competition Microsoft Azure and Google Cloud have 21%, and 10% respectively, in step with Synergy Group.

3.4.4 Freshdesk

It is an American cloud based customer engagement company.

Features:

1. Support channels:

- It has different channels like chat, email, phone, twitter and facebook etc.

2. Productivity hacks:

- Tags: It helps in categorizing the tickets.
- Dispatch: Rules can be created for tickets and workflows automation can be done for support.
- Automatic email notification: Agents and customers are sent an email when changes are made.
- Canned responses: Templates can be created and saved for reusing while replying to a customer.
- Customizable help desk:

3. Helpdesk management:

- Notes: Public notes can be added for informing customers and private notes can be added to inform fellow team members.
- Ticket Activities: Changes made in the ticket can be viewed in the history for that day.
- Team Inbox: Shared inbox is present for the team members.
- Merge Tickets: Tickets from different channels can be merged in chronological order.

- To-dos: Add a task in the ticket or in the dashboard. Prioritize your work.
- Freshconnect collaboration: Connect with other team members within the freshdesk.

4. Self- Service:

- Knowledge Base: Knowledge pages can be created to share important information with customers.
- Email to knowledge base:

5. Reporting:

- Default Dashboard: We can view trends in the tickets, recent activities, forums etc.
- Freshdesk analytics(beta): View performances of agent ,ticket lifecycle, group performance etc.

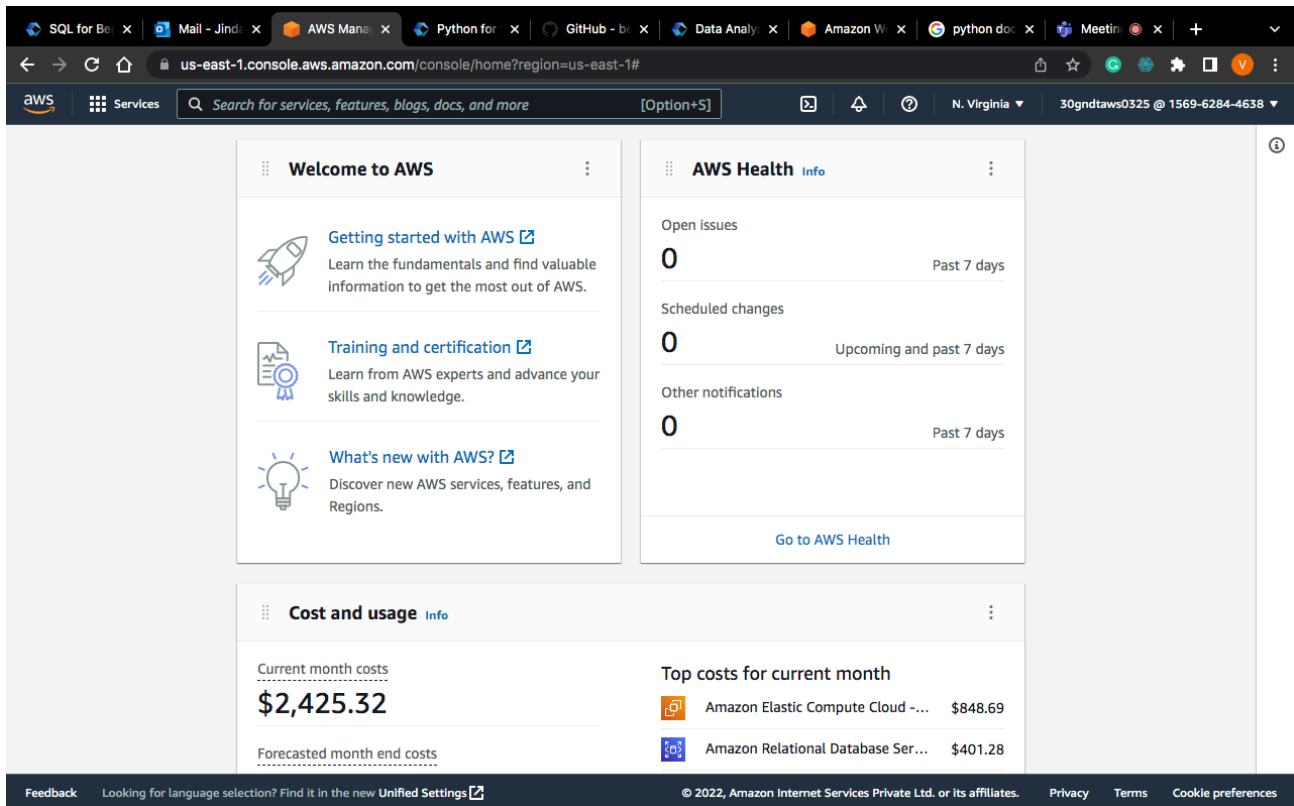


Figure 5 : AWS Console

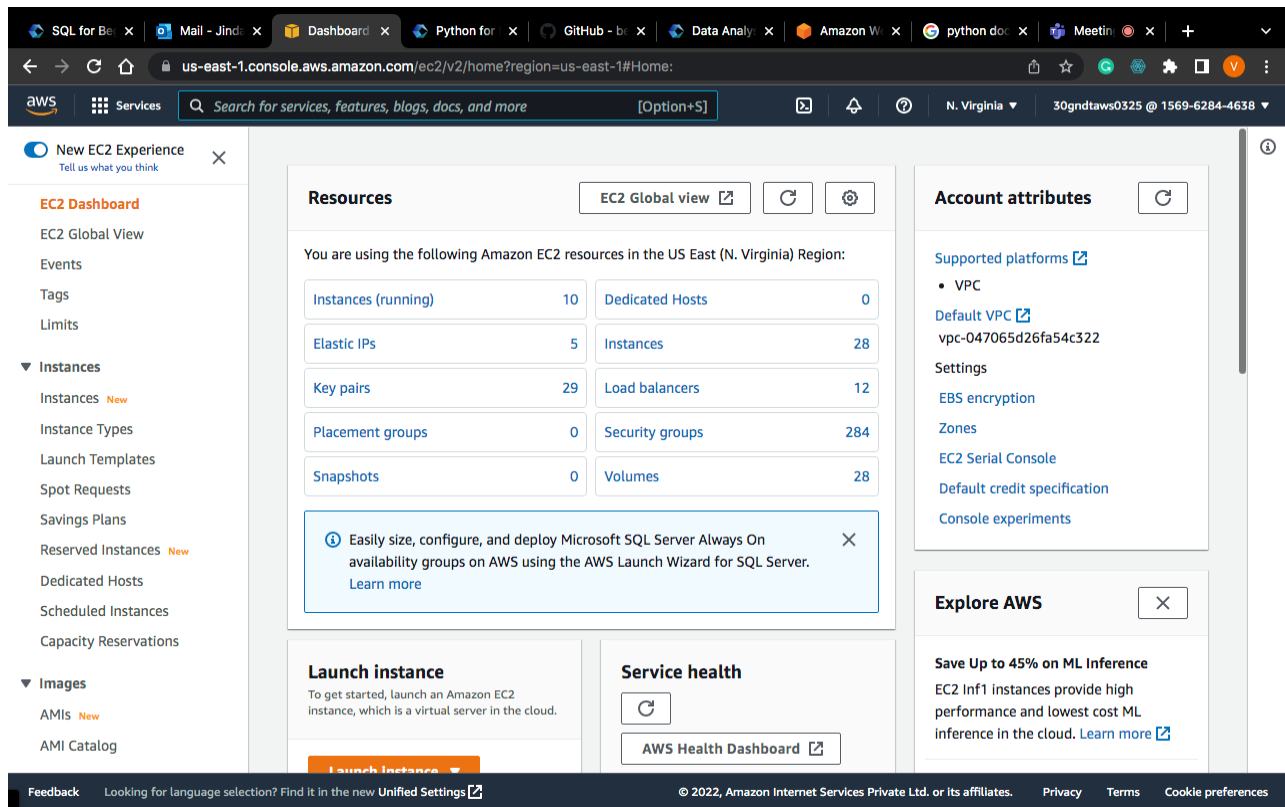


Figure 6 : Welcome to AWS

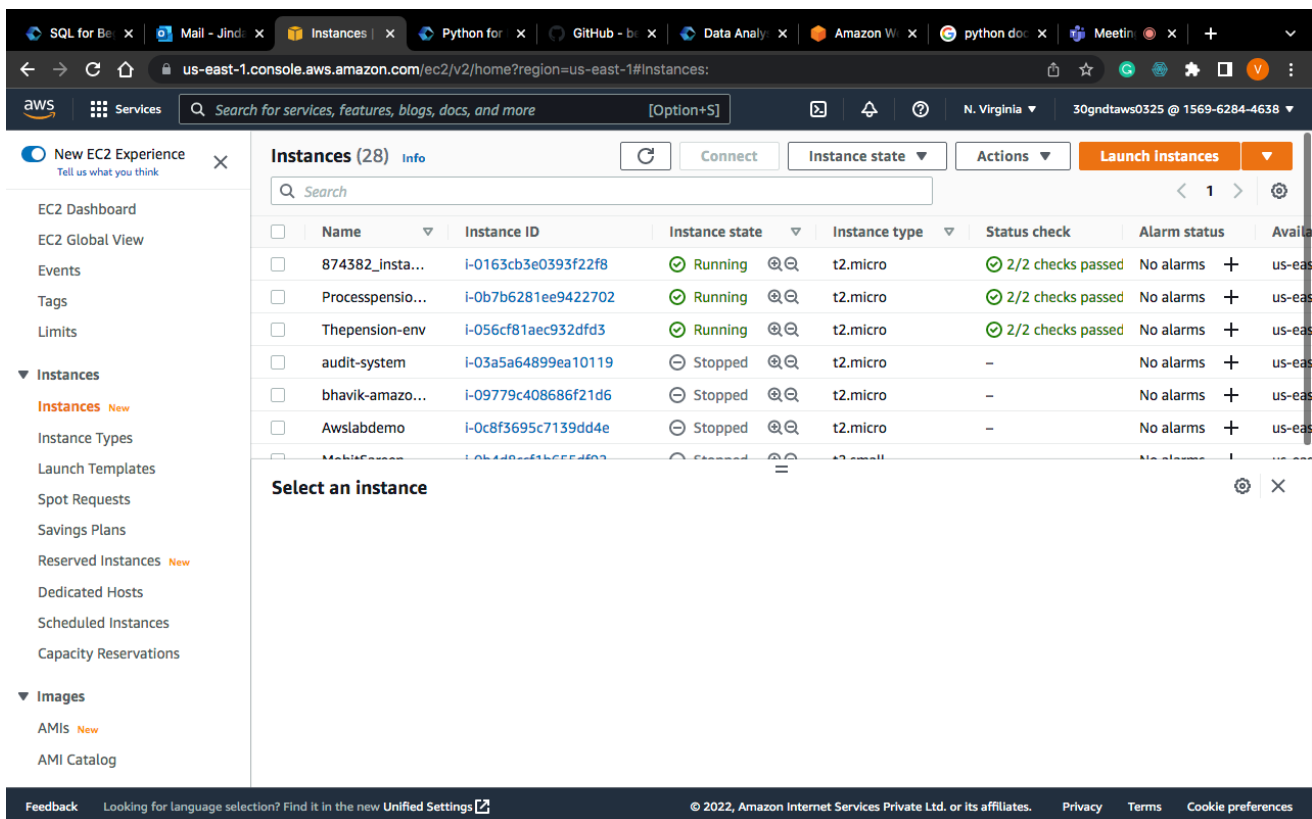


Figure 7 : EC2 Dashboard

3.5 HTML and CSS

We completed Udemy courses provided by the Cognizant during the internship, completed the hands-ons, assessment(important), and completed the integrated capability test also.

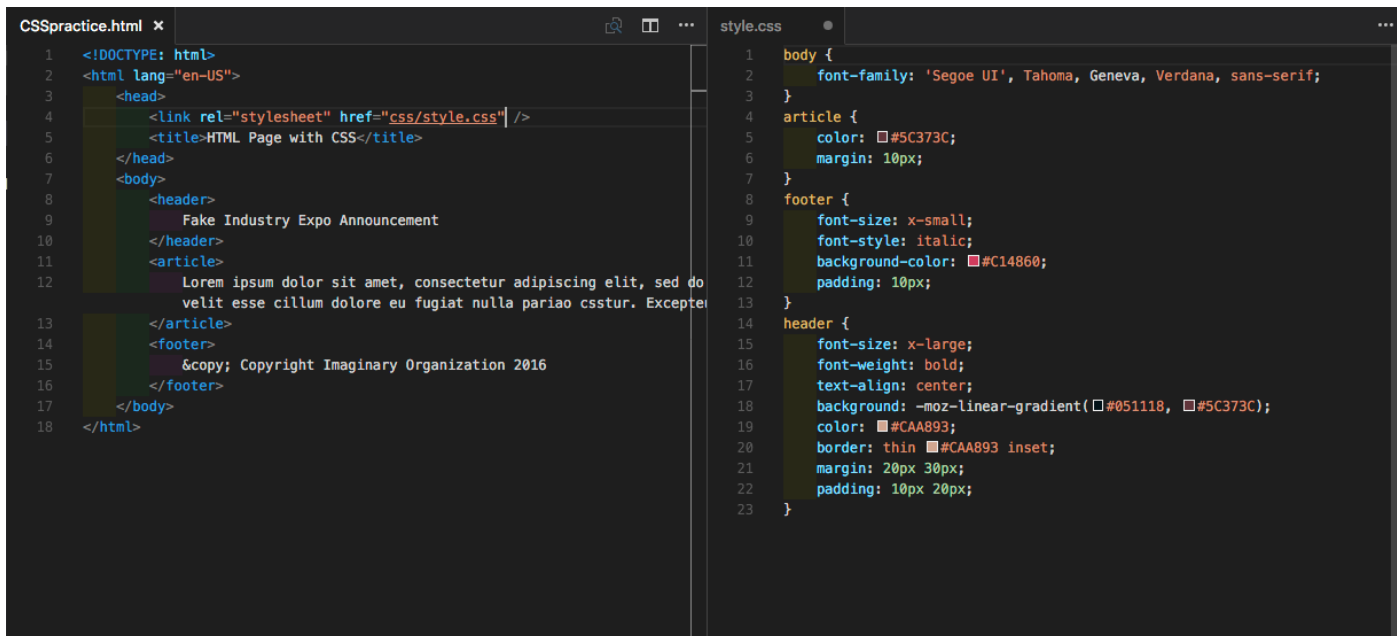
We learned about the designing part from scratch with the help of the HTML5 and CSS and also JavaScript.

After we completed the online Udemy courses, we did the hands-ons and completing the hands- on is mandatory for every interns and then, after completing the hands-ons we use to give assessment, a small test whose marks were taken into account, for the calculation of the in the final overall performance.

The most important part of this week was covering all the basis aspects of the designing and learning HTML and CSS and JavaScript from scratch because learning HTML and CSS is very important in designing.

It also covered integrating of HTML and CSS with the JavaScript to form proper webpage.

Below is the sample HTML and CSS, and JavaScript Code.



```

CSSpractice.html x
1 <!DOCTYPE: html>
2 <html lang="en-US">
3   <head>
4     <link rel="stylesheet" href="css/style.css" />
5     <title>HTML Page with CSS</title>
6   </head>
7   <body>
8     <header>
9       Fake Industry Expo Announcement
10    </header>
11    <article>
12      Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do
13      velit esse cillum dolore eu fugiat nulla pariao csstur. Excepte
14    </article>
15    <footer>
16      &copy; Copyright Imaginary Organization 2016
17    </footer>
18  </body>
19 </html>

style.css
1 body {
2   font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
3 }
4 article {
5   color: #5C373C;
6   margin: 10px;
7 }
8 footer {
9   font-size: x-small;
10  font-style: italic;
11  background-color: #C14860;
12  padding: 10px;
13 }
14 header {
15  font-size: x-large;
16  font-weight: bold;
17  text-align: center;
18  background: -moz-linear-gradient(#051118, #5C373C);
19  color: #CAA893;
20  border: thin #CAA893 inset;
21  margin: 20px 30px;
22  padding: 10px 20px;
23 }
```

Figure 8 : HTML and CSS Code

CHAPTER - 4
PERFORMANCE ANALYSIS

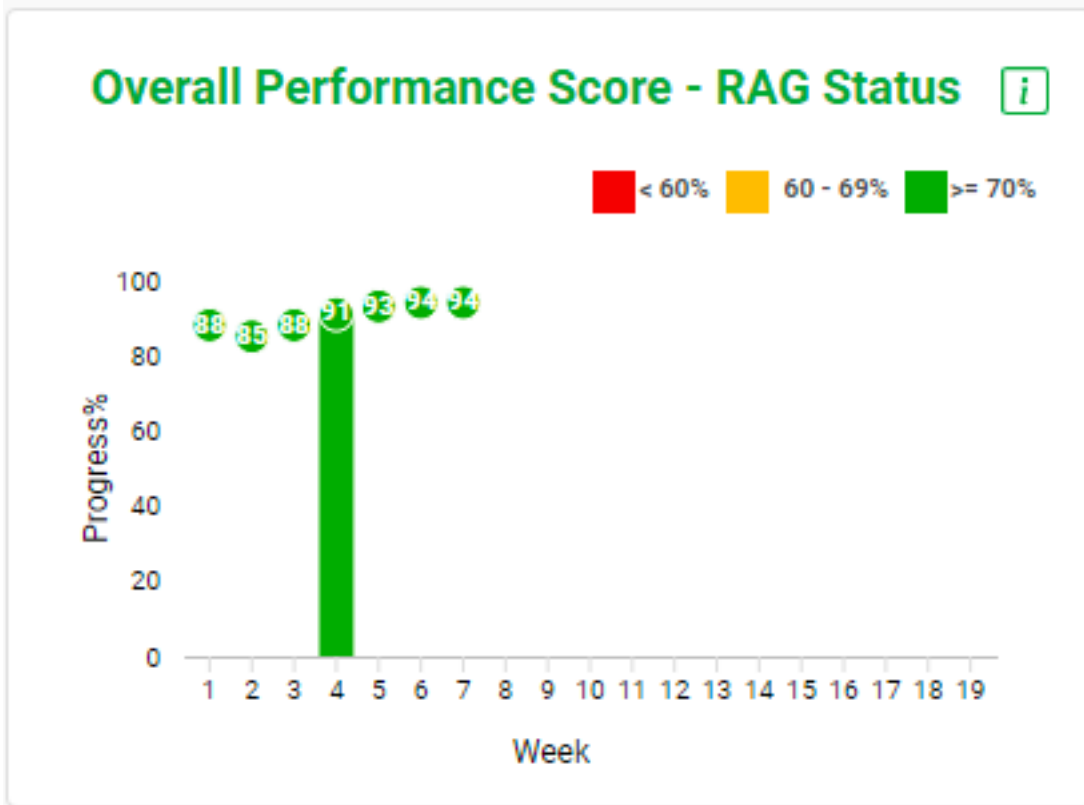


Figure 9 : Overall performance Score

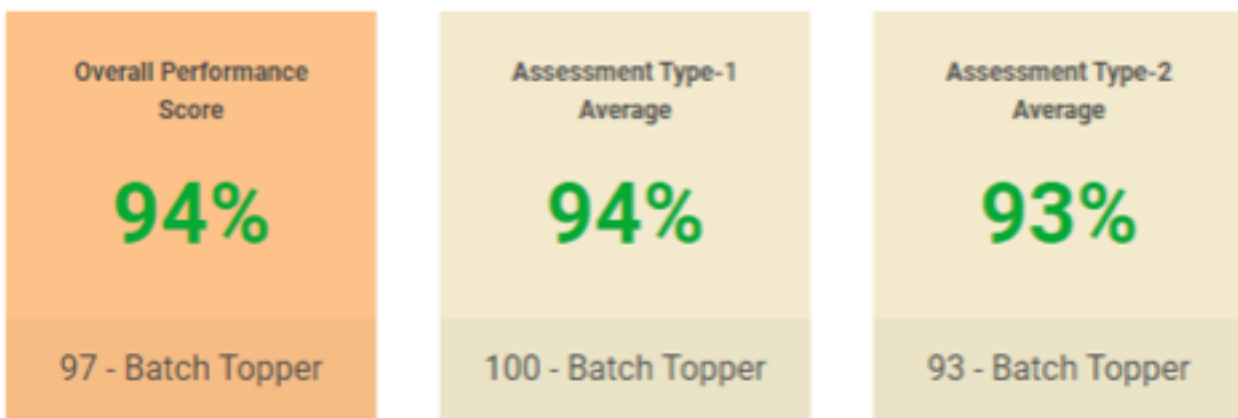


Figure 10 : Assessment scores

Practice progress



Hands-on Progress

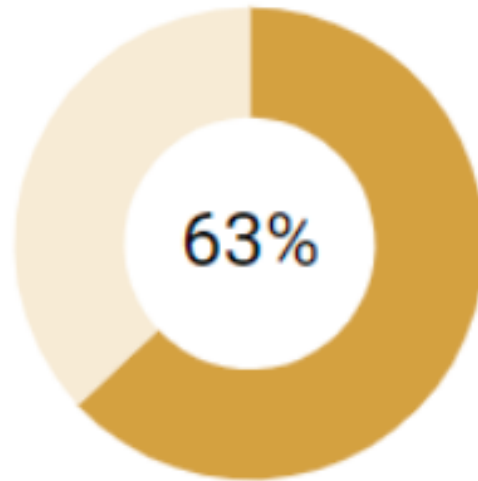


Figure 11 : Hands on Progress

CHAPTER 5

CONCLUSION

5.1 Conclusion

The past two months had been great for learning. The concepts I learnt played a major role in clearing my basics. They raise my curiosity and I wish to learn more. Python and DBMS are one of the main concerns in today's domain. Cognizant is on a mission to provide services through intelligent, cloud-first identity governance and by providing different solutions to a team. Cognizant was created to challenge the status quo. We always realized that identity could be so much more – and the leading solutions weren't up to the task.

It hasn't always been easy, but we're here to solve challenges, not hide from them. Today we've grown into a global organization scaling at breakneck speed to help the largest enterprises in the world transform their identity programs and protect their people, assets, and infrastructure.

REFERENCES

1. https://cognizantlearning.sumtotal.host/rcore/c/pillarRedirect?isDeepLink=1&relyingParty=LM&url=https%3A%2F%2FCOGNIZANTLEARNING.sumtotal.host%2Flearning%2Fapp%2Fmanagement%2FLMS_CNT_LaunchCourse.aspx%3FUserMode%3D0%26ActivityID%3D1077341%26RegId%3D102335692%26RootActID%3D1077341%26EventID%3D1077341%26AttCnt%3D1%26AttemptId%3D639488887
2. https://cognizantlearning.sumtotal.host/rcore/c/pillarRedirect?isDeepLink=1&relyingParty=LM&url=https%3A%2F%2FCOGNIZANTLEARNING.sumtotal.host%2Flearning%2Fapp%2Fmanagement%2FLMS_CNT_LaunchCourse.aspx%3FUserMode%3D0%26ActivityID%3D1126097%26RegId%3D101102081%26RootActID%3D1126097%26EventID%3D1126097%26AttCnt%3D1%26AttemptId%3D635298795
3. https://cognizantlearning.sumtotal.host/rcore/c/pillarRedirect?isDeepLink=1&relyingParty=LM&url=https%3A%2F%2FCOGNIZANTLEARNING.sumtotal.host%2Flearning%2Fapp%2Fmanagement%2FLMS_CNT_LaunchCourse.aspx%3FUserMode%3D0%26ActivityID%3D1126982%26RegId%3D101060228%26RootActID%3D1126982%26EventID%3D1126982%26AttCnt%3D1%26AttemptId%3D635177531
4. https://cognizantlearning.sumtotal.host/rcore/c/pillarRedirect?isDeepLink=1&relyingParty=LM&url=https%3A%2F%2FCOGNIZANTLEARNING.sumtotal.host%2Flearning%2Fapp%2Fmanagement%2FLMS_CNT_LaunchCourse.aspx%3FUserMode%3D0%26ActivityID%3D1126460%26RegId%3D100675604%26RootActID%3D1126460%26EventID

[%3D1126460%26AttCnt%3D1%26AttemptId%3D633834703](#)

5. <https://cognizant.udemy.com/course/data-analysis-with-pandas/learn/lecture/5931582#overview>

6. <https://cognizant.udemy.com/course/python-for-programmers/learn/quiz/268140#overview>