



Journey Presentation

Harsh Thakur

Harsh.Thakur@shell.com

DAY-5

Azure Storage Types- Day 1

1. Create A Resource -> Azure Storage Account
2. Resource Group
3. Storage Account
4. Storage Services
5. Redundancy
 1. Local Redundancy
 2. Zone Redundancy
 3. Geo Redundancy
 4. Geo Zone Redundancy
6. RAGRS (Read Access Geo Redundant Storage)
 1. Failover
 2. Synchronous Replication
7. Security
8. Access Protocols
9. Access Tiers
 1. Hot
 2. Cold
 3. Cool
 4. Archive Access
10. Purging & Archiving
11. Containers - Folder For Storing Files
12. Azure Tables
13. Azure Queues
14. File Shares
15. Snapshots
16. Ingress & Egress
17. Static Website

Resource group (Ingest): RG_Ansukha_04

Location: East US

Primary/Secondary Location: Primary: East US, Secondary: West US

Subscription (Ingest): anushkastorage_1693810230369

Subscription ID: 6f5ab03-77db-43fb-9cde-b251f74e513

Disk state: Available

Created: 9/4/2023, 11:26:34 AM

Resource group (Ingest): RG_Ansukha_04

Location: East US

Primary/Secondary Location: Primary: East US, Secondary: West US

Subscription (Ingest): anushkastorage_1693810230369

Subscription ID: 6f5ab03-77db-43fb-9cde-b251f74e513

Disk state: Available

Created: 9/4/2023, 12:05:53 PM

Name: anufs

Type: File

Size: 912 B

Deployment name: gen2anushkastorage_1693810230369

Start time: 9/4/2023, 12:04:11 PM

Subscription: anushkastorage_1693810230369

Resource group: RG_Ansukha_04

Deployment ID: c1ed509-4048-42fb-9105-7221b667587

Deployment details

- Go to resource
- Give feedback

Cost Management: Get notified to stay within your budget and prevent unexpected charges on your bill. Set up cost alerts.

Microsoft Defender for Cloud: Secure your apps and infrastructure. Go to Microsoft Defender for Cloud >

Free Microsoft Tutorials: Start learning today >

Work with an expert: Azure experts are service provider partners who can help manage your assets on Azure and be your first line of support. Find an Azure expert >

Name: samplefileshare

Tier: Transaction optimized

Modified: 9/4/2023, 2:03:57 PM

Quota: 5 TiB

Notepad content:

1. Set up Azure SQL services by logging in on Azure portal.
2. SQL Deployment options for Azure SQL:
 - a. SQL Database
 - b. SQL Managed Instances
 - c. SQL Virtual Machines
3. Select a database & Single database -> Create database.
4. Introduction to SQL
5. Database created on Microsoft Azure SQL.
6. Running Query Editor on our server.
7. DDL (Data Definition Language): CREATE, ALTER, DROP, TRUNCATE
8. DML (Data Manipulation Language): INSERT, UPDATE, DELETE
9. DCL (Data Control Language): GRANT, REVOKE
10. Common table expression (CTE)
11. Between operator between DELETE, DROP, TRUNCATE.
12. Activity : Create tables for online book store
13. Order By
14. Having Clause : For filtration
15. Group By
16. GRANT : grant select permission on <Table_name> to <User>
17. REVOKE

Microsoft Edge content:

HELLO WORLD!

storage1account | Static website

Name: 2023-09-04T09:30:06.000000Z

Date created: 9/4/2023, 3:00:06 PM

Initiator: Manual

Storage account: storage1account

Static website

Static website configuration

Static website URL: https://storage1account.blob.core.windows.net/

Static website root blob: index.html

Static website index blob: index.html

Static website error blob: error.html

Static website blob type: Block blob

Static website blob container: static

Static website blob prefix: static

Static website blob suffix: static

Static website blob content type: application/javascript

Static website blob content encoding: gzip

Static website blob content disposition: attachment

Static website blob content security policy: Content-Security-Policy: default-src 'self'; script-src 'self' 'unsafe-eval'; style-src 'self' 'unsafe-style-src'

Static website blob content type: application/javascript

Static website blob content encoding: gzip

Static website blob content disposition: attachment

Static website blob content security policy: Content-Security-Policy: default-src 'self'; script-src 'self' 'unsafe-eval'; style-src 'self' 'unsafe-style-src'

DAY-6

Azure Storage Types- Day 2 & Azure Data Factory

1. Access Keys : Lifetime
2. SAS (Shared Access Signature) : Time-bounded
3. Azure Key Vaults
4. Access Control (IAM) : Check Access Here
5. Azure Data Factory :
 - a. Data Engineer Activities:
 - i. ETL (Extract, Transform, Load)
 - ii. ELT
 - iii. Azure Databricks - Transformation
 - b. Data Orchestration
 - c. ADF Components:
 - i. Pipelines : Combination Of One Or More Activities
 - ii. Activities
 - iii. Datasets
 - iv. Linked Services : To Access Particular Directory This Is Used.
 - v. Integration Runtime : Gateway Or Bridge To Connect For Data Extraction.
 1. Azure Auto Resolved (Default)
 2. Self-hosted
 3. Azure SSIS
 - vi. Triggers
 - vii. Data Flow
 - d. Parameterization
 - e. Pipelines

anushkashell | Shared access signature

Some routing options are disabled because the endpoints are not published.

keyuser | Overview

keyuser | Secrets

anushkashell | Access Control (IAM)

anushkashell | Access Control (IAM)

Shellunext unexIDA90 assignments - anushkashell

Microsoft Azure Data Factory

DAY-7

Azure Data Factory - Day 1

1.Create Data Factory

2.Activities In Pipeline :

- a. String, Boolean, Array And Integer Activities In Pipelines
- b. Execute Pipeline Activity
- c. SSIS Package (Azure SSIS IR - Required For This Package)
- d. Get Metadata Activity

1.Copy Data Activity Implementation

2.Parameter Concept Implementation

3.Triggers

4.Get Metadata Activity

7.If-else Condition For Get Metadata In Pipelines.

8.For Each Loop

Microsoft Azure | Search resources, services, and docs (G+)

All services > Microsoft.DataFactory-20230906091019 | Overview >

df-anushkashell

Data factory (V2)

Search

Essentials

- Resource group (move) : RG_Anushka
- Status : Succeeded
- Location : East US
- Subscription (move) : npunext-1680261348707
- Subscription ID : 6f5f0ab3-77b8-43f8-9c0e-b251f714e513

Type : Data factory (V2)

Getting started : Quick start

Azure Data Factory Studio

Launch studio

Quick Starts, **Tutorials**, **Template Gallery**, **Training Modules**

https://portal.azure.com/#@npunext.onmicrosoft.com/resource/subscriptions/6f5f0ab3-77b8-43f8-9c0e-b251f714e513/resourceGroups/RG_Anushka/providers/Microsoft.DataFactory/factories/df-anush...

Microsoft Azure | Data Factory > df-anushkashell | Data Factory | Validate all | Publish all 1

Microsoft recently announced the public preview of Microsoft Fabric, a brand new and exciting way to build cloud-first data analytics. Click [here](#) to get started with Fabric Data Factory!

Factory Resources

- Pipelines: PL_IDA_01
- Datasets: DS_source, DS_Output
- Data flows: 0
- Power Query: 0

Activities

- Copy data
- Get Metadata
- Set variable
- Append variable
- Delete
- Execute Pipeline
- Execute SSIS package
- Fail
- Get Metadata
- Lookup
- Stored procedure
- Script
- (X) Set variable
- Validation
- Web
- WebHook
- Wait
- > HDInsight
- > Iteration & conditionals
- > Machine Learning

Preview experience Off

PL_IDA_01

DS_Source

DS_Destination

PL_IDA_02

Activities

Validate, Debug, Add trigger

Copy data1 → **Get Metadata1** → **Set variable1**

Parameters, **Variables**, **Settings**, **Output**

Pipeline run ID: d45d2417-5ff1-426f-bb88-87c17d7123ea | **Pipeline status:** Succeeded

Activity name	Activity status	Activity type	Run start	Duration	Log	Integration runt
Set variable1	Succeeded	Set variable	9/6/2023, 2:45:18 PM	Less than 1s		
Get Metadata1	Succeeded	Get Metadata	9/6/2023, 2:45:13 PM	4s		AutoResolveInteg
Copy data1	Succeeded	Copy data	9/6/2023, 2:45:03 PM	9s		AutoResolveInteg

Microsoft Azure | Data Factory | Validate all | Publish all 1

Microsoft recently announced the public preview of Microsoft Fabric, a brand new and exciting way to build cloud-first data analytics. Click [here](#) to get started with Fabric Data Factory!

Activities

- Set variable
- Validation
- Web
- WebHook
- Wait
- > HDInsight
- > Iteration & conditionals
- Filter
- ForEach
- If Condition
- Switch
- Until

PL_IDA_01

DS_Source

DS_Destination

PL_IDA_02

Activities

Validate, Debug, Add trigger

If Condition

```

graph TD
    GetMetadata1[Get Metadata] --> IfCondition{If Condition}
    IfCondition -- True --> CopySourceData[Copy Source Data]
    IfCondition -- False --> End1(( ))
    CopySourceData --> End1

```

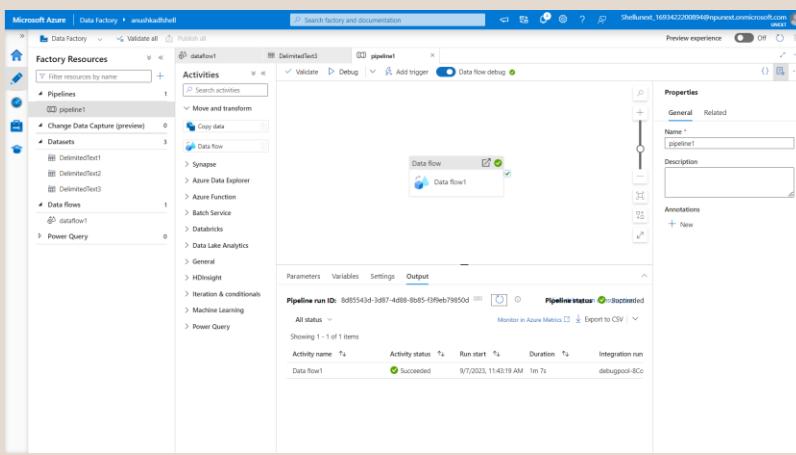
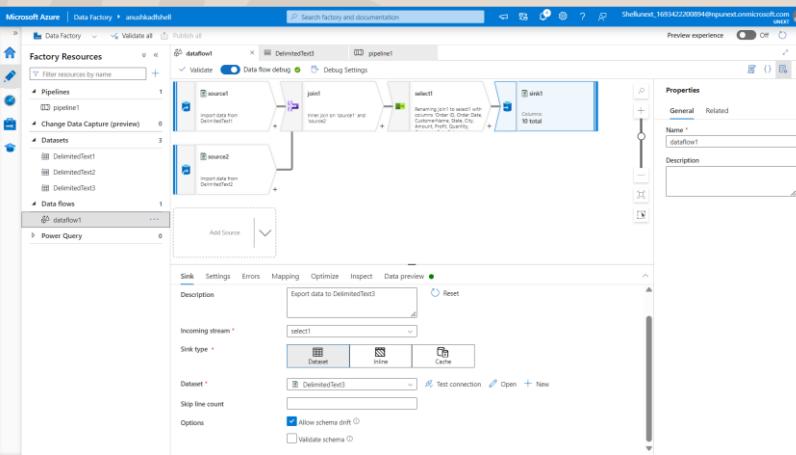
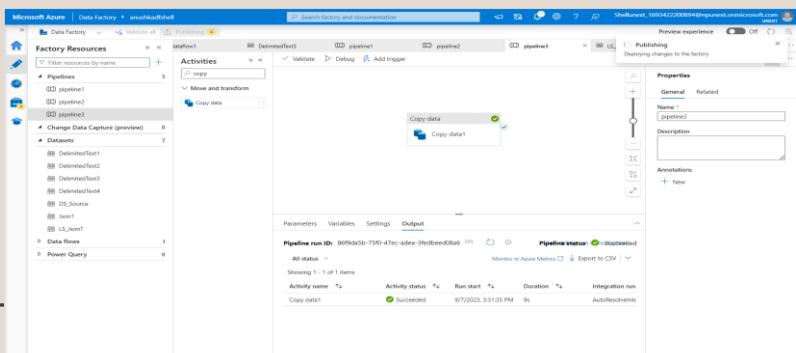
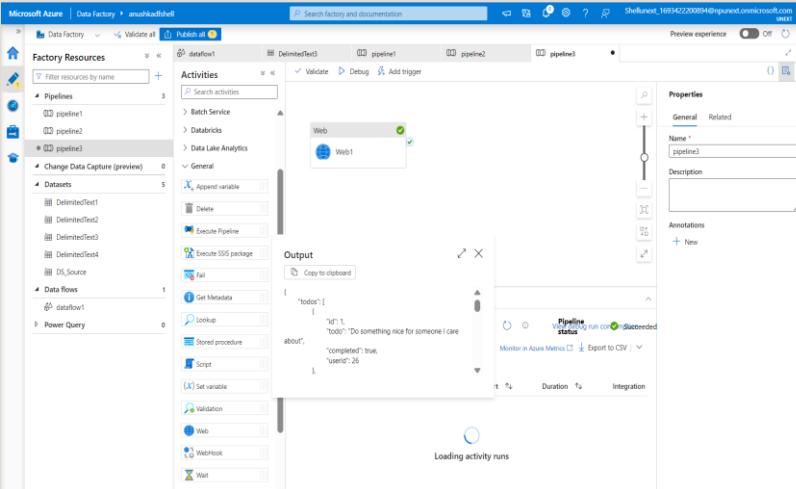
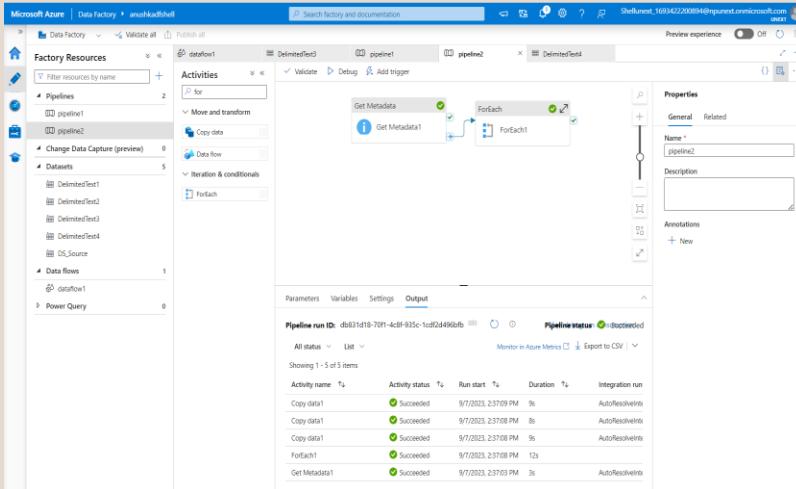
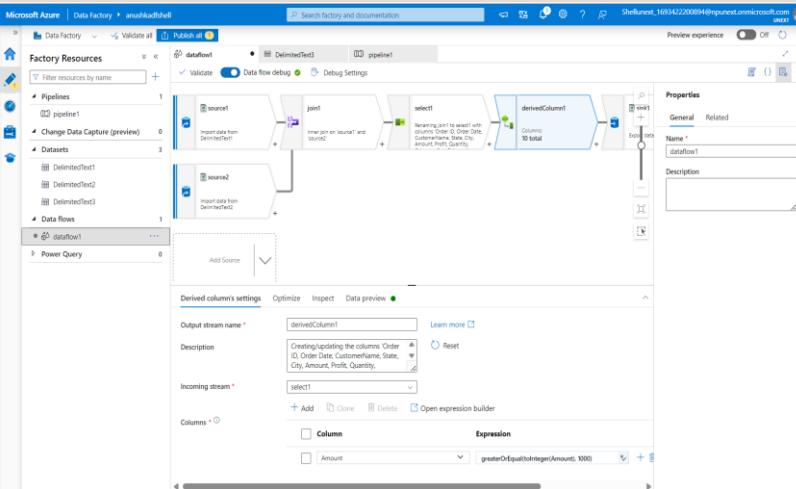
Parameters, **Variables**, **Settings**, **Output**

Activity name	Activity status	Activity type	Run start	Duration	Log
Copy data1	Succeeded	Copy data	9/6/2023, 2:52:44 PM	10s	
Copy Data Condition	Succeeded	If Condition	9/6/2023, 2:52:43 PM	11s	
Get Metadata1	Succeeded	Get Metadata	9/6/2023, 2:52:40 PM	2s	

DAY-8

Azure Data Factory - Day 2

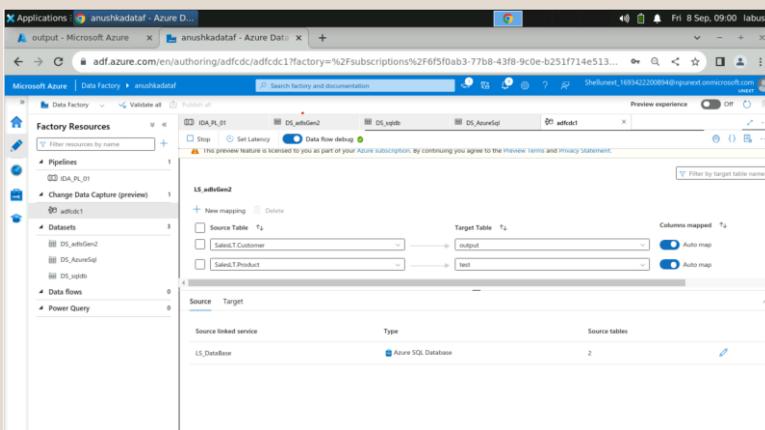
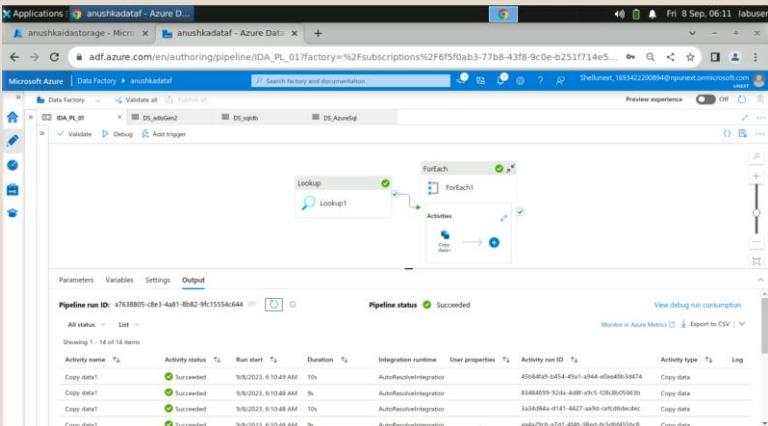
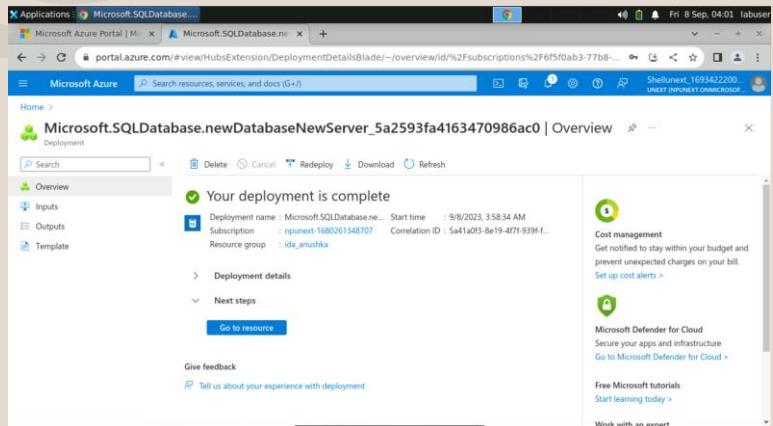
1. Data(resource)transformation Using Data Flow.
2. For Each Activity
3. Data Flows
4. Dataflow Expression Builder And Derived Columns
5. Surrogate Key
6. For Each Container And Get Metadata
7. API

DAY-9

Azure Data Factory - Day 3

- 1.Create SQL Database On VM.
- 2.Lookup Table In Sql Database
- 1.Lookup Table With Foreach Pipeline
- 4.CDC (Change Data Capture)
- 5.Runbook Or Automation



DAY-10

Azure Synapse Analytics

1. Azure Synapse Analytics : Used To Create ADF Services, Spark Notebooks, SQL Data Warehouse
2. SQL Pool
 - Serverless
 - Dedicated Pool
3. Spark Pool
4. Data Exploration
5. SQL Database
6. Collate
7. Common Table Expression (CTE)
8. View
9. DWU (Data Warehousing Unit)
10. Storage Node
11. Compute Node
12. Data Movement Service (DMS)
13. Massive Parallel Processing
14. Replicated Tables
15. Hash Distributed Table
16. Implementing Partitions For An SQL Data Warehouse

Microsoft Azure | Search resources, services, and docs (F1)

Home > Microsoft.Azure.SynapseAnalytics-20230912034630 | Overview

Deployment

Search

Delete Cancel Redeploy Download Refresh

Your deployment is complete

Deployment name : Microsoft.Azure.SynapseAnalytics-20230912034630 Start time : 9/12/2023, 3:56:42 AM
Subscription : rpuwest188026134877 Correlation ID : 3eab269-82d0-4cb6-b6fc-bc47b0ffa78
Resource group : RG-Anushka

Deployment details

Next steps

Go to resource group

Give feedback

Tell us about your experience with deployment

Cost management Get notified to stay within your budget and prevent unexpected charges on your bill Set up cost alerts >

Microsoft Defender for Cloud Secure your apps and infrastructure Go to Microsoft Defender for Cloud >

Free Microsoft tutorials Start learning today >

Work with an expert Azure experts are service provider partners who can help manage your assets on Azure and be your first line of support. Find an Azure expert >

Applications : anushkasynapse - Azure... Tue 12 Sep, 04:58 labus

Microsoft Azure Portal | Microsoft Edge | anushkasynapse - Microsoft Edge | anushkasynapse - Azure | +

web.azuressynapse.net/en/authoring/explore/workspace/sqlscripts/SQL%20script%20?workspace=%2Fsubscriptions%2F6... Tue 12 Sep, 04:58 labus

Synapse Analytics - anushkasynapse

Microsoft Azure | Synapse Analytics | anushkasynapse | Search

Synapse live | Validate all | Publish all

Data Workspace Linked

Filter resources by name

SQL database

anushka (SQL)

External tables

External resources

Views

Schemas

Security

math (SQL)

SQL script 1

Run Undo Publish Query plan Connect to Built-in Use database master

alter database anushka collate Latin1_general_100 BIN2 UTF8

Results Messages

No results to show

Your query yielded no displayable results

1000011 Queries executed successfully.

The screenshot displays two separate browser windows, both titled "anushkasynapse - Azure..." and connected to the same Microsoft Azure Portal instance.

Top Window:

- Shows a "Data" workspace named "anushka".
- The "Linked" tab is selected.
- A query titled "SQL script 1" is running, showing T-SQL code for reading a CSV file from a blob storage location using BULK INSERT.
- The results table shows 10 rows of zip code data.
- Message bar: "00:00:14 Query executed successfully."

Bottom Window:

- Shows a "Data" workspace named "anushka".
- The "Linked" tab is selected.
- A query titled "SQL script 2" is running, showing T-SQL code for reading a CSV file from a blob storage location using BULK INSERT.
- The results table shows 10 rows of zip code data.
- Message bar: "00:00:14 Query executed successfully."

The screenshot shows a Microsoft Azure Synapse Analytics workspace titled 'anushkasynapse - Azure...'. The left sidebar displays the 'Data' workspace, with a 'Linked' section showing a connection to 'anushka'. The main area contains a code editor with the following SQL script:

```
14 Select * from cte test1/
15
16 CREATE view anushka_view as
17     SELECT
18         TOP 100 *
19     FROM
20     OPENROWSET(
21         BULK 'https://anushkaautomationshell.dfs.core.windows.net/anushkafile/zipcodes.csv',
22         FORMAT = 'CSV',
23         PARSER_VERSION = '2.0',
24         HEADER_ROW = TRUE
25     ) AS [result]
26
27 |
```

The 'Results' tab is selected, showing the message 'No results to show' and 'Your query yielded no displayable results'. To the right, the 'Properties' panel shows the script is named 'SQL script 2' and has a size of 241 bytes. The 'Results settings per query' section is set to 'First 5000 rows (default)'.

DAY-11

Data Visualization - Power BI

1. PBI Desktop
2. Power BI Supports Multiple Data Connectors.
3. Power Query Editor: Used To Transform Data
 - Language Used At Back-end - M Language
 - Different Fields Of This Editor
4. Get Data In PBI Report And Transform It In Power Query Editor
5. Getting Familiar With Different Visuals
6. Parameter
7. Filters :
 - Visual Level
 - Page Level
 - Report Level
8. Slicers
9. DAX Measures
10. DAX Calculated Columns
11. RLS (Manage Roles)

The screenshot shows the Microsoft Power Query Editor interface. A query named "financials" is selected. The data preview shows 19 rows of financial data with columns: Year, Total_Gross_Revenue, and Short_Name. A context menu is open over the "Short_Name" column, with the "Changed Type3*" option highlighted. The "APPLIED STEPS" pane on the right shows the history of changes made to the query, including "Added Condition...".

The screenshot shows the Power Query Editor interface with a query named "financials". The "Applied Steps" pane on the right highlights the "Changed Type3" step, which was used to change the type of the "Short_Name" column. The data table contains 19 columns and 790 rows, with the first few rows showing columns like "Country", "Product", and "Discount Band".

	Country	Product	Discount Band
1	Canada	Carretera	None
2	Germany	Carretera	None
3	France	Carretera	None
4	Germany	Carretera	None
5	Mexico	Carretera	None
6	Germany	Carretera	None
7	Germany	Montana	None
8	Canada	Montana	None
9	France	Montana	None
10	Channel Partners	Montana	None
11	Germany	Montana	None
12	Mexico	Montana	None
13	Enterprise	Canada	None
14	Mexico	Montana	None
15	Small Business	Montana	None
16	Government	Montana	None
17	Enterprise	Canada	None
18	Midmarket	United States of America	Montana
19	Government	Canada	Paseo
20	Channel Partners	Mexico	Paseo
21	Small Business	Canada	Paseo

Untitled - Power BI Desktop

https://wai.nuvepro.com/guacamole#/client/a50WymFYTRjN2M3NDHZGE3ZABjAG51dmVsW5r?h...

Search Sign in

Name: financials

File Home Insert Modeling View Help Format Data / Drill Table tools

Mark as date table ▾ Calendars Relationships

New Quick New measure New measure column New table Calculations

Structure

Sum of Sales by Country

Country: United States (25.00%), Canada (24.25%), France (24.25%), Germany (24.25%), Mexico (24.25%)

25.00% (25.00%)

24.25% (24.25%)

24.25% (24.25%)

24.25% (24.25%)

24.25% (24.25%)

Country: United States Canada India France Germany Mexico

Sum of Revenue by Country

Country: United States (2.0M), Canada (1.8M), India (1.8M), France (1.8M), Germany (1.8M)

2M

1M

0M

United States Canada India France Germany

Filters

Search

Filters on this visual

Country is (All)

Add data fields here

Visual General

Slicer settings

Values

Filters on this page

Add data fields here

Filters on all pages

Add data fields here

Format visual

Search

Visualizations

Fields

financials

Discount Band

Discounts

Gross Sales

Manufacturing

Month Name

Month Number

Product

Profit

Sale Price

SegCountry

Page 1 of 1

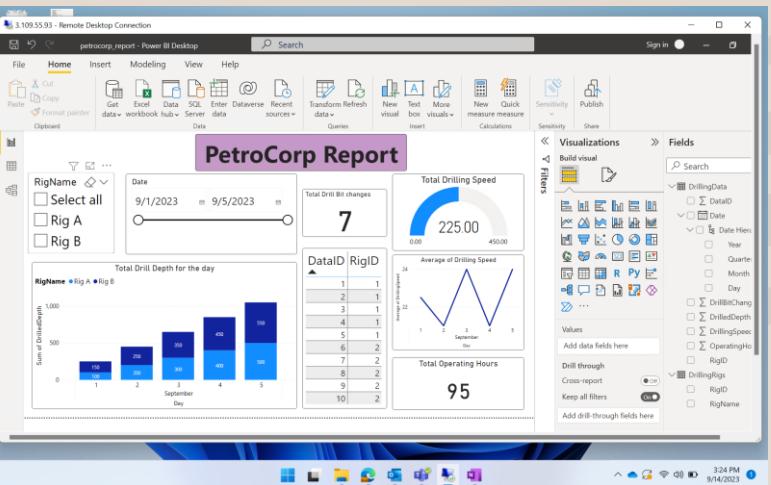
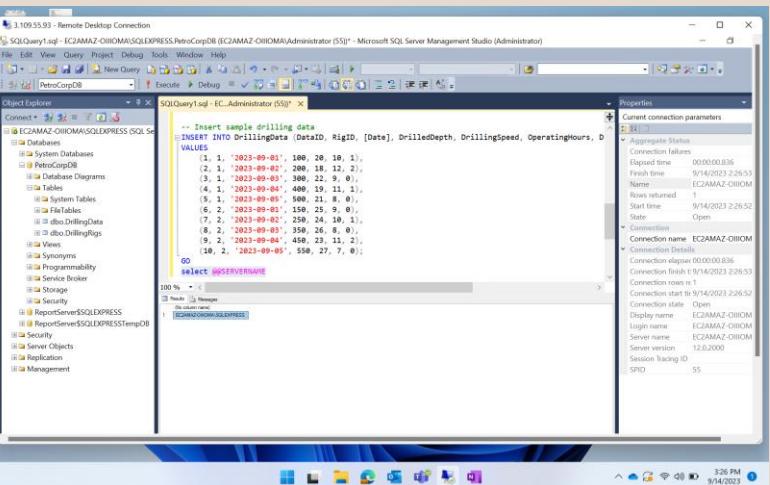
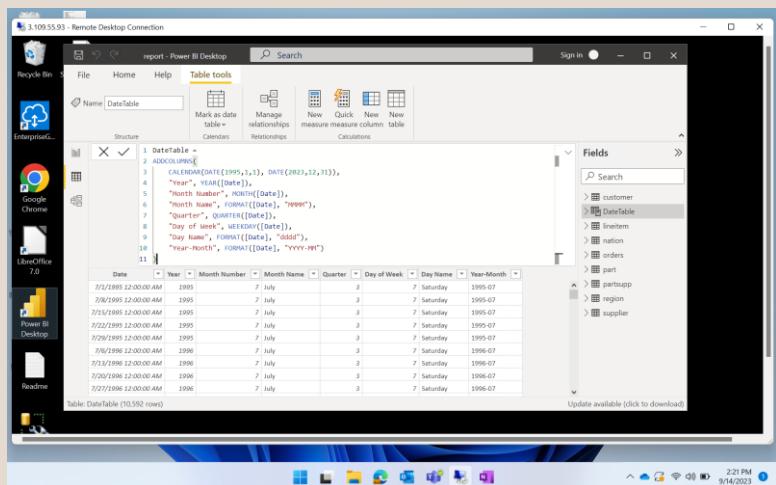
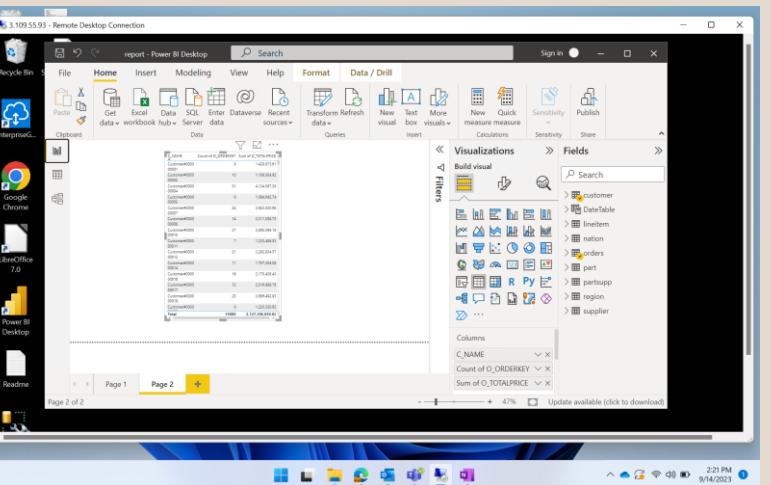
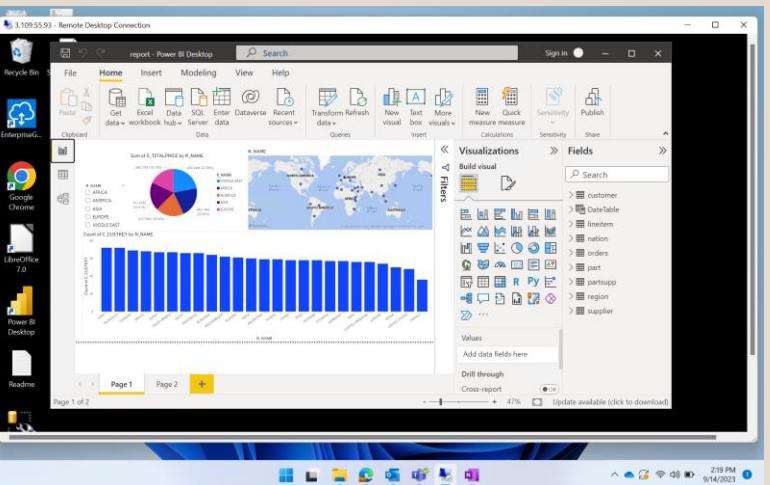
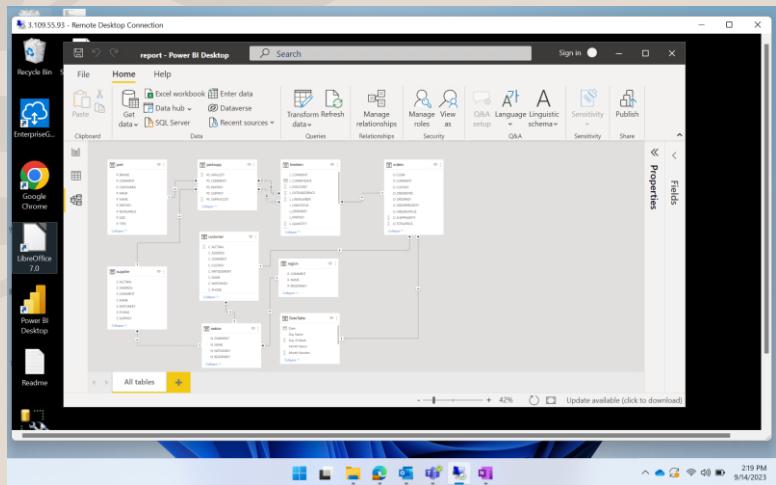
Update available (click to download)

23:57 PM 9/13/2023 ENG IN

DAY-12

Power BI - Day 2

- 1.Data Modeling : Establishing Relationships Between Tables
- 2.Univariant
- 3.Bivariant
- 4.Multivariant
- 5.Drill Through
- 6.Date Table
- 7.Connection Of SQL Server And Power BI



PetroCorp Report

Rig Name

Rig A

Rig B

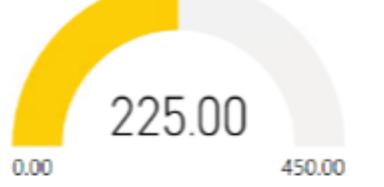
Date

9/1/2023  9/5/2023

Total Drill Bit changes

7

Total Drilling Speed

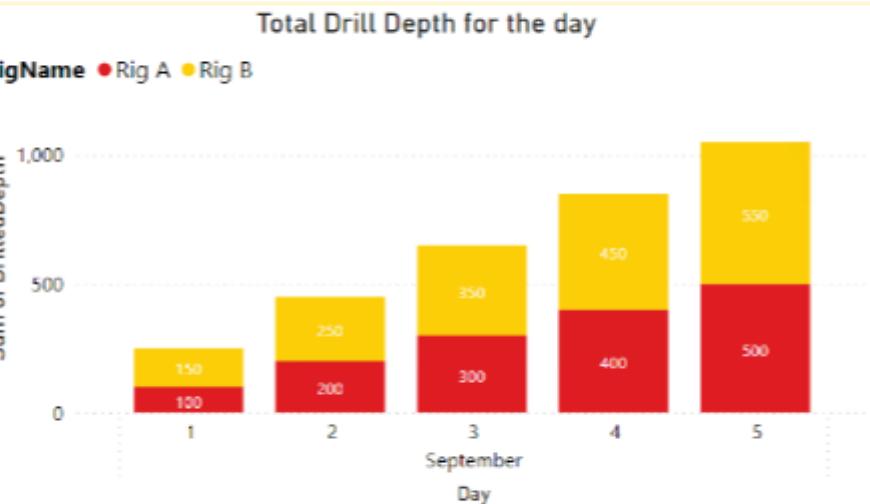


225.00

0.00 450.00

Total Drill Depth for the day

RigName • Rig A • Rig B

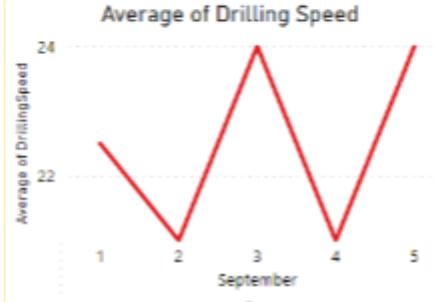


Day	Rig A	Rig B	Total
1	100	150	250
2	200	250	450
3	300	350	650
4	400	450	850
5	500	550	1050

DataID | RigID

DataID	RigID
10	2
9	2
8	2
7	2
6	2
5	1
4	1
3	1
2	1
1	1

Average of Drilling Speed



24

22

1 2 3 4 5

September Day

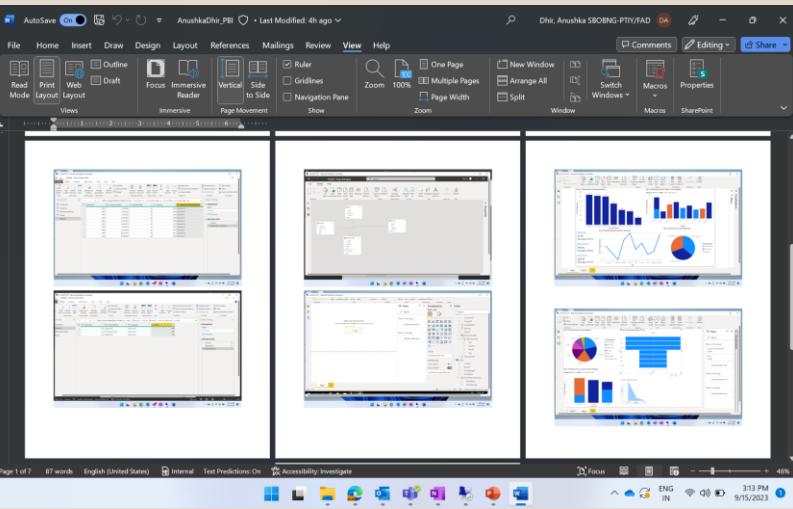
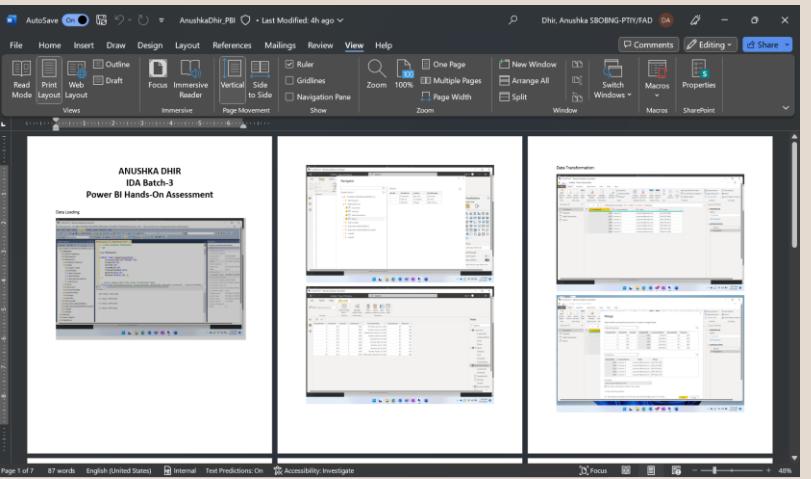
Total Operating Hours

95

DAY-13

Python – Day 1

1. Hands-on Assessment - Power BI
2. Jupyter Notebook Launch
3. Datatypes In Python:
 - Integer
 - Float
 - String
 - List
 - Dictionary
 - Tuple
 - Boolean
4. Arithmetic
5. Logic



```
In [1]: # Integers
age = 20
print("Age is ", age)

# Floating Point Numbers
price = 19.99
print("Price is ", price)

# Strings
name = "Alice"
message = "Hello, World!"
print("My name is ", name)

# Boolean values
is_python_installed = True
is_learning = False

# Lists
numbers = [1, 2, 3, 4, 5]
fruits = ["apple", "banana", "cherry"]

# Tuples
coordinates = (3, 4)

# Dictionaries
person = {"first": "Alice", "age": 20, "city": "New York"}
grades = {"math": 90, "science": 85, "history": 80}

# Accessing elements in lists and dictionaries
first_number = numbers[0]
last_fruit = fruits[-1]
name1 = person["first"]
math_grade = grades["math"]

print(first_number)
print(last_fruit)
print(name1)
print(math_grade)

Age is 20
Price is 19.99
My name is Alice
1
cherry
Alice
92
```

```
In [1]: #!/usr/bin/python
# This script calculates profit based on user satisfaction and price.

class User:
    profit = False
    user_satisfaction = None

    def __init__(self, profit, low_price, user_satisfaction):
        self.profit = profit
        self.low_price = low_price
        self.user_satisfaction = user_satisfaction

    def set_low_price(self):
        self.low_price = low_price

    def set_user_satisfaction(self):
        self.user_satisfaction = user_satisfaction

    def calculate_profit(self):
        self.profit = self.user_satisfaction * self.low_price

    def __str__(self):
        return f"User Profit: {self.profit}, Low Price: {self.low_price}, Satisfaction: {self.user_satisfaction}"

user1 = User(profit=False, low_price=10, user_satisfaction=0.5)
user2 = User(profit=False, low_price=15, user_satisfaction=0.8)
user3 = User(profit=False, low_price=20, user_satisfaction=0.9)

Region = [Region(i) for i in range(5)]
for r in Region:
    r.set_low_price(10)
    r.set_user_satisfaction(0.5)
    r.calculate_profit()

print(user1)
print(user2)
print(user3)
```

DAY-14

Python – Day 2

1. Function
2. Function Composition
3. Recursive Function
4. Generator Function
5. Decorator (Wrap Function)
6. Types Of Argument
7. Class
8. Date Stripping
9. Date Conversion
10. Exception Handling
11. List Comprehension
12. Lambda Function

The screenshot shows a Jupyter Notebook interface with several code cells and their outputs. The code is related to calculating gas volume and mass based on pressure, gas constant, and temperature.

```
print(seats)
['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S', 'T', 'U', 'V', 'W', 'X', 'Y', 'Z']

In [3]: def func(pressure,gas_constant,temperature):
    gas = Gas(gas_constant,pressure,temperature)
    return vol

gas.vol_func(100000, 1.9, 300)
print(gas.vol_func(100000, 1.9, 300))
Gas volume=4.0930157964623 m3

In [4]: def mass(reserve_gas_constant,temp,pressure,mass):
    func(pressure,gas_constant,temp)*mass
    return mass

mass(100000, 1.9, 300, 445)
174.488295821209 kg

In [5]: def seg(n):
    if n == 0:
        return []
    else:
        return arr[n-1]
print(seg(1))
[]

In [6]: def calc_tot(segments):
    if not segments:
        pass
    else:
        curr_seg_depth = segments[0]
        remaining_seg = segments[1:]

Remaining 36 hrs/51 mins
```

The screenshot shows a Jupyter Notebook interface with several code cells and their outputs. The notebook title is "Untitled" and it was last checked 2 hours ago.

- In [1]:** `print(calc_tet_depth(0))`
Output: 120
- In [10]:** `def calc_tet_depth(segments):`
If not segments:
 return 0
else:
 curr_seg_depth = segments[0]
 remaining_seg = segments[1:]
 recursive_call = calc_tet_depth(remaining_seg)
 print(calc_tet_depth([1,2,3]))
 return curr_seg_depth + recursive_call
Output: 6
- In [11]:** `def generate_squares(n):`
 yield 1 ** n
 for i in range(1, n):
 print(i ** n)
 yield i ** n
 1
 4
 9
 16
 25
- In [13]:** `import datetime`
`def calculateMonthlyProduction(yearlyProduction):`
month = 1
for month in range(1,13):
 x = datetime.datetime(2023, month, 1)
 print(f"Month {x.month} Production: {yearlyProduction * yearlyProduction / 12})")
 month = month + 1

The screenshot shows a Jupyter Notebook interface with the following content:

```
13.232.140.15 - Remote Desktop Connection
```

In [1]:

```
10  
11  
12
```

In [13]:

```
insert dataset
def calculateMonthlyProduction(yearlyProduction):
    for i in range (1,13):
        yearlyProduction[i] = str(datetime.datetime(1981, month, 1))
        yearlyProduction[i] = yearlyProduction[i].strftime("%M", yearlyProduction[i])
    monthlyProduction = yearlyProduction[1:13]
```

In [14]:

```
# To Calculate Monthly Production(2000)
print("Production in {} month is {} production".format(*monthlyProduction))
```

Production in January is 1800.0
Production in February is 1800.0
Production in March is 1800.0
Production in April is 1800.0
Production in May is 1800.0
Production in June is 1800.0
Production in July is 1800.0
Production in August is 1800.0
Production in September is 1800.0
Production in October is 1800.0
Production in November is 1800.0
Production in December is 1800.0

In [15]:

```
def my_decorator(function):
    def inner():
        print("I am starting")
        function()
        print("I am completed")
    return inner

@my_decorator
def calculate():
    print("Hello")
```

In [16]:

```
i am starting
Hello
I am completed
```

In [17]:

```
13.232.140.15 - Remote Desktop Connection

File Edit View Insert Cell Kernel Help

In [10]: def my_decorator():
    def wrapper(*args, **kwargs):
        print("I am starting")
        func = args[0]
        func(*args, **kwargs)
        print("I am completed")
    return wrapper

@my_decorator
def hello():
    print("Hello")

I am starting
Hello
I am completed

In [11]: @parent(logging
def logging(*args, **kwargs):
    def wrapper(func):
        func(*args, **kwargs)
        logging.warning('started')
        func(*args, **kwargs)
        logging.warning('finishing')
    return wrapper
    return wrapper

@parent
def my_func():
    print("My func")
    my_func()

WARNING:root:root started
WARNING:root:root finished

Hello

In [12]:
```

13.232.140.15 - Remote Desktop Connection

File Edit View Insert Cell Kernel Help

Not Tracked Python 3 (ipython)

```
my sum()

WARNING:root:started
WARNING:root:teaching
None

In [2]: class my_cis:
    def __init__(self,p,t):
        self.p = p
        self.t = t
        x = np.zeros((p,t))
        self.x = x
        self.y = np.zeros((p,t))

    def max(self,p,t,m):
        return np.max(np.multiply(self.p,x),t=m)

    def fact(self,n):
        if n <= 1:
            else:
                return n*self.factorial(n-1)

    def calc_tot_depth(self,segments):
        if not segments:
            else:
                curr_sep_depth = segments[0]
                remaining_sep = segments[1:]
                return curr_sep_depth + self.calc_tot_depth(remaining_sep)

    x = my_cis(10,10)
    print(x.max(100000,200))
    print(x.max(100000,200))
    print(x.fact(10))
    print(x.calc_tot_depth([1,2,3]))
    4.00000157964813
    8.00000157964813
    120
    In [1]:
```

Parumaranthan, Sumita D. 8/25

The screenshot shows a Jupyter Notebook interface running in a browser window titled "12.232.140.15 - Remote Desktop Connection". The notebook has two visible cells:

```
# In [1]:  
import time  
from datetime import datetime  
curr_time = time.time()  
testdate = datetime.fromtimestamp(curr_time).strftime('%Y-%m-%d')  
print(testdate)  
2023-09-19  
  
In [2]:  
from datetime import datetime  
curr_datetime = datetime.now()  
print(curr_datetime)  
(curr_datetime.replace(minute=0)).strftime("%H")  
print(curr_hour)  
0  
  
In [3]:
```

```
69

In [8]: a=10
         b=0
         try:
             result=a/b
             print(result)
         except:
             print("Error : Someone divided by zero")
Error : Someone divided by zero

In [9]:
```

```
In [10]: a = [i**2 for i in range(1,6)]
          print(a)
```

[1, 4, 9, 16, 25]

The screenshot shows a Jupyter Notebook interface with several code cells and their corresponding outputs:

- In [1]:** `print(datetime.datetime.now())`
Output: `2023-09-10 14:49:48.918791`
- In [2]:** `from datetime import datetime
curr_datetime = datetime.now()
curr_minute = curr_datetime.strftime('%M')
curr_hour = curr_datetime.strftime('%H')
curr_date = curr_datetime.strftime('%d')
curr_year = curr_datetime.strftime('%Y')`
Output: `2023-09-10 09:49:48.918791`
- In [3]:** `a=20
b=0
try:
 result=a/b
 print(result)
except:
 print("Error: Someone divided by zero")`
Output: `Error: Someone divided by zero`
- In [4]:** `b = 1
for i in range(1,10) if i%2==1:
 print(i)`
Output: `1, 3, 5, 7, 9, 11, 13, 15, 17, 19`
- In [5]:** `Lambda avg:avg`