Databricks:

Databricks is a cloud-based data analytics platform that provides an interactive workspace for data engineers, data scientists, and analysts to collaborate on big data and machine learning projects. It is built on **Apache Spark** and integrates well with **Azure, AWS, and Google Cloud**.

Key Features of Databricks

1. **Unified Data Platform** – Combines data engineering, data science, and machine learning.
2. **Apache Spark Optimization** – Provides a managed Spark environment with auto-scaling and performance improvements.
3. **Multi-Cloud Support** – Available on **Azure (Azure Databricks), AWS (Databricks on AWS), and GCP**.
4. **Notebooks** – Supports Python, Scala, SQL, and R for interactive data processing.
5. **Delta Lake** – An optimized storage layer that enhances **data reliability and performance**.
6. **Security & Compliance** – Provides enterprise-grade security and compliance support.
7. **Collaborative Workspace** – Allows multiple users to work together in shared notebooks.

Databricks uses its own filesystem for data processing operation, called DBFS (Databricks File system). It is a Linux based file system and we will Unix commands for various data processing operations.

Databricks File System (DBFS)

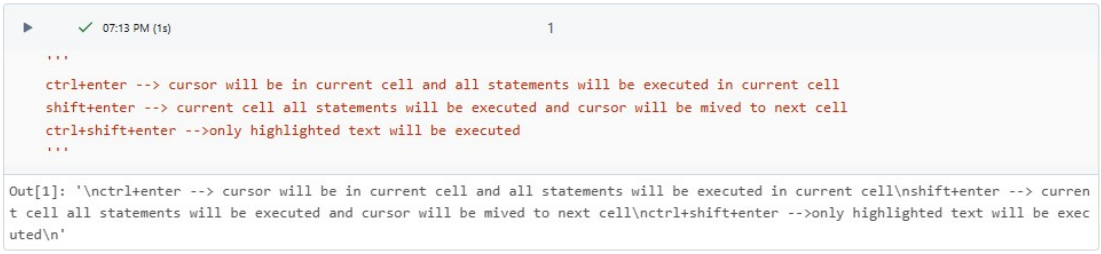
DBFS is a **distributed file system** built into Databricks. It allows users to interact with files and directories in **Databricks clusters** as if they were a regular file system.

Key Features of DBFS

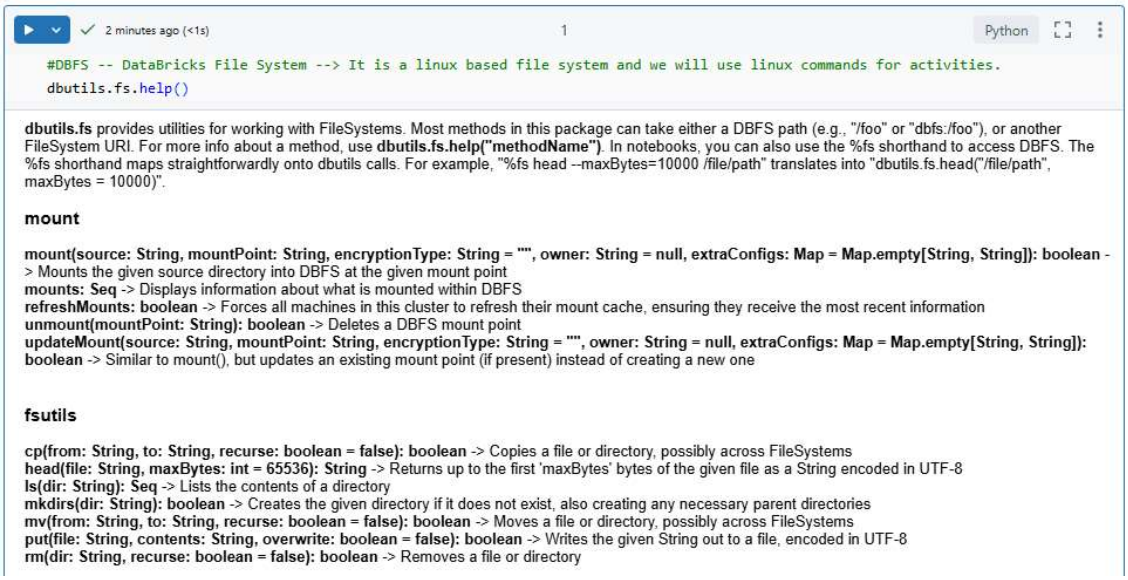
- **Mountable Storage** – Can mount cloud storage (Azure ADLS, AWS S3, Google Cloud Storage).
- **Hierarchical File Structure** – Files are organized in directories.
- **Supports Different File Formats** – CSV, JSON, Parquet, Delta, etc.
- **Access Control** – Permissions for files and directories.

- In Databricks, we will use dbutils (databricks utilities) to interact DBFS and for data processing operations.

Imp points#



dbutils.fs.help() → This command will display the different types of supported commands in DBFS.



Main commands used are:

mkdirs → To create a new directory

ls → It is used to list files in a specified directory

cp → To copy a file from one directory to another directory

mv → used to move the file from one directory to other directory

head →used to display the content of the file

put → used to put the content into a file

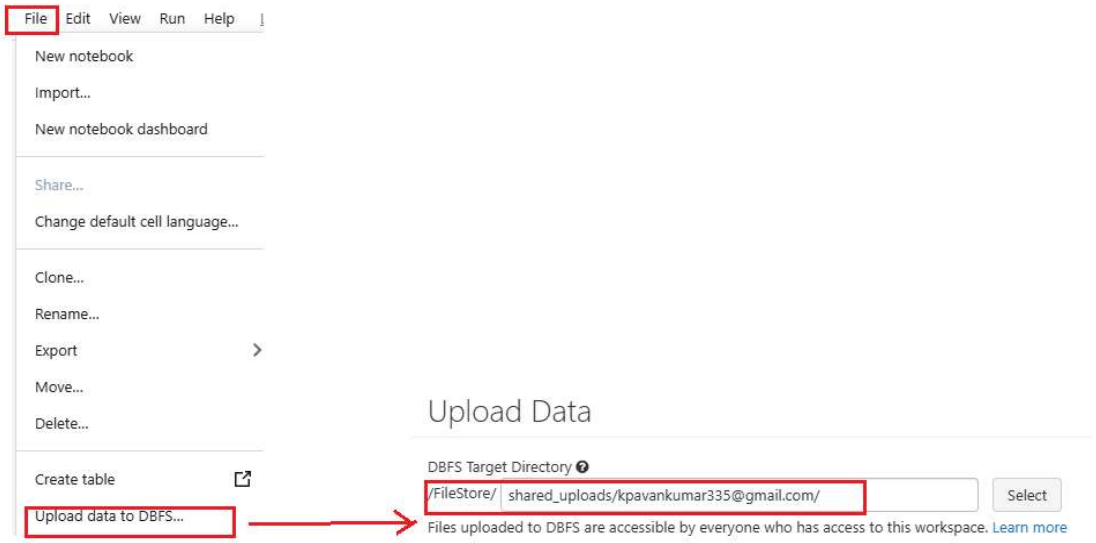
rm → used to delete the file permanently

1. **Create a new directory:**

#Syntax → . dbutils.fs.mkdirs(‘file system path”)

The path of the file system, can be taken by below process.

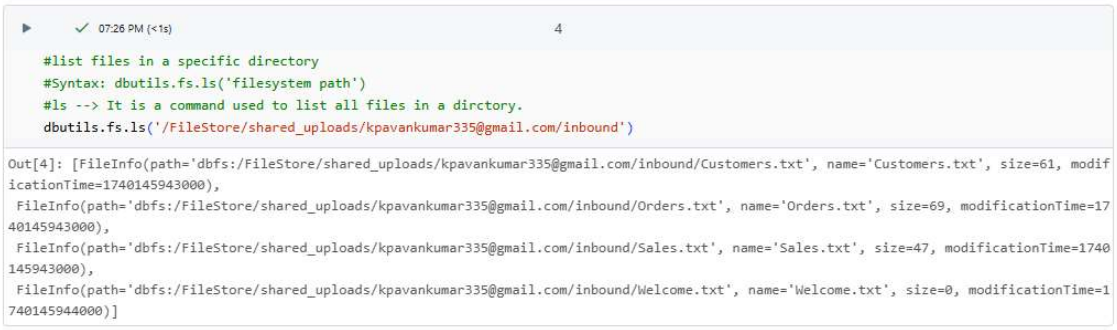
Click on File → Upload data to DBFS → Then take the path



3 directories are created. Inbound, outbound, srcfiles.

2. **I uploaded the files using above process in a pic, lets list the files in a inbound folder.**

#Syntax → dbutils.fs.ls(‘path to the directory)



3. Copy a file between directories.

#Syntax → dbutils.fs.cp('source path', 'Target path')

```
Just now (1s)                                                    5

#copy files between inbound and outbound folders
#Syntax: dbutils.fs.cp(source path, target path)
#cp --> It is a command used to copy file between source and target.

dbutils.fs.cp(
    '/FileStore/shared_uploads/kpavankumar335@gmail.com/inbound/Customers.txt',
    '/FileStore/shared_uploads/kpavankumar335@gmail.com/outbound'
)

Out[8]: True
```

Here we can the customer.txt file copied to outbound foler.

```
Just now (<1s)                                   6                Python

dbutils.fs.ls('/FileStore/shared_uploads/kpavankumar335@gmail.com/outbound/')

Out[9]: [FileInfo(path='dbfs:/FileStore/shared_uploads/kpavankumar335@gmail.com/outbound/Customers.txt', name='Customers.txt', size=61, modi
ficationTime=1740147386000)]
```

Case:

If we want to copy all files from source to target, we can use recursive argument.

By default, the recursive option will be false. We just mention it as True.

```
Just now (1s)                                                    7

#Copy all files from inbound to outbound using recursive
#By default the recursive option will set to false, we need to make true.
#Syntax --> dbutils.fs.cp('src path','target path',True)

dbutils.fs.cp(
    '/FileStore/shared_uploads/kpavankumar335@gmail.com/inbound',
    '/FileStore/shared_uploads/kpavankumar335@gmail.com/outbound', True
    )

Out[10]: True
```

Inbound directory:

```
Just now (<1s)                                   8

dbutils.fs.ls('/FileStore/shared_uploads/kpavankumar335@gmail.com/inbound')

Out[11]: [FileInfo(path='dbfs:/FileStore/shared_uploads/kpavankumar335@gmail.com/inbound/Customers.txt', name='Customers.txt', size=61, modi
ficationTime=1740145943000),
 FileInfo(path='dbfs:/FileStore/shared_uploads/kpavankumar335@gmail.com/inbound/Orders.txt', name='Orders.txt', size=69, modificationTime=17
40145943000),
 FileInfo(path='dbfs:/FileStore/shared_uploads/kpavankumar335@gmail.com/inbound/Sales.txt', name='Sales.txt', size=47, modificationTime=1740
145943000),
 FileInfo(path='dbfs:/FileStore/shared_uploads/kpavankumar335@gmail.com/inbound/Welcome.txt', name='Welcome.txt', size=0, modificationTime=1
740145944000)]
```

Outbound directory:

```
Just now (<1s)   9   Python
dbutils.fs.ls('/FileStore/shared_uploads/kpavankumar335@gmail.com/outbound')

Out[12]: [FileInfo(path='dbfs:/FileStore/shared_uploads/kpavankumar335@gmail.com/outbound/Customers.txt', name='Customers.txt', size=61, modificationTime=1740147755000),
 FileInfo(path='dbfs:/FileStore/shared_uploads/kpavankumar335@gmail.com/outbound/Orders.txt', name='Orders.txt', size=69, modificationTime=1740147756000),
 FileInfo(path='dbfs:/FileStore/shared_uploads/kpavankumar335@gmail.com/outbound/Sales.txt', name='Sales.txt', size=47, modificationTime=1740147756000),
 FileInfo(path='dbfs:/FileStore/shared_uploads/kpavankumar335@gmail.com/outbound/Welcome.txt', name='Welcome.txt', size=0, modificationTime=1740147756000)]
```

4. Move a file between directories.

Source – Inbound

Target – srcfiles

#Syntax – dbutils.fs.mv('source path', 'target path')

```
Just now (<1s)   10
dbutils.fs.ls('/FileStore/shared_uploads/kpavankumar335@gmail.com/srcfiles')

Out[13]: []
```

```
Just now (1s)   11
#copy files between inbound and outbound folders
#Syntax: dbutils.fs.mv(source path, target path)
#mv --> It is a command used to move file between source and target.

dbutils.fs.mv(
    '/FileStore/shared_uploads/kpavankumar335@gmail.com/inbound/Customers.txt',
    '/FileStore/shared_uploads/kpavankumar335@gmail.com/srcfiles'
)

Out[14]: True
```

```
Just now (<1s)   12   Python
dbutils.fs.ls('/FileStore/shared_uploads/kpavankumar335@gmail.com/inbound')

Out[15]: [FileInfo(path='dbfs:/FileStore/shared_uploads/kpavankumar335@gmail.com/inbound/Orders.txt', name='Orders.txt', size=69, modificationTime=1740145943000),
 FileInfo(path='dbfs:/FileStore/shared_uploads/kpavankumar335@gmail.com/inbound/Sales.txt', name='Sales.txt', size=47, modificationTime=1740145943000),
 FileInfo(path='dbfs:/FileStore/shared_uploads/kpavankumar335@gmail.com/inbound/Welcome.txt', name='Welcome.txt', size=0, modificationTime=1740145944000)]
```

```
Just now (<1s)   13   Python
dbutils.fs.ls('/FileStore/shared_uploads/kpavankumar335@gmail.com/srcfiles')

Out[16]: [FileInfo(path='dbfs:/FileStore/shared_uploads/kpavankumar335@gmail.com/srcfiles/Customers.txt', name='Customers.txt', size=61, modificationTime=1740148080000)]
```

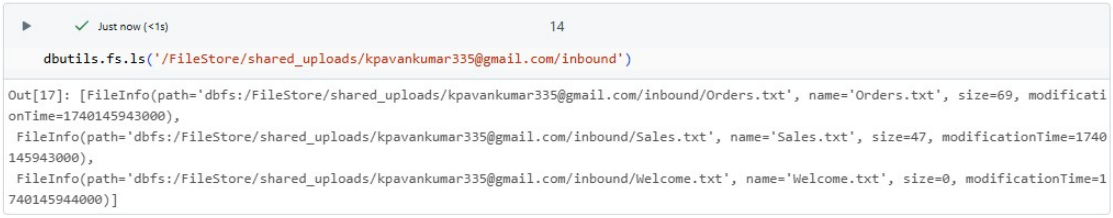
Case:

If we want to move all files from source to target, we can use recursive argument.

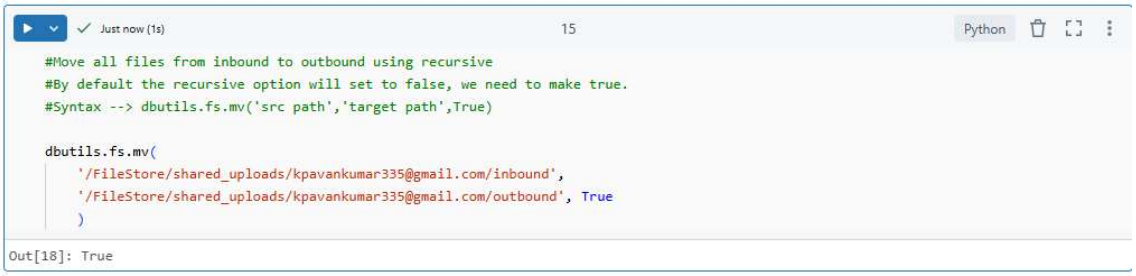
By default, the recursive option will be false. We just mention it as True.

But, be aware that after moving all the files the source directory will be deleted.

List of files in inbound folder before:

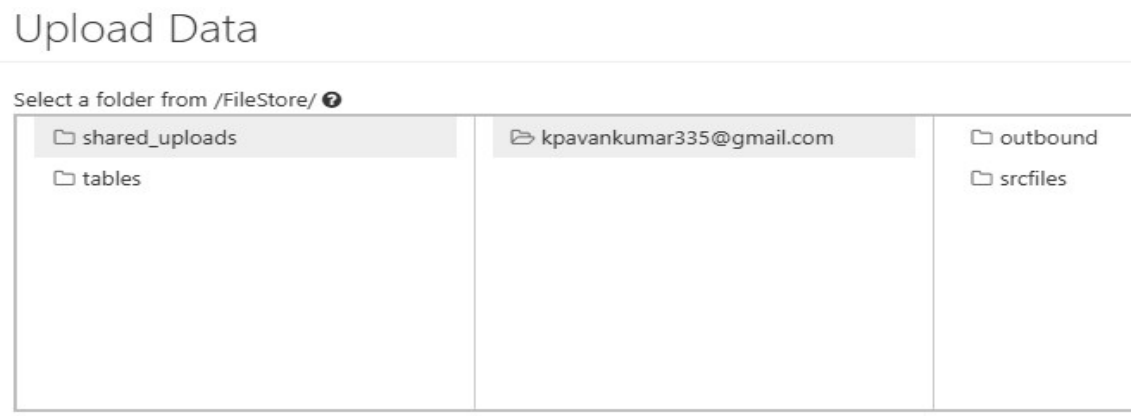


Execution: From inbound to outbound

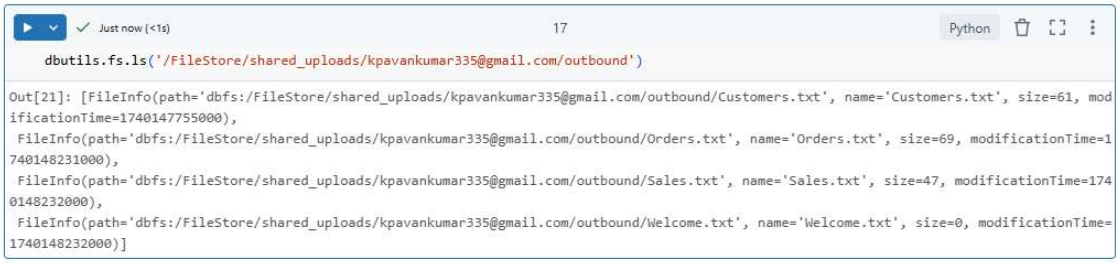


List of files in inbound folder post execution:

We can observe that, directory got deleted. There is no inbound folder.



Files in outbound folder:



```
dbutils.fs.ls('/FileStore/shared_uploads/kpavankumar335@gmail.com/outbound')
```

```
Out[21]: [FileInfo(path='dbfs:/FileStore/shared_uploads/kpavankumar335@gmail.com/outbound/Customers.txt', name='Customers.txt', size=61, modificationTime=1740147755000),
 FileInfo(path='dbfs:/FileStore/shared_uploads/kpavankumar335@gmail.com/outbound/Orders.txt', name='Orders.txt', size=69, modificationTime=1740148231000),
 FileInfo(path='dbfs:/FileStore/shared_uploads/kpavankumar335@gmail.com/outbound/Sales.txt', name='Sales.txt', size=47, modificationTime=1740148232000),
 FileInfo(path='dbfs:/FileStore/shared_uploads/kpavankumar335@gmail.com/outbound/Welcome.txt', name='Welcome.txt', size=0, modificationTime=1740148232000)]
```

5. **Head- used to display the content of the file**

#Syntax- dbutils.fs.head('file path')

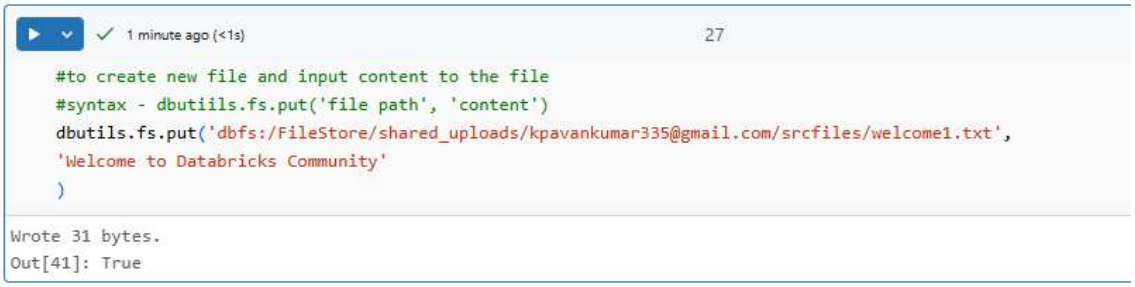
```
#to display the content of the file.
#syntax - dbutils.fs.head('file path)

dbutils.fs.head('/FileStore/shared_uploads/kpavankumar335@gmail.com/outbound/Customers.txt')
```

```
Out[25]: 'Cust_id,Cust_name\r\n1, Srinivas\r\n2, Pavan\r\n3, Phani\r\n4, Naresh'
```

6. **Put – To input content to the file.**

#Syntax – dbutils.fs.put('file path', 'content to be added')



```
#to create new file and input content to the file
#syntax - dbutiils.fs.put('file path', 'content')
dbutils.fs.put('dbfs:/FileStore/shared_uploads/kpavankumar335@gmail.com/srcfiles/welcome1.txt',
'Welcome to Databricks Community'
)
```

```
Wrote 31 bytes.
Out[41]: True
```

We can see below, content of the welcome1.txt file.

```
dbutils.fs.ls('/FileStore/shared_uploads/kpavankumar335@gmail.com/srcfiles')
```

```
Out[42]: [FileInfo(path='dbfs:/FileStore/shared_uploads/kpavankumar335@gmail.com/srcfiles/Customers.txt', name='Customers.txt', size=61, modificationTime=1740148080000),
 FileInfo(path='dbfs:/FileStore/shared_uploads/kpavankumar335@gmail.com/srcfiles/welcome.txt', name='welcome.txt', size=0, modificationTime=1740149612000),
 FileInfo(path='dbfs:/FileStore/shared_uploads/kpavankumar335@gmail.com/srcfiles/welcome1.txt', name='welcome1.txt', size=31, modificationTime=1740149710000)]
```

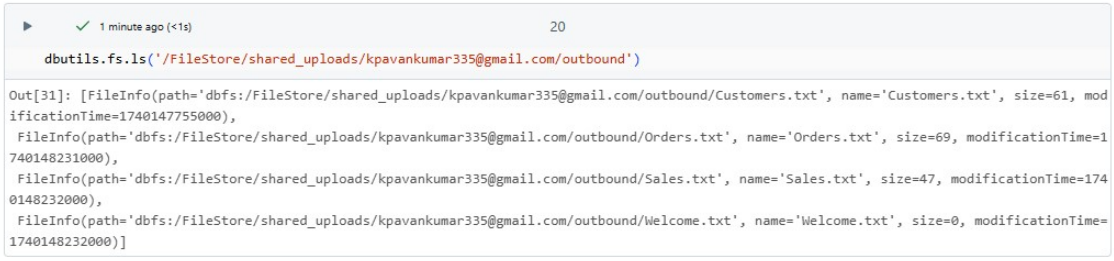
```
dbutils.fs.head('dbfs:/FileStore/shared_uploads/kpavankumar335@gmail.com/srcfiles/welcome1.txt')
```

```
Out[43]: 'Welcome to Databricks Community'
```

7. **To delete the files in a directory.**

syntax - dbutils.fs.rm('filepath')

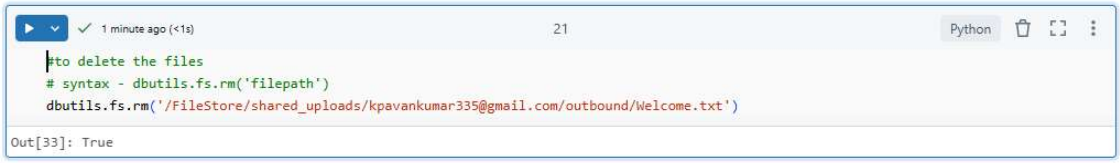
Before execution:



```
1 minute ago (<1s)  20
dbutils.fs.ls('/FileStore/shared_uploads/kpavankumar335@gmail.com/outbound')

Out[31]: [FileInfo(path='dbfs:/FileStore/shared_uploads/kpavankumar335@gmail.com/outbound/Customers.txt', name='Customers.txt', size=61, modificationTime=1740147755000),
 FileInfo(path='dbfs:/FileStore/shared_uploads/kpavankumar335@gmail.com/outbound/Orders.txt', name='Orders.txt', size=69, modificationTime=1740148231000),
 FileInfo(path='dbfs:/FileStore/shared_uploads/kpavankumar335@gmail.com/outbound/Sales.txt', name='Sales.txt', size=47, modificationTime=1740148232000),
 FileInfo(path='dbfs:/FileStore/shared_uploads/kpavankumar335@gmail.com/outbound/Welcome.txt', name='Welcome.txt', size=0, modificationTime=1740148232000)]
```

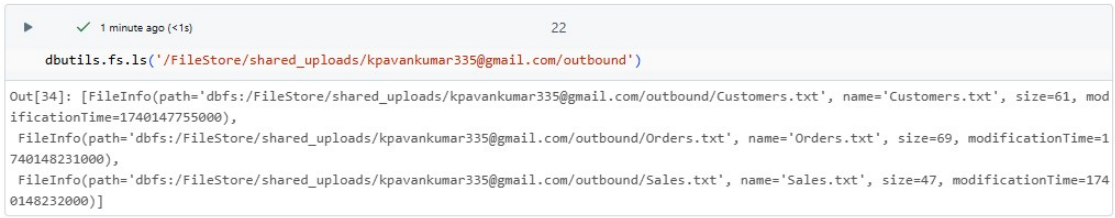
Execution:



```
1 minute ago (<1s)  21  Python
#to delete the files
# syntax - dbutils.fs.rm('filepath')
dbutils.fs.rm('/FileStore/shared_uploads/kpavankumar335@gmail.com/outbound/Welcome.txt')

Out[33]: True
```

Post Execution:



```
1 minute ago (<1s)  22
dbutils.fs.ls('/FileStore/shared_uploads/kpavankumar335@gmail.com/outbound')

Out[34]: [FileInfo(path='dbfs:/FileStore/shared_uploads/kpavankumar335@gmail.com/outbound/Customers.txt', name='Customers.txt', size=61, modificationTime=1740147755000),
 FileInfo(path='dbfs:/FileStore/shared_uploads/kpavankumar335@gmail.com/outbound/Orders.txt', name='Orders.txt', size=69, modificationTime=1740148231000),
 FileInfo(path='dbfs:/FileStore/shared_uploads/kpavankumar335@gmail.com/outbound/Sales.txt', name='Sales.txt', size=47, modificationTime=1740148232000)]
```

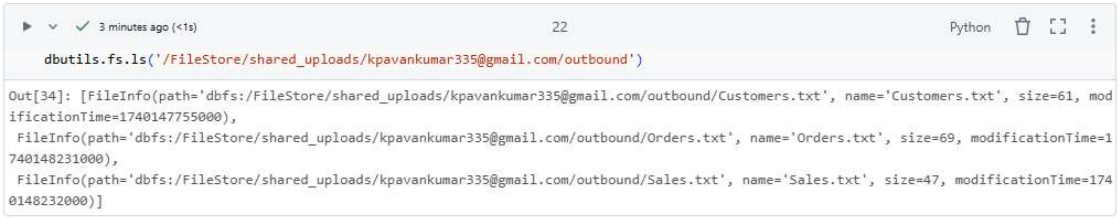
Case:

If we want to delete all files from directory, we can use recursive argument.

By default, the recursive option will be false. We just mention it as True.

But, be sure to check while deleting in production environment.

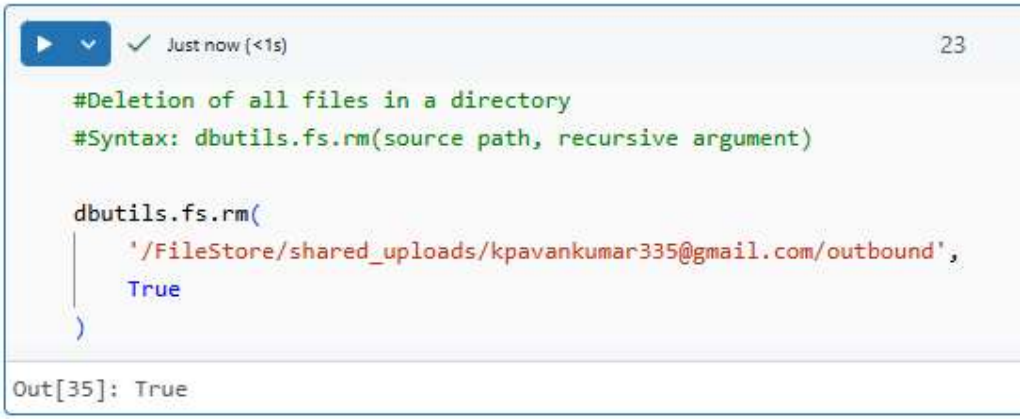
List of files in outbound folder before:



```
3 minutes ago (<1s)  22  Python
dbutils.fs.ls('/FileStore/shared_uploads/kpavankumar335@gmail.com/outbound')

Out[34]: [FileInfo(path='dbfs:/FileStore/shared_uploads/kpavankumar335@gmail.com/outbound/Customers.txt', name='Customers.txt', size=61, modificationTime=1740147755000),
 FileInfo(path='dbfs:/FileStore/shared_uploads/kpavankumar335@gmail.com/outbound/Orders.txt', name='Orders.txt', size=69, modificationTime=1740148231000),
 FileInfo(path='dbfs:/FileStore/shared_uploads/kpavankumar335@gmail.com/outbound/Sales.txt', name='Sales.txt', size=47, modificationTime=1740148232000)]
```

Execution step:



Post deletion, all the files were deleted including the directory as well.