Custom Learnings

Day 20

Databricks

Structured Streaming: Modes:

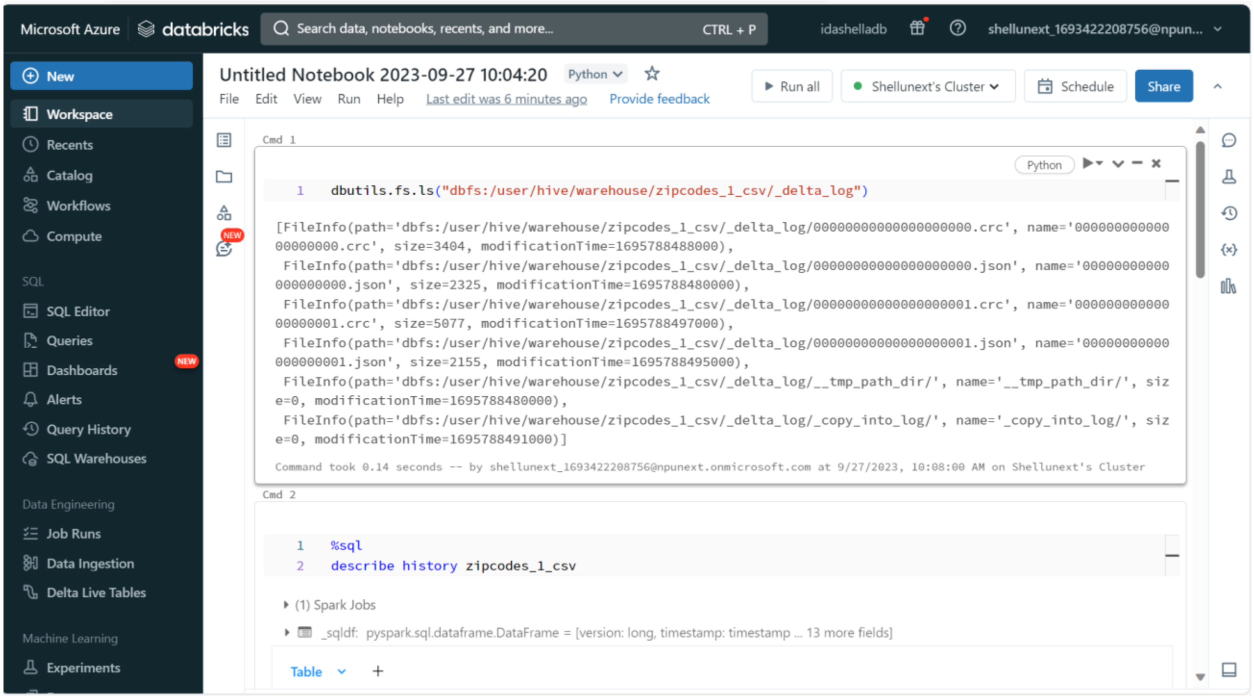
1. Append: Optional when not using aggregation.
2. Complete: Mandatory in case of aggregation.
3. Update: On the go will check for the updating if required it will do.

This involves more cost than ADF.

Parquet is columnar format which goes well with big data. But it doesn’t support deletion operation.

Delta Table: Whenever we create a table in databricks, delta table is created.

We can do time-travelling in delta table. We can either mimic the table for that time period or rollback to that particular version. Inside the delta.log folder, the json file will store these versions. Vacuum command is used to clean the versions in periodic intervals.

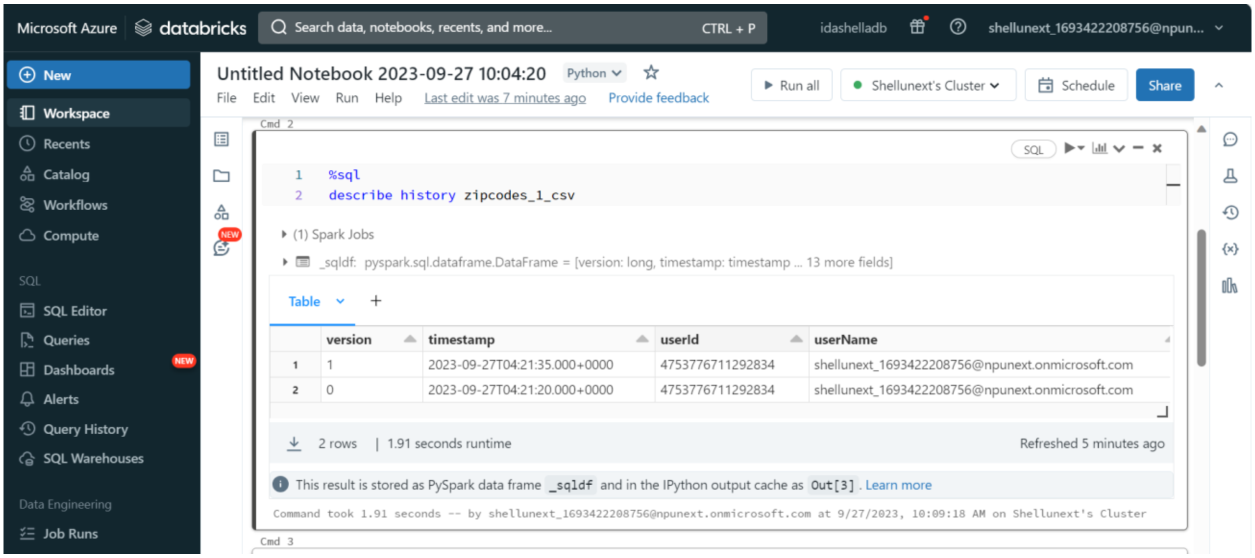


```
dbutils.fs.ls("dbfs:/user/hive/warehouse/zipcodes_1_csv/_delta_log")
```

```
[FileInfo(path='dbfs:/user/hive/warehouse/zipcodes_1_csv/_delta_log/00000000000000000000.crc', name='000000000000
00000000.crc', size=3404, modificationTime=1695788488000),
 FileInfo(path='dbfs:/user/hive/warehouse/zipcodes_1_csv/_delta_log/00000000000000000000.json', name='00000000000
000000000.json', size=2325, modificationTime=1695788480000),
 FileInfo(path='dbfs:/user/hive/warehouse/zipcodes_1_csv/_delta_log/00000000000000000001.crc', name='000000000000
00000001.crc', size=5077, modificationTime=1695788497000),
 FileInfo(path='dbfs:/user/hive/warehouse/zipcodes_1_csv/_delta_log/00000000000000000001.json', name='00000000000
000000001.json', size=2155, modificationTime=1695788495000),
 FileInfo(path='dbfs:/user/hive/warehouse/zipcodes_1_csv/_delta_log/__tmp_path_dir/', name='__tmp_path_dir/', siz
e=0, modificationTime=1695788480000),
 FileInfo(path='dbfs:/user/hive/warehouse/zipcodes_1_csv/_delta_log/_copy_into_log/', name='_copy_into_log/', siz
e=0, modificationTime=1695788491000)]
```

Command took 0.14 seconds -- by shellunext_1693422208756@npunext.onmicrosoft.com at 9/27/2023, 10:08:00 AM on Shellunext's Cluster

```
%sql
describe history zipcodes_1_csv
```



```
%sql
describe history zipcodes_1_csv
```

(1) Spark Jobs

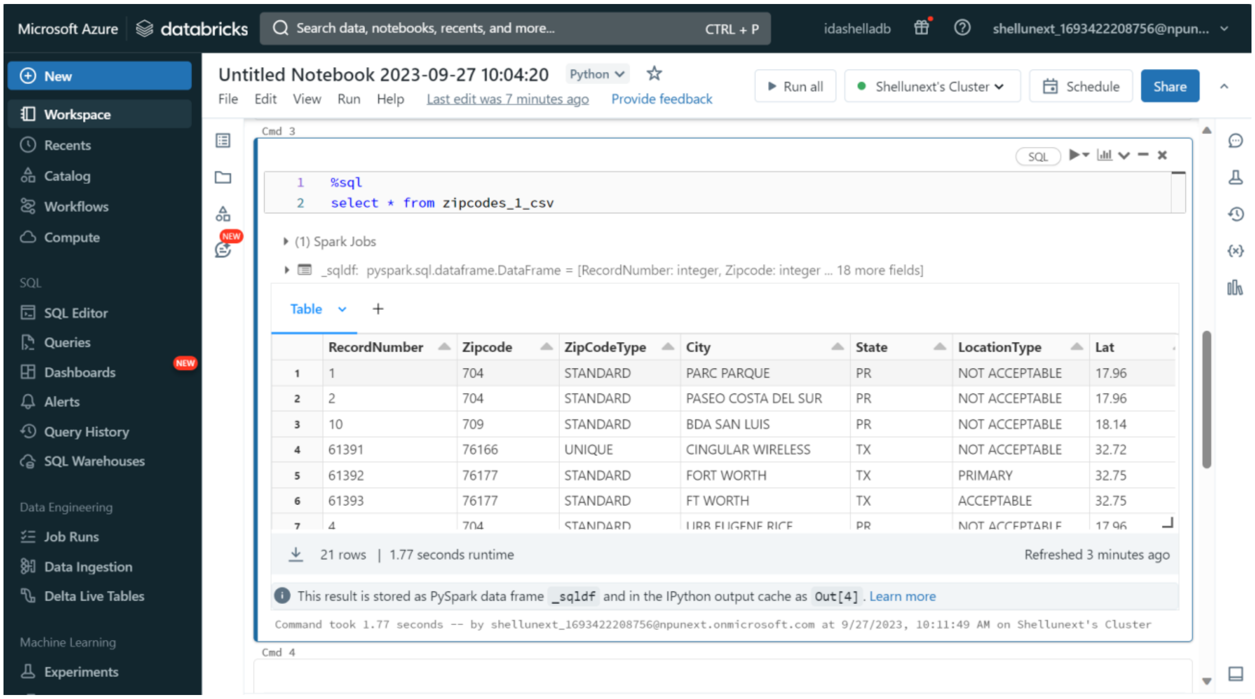
_sqldf: pyspark.sql.dataframe.DataFrame = [version: long, timestamp: timestamp ... 13 more fields]

	version	timestamp	userId	userName
1	1	2023-09-27T04:21:35.000+0000	4753776711292834	shellunext_1693422208756@npunext.onmicrosoft.com
2	0	2023-09-27T04:21:20.000+0000	4753776711292834	shellunext_1693422208756@npunext.onmicrosoft.com

2 rows | 1.91 seconds runtime Refreshed 5 minutes ago

This result is stored as PySpark data frame `_sqldf` and in the IPython output cache as `Out[3]`. Learn more

```
Command took 1.91 seconds -- by shellunext_1693422208756@npunext.onmicrosoft.com at 9/27/2023, 10:09:18 AM on Shellunext's Cluster
```



```
%sql
select * from zipcodes_1_csv
```

(1) Spark Jobs

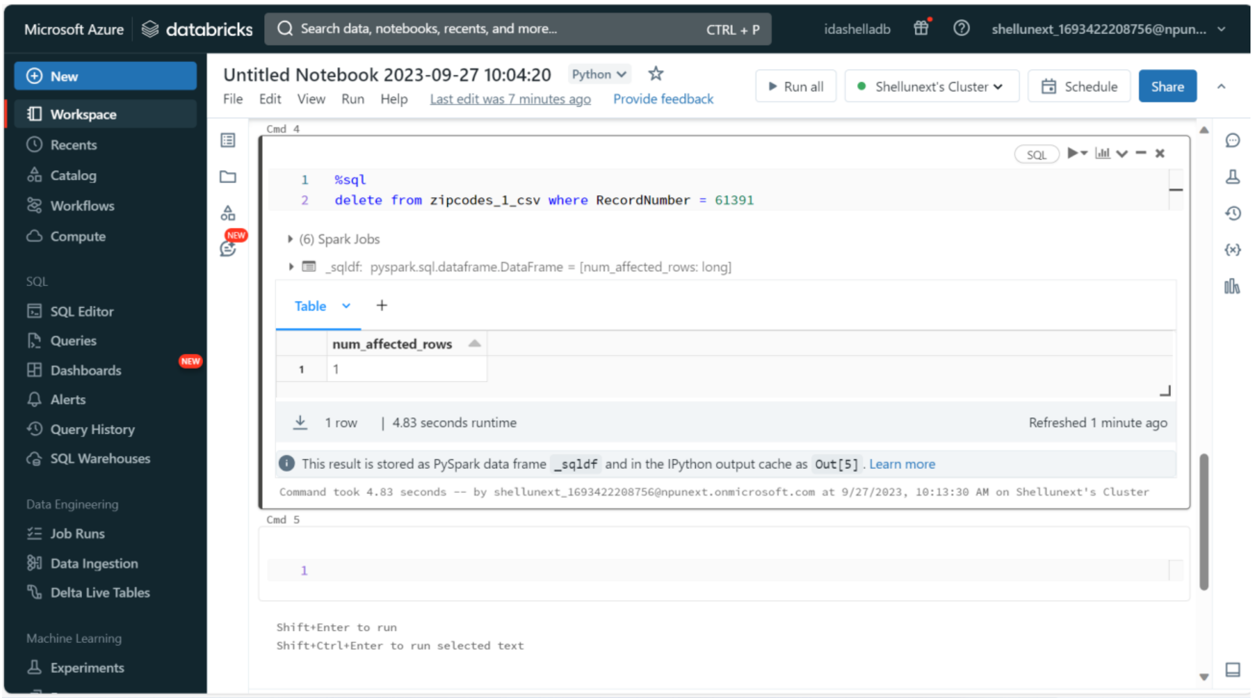
_sqldf: pyspark.sql.dataframe.DataFrame = [RecordNumber: integer, Zipcode: integer ... 18 more fields]

	RecordNumber	Zipcode	ZipCodeType	City	State	LocationType	Lat
1	1	704	STANDARD	PARC PARQUE	PR	NOT ACCEPTABLE	17.96
2	2	704	STANDARD	PASEO COSTA DEL SUR	PR	NOT ACCEPTABLE	17.96
3	10	709	STANDARD	BDA SAN LUIS	PR	NOT ACCEPTABLE	18.14
4	61391	76166	UNIQUE	CINGULAR WIRELESS	TX	NOT ACCEPTABLE	32.72
5	61392	76177	STANDARD	FORT WORTH	TX	PRIMARY	32.75
6	61393	76177	STANDARD	FT WORTH	TX	ACCEPTABLE	32.75
7	4	704	STANDARD	URB EUGENE RICE	PR	NOT ACCEPTABLE	17.96

21 rows | 1.77 seconds runtime Refreshed 3 minutes ago

This result is stored as PySpark data frame `_sqldf` and in the IPython output cache as `Out[4]`. Learn more

```
Command took 1.77 seconds -- by shellunext_1693422208756@npunext.onmicrosoft.com at 9/27/2023, 10:11:49 AM on Shellunext's Cluster
```



```
%sql
delete from zipcodes_1_csv where RecordNumber = 61391
```

(6) Spark Jobs

_sqldf: pyspark.sql.dataframe.DataFrame = [num_affected_rows: long]

	num_affected_rows
1	1

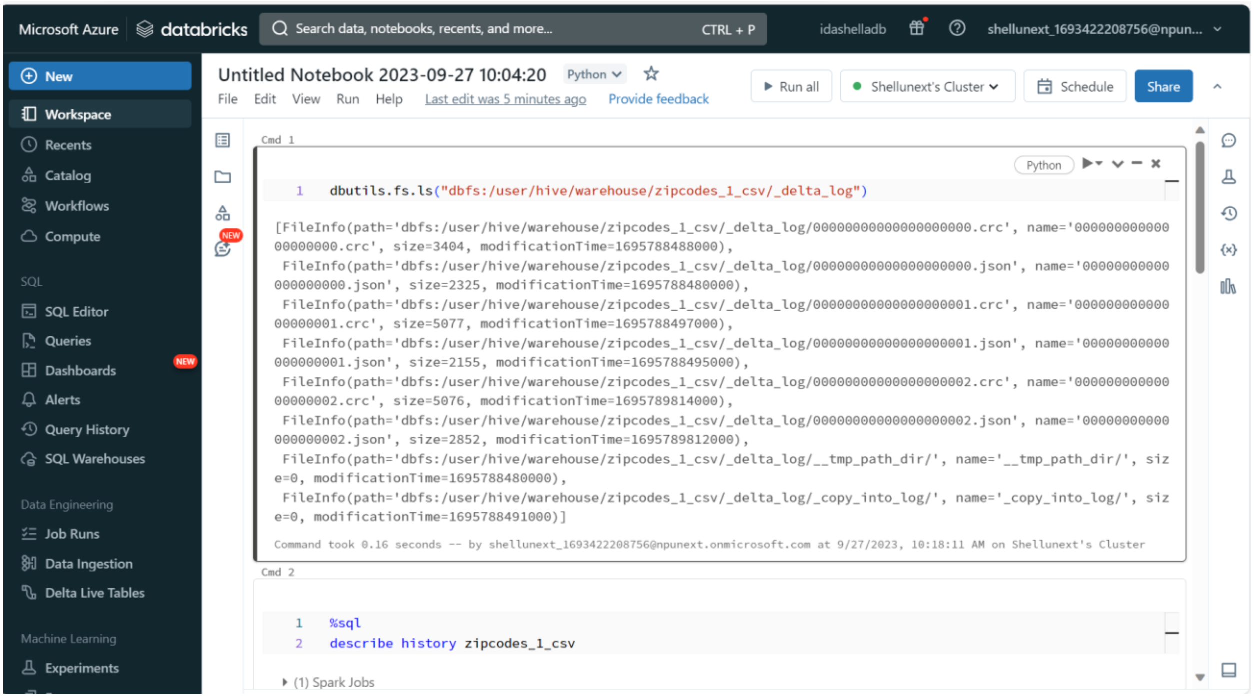
1 row | 4.83 seconds runtime   Refreshed 1 minute ago

This result is stored as PySpark data frame `_sqldf` and in the IPython output cache as `Out[5]`. Learn more

```
Command took 4.83 seconds -- by shellunext_1693422208756@npunext.onmicrosoft.com at 9/27/2023, 10:13:30 AM on Shellunext's Cluster
```

```
Shift+Enter to run
Shift+Ctrl+Enter to run selected text
```

After deleting:



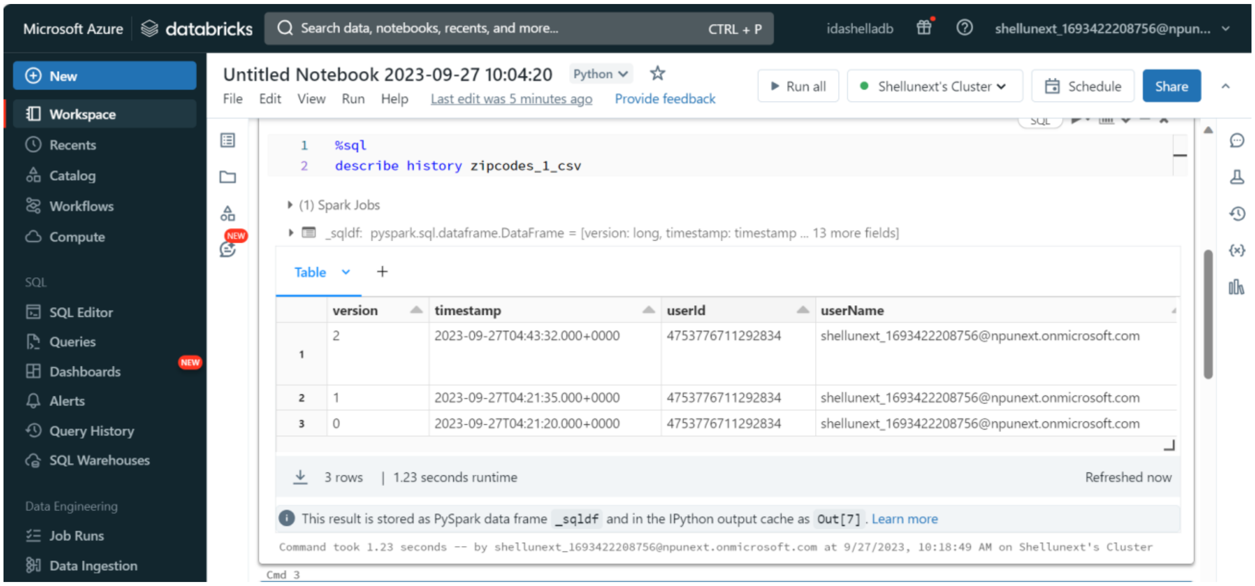
```
dbutils.fs.ls("dbfs:/user/hive/warehouse/zipcodes_1_csv/_delta_log")
```

```
[FileInfo(path='dbfs:/user/hive/warehouse/zipcodes_1_csv/_delta_log/00000000000000000000.crc', name='000000000000
00000000.crc', size=3404, modificationTime=1695788488000),
 FileInfo(path='dbfs:/user/hive/warehouse/zipcodes_1_csv/_delta_log/00000000000000000000.json', name='00000000000
000000000.json', size=2325, modificationTime=1695788480000),
 FileInfo(path='dbfs:/user/hive/warehouse/zipcodes_1_csv/_delta_log/00000000000000000001.crc', name='000000000000
00000001.crc', size=5077, modificationTime=1695788497000),
 FileInfo(path='dbfs:/user/hive/warehouse/zipcodes_1_csv/_delta_log/00000000000000000001.json', name='00000000000
000000001.json', size=2155, modificationTime=1695788495000),
 FileInfo(path='dbfs:/user/hive/warehouse/zipcodes_1_csv/_delta_log/00000000000000000002.crc', name='000000000000
00000002.crc', size=5076, modificationTime=1695789814000),
 FileInfo(path='dbfs:/user/hive/warehouse/zipcodes_1_csv/_delta_log/00000000000000000002.json', name='00000000000
000000002.json', size=2852, modificationTime=1695789812000),
 FileInfo(path='dbfs:/user/hive/warehouse/zipcodes_1_csv/_delta_log/__tmp_path_dir/', name='__tmp_path_dir/', siz
e=0, modificationTime=1695788480000),
 FileInfo(path='dbfs:/user/hive/warehouse/zipcodes_1_csv/_delta_log/_copy_into_log/', name='_copy_into_log/', siz
e=0, modificationTime=1695788491000)]
```

```
Command took 0.16 seconds -- by shellunext_1693422208756@npunext.onmicrosoft.com at 9/27/2023, 10:18:11 AM on Shellunext's Cluster
```

```
%sql
describe history zipcodes_1_csv
```

(1) Spark Jobs



```
%sql
describe history zipcodes_1_csv
```

(1) Spark Jobs

_sqldf: pyspark.sql.dataframe.DataFrame = [version: long, timestamp: timestamp ... 13 more fields]

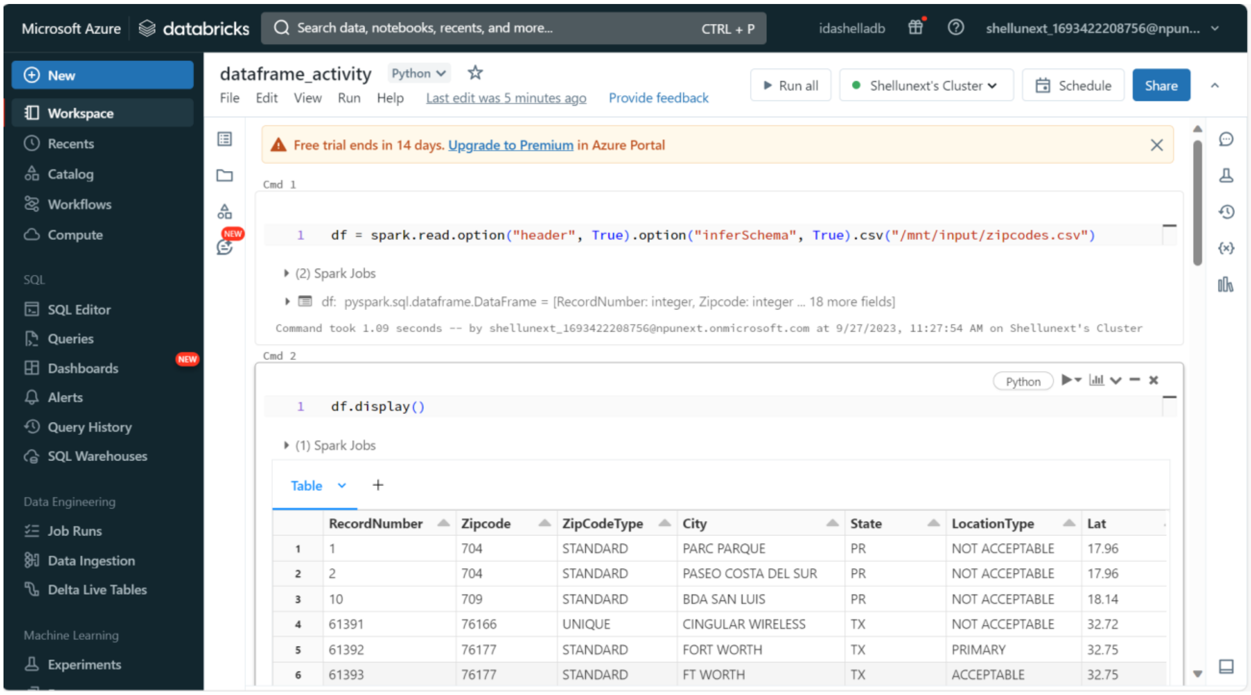
	version	timestamp	userId	userName
1	2	2023-09-27T04:43:32.000+0000	4753776711292834	shellunext_1693422208756@npunext.onmicrosoft.com
2	1	2023-09-27T04:21:35.000+0000	4753776711292834	shellunext_1693422208756@npunext.onmicrosoft.com
3	0	2023-09-27T04:21:20.000+0000	4753776711292834	shellunext_1693422208756@npunext.onmicrosoft.com

3 rows | 1.23 seconds runtime

This result is stored as PySpark data frame `_sqldf` and in the IPython output cache as `Out[7]`.

```
Command took 1.23 seconds -- by shellunext_1693422208756@npunext.onmicrosoft.com at 9/27/2023, 10:18:49 AM on Shellunext's Cluster
```

Dataframe Activity:



Free trial ends in 14 days. Upgrade to Premium in Azure Portal

```
df = spark.read.option("header", True).option("inferSchema", True).csv("/mnt/input/zipcodes.csv")
```

(2) Spark Jobs

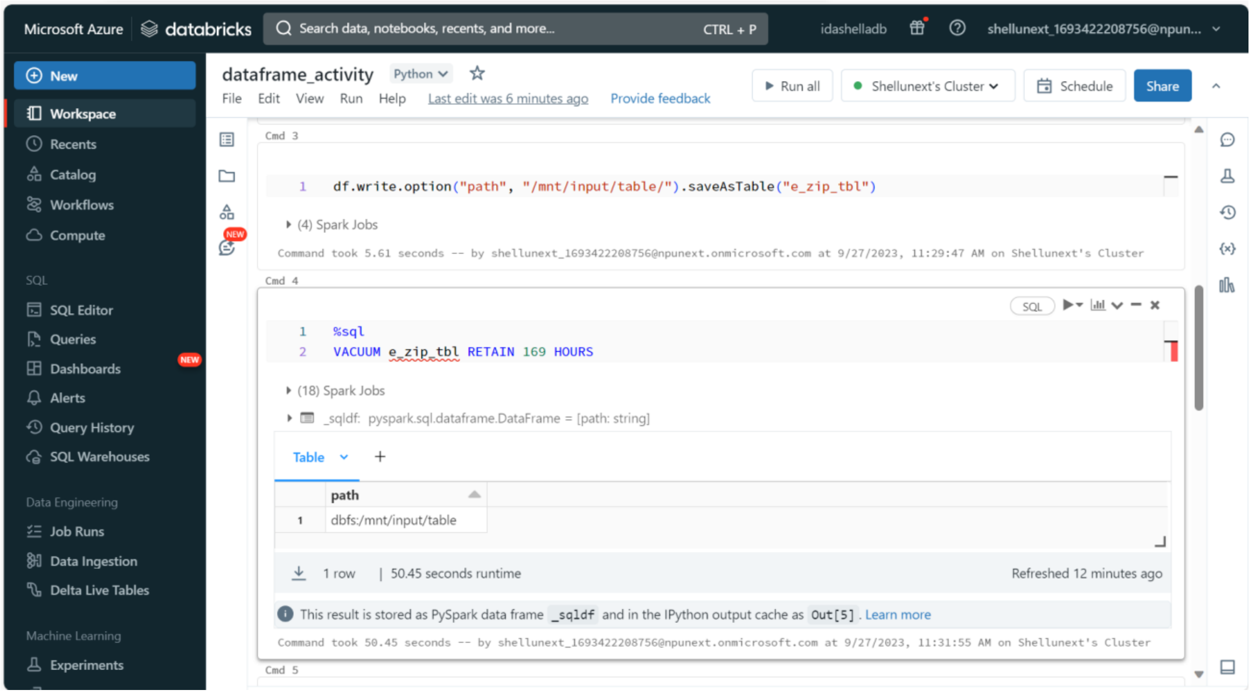
df: pyspark.sql.dataframe.DataFrame = [RecordNumber: integer, Zipcode: integer ... 18 more fields]

```
Command took 1.09 seconds -- by shellunext_1693422208756@npunext.onmicrosoft.com at 9/27/2023, 11:27:54 AM on Shellunext's Cluster
```

```
df.display()
```

(1) Spark Jobs

	RecordNumber	Zipcode	ZipCodeType	City	State	LocationType	Lat
1	1	704	STANDARD	PARC PARQUE	PR	NOT ACCEPTABLE	17.96
2	2	704	STANDARD	PASEO COSTA DEL SUR	PR	NOT ACCEPTABLE	17.96
3	10	709	STANDARD	BDA SAN LUIS	PR	NOT ACCEPTABLE	18.14
4	61391	76166	UNIQUE	CINGULAR WIRELESS	TX	NOT ACCEPTABLE	32.72
5	61392	76177	STANDARD	FORT WORTH	TX	PRIMARY	32.75
6	61393	76177	STANDARD	FT WORTH	TX	ACCEPTABLE	32.75



```
df.write.option("path", "/mnt/input/table/").saveAsTable("e_zip_tbl")
```

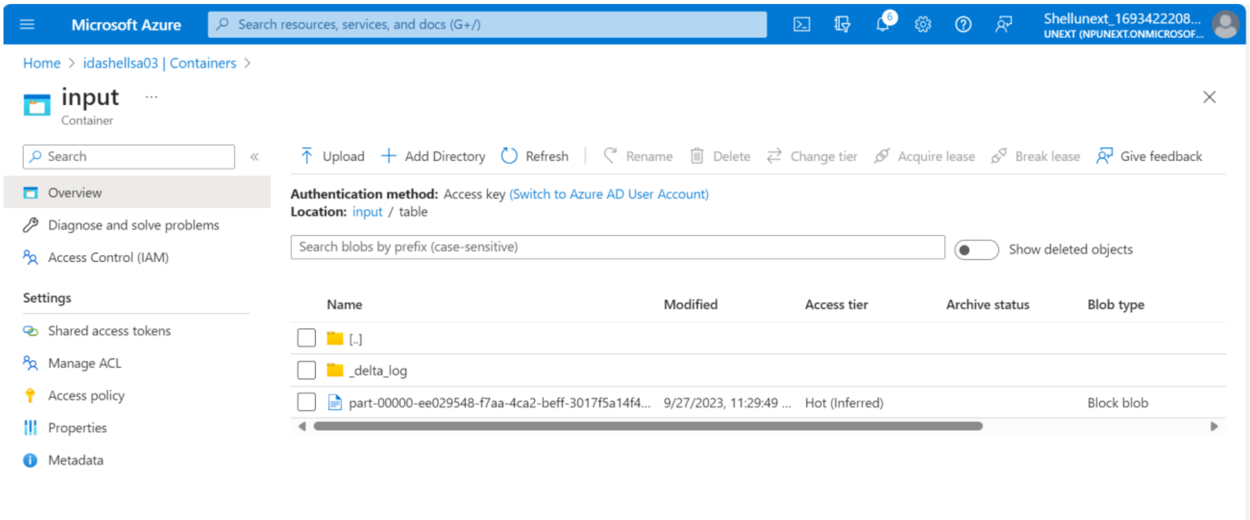
```
▸ (4) Spark Jobs
Command took 5.61 seconds -- by shellunext_1693422208756@npunext.onmicrosoft.com at 9/27/2023, 11:29:47 AM on Shellunext's Cluster
```

```
%sql
VACUUM e_zip_tbl RETAIN 169 HOURS
```

```
▸ (18) Spark Jobs
▸ _sqldf: pyspark.sql.dataframe.DataFrame = [path: string]
```

	path
1	dbfs:/mnt/input/table

```
1 row | 50.45 seconds runtime                                    Refreshed 12 minutes ago
This result is stored as PySpark data frame _sqldf and in the IPython output cache as Out[5]. Learn more
Command took 50.45 seconds -- by shellunext_1693422208756@npunext.onmicrosoft.com at 9/27/2023, 11:31:55 AM on Shellunext's Cluster
```

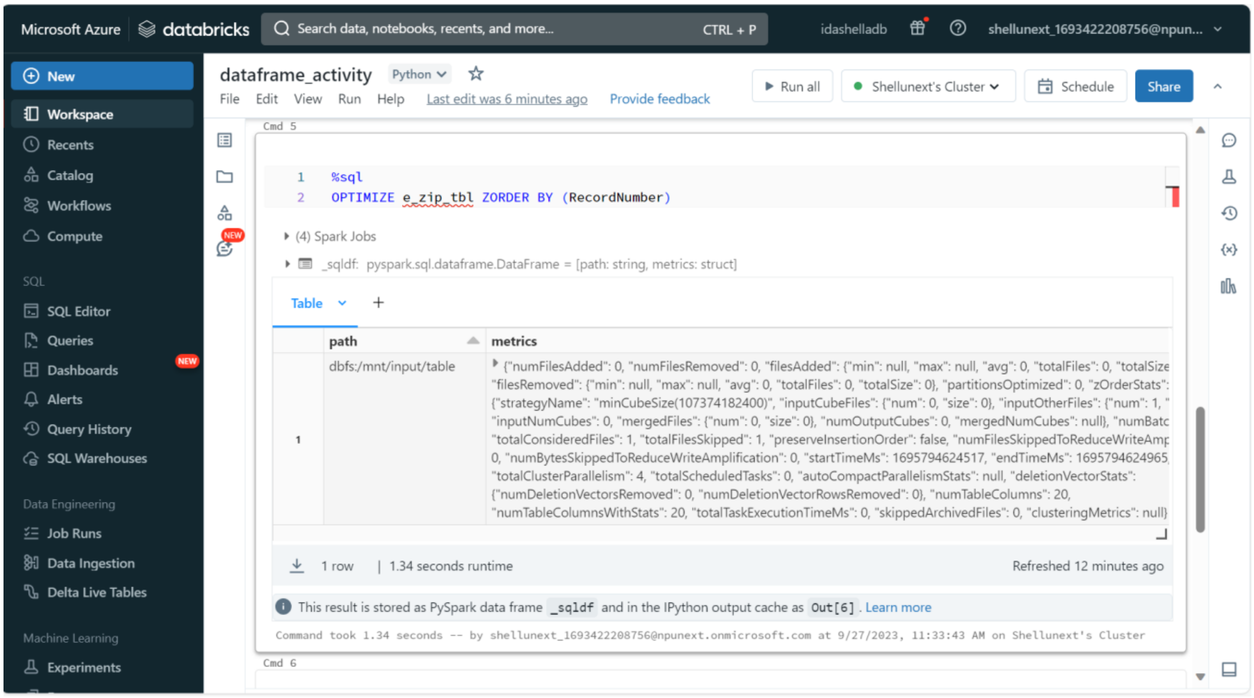


Authentication method: Access key (Switch to Azure AD User Account)
Location: input / table

Search blobs by prefix (case-sensitive)　○ Show deleted objects

	Name	Modified	Access tier	Archive status	Blob type
□	[..]				
□	_delta_log				
□	part-00000-ee029548-f7aa-4ca2-beff-3017f5a14f4...	9/27/2023, 11:29:49 ...	Hot (Inferred)		Block blob

We optimize the partitions so that each partitioned file consists of the same no. of records.



```
%sql
OPTIMIZE e_zip_tbl ZORDER BY (RecordNumber)
```

(4) Spark Jobs

_sqldf: pyspark.sql.dataframe.DataFrame = [path: string, metrics: struct]

Table

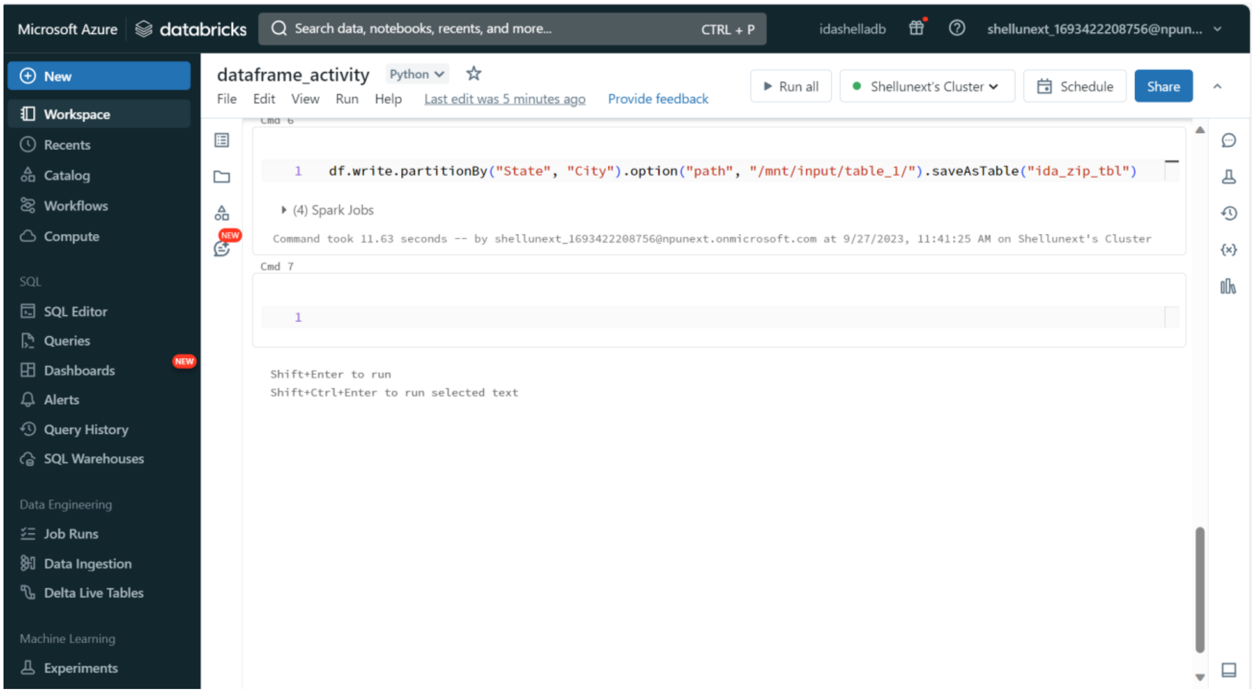
	path	metrics
1	dbfs:/mnt/input/table	{"numFilesAdded": 0, "numFilesRemoved": 0, "filesAdded": {"min": null, "max": null, "avg": 0, "totalFiles": 0, "totalSize "filesRemoved": {"min": null, "max": null, "avg": 0, "totalFiles": 0, "totalSize": 0}, "partitionsOptimized": 0, "zOrderStats": {"strategyName": "minCubeSize(107374182400)", "inputCubeFiles": {"num": 0, "size": 0}, "inputOtherFiles": {"num": 1, " "inputNumCubes": 0, "mergedFiles": {"num": 0, "size": 0}, "numOutputCubes": 0, "mergedNumCubes": null}, "numBatc "totalConsideredFiles": 1, "totalFilesSkipped": 1, "preserveInsertionOrder": false, "numFilesSkippedToReduceWriteAmp 0, "numBytesSkippedToReduceWriteAmplification": 0, "startTimeMs": 1695794624517, "endTimeMs": 1695794624965, "totalClusterParallelism": 4, "totalScheduledTasks": 0, "autoCompactParallelismStats": null, "deletionVectorStats": {"numDeletionVectorsRemoved": 0, "numDeletionVectorRowsRemoved": 0}, "numTableColumns": 20, "numTableColumnsWithStats": 20, "totalTaskExecutionTimeMs": 0, "skippedArchivedFiles": 0, "clusteringMetrics": null}

1 row | 1.34 seconds runtime   Refreshed 12 minutes ago

This result is stored as PySpark data frame `_sqldf` and in the IPython output cache as `Out[6]`. Learn more

```
Command took 1.34 seconds -- by shellunext_1693422208756@npunext.onmicrosoft.com at 9/27/2023, 11:33:43 AM on Shellunext's Cluster
```

By Partition creating table:

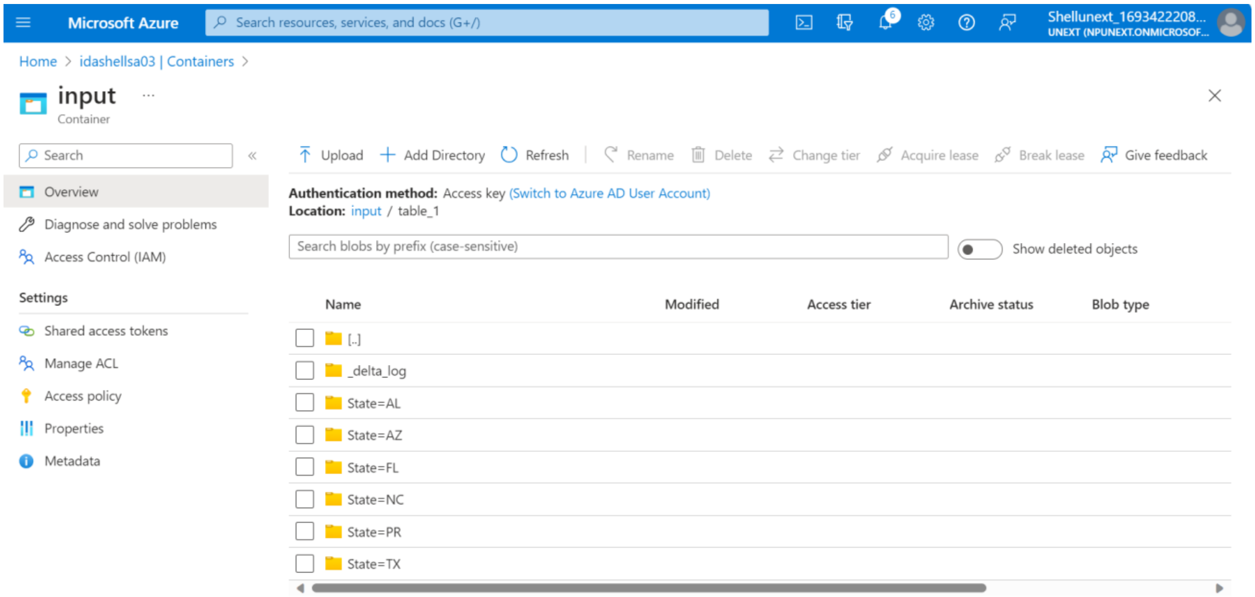


```
df.write.partitionBy("State", "City").option("path", "/mnt/input/table_1/").saveAsTable("ida_zip_tbl")
```

(4) Spark Jobs

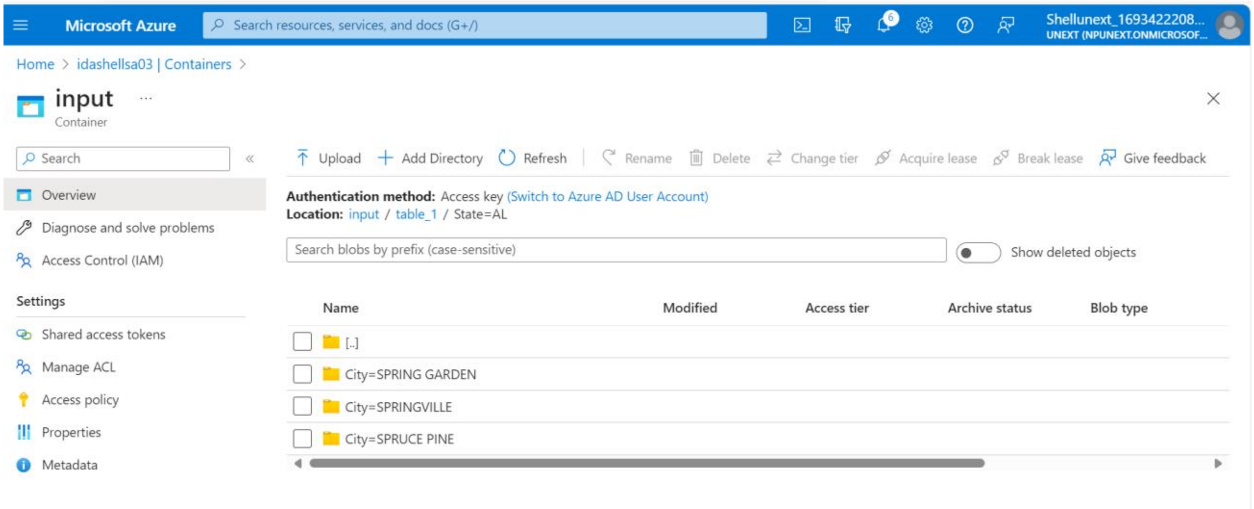
```
Command took 11.63 seconds -- by shellunext_1693422208756@npunext.onmicrosoft.com at 9/27/2023, 11:41:25 AM on Shellunext's Cluster
```

```
Shift+Enter to run
Shift+Ctrl+Enter to run selected text
```



Authentication method: Access key (Switch to Azure AD User Account)
Location: input / table_1

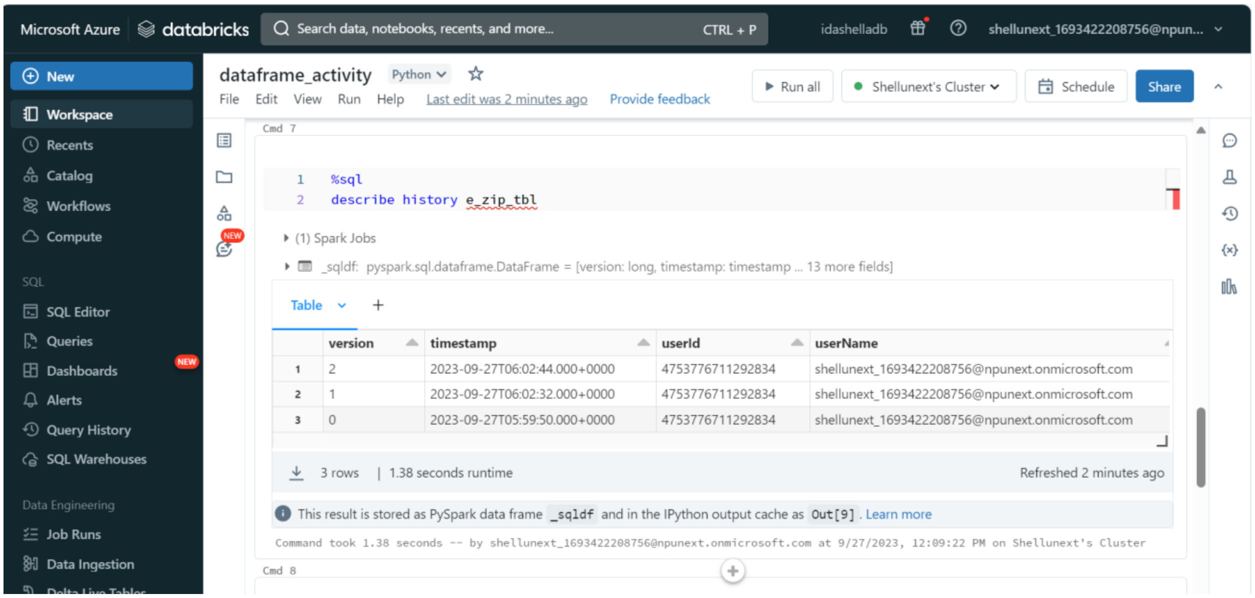
	Name	Modified	Access tier	Archive status	Blob type
□	[..]				
□	_delta_log				
□	State=AL				
□	State=AZ				
□	State=FL				
□	State=NC				
□	State=PR				
□	State=TX				



Authentication method: Access key (Switch to Azure AD User Account)
Location: input / table_1 / State=AL

	Name	Modified	Access tier	Archive status	Blob type
□	[..]				
□	City=SPRING GARDEN				
□	City=SPRINGVILLE				
□	City=SPRUCE PINE				

Restore previous version:



```
%sql
describe history e_zip_tbl
```

(1) Spark Jobs

_sqldf: pyspark.sql.dataframe.DataFrame = [version: long, timestamp: timestamp ... 13 more fields]

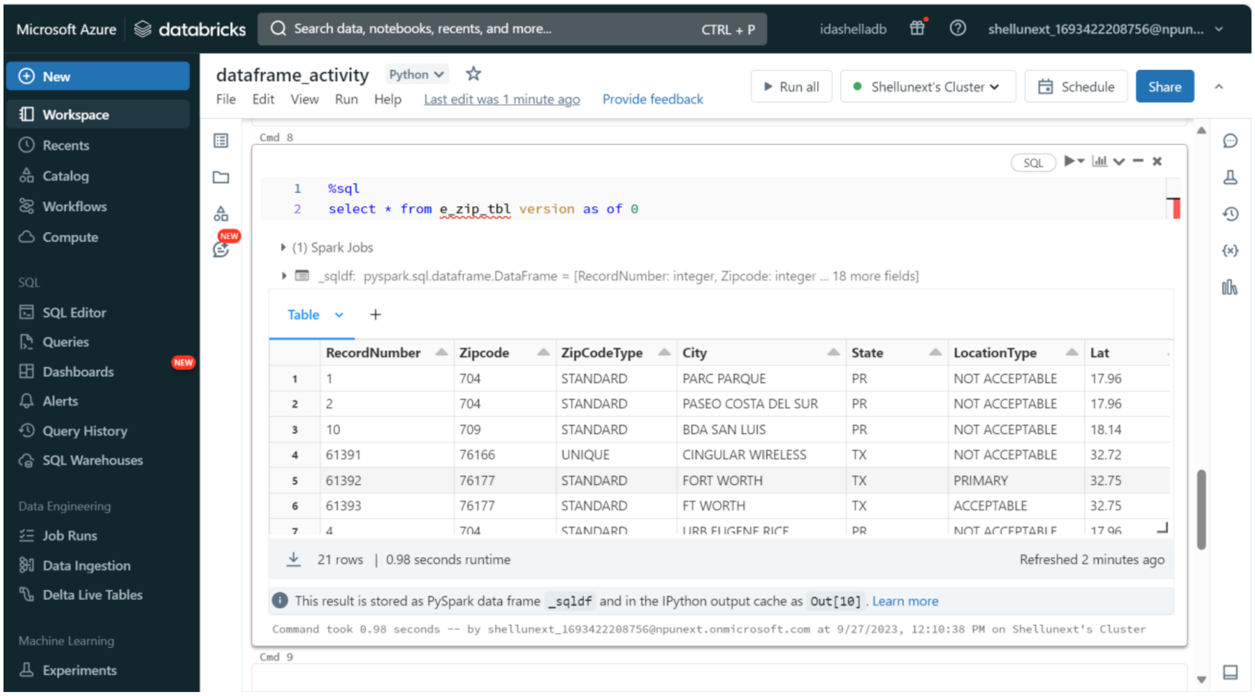
	version	timestamp	userId	userName
1	2	2023-09-27T06:02:44.000+0000	4753776711292834	shellunext_1693422208756@npunext.onmicrosoft.com
2	1	2023-09-27T06:02:32.000+0000	4753776711292834	shellunext_1693422208756@npunext.onmicrosoft.com
3	0	2023-09-27T05:59:50.000+0000	4753776711292834	shellunext_1693422208756@npunext.onmicrosoft.com

3 rows | 1.38 seconds runtime

Refreshed 2 minutes ago

This result is stored as PySpark data frame `_sqldf` and in the IPython output cache as `Out[9]`. Learn more

```
Command took 1.38 seconds -- by shellunext_1693422208756@npunext.onmicrosoft.com at 9/27/2023, 12:09:22 PM on Shellunext's Cluster
```



```
%sql
select * from e_zip_tbl version as of 0
```

(1) Spark Jobs

_sqldf: pyspark.sql.dataframe.DataFrame = [RecordNumber: integer, Zipcode: integer ... 18 more fields]

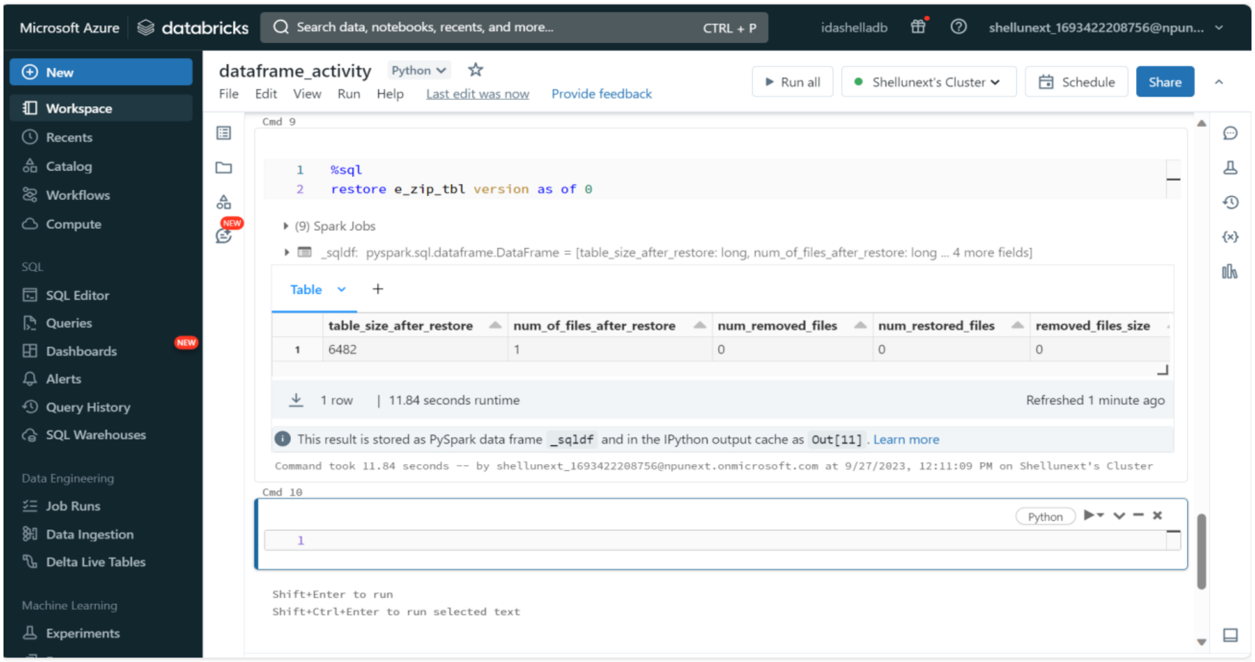
	RecordNumber	Zipcode	ZipCodeType	City	State	LocationType	Lat
1	1	704	STANDARD	PARC PARQUE	PR	NOT ACCEPTABLE	17.96
2	2	704	STANDARD	PASEO COSTA DEL SUR	PR	NOT ACCEPTABLE	17.96
3	10	709	STANDARD	BDA SAN LUIS	PR	NOT ACCEPTABLE	18.14
4	61391	76166	UNIQUE	CINGULAR WIRELESS	TX	NOT ACCEPTABLE	32.72
5	61392	76177	STANDARD	FORT WORTH	TX	PRIMARY	32.75
6	61393	76177	STANDARD	FT WORTH	TX	ACCEPTABLE	32.75
7	4	704	STANDARD	URB EUGENE RICE	PR	NOT ACCEPTABLE	17.96

21 rows | 0.98 seconds runtime

Refreshed 2 minutes ago

This result is stored as PySpark data frame `_sqldf` and in the IPython output cache as `Out[10]`. Learn more

```
Command took 0.98 seconds -- by shellunext_1693422208756@npunext.onmicrosoft.com at 9/27/2023, 12:10:38 PM on Shellunext's Cluster
```



```
%sql
restore e_zip_tbl version as of 0
```

(9) Spark Jobs

_sqldf: pyspark.sql.dataframe.DataFrame = [table_size_after_restore: long, num_of_files_after_restore: long ... 4 more fields]

	table_size_after_restore	num_of_files_after_restore	num_removed_files	num_restored_files	removed_files_size
1	6482	1	0	0	0

1 row | 11.84 seconds runtime Refreshed 1 minute ago

This result is stored as PySpark data frame `_sqldf` and in the IPython output cache as `Out[11]`. Learn more

```
Command took 11.84 seconds -- by shellunext_1693422208756@npunext.onmicrosoft.com at 9/27/2023, 12:11:09 PM on Shellunext's Cluster
```

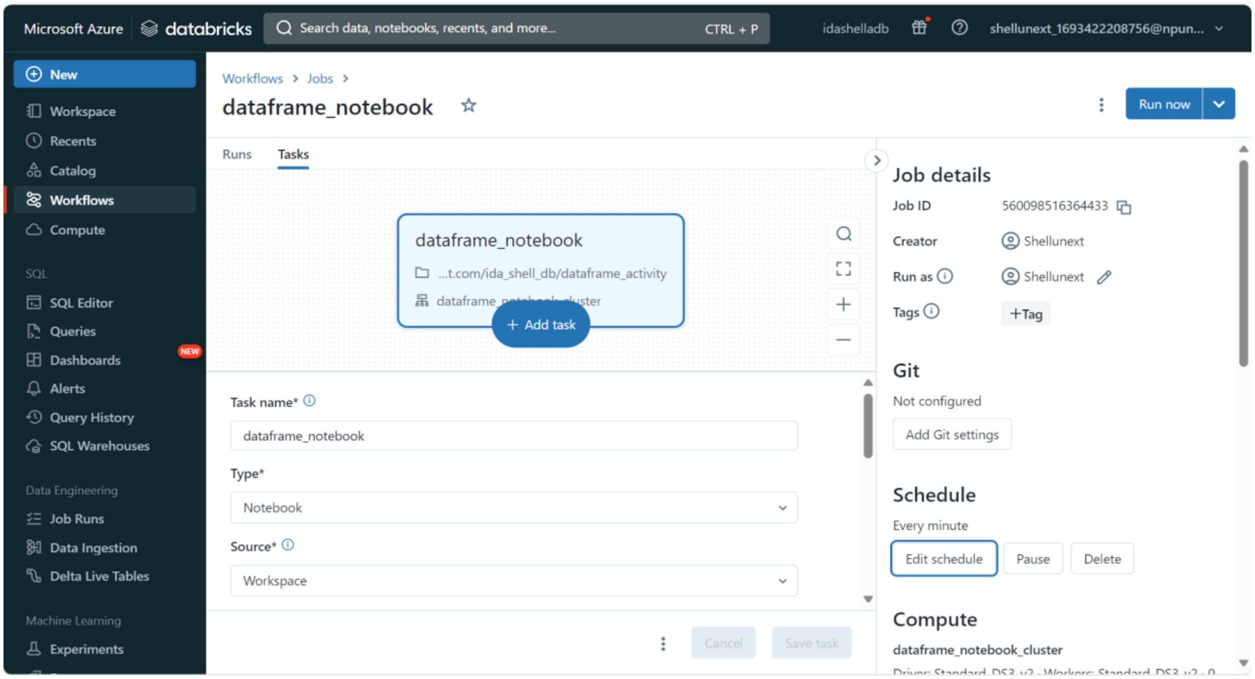
```
Shift+Enter to run
Shift+Ctrl+Enter to run selected text
```

Medallion/ Multi Hop Architecture

We follow three folder architecture:

1. Raw/Bronze: All raw data from all the sources are kept in this folder.
2. Silver: Cleansing and formatting the data files into one particular file format.
3. Gold: Performing some aggregations and joins and keeping the data files needed according to the requirements.

Scheduling the task:



dataframe_notebook

dataframe_notebook
...t.com/ida_shell_db/dataframe_activity
dataframe_notebook_cluster

Task name*: dataframe_notebook

Type*: Notebook

Source*: Workspace

Job details

Job ID: 560098516364433

Creator: Shellunext

Run as: Shellunext

Tags: +Tag

Git

Not configured

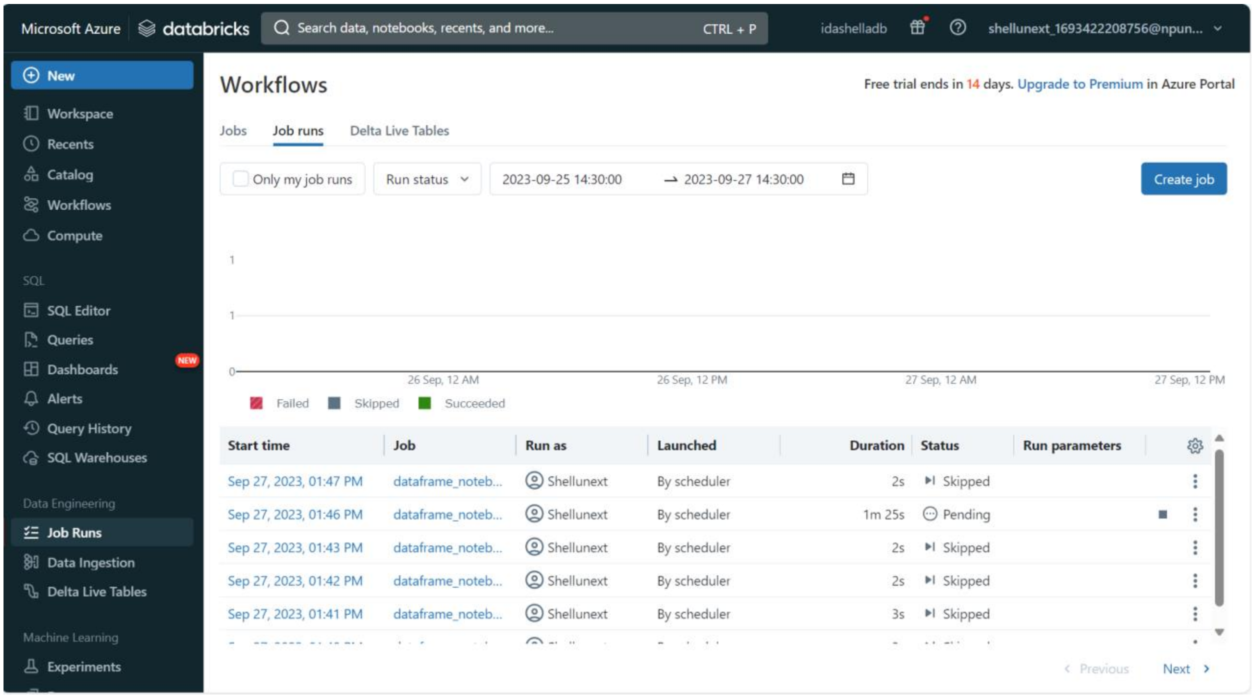
Schedule

Every minute

Compute

dataframe_notebook_cluster

Driver: Standard_DS3_v2 · Workers: Standard_DS3_v2 · 0

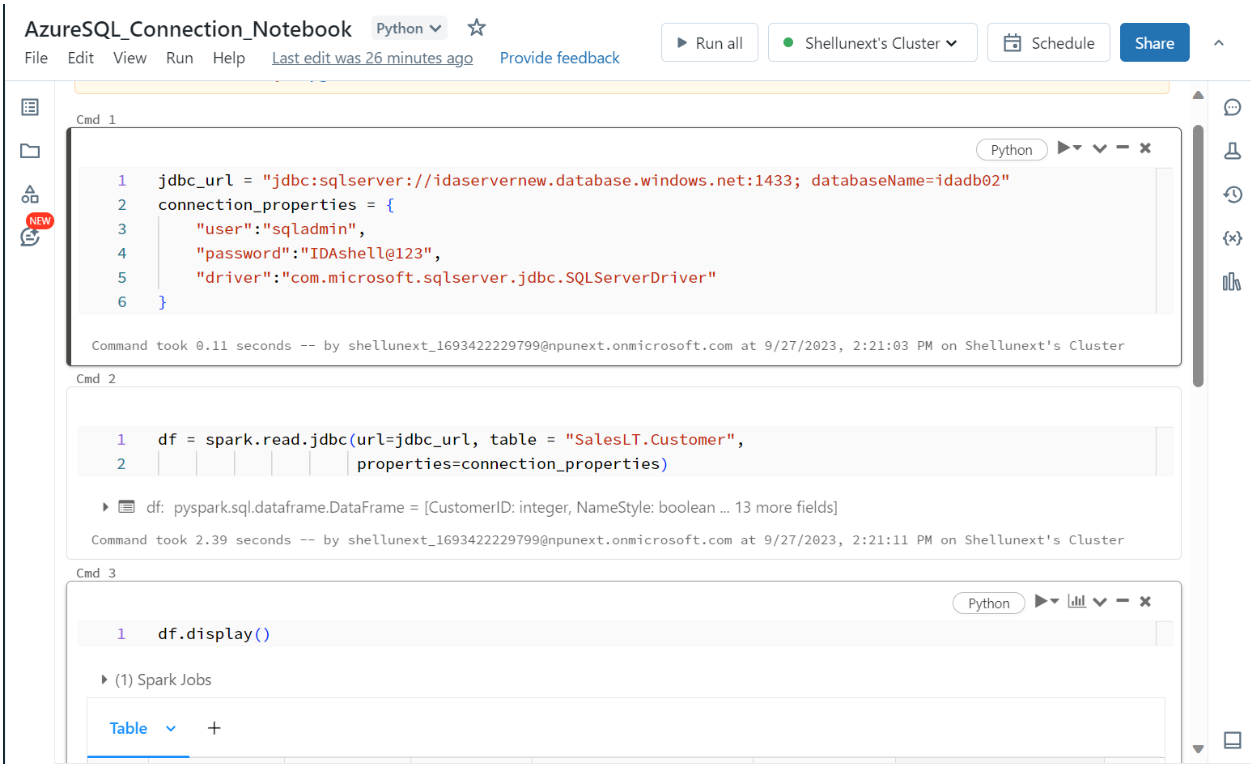


Workflows

Free trial ends in 14 days. Upgrade to Premium in Azure Portal

Start time	Job	Run as	Launched	Duration	Status	Run parameters
Sep 27, 2023, 01:47 PM	dataframe_noteb...	Shellunext	By scheduler	2s	Skipped	
Sep 27, 2023, 01:46 PM	dataframe_noteb...	Shellunext	By scheduler	1m 25s	Pending	
Sep 27, 2023, 01:43 PM	dataframe_noteb...	Shellunext	By scheduler	2s	Skipped	
Sep 27, 2023, 01:42 PM	dataframe_noteb...	Shellunext	By scheduler	2s	Skipped	
Sep 27, 2023, 01:41 PM	dataframe_noteb...	Shellunext	By scheduler	3s	Skipped	

Reading the data from azure sql:



Cmd 1

```python
jdbc_url = "jdbc:sqlserver://idaservernew.database.windows.net:1433; databaseName=idadb02"
connection_properties = {
    "user":"sqladmin",
    "password":"IDAshell@123",
    "driver":"com.microsoft.sqlserver.jdbc.SQLServerDriver"
}
```

```
Command took 0.11 seconds -- by shellunext_1693422229799@npunext.onmicrosoft.com at 9/27/2023, 2:21:03 PM on Shellunext's Cluster
```

Cmd 2

```python
df = spark.read.jdbc(url=jdbc_url, table = "SalesLT.Customer",
                     properties=connection_properties)
```

```
df:  pyspark.sql.dataframe.DataFrame = [CustomerID: integer, NameStyle: boolean ... 13 more fields]
Command took 2.39 seconds -- by shellunext_1693422229799@npunext.onmicrosoft.com at 9/27/2023, 2:21:11 PM on Shellunext's Cluster
```

Cmd 3

```python
df.display()
```

(1) Spark Jobs

- For the same workspace, there can be multiple users that work on different notebooks.
- For development, production, UAT env there are different workspaces.
- To share notebooks present under different users in different workspaces, the admin must provide access separately to every user for any file present in different workspaces.
- This is a very inefficient and time-consuming process.
- With unity catalog, workspace sharing can be enabled.
- This provides data access without needing admin permission.
- Azure Access Connector is used to provide connection to different workspace files.