

Context

Create Azure Synapse Analytics Workspace	2 – 4
Create SQL Pool	5 – 6
Analyze data with a dedicated SQL Pool	6 – 10
Create Pyspark Pool	11
Analyze Data with Pyspark Pool	12 – 14
Analyze data in the Storage Account	14 – 15
Integrate Pipelines	16 – 17
Mapping Data Flow	18 – 24
Querying and Analysing Data	24 – 27
Write and Execute Queries	27 – 32

Create Azure Synapse Analytics Workspace

1. Go to Resources and search for Synapse then click on Azure Synapse Analytics.

The screenshot shows the Azure Marketplace search results for 'synapse'. The search bar at the top contains 'synapse'. Below it, there are three main items listed:

- Azure Synapse Analytics** (Microsoft): Described as a 'Limitless analytics service with unmatched time to insight'.
- Azure Synapse Analytics (private link hubs)** (Microsoft): Described as a 'Connect to Azure Synapse Studio using private endpoints'.
- Datometry Hyper-Q for Azure Synapse Analytics** (Datometry): Described as a 'Re-platform from Teradata and Exadata to Azure Synapse at a fraction of the time, cost, and risk'.

2. Click on Create.

The screenshot shows the 'Create' page for Azure Synapse Analytics. At the top, it says 'Azure Synapse Analytics' and has a 'Create' button highlighted in blue.

3. Under Overview, create a New Resource group, give the name, create a new account name, and create a new File System Name. Click on Review + Create.

Create Synapse workspace

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all of your resources.

Subscription *	<input type="text" value="Microsoft Partner Network"/>
Resource group *	<input type="text" value="SynapseGroup"/> Create new
Managed resource group	<input type="text" value="Enter managed resource group name"/>

Workspace details

Name your workspace, select a location, and choose a primary Data Lake Storage Gen2 file system to serve as the default location for logs and job output.

Workspace name *	<input type="text" value="synapsetraining1619"/>
Region *	<input type="text" value="Australia East"/>
Select Data Lake Storage Gen2 *	<input checked="" type="radio"/> From subscription <input type="radio"/> Manually via URL
Account name *	<input type="text" value="synapseaccdemo"/> Create new
File system name *	<input type="text" value="demofile"/> Create new
<input checked="" type="checkbox"/> Assign myself the Storage Blob Data Contributor role on the Data Lake Storage Gen2 account to interactively query it in the workspace.	

[Review + create](#) [< Previous](#) [Next: Security >](#)

4. Click on Create.

Create Synapse workspace ...

The screenshot shows the 'Create Synapse workspace' wizard. At the top, a green bar indicates 'Validation succeeded'. Below it, the 'Review + create' tab is selected. The 'Product Details' section shows 'Azure Synapse Analytics workspace by Microsoft' with a 'Serverless SQL est. cost/TB' of '392.67 INR'. Terms of use and Privacy policy links are also present. The 'Terms' section contains legal text about agreeing to terms and privacy statements. The 'Basics' section shows the following configuration:

Subscription	Microsoft Partner Network
Resource group	(new) SynapseGroup
Region	Australia East
Workspace name	(new) synapsews1

At the bottom are 'Create' and 'Next >' buttons, along with a link to 'Download a template for automation'.

5. After creating you will see the below screen.

6. Click on Go to the resource group.

The screenshot shows the 'Microsoft.Azure.SynapseAnalytics-20230901095223 | Overview' page. It displays deployment details: Deployment name: Microsoft.Azure.SynapseAnalytics-20230901095223, Subscription: Microsoft Partner Network, Resource group: SynapseGroup, Start time: 9/1/2023, 9:53:46 AM, Correlation ID: b5ffa42e-ed71-4880-8ff2-ddc8867b6cc4. A message says 'Your deployment is complete'. Navigation links include 'Overview', 'Inputs', 'Outputs', and 'Template'. A 'Go to resource group' button is at the bottom.

7. Here click on Synapse Workspace.

The screenshot shows the 'SynapseGroup' resource group overview. The left sidebar lists 'Overview', 'Activity log', 'Access control (IAM)', 'Tags', 'Resource visualizer', 'Events', 'Deployments', and 'Security'. The main area shows 'Essentials' and 'Resources' sections. Under 'Resources', there are two records: 'synapseaccdemo' and 'synapsetraining1619'. A search bar and filter options are also present.

8. Now under Open Synapse Studio click on Open option.

The screenshot shows the Azure Synapse Analytics workspace overview page for 'synapsetraining1619'. The left sidebar includes sections for Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Settings (Azure Active Directory, Properties, Locks), Analytics pools (SQL pools, Apache Spark pools, Data Explorer pools (preview)), and Security. The main content area displays 'Essentials' information such as Resource group (move) : SynapseGroup, Status : Succeeded, Location : Australia East, Subscription (move) : Microsoft Partner Network, Subscription ID : df295c6e-136c-4598-9168-19fa9dba7fe1, Managed virtual network : No, Managed Identity object ID : 20acf47d-13b2-455f-b4c6-0b6a117310c6, Workspace web URL : <https://web.azuresynthesize.net?workspace=%2fsynapsetraining1619>, and Tags (edit) : Add tags. Below this is a 'Getting started' section with a 'Open Synapse Studio' button and a 'Read doc' button.

9. This is our synapse analytics workspace.

The screenshot shows the Microsoft Azure Synapse Analytics workspace interface for 'synapsetraining1619'. The top navigation bar includes 'Accept', 'Reject', and 'More options' buttons. The main content area features a large 3D bar chart visualization. On the left, there's a sidebar with icons for Home, Databases, Pipelines, Triggers, and Jupyter Notebooks. Below the sidebar, three cards are displayed: 'Ingest' (Perform a one-time or scheduled data load.), 'Explore and analyze' (Learn how to get insights from your data.), and 'Visualize' (Build interactive reports with Power BI capabilities.). At the bottom, there are links for 'Discover more' (Knowledge center, Browse partners) and 'Recent resources'.

Create SQL Pool

- Under Manage, in the SQL Pool click on Plus symbol New.

The screenshot shows the Azure portal's 'Manage' section. On the left, there's a sidebar with icons for Home, Analytics pools, SQL pools (which is selected and highlighted in grey), Apache Spark pools, Data Explorer pools (preview), External connections, Linked services, Microsoft Purview, and Integration. The main area is titled 'SQL pools' and contains a message: 'The serverless SQL pool, Built-in, is immediately available for use.' Below this are buttons for '+ New' and 'Refresh'. A 'Filter by name' input field is present. The text 'Showing 1-1 of 1 items (1 Serverless, 0 Dedicated)' is displayed. A table row shows the name 'Built-in'.

- Give the name and set the performance level then click on Review + Create.

The screenshot shows the 'New dedicated SQL pool' creation wizard on the 'Basics' tab. The tabs at the top are 'Basics *' (selected), 'Additional settings *', 'Tags', and 'Review + create'. A note below says: 'Create a dedicated SQL pool with your preferred configurations. Complete the Basics tab then go to Review + Create to provision with smart defaults. [Learn more](#)'.

Dedicated SQL pool details

Name your dedicated SQL pool and choose its initial settings.

Dedicated SQL pool name *:

Performance level: DW100c

Estimated price: View pricing details"/>

At the bottom are buttons for 'Review + create' (highlighted in blue) and 'Next: Additional settings >'

3. Now click on Create.

The screenshot shows the 'Review + create' step of the Azure Marketplace wizard. At the top, a green success message box says 'Validation succeeded.' Below it, tabs for 'Basics', 'Additional settings', 'Tags', and 'Review + create' are visible, with 'Review + create' being the active tab. The 'Product details' section shows 'Azure Synapse Analytics dedicated SQL pool by Microsoft' and an 'Est. cost per hour' of '132.72 INR'. A link to 'View pricing details' is also present. The 'Terms' section contains a detailed legal agreement text. The 'Data source' section lists 'Dedicated SQL pool name' as 'SQLPool1' and 'Performance level' as 'DW100c'. The 'Additional settings' section includes 'Use existing data' set to 'Blank' and 'Collation' set to 'SQL_Latin1_General_CI_AS'. At the bottom, there are three buttons: a blue 'Create' button, a grey '< Previous' button, and a grey 'Download template for automation' button.

4. Here we have created a Dedicated SQLPool.

The screenshot shows the 'SQL pools' blade in the Azure portal. It displays a table with two items: 'Built-in' (Serverless, Online) and 'SQLPool1' (Dedicated, Online). The table has columns for Name, Type, Status, and Size. A 'New' button and a 'Refresh' button are at the top left. A 'Filter by name' input field is below the table. The status bar indicates 'Showing 1-2 of 2 items (1 Serverless, 1 Dedicated)'.

Name	Type	Status	Size ↑
Built-in	Serverless	Online	Auto
SQLPool1	Dedicated	Online	DW100c

Analyze data with a dedicated SQL Pool

- Under Data, in the Linked, go to demofile then create a folder name as Data using the New Folder option.

The screenshot shows the 'Linked' blade of the Azure Data Lake Storage Gen2 'demofile' container. On the left, a tree view shows 'Azure Data Lake Storage Gen2' and 'synapsetraining1619 (Primary - syn...'. Under 'synapsetraining1619', 'demofile' is selected. In the center, a file browser shows a single item 'Data'. At the top right, there are buttons for 'New SQL script', 'New data flow', 'New integration dataset', 'Upload', 'Download', 'New folder', and 'Select all'. A 'Name' filter input field is also present. The status bar indicates 'Showing 1-2 of 2 items (1 Serverless, 1 Dedicated)'.

2. Go into the folder and click on the Upload option.

The screenshot shows the Azure Data Lake Storage Gen2 interface. On the left, there's a navigation bar with 'Data' and 'Linked' tabs. Under 'Linked', it shows 'Workspace' and 'Azure Data Lake Storage Gen2'. The 'Azure Data Lake Storage Gen2' section has a sub-item 'synapsetraining1619 (Primary - syn...)' and a file 'demofile (Primary)'. The right pane is titled 'demofile' and shows a file tree with 'demofile > Data'. There are buttons for 'Upload', 'New folder', and 'Select all'. Below the tree, it says 'No data available in this blob container'.

3. Select the file from your local system and click on Upload.

This is a 'Upload Files' dialog. It has sections for 'Upload Files' (with a dropdown for 'demofile'), 'Destination folder' (set to 'Data'), and 'File Upload' (with a file input field containing 'NYCTripSmall.parquet'). There's also a checkbox for 'Overwrite existing files'. Below the input field is a table showing the uploaded file: 'File name' is 'NYCTripSmall.parquet', 'Size' is '4.93 MB', and 'Action' is 'Remove'.

4. We are going to read this file using SQLPool.

The screenshot shows the Azure Data Lake Storage Gen2 interface. On the left, there's a navigation bar with 'Data' and 'Linked' tabs. Under 'Linked', it shows 'Workspace' and 'Azure Data Lake Storage Gen2'. The 'Azure Data Lake Storage Gen2' section has a sub-item 'synapsetraining1619 (Primary - syn...)' and a file 'demofile (Primary)'. The right pane is titled 'demofile' and shows a file tree with 'demofile > Data'. There are buttons for 'New SQL script', 'New notebook', 'New data flow', and 'New integration dataset'. Below the tree, it shows a file 'NYCTripSmall.parquet' with a timestamp of '1/9/2023, 10:28:24 am'.

5. Right-click on it and click on Properties.

A context menu is open over the file 'NYCTripSmall.parquet'. The menu items include: 'New SQL script', 'New notebook', 'New data flow', 'New integration dataset', 'Upload', 'Name', 'Last Modified', 'New SQL script', 'New notebook', 'New data flow', 'New integration dataset', 'Manage access...', 'Rename...', 'Download', 'Delete', and 'Properties...'. The 'Properties...' option is highlighted with a red box.

6. Copy the URL and paste it somewhere safe.

Properties

Name	Data/NYCTripSmall.parquet
URL	https://synapseaccdemo.dfs.core.windows.net/demofile/Data/NYCTripSmall.parquet
ABFSS Path	abfss://demofile@synapseaccdemo.dfs.core.windows.net/Data/NYCTripSmall.parquet
Last modified	

7. Under Develop click on the plus symbol and click on SQL script.

The screenshot shows the Microsoft Azure Synapse Analytics 'Develop' hub. On the left, there's a sidebar with icons for Synapse live, Validate all, Publish all, and a 'demofile' folder. The main area has a search bar 'Filter resources by name' and a dropdown menu with options: SQL script (which is highlighted), KQL script, Notebook, Data flow, Apache Spark job definition, Browse gallery, and Import. Below the menu is a list of resources starting with 'demofile'.

8. Under this link copy the below code.

Link: <https://learn.microsoft.com/en-us/azure/synapse-analytics/get-started-analyze-sql-pool>

The screenshot shows a web page titled 'Load the NYC Taxi Data into SQLPOOL1'. The left sidebar has a navigation menu with sections like 'Get started', 'Quickstarts', 'Overview', 'Azure Synapse Analytics', 'Switch to dedicated SQL pool (formerly SQL DW)', and 'Download PDF'. The main content area has a heading 'Load the NYC Taxi Data into SQLPOOL1' and three numbered steps:

- In Synapse Studio, navigate to the **Develop** hub, click the **+** button to add new resource, then create new SQL script.
- Select the pool 'SQLPOOL1' (pool created in **STEP 1** of this tutorial) in **Connect to** drop down list above the script.
- Enter the following code:

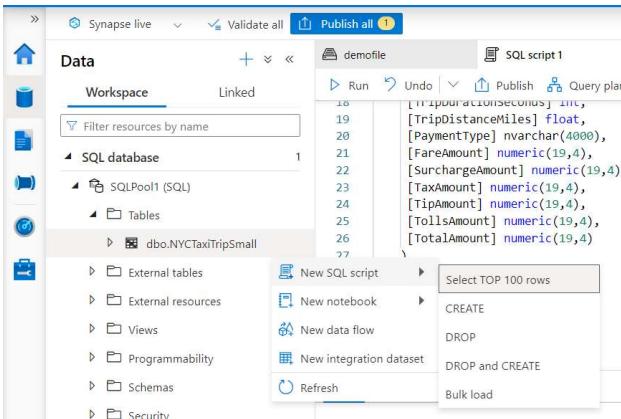
```
IF NOT EXISTS (SELECT * FROM sys.objects O JOIN sys.schemas S ON O.schema_id = S.schema_id WHERE O.name = 'NYCTaxiTripSmall')
CREATE TABLE dbo.NYCTaxiTripSmall
(
    [DateID] int,
    [MedallionID] int,
    [HackneyLicenseID] int,
    [PickupTimeID] int,
    [DropoffTimeID] int,
    [PickupGeographyID] int,
    [DropoffGeographyID] int,
    [PickupLatitude] float,
    [PickupLongitude] float,
    [PickupLatLong] nvarchar(4000),
    [DropoffLatitude] float,
    [DropoffLongitude] float,
    [DropoffLatLong] nvarchar(4000),
    [PassengerCount] int,
    [TripDurationSeconds] int,
    [TripDistanceMiles] float,
    [PaymentType] nvarchar(4000),
    [FareAmount] numeric(19,4),
    [TipAmount] numeric(19,4),
    [TollsAmount] numeric(19,4),
    [TotalAmount] numeric(19,4)
)
```

- Paste it into SQL Script and connect to SQLPool1 then change the URL to the File URL that we copied in the previous step.
- Next click on Run.

```

10    [TripDurationInSeconds] int,
11    [TripDistanceMiles] float,
12    [PaymentType] nvarchar(4000),
13    [FareAmount] numeric(19,4),
14    [SurchargeAmount] numeric(19,4),
15    [TaxAmount] numeric(19,4),
16    [TipAmount] numeric(19,4),
17    [TollsAmount] numeric(19,4),
18    [TotalAmount] numeric(19,4)
19 )
20 WITH
21 (
22    DISTRIBUTION = ROUNDRobin,
23    CLUSTERED COLUMNSTORE INDEX
24    -- HEAP
25 )
26 GO
27
28 COPY INTO dbo.NYCTaxiTripSmall
29    ([DateID], [MedallionID], [HackneyLicenseID], [PickupTimeID], [DropoffTimeID], [PickupGeographyID], [DropoffGeographyID], [PickupLatitude], [PickupLongitude], [DropoffLatitude], [DropoffLongitude], [PassengerCount], [TripDurationSeconds], [TripDistanceMiles], [PaymentType], [FareAmount], [SurchargeAmount], [TaxAmount], [TipAmount], [TollsAmount], [TotalAmount])
30    FROM https://synapseaccdemo.dfs.core.windows.net/demofile/Data/NYCTripSmall.parquet
31    WITH
32    (
33        FILE_TYPE = 'PARQUET'
34        ,MAXROWS = 0
35        ,IDENTITY_INSERT = 'OFF'
36    )
37 
```

- Now go to Data, In the Workspace under SQLPool click on the Tables and right click on the table go to New SQL Script, and click on Select TOP 100 rows.



- Here we can see the data inside that file.

DateID	MedallionID	HackneyLicens...	PickupTimeID	DropoffTimeID	PickupGeogra...	DropoffGeogr...	PickupLatitude	PickupLongitu...	Pickup...
20131221	3882	37099	617	1434	271412	231617	40.7622	-73.9861	40.762
20131220	5214	38968	49786	49933	87520	179427	40.7649	-73.9706	40.764
20131219	1074	11654	4680	5100	293054	74565	40.761	-73.9721	40.761
20131219	2780	25002	82740	84000	37119	44741	40.7552	-73.9918	40.755
20131231	9467	40793	46260	47160	62428	9539	40.7677	-73.9643	40.767
20131218	9314	41450	68100	68580	35918	293067	40.75	-74.0059	40.75
20131218	7402	33967	59820	60060	177929	260868	40.7476	-73.9938	40.747
20131218	7873	16092	71040	77720	44002	277054	40.7807	-73.9602	40.780

13. Replace the text of the SQL script with this code and run it.

Query: SELECT PassengerCount,
SUM(TripDistanceMiles) as SumTripDistance,
AVG(TripDistanceMiles) as AvgTripDistance
INTO dbo.PassengerCountStats
FROM dbo.NYCTaxiTripSmall
WHERE TripDistanceMiles > 0 AND PassengerCount > 0
GROUP BY PassengerCount;
SELECT * FROM dbo.PassengerCountStats
ORDER BY PassengerCount;

14. This query shows how the total trip distances and average trip distances relate to the number of passengers.

The screenshot shows a SQL query editor interface. At the top, there are buttons for Run, Undo, Publish, Query plan, and Connect to (SQLPool1). Below the toolbar is the SQL query itself, numbered from 1 to 9. The results tab is selected, showing a table with three columns: PassengerCount, SumTripDistance, and AvgTripDistance. The data is as follows:

PassengerCount	SumTripDistance	AvgTripDistance
1	190167.34	2.87357339297048
2	49399.74	3.13072691552063
3	14790.39	2.88480397893505
4	8558.35	2.9976707530648
5	17187.24	2.85502325581395
6	10741.91	2.87986863270778

Create Pyspark Pool

1. Under Manage, in the Apache Spark Pools click on the New plus symbol.
2. And give the properties as shown below. Now click on Review + Create.

New Apache Spark pool

Basics Additional settings Tags Review + create

Create a Synapse Analytics Apache Spark pool with your preferred configurations. Complete the Basics tab then go to Review + Create to provision with smart defaults, or visit each tab to customize.

Apache Spark pool details

Name your Apache Spark pool and choose its initial settings.

Apache Spark pool name * sparkdemo

Node size family * Memory Optimized

Node size * Medium (8 vCores / 64 GB)

Autoscale * Enabled

Number of nodes * 3

Estimated price * Est. cost per hour 291.32 to 291.32 INR View pricing details

Dynamically allocate executors * Enabled

Review + create Next: Additional settings >

3. Next click on Create.

New Apache Spark pool

Validation succeeded.

Basics Additional settings Tags Review + create

Azure Synapse Analytics Apache Spark pool by Microsoft Terms of use | Privacy policy Est. cost per hour 291.32 to 291.32 INR View pricing details

Product details

Terms

By clicking "Create", I agree to the legal terms and privacy statement(s) associated with the Marketplace offering(s) listed above; (b) authorize Microsoft to bill my current payment method for the fees associated with the offering(s), with the same billing frequency as my Azure subscription; and (c) agree that Microsoft may share my contact, usage and transactional information with the provider(s) of the offering(s) for support, billing and other transactional activities. Microsoft does not provide rights for third-party offerings. For additional details see Azure Marketplace Terms

Basics

Subscription Microsoft Partner Network

Resource group SynapseGroup

Apache Spark pool name sparkdemo

Node size family Memory Optimized

Node size Medium (8 vCores / 64 GB)

Create < Previous Download template for automation Cancel

4. Here we have created our Pyspark pool.

Analytics pools

Apache Spark pools

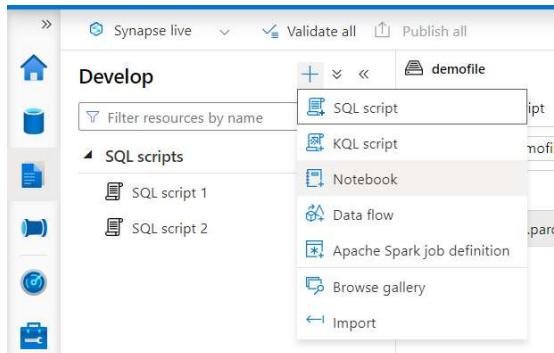
Pool : All

Showing 1 - 1 of 1 items

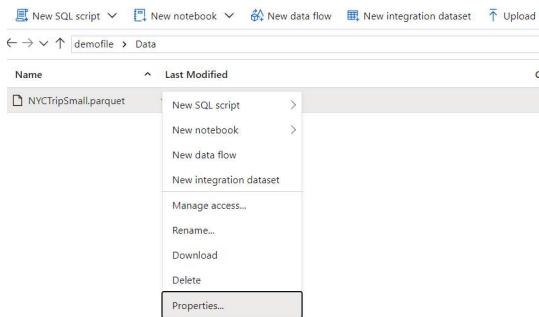
Pool name	Size	Active users	Allocated vCores
sparkdemo	Medium (8 vCores / 64 GB) - 3 to 0	0	24

Analyze Data with Pyspark Pool

5. In the Develop click on the plus symbol and click on a notebook.



15. Right-click on the file and click on Properties.



16. Copy the ABFSS Path and paste it somewhere safe.

Properties

Name

Data/NYCTripSmall.parquet



URL

<https://synapseaccdemo.dfs.core.windows.net/demofile/Data/NYCTripSmall.parquet>



ABFSS Path

abfss://demofile@synapseaccdemo.dfs.core.windows.net/Data/NYCTripSmall.parquet



Last modified

17. In the Notebook set the sparkdemo Attack.

18. And paste the below query. But in the query give the path you copied in the previous step.

19. Next click on the Run Cell option.

Query: %%pyspark

df =

```
spark.read.load(path='abfss://demofile@synapseaccdemo.dfs.core.windows.net/Data/NYC  
TripSmall.parquet')
```

Ready

```

1 %%pyspark
2 df = spark.read.load(path='abfss://demofile@synapseaccdemo.
[2] ✓ 17 sec - Command executed in 16 sec 904 ms by Lacchi.Balaji on 11:13:47 AM

```

20. Next click on the Plus symbol code and give the below code.

Code: `spark.sql("create Database if not exists nyctaxi")
df.write.mode("overwrite").saveAsTable("nyctaxi.trip")`

```

1 spark.sql("create Database if not exists nyctaxi")
2 df.write.mode("overwrite").saveAsTable("nyctaxi.trip")
[4] ✓ 17 sec - Command executed in 16 sec 982 ms by Lacchi.Balaji on 11:16:51 AM

```

21. Click on the Plus symbol of the code and give the below code.

Code: `%%pyspark
df = spark.sql("""
SELECT PassengerCount,
 SUM(TripDistanceMiles) as SumTripDistance,
 AVG(TripDistanceMiles) as AvgTripDistance
FROM nyctaxi.trip
WHERE TripDistanceMiles > 0 AND PassengerCount > 0
GROUP BY PassengerCount
ORDER BY PassengerCount
""")
display(df)
df.write.saveAsTable("nyctaxi.passengercountstats")`

```

1 %%pyspark
2 df = spark.sql("""
3 SELECT PassengerCount,
4     SUM(TripDistanceMiles) as SumTripDistance,
5     AVG(TripDistanceMiles) as AvgTripDistance
6 FROM nyctaxi.trip
7 WHERE TripDistanceMiles > 0 AND PassengerCount > 0
8 GROUP BY PassengerCount
9 ORDER BY PassengerCount
10 """
11 display(df)
12 df.write.saveAsTable("nyctaxi.passengercountstats")
[6] ✓ 7 sec - Command executed in 7 sec 45 ms by Lacchi.Balaji on 11:24:51 AM,

```

> Job execution Succeeded Spark 2 executors 16 cores

View Table Chart Export results

PassengerCount	SumTripDistance	Avg1
1	190167.33999999636	2.87:

22. Under the pyspark pool we have a table.

The screenshot shows the Azure Data Studio interface. On the left, there's a sidebar with icons for databases, tables, and views. The main workspace shows a tree view of a 'Lake database'. Under it, 'default' has a child 'nyctaxi', which contains a 'Tables' folder. Inside 'Tables', there are two entries: 'passengercountstats' and 'trip'. A 'Views' folder is also present. At the top right, there are buttons for 'Run all' and 'Session failed: R'. Below the workspace, there's a message: '+ Coc'.

Analyze data in the Storage Account

1. In this example we are converting the passengercountstatus table into csv and store in the Folder.
2. Create a Notebook.

The screenshot shows the 'Develop' tab in Azure Data Studio. On the left, there's a sidebar with icons for home, databases, tables, and notebooks. The main area shows a list of resources: 'SQL script', 'KQL script', 'Notebook', 'Data flow', 'Apache Spark job definition', 'Browse gallery', and 'Import'. A context menu is open over the 'SQL script' item, listing options like 'SQL script', 'KQL script', 'Notebook', 'Data flow', 'Apache Spark job definition', 'Browse gallery', and 'Import'. The 'Notebook' option is highlighted.

3. Give the below code and run it.

Code: %%pyspark

```
df = spark.sql("select * from nyctaxi.passengercountstats")
df1 = df.repartition(1)
df1.write.mode("overwrite").csv("/NYCTaxi/csvfile")
df1.write.mode("overwrite").parquet("/NYCTaxi/Parquetfile")
```

The screenshot shows the execution results of a notebook cell. The cell content is the same as the code provided above. The output shows a green checkmark and the text 'Ready'. Below the code, a message indicates a successful execution: '✓ 3 min 57 sec - Apache Spark session started in 3 min 6 sec 683 ms. Command executed in 50 sec 407 ms by Lacchi.Balaji ...'. At the bottom, it says 'Job execution Succeeded' with 'Spark 2 executors 16 cores', and links to 'View in monitoring' and 'Open Spark UI'.

4. Now go to Data and check the folder.

The screenshot shows the 'Data' section of the Azure portal. On the left, under 'Linked', there is a tree view of storage accounts: 'Azure Data Lake Storage Gen2' (2 items) and 'synapsetraining1619 (Primary - syn...)' (2 items). The second item under 'synapsetraining1619' is 'demofile (Primary)', which is selected. On the right, a detailed view of the 'demofile' folder is shown. It contains three subfolders: 'Data', 'NYCTaxi', and 'synapse'. The 'Data' folder was last modified on 1/9/2023 at 10:23. The 'NYCTaxi' folder was last modified on 1/9/2023 at 11:52. The 'synapse' folder was last modified on 1/9/2023 at 10:52.

5. Go into the folder and you will see a CSV folder.

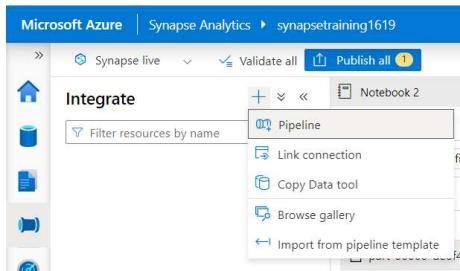
The screenshot shows the 'Data' section of the Azure portal. The 'Linked' view shows the same storage account structure as the previous screenshot. The 'demofile' folder is selected. On the right, the 'NYCTaxi' folder is expanded, showing two subfolders: 'csvfile' and 'Parquetfile'. Both were last modified on 1/9/2023.

6. Go into the CSV folder and double-click on the file, the csv file automatically will download.

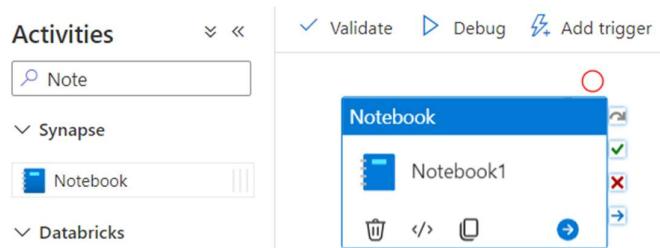
The screenshot shows the Microsoft Azure Synapse Analytics workspace for 'synapsetraining1619'. The top navigation bar includes 'Synapse live', 'Validate all', 'Publish all (1)', and a search bar. The left sidebar shows 'Data' and 'Linked' sections with the same storage account structure as before. The 'Linked' section shows the 'demofile' folder selected. On the right, the 'NYCTaxi' folder is selected, and its 'csvfile' subfolder is expanded. Two files are listed: '_SUCCESS' (last modified 1/9/2023) and 'part-00000-de6f4ee1-e754-47e4-9b79-5badca087ee3-c000.csv' (last modified 1/9/2023).

Integrate Pipelines

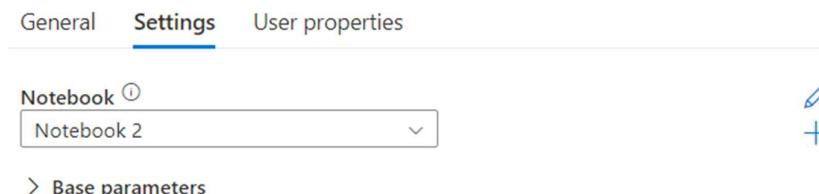
1. In the Integrate click on plus symbol and click on Pipeline.



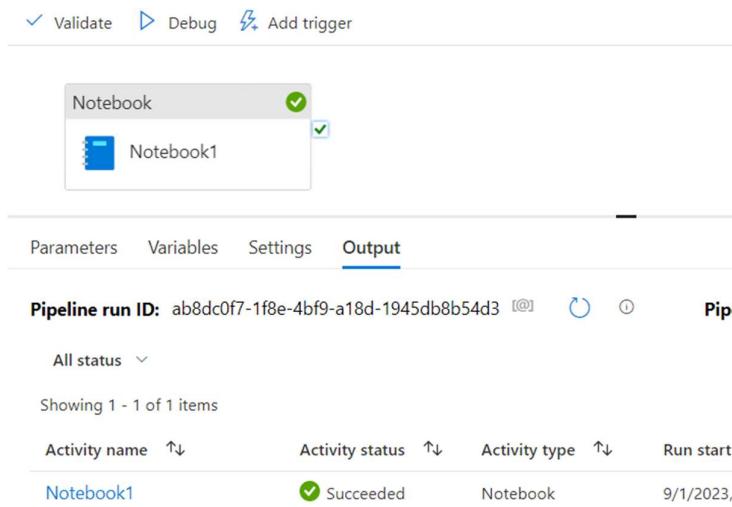
2. Drag and drop the Notebook.



3. Select the Notebook that we created in the previous example.



4. Here our pipeline executed successfully.



5. Click on the suynapseaccdemo storage account.

The screenshot shows the 'SynapseGroup' resource group overview. In the left sidebar, under 'Resources', there are four items listed: 'sparkdemo (synapsetraining1619/sparkdemo)', 'SQLPool1 (synapsetraining1619/SQLPool1)', 'synapseaccdemo', and 'synapsetraining1619'. The 'synapseaccdemo' item is highlighted with a blue border.

6. Under container click on Demo file.

The screenshot shows the 'synapseaccdemo' storage account containers page. In the left sidebar, under 'Data storage', 'Containers' is selected. A single container named 'demofile' is listed in the main table. The 'demofile' row is highlighted with a blue border.

7. You will see the folder that we created in the notebook.

The screenshot shows the 'demofile' storage container page. In the left sidebar, under 'Settings', 'Shared access tokens', 'Manage ACL', 'Access policy', and 'Properties' are listed. In the main area, there is a table showing blobs: 'Data' (unchecked), 'NYCTaxi' (checked and highlighted with a blue border), and 'synapse' (unchecked). The 'NYCTaxi' row is also highlighted with a blue border.

Mapping Data Flow

1. In this Example we are going to use the below csv file data.

	A	B	C	D	E
1	SID	NAME	SUBJECT	MARKS	
2	1	Mike	Maths	80	
3	1	Mike	Science	75	
4	1	Mike	English	76	
5	1	Mike	Hindi	68	
6	2	Dean	Maths	78	
7	2	Dean	Science	45	
8	2	Dean	English	65	
9	2	Dean	Hindi	69	

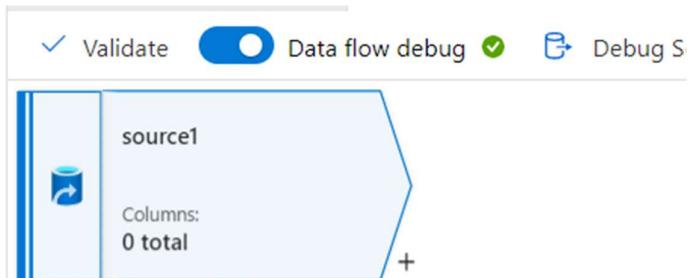
2. I uploaded my file into the Data folder.

The screenshot shows the Azure Data Lake Storage Gen2 Data folder. On the left, there's a navigation sidebar with icons for Home, Data, Workspace, and Linked. Under 'Linked', 'synapsetraining1619' is selected. The main area shows a list of files: 'NYCTripSmall.parquet' and 'Std_details.csv'. The 'Std_details.csv' file is highlighted.

3. Now go to Develop and click on the plus symbol then click on Dataflow.

The screenshot shows the Microsoft Azure Synapse Analytics Develop section. On the left, there's a navigation sidebar with icons for Home, Data, Develop, Workspace, and Linked. Under 'Develop', 'SQL scripts' is selected. A dropdown menu is open over the 'Dataflow' option, showing other options like 'SQL script', 'KQL script', 'Notebook', 'Apache Spark job definition', 'Browse gallery', and 'Import'.

4. Add a Data source.



5. Click on the Plus symbol to create a Dataset.

The screenshot shows the 'Source settings' tab. It includes fields for 'Output stream name' (set to 'source1'), 'Description' (set to 'Add source dataset'), and 'Source type' (set to 'Integration dataset'). There are also dropdowns for 'Dataset' (set to 'Select...') and 'Options' (with 'Allow schema drift' checked). A 'New' button is visible next to the 'Dataset' dropdown.

6. Select the Azure Blob Storage and click on continue.

7. Then select Delimited text and click on Continue.

8. Give the name and click on the plus symbol to create a New linked service.

Set properties

The screenshot shows the 'Set properties' dialog. It has a 'Name' field containing 'Std_DS' and a 'Linked service *' section. The 'Linked service *' section contains a 'Select...' dropdown, a 'Filter...' input field, and a '+ New' button.

9. Set the below properties and click on Create.

New linked service

Azure Blob Storage [Learn more](#)

Name *
SynapseDemo

Description

Connect via integration runtime * ⓘ
AutoResolveIntegrationRuntime

Authentication type
Account key

Connection string Azure Key Vault

Account selection method ⓘ
From Azure subscription Enter manually

Azure subscription ⓘ
Microsoft Partner Network (df295c6e-136c-4598-9168-19fa9dba7fe1)

Storage account name *
synapseaccdemo

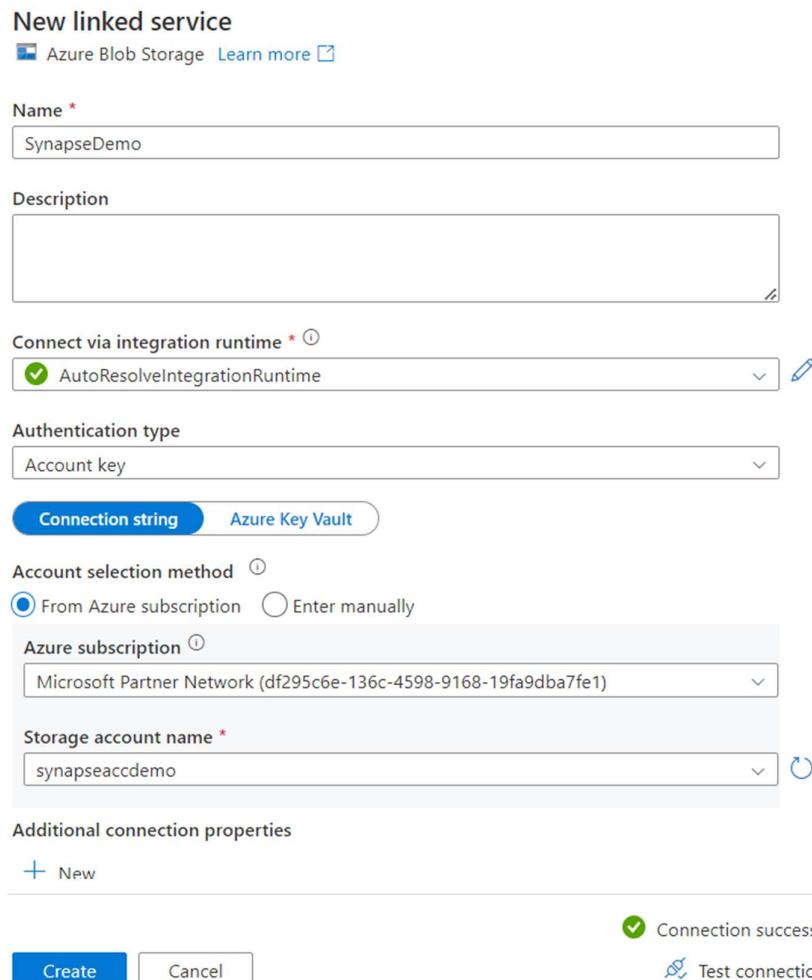
Additional connection properties

+ New

✓ Connection successful

Create Cancel

Test connection



10. Select the file and click on OK.

Set properties

Name
Std_DS

Linked service *
SynapseDemo

Connect via integration runtime * ⓘ
AutoResolveIntegrationRuntime

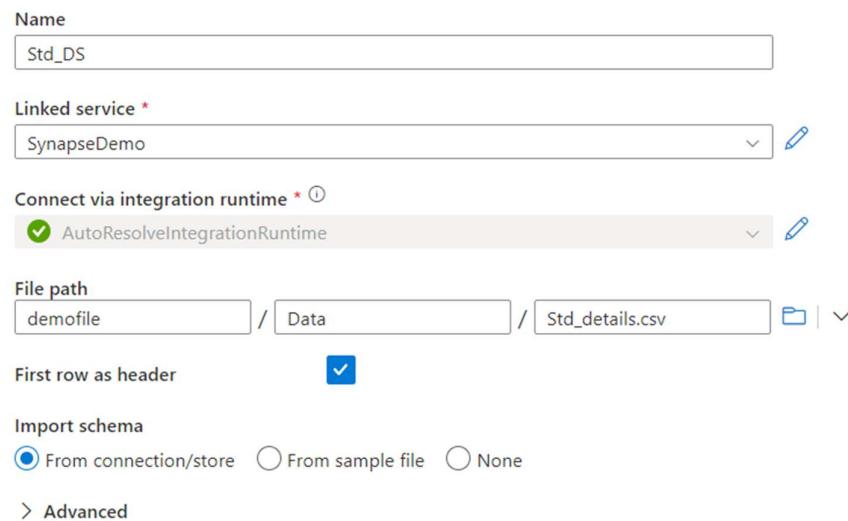
File path
demofile / Data / Std_details.csv

First row as header

Import schema

From connection/store From sample file None

> Advanced



11. Here we have assigned our source dataset.

Source settings Source options Projection Optimize Inspect Data preview

Output stream name * Learn more

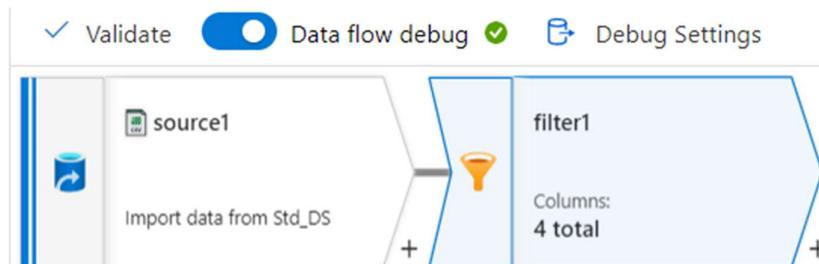
Description Reset

Source type * Integration dataset Inline Workspace DB

Dataset * Test connection Open New

Options Allow schema drift

12. Add filter Transformation.



13. In the Filter Settings, under Filter On give the below expression.

Exp: equals(NAME, 'Dean')

Filter settings Optimize Inspect Data preview

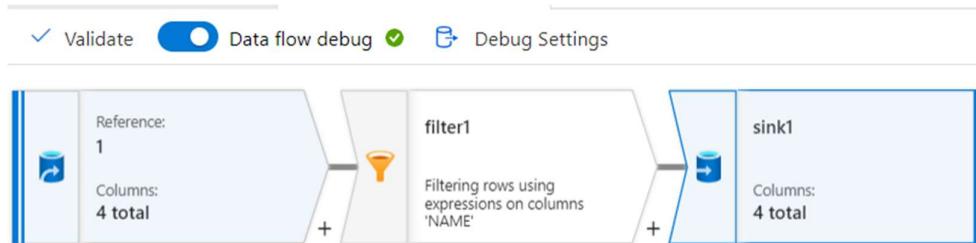
Output stream name * Learn more

Description Reset

Incoming stream *

Filter on *

14. Add Sink Transformation.



15. Click on the Plus symbol to create a Target Dataset.

Sink Settings Errors Mapping Optimize Inspect Data preview ●

Output stream name * sink1 Learn more ↗

Description Add sink dataset Reset

Incoming stream * filter1

Sink type * Integration dataset Inline Workspace DB Cache

Dataset * Select... + New

Options Allow schema drift ⓘ

16. Select the Azure Blob Storage and click on continue.

17. Then select Delimited text and click on Continue.

18. Give the name, linked service, and file path. Don't give a file just give the location.

Set properties

Name TargetDS

Linked service * SynapseDemo edit

Connect via integration runtime * ⓘ AutoResolveIntegrationRuntime edit

File path demofile / NYCTaxi / File name edit

First row as header

Import schema From connection/store From sample file None

19. Here we assigned our Target dataset.

Sink Settings Errors Mapping Optimize Inspect Data preview ●

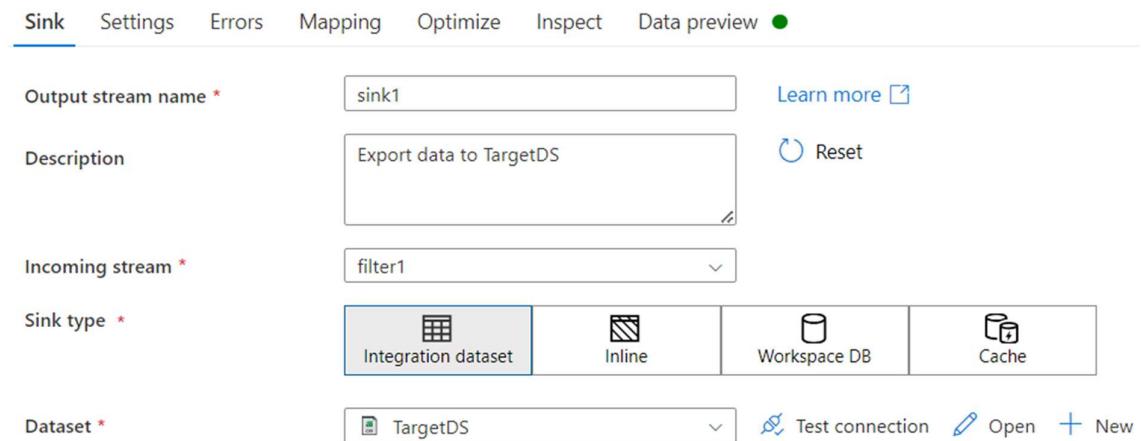
Output stream name * sink1 [Learn more](#)

Description Export data to TargetDS [Reset](#)

Incoming stream * filter1

Sink type * [Integration dataset](#) [Inline](#) [Workspace DB](#) [Cache](#)

Dataset * [TargetDS](#) [Test connection](#) [Open](#) [New](#)



20. Under Settings select the file option and give the file name.

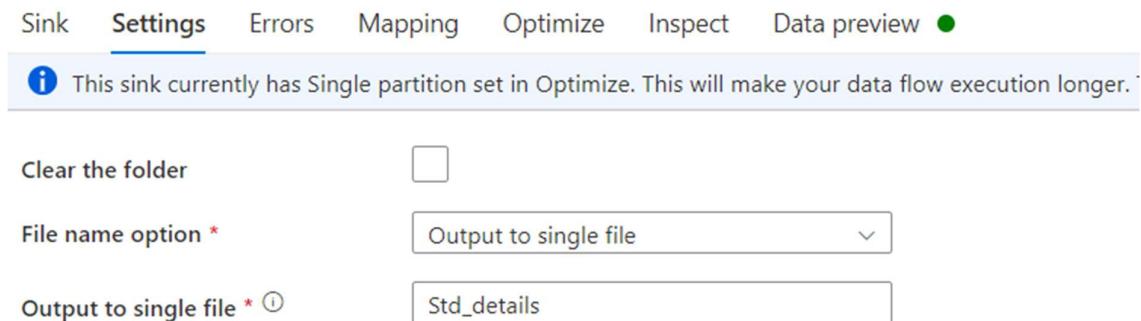
Sink **Settings** Errors Mapping Optimize Inspect Data preview ●

i This sink currently has Single partition set in Optimize. This will make your data flow execution longer.

Clear the folder

File name option * Output to single file

Output to single file * Std_details



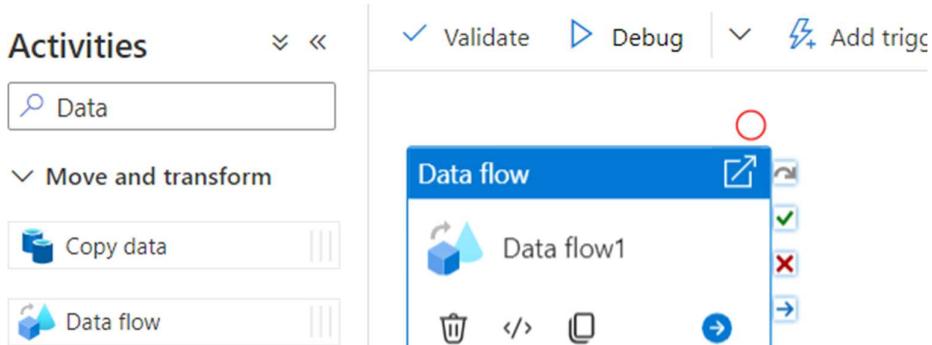
21. Create a Pipeline, Drag and Drop the data flow.

Activities [Validate](#) [Debug](#) [Add trigger](#)

Data

Move and transform [Copy data](#) [Data flow](#)

Data flow Data flow1



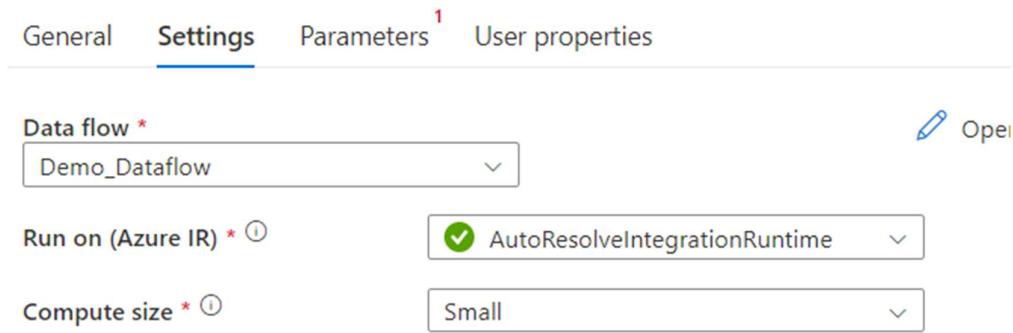
22. Under Settings select the Dataflow.

General **Settings** Parameters ¹ User properties

Data flow * Demo_Dataflow [Open](#)

Run on (Azure IR) * AutoResolveIntegrationRuntime

Compute size * Small



23. Validate it and click on Debug.
24. Here our pipeline executed successfully.

Pipeline run ID: 62dc7c73-a68e-444b-a224-8e5da7ce9355

Activity name ↑↓	Activity status ↑↓	Activity type ↑↓
Data flow1	✓ Succeeded	Data flow

Querying and Analysing Data

1. Under develop click on plus symbol and click on SQL Script.

2. Create a Table with the below query.

```
Query: CREATE TABLE Sales (
    SaleID INT IDENTITY(1, 1),
    CustomerID INT,
    SaleDate DATE,
    Amount DECIMAL(10, 2),
    PRIMARY KEY NONCLUSTERED (SaleID) NOT ENFORCED
);
```

3. Give the above query and click on Run.

The screenshot shows a SQL script editor interface. The title bar says "SQL script 3". There are tabs for "Run", "Undo", "Publish", "Query plan", and "Connect to". A message bubble says "Other users in your workspace may have access to this workspace." The code in the editor is:

```
1 CREATE TABLE Sales (
2     SaleID INT IDENTITY(1, 1),
3     CustomerID INT,
4     SaleDate DATE,
5     Amount DECIMAL(10, 2),
6     PRIMARY KEY NONCLUSTERED (SaleID) NOT ENFORCED
7 );
```

4. Insert sample data.

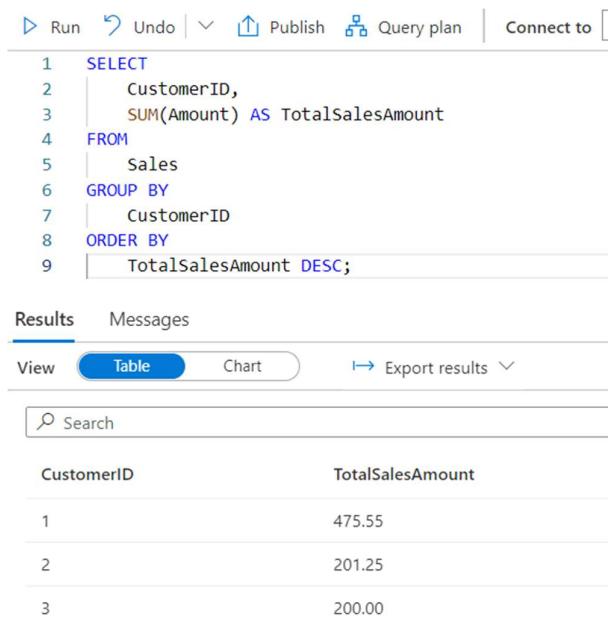
Query: INSERT INTO Sales (CustomerID, SaleDate, Amount)
SELECT 1, '2023-09-01', 100.00
UNION ALL
SELECT 2, '2023-09-02', 150.50
UNION ALL
SELECT 1, '2023-09-03', 75.25
UNION ALL
SELECT 3, '2023-09-04', 200.00
UNION ALL
SELECT 2, '2023-09-05', 50.75
UNION ALL
SELECT 1, '2023-09-06', 300.30;

The screenshot shows a SQL script editor interface. The title bar says "SQL script 3". There are tabs for "Run", "Undo", "Publish", "Query plan", and "Connect to". A message bubble says "Other users in your workspace may have access to this workspace." The code in the editor is:

```
1 INSERT INTO Sales (CustomerID, SaleDate, Amount)
2 SELECT 1, '2023-09-01', 100.00
3 UNION ALL
4 SELECT 2, '2023-09-02', 150.50
5 UNION ALL
6 SELECT 1, '2023-09-03', 75.25
7 UNION ALL
8 SELECT 3, '2023-09-04', 200.00
9 UNION ALL
10 SELECT 2, '2023-09-05', 50.75
11 UNION ALL
12 SELECT 1, '2023-09-06', 300.30;
```

5. Now, let's perform some SQL queries to extract insights from the sample data.
6. Total Sales Amount by Customer.

Query: SELECT
 CustomerID,
 SUM(Amount) AS TotalSalesAmount
 FROM
 Sales
 GROUP BY
 CustomerID
 ORDER BY
 TotalSalesAmount DESC;



The screenshot shows a SQL query editor interface. At the top, there are buttons for Run, Undo, Publish, Query plan, and Connect to []. Below the buttons is a code editor containing the following SQL query:

```

1  SELECT
2  |   CustomerID,
3  |   SUM(Amount) AS TotalSalesAmount
4  FROM
5  |   Sales
6  GROUP BY
7  |   CustomerID
8  ORDER BY
9  |   TotalSalesAmount DESC;

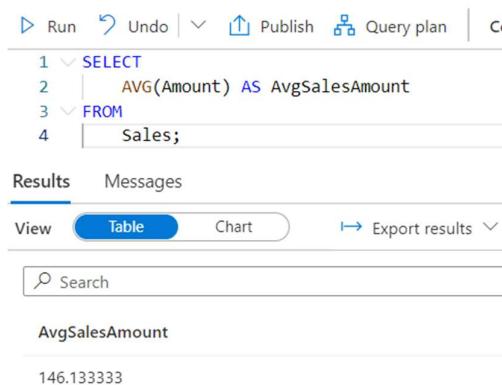
```

Below the code editor, there are tabs for Results and Messages, with Results being selected. Under the Results tab, there are buttons for View (set to Table), Chart, and Export results. A search bar is also present. The results table displays the following data:

CustomerID	TotalSalesAmount
1	475.55
2	201.25
3	200.00

7. This query will give you the total sales amount for each customer, ordered from the highest total sales to the lowest.
8. Average Sales Amount per Transaction.

Query: SELECT
 AVG(Amount) AS AvgSalesAmount
 FROM
 Sales;



The screenshot shows a SQL query editor interface. At the top, there are buttons for Run, Undo, Publish, Query plan, and Connect to []. Below the buttons is a code editor containing the following SQL query:

```

1  SELECT
2  |   AVG(Amount) AS AvgSalesAmount
3  FROM
4  |   Sales;

```

Below the code editor, there are tabs for Results and Messages, with Results being selected. Under the Results tab, there are buttons for View (set to Table), Chart, and Export results. A search bar is also present. The results table displays the following data:

AvgSalesAmount
146.133333

9. This query calculates the average sales amount per transaction.
10. Sales on a Specific Date.

Query: SELECT

```

SaleDate,
SUM(Amount) AS TotalSalesAmount
FROM
Sales
WHERE
SaleDate = '2023-09-03'
GROUP BY
SaleDate;
```

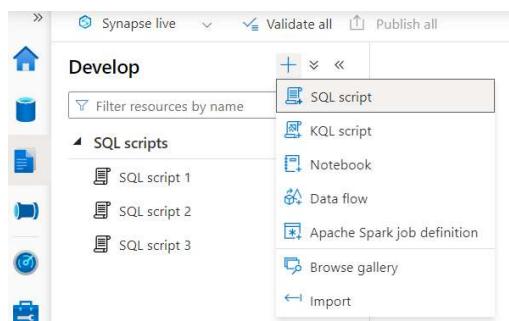
The screenshot shows the Azure Synapse Studio interface. At the top, there's a toolbar with 'Run', 'Undo', 'Publish', 'Query plan', and 'Connect' buttons. Below the toolbar is a code editor window containing the SQL query provided above. The code is numbered from 1 to 9. The results pane below the code editor has tabs for 'Results' and 'Messages', with 'Results' selected. Under 'Results', there are buttons for 'View' (set to 'Table'), 'Chart', and 'Export results'. A search bar is also present. The results table shows one row with columns 'SaleDate' and 'TotalSalesAmount'. The data is: SaleDate: 2023-09-03T00:00:00.0000000 and TotalSalesAmount: 75.25.

SaleDate	TotalSalesAmount
2023-09-03T00:00:00.0000000	75.25

11. This query shows the total sales amount on a specific date (in this case, '2023-09-03').
12. You can execute these queries in Azure Synapse Studio or any SQL client tool connected to your Azure Synapse SQL Data Warehouse to analyze the sample data and gain meaningful insights.

Write and Execute Queries

1. Create another SQL Script.



2. Create two sample tables: "Customers" and "Orders."

Query: CREATE TABLE Customers (

```
CustomerID INT,  
CustomerName VARCHAR(255),  
Email VARCHAR(255),  
CONSTRAINT UQ_CustomerID UNIQUE NONCLUSTERED (CustomerID) NOT ENFORCED  
);
```

CREATE TABLE Orders (

```
OrderID INT,  
CustomerID INT,  
OrderDate DATE,  
TotalAmount DECIMAL(10, 2),  
CONSTRAINT UQ_OrderID UNIQUE NONCLUSTERED (OrderID) NOT ENFORCED  
);
```

The screenshot shows a SQL query editor interface. At the top, there are buttons for Run, Undo, Publish, Query plan, Connect to (set to SQLPool1), and Use database. The main area contains the SQL code for creating the 'Customers' and 'Orders' tables. The code is numbered from 1 to 14. Lines 1-6 define the 'Customers' table, and lines 8-14 define the 'Orders' table. The 'NOT ENFORCED' part of the constraint definition is highlighted in red.

```
1 CREATE TABLE Customers (  
2     CustomerID INT,  
3     CustomerName VARCHAR(255),  
4     Email VARCHAR(255),  
5     CONSTRAINT UQ_CustomerID UNIQUE NONCLUSTERED (CustomerID) NOT ENFORCED  
6 );  
7  
8 CREATE TABLE Orders (  
9     OrderID INT,  
10    CustomerID INT,  
11    OrderDate DATE,  
12    TotalAmount DECIMAL(10, 2),  
13    CONSTRAINT UQ_OrderID UNIQUE NONCLUSTERED (OrderID) NOT ENFORCED  
14 );
```

3. Now, let's insert some sample data into these tables.

Query: INSERT INTO Customers (CustomerID, CustomerName, Email)

```
SELECT 1, 'John Smith', 'john@example.com'  
UNION ALL  
SELECT 2, 'Alice Johnson', 'alice@example.com'  
UNION ALL  
SELECT 3, 'Bob Brown', 'bob@example.com';
```

INSERT INTO Orders (OrderID, CustomerID, OrderDate, TotalAmount)

```
SELECT 101, 1, '2023-09-01', 100.00
```

```
UNION ALL
```

```
SELECT 102, 2, '2023-09-02', 150.50
```

```
UNION ALL
```

```
SELECT 103, 1, '2023-09-03', 75.25
```

```
UNION ALL
```

```
SELECT 104, 3, '2023-09-04', 200.00
```

```
UNION ALL
```

```
SELECT 105, 2, '2023-09-05', 50.75
```

```
UNION ALL
SELECT 106, 1, '2023-09-06', 300.30;
```

```
1 INSERT INTO Customers (CustomerID, CustomerName, Email)
2 SELECT 1, 'John Smith', 'john@example.com'
3 UNION ALL
4 SELECT 2, 'Alice Johnson', 'alice@example.com'
5 UNION ALL
6 SELECT 3, 'Bob Brown', 'bob@example.com';
7
8 INSERT INTO Orders (OrderID, CustomerID, OrderDate, TotalAmount)
9 SELECT 101, 1, '2023-09-01', 100.00
10 UNION ALL
11 SELECT 102, 2, '2023-09-02', 150.50
12 UNION ALL
13 SELECT 103, 1, '2023-09-03', 75.25
14 UNION ALL
15 SELECT 104, 3, '2023-09-04', 200.00
```

4. With the tables and data in place, let's write some SQL queries to analyze the data.
5. Retrieve Customer Information with Orders.
6. This query retrieves customer information along with the number of orders and the total amount spent by each customer.

Query: SELECT
c.CustomerName,
c.Email,
COUNT(o.OrderID) AS OrderCount,
SUM(o.TotalAmount) AS TotalSpent
FROM
Customers c
LEFT JOIN Orders o ON c.CustomerID = o.CustomerID
GROUP BY c.CustomerName, c.Email
ORDER BY TotalSpent DESC;

```
1 SELECT
2     c.CustomerName,
3     c.Email,
4     COUNT(o.OrderID) AS OrderCount,
5     SUM(o.TotalAmount) AS TotalSpent
6 FROM
7     Customers c
8 LEFT JOIN Orders o ON c.CustomerID = o.CustomerID
9 GROUP BY c.CustomerName, c.Email
10 ORDER BY TotalSpent DESC;
```

Results

CustomerName	Email	OrderCount	TotalSpent
John Smith	john@example.com	3	475.55
Alice Johnson	alice@example.com	2	201.25
Bob Brown	bob@example.com	1	200.00

7. Retrieve Orders Placed in September 2023.
8. This query retrieves all orders placed in September 2023.

Query: SELECT

```

o.OrderID,
c.CustomerName,
o.OrderDate,
o.TotalAmount
FROM
Orders o
JOIN Customers c ON o.CustomerID = c.CustomerID
WHERE o.OrderDate >= '2023-09-01' AND o.OrderDate < '2023-10-01';

```

OrderID	CustomerName	OrderDate	TotalAmount
101	John Smith	2023-09-01T00:00:00.0000000	100.00
102	Alice Johnson	2023-09-02T00:00:00.0000000	150.50
103	John Smith	2023-09-03T00:00:00.0000000	75.25
104	Bob Brown	2023-09-04T00:00:00.0000000	200.00
105	Alice Johnson	2023-09-05T00:00:00.0000000	50.75
106	John Smith	2023-09-06T00:00:00.0000000	300.30

9. Without Indexing.
10. This query represents the initial query without any indexing or optimization. It may perform slowly, especially on a large dataset.

Query: SELECT

```

c.CustomerName,
SUM(o.TotalAmount) AS TotalSalesAmount
FROM
Customers c
JOIN Orders o ON c.CustomerID = o.CustomerID
WHERE o.OrderDate BETWEEN '2023-09-01' AND '2023-09-30'
GROUP BY c.CustomerName;

```

11. Give the Query and click on Run.

```
1 SELECT
2     c.CustomerName,
3     SUM(o.TotalAmount) AS TotalSalesAmount
4 FROM
5     Customers c
6 JOIN Orders o ON c.CustomerID = o.CustomerID
7 WHERE o.OrderDate BETWEEN '2023-09-01' AND '2023-09-30'
8 GROUP BY c.CustomerName;
```

Results Messages

View Table Chart Export results

Search

CustomerName	TotalSalesAmount
Alice Johnson	201.25
John Smith	475.55
Bob Brown	200.00

12. With Appropriate Indexing.

13. It adds an index on the OrderDate column in the Orders table. This index can significantly improve performance for date-range queries.

Query: CREATE INDEX IX_OrderDate ON Orders (OrderDate);

```
SELECT
    c.CustomerName,
    SUM(o.TotalAmount) AS TotalSalesAmount
FROM
    Customers c
JOIN Orders o ON c.CustomerID = o.CustomerID
WHERE o.OrderDate BETWEEN '2023-09-01' AND '2023-09-30'
GROUP BY c.CustomerName;
```

```
1 CREATE INDEX IX_OrderDate ON Orders (OrderDate);
2
3 SELECT
4     c.CustomerName,
5     SUM(o.TotalAmount) AS TotalSalesAmount
6 FROM
7     Customers c
8 JOIN Orders o ON c.CustomerID = o.CustomerID
9 WHERE o.OrderDate BETWEEN '2023-09-01' AND '2023-09-30'
10 GROUP BY c.CustomerName;
```

Results Messages

View Table Chart Export results

Search

CustomerName	TotalSalesAmount
Alice Johnson	201.25
John Smith	475.55
Bob Brown	200.00

14. With Columnstore Index.
15. It introduces a clustered columnstore index on the Orders table. Columnstore indexes are optimized for large analytical queries and can provide excellent performance for aggregations.

Query: CREATE CLUSTERED COLUMNSTORE INDEX IX_Columnstore ON Orders;

```

SELECT
    c.CustomerName,
    SUM(o.TotalAmount) AS TotalSalesAmount
FROM
    Customers c
JOIN Orders o ON c.CustomerID = o.CustomerID
WHERE o.OrderDate BETWEEN '2023-09-01' AND '2023-09-30'
GROUP BY c.CustomerName;

```

The screenshot shows a SQL query editor interface. At the top, there are navigation buttons for Run, Undo, Publish, Query plan, and Connect to (SQLPool1). Below the toolbar, the query code is displayed with line numbers:

```

1  CREATE CLUSTERED COLUMNSTORE INDEX IX_Columnstore ON Orders;
2
3  SELECT
4      c.CustomerName,
5      SUM(o.TotalAmount) AS TotalSalesAmount
6  FROM
7      Customers c
8  JOIN Orders o ON c.CustomerID = o.CustomerID
9  WHERE o.OrderDate BETWEEN '2023-09-01' AND '2023-09-30'
10 GROUP BY c.CustomerName;

```

Below the code, there are tabs for Results and Messages, with Results being selected. Under View, the Table tab is highlighted. The results section displays the query output:

CustomerName	TotalSalesAmount
Alice Johnson	201.25
John Smith	475.55
Bob Brown	200.00