

ASSIGNMENT 11

NAME – JYOTHI CHANDANA VOLETI
BATCH – DXC-262-ANALYTICS-B12-AZURE
EMPLOYEE DOMAIN –AZURE ANALYTICS
TRAINING UNDER – MANIPAL PRO LEARN
DATE OF SUBMISSION – 17TH JUNE 2022

ROLL NUMBER – DXC-262-AB-1218
COMPANY – DXC TECHNOLOGY

TRAINER NAME – MR. AJAY KUMAR
NO.OF QUESTIONS: 8

1. Write a python program to predict car sales of a company by using below data,

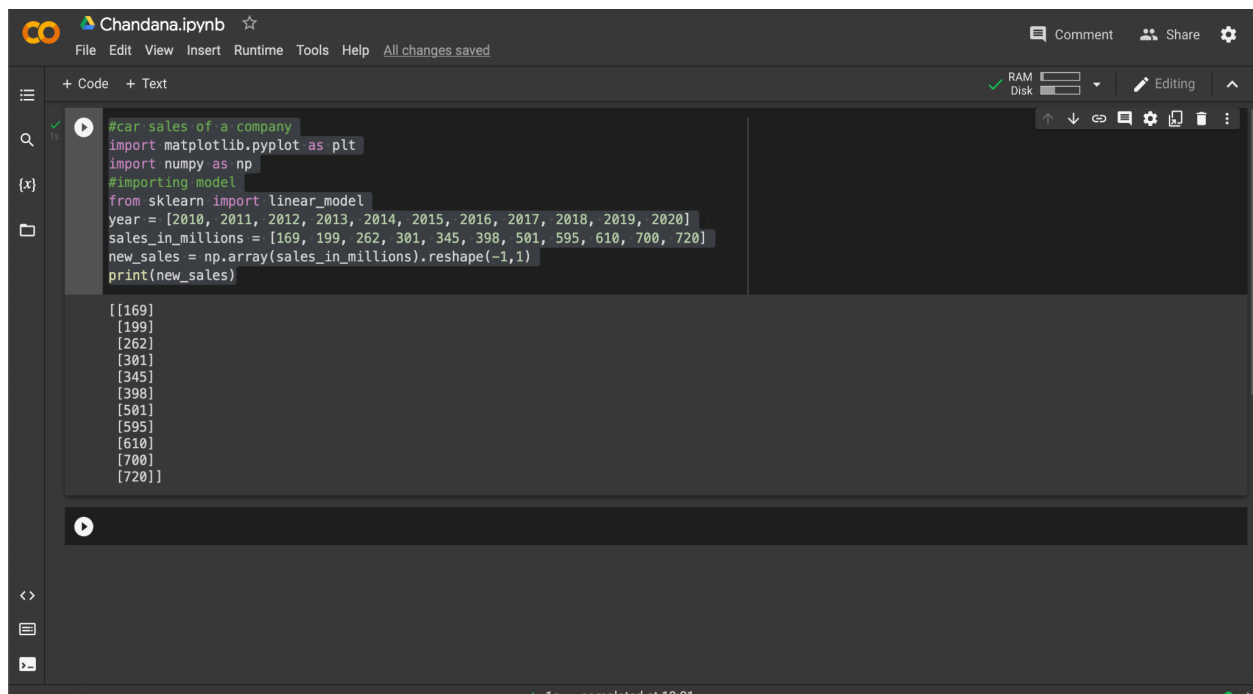
year : 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019 2020

Sales in millions: 169 199 262 301 345 398 501 595 610 700 720

display outcome using linear regression method.

Ans: PYTHON PROGRAM

```
#car sales of a company
import matplotlib.pyplot as plt
import numpy as np
#importing model
from sklearn import linear_model
year = [2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019, 2020]
sales_in_millions = [169, 199, 262, 301, 345, 398, 501, 595, 610, 700, 720]
new_sales = np.array(sales_in_millions).reshape(-1,1)
print(new_sales)
```



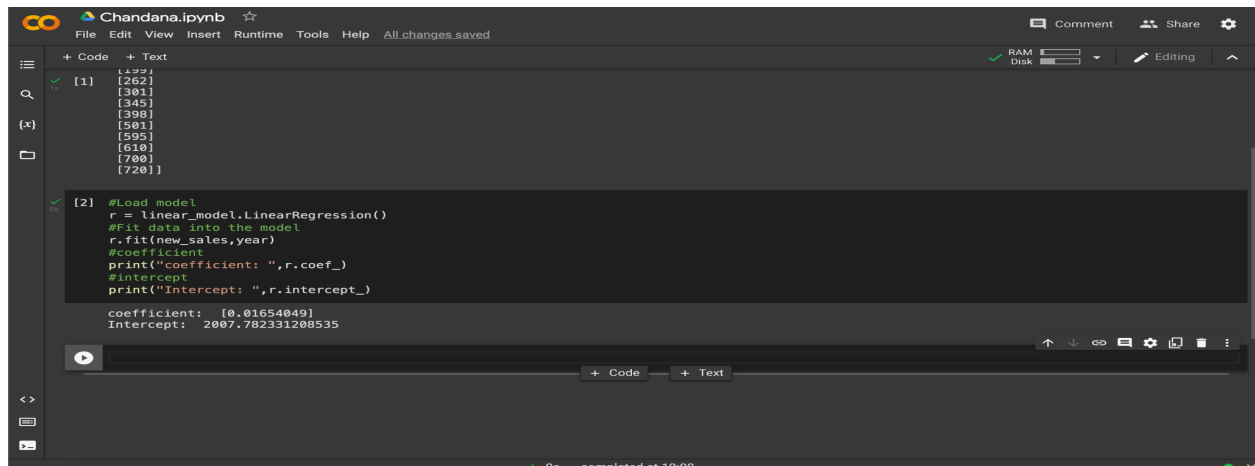
```
Chandana.ipynb
File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text
#car sales of a company
import matplotlib.pyplot as plt
import numpy as np
#importing model
from sklearn import linear_model
year = [2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019, 2020]
sales_in_millions = [169, 199, 262, 301, 345, 398, 501, 595, 610, 700, 720]
new_sales = np.array(sales_in_millions).reshape(-1,1)
print(new_sales)

[[169]
 [199]
 [262]
 [301]
 [345]
 [398]
 [501]
 [595]
 [610]
 [700]
 [720]]
```

```

#Load model
r = linear_model.LinearRegression()
#Fit data into the model
r.fit(new_sales,year)
#coefficient
print("coefficient: ",r.coef_)
#intercept
print("Intercept: ",r.intercept_)

```



```

Chandana.ipynb
File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text
[1] [1]
[262]
[301]
[345]
[398]
[501]
[595]
[610]
[700]
[720]

[2] #Load model
r = linear_model.LinearRegression()
#Fit data into the model
r.fit(new_sales,year)
#coefficient
print("coefficient: ",r.coef_)
#intercept
print("Intercept: ",r.intercept_)

coefficient: [0.01654049]
Intercept: 2007.782331208535

```

```

def graph(formula,x_range):
    x=np.array(x_range)
    y=eval(formula)
    plt.plot(x,y)
#plotting the prediction
graph('r.coef_*x+r.intercept_',range(0,2500))
plt.scatter(new_sales,year,color='purple')
plt.ylabel('year')
plt.xlabel('Sales in millions')
plt.show()

```



2. Write a python program to generate possible tuples from any two sample Lists.

Ans: PYTHON PROGRAM TO GENERATE POSSIBLE TUPLES FROM ANY TWO SAMPLE LISTS.

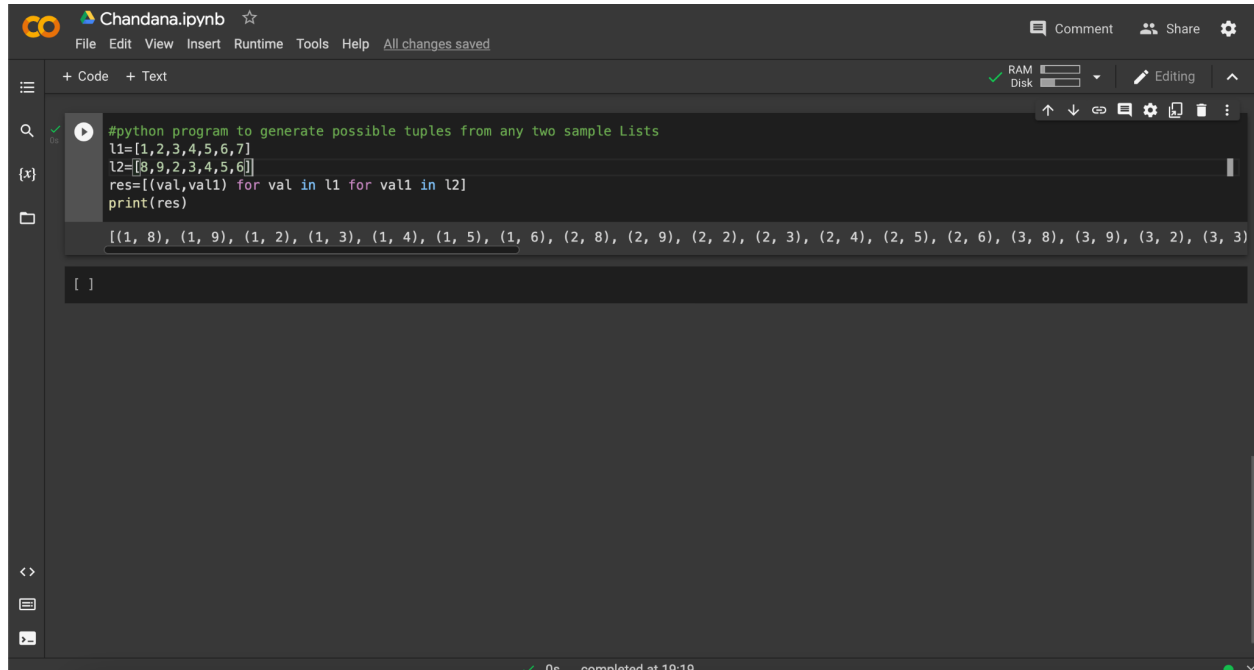
#python program to generate possible tuples from any two sample Lists

```
l1=[1,2,3,4,5,6,7]
```

```
l2=[8,9,2,3,4,5,6]
```

```
res=[(val,val1) for val in l1 for val1 in l2]
```

```
print(res)
```



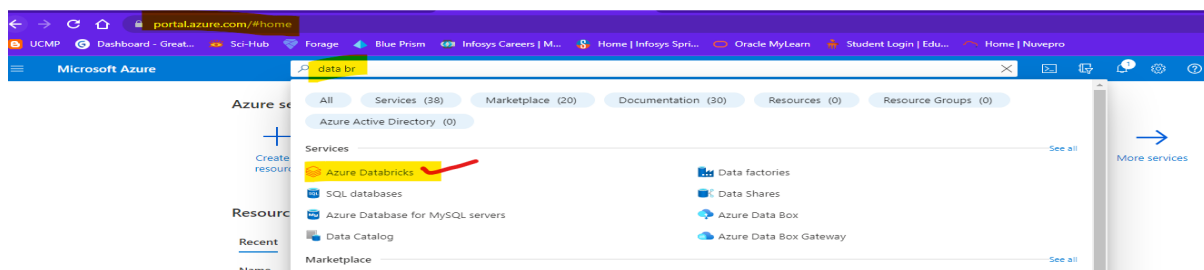
```
#python program to generate possible tuples from any two sample Lists
l1=[1,2,3,4,5,6,7]
l2=[8,9,2,3,4,5,6]
res=[(val,val1) for val in l1 for val1 in l2]
print(res)
```

```
[(1, 8), (1, 9), (1, 2), (1, 3), (1, 4), (1, 5), (1, 6), (2, 8), (2, 9), (2, 2), (2, 3), (2, 4), (2, 5), (2, 6), (3, 8), (3, 9), (3, 2), (3, 3), (3, 4), (3, 5), (3, 6), (4, 8), (4, 9), (4, 2), (4, 3), (4, 4), (4, 5), (4, 6), (5, 8), (5, 9), (5, 2), (5, 3), (5, 4), (5, 5), (5, 6), (6, 8), (6, 9), (6, 2), (6, 3), (6, 4), (6, 5), (6, 6), (7, 8), (7, 9), (7, 2), (7, 3), (7, 4), (7, 5), (7, 6)]
```

3. Create Azure Databricks & try to connect databricks & powerBI , explain the steps with screenshots.

Ans: To create Data Bricks we have to follow the following steps.

Step-1: Login into the azure portal and search for the Data Bricks.



Step-2: Click on “Azure Data Dricks” and it will navigate you to the page. And click on “create” to create Data Bricks.

Home > Azure Databricks >

Create an Azure Databricks workspace

Basics Networking Advanced Tags Review + create

Project Details

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

Subscription * ⓘ Azure-DXC262AB12Lab

Resource group * ⓘ dxcorg231
[Create new](#)

Instance Details

Workspace name * dxcdb231

Region * East US

Pricing Tier * ⓘ Standard (Apache Spark, Secure with Azure AD)

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

Step-3: After checking all these reviews and creating and waiting for deployment after deployment we will get like this.

Notifications

[More events in the activity log →](#) [Dismiss](#)

✔ **Deployment succeeded**

Deployment 'dxcorg231_dxcdb231' to resource group 'dxcorg231' was successful.

[Go to resource](#) [Pin to dashboard](#)

Step-4: After clicking the go to resource button you are navigated to the Data Bricks.

Home > **dxcdb231** ⚙️ ☆ ...

Search (Ctrl+/) < Delete

Overview Essentials JSON View

Activity log

Access control (IAM)

Tags

Settings

Virtual Network Peerings

Encryption

Properties

Locks

Automation

Tasks (preview)

Export template

Support + troubleshooting

New Support Request

Status : Active

Resource group : dxcorg231

Location : East US

Subscription : Azure-DXC262AB12Lab

Subscription ID : 48ef8049-33b2-4e37-bec3-5f6394096bd2

Tags [edit](#) [Click here to add tags](#)

Managed Resource Group : databricks-rg-dxcdb231-gpoon7jsh2zg

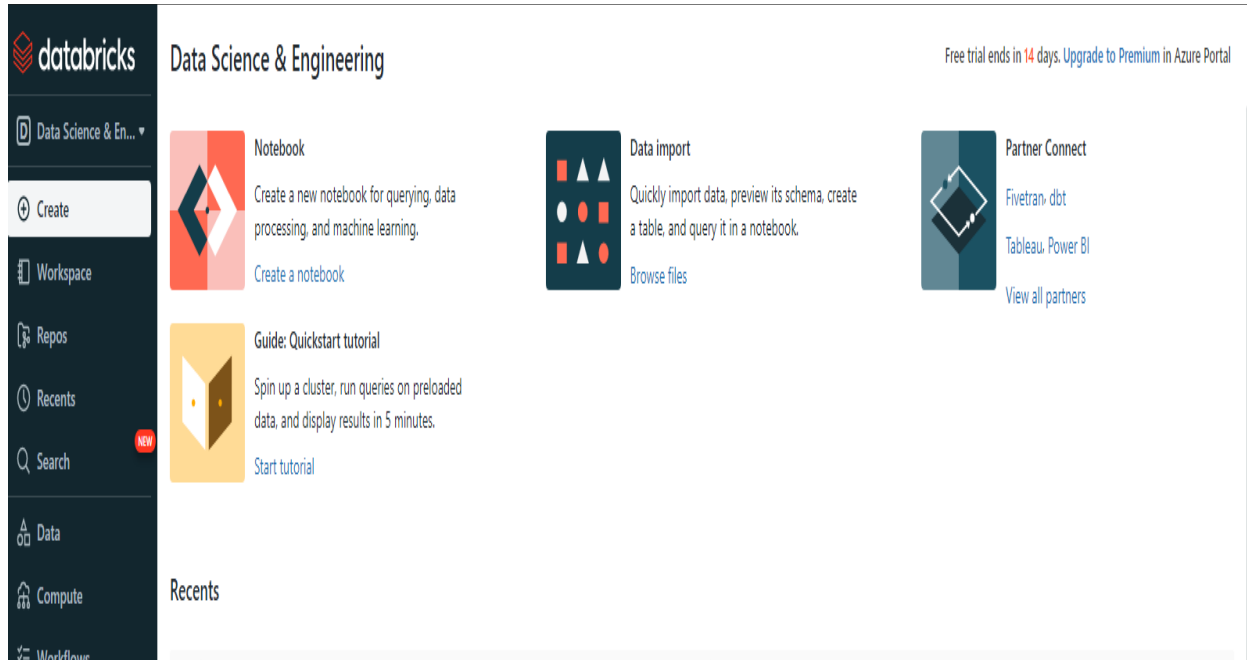
URL : <https://adb-7548927456012596.16.azuredatabricks.net>

Pricing Tier : Trial (Premium - 14-Days Free DBUs)

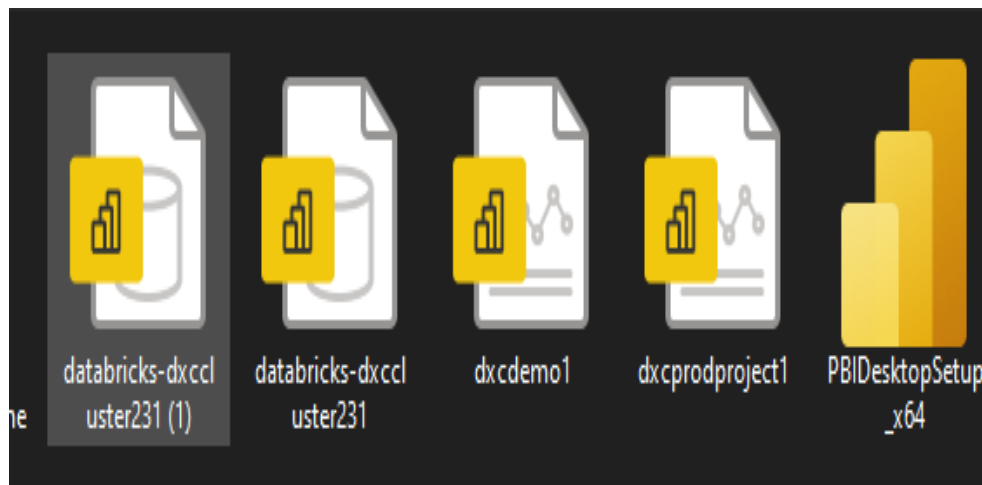
Launch Workspace

[Upgrade to Premium](#)

Step-5: After clicking on launch workspace, you are in Data Bricks.



Step-6: After that open user settings and generate a token . After that, go to tables and click on partner connect and select the “**power BI**” and attach the cluster and download the file as shown below.



4. Create Azure Synapse & connect with Azure Blob, explain the steps with screenshots.

Ans: Step-1: Create an “Azure Synapse Account”.

The screenshot shows the 'Create Synapse workspace' wizard in the Azure portal. The 'Basics' tab is selected. The form includes the following fields and options:

- Subscription:** Azure-DXC262AB12Lab
- Resource group:** (New) DXCRG23178, with a 'Create new' link below.
- Managed resource group:** Enter managed resource group name, with a dropdown arrow.
- Workspace name:** dxcsynapse231789
- Region:** East US
- Select Data Lake Storage Gen2:** From subscription (selected) or Manually via URL.
- Account name:** (New) dxcdatalakeaccount, with a 'Create new' link below.
- File system name:** (New) dxdfs231789, with a 'Create new' link below.

Below the form, there is a warning box with an orange triangle icon:

Additional configuration is required. After you create your workspace, perform these tasks:

- Assign other users to the **Contributor** role on workspace
- Assign other users the appropriate **Synapse RBAC roles** using Synapse Studio

Contact an **Owner** of the storage account, and ask them to perform the following tasks:

At the bottom, there are three buttons: 'Review + create' (highlighted in yellow), '< Previous', and 'Next: Security >'.

Step-2: Wait for the Deployment.

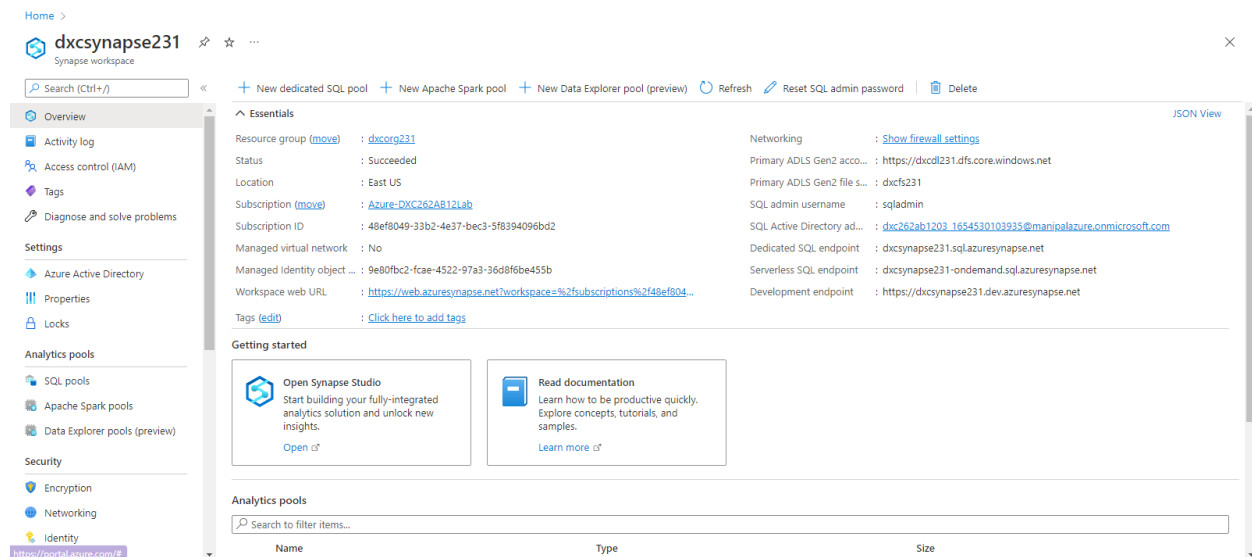
The screenshot shows the 'Deployment' overview page in the Azure portal. The deployment is successful, and the page displays the following information:

- Deployment name:** dxcdb231_1655450738326
- Subscription:** Azure-DXC262AB12Lab
- Resource group:** dxccorg231
- Start time:** 6/17/2022, 12:55:48 PM
- Correlation ID:** b6c78043-98df-45be-a516-fd568864fd07

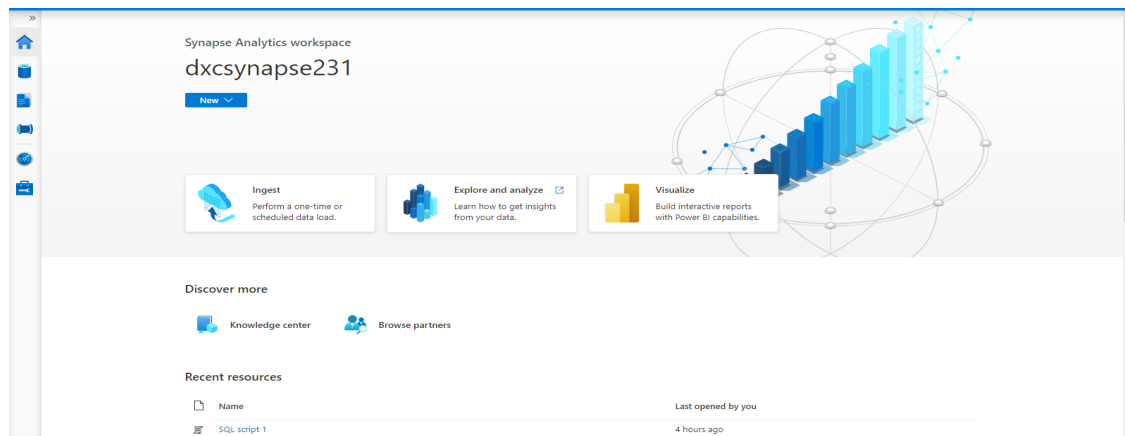
The page also includes a 'Go to resource' button and a 'Next steps' section with a 'Go to resource' button. On the right side, there are several informational cards:

- 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 >](#)

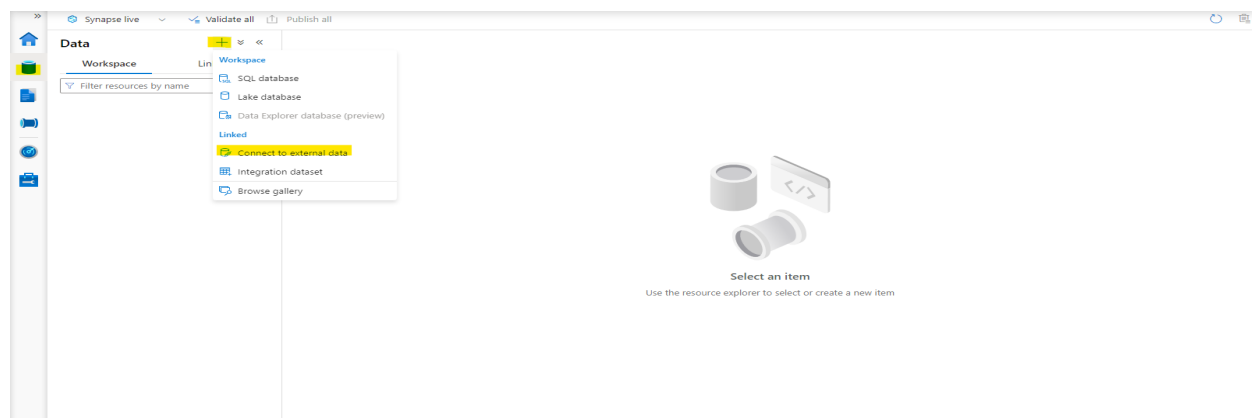
Step-3: Click on goto resource and navigate to the synapse studio and click on open.



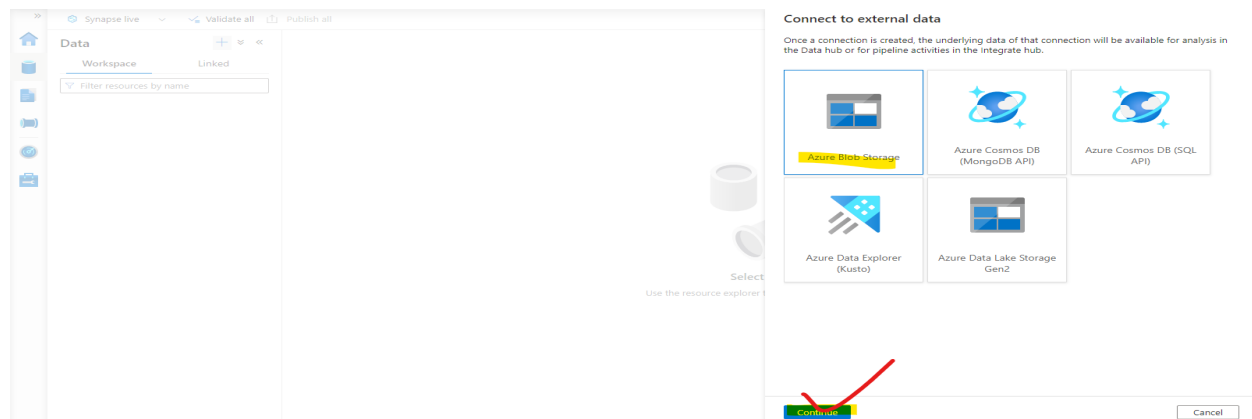
Step-4: This page will appear.



Step-5: Click on data after that click on connect external data as shown in the screen.



Step-6: Click on azure blob storage and click on create.



Step-7: New linked services page will open give the info required and check connection and later click on create.

New linked service
Azure Blob Storage [Learn more](#)

Choose a name for your linked service. This name cannot be updated later.

Name *
AzureBlobStoragee2

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
Azure-DXC262AB12Lab (48ef8049-33b2-4e37-bec3-5f8394096bd2)

Storage account name *
dxcdb231

Additional connection properties
+ New

Create Back

Connection successful
Test connection Cancel

Step-8: After that the blob storage is connected successfully.

The screenshot shows the Synapse Studio interface. On the left, the 'Data' pane is open, showing a list of resources. The 'Linked' section is expanded, showing 'Azure Blob Storage (dxcdb231)'. The 'source' folder is selected, and its contents are displayed in a table.

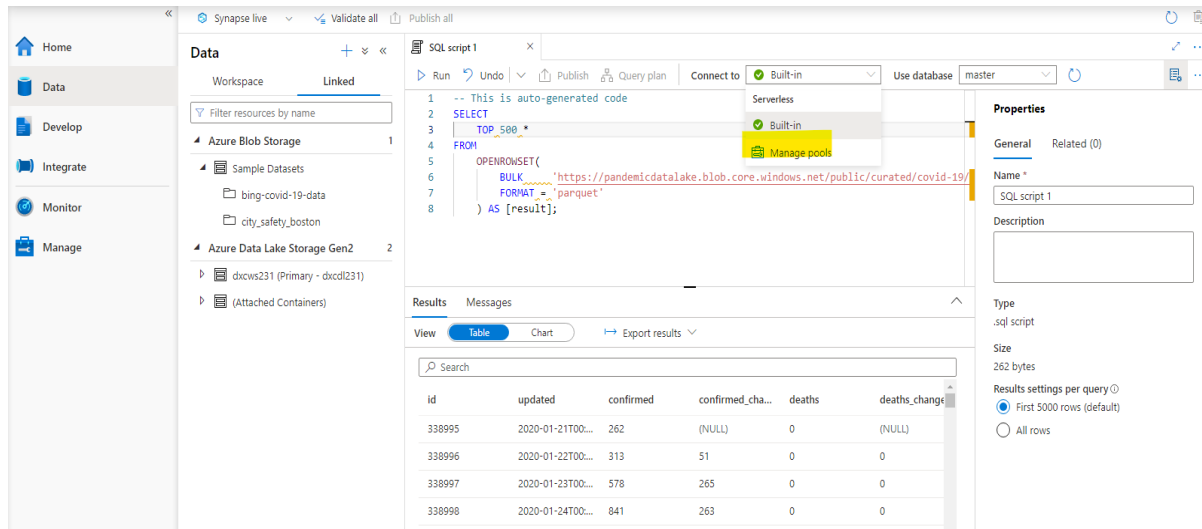
Name	Access Tier	Access Tier Last Modified	Last Modified	Blob Type	Content Type	Size	Status	Remaining Days	Deleted Time	Lease Stat
ass-5.csv	Hot		6/17/2022, 2:56:19 PM	Block Blob	application/octet-stream	271.3 KB	Active			
circuits.csv	Hot		6/17/2022, 12:59:45 PM	Block Blob	text/csv	9.8 KB	Active			
constructors.json	Hot		6/17/2022, 12:59:11 PM	Block Blob	application/json	29.7 KB	Active			
pit_stops.json	Hot		6/17/2022, 1:00:08 PM	Block Blob	application/json	1.3 MB	Active			
races.csv	Hot		6/17/2022, 12:59:45 PM	Block Blob	text/csv	114.1 KB	Active			

Showing 1 to 5 of 5 cached items

5. Create Azure Synapse spark pool & query sample sample JSON file, explain the steps with screenshots.

Ans: To create a spark pool we have to follow the steps mentioned below.

Step-1: Click on manage pools in Synapse.



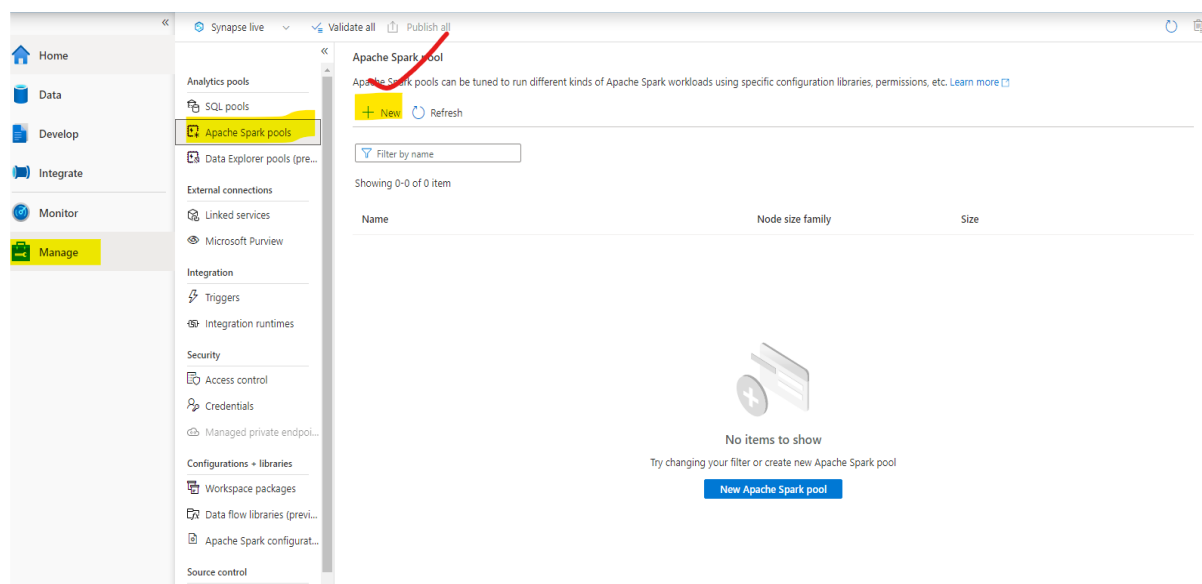
The screenshot shows the Azure Synapse Studio interface. On the left, the 'Data' pane is expanded, showing 'Azure Blob Storage' and 'Azure Data Lake Storage Gen2'. The main pane displays a SQL script in a text editor. The script is as follows:

```
1 -- This is auto-generated code
2 SELECT
3   TOP 500 *
4 FROM
5   OPENROWSET(
6     BULK 'https://pandemicdatalake.blob.core.windows.net/public/curated/covid-19/
7     FORMAT = 'parquet'
8   ) AS [result];
```

The 'Connect to' dropdown is set to 'Built-in', and the 'Use database' is set to 'master'. The 'Results' pane shows a table with the following data:

id	updated	confirmed	confirmed_cha...	deaths	deaths_change
338995	2020-01-21T00:...	262	(NULL)	0	(NULL)
338996	2020-01-22T00:...	313	51	0	0
338997	2020-01-23T00:...	578	265	0	0
338998	2020-01-24T00:...	841	263	0	0

Step-2: After that it navigates to the manage page and selects spark pool there, refer screenshot.



The screenshot shows the 'Manage' page in Azure Synapse Studio. The 'Apache Spark pools' section is selected, and the 'New' button is highlighted. The 'Filter by name' field is empty. The page shows 'Showing 0-0 of 0 item' and a 'No items to show' message. The 'New Apache Spark pool' button is visible at the bottom.

Step-3: Fill all the requirements and click on review and create.

New Apache Spark pool

Basics * Additional settings * Tags Review + create

Create an 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 *

Node size family *

Node size *

Autoscale * ☐ Enabled ☒ Disabled


Number of nodes *

Estimated price

Dynamically allocate executors * ☐ Enabled ☒ Disabled

Step-4 : It takes a few minutes to Deploy.

Notifications

 **Deploying**

Deploying dxcap231 (Apache Spark pool) to the workspace. This can take 1-2 minutes.

a few seconds ago

Step-5: And Done.

Apache Spark pools

Showing 1-1 of 1 item

Name
dxcap231

Step-6: Now querying a sample json file. Attach to the pool and click on run all. It takes some time.

The screenshot shows the Synapse Notebook 1 interface. The notebook is attached to the 'dxcap' pool. The code cell contains the following PySpark code:

```
1 %%pyspark
2 blob_account_name = "dxcdb231"
3 blob_container_name = "source"
4 from pyspark.sql import SparkSession
5
6 sc = SparkSession.builder.getOrCreate()
7 token_library = sc._jvm.com.microsoft.azure.synapse.tokenlibrary.TokenLibrary
8 blob_sas_token = token_library.getConnectionString("AzureBlobStorage1")
9
10 spark.conf.set(
11     'fs.azure.sas.%s.%s.blob.core.windows.net' % (blob_container_name, blob_account_name),
12     blob_sas_token)
13 df = spark.read.load('wasbs://source@dxcdb231.blob.core.windows.net/constructors.json', format='json')
14 display(df.limit(10))
```

The status bar indicates "Please wait a few minutes while your session starts."

Step-7: The data will be queried successfully as shown in the screen.

The screenshot shows the Synapse Notebook 1 interface after successful execution. The status bar indicates "Ready". The code cell shows the same PySpark code as in Step 6. The output shows a table of constructor data:

```
[1] ✓ 4 min 31 sec - Apache Spark session started in 4 min 5 sec 507 ms. Command executed in 24 sec 3 ms by dxc262ab1203_1654530103935 on 5:08:56 PM, 6/17/22
```

Job execution Succeeded Spark 2 executors 8 cores

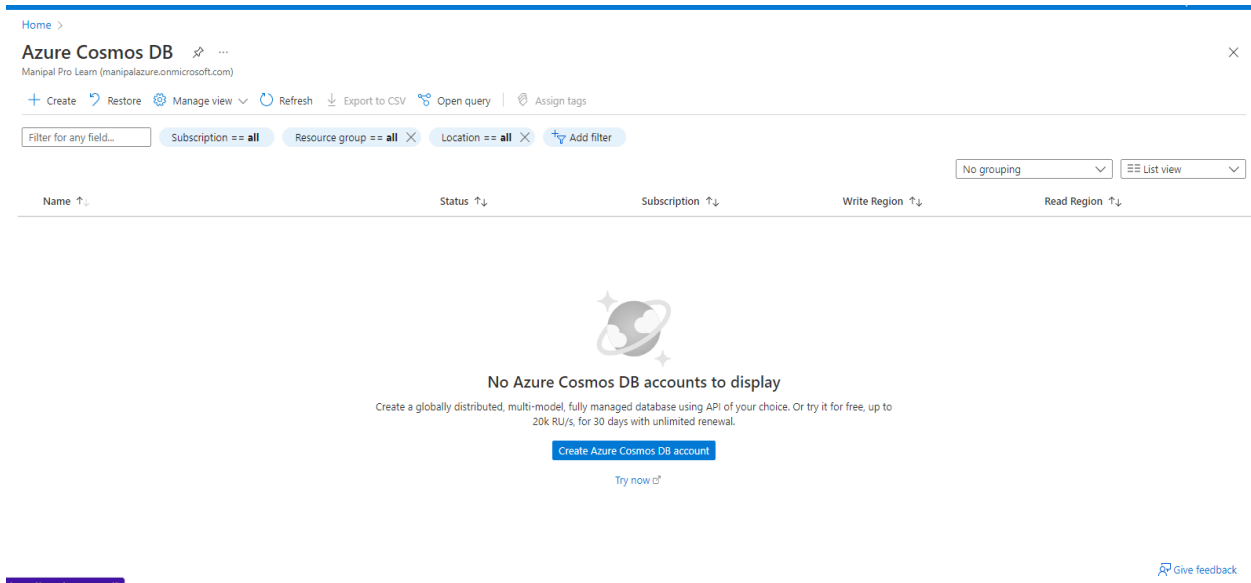
View: Table Chart Export results

constructorid	constructorRef	name	nationality	url
1	mclaren	McLaren	British	http://en.wikipedia.org/wiki/McL...
2	bmw_sauber	BMW Sauber	German	http://en.wikipedia.org/wiki/BM...
3	williams	Williams	British	http://en.wikipedia.org/wiki/Willi...
4	renault	Renault	French	http://en.wikipedia.org/wiki/Ren...
5	toro_rosso	Toro Rosso	Italian	http://en.wikipedia.org/wiki/Scu...
6	ferrari	Ferrari	Italian	http://en.wikipedia.org/wiki/Scu...
7	toyota	Toyota	Japanese	http://en.wikipedia.org/wiki/Toyo...

6.Create Azure Cosmos DB & import sample JSON file, explain the steps with screenshots.

Ans: To create Azure cosmos DB we need to follow the below mentioned steps.

Step-1: Go to azure portal and search for “Azure cosmos DB”.



Step-2: We have to select the API option and we are recommended with core sql.

Home > Azure Cosmos DB >

Select API option

Which API best suits your workload?

Azure Cosmos DB is a fully managed NoSQL database service for building scalable, high performance applications. [Learn more](#)

To start, select the API to create a new account. The API selection cannot be changed after account creation.

Core (SQL) - Recommended Azure Cosmos DB's core, or native API for working with documents. Supports fast, flexible development with familiar SQL query language and client libraries for .NET, JavaScript, Python, and Java. Create Learn more	Azure Cosmos DB API for MongoDB Fully managed database service for apps written for MongoDB. Recommended if you have existing MongoDB workloads that you plan to migrate to Azure Cosmos DB. Create Learn more	Cassandra Fully managed Cassandra database service for apps written for Apache Cassandra. Recommended if you have existing Cassandra workloads that you plan to migrate to Azure Cosmos DB. Create Learn more
Azure Table Fully managed database service for apps written for Azure Table storage. Recommended if you have existing Azure Table storage workloads that you plan to migrate to Azure Cosmos DB, but do not want to re-write your application to use the SQL API. Create Learn more	Gremlin (Graph) Fully managed graph database service using the Gremlin query language, based on Apache TinkerPop project. Recommended for new workloads that need to store relationships between data. Create Learn more	

Step-3: We have to fill in the following details.

Home > Azure Cosmos DB > Select API option >

Create Azure Cosmos DB Account - Core (SQL)

Basics Global Distribution Networking Backup Policy Encryption Tags Review + create

Azure Cosmos DB is a fully managed NoSQL database service for building scalable, high performance applications. [Try it for free](#), for 30 days with unlimited renewals. Go to production starting at \$24/month per database, multiple containers included. [Learn more](#)

Project Details

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

Subscription * Azure-DXC262AB12Lab

Resource Group * dxcorg231
[Create new](#)

Instance Details

Account Name * dxccosmosdb231

Location * (US) West US

Capacity mode ☐ Provisioned throughput ☒ Serverless
[Learn more about capacity mode](#)

[Review + create](#) [Previous](#) [Next: Global Distribution](#)

Step-4: After successful validation click on create.

Home > Azure Cosmos DB > Select API option >

Create Azure Cosmos DB Account - Core (SQL)

Basics Global Distribution Networking Backup Policy Encryption Tags **Review + create**

Validation Success

Creation Time

Estimated Account Creation Time (in minutes) 2

The estimated creation time is calculated based on the location you have selected

Basics

Subscription Azure-DXC262AB12Lab

Resource Group dxcorg231

Location West US

Account Name (new) dxccosmosdb231

API Core (SQL)

Capacity mode Serverless

Backup Policy

Backup policy Periodic

Backup storage redundancy Geo-redundant backup storage

[Create](#) [Previous](#) [Next](#) [Download a template for automation](#)

Step-5: Wait for the deployment to complete.

Home >

Microsoft.Azure.CosmosDB-20220617172353 | Overview

Deployment

Search (Ctrl+F) Delete Cancel Redeploy Refresh

We'd love your feedback →

=== Deployment is in progress

Deployment name: Microsoft.Azure.CosmosDB-20220617172353 Start time: 6/17/2022, 5:24:03 PM
Subscription: Azure-DXC262AB12Lab Correlation ID: 42e1ad9c-f540-4314-adf1-6c7e4e80dde4
Resource group: dxcorg231

Deployment details (Download)

Resource	Type	Status	Operation details
dxccosmosdb231	Microsoft.DocumentDb/databaseAccounts	OK	Operation details

Step-6: After completion of deployment click on go to resource.

The screenshot shows the Azure portal interface for a deployment. The breadcrumb navigation at the top reads "Home > Microsoft.Azure.CosmosDB-20220617172353 | Overview". Below this, there's a search bar and action buttons: "Delete", "Cancel", "Redeploy", and "Refresh". A left sidebar contains links for "Overview", "Inputs", "Outputs", and "Template". The main content area has a purple banner that says "We'd love your feedback". Below this, a green checkmark icon indicates "Your deployment is complete". The deployment details show: "Deployment name: Microsoft.Azure.CosmosDB-20220617172353", "Subscription: Azure-DXC262AB12Lab", "Resource group: dxccorg231", "Start time: 6/17/2022, 5:24:03 PM", and "Correlation ID: 42e1ad9c-f540-4314-adf1-6c7e4e80dde4". There are links for "Deployment details (Download)" and "Next steps". A prominent blue button labeled "Go to resource" is at the bottom of the deployment details. On the right, there are three promotional cards: "Cost Management" (Get notified to stay within your budget), "Microsoft Defender for Cloud" (Secure your apps and infrastructure), and "Free Microsoft tutorials" (Start learning today).

Step-7: After going to the cosmos DB and follow the below mentioned steps.

The screenshot shows the Azure Cosmos DB Data Explorer interface. The breadcrumb navigation at the top reads "Home > Microsoft.Azure.CosmosDB-20220617172353 > dxccosmosdb231". Below this, there's a search bar and action buttons: "New Container", "Enable Azure Synapse Link", "New Notebook", and "Connect to GitHub". A left sidebar contains links for "Overview", "Activity log", "Access control (IAM)", "Tags", "Diagnose and solve problems", "Quick start", "Notifications", and "Data Explorer". The main content area has a "Welcome to Cosmos DB" header with the tagline "Globally distributed, multi-model database service for any scale". Below this, there are three main action cards: "Launch quick start" (Launch a quick start tutorial to get started with sample data), "New Container" (Create a new container for storage and throughput), and "Connect" (Prefer using your own choice of tooling? Find the connection string you need to connect). At the bottom, there are three sections: "Recents", "Top 3 things you need to know" (with links for "Advanced Modeling Patterns", "Partitioning Best Practices", and "Migrate Your Data"), and "Learning Resources" (with links for "Get Started using an SDK", "Master Complex Queries", and "Migrate Your Data").

Step-8: Create the container.

New Container

✕

★ Container id ⓘ

source

★ Partition key ⓘ

For small workloads, the item ID is a suitable choice for the partition key.

/example

Unique keys ⓘ

+ Add unique key

Analytical store ⓘ

☐ On ☒ Off

Azure Synapse Link is required for creating an analytical store container. Enable Synapse Link for this Cosmos DB account. [Learn more](#)

Enable

▼ Advanced

☐ My partition key is larger than 101 bytes

OK

Step-8: Upload the data as shown below screen and After clicking on execute query then the data will be queried.

Home > Microsoft Azure Cosmos DB - 20220617172353 > dxccosmosdb231

dxccosmosdb231 | Data Explorer

Azure Cosmos DB account

Search (Ctrl+/)

- Overview
- Activity log
- Access control (IAM)
- Tags
- Diagnose and solve problems
- Quick start
- Notifications
- Data Explorer

Settings

- Features
- Default consistency
- Backup & Restore
- Firewall and virtual networks
- Private Endpoint Connections
- CORS
- Dedicated Gateway
- Keys
- Advisor Recommendations

SQL API

source - Items

Settings

Query 1

1 SELECT * FROM c

Results

Query Stats

1 - 100 | Load more

NOTEBOOKS

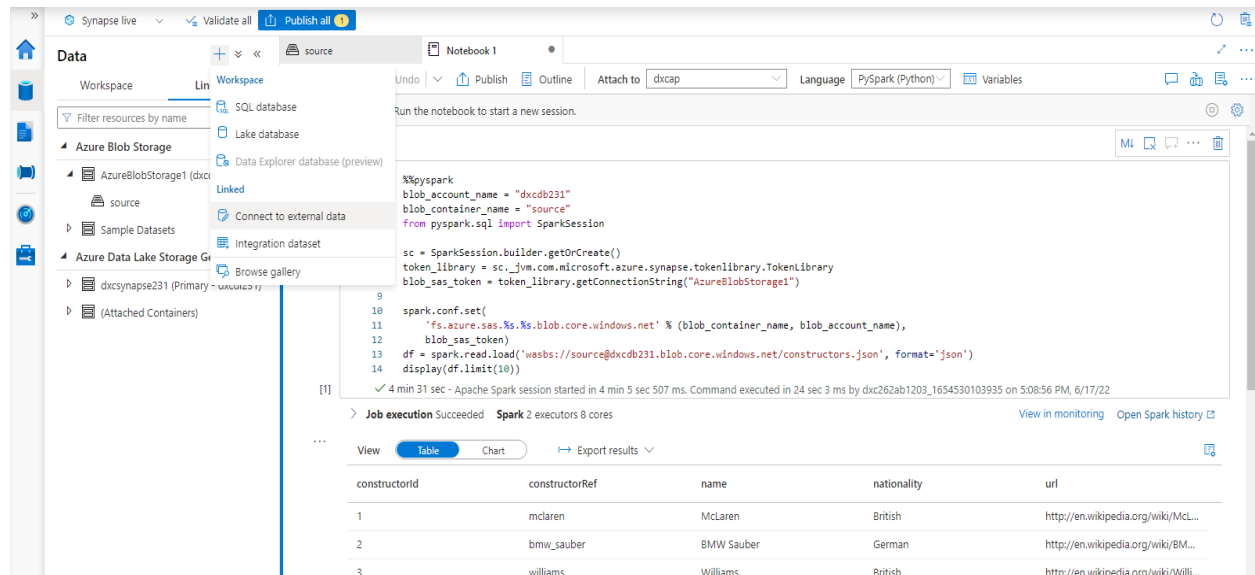
Notebooks is currently not available. We are working on it.

```
{
  "traceId": "841",
  "driverId": "153",
  "stop": 1,
  "lap": 1,
  "time": "17:05:23",
  "duration": 26.898,
  "milliseconds": 26898,
  "id": "fc258938-6684-4ac2-979f-6a41b9163bcc",
  "rid": "fmgSAOJIIIBAAAAA",
  "self": "dbs/fmgSAOJIIIBAAAAA/colls/fmgSAOJIIIBAAAAA",
  "etag": "\"14608a833-0000-0700-0000-62ac6fbc0000\"",
  "attachments": "attachments/",
  "ts": 1655467964
},
{
  "traceId": "841",
```

7. Connect COSMOS DB & Azure Synapse analytics & explain the steps with screenshots.

Ans: To connect the Cosmos DB and Azure synapse we have to follow the below mentioned steps.

Step-1: Open azure synapse and click on data and click on + icon and select connect external data.



The screenshot shows the Azure Synapse Studio interface. On the left, the 'Data' pane is open, showing a workspace with 'Azure Blob Storage' and 'Azure Data Lake Storage Gen2'. The 'Connect to external data' option is selected. The main area displays a notebook with the following code:

```
%pyspark
blob_account_name = "dxcdb231"
blob_container_name = "source"
from pyspark.sql import SparkSession

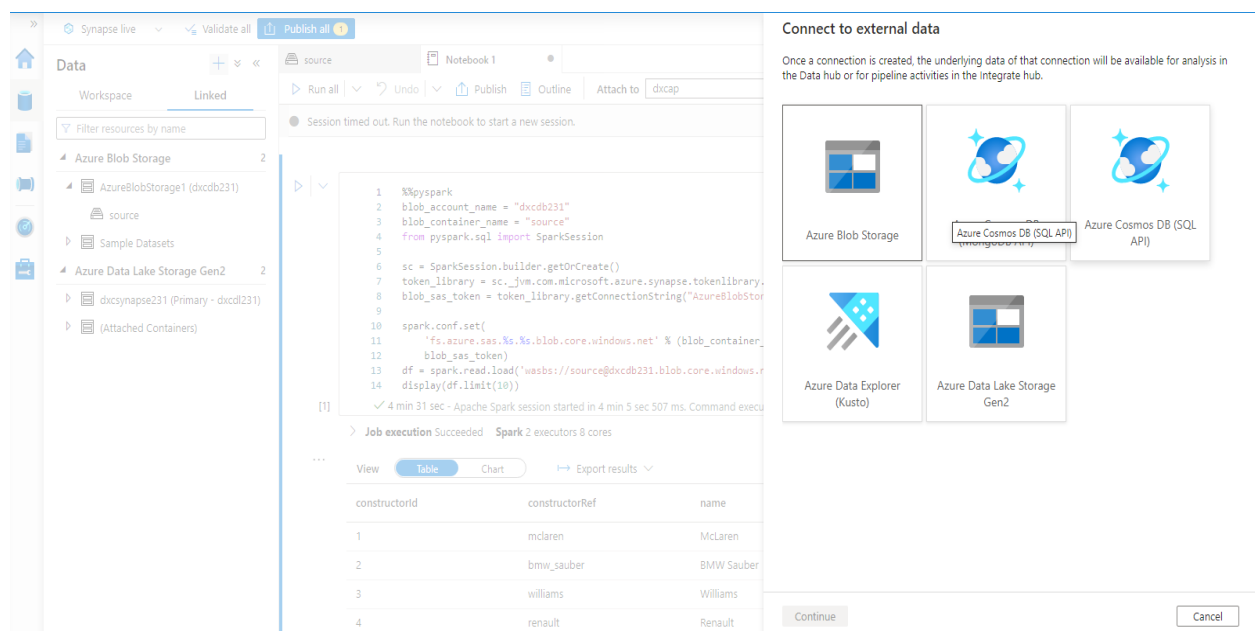
sc = SparkSession.builder.getOrCreate()
token_library = sc._jvm.com.microsoft.azure.synapse.tokenlibrary.TokenLibrary
blob_sas_token = token_library.getConnectionString("AzureBlobStorage1")

spark.conf.set(
    'fs.azure.sas.%s.%s.blob.core.windows.net' % (blob_container_name, blob_account_name),
    blob_sas_token)
df = spark.read.load("wasbs://source@dxcdb231.blob.core.windows.net/constructors.json", format='json')
display(df.limit(10))
```

The notebook output shows a successful job execution and a table of constructor data:

constructorid	constructorRef	name	nationality	url
1	mclaren	McLaren	British	http://en.wikipedia.org/wiki/McL...
2	bmw_sauber	BMW Sauber	German	http://en.wikipedia.org/wiki/BM...
3	williams	Williams	British	http://en.wikipedia.org/wiki/Willi...

Step-2: Then select the cosmos DB SQL API.



The screenshot shows the 'Connect to external data' dialog in Azure Synapse Studio. The dialog lists several data sources, and 'Azure Cosmos DB (SQL API)' is selected. The background shows the same notebook as in Step 1, but the code is not yet executed.

Connect to external data

Once a connection is created, the underlying data of that connection will be available for analysis in the Data hub or for pipeline activities in the integrate hub.

Azure Blob Storage

Azure Cosmos DB (SQL API)

Azure Cosmos DB (SQL API)

Azure Data Explorer (Kusto)

Azure Data Lake Storage Gen2

Continue

Cancel

Step-3: We have to fill the below mentioned fields.

New linked service
Azure Cosmos DB (SQL API) [Learn more](#)

Choose a name for your linked service. This name cannot be updated later.

Name *
CosmosDb1

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
Azure-DXC262AB12Lab (48ef8049-33b2-4e37-bec3-5f8394096bd2)

Azure Cosmos DB account name *
dxccosmosdb231

Database name *
dxcdb231

Create **Back**

Connection successful **Test connection** **Cancel**

Step-4: After that cosmos DB is successfully connected with synapse.

Synapse live [Validate all](#) [Publish all](#)

Data [Workspace](#) [Linked](#)

Filter resources by name

- Azure Blob Storage 2
 - AzureBlobStorage1 (dxcdb231)
- Sample Datasets
- Azure Cosmos DB 1
 - CosmosDb1 (dxcdb231)
 - source
- Azure Data Lake Storage Gen2 2
 - dxcsynapse231 (Primary - dxcdl231)
 - (Attached Containers)

Ready

For Spark 3 Cosmos DB connector has slightly different configuration. Please select different snippet if a Spark 3.1+ connector is attached.

Code **Markdown**

```
1 # Write a Spark DataFrame into a Cosmos DB container
2 # To select a preferred list of regions in a multi-region Cosmos DB account, add .option("spark.cosmos.regions", "us-east-1")
3
4 # For Spark 2.4
5 YOURDATAFRAME.write()
6   .format("cosmos.oltp")\
7   .option("spark.synapse.linkedService", "CosmosDb1")\
8   .option("spark.cosmos.container", "source")\
9   .option("spark.cosmos.write.upsertEnabled", "true")\
10  .mode("append")\
11  .save()
12
13 # For Spark 3.1 +
14 # YOURDATAFRAME.write()
15 #   .format("cosmos.oltp")\
16 #   .option("spark.synapse.linkedService", "CosmosDb1")\
17 #   .option("spark.cosmos.container", "source")\
18 #   .option("spark.cosmos.write.strategy", "ItemOverwrite")\
19 #   .option("spark.cosmos.write.bulk.enabled", "true")\
20 #   .mode("append")\
21 #   .save()
```

Properties

General **Related (0)**

Name *
Notebook 1

Description

Type
.ipynb notebook

Size
1,988 bytes

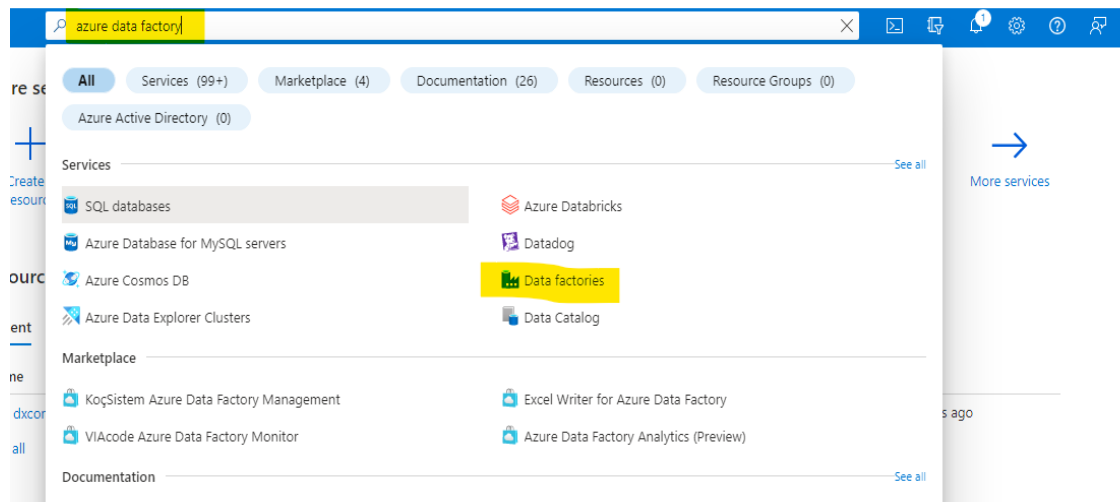
Notebook settings
☒ Include cell output when saving
☐ Enable unpublished notebook reference

Session
[Configure session](#)

8. Create azure Data factory & azure Blob, connect Blob & ADF, import blob files into Data factory & explain the steps with screenshots.

Ans: Azure data factory can be created by using the following steps

Step-1: Login to the azure portal and search for azure data factory as shown in figure.



Step-2: After navigating to the azure data factory page click on create and select the resource group, name, region, version after that click on next follow the figure.

A screenshot of the 'Basics' configuration page for creating a new Azure Data Factory. The page has a navigation bar with tabs: 'Basics', 'Git configuration', 'Networking', 'Advanced', 'Tags', and 'Review + create'. The 'Basics' tab is selected. Under the 'Project details' section, there is a description: 'Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.' Below this, there are two dropdown menus: 'Subscription' (set to 'Azure-DXC262AB12Lab') and 'Resource group' (set to 'dxccorg231'). A 'Create new' link is visible below the 'Resource group' dropdown. Under the 'Instance details' section, there are three dropdown menus: 'Name' (set to 'dxcdf231'), 'Region' (set to 'East US'), and 'Version' (set to 'V2 (Recommended)'). At the bottom of the page, there are three buttons: 'Review + create', '< Previous', and 'Next : Git configuration >'. The 'Next : Git configuration >' button is highlighted with a yellow box.

Step-3: Click on git configurations and checkbox it as configure git later as shown in figure.

[Home](#) > [Data factories](#) >

Create Data Factory ...

Basics **Git configuration** Networking Advanced Tags Review + create

Azure Data Factory allows you to configure a Git repository with either Azure DevOps or GitHub. Git is a version control system that allows for easier change tracking and collaboration.

[Learn more about Git integration in Azure Data Factory](#)

Configure Git later ⓘ



Step-4: Go through the next steps followed by successful completion of validation click on create as shown in figure.

[Home](#) > [Data factories](#) >

Create Data Factory ...

✓ Validation Passed

Basics **Git configuration** Networking Advanced Tags **Review + create**

TERMS

By clicking "Create", I (a) agree to the legal terms and privacy statement(s) associated with the Marketplace offering(s) listed above; and (b) 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. See the [Azure Marketplace Terms](#) for additional details.

Basics

Subscription	Azure-DXC262AB12Lab
Resource group	dxccorg231
Name	dxcdf231
Region	East US
Version	V2 (Recommended)

Networking

Connect via	Public endpoint
-------------	-----------------

Create


< Previous


Next

[Download a template for automation](#)

Step-5: After clicking on create it takes some time for deployment after deployment you can see the following as shown in figure.

✓ Your deployment is complete

 Deployment name: Microsoft.DataFactory-20220610162331
Subscription: [Azure-DXC262AB12Lab](#)
Resource group: [dxcorg231](#)

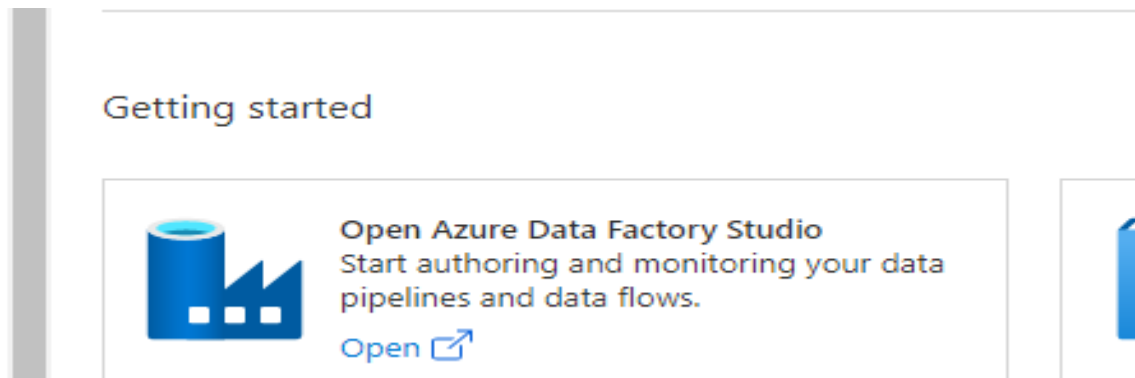
Start time: 6/10/2022, 4:31:58 PM
Correlation ID: 4b58f579-505b-44a3-9760-552ded084f0b 

▼ Deployment details [\(Download\)](#)

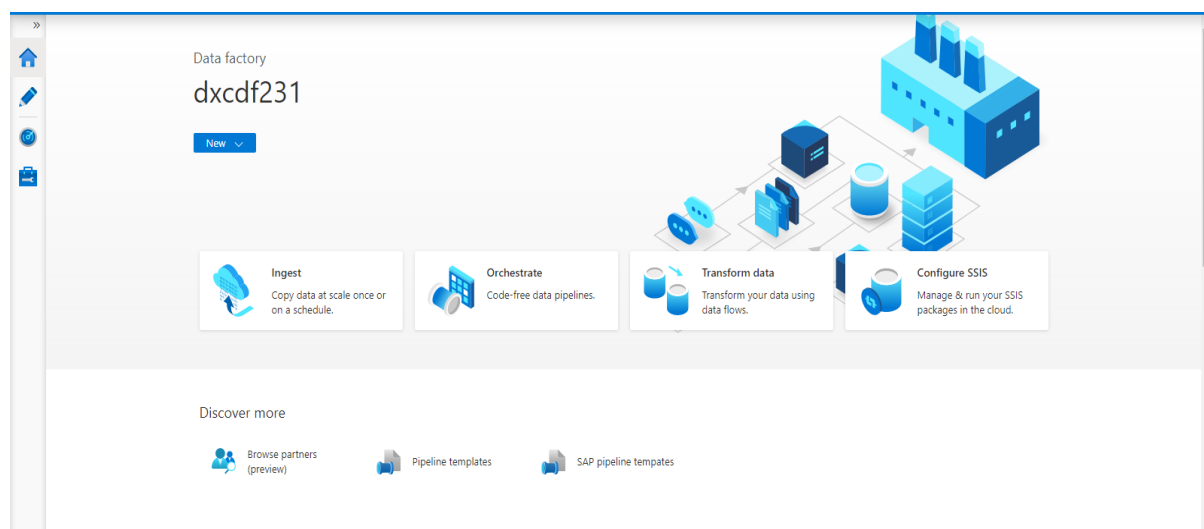
^ Next steps

[Go to resource](#)

Step-6: Click on go to resources after that click on open as shown in the figure.



Step-7: After clicking over it will open the azure data factory in a new window as mentioned in figure and we can use this creation of pipelines.



Step-8 : We are moving the data from the source to the destination using the copy data tool and creating a pipeline.

The screenshot shows the 'Copy Data tool' configuration interface. On the left, a vertical navigation pane lists steps: Properties, Source, Target, Settings, Review and finish (highlighted), Review, and Deployment. The main area is titled 'Copy Data tool' and contains a 'Summary' section with a diagram showing data flow from 'Azure Blob Storage' to 'Azure Blob Storage'. Below this is a 'Properties' section with fields for Task name (pipeline1), Task description, Source (Connection name: AzureBlobStorage1, Dataset name: SourceDataset_v7x, Column delimiter: , , Row delimiter, Escape character: \, Quote char: ", First row as header: true), and File name (1000 Companies.csv). At the bottom are '< Previous' and 'Next >' buttons.

Step-9: The validation and deployment is done and the pipeline is created successfully.

The screenshot shows the 'Copy Data tool' deployment completion screen. The left navigation pane is the same as in Step 8, but 'Deployment' is now highlighted. The main area is titled 'Copy Data tool' and features a 'Deployment complete' section with a diagram showing data flow from 'Azure Blob Storage' to 'Azure Blob Storage'. Below this is a table showing the deployment steps and their status:

Deployment step	Status
Validating copy runtime environment	✓ Succeeded
> Creating datasets	✓ Succeeded
> Creating pipelines	✓ Succeeded
> Running pipelines	✓ Succeeded

Below the table, a message states: 'Datasets and pipelines have been created. You can now monitor and edit the copy pipelines or click finish to close Copy Data Tool.' At the bottom, there is a 'Finish' button (highlighted with a red checkmark), an 'Edit pipeline' button, and a 'Monitor' button.

Step-10: The data is successfully copied from source to destination.


Authentication method: Access key ([Switch to Azure AD User Account](#))

Location: source

Search blobs by prefix (case-sensitive)

☒ Show deleted blobs

+ Add filter

	Name	Modified	Access tier
<input type="checkbox"/>	 1000_Companies.csv	6/10/2022, 5:01:31 PM	Hot (Inferred)


Authentication method: Access key ([Switch to Azure AD User Account](#))

Location: destination

Search blobs by prefix (case-sensitive)

☒ Show deleted blobs

+ Add filter

	Name	Modified	Access tier
<input type="checkbox"/>	 1000_Companies.txt	6/10/2022, 5:13:30 PM	Hot (Inferred)