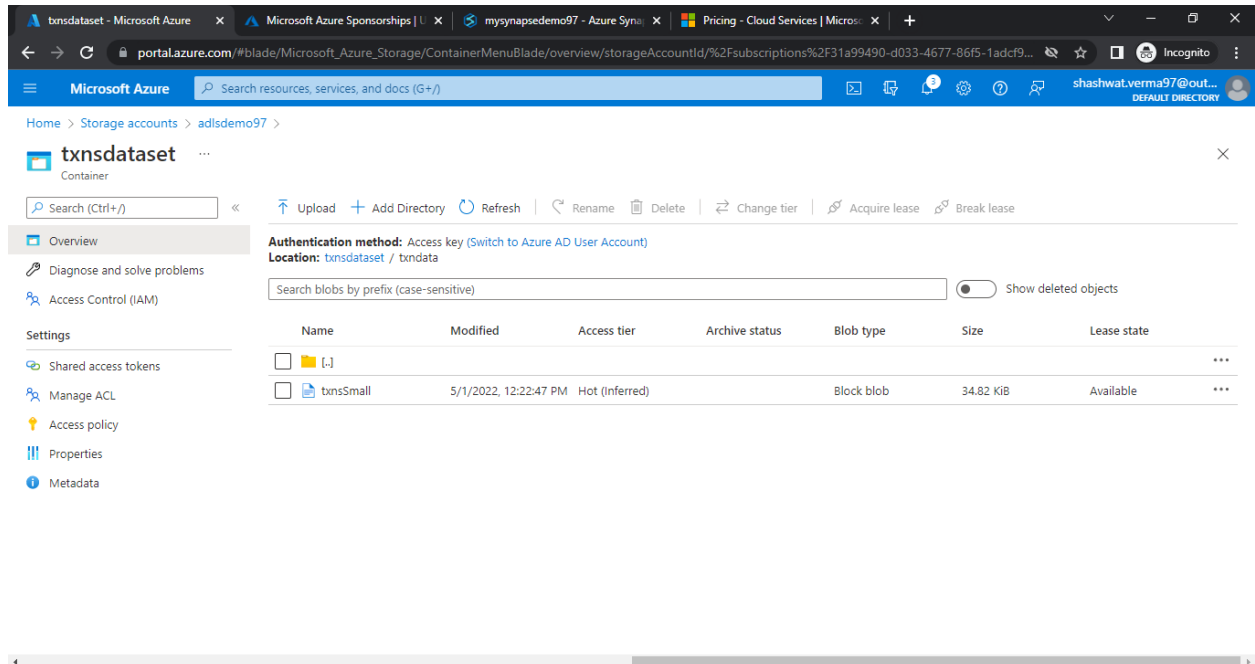


# Assignment 1: Azure Synapse Analytics

*In the 'adlsdemo97' storage account we created a container named 'txnsdataset'. Inside txns data set we created a directory 'txndata' where we uploaded 'txnsSmall' file as shown in screenshot below:*



## Source Code:

```
-- First we need to create a database in the serverless pool
CREATE DATABASE [walmart]

-- Ensure to switch the context to the new database first
USE [walmart]

-- Here we are creating a database master key. This key will be
used to protect the Shared Access Signature which is specified
in the next step
CREATE MASTER KEY ENCRYPTION BY PASSWORD = 'P@ssw0rd@123';
```

```
-- Here we are using the Shared Access Signature to authorize
the use of the Azure Data Lake Storage account
CREATE DATABASE SCOPED CREDENTIAL SasToken
WITH IDENTITY='SHARED ACCESS SIGNATURE'
, SECRET =
'?sv=2020-08-04&ss=bfqt&srt=sco&sp=rwdlacupx&se=2022-05-31T15:49
:55Z&st=2022-05-01T07:49:55Z&spr=https&sig=a%2FTrdAsJ0kGYldFAqi3
yXoA0soWRUDYcdtUfpjMEpdU%3D';
```

```
-- This defines the source of the data.
CREATE EXTERNAL DATA SOURCE txn_data
WITH ( LOCATION =
'https://adlsdemo97.dfs.core.windows.net/txnsdataset',
CREDENTIAL = SasToken
)
```

/\* This creates an External File Format object that defines the external data that can be present in Hadoop, Azure Blob storage or Azure Data Lake Store

Here with FIRST\_ROW, we are saying please skip the first row because this contains header information

Ref:

<https://docs.microsoft.com/en-us/sql/t-sql/statements/create-external-file-format-transact-sql?view=sql-server-ver15&tabs=delimited>

\*/

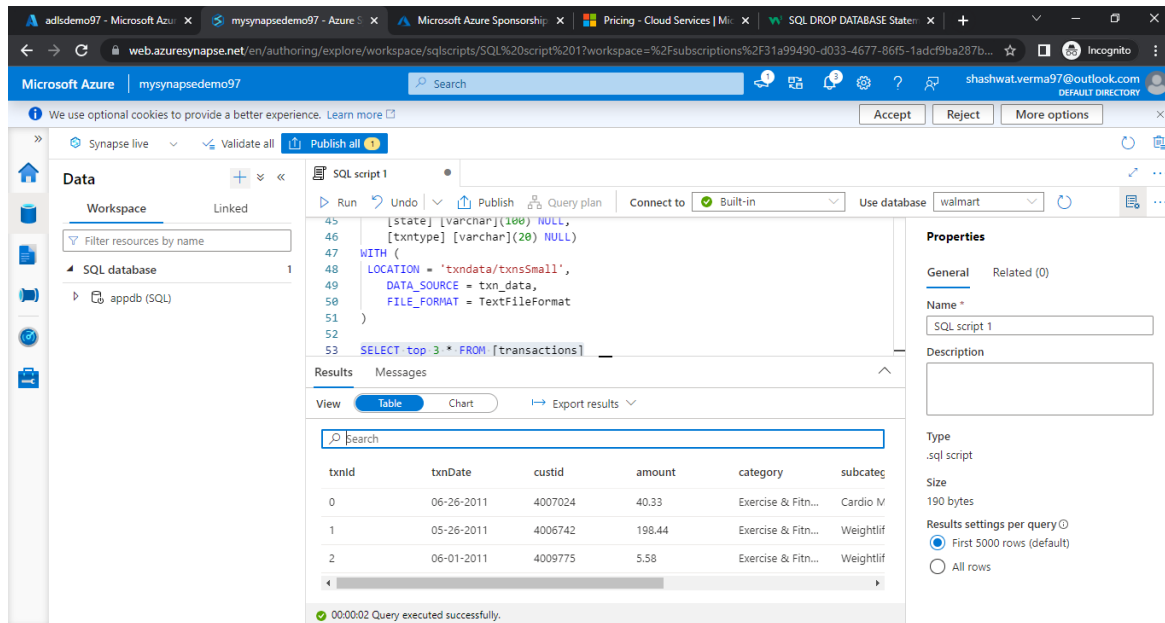
```
CREATE EXTERNAL FILE FORMAT TextFileFormat WITH (
    FORMAT_TYPE = DELIMITEDTEXT,
    FORMAT_OPTIONS (
        FIELD_TERMINATOR = ',',
        FIRST_ROW = 1))
```

-- Here we define the external table

```
CREATE EXTERNAL TABLE [transactions]
(
    [txnId] [bigint] NULL,
    [txnDate] [varchar](20) NULL,
    [custid] [bigint] NULL,
    [amount] [float] NULL,
    [category] [varchar](100) NULL,
    [subcategory] [varchar](200) NULL,
    [city] [varchar](100) NULL,
    [state] [varchar](100) NULL,
    [txntype] [varchar](20) NULL)
WITH (
    LOCATION = 'txndata/txnsSmall',
    DATA_SOURCE = txn_data,
    FILE_FORMAT = TextFileFormat
)
```

--To display top 3 rows for confirmation of successful table creation:

```
SELECT top 10 * FROM [transactions]
```



The screenshot shows the Microsoft Azure Synapse Studio interface. The SQL script editor contains the following code:

```
45 [state] [varchar](100) NULL,
46 [txntype] [varchar](20) NULL
47 WITH (
48     LOCATION = 'txndata/txnsSmall',
49     DATA_SOURCE = txn_data,
50     FILE_FORMAT = TextFileFormat
51 )
52
53 SELECT top 10 * FROM [transactions]
```

The query was executed successfully, and the results are displayed in a table view. The table has the following columns: txnId, txnDate, custid, amount, category, and subcateg. The first three rows of data are shown:

txnId	txnDate	custid	amount	category	subcateg
0	06-26-2011	4007024	40.33	Exercise & Fitn...	Cardio M
1	05-26-2011	4006742	198.44	Exercise & Fitn...	Weightlif
2	06-01-2011	4009775	5.58	Exercise & Fitn...	Weightlif

The status bar at the bottom indicates: 00:00:02 Query executed successfully.

## Problems:

--a. Find the total revenue generated based on category

```
SELECT [category], SUM([amount]) AS revenue FROM [transactions]
GROUP BY [category]
```

The screenshot shows the Microsoft Azure Synapse Studio interface. On the left, the 'Data' pane displays a list of columns for the 'dbo.transactions' table, including 'txnId', 'txnDate', 'custId', 'amount', 'category', 'subCategory', 'city', 'state', and 'txnType'. The main editor shows a SQL script with the following content:

```
52 SELECT top 10 * FROM [transactions]
53
54
55 --a. Find the total revenue generated based on category
56
57 SELECT [category], SUM([amount]) AS revenue FROM [transactions]
58 GROUP BY [category]
59
```

The 'Results' pane shows the output of the query, displaying a table with two columns: 'category' and 'revenue'. The data is as follows:

category	revenue
Jumping	2078.58
Water Sports	4379.96
Games	2899.91
Outdoor Recreation	6190.07

The status bar at the bottom indicates '00:00:07 Query executed successfully.'

--b. Find the total number of transactions done my cash

```
SELECT COUNT(*) as numberofcashtxns FROM [transactions] WHERE
[txnType] = 'cash'
```

The screenshot shows the Microsoft Azure Synapse Studio interface. On the left, the 'Data' pane displays a list of columns for the 'dbo.transactions' table, including 'txnId', 'txnDate', 'custId', 'amount', 'category', 'subCategory', 'city', 'state', and 'txnType'. The main editor shows a SQL script with the following content:

```
58 GROUP BY [category]
59
60
61 --b. Find the total number of transactions done my cash
62
63 SELECT COUNT(*) as numberofcashtxns FROM [transactions] WHERE [txnType] = 'cash'
64
```

The 'Results' pane shows the output of the query, displaying a table with one column: 'numberofcashtxns'. The data is as follows:

numberofcashtxns
64

The status bar at the bottom indicates '00:00:02 Query executed successfully.'

--c. Find the total collection of amount done by credit

```
SELECT SUM([amount]) AS credit_total_collection FROM
[transactions] WHERE [txntype] = 'credit'
```

The screenshot shows the Microsoft Azure Synapse Studio interface. On the left, the 'Data' pane displays a workspace with a linked database 'walmart' and a table 'dbo.transactions'. The 'Columns' pane lists fields: txnid, txnDate, custid, amount, category, subcategory, city, state, and txntype. The central editor shows a SQL script with the following content:

```
63
64 SELECT COUNT(*) as numerofcashtxns FROM [transactions] WHERE [txntype] = 'cash'
65
66
67 --c. Find the total collection of amount done by credit
68
69 SELECT SUM([amount]) AS credit_total_collection FROM [transactions] WHERE [txntype]
```

The 'Results' pane shows a single row of data:

credit_total_collection
38654.25

The status bar indicates '00:00:03 Query executed successfully.' The right-hand 'Properties' pane shows details for the 'SQL script 1' file.

--d. Find the highest selling category

```
SELECT TOP 1 [category], SUM([amount]) AS revenue FROM
[transactions] GROUP BY [category] ORDER BY revenue DESC
```

The screenshot shows the Microsoft Azure Synapse Studio interface. The central editor displays a SQL script with the following content:

```
69 SELECT SUM([amount]) AS credit_total_collection FROM [transactions] WHERE [txntype]
70
71
72 --d. Find the highest selling category
73
74 SELECT TOP 1 [category], SUM([amount]) AS revenue FROM [transactions]
75 GROUP BY [category] ORDER BY revenue DESC
```

The 'Results' pane shows a single row of data:

category	revenue
Outdoor Recreation	6190.07

The status bar indicates '00:00:02 Query executed successfully.' The right-hand 'Properties' pane shows details for the 'SQL script 1' file.

--e. Find the lowest selling category

```
SELECT TOP 1 [category], SUM([amount]) AS revenue FROM  
[transactions] GROUP BY [category] ORDER BY revenue
```

The screenshot displays the Microsoft Azure Synapse Studio interface. The left sidebar shows the 'Data' section with a tree view of resources, including 'External tables', 'dbo.transactions', and 'Columns'. The main editor area shows a SQL script titled 'SQL script 1' with the following code:

```
75 GROUP BY [category] ORDER BY revenue DESC  
76  
77  
78 --e. Find the lowest selling category  
79  
80 SELECT TOP 1 [category], SUM([amount]) AS revenue FROM [transactions]  
81 GROUP BY [category] ORDER BY revenue
```

The 'Results' tab is active, showing a table with two columns: 'category' and 'revenue'. The table contains one row with the value 'Puzzles' in the 'category' column and '508.86' in the 'revenue' column.

The 'Properties' panel on the right shows the 'General' tab with the following information:

- Name: SQL script 1
- Description: (empty)
- Type: .sql script
- Size: 190 bytes
- Results settings per query: ☒ First 5000 rows (default), ☐ All rows

The status bar at the bottom indicates '00:00:02 Query executed successfully.'