

Lab 8 Data Warehouse

BUAN 6390.001 – Analytics Practicum

Name: Swastik Bhatnagar
Net ID: Sxb220210

Connecting to PowerShell:

The screenshot displays the Microsoft Azure portal interface. At the top, the 'Azure services' section includes links for 'Create a resource', 'Quickstart Center', 'Azure AI services', 'Kubernetes services', 'Virtual machines', 'App Services', 'Storage accounts', 'SQL databases', 'Azure Cosmos DB', and 'More services'. Below this is the 'Resources' section, which is currently empty, showing a message 'No resources have been viewed recently' and a 'View all resources' button. The 'Navigate' section at the bottom includes links for 'Subscriptions', 'Resource groups', 'All resources', and 'Dashboard'.

The bottom of the screen shows the Azure Cloud Shell interface. The left pane displays the 'Azure services' and 'Resources' sections. The right pane shows a PowerShell session with the following commands and output:

```
rm -r dp203 -f
git clone https://github.com/Micro

cd dp203/Allfiles/Labs/08
./setup.ps1
```

The output of the PowerShell session shows the registration of various Azure services and the generation of a random suffix for Azure resources. The output includes the following text:

```
At least one lower case English letter [a-z]
At least one digit [0-9]
At least one special character (!, @, #, %, ^, &, $)
Password S@015719952261 accepted; Make sure you remember this!
Registering resource providers...
Microsoft.Synapse : Registering
Microsoft.Sql : Registering
Microsoft.Storage : Registering
Microsoft.Compute : Registering
Your randomly-generated suffix for Azure resources is mx2otfd
Finding an available region. This may take several minutes...
```

On the right side of the screen, there is a 'Data Warehouse' sidebar with a '1 Hr 57 Min Remaining' timer. It contains a 'Note' about using a Bash environment and a list of instructions for setting up the data warehouse. The instructions include:

- Note that you can resize the cloud shell by dragging the separator bar at the top of the pane, or by using the \rightarrow , \leftarrow , and \times icons at the top right of the pane to minimize, maximize, and close the pane. For more information about using the Azure Cloud Shell, see the [Azure Cloud Shell documentation](#).
- In the PowerShell pane, enter the following commands to clone this repo:

```
rm -r dp203 -f
git clone https://github.com/Micro
```

- After the repo has been cloned, enter the following commands to change to the folder for this lab and run the `setup.ps1` script it contains:

```
cd dp203/Allfiles/Labs/08
./setup.ps1
```

- If prompted, choose which subscription you want to use (this will only happen if you have access to multiple Azure subscriptions).
- When prompted, enter a suitable password to be set for your Azure Synapse SQL pool.

Note: Be sure to remember this password!

- Wait for the script to complete - this typically takes around 15 minutes, but in some cases may take longer. While you are waiting, review the [What is dedicated SQL pool in Azure Synapse Analytics?](#) article in the Azure Synapse Analytics documentation.

Explore the data warehouse schema

End >

Starting the SQL Pool:

Microsoft Azure | Synapse Analytics | synapseeu2otfd

We use optional cookies to provide a better experience. [Learn more](#)

Accept Reject More options

Synapse live Validate all Publish all

Home Data Develop Integrate Monitor Manage

Analytics pools

- SQL pools
- Apache Spark pools
- Data Explorer pools (preview)

External connections

- Linked services
- Microsoft Purview

Integration

- Triggers
- Integration runtimes

Security

- Access control
- Credentials
- Managed private endpoints

Configurations + libraries

- Workspace packages
- Data flow libraries
- Apache Spark configurations

Source control

- Git configuration

SQL pools

The serverless SQL pool, Built-in, is immediately available for your workspace. Dedicated SQL pools can be configured to adapt to team or organizational requirements and constraints. [Learn more](#)

+ New Refresh

Filter by name

Showing 1-2 of 2 items (1 Serverless, 1 Dedicated)

Name	Type	Status	Size
Built-in	Serverless	Online	Auto
sqlv2otfd	Dedicated	Online	DW100c

ENG US 12:56 PM 3/28/2025

Viewing the table in Database:

Microsoft Azure | Synapse Analytics | synapseeu2otfd

We use optional cookies to provide a better experience. [Learn more](#)

Accept Reject More options

Synapse live Validate all Publish all

Home Data Develop Integrate Monitor Manage

Data

Workspace Linked

Filter resources by name

- SQL database
- sqlv2otfd (SQL)
 - Tables
 - dbo.DimAccount
 - dbo.DimCurrency
 - dbo.DimCustomer
 - dbo.DimDate
 - Columns
 - DateKey (int, not null)
 - FullDateAlternateKey (...)
 - DayNumberOfWeek (...)
 - EnglishDayNameOfWeek (...)
 - SpanishDayNameOfWeek (...)
 - FrenchDayNameOfWeek (...)
 - DayNumberOfMonth (...)
 - DayNumberOfYear (...)
 - WeekNumberOfWeek (...)
 - EnglishMonthName (...)
 - SpanishMonthName (...)
 - FrenchMonthName (...)
 - MonthNumberOfYear (...)
 - CalendarQuarter (tinyint)
 - CalendarYear (smallint)

Select an item

Use the resource explorer to select or create a new item

View the tables in the database

1. In Synapse Studio, select the **Data** page and ensure that the **Workspace** tab is selected and contains a **SQL database** category.
2. Expand **SQL database**, the **sqlv2otfd** pool, and its **Tables** folder to see the tables in the database.

A relational data warehouse is typically based on a schema that consists of **fact** and **dimension** tables. The tables are optimized for analytical queries in which numeric metrics in the fact tables are aggregated by attributes of the entities represented by the dimension tables - for example, enabling you to aggregate Internet sales revenue by product, customer, date, and so on.
3. Expand the **dbo.FactInternetSales** table and its **Columns** folder to see the columns in this table. Note that many of the columns are keys that reference rows in the dimension tables. Others are numeric values (**measures**) for analysis.

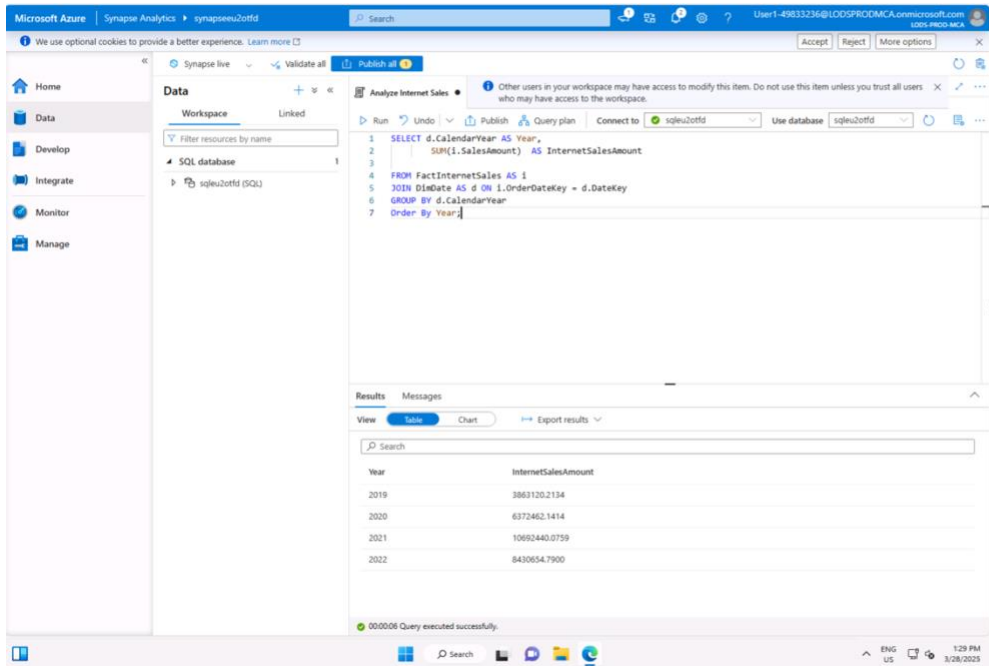
The keys are used to relate a fact table to one or more dimension tables, often in a **star** schema; in which the fact table is directly related to each dimension table (forming a multi-pointed "star" with the fact table at the center).
4. View the columns for the **dbo.DimPromotion** table, and note that it has a unique **PromotionKey** that uniquely identifies each row in the table. It also has an **AlternateKey**.

Usually, data in a data warehouse has been imported from one or more transactional sources. The alternate key reflects the business identifier for the instance of this entry in the source, but a unique numeric surrogate key is usually generated to uniquely identify each row in the data warehouse dimension table. One of the benefits of this approach is that it enables the data warehouse to contain multiple instances of the same entity at different points in time (for example, records for the same customer reflecting their address at the time an order was placed).
5. View the columns for the **dbo.DimProduct**, and note that it contains a **ProductSubcategoryKey** column, which references the **dbo.DimProductSubcategory** table, which in turn contains a **ProductCategoryKey** column that

End >

ENG US 1:22 PM 3/28/2025

Querying the Data warehouse tables:



The screenshot shows the Microsoft Azure Synapse Analytics interface. The left sidebar contains navigation options: Home, Data, Develop, Integrate, Monitor, and Manage. The main workspace is titled 'Analyze Internet Sales'. It shows a SQL query in the 'Query plan' tab, which is connected to the 'sqlue20fdd' database. The query is as follows:

```
1 SELECT d.CalendarYear AS Year,
2       SUM(i.SalesAmount) AS InternetSalesAmount
3
4 FROM FactInternetSales AS i
5 JOIN DimDate AS d ON i.OrderDateKey = d.DateKey
6 GROUP BY d.CalendarYear
7 ORDER BY Year
```

The 'Results' tab shows the output of the query, which is a table with two columns: 'Year' and 'InternetSalesAmount'. The data is as follows:

Year	InternetSalesAmount
2019	3861202.2134
2020	6372462.1414
2021	10692440.0759
2022	8430654.7900

The status bar at the bottom indicates '00:00:06 Query executed successfully.'

Query fact and dimension tables

Numeric values in a relational data warehouse are stored in fact tables with related dimension tables that you can use to aggregate the data across multiple attributes. This design means that most queries in a relational data warehouse involve aggregating and grouping data (using aggregate functions and GROUP BY clauses) across related tables (using JOIN clauses).

- On the **Data** page, select the **sql*xxxxxxx** SQL pool and in its **...** menu, select **New SQL script** > **Empty script**.
- When a new **SQL Script** tab opens, in its **Properties** pane, change the name of the script to **Analyze Internet Sales** and change the **Result settings per query** to return all rows. Then use the **Publish** button on the toolbar to save the script, and use the **Properties** button (which looks similar to **CL**) on the right end of the toolbar to close the **Properties** pane so you can see the script pane.
- In the empty script, add the following code:

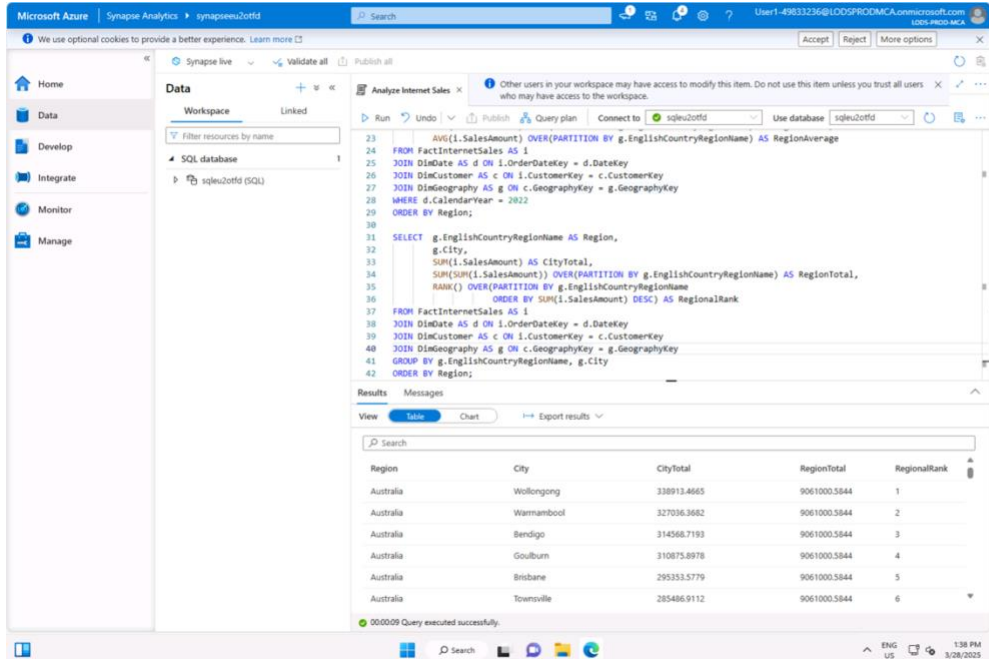
```
sql
T SELECT d.CalendarYear AS Year,
      SUM(i.SalesAmount) AS Intern
FROM FactInternetSales AS i
JOIN DimDate AS d ON i.OrderDateKey
GROUP BY d.CalendarYear
ORDER BY Year;
```

- Use the **Run** button to run the script, and review the results, which should show the Internet sales totals for each year. This query joins the fact table for Internet sales to a time dimension table based on the order date, and aggregates the sales amount measure in the fact table by the calendar month attribute of the dimension table.
- Modify the query as follows to add the month attribute from the time dimension, and then run the modified query.

```
sql
T SELECT d.CalendarYear AS Year,
      d.MonthNumberOffYear AS Month,
      SUM(i.SalesAmount) AS Intern
FROM FactInternetSales AS i
```

End >

Using Ranking Functions:



The screenshot shows the Microsoft Azure Synapse Analytics interface. The left sidebar contains navigation options: Home, Data, Develop, Integrate, Monitor, and Manage. The main workspace is titled 'Analyze Internet Sales'. It shows a SQL query in the 'Query plan' tab, which is connected to the 'sqlue20fdd' database. The query is as follows:

```
23 AVG(i.SalesAmount) OVER(PARTITION BY g.EnglishCountryRegionName) AS RegionAverage
24 FROM FactInternetSales AS i
25 JOIN DimDate AS d ON i.OrderDateKey = d.DateKey
26 JOIN DimCustomer AS c ON i.CustomerKey = c.CustomerKey
27 JOIN DimGeography AS g ON i.GeographyKey = g.GeographyKey
28 WHERE d.CalendarYear = 2022
29 ORDER BY Region;
30
31 SELECT g.EnglishCountryRegionName AS Region,
32       g.City,
33       SUM(i.SalesAmount) AS CityTotal,
34       SUM(SUM(i.SalesAmount) OVER(PARTITION BY g.EnglishCountryRegionName) AS RegionTotal,
35     RANK() OVER(PARTITION BY g.EnglishCountryRegionName
36     ORDER BY SUM(i.SalesAmount) DESC) AS RegionalRank
37 FROM FactInternetSales AS i
38 JOIN DimDate AS d ON i.OrderDateKey = d.DateKey
39 JOIN DimCustomer AS c ON i.CustomerKey = c.CustomerKey
40 JOIN DimGeography AS g ON i.GeographyKey = g.GeographyKey
41 GROUP BY g.EnglishCountryRegionName, g.City
42 ORDER BY Region;
```

The 'Results' tab shows the output of the query, which is a table with five columns: 'Region', 'City', 'CityTotal', 'RegionTotal', and 'RegionalRank'. The data is as follows:

Region	City	CityTotal	RegionTotal	RegionalRank
Australia	Wollongong	338913.4665	9061000.5844	1
Australia	Warrambool	327036.3682	9061000.5844	2
Australia	Bendigo	314568.7193	9061000.5844	3
Australia	Goulburn	310875.8978	9061000.5844	4
Australia	Brisbane	295353.5779	9061000.5844	5
Australia	Townsville	285486.9112	9061000.5844	6

The status bar at the bottom indicates '00:00:09 Query executed successfully.'

Retrieve an approximate count

When exploring very large volumes of data, queries can take significant time and resources to run. Often, data analysis doesn't require absolutely precise values - a comparison of approximate values may be sufficient.

- Under the existing queries, add the following code to retrieve the number of sales orders for each calendar year:

```
sql
T SELECT d.CalendarYear AS CalendarYe
COUNT(DISTINCT i.SalesOrderNumb
FROM FactInternetSales AS i
JOIN DimDate AS d ON i.OrderDateKey
GROUP BY d.CalendarYear
ORDER BY CalendarYear;
```

- Select only the new query code, and use the **Run** button to run it. Then review the output that is returned:

- On the **Results** tab under the query, view the order counts for each year.
- On the **Messages** tab, view the total execution time for the query.

- Modify the query as follows, to return an approximate count for each year. Then re-run the query.

```
sql
T SELECT d.CalendarYear AS CalendarYe
APPROX_COUNT(DISTINCT i.SalesOrder
FROM FactInternetSales AS i
```

End >

Retrieving approximate count:

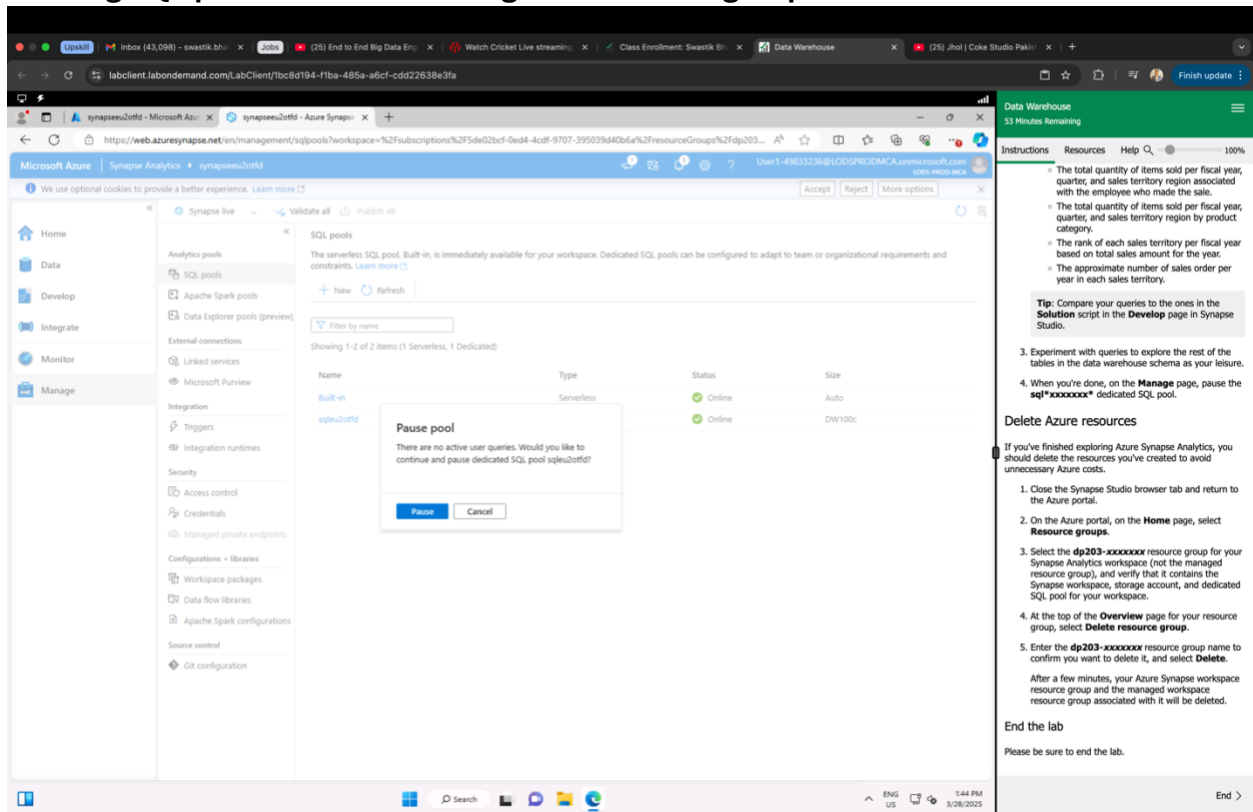
The screenshot displays the Microsoft Azure Synapse Analytics interface. The left sidebar contains navigation options: Home, Data, Develop, Integrate, Monitor, and Manage. The 'Data' section is active, showing a workspace with a filter for 'SQL database' and a list of resources including 'sqlu2otfd (SQL)'. The main pane shows a SQL query titled 'Analyze Internet Sales' with a 'Run' button. The query is a complex T-SQL statement involving multiple joins and aggregate functions. Below the query editor, the 'Results' tab is selected, showing a table with two columns: 'CalendarYear' and 'Orders'. The table contains data for the years 2020, 2021, and 2022. A status bar at the bottom indicates '00:00:05 Query executed successfully.'.

```
27 JOIN DimGeography AS g ON c.GeographyKey = g.GeographyKey
28 WHERE d.CalendarYear = 2022
29 ORDER BY Region;
30
31 SELECT g.EnglishCountryRegionName AS Region,
32        g.City,
33        SUM(i.SalesAmount) AS CityTotal,
34        SUM(SUM(i.SalesAmount)) OVER(PARTITION BY g.EnglishCountryRegionName) AS RegionTotal,
35        RANK() OVER(PARTITION BY g.EnglishCountryRegionName
36                   ORDER BY SUM(i.SalesAmount) DESC) AS RegionalRank
37 FROM FactInternetSales AS i
38 JOIN DimDate AS d ON i.OrderDateKey = d.DateKey
39 JOIN DimCustomer AS c ON i.CustomerKey = c.CustomerKey
40 JOIN DimGeography AS g ON c.GeographyKey = g.GeographyKey
41 GROUP BY g.EnglishCountryRegionName, g.City
42 ORDER BY Region;
43
44
45 SELECT d.CalendarYear AS CalendarYear,
46        APPROX_COUNT_DISTINCT(i.SalesOrderNumber) AS Orders
47 FROM FactInternetSales AS i
48 JOIN DimDate AS d ON i.OrderDateKey = d.DateKey
49 GROUP BY d.CalendarYear
50 ORDER BY CalendarYear;
```

CalendarYear	Orders
2020	2721
2021	12533
2022	11109

00:00:05 Query executed successfully.

Pausing SQL pool and then deleting the Resource group:



The screenshot shows the Azure Synapse Studio interface. On the left, the 'Manage' tab is selected. The main area displays the 'SQL pools' section. A 'Pause pool' dialog box is open, asking for confirmation to pause the 'sqlueu2otfd' pool. The background shows a table of SQL pools with columns for Name, Type, Status, and Size.

Name	Type	Status	Size
Built-in	Serverless	Online	Auto
sqlueu2otfd	Serverless	Online	DW100c

Pause pool
There are no active user queries. Would you like to continue and pause dedicated SQL pool sqlueu2otfd?

Buttons: Pause, Cancel

Data Warehouse
53 Minutes Remaining

Instructions Resources Help

- The total quantity of items sold per fiscal year, quarter, and sales territory region associated with the employee who made the sale.
- The total quantity of items sold per fiscal year, quarter, and sales territory region by product category.
- The rank of each sales territory per fiscal year based on total sales amount for the year.
- The approximate number of sales order per year in each sales territory.

Tip: Compare your queries to the ones in the **Solution** script in the **Develop** page in Synapse Studio.

3. Experiment with queries to explore the rest of the tables in the data warehouse schema as your leisure.

4. When you're done, on the **Manage** page, pause the **sql*xxxxxxx*** dedicated SQL pool.

Delete Azure resources

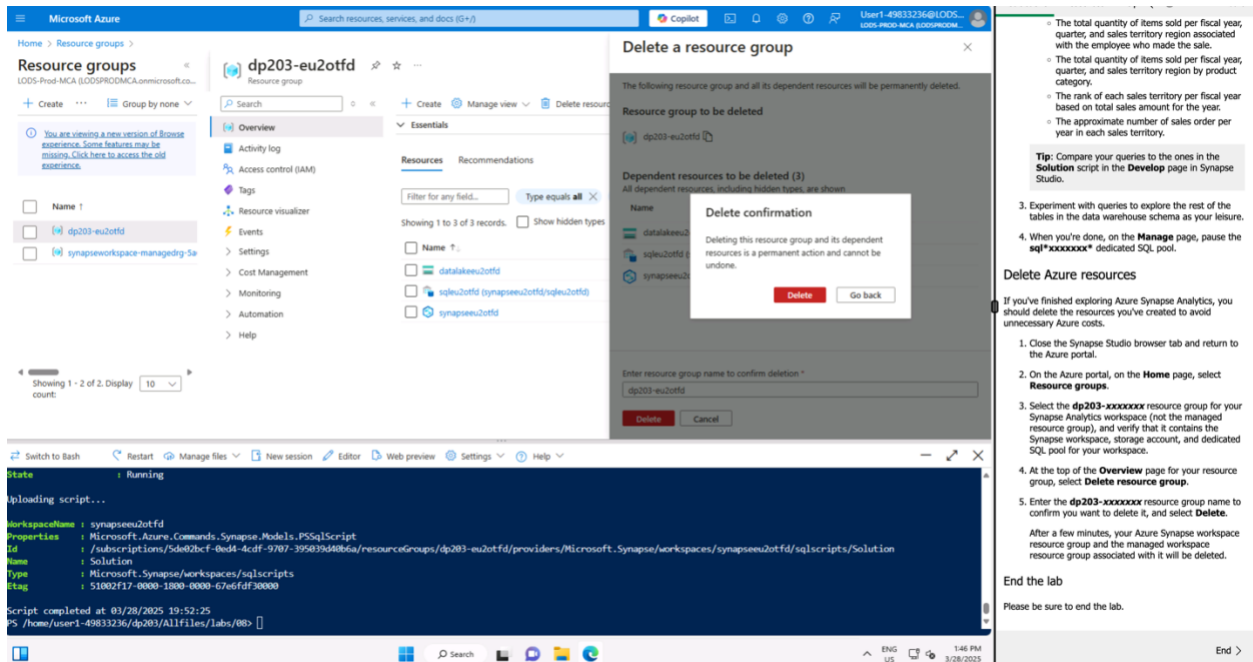
If you've finished exploring Azure Synapse Analytics, you should delete the resources you've created to avoid unnecessary Azure costs.

- Close the Synapse Studio browser tab and return to the Azure portal.
 - On the Azure portal, on the **Home** page, select **Resource groups**.
 - Select the **dp203-xxxxxxx** resource group for your Synapse Analytics workspace (not the managed resource group), and verify that it contains the Synapse workspace, storage account, and dedicated SQL pool for your workspace.
 - At the top of the **Overview** page for your resource group, select **Delete resource group**.
 - Enter the **dp203-xxxxxxx** resource group name to confirm you want to delete it, and select **Delete**.
- After a few minutes, your Azure Synapse workspace resource group and the managed workspace resource group associated with it will be deleted.

End the lab

Please be sure to end the lab.

End >



The screenshot shows the Azure portal interface. The 'Resource groups' section is selected, and the 'dp203-eu2otfd' resource group is highlighted. A 'Delete a resource group' dialog box is open, showing the resource group name and its dependent resources. A 'Delete confirmation' dialog box is also open, asking for confirmation to delete the resource group.

Delete a resource group

The following resource group and all its dependent resources will be permanently deleted.

Resource group to be deleted

- dp203-eu2otfd

Dependent resources to be deleted (3)

- datalakeueu2otfd
- sqlueu2otfd
- synapseueu2otfd

Delete confirmation

Deleting this resource group and its dependent resources is a permanent action and cannot be undone.

Buttons: Delete, Go back

The total quantity of items sold per fiscal year, quarter, and sales territory region associated with the employee who made the sale.

The total quantity of items sold per fiscal year, quarter, and sales territory region by product category.

The rank of each sales territory per fiscal year based on total sales amount for the year.

The approximate number of sales order per year in each sales territory.

Tip: Compare your queries to the ones in the **Solution** script in the **Develop** page in Synapse Studio.

3. Experiment with queries to explore the rest of the tables in the data warehouse schema as your leisure.

4. When you're done, on the **Manage** page, pause the **sql*xxxxxxx*** dedicated SQL pool.

Delete Azure resources

If you've finished exploring Azure Synapse Analytics, you should delete the resources you've created to avoid unnecessary Azure costs.

- Close the Synapse Studio browser tab and return to the Azure portal.
 - On the Azure portal, on the **Home** page, select **Resource groups**.
 - Select the **dp203-xxxxxxx** resource group for your Synapse Analytics workspace (not the managed resource group), and verify that it contains the Synapse workspace, storage account, and dedicated SQL pool for your workspace.
 - At the top of the **Overview** page for your resource group, select **Delete resource group**.
 - Enter the **dp203-xxxxxxx** resource group name to confirm you want to delete it, and select **Delete**.
- After a few minutes, your Azure Synapse workspace resource group and the managed workspace resource group associated with it will be deleted.

End the lab

Please be sure to end the lab.

End >

Conclusion:

In this lab I gained hands-on experience in managing and querying a data warehouse using SQL Pool. I learned how to connect to PowerShell, initiate and pause SQL pools, retrieve data using ranking functions, and efficiently manage resources within the database environment. This hands-on experience enhanced my understanding of how data warehouses operate and how to apply SQL queries for data retrieval and analysis.