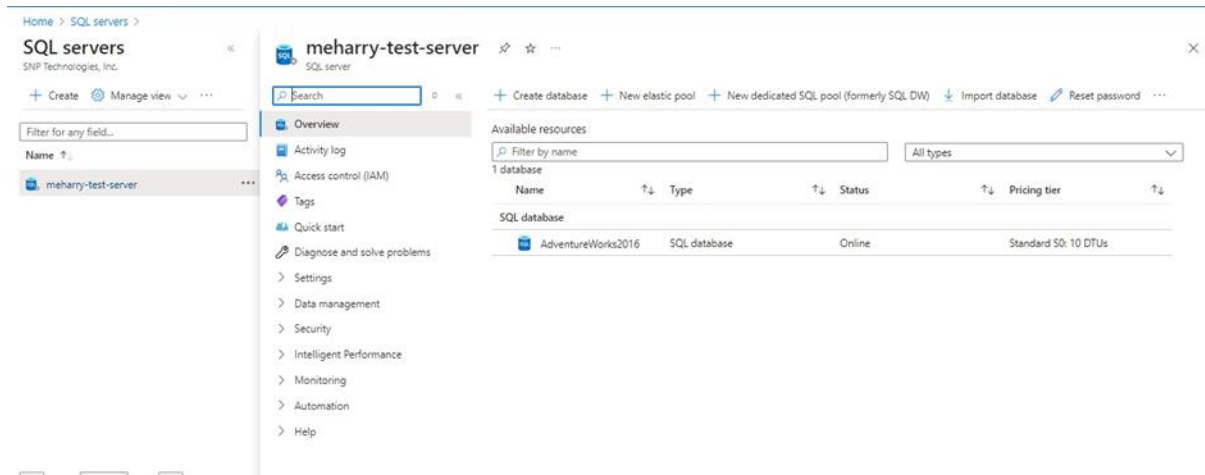


PRCL – 01 Data Warehousing Project Report

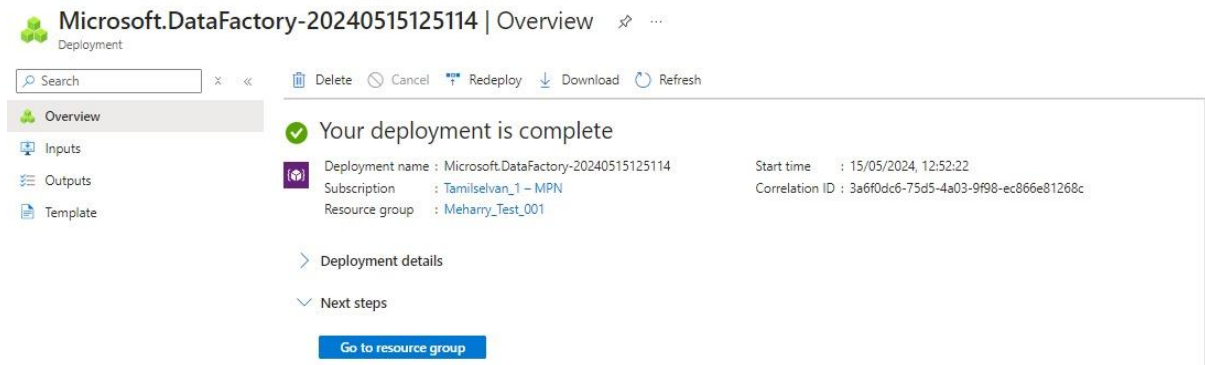
Step1:

In the Azure SQL server, first I created a new SQL database named AdventureWorks2016.



Step 2:

Here I created an Azure data factory named Microsoft_DataFactory-20240515125114.



Step 3:

After creating an Azure data factory, I created two linked services of LS_onpremysql for source and AZ_sql_server for the target.

The screenshot shows the 'Edit linked service' page for a linked service named 'LS_onpremysql'. The left sidebar shows the 'Data Factory' navigation pane with 'Linked services' selected. The main pane displays the 'Linked services' table with two items: 'AZ_sql_server' (Azure SQL Database) and 'LS_onpremysql' (MySQL). The right pane shows the configuration for 'LS_onpremysql'.

Name	Type	Related
AZ_sql_server	Azure SQL Database	2
LS_onpremysql	MySQL	1

Edit linked service
MySQL Learn more

Name *
LS_onpremysql

Description

Connect via integration runtime *
☒ AutoResolveIntegrationRuntime (Managed Virtual Network)
☐ Interactive authoring disabled

Driver version
☒ Recommended ☐ Legacy

Server name *
18.136.157.135

Port
3306

Database name *
datamites_sql

Save Cancel Test connection

The screenshot shows the 'Edit linked service' page for a linked service named 'AZ_sql_server'. The left sidebar shows the 'Data Factory' navigation pane with 'Linked services' selected. The main pane displays the 'Linked services' table with two items: 'AZ_sql_server' (Azure SQL Database) and 'LS_onpremysql' (MySQL). The right pane shows the configuration for 'AZ_sql_server'.

Name	Type	Related
AZ_sql_server	Azure SQL Database	2
LS_onpremysql	MySQL	1

Edit linked service
Azure SQL Database Learn more

AZ_sql_server

Description

Connect via integration runtime *
☒ AutoResolveIntegrationRuntime (Managed Virtual Network)
☐ Interactive authoring disabled

Connection string Azure Key Vault

Account selection method
☐ From Azure subscription ☒ Enter manually

Fully qualified domain name *
meharry-test-server.database.windows.net

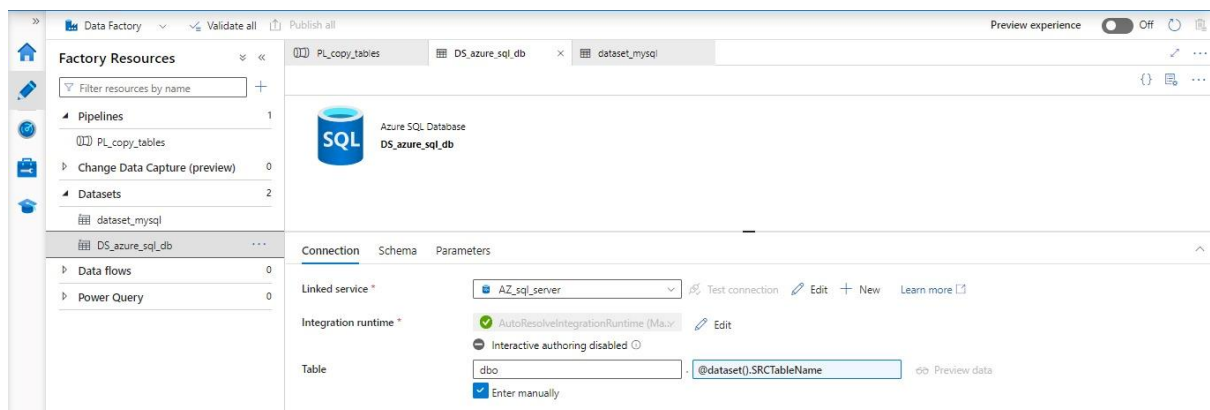
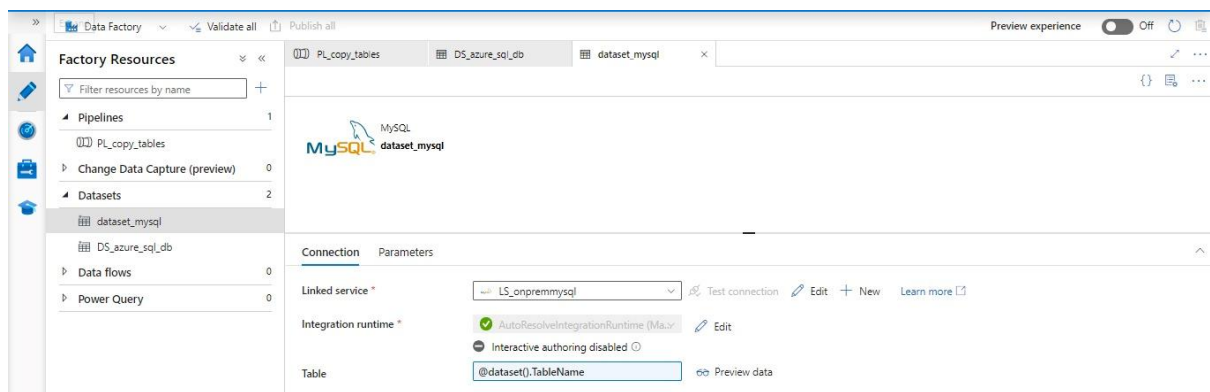
Database name *
AdventureWorks2016

Managed private endpoint

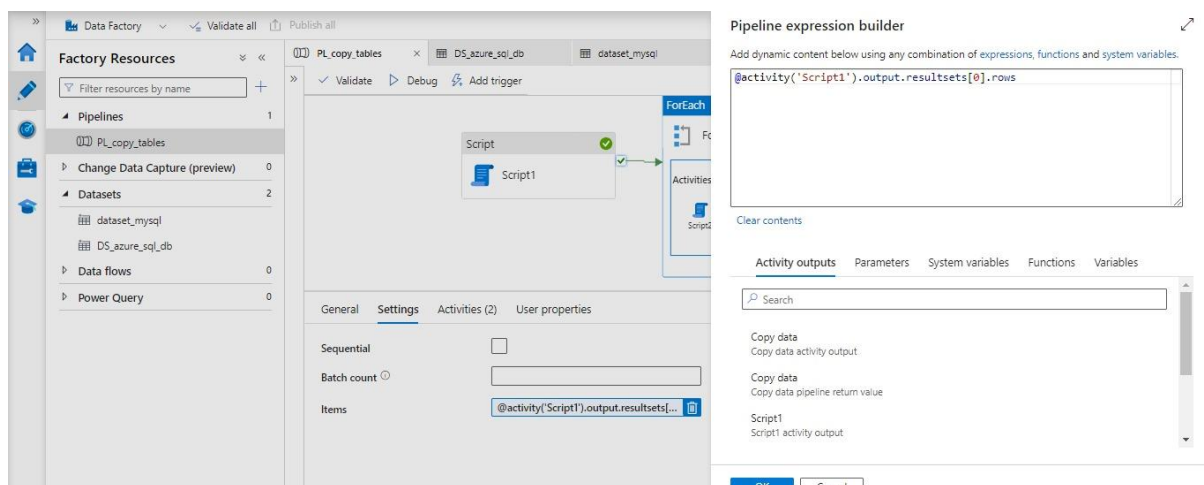
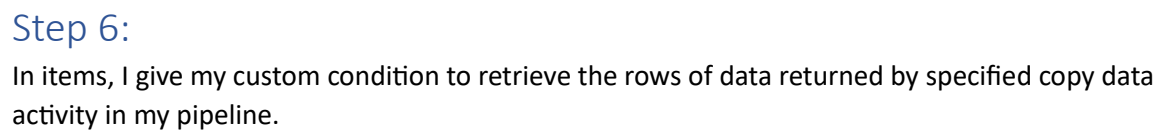
Apply Cancel Test connection

Step 4:

In the data factory, I created a dataset to connect between linked services of the source and target.

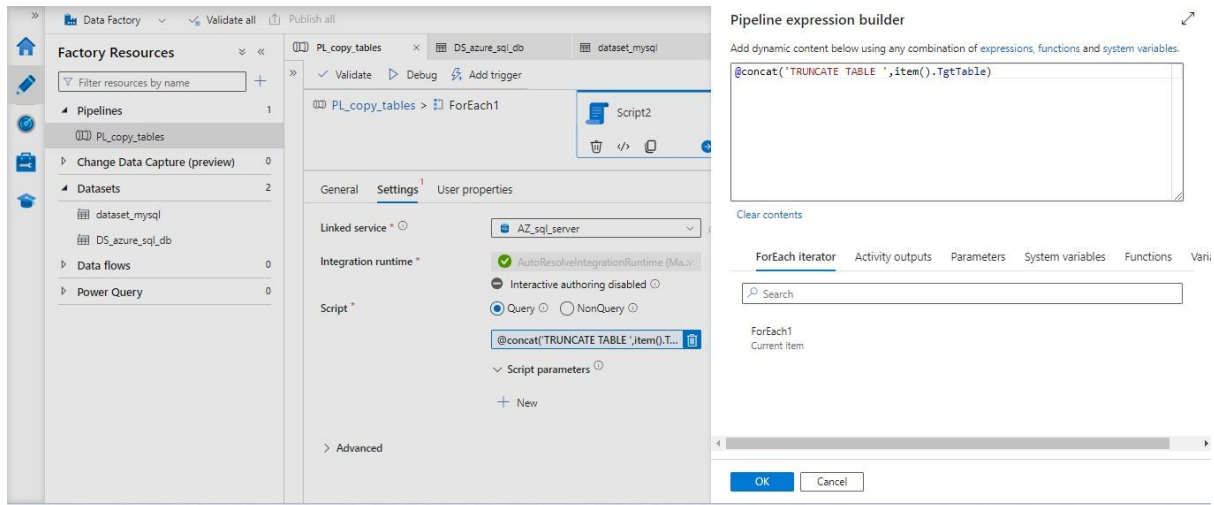


By using the script I created a custom pipeline and used it for loop to logical conditions my source and target tables .



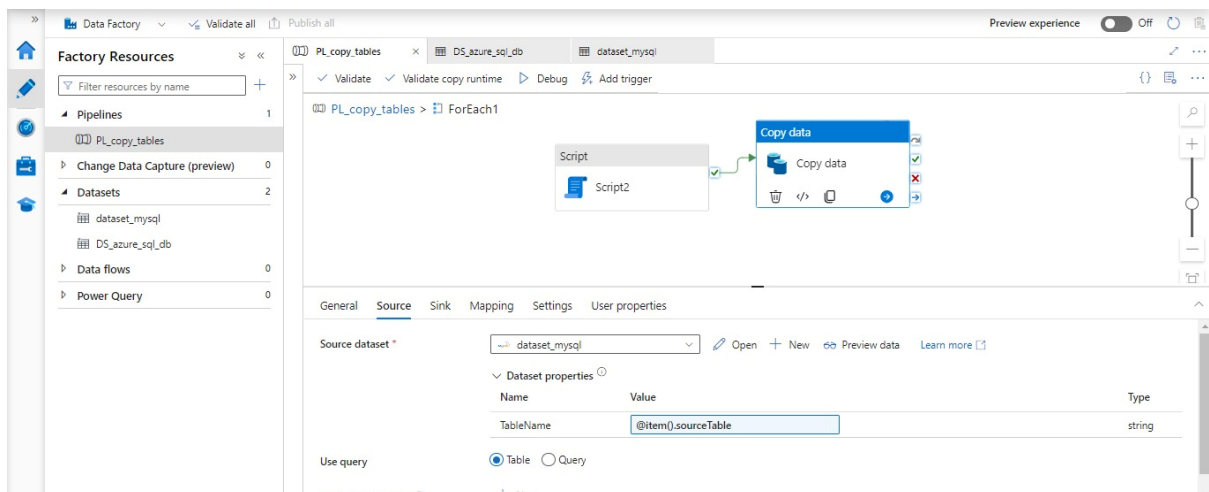
Step 7:

I used full truncate load, to remove the rows from the table to avoid the duplication in target .



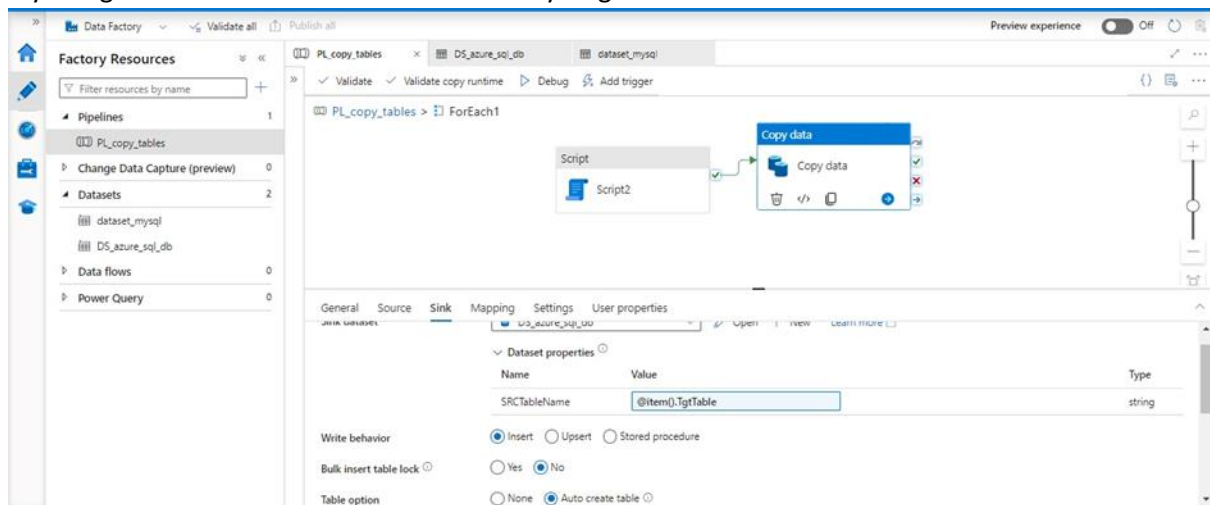
Step 8:

From the data factory, I connect source dataset of dataset_mysql.



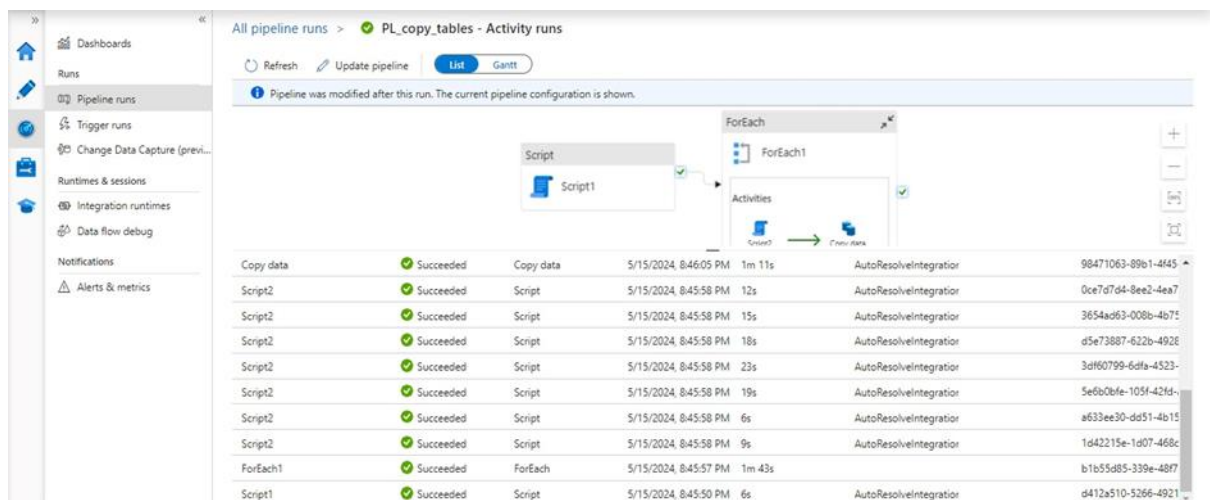
Step 9:

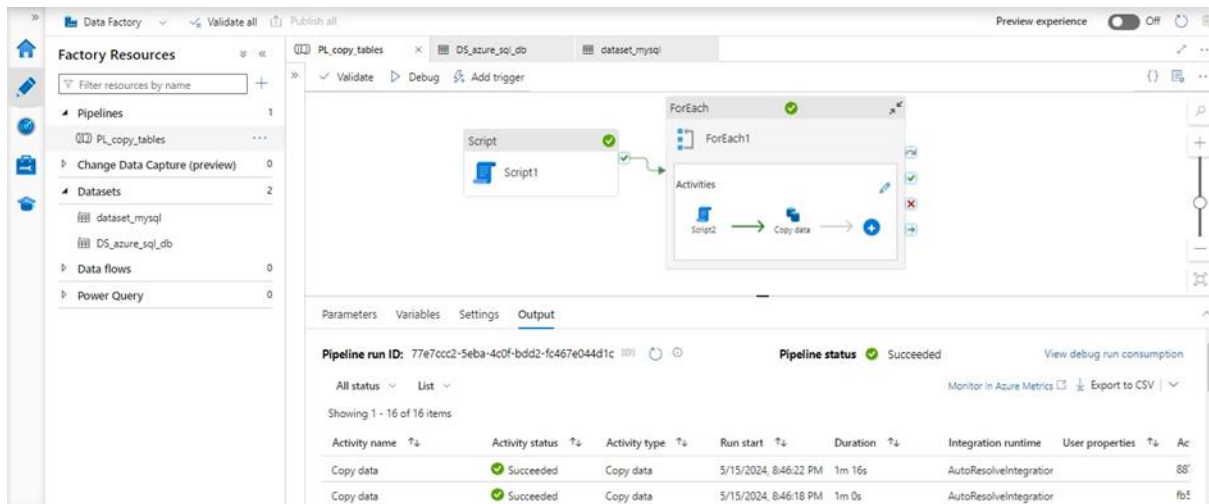
By using a sink I transformed the data into my target.



Step 10:

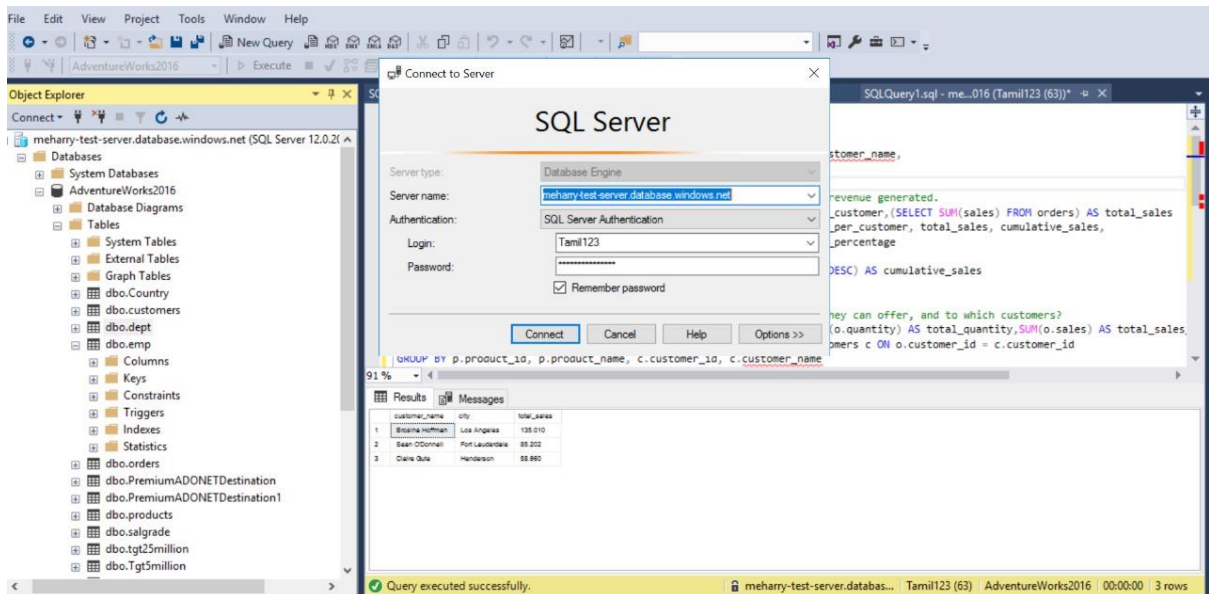
At last, I debug the pipeline it runs.





Step 12:

I connected my SQL server and retrieve the queries .



Queries:

1. Top 3 customers belong to which city.

Select top 3 c.customer_name ,c.city,sum(o.sales) as total_sales from customers c join orders o on c.customer_id =o.customer_id group by c.customer_name,c.city order by total_sales desc;

2. What percentage of top customers are responsible for 50% of the revenue generated.
 WITH SalesSummary AS (SELECT customer_id,SUM(sales) AS total_sales_per_customer,(SELECT SUM(sales) FROM orders) AS total_sales FROM orders GROUP BY customer_id)SELECT s.customer_id, s.total_sales_per_customer, total_sales, cumulative_sales, CAST(cumulative_sales * 1.0 / total_sales AS decimal(18, 4)) AS sales_percentage FROM (SELECT customer_id,total_sales_per_customer,total_sales, SUM(total_sales_per_customer) OVER (ORDER BY total_sales_per_customer DESC) AS cumulative_sales FROM SalesSummary) AS s WHERE cumulative_sales <= 0.5 * total_sales;

3. If the organization wants to offer discounts on which products they can offer, and to which customers?

```
SELECT p.product_id,p.product_name,c.customer_id,c.customer_name,COUNT(o.quantity) AS
total_quantity,SUM(o.sales) AS total_sales_amount
FROM orders o JOIN products p ON o.product_id = p.product_id JOIN customers c ON
o.customer_id = c.customer_id
GROUP BY p.product_id, p.product_name, c.customer_id, c.customer_name
HAVING COUNT(o.quantity) >= 1 AND SUM(o.sales) >= 50
ORDER BY total_quantity DESC, total_sales_amount DESC;
```

4. Explain the hierarchy of the employee in the emp table using SQL query.

```
SELECT e1.empno AS employee_id, e1.ename AS employee_name, e1.job AS employee_job,
e1.mgr AS manager_id,
CONCAT(e2.empno, ' - ', e2.ename) AS manager_name,0 AS [level]
FROM emp e1 LEFT JOIN emp e2 ON e1.mgr = e2.empno UNION ALL
SELECT e1.empno AS employee_id,e1.ename AS employee_name,e1.job AS employee_job,e1.mgr
AS manager_id,
CONCAT(e2.empno, ' - ', e2.ename) AS manager_name,1 AS [level]
FROM emp e1 LEFT JOIN emp e2 ON e1.mgr = e2.empno
LEFT JOIN emp e3 ON e2.mgr = e3.empno WHERE e3.empno IS NULL
ORDER BY [level], employee_id;
```

5. Who was the first employee to get hired in the organization?

```
select empno,ename,hiredate from emp where hiredate=(select min(hiredate) from emp);
```

The screenshot shows the Microsoft SQL Server Enterprise Manager interface. The left pane displays the 'Object Explorer' with the 'AdventureWorks2016' database selected. The right pane shows a query window with three SQL queries and their results.

Query 1: Top 3 customers by total sales

```
-- 1 Top 3 customers belong to which city.
SELECT top 3 c.customer_name,c.city,SUM(o.sales) as total_sales from
customers c join orders o on c.customer_id=o.customer_id group by c.customer_name,
c.city order by total_sales desc;
```

Results:

Customer Name	City	Total Sales
Boston (Hoffman)	Los Angeles	138,010
Sean O'Donner	Portland, Ore.	\$5,202
Chae Guk	Memphis	\$5,860

Query 2: Sales Summary

```
-- 2. What percentage of top customers are responsible for 50% of the revenue generated.
WITH SalesSummary AS (SELECT customer_id,SUM(sales) AS total_sales_per_customer,(SELECT SUM(sales) FROM orders) AS total_sales
FROM orders GROUP BY customer_id)SELECT s.customer_id, s.total_sales_per_customer, total_sales, cumulative_sales,
CAST((cumulative_sales * 1.0 / total_sales AS decimal(18, 4)) AS sales_percentage
FROM ( SELECT customer_id,total_sales_per_customer,total_sales,
SUM(total_sales_per_customer) OVER (ORDER BY total_sales_per_customer DESC) AS cumulative_sales
FROM SalesSummary ) AS s
WHERE cumulative_sales <= 0.5 * total_sales;
```

Query 3: Products and customers for discounts

```
-- 3. If the organization wants to offer discounts on which products they can offer, and to which customers?
SELECT p.product_id,p.product_name,c.customer_id,c.customer_name,COUNT(o.quantity) AS total_quantity,SUM(o.sales) AS total_sales
FROM orders o JOIN products p ON o.product_id = p.product_id JOIN customers c ON o.customer_id = c.customer_id
GROUP BY p.product_id, p.product_name, c.customer_id, c.customer_name
```

The status bar at the bottom indicates: "Query executed successfully. AdventureWorks2016 00:00:00 3 rows".

File Edit View Query Project Tools Window Help

AdventureWorks2016 Execute

Object Explorer

Connect - meaharry-test-server.database.windows.net (SQL Server 12.0.2000.8)

Databases

- System Databases
- AdventureWorks2016
 - Database Diagrams
 - Tables
 - Views
 - External Resources
 - Synonyms
 - Programmability
 - Query Store
 - Extended Events
 - Storage
 - Security
 - Integration Services Catalogs

SQLQuery1.sql - me_016 (Tamil123 (53))

```
-- 2. What percentage of top customers are responsible for 50% of the revenue generated.
WITH SalesSummary AS (SELECT customer_id,SUM(sales) AS total_sales_per_customer,(SELECT SUM(sales) FROM orders) AS total_sales
FROM orders GROUP BY customer_id)SELECT s.customer_id, s.total_sales_per_customer, total_sales, cumulative_sales,
CAST(cumulative_sales * 1.0 / total_sales AS decimal(18, 4)) AS sales_percentage
FROM ( SELECT customer_id,total_sales_per_customer,total_sales,
SUM(total_sales_per_customer) OVER (ORDER BY total_sales_per_customer DESC) AS cumulative_sales
FROM SalesSummary ) AS s
WHERE cumulative_sales <= 0.5 * total_sales;

-- 3. If the organization wants to offer discounts on which products they can offer, and to which customers?
SELECT p.product_id,p.product_name,c.customer_id,c.customer_name,COUNT(o.quantity) AS total_quantity,SUM(o.sales) AS total_sale
FROM orders o JOIN products p ON o.product_id = p.product_id JOIN customers c ON o.customer_id = c.customer_id
GROUP BY p.product_id, p.product_name, c.customer_id, c.customer_name
HAVING COUNT(o.quantity) >= 1 AND SUM(o.sales) >= 50
ORDER BY total_quantity DESC, total_sales_amount DESC;
```

Results Messages

customer_id	total_sales_per_customer	total_sales	cumulative_sales	sales_percentage
1	324.896	135.010	135.010	0.4155

File Edit View Query Project Tools Window Help

AdventureWorks2016 Execute

Object Explorer

Connect - meaharry-test-server.database.windows.net (SQL Server 12.0.2000.8)

Databases

- System Databases
- AdventureWorks2016
 - Database Diagrams
 - Tables
 - Views
 - External Resources
 - Synonyms
 - Programmability
 - Query Store
 - Extended Events
 - Storage
 - Security
 - Integration Services Catalogs

SQLQuery1.sql - me_016 (Tamil123 (53))

```
-- 2. What percentage of top customers are responsible for 50% of the revenue generated.
WITH SalesSummary AS (SELECT customer_id,SUM(sales) AS total_sales_per_customer,(SELECT SUM(sales) FROM orders) AS total_sales
FROM orders GROUP BY customer_id)SELECT s.customer_id, s.total_sales_per_customer, total_sales, cumulative_sales,
CAST(cumulative_sales * 1.0 / total_sales AS decimal(18, 4)) AS sales_percentage
FROM ( SELECT customer_id,total_sales_per_customer,total_sales,
SUM(total_sales_per_customer) OVER (ORDER BY total_sales_per_customer DESC) AS cumulative_sales
FROM SalesSummary ) AS s
WHERE cumulative_sales <= 0.5 * total_sales;

-- 3. If the organization wants to offer discounts on which products they can offer, and to which customers?
SELECT p.product_id,p.product_name,c.customer_id,c.customer_name,COUNT(o.quantity) AS total_quantity,SUM(o.sales) AS total_sale
FROM orders o JOIN products p ON o.product_id = p.product_id JOIN customers c ON o.customer_id = c.customer_id
GROUP BY p.product_id, p.product_name, c.customer_id, c.customer_name
HAVING COUNT(o.quantity) >= 1 AND SUM(o.sales) >= 50
ORDER BY total_quantity DESC, total_sales_amount DESC;
```

Results Messages

product_id	product_name	customer_id	customer_name	total_quantity	total_sales_amount
1	FLUR-TA-100000772	BH-11710	Brianne Poffman	1	71.530
2	FLUR-TA-100000772	SO-20335	Steven O'Donnell	1	69.650
3	FLUR-BD-10001790	CO-13520	Chloe Gula	1	55.980

File Edit View Query Project Tools Window Help

AdventureWorks2016 Execute

Object Explorer

Connect - meharry-test-server.database.windows.net (SQL Server 12.0.2000)

Databases

- System Databases
- AdventureWorks2016
 - Database Diagrams
 - Tables
 - Views
 - External Resources
 - Synonyms
 - Programmability
 - Query Store
 - Extended Events
 - Storage
 - Security
- Integration Services Catalogs

SQLQuery1.sql - me...016 (Tamil123 (53))

```
-- 4. Explain the hierarchy of the employee in the emp table using SQL query.
SELECT e1.empno AS employee_id, e1.ename AS employee_name, e1.job AS employee_job, e1.mgr AS manager_id,
CONCAT(e2.empno, ' - ', e2.ename) AS manager_name, 0 AS [level]
FROM emp e1 LEFT JOIN emp e2 ON e1.mgr = e2.empno UNION ALL
SELECT e1.empno AS employee_id, e1.ename AS employee_name, e1.job AS employee_job, e1.mgr AS manager_id,
CONCAT(e2.empno, ' - ', e2.ename) AS manager_name, 1 AS [level]
FROM emp e1 LEFT JOIN emp e2 ON e1.mgr = e2.empno
LEFT JOIN emp e3 ON e2.mgr = e3.empno WHERE e3.empno IS NULL
ORDER BY [level], employee_id;
```

-- 5. Who was the first employee to get hired in the organization?

90 %

Results Messages

employee_id	employee_name	employee_job	manager_id	manager_name	level
7789	SMITH	CLERK	7802	7802-FORD	0
7499	ALLEN	SALESMAN	7668	7668-BLAKE	0
7521	WARD	SALESMAN	7668	7668-BLAKE	0
7566	JONES	MANAGER	7839	7839-KING	0
7554	MARTIN	SALESMAN	7668	7668-BLAKE	0
7698	BLAKE	MANAGER	7839	7839-KING	0
7782	CLARK	MANAGER	7839	7839-KING	0
7788	SCOTT	ANALYST	7566	7566-JONES	0
7839	KING	PRESIDENT	NULL		0
7844	TURNER	SALESMAN	7668	7668-BLAKE	0
7876	ADAMS	CLERK	7788	7788-SCOTT	0
7900	JONES	CLERK	7668	7668-BLAKE	0
7902	FORD	ANALYST	7566	7566-JONES	0
7934	MILLER	CLERK	7782	7782-CLARK	0
7968	JONES	MANAGER	7839	7839-KING	1
7698	BLAKE	MANAGER	7839	7839-KING	1
7782	CLARK	MANAGER	7839	7839-KING	1
7839	KING	PRESIDENT	NULL		1

File Edit View Query Project Tools Window Help

AdventureWorks2016 Execute

Object Explorer

Connect - meharry-test-server.database.windows.net (SQL Server 12.0.2000)

Databases

- System Databases
- AdventureWorks2016
 - Database Diagrams
 - Tables
 - Views
 - External Resources
 - Synonyms
 - Programmability
 - Query Store
 - Extended Events
 - Storage
 - Security
- Integration Services Catalogs

SQLQuery1.sql - me...016 (Tamil123 (53))

```
CONCAT(e2.empno, ' - ', e2.ename) AS manager_name, 1 AS [level]
FROM emp e1 LEFT JOIN emp e2 ON e1.mgr = e2.empno
LEFT JOIN emp e3 ON e2.mgr = e3.empno WHERE e3.empno IS NULL
ORDER BY [level], employee_id;
```

-- 5. Who was the first employee to get hired in the organization?

```
select empno,ename,hiredate from emp where hiredate=(select min(hiredate) from emp);
```

90 %

Results Messages

empno	ename	hiredate
7839	KING	1980-06-09 00:00:00

Query executed successfully. meharry-test-server.databases... Tamil123 (53) AdventureWorks2016 00:00:00 1 rows