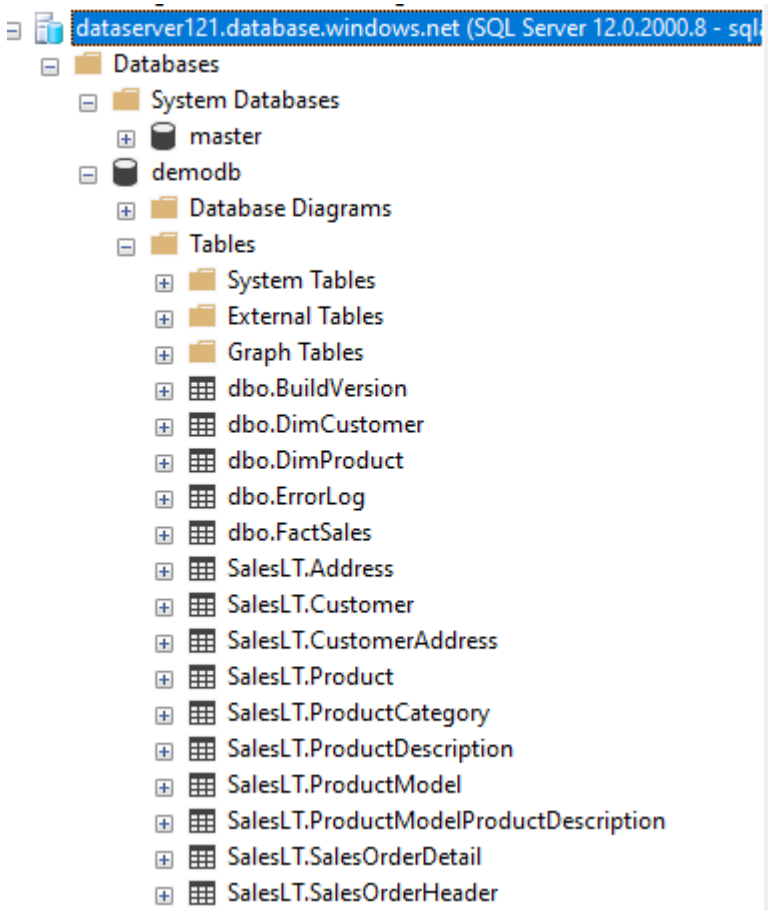
# 😊 Using query for Data Transfer

- In this lab, we're focusing on utilizing Azure Data Factory to facilitate data transfer between Azure SQL Database and Azure Synapse Analytics. The primary objective is to create a fact table within Synapse Analytics by extracting data from SQL Database tables using custom SQL queries. Here's a detailed summary of the process and its end goal:
- Firstly, we start by connecting to both Azure SQL Database and Azure Synapse Analytics using SQL Server Management Studio (SSMS). Within SQL Database, we've previously loaded sample data into various tables, providing the foundation for our data transfer operations.
- Within SSMS, we prepare for the creation of a fact table within the dedicated SQL pool of Azure Synapse Analytics. This involves writing a custom SQL query that defines the logic for aggregating and transforming the data from multiple SQL Database tables into the desired structure for the fact table.
- Moving to Azure Data Factory, we initiate the creation of a new pipeline specifically designed for the data transfer task. Within the pipeline creation wizard, we opt for a built-in copy task and configure the source connection to our Azure SQL Database.
- After specifying necessary details such as subscription, server name, database name, username, and password for authentication, we test the connection to ensure its success. Unlike previous data transfer tasks where we selected tables directly, this time we utilize the "Query" option to specify our custom SQL query, which defines the logic for building our fact table.
- Once the SQL query is pasted into the designated area, we proceed to configure the destination connection, selecting our Synapse Analytics dedicated SQL pool and choosing the existing fact table as the destination. The pipeline wizard automatically generates a column mapping based on the query and the destination table schema, simplifying the setup process.
- Before deploying the pipeline, we give the pipeline activity a name, disable staging, and opt for bulk insert as the transfer method. After reviewing the configurations, we deploy the pipeline and monitor its deployment progress in the monitor section of Azure Data Factory.
- Upon successful deployment, we verify the execution status, ensuring that the data transfer operation completes without errors. Returning to SSMS, we execute a select statement on the fact table to confirm that the data has been successfully transferred and reflects the expected results.

**In summary, the end goal of this lab is to automate the process of aggregating and transforming data from Azure SQL Database tables into a structured fact table within Azure Synapse Analytics, enabling efficient data integration and analytics within the Azure ecosystem. By leveraging Azure Data Factory and custom SQL queries, we streamline the data transfer process, ultimately empowering organizations to derive valuable insights from their data assets.**

## 😄 To begin with the Lab:

1. Now go to SSMS and connect your SQL Database and Synapse Workspace.
2. So, while creating our SQL Database we had loaded sample data into it.
3. Below you can see those sample tables.




4. Now with reference to these tables we are going to create a fact that we had done while working on Synapse.
5. Open a new query in your Pool db. Then create your fact sales table.

```
CREATE TABLE [dbo].[FactSales](
    [ProductID] [int] NOT NULL,
    [SalesOrderID] [int] NOT NULL,
    [CustomerID] [int] NOT NULL,
    [OrderQty] [smallint] NOT NULL,
    [UnitPrice] [money] NOT NULL,
    [OrderDate] [datetime] NULL,
    [TaxAmt] [money] NULL
)
WITH
(
    DISTRIBUTION = HASH (CustomerID)
)
```

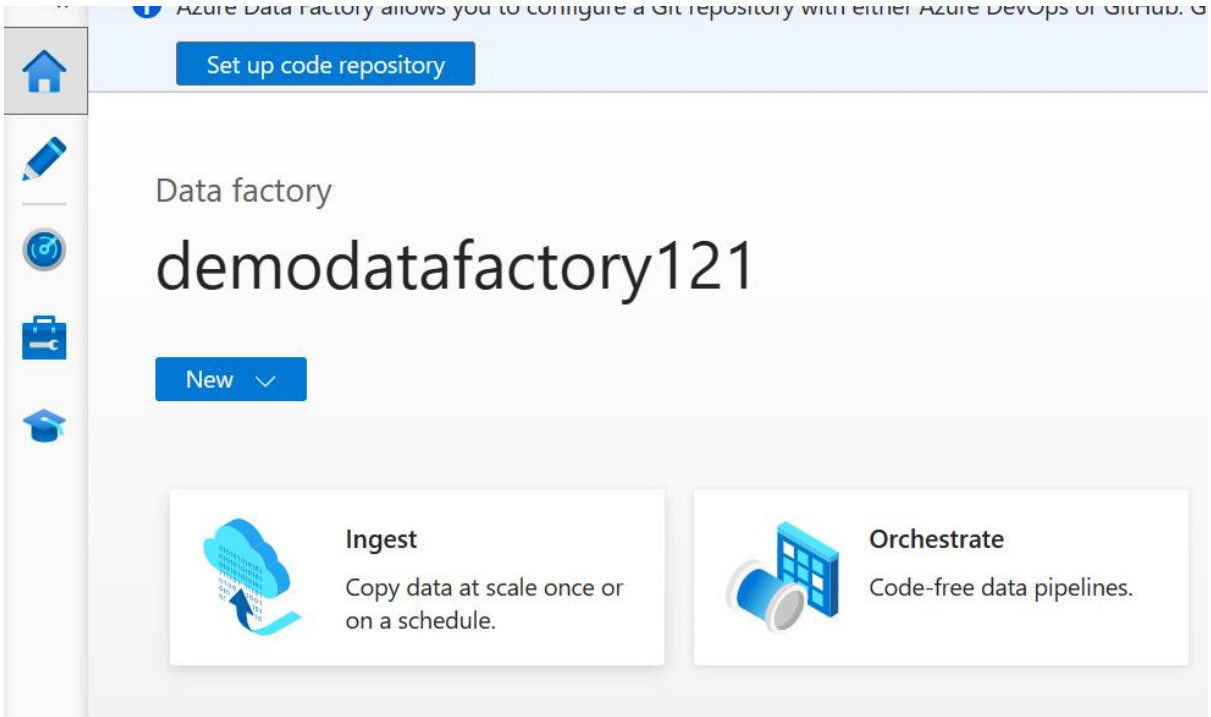
150 %

Messages

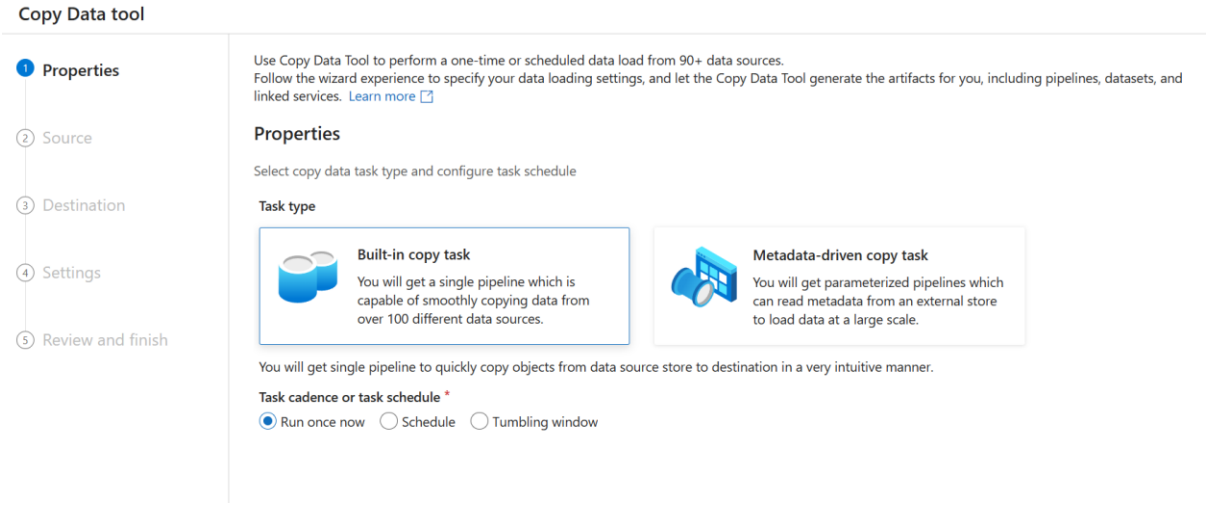
```
Commands completed successfully.

Completion time: 2024-04-24T16:44:19.3338815+05:30
```

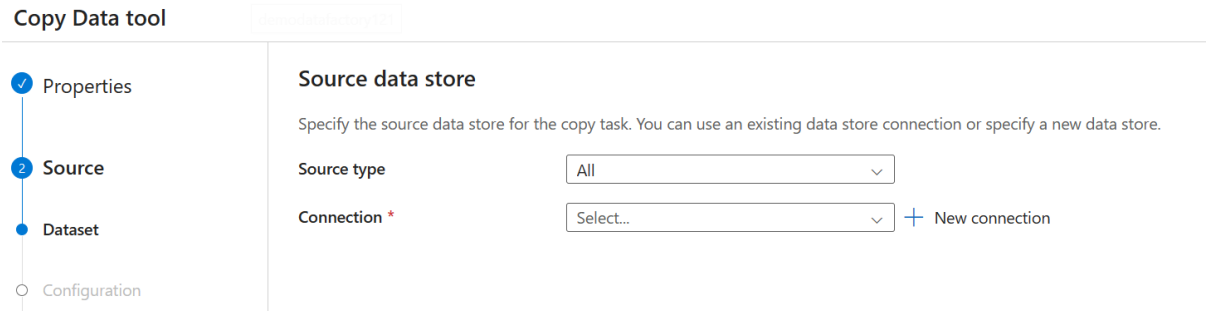
6. Now go to the homepage of your data factory wizard and click on ingest.



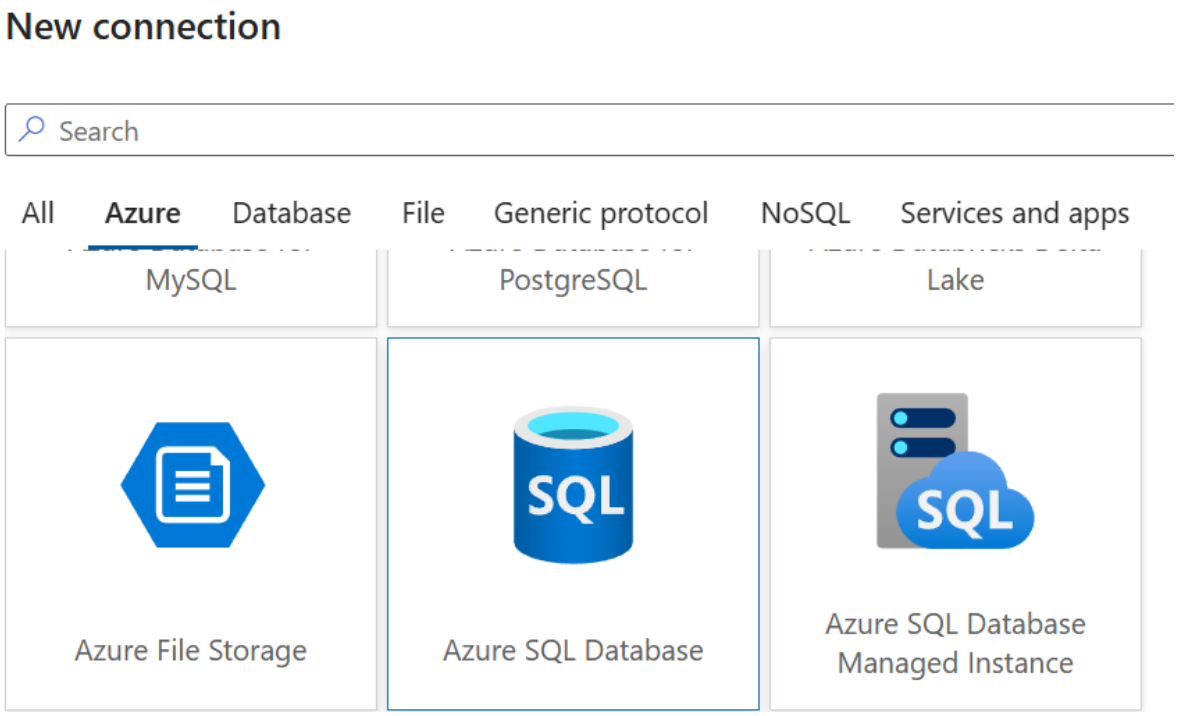
7. There we are going to create a pipeline using a built-in copy task. Click n next.



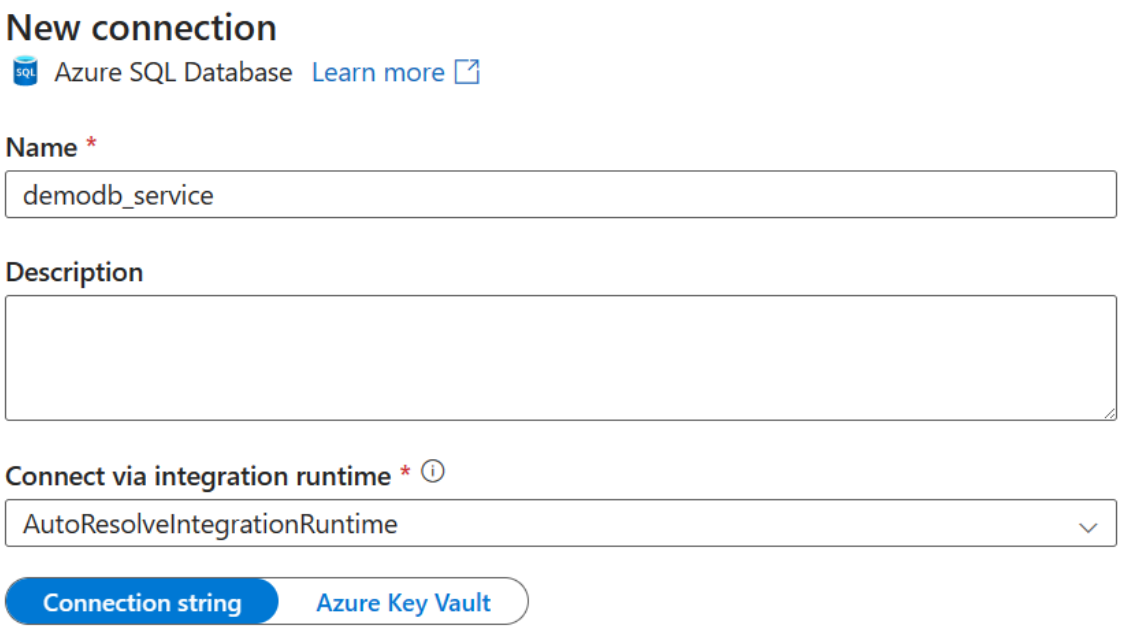

8. Then we will be needing a new connection to our Azure SQL Database.



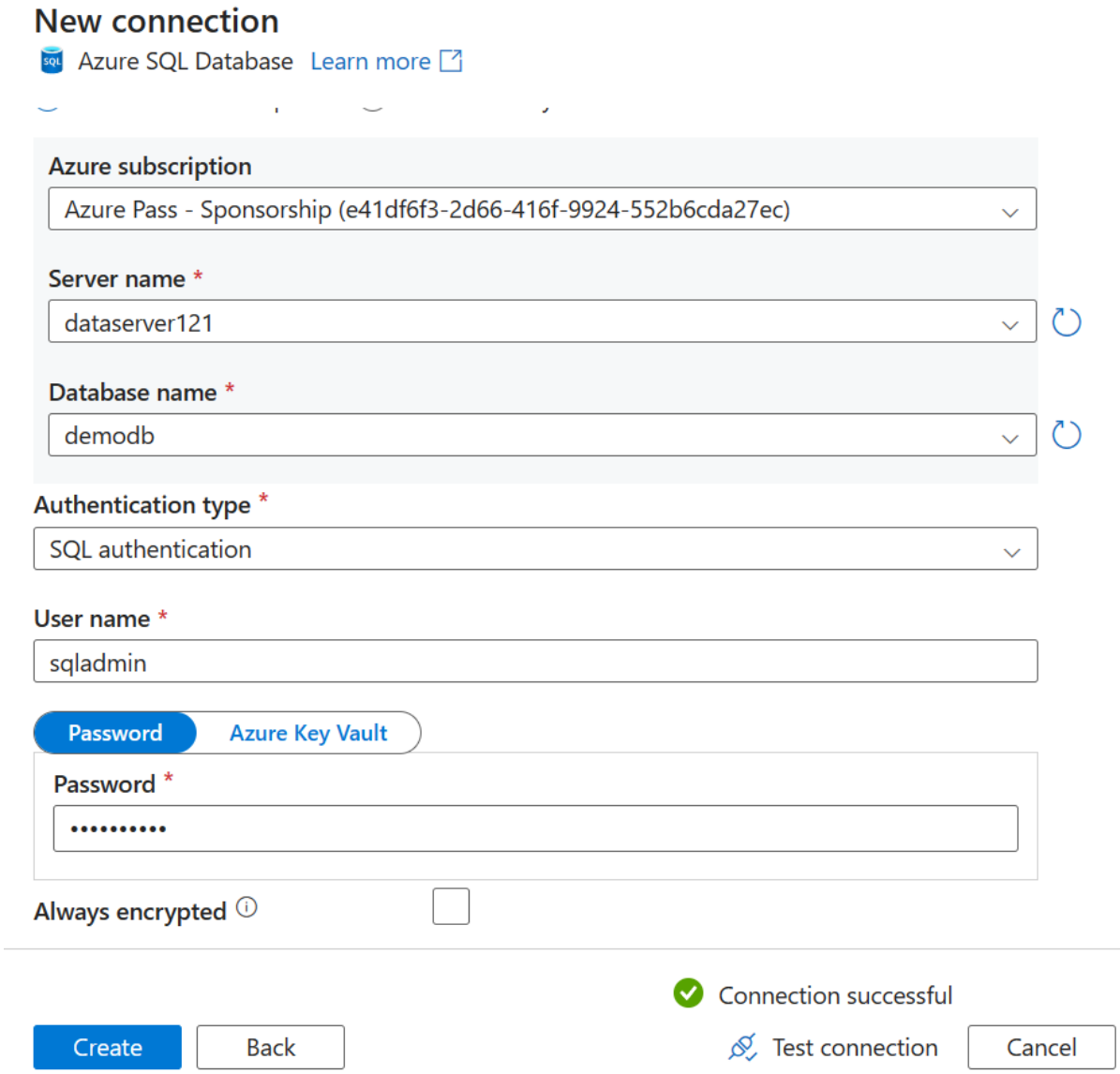

9. Scroll down and search for Azure SQL Database. Then hit on continue.



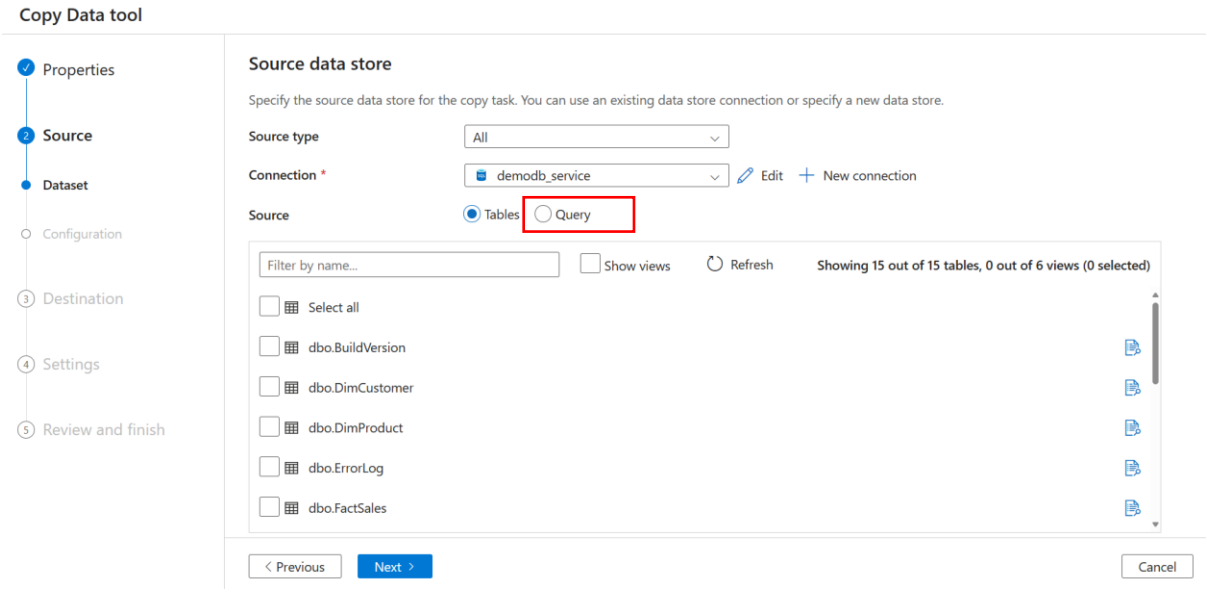

10. Give it a name and scroll down.



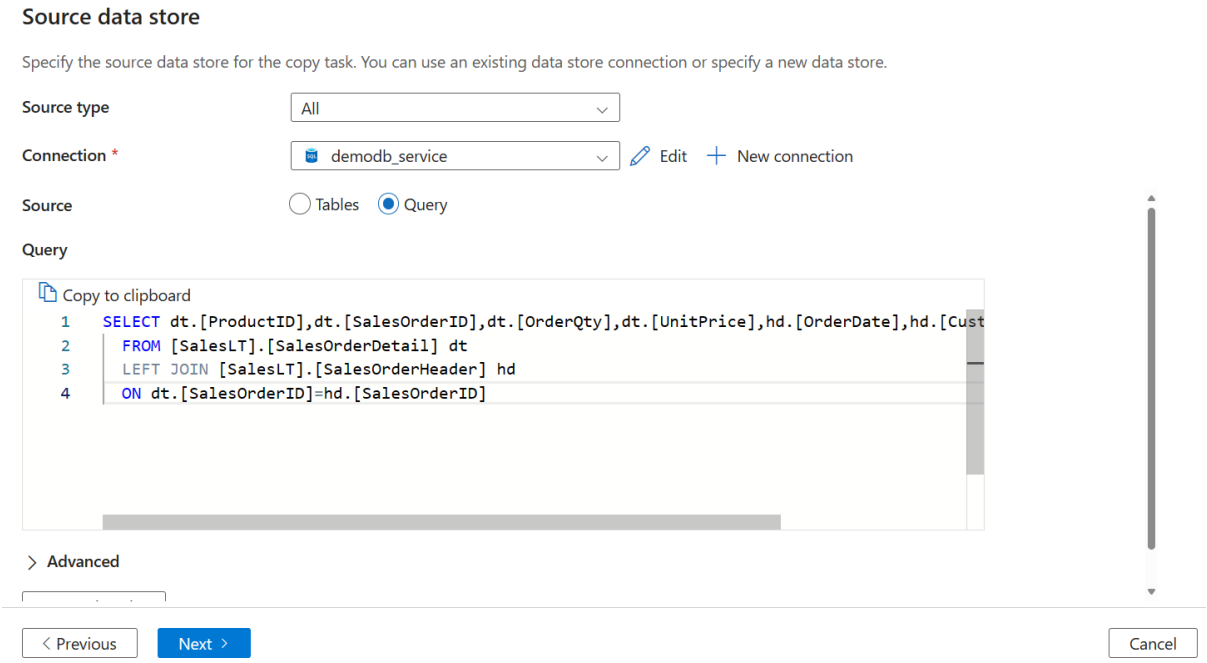

11. After that choose your subscription.
12. Then choose your server's name and database name.
13. In the authentication type give your username and password.
14. In the end test your connection.
15. At last if your connection is successful then click on create.



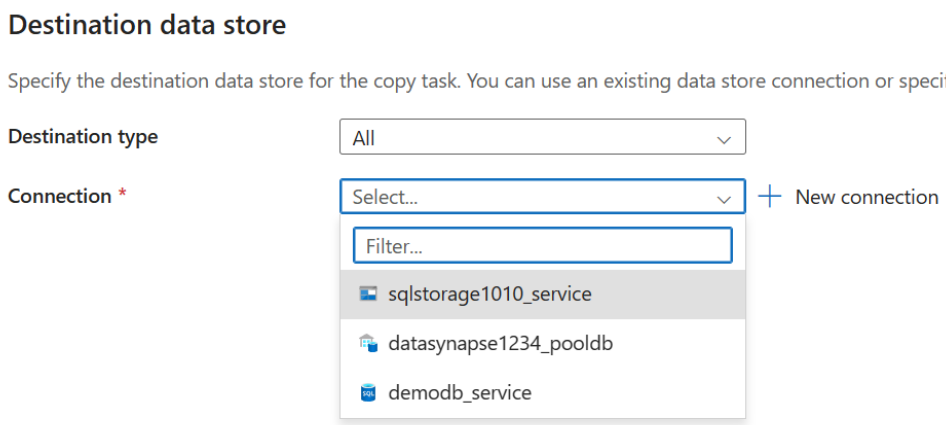

16. In Azure Synapse if you remember here, we have chosen our table from the list below. But this time we will be using the Query option.



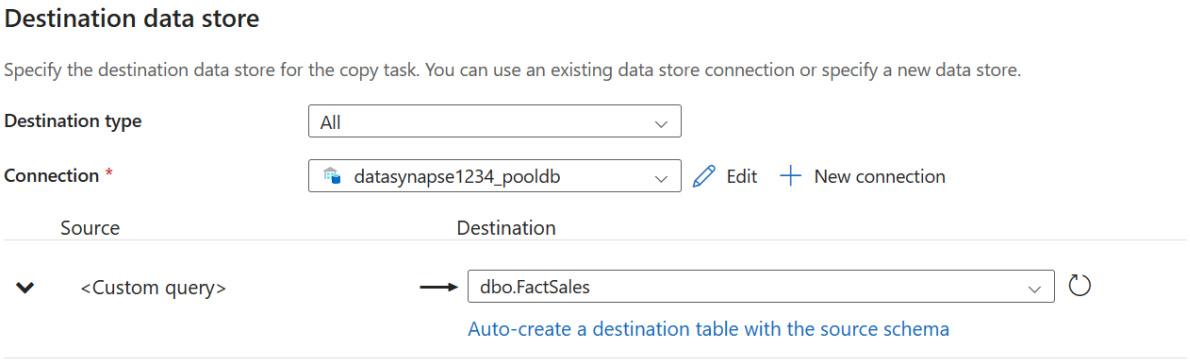

17. Then you have to paste the query which will help to build our fact sales table. Then click on next.



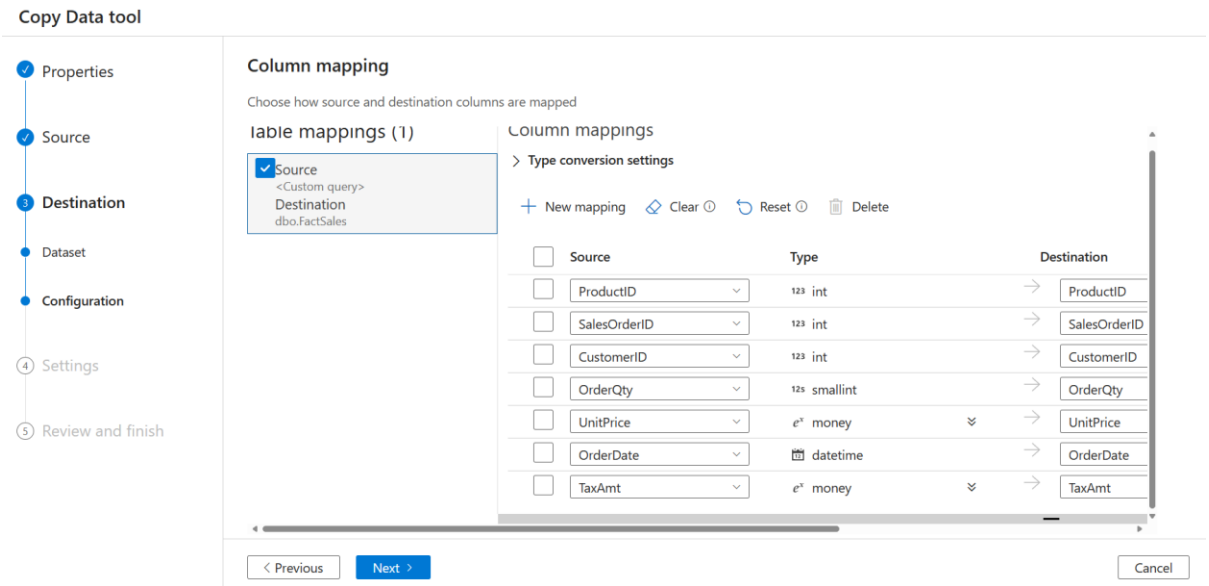

18. Now on the destination we will choose our pool db service.



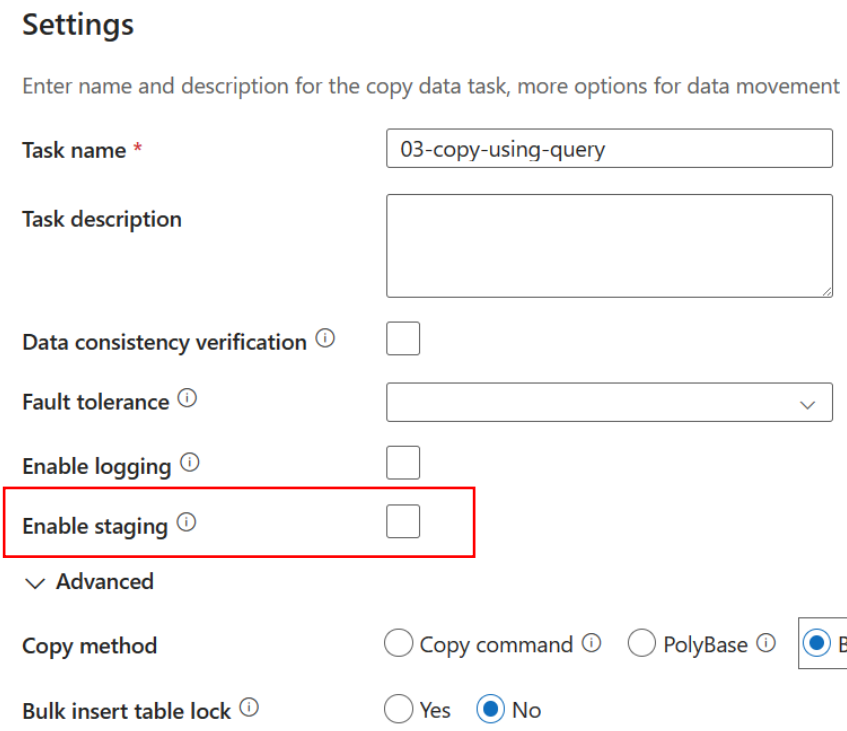

19. Here then you have to choose an existing table. Choose your fact sales table.



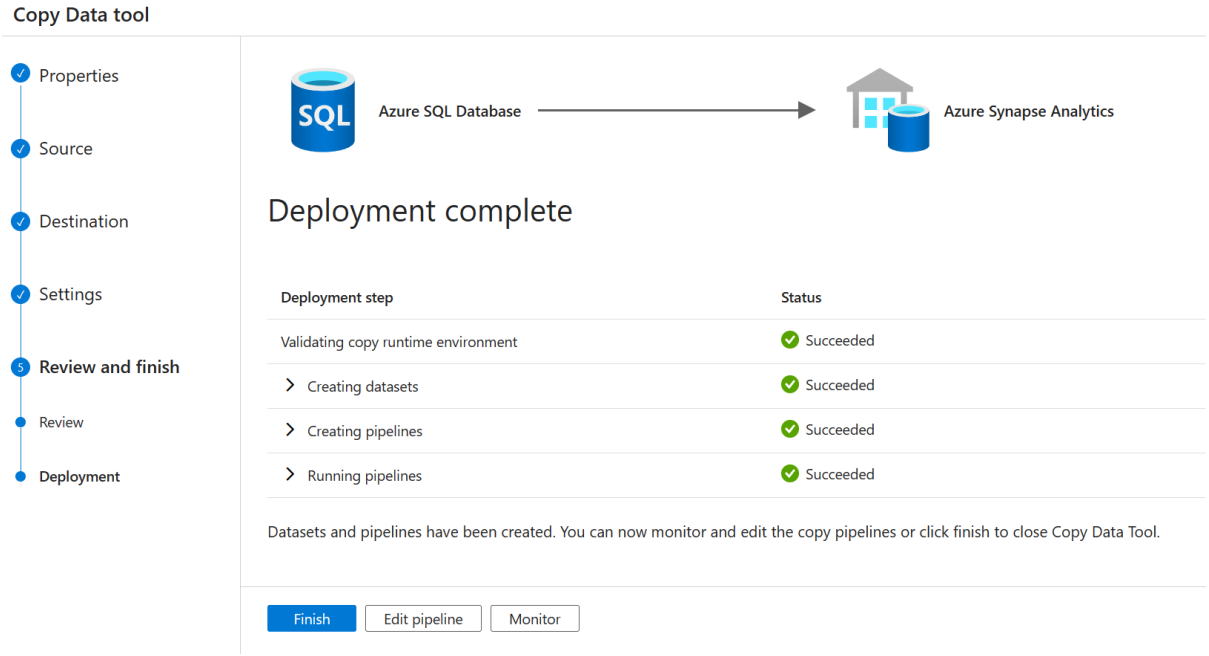

20. After that it will give you a column mapping. Click on next.



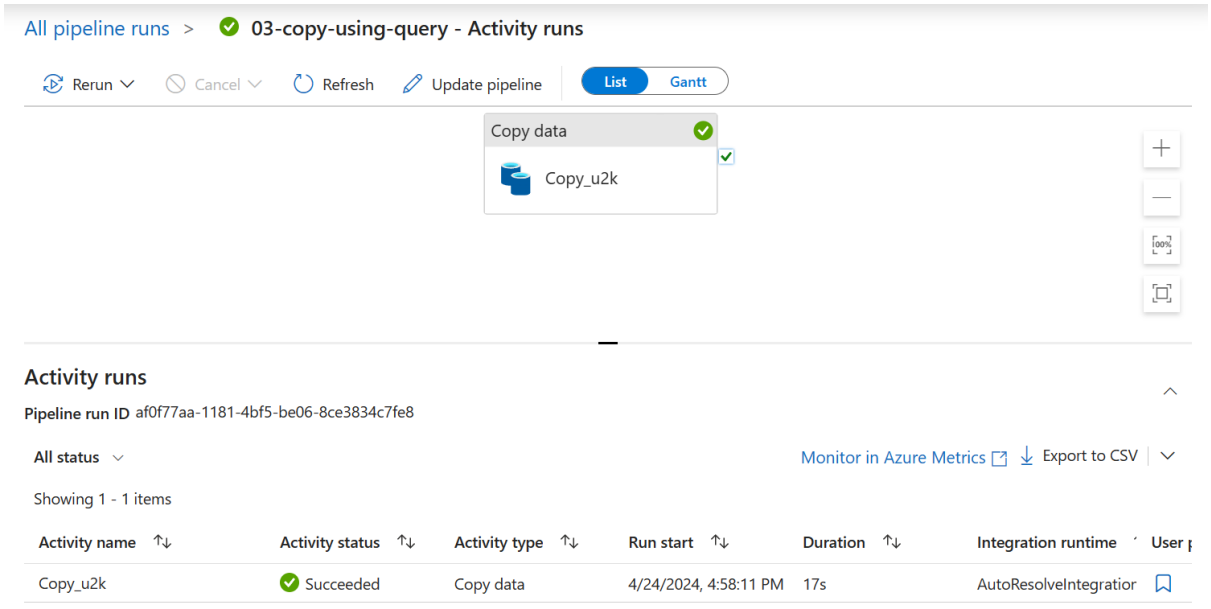

21. Now you have to give it a name and disable staging and choose bulk insert.



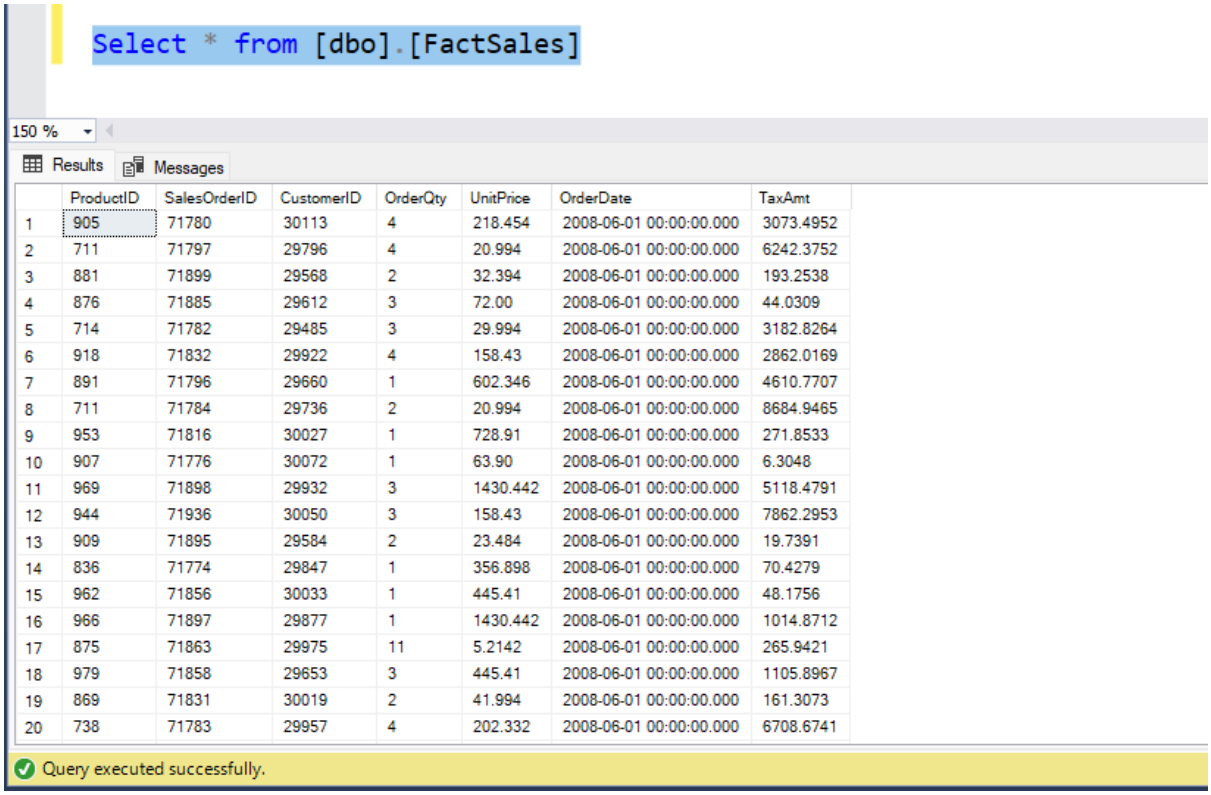

22. Then just click on next go to the review page and wait for the deployment to complete.
23. Once the deployment is completed then click on finish or you can directly go to the monitor section.




24. Below you can see that your query was a success.




25. Now come back to SSMS and run the select statement for fact sales table.
26. As expected below you can see the data.



```
Select * from [dbo].[FactSales]
```

150 %

Results | Messages

| | ProductID | SalesOrderID | CustomerID | OrderQty | UnitPrice | OrderDate | TaxAmt |
|---|---|---|---|---|---|---|---|
| 1 | 905 | 71780 | 30113 | 4 | 218.454 | 2008-06-01 00:00:00.000 | 3073.4952 |
| 2 | 711 | 71797 | 29796 | 4 | 20.994 | 2008-06-01 00:00:00.000 | 6242.3752 |
| 3 | 881 | 71899 | 29568 | 2 | 32.394 | 2008-06-01 00:00:00.000 | 193.2538 |
| 4 | 876 | 71885 | 29612 | 3 | 72.00 | 2008-06-01 00:00:00.000 | 44.0309 |
| 5 | 714 | 71782 | 29485 | 3 | 29.994 | 2008-06-01 00:00:00.000 | 3182.8264 |
| 6 | 918 | 71832 | 29922 | 4 | 158.43 | 2008-06-01 00:00:00.000 | 2862.0169 |
| 7 | 891 | 71796 | 29660 | 1 | 602.346 | 2008-06-01 00:00:00.000 | 4610.7707 |
| 8 | 711 | 71784 | 29736 | 2 | 20.994 | 2008-06-01 00:00:00.000 | 8684.9465 |
| 9 | 953 | 71816 | 30027 | 1 | 728.91 | 2008-06-01 00:00:00.000 | 271.8533 |
| 10 | 907 | 71776 | 30072 | 1 | 63.90 | 2008-06-01 00:00:00.000 | 6.3048 |
| 11 | 969 | 71898 | 29932 | 3 | 1430.442 | 2008-06-01 00:00:00.000 | 5118.4791 |
| 12 | 944 | 71936 | 30050 | 3 | 158.43 | 2008-06-01 00:00:00.000 | 7862.2953 |
| 13 | 909 | 71895 | 29584 | 2 | 23.484 | 2008-06-01 00:00:00.000 | 19.7391 |
| 14 | 836 | 71774 | 29847 | 1 | 356.898 | 2008-06-01 00:00:00.000 | 70.4279 |
| 15 | 962 | 71856 | 30033 | 1 | 445.41 | 2008-06-01 00:00:00.000 | 48.1756 |
| 16 | 966 | 71897 | 29877 | 1 | 1430.442 | 2008-06-01 00:00:00.000 | 1014.8712 |
| 17 | 875 | 71863 | 29975 | 11 | 5.2142 | 2008-06-01 00:00:00.000 | 265.9421 |
| 18 | 979 | 71858 | 29653 | 3 | 445.41 | 2008-06-01 00:00:00.000 | 1105.8967 |
| 19 | 869 | 71831 | 30019 | 2 | 41.994 | 2008-06-01 00:00:00.000 | 161.3073 |
| 20 | 738 | 71783 | 29957 | 4 | 202.332 | 2008-06-01 00:00:00.000 | 6708.6741 |

Query executed successfully.