



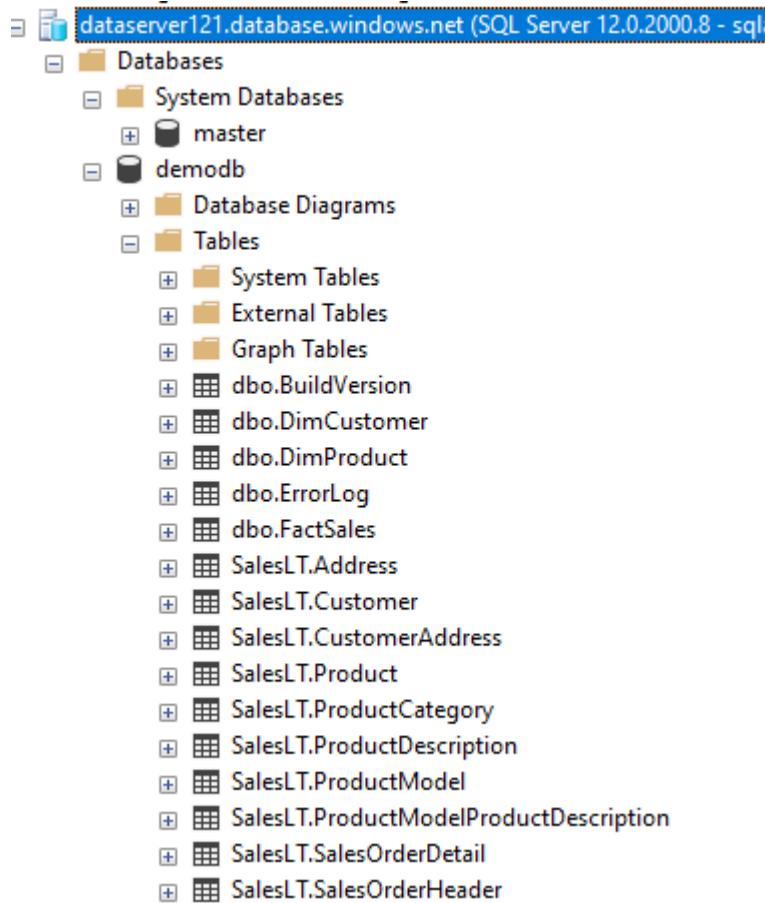
Using query for Data Transfer

- 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.

The screenshot shows the Azure Data Factory homepage. On the left, there's a vertical toolbar with icons for Home, Data factory, Pipeline, Dataset, and New. The Data factory icon is selected. The main area displays the name of the data factory: "demodatafactory121". Below the name are two large cards: "Ingest" and "Orchestrate". The "Ingest" card features an icon of a blue cloud with a white arrow pointing into it, and the text "Copy data at scale once or on a schedule.". The "Orchestrate" card features an icon of a blue cylinder with a grid, and the text "Code-free data pipelines.". At the top of the page, there's a banner with the text "Azure Data Factory allows you to configure a Git repository with either Azure DevOps or GitHub. G" and a "Set up code repository" button.

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

Copy Data tool

① Properties
② Source
③ Destination
④ Settings
⑤ Review and finish

Use Copy Data Tool to perform a one-time or scheduled data load from 90+ data sources. Follow the wizard experience to specify your data loading settings, and let the Copy Data Tool generate the artifacts for you, including pipelines, datasets, and linked services. [Learn more](#)

Properties
Select copy data task type and configure task schedule

Task type

Built-in copy task
You will get a single pipeline which is capable of smoothly copying data from over 100 different data sources.

Metadata-driven copy task
You will get parameterized pipelines which can read metadata from an external store to load data at a large scale.

You will get single pipeline to quickly copy objects from data source store to destination in a very intuitive manner.

Task cadence or task schedule *

Run once now Schedule Tumbling window

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

Copy Data tool

① Properties
② Source
③ Dataset
④ Configuration

Source data store
Specify the source data store for the copy task. You can use an existing data store connection or specify a new data store.

Source type: All

Connection *: Select... + New connection

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

New connection

Search

All **Azure** Database File Generic protocol NoSQL Services and apps

MySQL PostgreSQL Lake

Azure File Storage Azure SQL Database Azure SQL Database Managed Instance

10. Give it a name and scroll down.

New connection

 Azure SQL Database [Learn more](#) 

Name *

Description

Connect via integration runtime * 

Connection string

[Azure Key Vault](#)

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.

New connection

 Azure SQL Database [Learn more](#) 

Azure subscription

Azure Pass - Sponsorship (e41df6f3-2d66-416f-9924-552b6cda27ec) 

Server name *

dataserver121  

Database name *

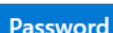
demodb  

Authentication type *

SQL authentication 

User name *

sqladmin

 Password

 Azure Key Vault

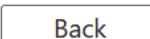
Password *

.....

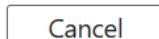
Always encrypted 

 Connection successful

 Create

 Back

 Test connection

 Cancel

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.

Copy Data tool

The screenshot shows the 'Source' step in the Copy Data tool. On the left, a navigation pane lists steps 1 through 5: Properties, Source, Dataset, Configuration, Destination, Settings, and Review and finish. Step 2, 'Source', is selected and highlighted with a blue circle. The main area is titled 'Source data store' and contains the following fields:

- Source type: A dropdown menu set to 'All'.
- Connection *: A dropdown menu set to 'demodb_service'. To its right are 'Edit' and 'New connection' buttons.
- Source: A radio button group where 'Tables' is selected (indicated by a red box), and 'Query' is the other option.

Below these settings is a list of tables from the source database:

- Filter by name... (text input field)
- Show views (checkbox)
- Refresh (button)
- Showing 15 out of 15 tables, 0 out of 6 views (0 selected)
- Table list:
 - Select all (checkbox)
 - dbo.BuildVersion
 - dbo.DimCustomer
 - dbo.DimProduct
 - dbo.ErrorLog
 - dbo.FactSales

At the bottom of the screen are navigation buttons: '< Previous', 'Next >', and 'Cancel'.

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

Source data store

Specify the source data store for the copy task. You can use an existing data store connection or specify a new data store.

The screenshot shows the 'Source' step in the Copy Data tool, specifically the 'Query' section. The interface includes:

- Source type: A dropdown menu set to 'All'.
- Connection *: A dropdown menu set to 'demodb_service'. To its right are 'Edit' and 'New connection' buttons.
- Source: A radio button group where 'Query' is selected (indicated by a blue circle), and 'Tables' is the other option.
- Query editor area:
 - Copy to clipboard (button)
 - Query text:

```
1  SELECT dt.[ProductID],dt.[SalesOrderID],dt.[OrderQty],dt.[UnitPrice],hd.[OrderDate],hd.[Cust
2    FROM [SalesLT].[SalesOrderDetail] dt
3    LEFT JOIN [SalesLT].[SalesOrderHeader] hd
4    ON dt.[SalesOrderID]=hd.[SalesOrderID]
```
- Advanced options (button)
- Navigation buttons: '< Previous', 'Next >', and 'Cancel'.

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

Destination data store

Specify the destination data store for the copy task. You can use an existing data store connection or specify a new data store.

Destination type

Connection *

sqlstorage1010_service
 datasynapse1234_pooldb
 demodb_service

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

Destination data store

Specify the destination data store for the copy task. You can use an existing data store connection or specify a new data store.

Destination type

Connection *

Source	Destination
<input checked="" type="checkbox"/> <Custom query>	<input type="button" value="→"/> <input type="button" value="dbo.FactSales"/> <input type="button" value="↻"/> Auto-create a destination table with the source schema

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

Copy Data tool

Properties
Source
3 Destination
Dataset
Configuration
4 Settings
5 Review and finish

Column mapping
Choose how source and destination columns are mapped

Table mappings (1)

<input checked="" type="checkbox"/> Source <Custom query> Destination dbo.FactSales
--

Column mappings
Type conversion settings

+ New mapping

Source	Type	Destination
ProductID	123 int	ProductID
SalesOrderID	123 int	SalesOrderID
CustomerID	123 int	CustomerID
OrderQty	12s smallint	OrderQty
UnitPrice	e ^x money	UnitPrice
OrderDate	datetime	OrderDate
TaxAmt	e ^x money	TaxAmt

< Previous Cancel

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

Settings

Enter name and description for the copy data task, more options for data movement

Task name *

03-copy-using-query

Task description

Data consistency verification ⓘ

Fault tolerance ⓘ

Enable logging ⓘ

Enable staging ⓘ

✓ Advanced

Copy method

Copy command ⓘ

PolyBase ⓘ

Bulk insert

Upsert

Bulk insert table lock ⓘ

Yes

No

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.

Copy Data tool

- Properties
- Source
- Destination
- Settings
- Review and finish
- Review
- Deployment



Azure SQL Database



Azure Synapse Analytics

Deployment complete

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

Datasets and pipelines have been created. You can now monitor and edit the copy pipelines or click finish to close Copy Data Tool.

Finish

Edit pipeline

Monitor

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

All pipeline runs > ✓ 03-copy-using-query - Activity runs

Rerun Cancel Refresh Update pipeline List Gantt

Copy data ✓
Copy_u2k

Activity runs

Pipeline run ID af0f77aa-1181-4bf5-be06-8ce3834c7fe8

All status ▾ Monitor in Azure Metrics Export to CSV ▾

Showing 1 - 1 items

Activity name	Activity status	Activity type	Run start	Duration	Integration runtime	User
Copy_u2k	✓ Succeeded	Copy data	4/24/2024, 4:58:11 PM	17s	AutoResolveIntegration	View

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.