

Create a star schema

Note:

You are not required to complete the processes, tasks, activities, or steps presented in this example. The various samples provided are for illustrative purposes only and it's likely that if you try this out you will encounter issues in your system.

In this exercise, you create a star schema in Azure Synapse dedicated pool. The first step is to create the base dimension and fact tables. You may notice some differences with creating tables in Synapse Analytics vs. SQL database, but the same data modeling principles apply.

When designing a star schema model for small or medium sized datasets you can use your preferred database, such as Azure SQL. For larger data sets you may benefit from implementing your data warehouse in Azure Synapse Analytics instead of SQL Server. It is important to understand some key differences when creating tables in Synapse Analytics.


In Synapse, you do not have foreign keys and unique value constraints like you do in SQL Server. Since these rules are not enforced at the database layer, the jobs used to load data have more responsibility to maintain data integrity. You still have the option to use clustered indexes, but for most dimension tables in Synapse you will benefit from using a clustered column store index (CCI). In this example, a few tables have data types which cannot be included in a clustered column store index so a clustered index was used instead.

Since Synapse Analytics is a [massively parallel processing](#) (MPP) system, you must consider how data is distributed in your table design, as opposed to symmetric multiprocessing (SMP) systems, such as OLTP databases like Azure SQL Database. The table category often determines which option to choose for distributing the table.



Table Category	Recommend distribution option
Fact	Use hash-distribution with clustered column store index. Performance improves when two hash tables are joined on the same distribution column.
Dimension	Use replicated for smaller tables. If tables are too large to store on each Compute node, use hash-distributed.
Staging	Use round-robin for the staging table. The load with CTAS is fast. Once the data is in the staging table, use INSERT...SELECT to move the data to production tables.

In the case of the dimension tables in this exercise, the amount of data stored per table falls well within the criteria for using a replicated distribution.

1. Sign in to the Azure portal (<https://portal.azure.com>).
2. Open the resource group that contains your Synapse workspace, then select the **Synapse workspace**.






<input type="checkbox"/> Name ↑↓	Type ↑↓
<input type="checkbox"/>  asagadatalakejdh013121	Storage account
<input type="checkbox"/>  asagakeyvaultjdh013121	Key vault
<input type="checkbox"/>  asagaworkspacejdh013121	Synapse workspace
<input type="checkbox"/>  dp203sqljdh013121	SQL server
<input type="checkbox"/>  SourceDB (dp203sqljdh013121/SourceDB)	SQL database
<input type="checkbox"/>  SQLPool01 (asagaworkspacejdh013121/SQLPool01)	Dedicated SQL pool

3. In your Synapse workspace Overview blade, select the **Open** link within **Open Synapse Studio**.




 **asagaworkspacejdh013121** 

Synapse workspace



[+ New dedicated SQL pool](#)
[+ New Apache Spark pool](#)
[Refresh](#)
[Reset SQL admin password](#)

 Overview
  Activity log
  Access control (IAM)
  Tags
  Diagnose and solve problems

Settings

-  SQL Active Directory admin
-  Properties
-  Locks

Analytics pools

-  SQL pools
-  Apache Spark pools

Security

Resource group (change) : [dp203-labs](#)
 Firewalls

Status : Succeeded
 Primary ADLS Gen2 a

Location : South Central US
 Primary ADLS Gen2 f

Subscription (change) : [Synapse Analytics Service and jobs](#)
 SQL admin username

Subscription ID : [8b8b8b8b-8b8b-8b8b-8b8b-8b8b8b8b8b8b](#)
 SQL Active Directory



Managed virtual network : No
 Dedicated SQL endpo



Managed Identity object ... : [https://msi.azure.com/identity/objects/...](#)
 Serverless SQL endpo

Workspace web URL : <https://web.azuresynapse.net?workspace=%2fsu...>
 Development endpoi

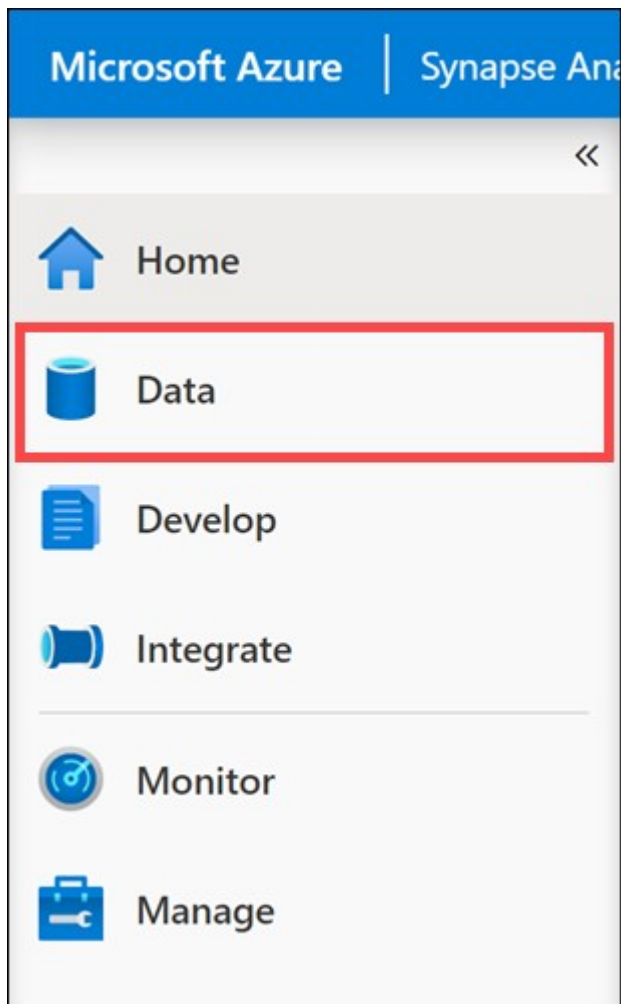
Tags (change) : [Click here to add tags](#)

Getting started

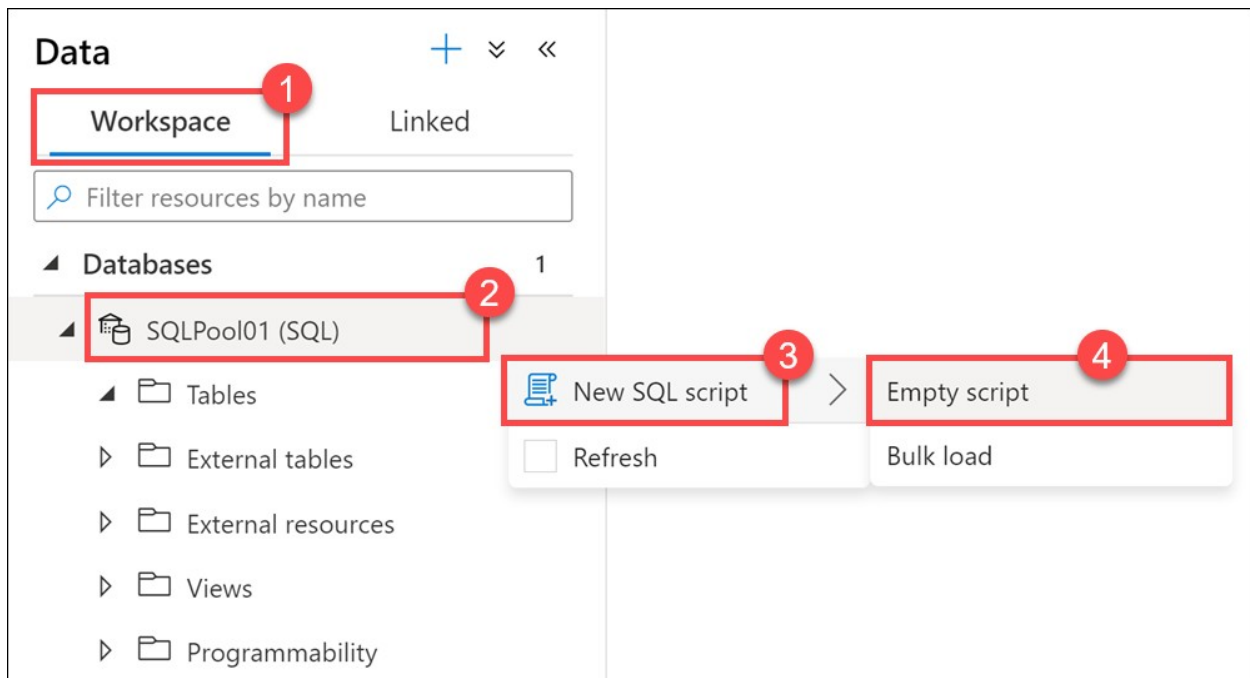

Open Synapse Studio
 Start building your fully-integrated analytics solution and unlock new insights.
[Open](#) 


Read documentation
 Learn how to be productive quickly. Explore concepts, tutorials, and samples.
[Learn more](#) 

4. In Synapse Studio, navigate to the **Data** hub.



5. Select the **Workspace** tab **(1)**, expand Databases, then right-click on your **SQLPool01 (2)**. Select **New SQL script (3)**, then select **Empty script (4)**.



6. Paste the following script into the empty script window, then select **Run** or hit **F5** to execute the query.

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24

25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40

```
CREATE TABLE [dbo].[FactResellerSales](
    [ProductKey] [int] NOT NULL,
    [OrderDateKey] [int] NOT NULL,
    [DueDateKey] [int] NOT NULL,
    [ShipDateKey] [int] NOT NULL,
    [ResellerKey] [int] NOT NULL,
    [EmployeeKey] [int] NOT NULL,
    [PromotionKey] [int] NOT NULL,
    [CurrencyKey] [int] NOT NULL,
    [SalesTerritoryKey] [int] NOT NULL,
    [SalesOrderNumber] [nvarchar](20) NOT NULL,
    [SalesOrderLineNumber] [tinyint] NOT NULL,
    [RevisionNumber] [tinyint] NULL,
    [OrderQuantity] [smallint] NULL,
    [UnitPrice] [money] NULL,
    [ExtendedAmount] [money] NULL,
    [UnitPriceDiscountPct] [float] NULL,
    [DiscountAmount] [float] NULL,
    [ProductStandardCost] [money] NULL,
    [TotalProductCost] [money] NULL,
    [SalesAmount] [money] NULL,
    [TaxAmt] [money] NULL,
    [Freight] [money] NULL,
    [CarrierTrackingNumber] [nvarchar](25) NULL,
    [CustomerPONumber] [nvarchar](25) NULL,
    [OrderDate] [datetime] NULL,
    [DueDate] [datetime] NULL,
    [ShipDate] [datetime] NULL
)
```

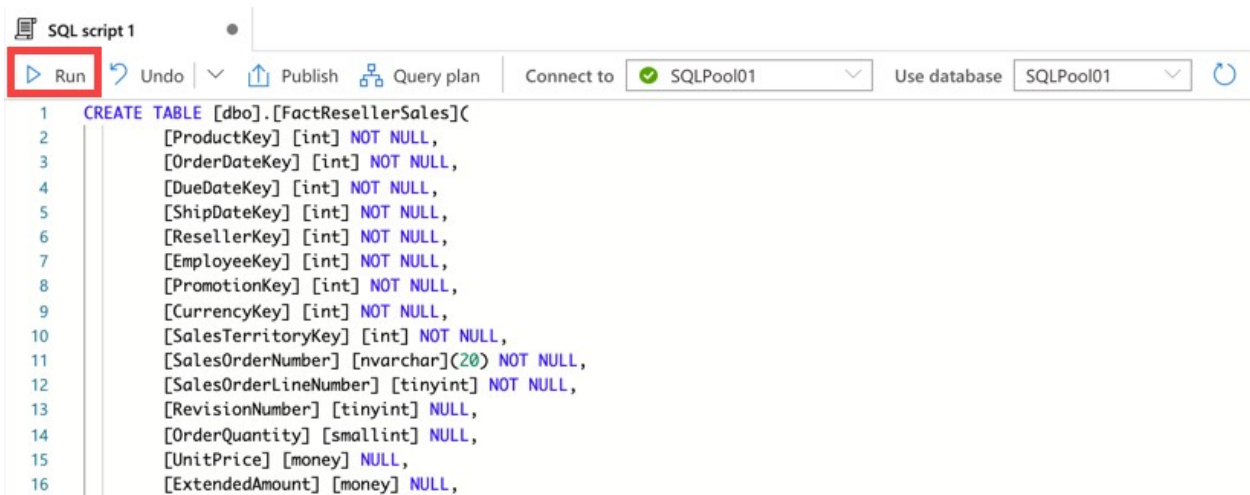
```

WITH
(
    DISTRIBUTION = HASH([SalesOrderNumber]),
    CLUSTERED COLUMNSTORE INDEX
);
GO

CREATE TABLE [dbo].[DimReseller](
    [ResellerKey] [int] NOT NULL,
    [GeographyKey] [int] NULL,
    [ResellerAlternateKey] [nvarchar](15) NULL,

```

You will find **Run** in the top left corner of the script window.



7. Replace **and execute** the following query to insert data into the fact and dimension tables:

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17

18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40

```
COPY INTO [dbo].[DimProduct]
FROM 'https://solliancepublicdata.blob.core.windows.net/dataengineering/dp-203/awdata/DimProduct.csv'
WITH (
    FILE_TYPE='CSV',
    FIELDTERMINATOR='|',
    FIELDQUOTE='',
    ROWTERMINATOR='\n',
    ENCODING = 'UTF16'
);
GO
```

```
COPY INTO [dbo].[DimReseller]
FROM 'https://solliancepublicdata.blob.core.windows.net/dataengineering/dp-203/awdata/DimReseller.csv'
WITH (
    FILE_TYPE='CSV',
    FIELDTERMINATOR='|',
    FIELDQUOTE='',
    ROWTERMINATOR='\n',
    ENCODING = 'UTF16'
);
```

GO

```
COPY INTO [dbo].[DimEmployee]
FROM 'https://solliancepublicdata.blob.core.windows.net/dataengineering/dp-203/
awdata/DimEmployee.csv'
WITH (
    FILE_TYPE='CSV',
    FIELDTERMINATOR='|',
    FIELDQUOTE='',
    ROWTERMINATOR='\n',
    ENCODING = 'UTF16'
);
GO
```

```
COPY INTO [dbo].[DimGeography]
FROM 'https://solliancepublicdata.blob.core.windows.net/dataengineering/dp-203/
awdata/DimGeography.csv'
WITH (
    FILE_TYPE='CSV',
    FIELDTERMINATOR='|',
    FIELDQUOTE='',
    ROWTERMINATOR='\n',
```

8. Replace and execute the following query to retrieve reseller sales data from the star schema at the reseller location, product, and month granularity:

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20

21
22
23
24
25
26
27
28
29
30
31
32

```
SELECT
    Coalesce(p.[ModelName], p.[EnglishProductName]) AS [Model]
    ,g.City AS ResellerCity
    ,g.StateProvinceName AS StateProvince
    ,Year(f.OrderDate) AS CalendarYear
    ,CASE
        WHEN Month(f.OrderDate) < 7 THEN Year(f.OrderDate)
        ELSE Year(f.OrderDate) + 1
    END AS FiscalYear -- Fiscal year runs from Jul to June)
    ,Month(f.OrderDate) AS [Month]
    ,Sum(f.OrderQuantity) AS Quantity
    ,Sum(f.ExtendedAmount) AS Amount
    ,Approx_count_distinct(f.SalesOrderNumber) AS UniqueOrders
FROM
    [dbo].[FactResellerSales] f
INNER JOIN [dbo].[DimReseller] r
    ON f.ResellerKey = r.ResellerKey
INNER JOIN [dbo].[DimGeography] g
    ON r.GeographyKey = g.GeographyKey
INNER JOIN [dbo].[DimProduct] p
    ON f.[ProductKey] = p.[ProductKey]
GROUP BY
    Coalesce(p.[ModelName], p.[EnglishProductName])
    ,g.City
    ,g.StateProvinceName
    ,Year(f.OrderDate)
    ,CASE
        WHEN Month(f.OrderDate) < 7 THEN Year(f.OrderDate)
        ELSE Year(f.OrderDate) + 1
    END
    ,Month(f.OrderDate)
ORDER BY Amount DESC
```

You should see an output similar to the following:

Results Messages

View Table Chart Export results

Search

Model	ResellerCity	StateProvince	CalendarYear	FiscalYear	Month	Quantity	Amount	UniqueOrders
Mountain-100	Seattle	Washington	2011	2011	3	272	552838.3680	2
Mountain-100	Toronto	Ontario	2011	2011	3	268	544738.3920	5
Mountain-100	Toronto	Ontario	2011	2012	8	256	520498.4640	4
Touring-1000	Sand City	California	2013	2013	3	372	483489.3960	1
Mountain-100	Orlando	Florida	2011	2011	5	236	473248.6024	1
Mountain-100	Orlando	Florida	2011	2011	1	224	448444.6760	1
Mountain-100	Toronto	Ontario	2011	2011	5	216	438898.7040	4
Mountain-100	Moline	Illinois	2011	2011	5	208	422698.7520	1
Mountain-100	Minneapolis	Minnesota	2011	2011	1	208	419250.7632	1
Mountain-100	Moline	Illinois	2011	2012	10	188	382498.8720	1
Mountain-100	Minneapolis	Minnesota	2011	2011	5	184	373858.8960	1
Mountain-100	La Mesa	California	2011	2011	5	180	362974.9296	1
Mountain-200	Toronto	Ontario	2012	2012	2	292	358314.1604	3
Mountain-100	Gulfport	Mississippi	2011	2011	5	176	358018.9440	1
Mountain-100	Minneapolis	Minnesota	2011	2012	10	176	357898.9440	1
Mountain-100	City Of Commerce	California	2011	2011	5	176	357178.9440	1
Mountain-200	Toronto	Ontario	2012	2012	5	292	357055.5944	3
Touring-1000	Sand City	California	2012	2013	12	292	350458.2900	1
Touring-1000	London	England	2013	2013	1	300	349981.4760	2

Remember to **pause your SQL Pool** to avoid extra cost if you are not continuing to another exercise.