

EXPERIMENT 1

Aim: Experiment to design Data Warehouse for given case study and perform ETL and OLAP operations on it.

Theory:

Data warehouse

A Data Warehouse (DW) is a relational database that is designed for query and analysis rather than transaction processing. It includes historical data derived from transaction data from single and multiple sources. A Data Warehouse provides integrated, enterprise-wide, historical data and focuses on providing support for decision-makers for data modeling and analysis. A Data Warehouse is a group of data specific to the entire organization, not only to a particular group of users. It is not used for daily operations and transaction processing but used for making decisions.

A Data Warehouse can be viewed as a data system with the following attributes:

- It is a database designed for investigative tasks, using data from various applications.
- It supports a relatively small number of clients with relatively long interactions.
- It includes current and historical data to provide a historical perspective of information.
- Its usage is read-intensive.
- It contains a few large tables.

"Data Warehouse is a subject-oriented, integrated, and time-variant store of information in support of management's decisions."

Characteristics of Data Warehouse

1. Subject-Oriented

A data warehouse target on the modeling and analysis of data for decision-makers. Therefore, data warehouses typically provide a concise and straightforward view around a particular subject, such as customer, product, or sales, instead of the global organization's ongoing operations. This is done by excluding data that are not useful concerning the subject and including all data needed by the users to understand the subject.

2. Integrated

A data warehouse integrates various heterogeneous data sources like RDBMS, flat files, and online transaction records. It requires performing data cleaning and integration during data warehousing to ensure consistency in naming conventions, attributes types, etc., among different data sources.

3. Time-Variant

Historical information is kept in a data warehouse. For example, one can retrieve files from 3 months, 6 months, 12 months, or even previous data from a data warehouse. These variations with a transactions system, where often only the most current file is kept.

4. Non-Volatile

The data warehouse is a physically separate data storage, which is transformed from the source operational RDBMS. The operational updates of data do not occur in the data warehouse, i.e., update, insert, and delete operations are not performed. It usually requires only two procedures in data accessing: Initial loading of data and access to data. Therefore, the DW does not require transaction processing, recovery, and concurrency capabilities, which allows for substantial speedup of data retrieval. Non-Volatile defines that once entered into the warehouse, and data should not change.

Data Warehouse Design

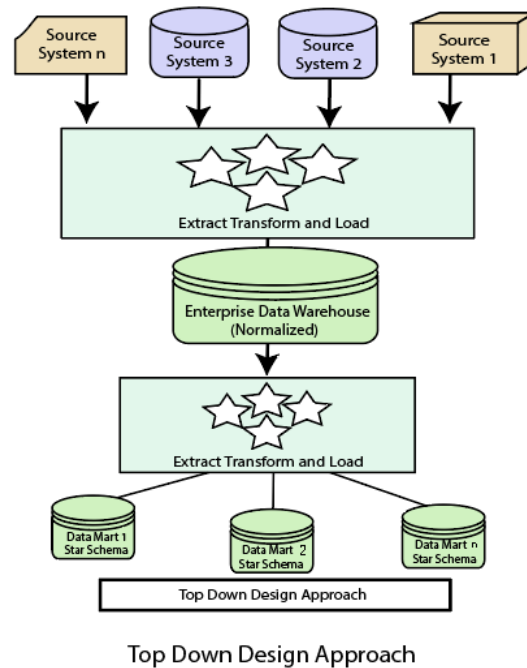
A data warehouse is a single data repository where a record from multiple data sources is integrated for online business analytical processing (OLAP). This implies a data warehouse needs to meet the requirements from all the business stages within the entire organization. Thus, data warehouse design is a hugely complex, lengthy, and hence error-prone process. Data warehouse design takes a method different from view materialization in the industries. It sees data warehouses as database systems with particular needs such as answering management related queries. The target of the design becomes how the record from multiple data sources should be extracted, transformed, and loaded (ETL) to be organized in a database as the data warehouse.

There are two approaches

1. "top-down" approach
2. "bottom-up" approach

Top-down Design Approach

In the "Top-Down" design approach, a data warehouse is described as a subject-oriented, time-variant, non-volatile and integrated data repository for the entire enterprise data from different sources are validated, reformatted and saved in a normalized (up to 3NF) database as the data warehouse. The data warehouse stores "atomic" information, the data at the lowest level of granularity, from which dimensional data marts can be built by selecting the data required for specific business subjects or particular departments. An approach is a data-driven approach as the information is gathered and integrated first and then business requirements by subjects for building data marts are formulated. The advantage of this method is which it supports a single integrated data source. Thus data marts built from it will have consistency when they overlap.



Components of the Top-Down Approach

A. The External Sources:

The data or the raw data is collected from the external source, that is, the source of truth, which the organization needs to decide as a best practice for designing the architecture of the data warehouse. The external source is a source from where the data is collected, irrespective of the type of data. The Data is mainly of three different forms, which are :

- Structured (CSV, excel sheets, relational database, etc)
- Semi-structured (HTML, JSON, XML)
- Unstructured (audio, video, pdf, etc)

B. The Stage Area:

After the extraction of data is done from the external sources, we see that the data does not follow a particular format, that is, some are logical values, numerical values, etc, so to make the standardized data format we need to validate this source data before we load into the data warehouse. And to solve this we have the ETL tool. The ETL tool or the Extract-Transform-Load tool helps in cleansing and transforming the data to serve the business needs.

E(Extracted): Here, the raw data is extracted from the external data source.

T(Transform): Here, the raw data that we received is transformed into the standard format, which is universally acceptable and reliable, along with serving the business needs. We make use of the Query Tools at this stage of data transformation.

L(Load): Here, the refined-transformed data is loaded into the data warehouse for further analysis to gain insights that can help a business grow.

C. The Data Marts:

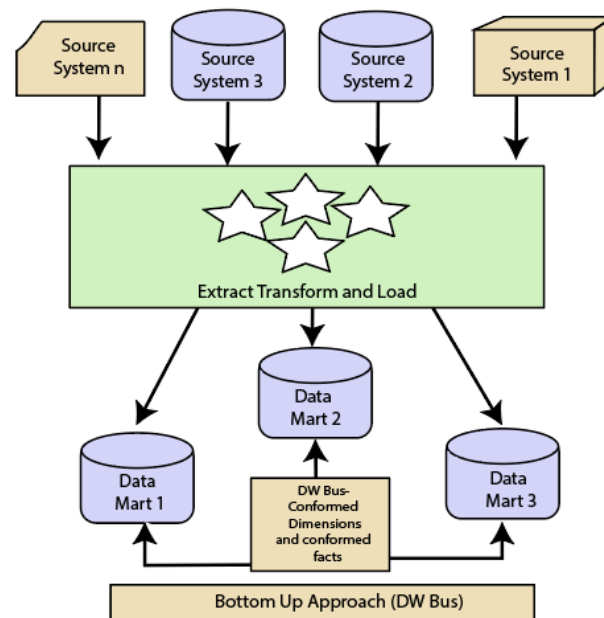
The third component in the Top-down approach while designing the architecture of the data warehouse is the Data Mart, which can also be suggested as part of the storage component. The data mart stores the information or the transformed data, which is of a particular function/theme of an enterprise mostly handled by a single authority.

D. The Data Mining:

The fourth major component in the Top-down approach while designing the architecture of the data warehouse is data mining. The raw data was gathered at the External source after the transformation reached the data warehouse. Now this cleansed and transformed data will be of no use until the analyst makes the best use of the same. So data mining can be explained as the ability to analyze the transformed data to find out the hidden patterns that are present in the database or the data warehouse with the help of an algorithm of data mining.

Bottom-Up Design Approach

In the "Bottom-Up" approach, a data warehouse is described as "a copy of transaction data specific architecture for query and analysis," term the star schema. In this approach, a data mart is created first to necessary reporting and analytical capabilities for particular business processes (or subjects). Thus it is needed to be a business-driven approach in contrast to Inmon's data-driven approach. Data marts include the lowest grain data and, if needed, aggregated data too. Instead of a normalized database for the data warehouse, a denormalized dimensional database is adapted to meet the data delivery requirements of data warehouses. The conformed dimensions connected the data marts to form a data warehouse, which is generally called a virtual data warehouse.

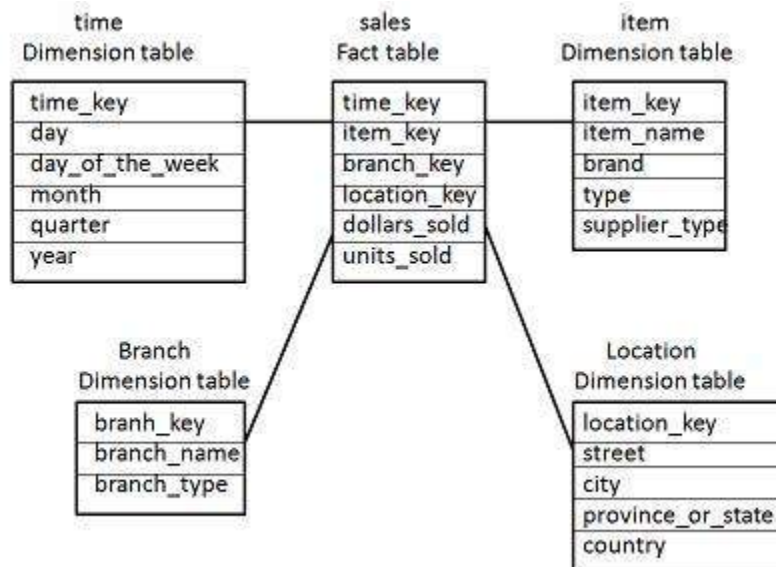


Dimensional Schema

Schema is a logical description of the entire database. It includes the name and description of records of all record types including all associated data-items and aggregates. Much like a database, a data warehouse also requires maintaining a schema. A database uses relational models, while a data warehouse uses Star, Snowflake, and Fact Constellation schema. In this chapter, we will discuss the schemas used in a data warehouse.

1. Star Schema

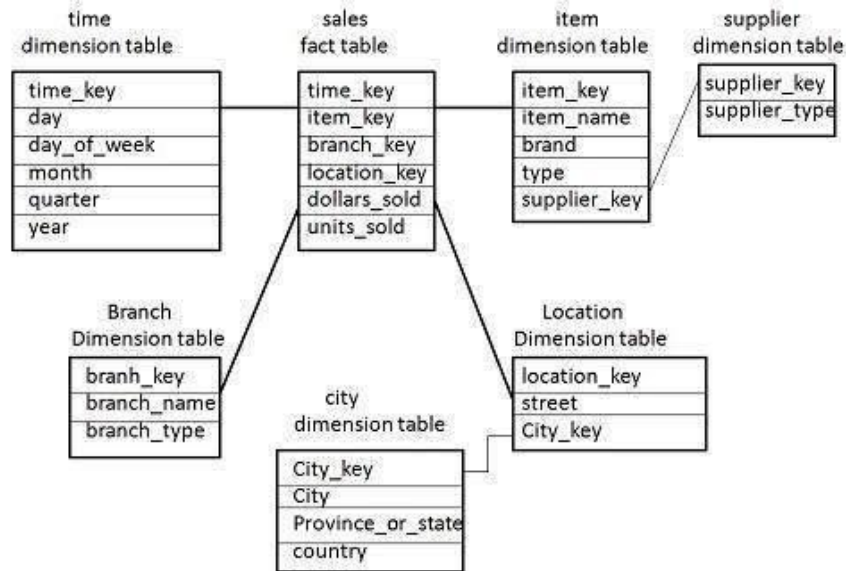
- Each dimension in a star schema is represented with only one-dimension table.
- This dimension table contains the set of attributes.
- The following diagram shows the sales data of a company with respect to the four dimensions, namely time, item, branch, and location.



- There is a fact table at the center. It contains the keys to each of four dimensions.
- The fact table also contains the attributes, namely dollars sold and units sold.

2. Snowflake Schema

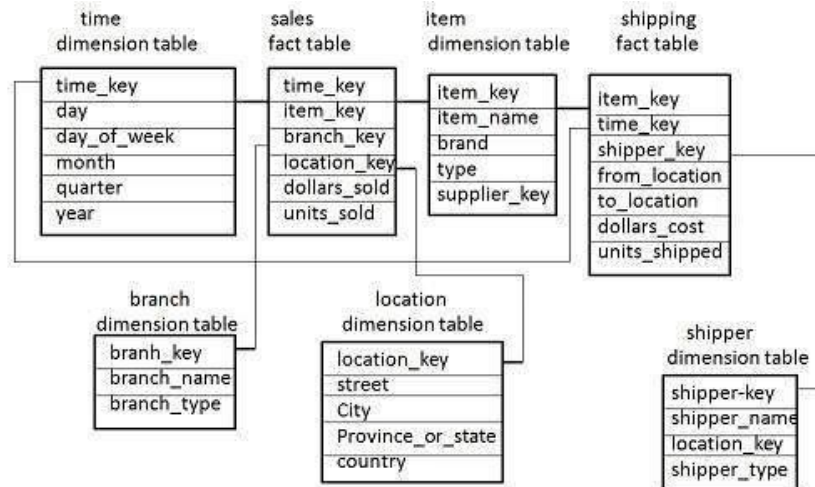
- Some dimension tables in the Snowflake schema are normalized.
- The normalization splits up the data into additional tables.
- Unlike Star schema, the dimensions table in a snowflake schema are normalized. For example, the item dimension table in star schema is normalized and split into two dimension tables, namely item and supplier table.



- Now the item dimension table contains the attributes item_key, item_name, type, brand, and supplier-key.
- The supplier key is linked to the supplier dimension table. The supplier dimension table contains the attributes supplier_key and supplier_type.

3. Fact Constellation Schema

- A fact constellation has multiple fact tables. It is also known as galaxy schema.
- The following diagram shows two fact tables, namely sales and shipping.



- The sales fact table is the same as that in the star schema.
- The shipping fact table has the five dimensions, namely item_key, time_key, shipper_key, from_location, to_location.
- The shipping fact table also contains two measures, namely dollars sold and units sold.
- It is also possible to share dimension tables between fact tables. For example, time, item, and location dimension tables are shared between the sales and shipping fact table.

OLAP Operations in the Multidimensional Data Model

In the multidimensional model, the records are organized into various dimensions, and each dimension includes multiple levels of abstraction described by concept hierarchies. This organization supports users with the flexibility to view data from various perspectives. A number of OLAP data cube operations exist to demonstrate these different views, allowing interactive queries and search of the record at hand. Hence, OLAP supports a user-friendly environment for interactive data analysis.

Basic operations of OLAP

Four types of analytical OLAP operations are:

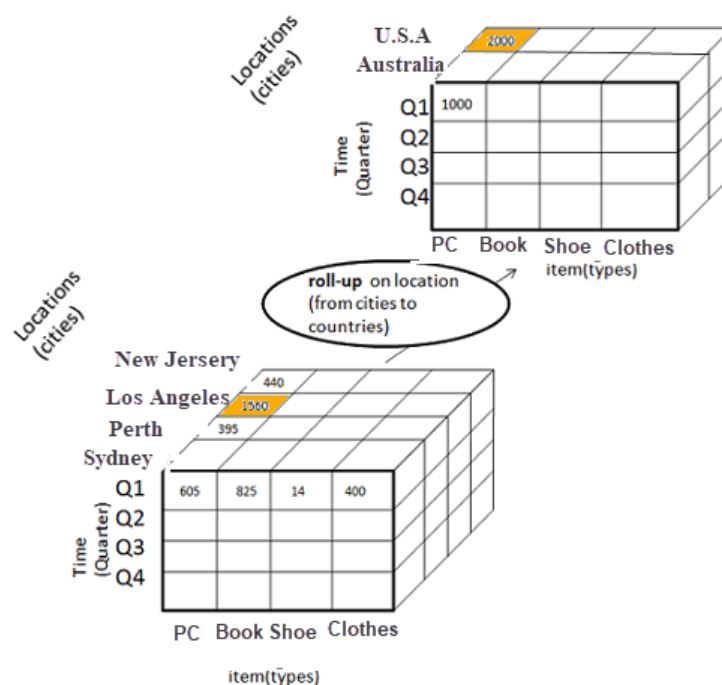
1. Roll-up
2. Drill-down
3. Slice and dice
4. Pivot (rotate)

1) Roll-up

Roll-up is also known as “consolidation” or “aggregation.” The Roll-up operation can be performed in 2 ways

1. Reducing dimensions
2. Climbing up concept hierarchy. Concept hierarchy is a system of grouping things based on their order or level.

Consider the following diagram



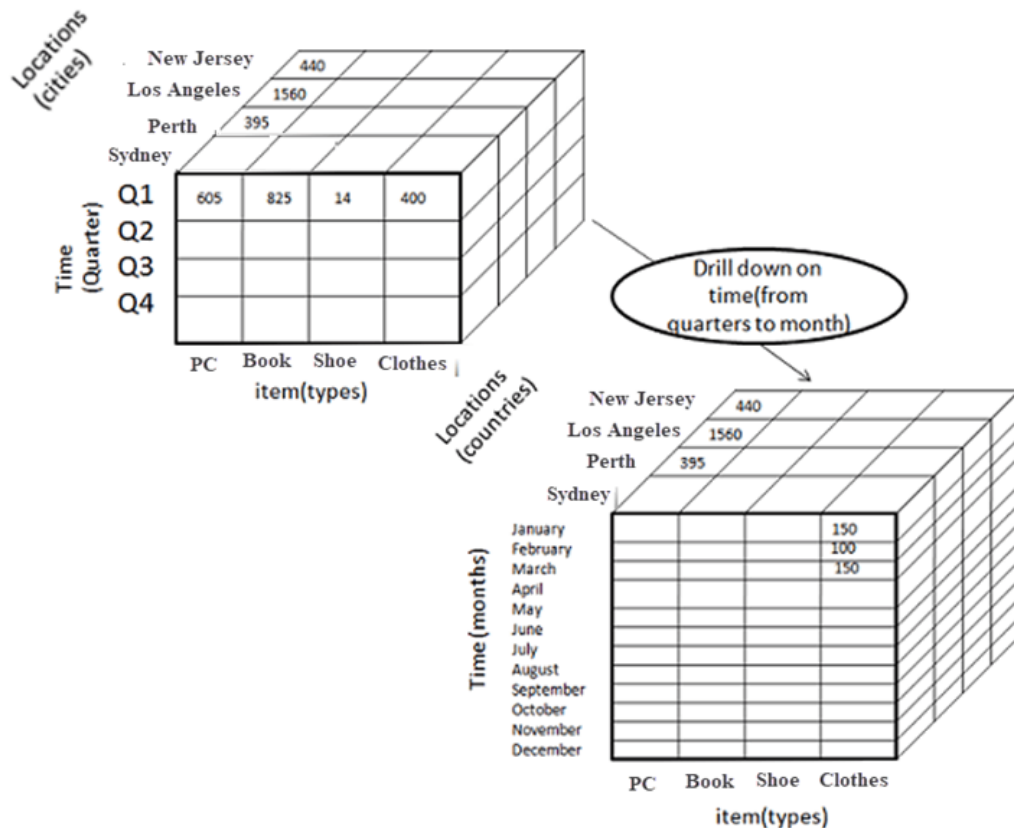
Roll-up operation in OLAP

- In this example, cities New Jersey and Los Angeles are rolled up into country USA
- The sales figures of New Jersey and Los Angeles are 440 and 1560 respectively. They become 2000 after roll-up
- In this aggregation process, data location hierarchy moves up from city to country.
- In the roll-up process at least one or more dimensions need to be removed. In this example, the Cities dimension is removed.

2) Drill-down

In drill-down data is fragmented into smaller parts. It is the opposite of the rollup process. It can be done via

- Moving down the concept hierarchy
- Increasing a dimension



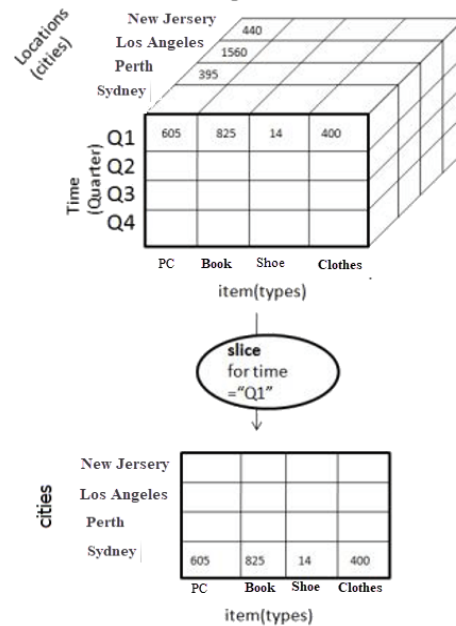
Drill-down operation in OLAP

Consider the diagram above

- Quarter Q1 is drilled down to months January, February, and March. Corresponding sales are also registered.
- In this example, dimension months are added.

3) Slice

Here, one dimension is selected, and a new sub-cube is created. Following diagram explain how slice operation performed:

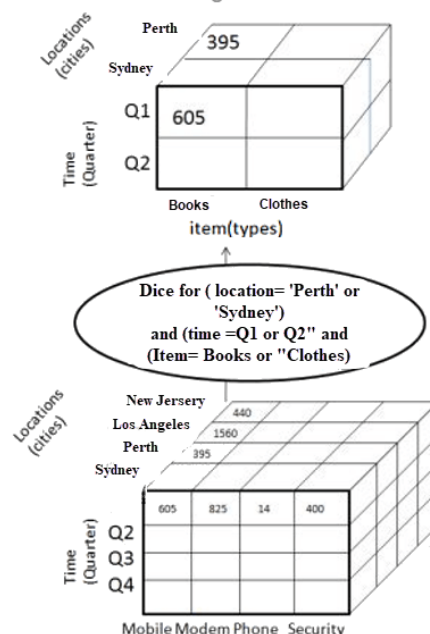


Slice operation in OLAP

- Dimension Time is Sliced with Q1 as the filter.
- A new cube is created altogether.

Dice:

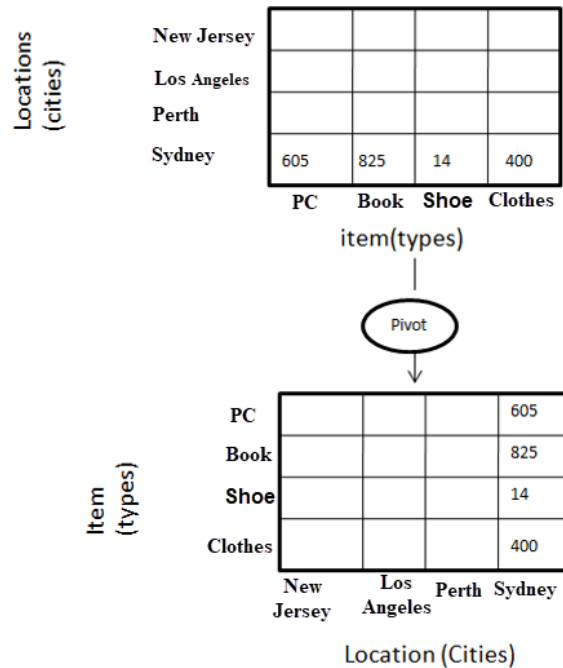
This operation is similar to a slice. The difference in dice is you select 2 or more dimensions that result in the creation of a sub-cube.



4) Pivot

In Pivot, you rotate the data axes to provide a substitute presentation of data.

In the following example, the pivot is based on item types.



Output :

1 : Create a text file with the first row filled with attributes and remaining rows filled with desired values.

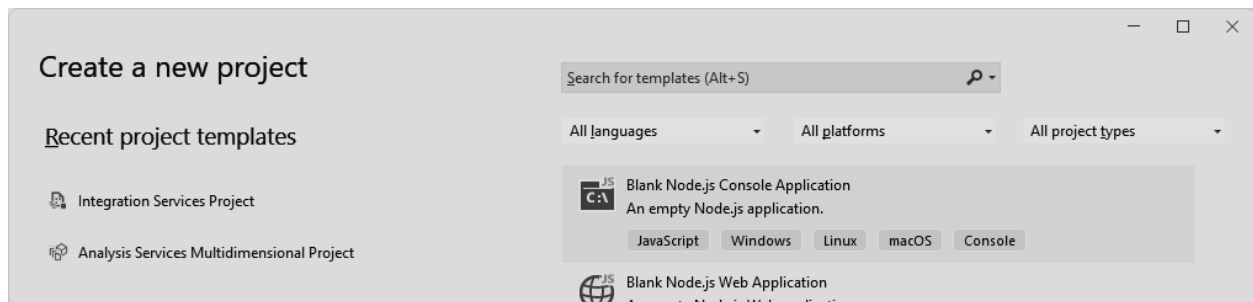
```
complaint
File Edit View

Complaint ID,Date Received,Date Sent to Company,Company Response to Consumer,Timely Response,Consumer Disputed
0,2013-07-29,2013-08-07,Closed with non-monetary relief,Yes,No
1,2013-07-29,2013-08-01,Closed with explanation,Yes,Yes
2,2013-07-29,2013-08-01,Closed with non-monetary relief,Yes,No
3,2013-07-29,2013-07-30,Closed with explanation,Yes,No
4,2013-07-29,2013-08-01,Closed with explanation,Yes,No
5,2013-07-29,2013-07-30,Closed with explanation,Yes,No
6,2013-07-29,2013-12-04,Closed with explanation,Yes,No
7,2013-07-29,2013-07-30,Closed with explanation,Yes,No
8,2013-07-29,2013-09-17,Closed,Yes,No
9,2013-07-31,2013-07-30,Closed with explanation,Yes,No
10,2013-07-30,2013-09-17,Closed with explanation,Yes,No
11,2013-07-30,2013-07-29,Closed with explanation,Yes,No
12,2013-07-31,2013-07-30,Closed with explanation,Yes,No
```

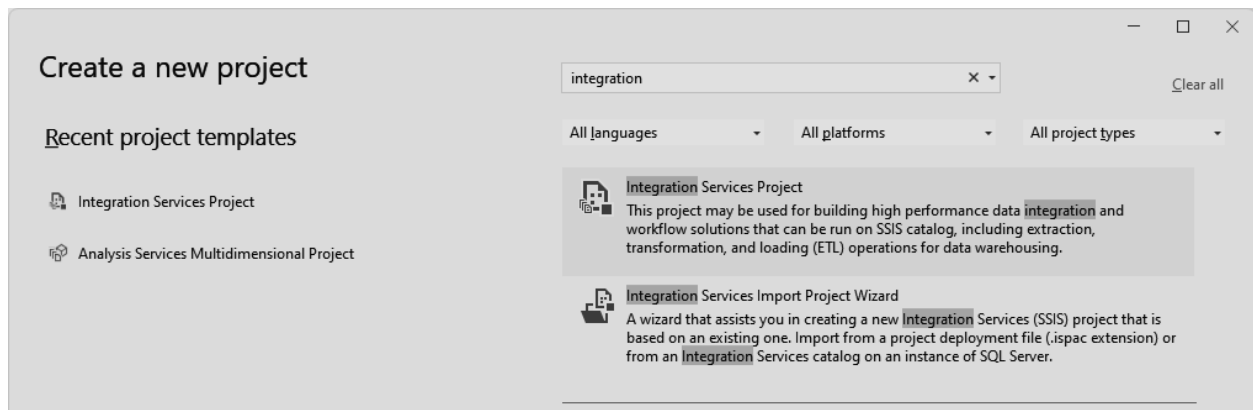
2 : Create a table in your database with the same attributes which were filled in a text file in previous steps with appropriate data type.

```
SQLQuery2.sql - INF...-12.test37 (sa (56))* X
create table complaint(
  Complaint_ID INT primary key,
  Date_Received date,
  Date_Sent_to_Company date,
  Company_Response_to_Consumer varchar(250),
  Timely_Response varchar(3),
  Consumer_Disputed varchar(3)
);
```

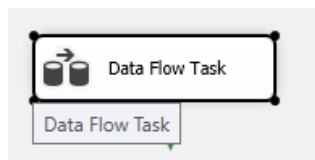
3 : Open Visual Studio 2022.



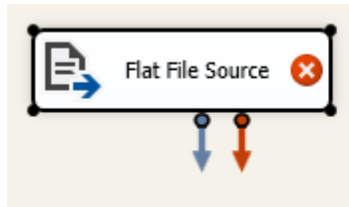
4 : Search for an integration service Project.



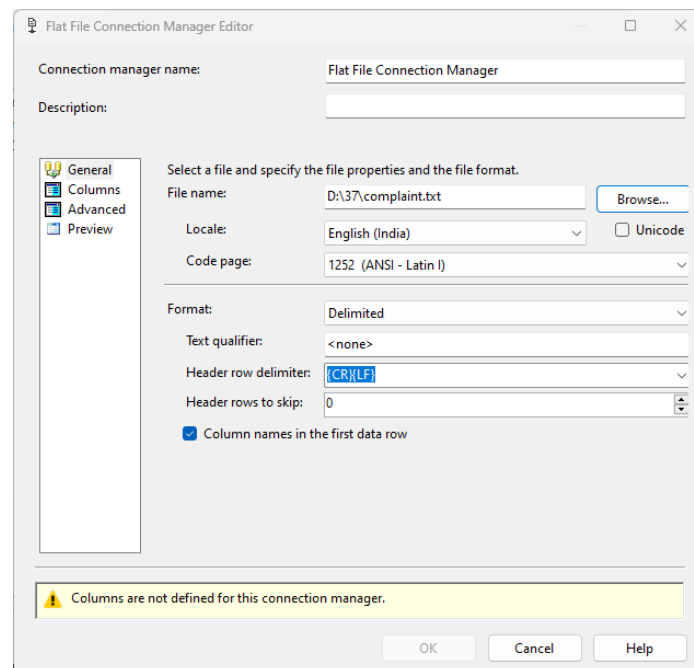
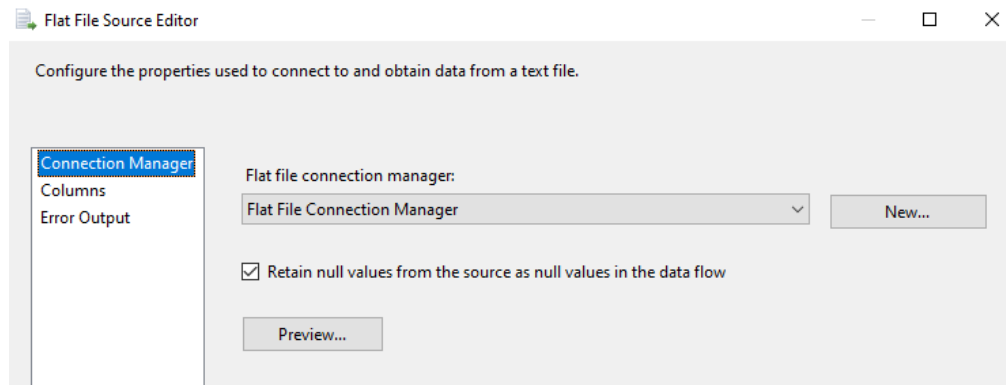
5 : Drag Data Flow Task to workspace.



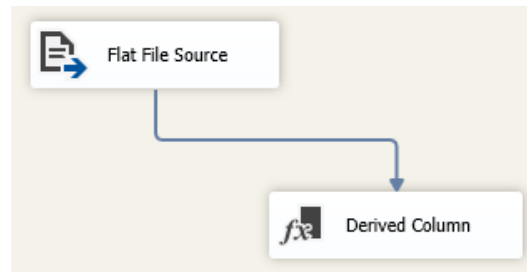
6 : Click on Data Flow to go into its workspace and then drag Flat File Source to its Workspace.



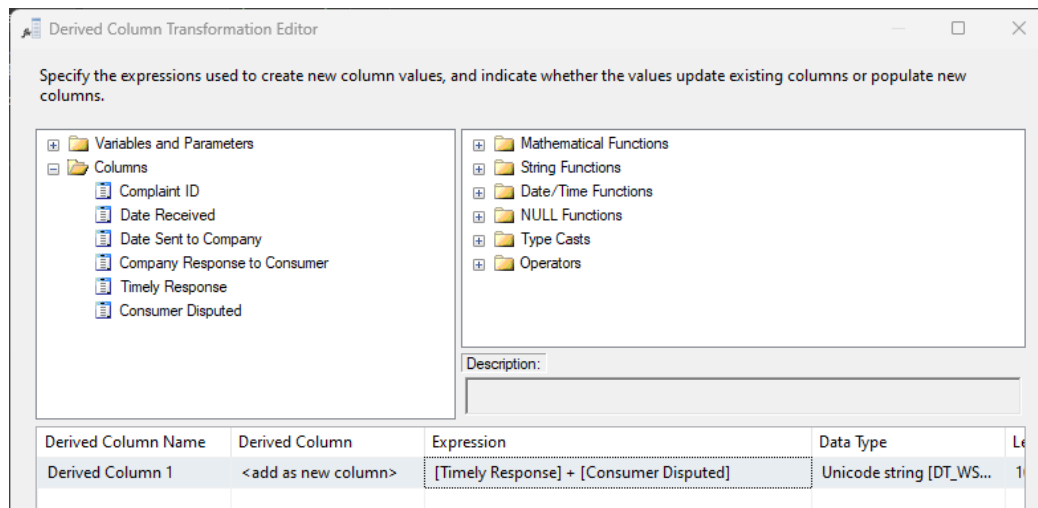
7 : After Clicking on Flat File Source click on New to add the complaint.csv which was created in the first step.



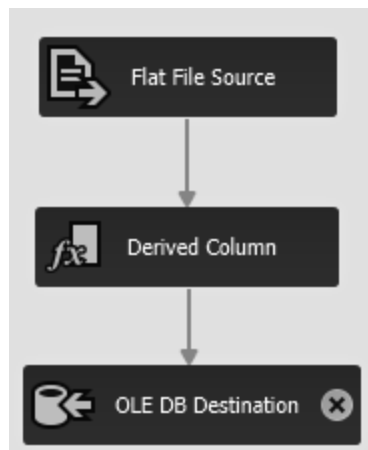
8 : Drag Derived Column to workspace and then connect Flat File Source to Derived Column.



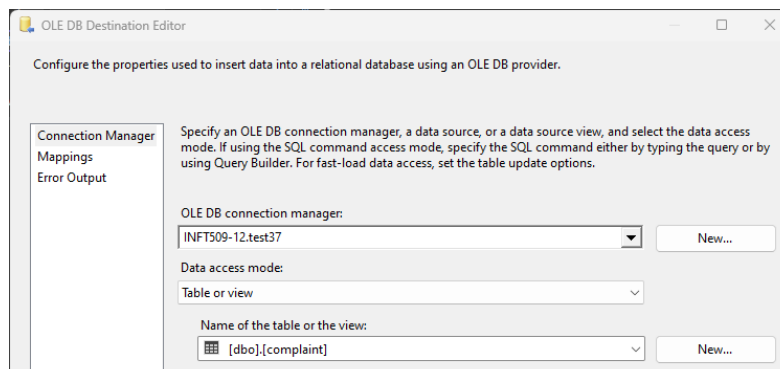
9 : After Clicking on the Derived Column we see the following dialog box, add all the column names to be mapped and transformed with needful transformation in the expression column.



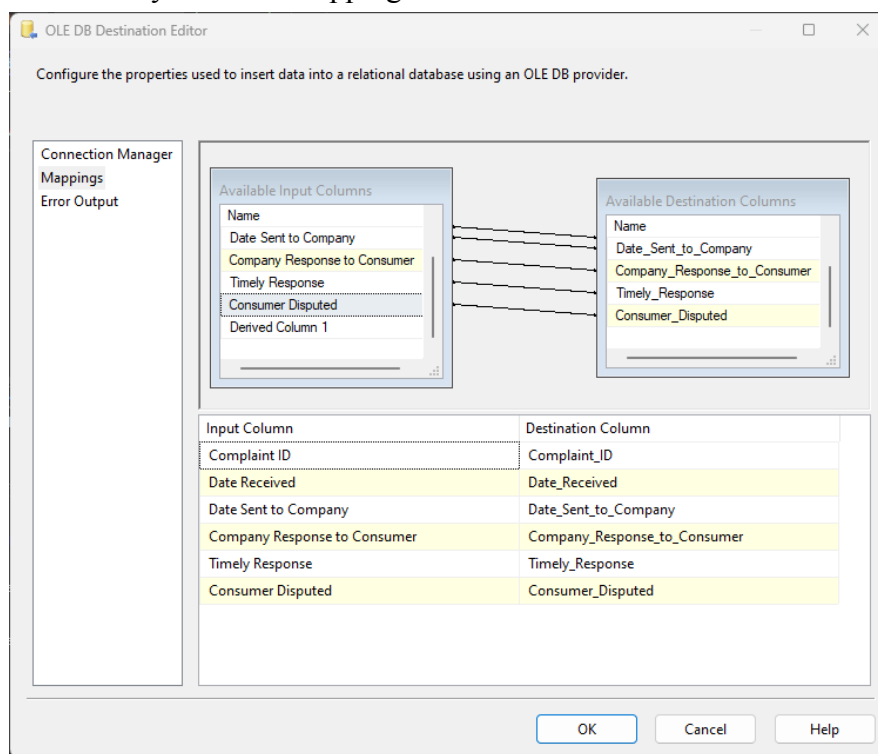
10 : After Dragging OLE DB Destination connects it with Derived Column we have to define a database which contains the table to transform.



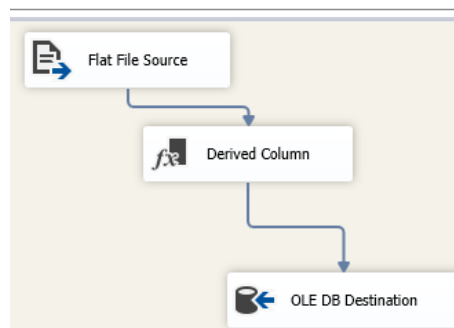
11 : Select the Table to be altered.



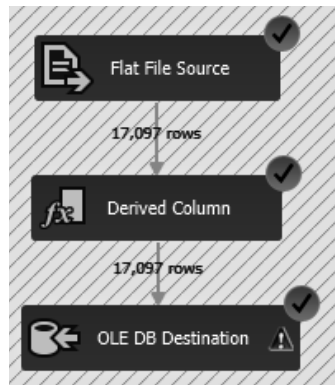
12 : Check if there are any errors in mappings.



13 : Check if there are any errors, if not then click on the run icon.



14 : After execution Check if there are any errors and warnings and check if the state is 'ready'.



15 : In dbo.test37 Table we can see new entries are mapped as per Derived Columns expressions.

SQLQuery4.sql - INF...-12.test37 (sa (54))

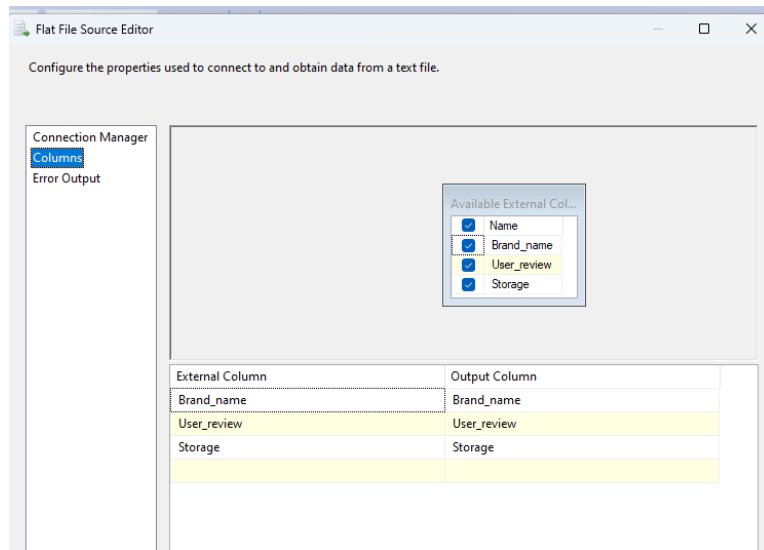
```
/****** Script for SelectTopNRows command from SSMS *****/  
SELECT TOP 1000 [Complaint_ID]  
      ,[Date_Received]  
      ,[Date_Sent_to_Company]  
      ,[Company_Response_to_Consumer]  
      ,[Timely_Response]  
      ,[Consumer_Disputed]  
FROM [test37].[dbo].[complaint]
```

100 %

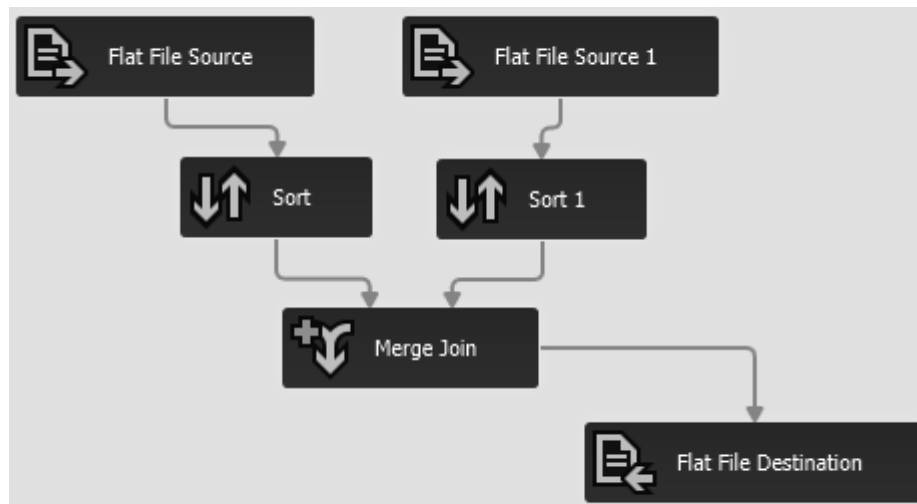
Results Messages

	Complaint_ID	Date_Received	Date_Sent_to_Company	Company_Response_to_Consumer	Timely_Response	Consumer_Disputed
1	0	2013-07-29	2013-08-07	Closed with non-monetary relief	Yes	No
2	1	2013-07-29	2013-08-01	Closed with explanation	Yes	Yes
3	2	2013-07-29	2013-08-01	Closed with non-monetary relief	Yes	No
4	3	2013-07-29	2013-07-30	Closed with explanation	Yes	No
5	4	2013-07-29	2013-08-01	Closed with explanation	Yes	No
6	5	2013-07-29	2013-07-30	Closed with explanation	Yes	No
7	6	2013-07-29	2013-12-04	Closed with explanation	Yes	No
8	7	2013-07-29	2013-07-30	Closed with explanation	Yes	No
9	8	2013-07-29	2013-09-17	Closed	Yes	No
10	9	2013-07-31	2013-07-30	Closed with explanation	Yes	No
11	10	2013-07-30	2013-09-17	Closed with explanation	Yes	No
12	11	2013-07-30	2013-07-29	Closed with explanation	Yes	No
13	12	2013-07-31	2013-07-30	Closed with explanation	Yes	No

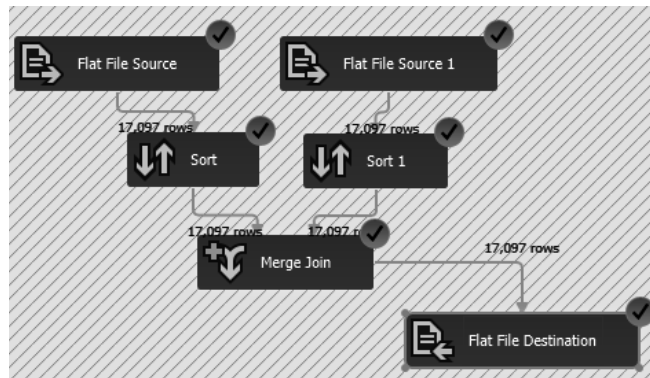
16 : We are then taking two flat file sources, adding the same txt file to it and checking for correct mappings.



17 : Create the following Structure for Performing **Merge Join**.



18 : After performing necessary operations we have to look for the errors before debugging and then click on 'start'.



19 : So we get Merge Join Output of above Operation

```
merge_output
File Edit View

0,2013-07-29,2013-08-07,Closed with non-monetary relief,Yes,No,0,Cont'd attempts collect debt not owed,Debt is not mine,Web
1,2013-07-29,2013-08-01,Closed with explanation,Yes,Yes,1,Cont'd attempts collect debt not owed,Debt was paid,Referral
10,2013-07-30,2013-09-17,Closed with explanation,Yes,No,10,Cont'd attempts collect debt not owed,Debt is not mine,Web
100,2013-07-25,2013-07-24,Closed with explanation,Yes,Yes,100,Improper contact or sharing of info,Talked to a third party about my debt,Web
1000,2013-09-11,2013-09-12,Closed with explanation,Yes,No,1000,Repaying your loan,Repaying your loan,Web
10000,2014-08-06,2014-08-13,Closed with explanation,Yes,No,10000,Disclosure verification of debt,Not given enough info to verify debt,Web
10001,2014-08-26,2014-08-26,Closed with explanation,Yes,No,10001,Cont'd attempts collect debt not owed,Debt is not mine,Web
10002,2014-08-20,2014-08-25,Closed with non-monetary relief,Yes,No,10002,Communication tactics,Frequent or repeated calls,Web
10003,2014-08-20,2014-08-20,Closed with explanation,Yes,No,10003,False statements or representation,Indicated committed crime not paying,Web
10004,2014-08-20,2014-10-06,Closed with explanation,Yes,No,10004,Disclosure verification of debt,Not given enough info to verify debt,Web
10005,2014-08-15,2014-08-19,Closed with explanation,Yes,No,10005,Dealing with my lender or servicer,Keep getting calls about my loan,Web
10006,2014-08-20,2014-08-20,Closed with explanation,Yes,No,10006,Cont'd attempts collect debt not owed,Debt is not mine,Web
10007,2014-08-15,2014-08-19,Closed with explanation,Yes,No,10007,Cont'd attempts collect debt not owed,Debt was paid,Web
10008,2014-08-15,2014-09-12,Closed with explanation,Yes,No,10008,Cont'd attempts collect debt not owed,Debt is not mine,Web
```

20 : We are then taking a dataset file and checking for the columns being used for sorting.

Flat File Source Editor

Configure the properties used to connect to and obtain data from a text file.

Connection Manager
Columns
Error Output

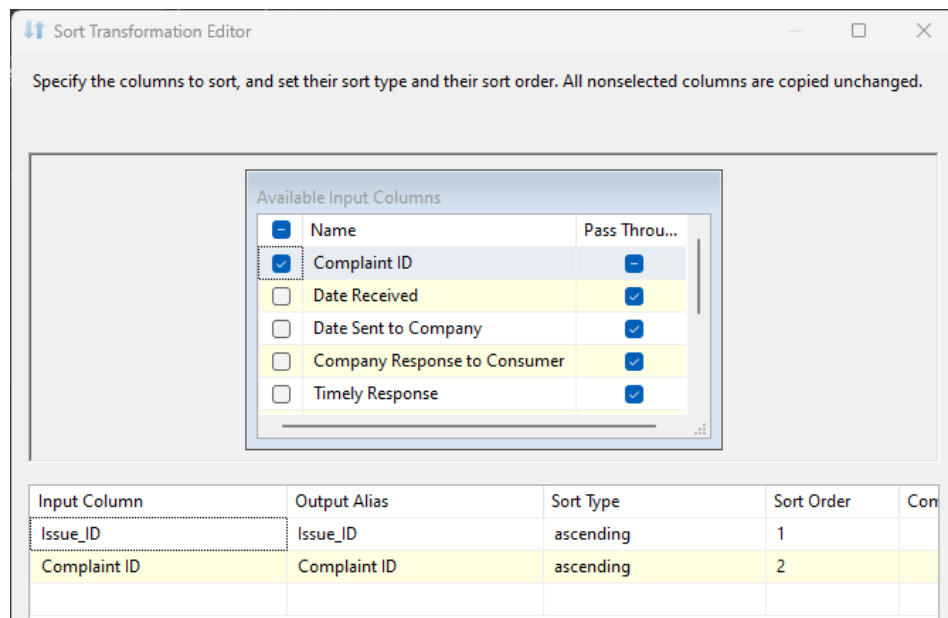
Available External Columns

- ☒ Name
- ☒ Complaint ID
- ☒ Date Received
- ☒ Date Sent to Company
- ☒ Company Response to Consumer
- ☒ Timely Response
- ☒ Consumer Disputed
- ☒ Issue_ID





External Column	Output Column
Complaint ID	Complaint ID
Date Received	Date Received
Date Sent to Company	Date Sent to Company
Company Response to Consumer	Company Response to Consumer
Timely Response	Timely Response
Consumer Disputed	Consumer Disputed
Issue_ID	Issue_ID
Issue	Issue
Sub Issue	Sub Issue

OK Cancel Help

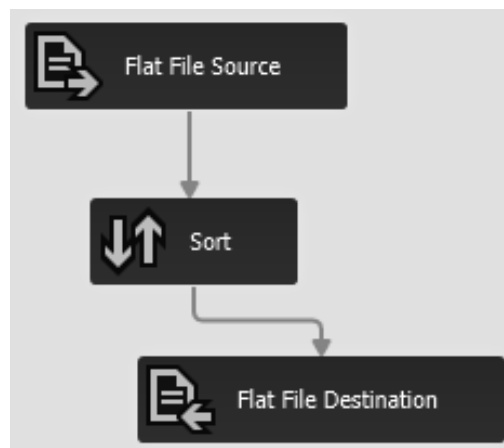
21 : In Sort we will be using User_Review as a parameter to rearrange the file contents



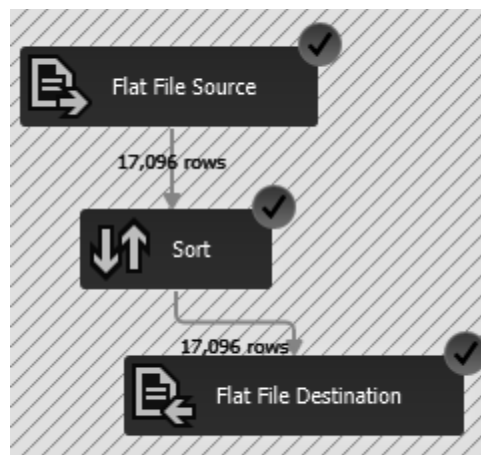
22 : We are choosing where the output will be stored i.e sortedOutput in Flat File Destination.

	P9-ConsumerComplaints	03-03-2023 15:25	XLS Worksheet	14,710 KB
	product	08-03-2023 15:36	Text Document	708 KB
	sort_output	08-03-2023 16:30	Text Document	2,199 KB
	test	08-03-2023 13:54	Jupyter Source File	36 KB

23 : Below is the necessary structure for **Sorting** Operation.



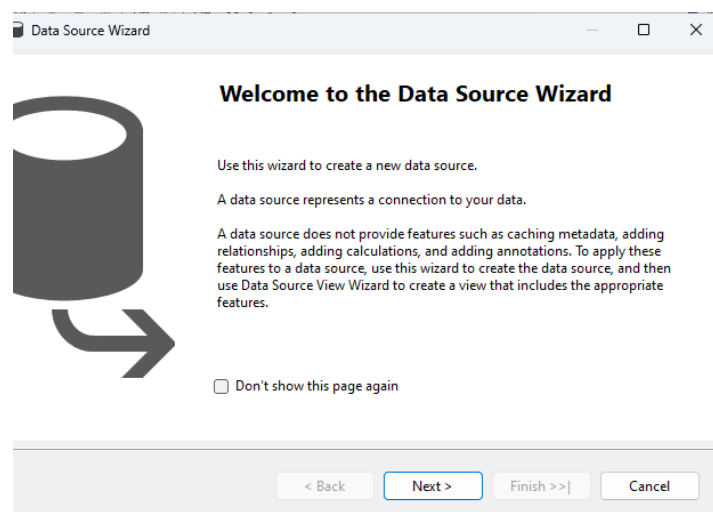
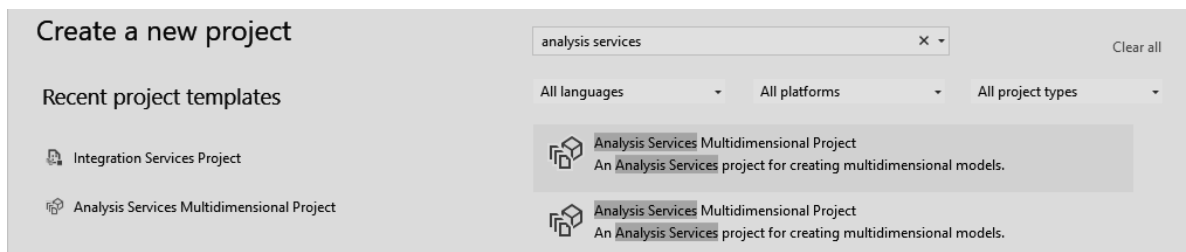
24 : We will get the image below after clicking on the start of the error free setup.



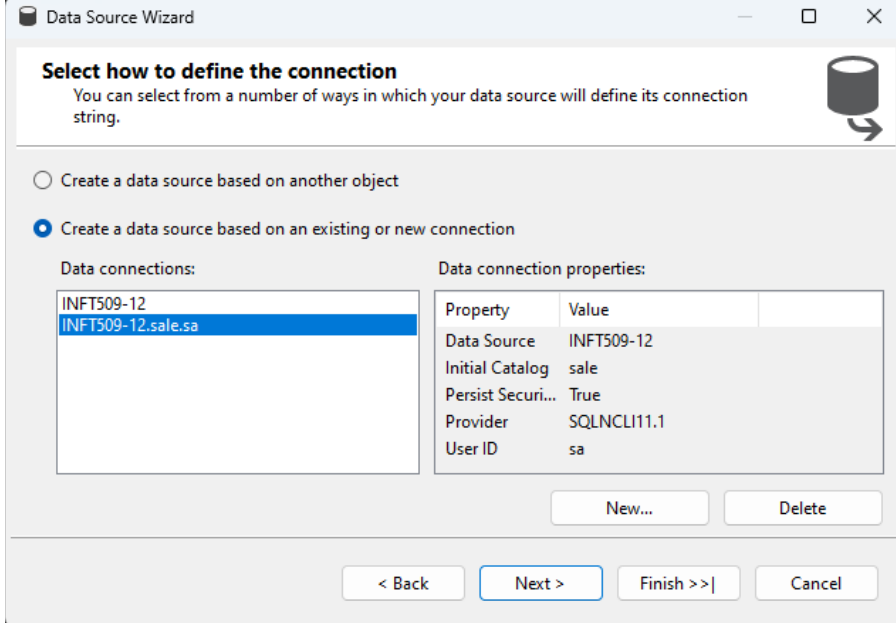
25 : This is the output of Sorted File.

The screenshot shows a text editor window titled 'sort_output'. The text inside is a single line of data, sorted alphabetically by the first column (Issue_ID). The data is as follows:
1,Cont'd attempts collect debt not owed,Debt was paid,Referral,1,2013-07-29,2013-08-01,Closed with explanation,Yes,Yes
10,Cont'd attempts collect debt not owed,Debt is not mine,Web,10,2013-07-30,2013-09-17,Closed with explanation,Yes,No
100,Improper contact or sharing of info,Talked to a third party about my debt,Web,100,2013-07-25,2013-07-24,Closed with explanation,Yes,Yes
1000,Repaying your loan,Repaying your loan,Web,1000,2013-09-11,2013-09-12,Closed with explanation,Yes,No
10000,Disclosure verification of debt,Not given enough info to verify debt,Web,10000,2014-08-06,2014-08-13,Closed with explanation,Yes,No
10001,Cont'd attempts collect debt not owed,Debt is not mine,Web,10001,2014-08-26,2014-08-26,Closed with explanation,Yes,No
10002,Communication tactics,Frequent or repeated calls,Web,10002,2014-08-20,2014-08-25,Closed with non-monetary relief,Yes,No
10003,Esco statements on representation,Indicated committed crime not paying Web,10003,2014-08-20,2014-08-20,Closed with explanation,Yes,No

26 : After creating the Analysis Project on Visual Studio we click on the new data source wizard.



27 : Then we select the required Database and **select the service account**.



Select how to define the connection
You can select from a number of ways in which your data source will define its connection string.

☐ Create a data source based on another object

☒ Create a data source based on an existing or new connection

Data connections:

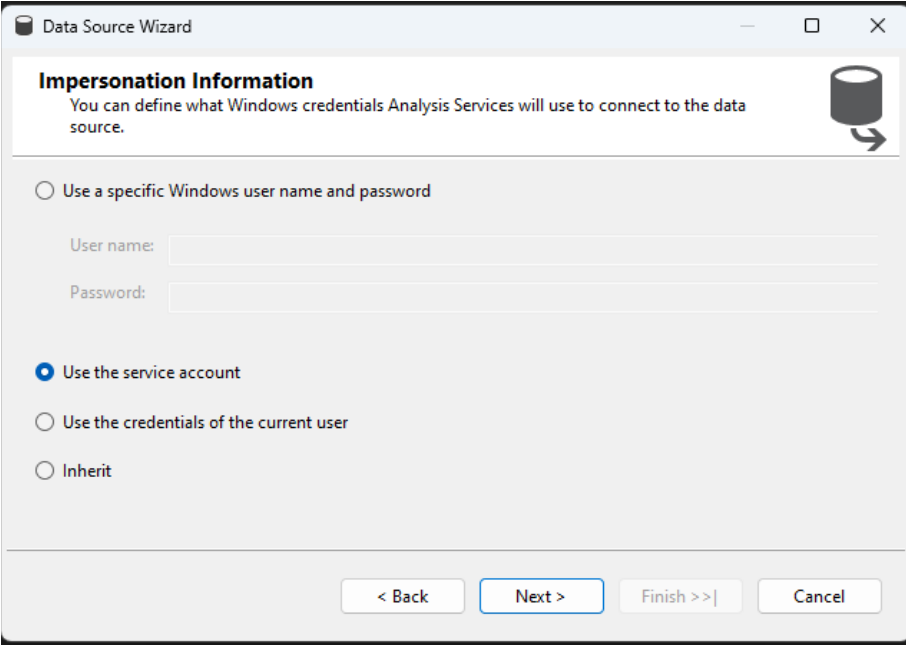
INFT509-12
INFT509-12.sale.sa

Data connection properties:

Property	Value
Data Source	INFT509-12
Initial Catalog	sale
Persist Securi...	True
Provider	SQLNCLI11.1
User ID	sa

New... Delete

< Back Next > Finish >>| Cancel



Impersonation Information
You can define what Windows credentials Analysis Services will use to connect to the data source.

☐ Use a specific Windows user name and password

User name:

Password:

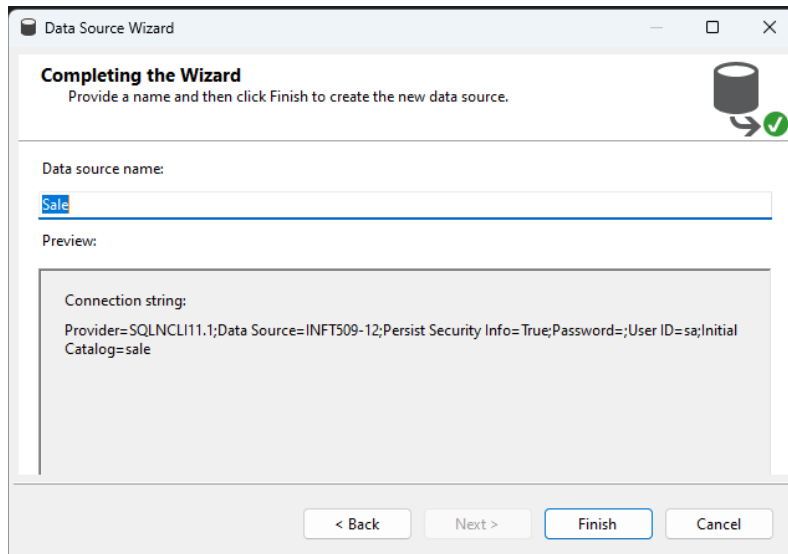
☒ Use the service account

☐ Use the credentials of the current user

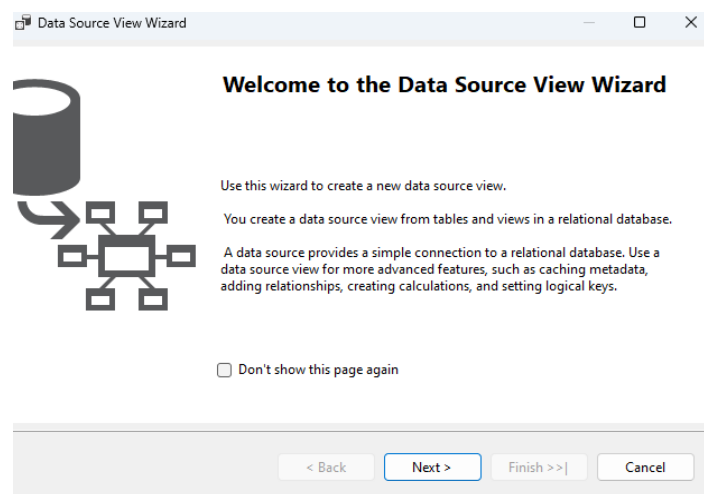
☐ Inherit

< Back Next > Finish >>| Cancel

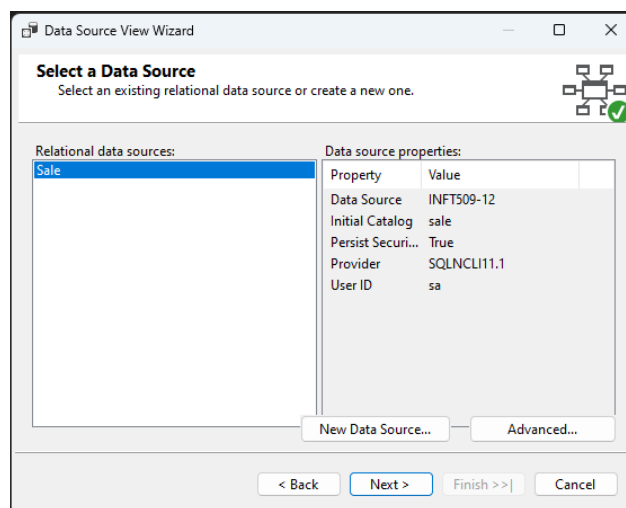
28 : Then we give the data source name.



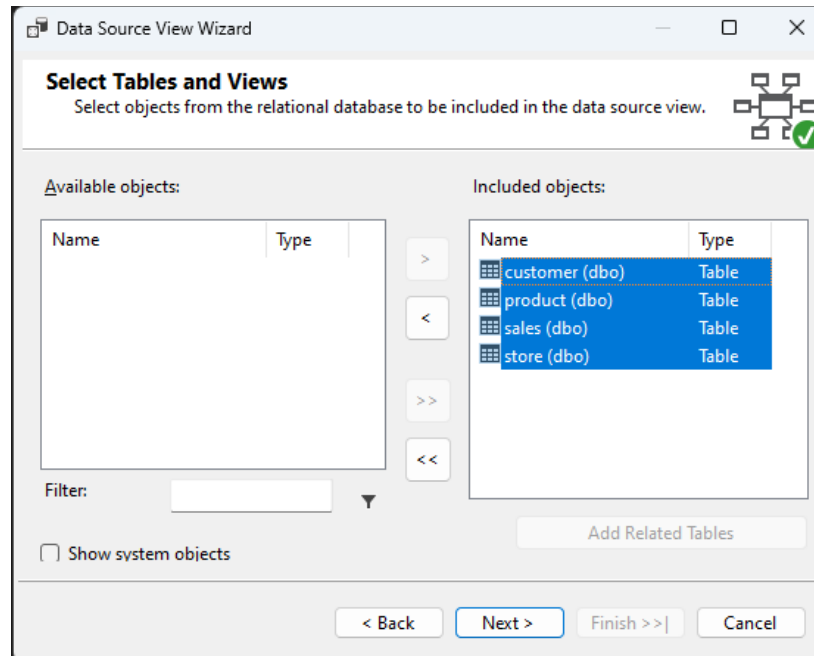
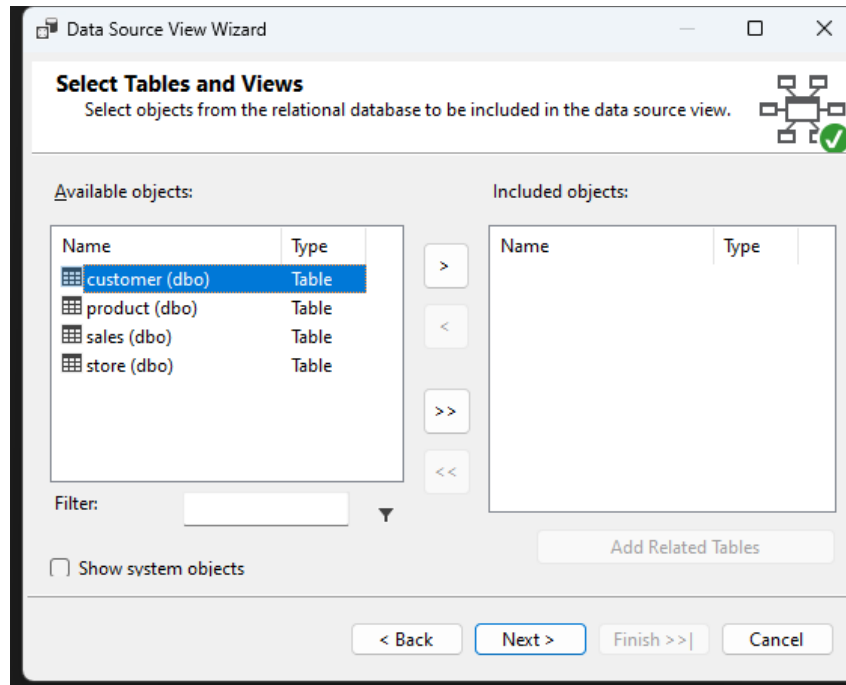
29 : Then we click on new data source views.



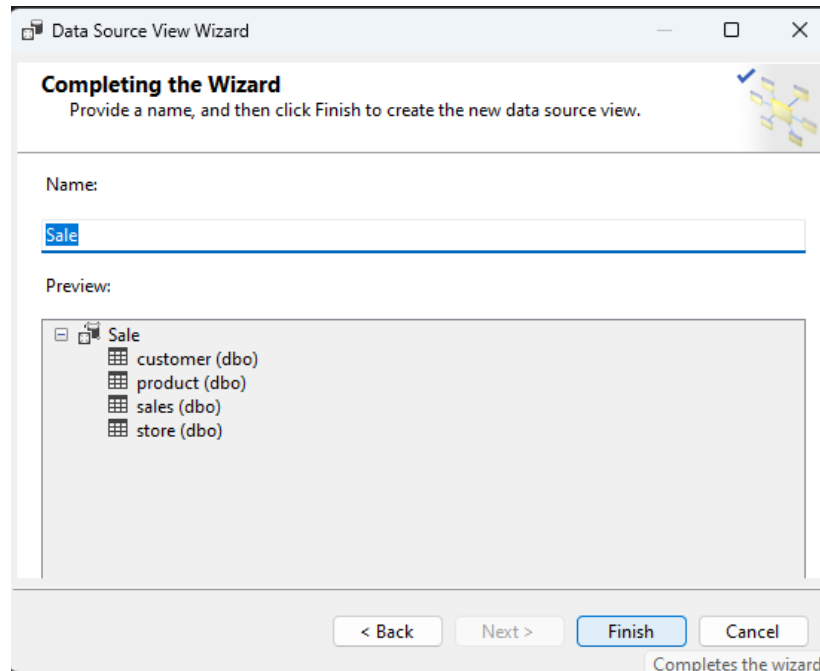
30 : In that we choose a data source.



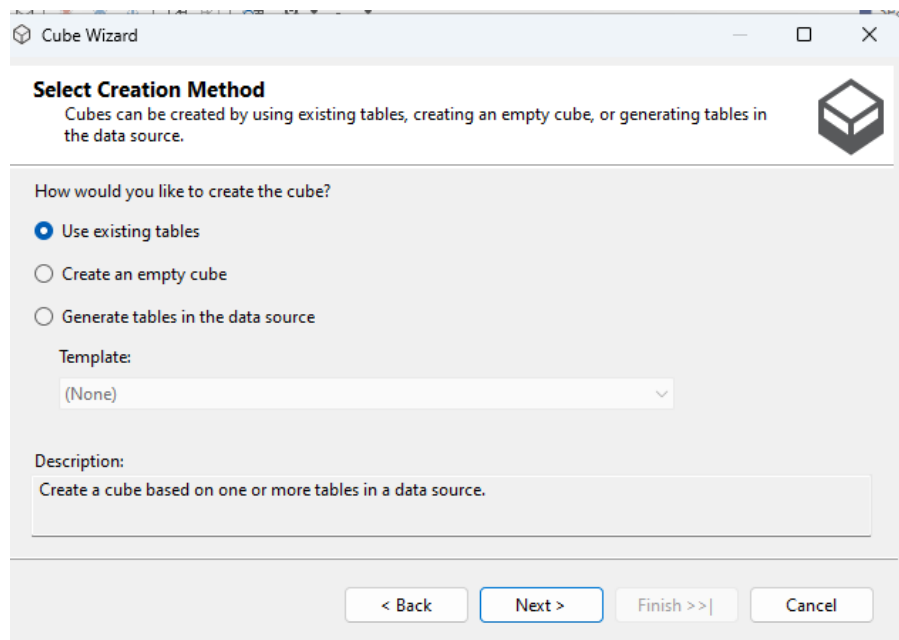
31 : In this we add the tables we will be using for creating star schema.



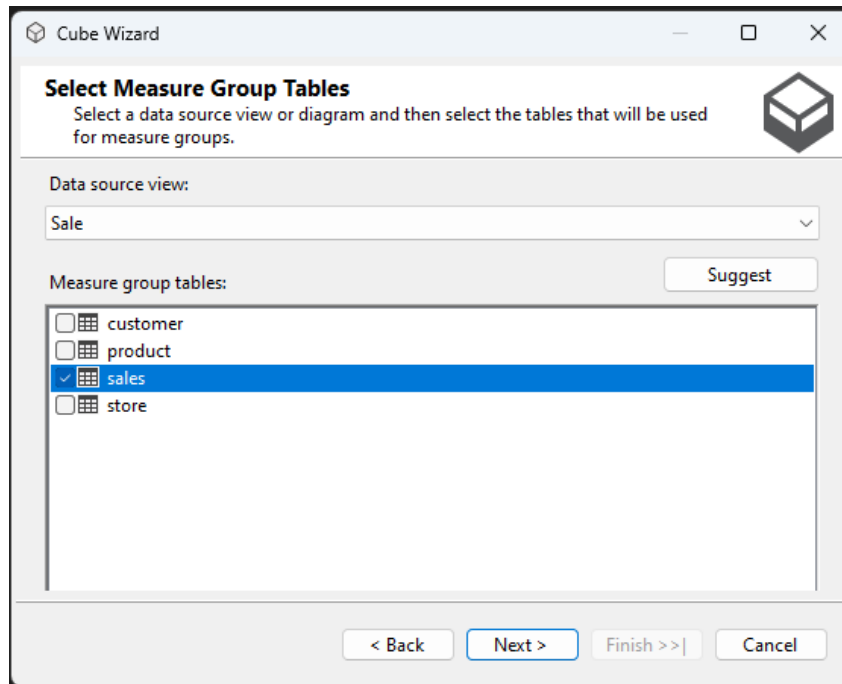
32 : We give names to our data source view.



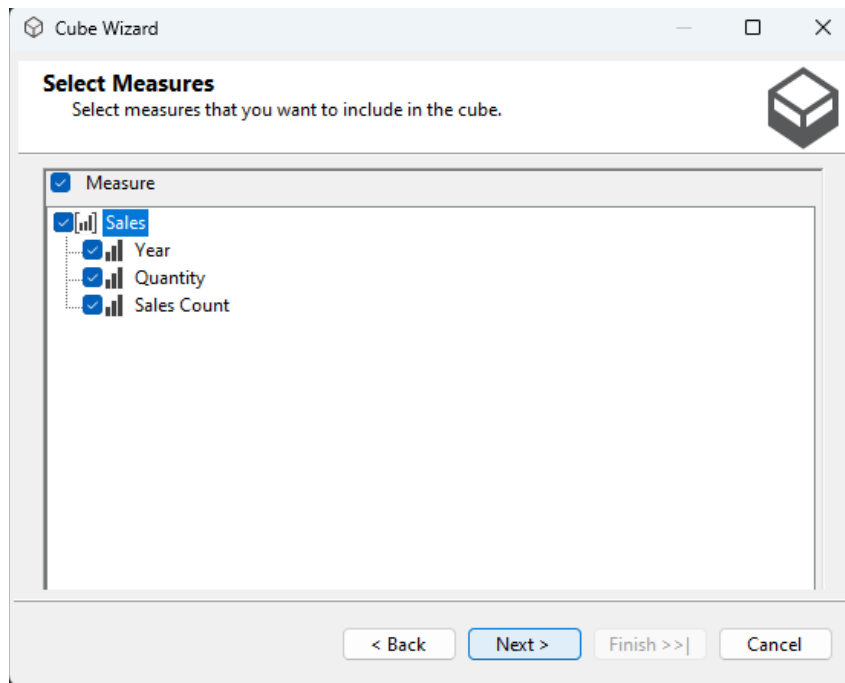
33 : Now click on New Cube.



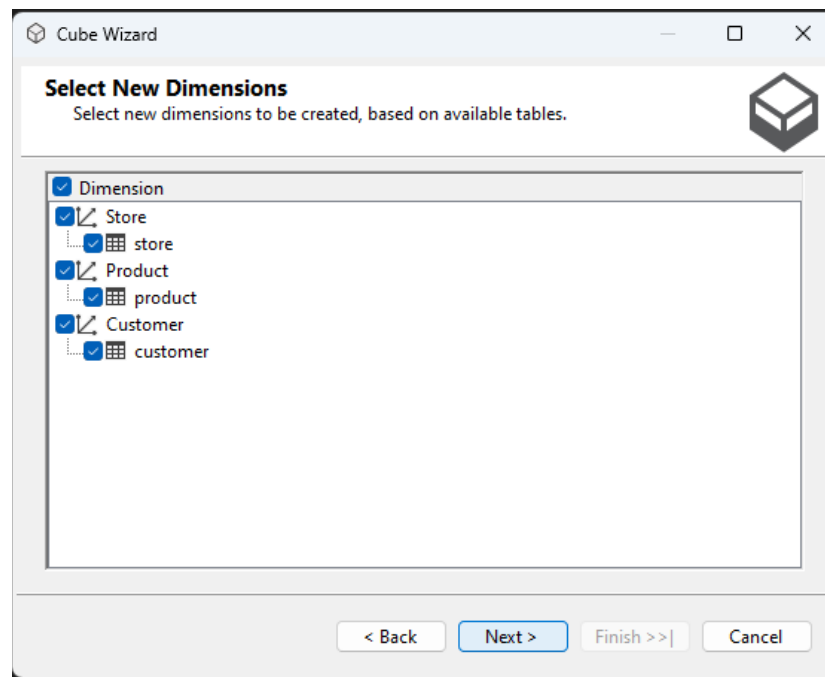
34 : Check on All the tables.



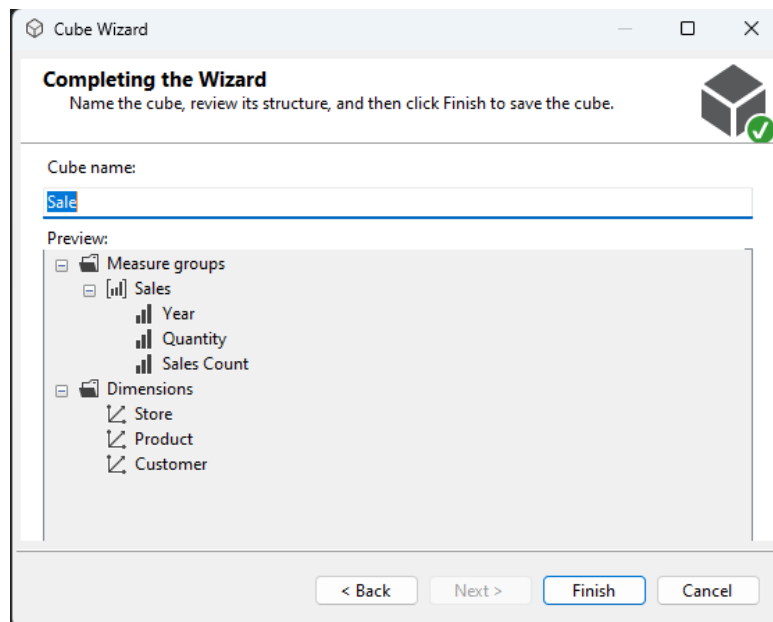
35 : Click on All the measures.



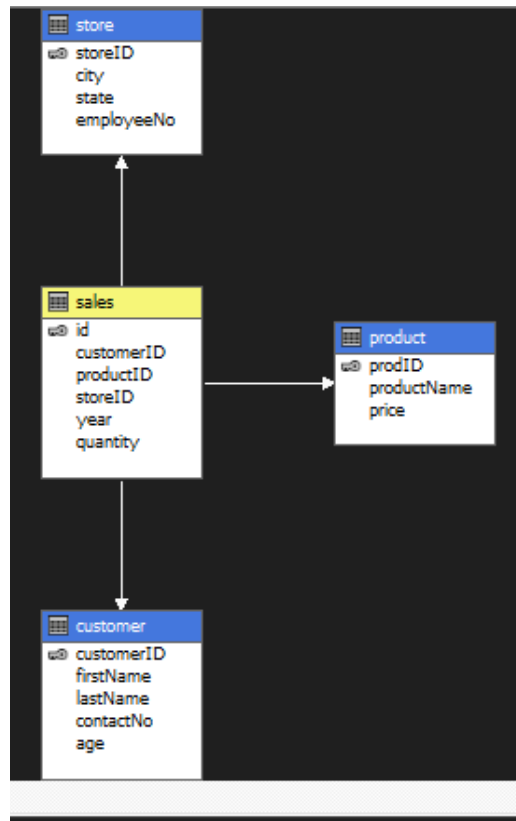
36 : Click on All the measures.



37 : Give name to the cube.



38: We can see the Data Source View.



39: We now create a new process and click on Run.

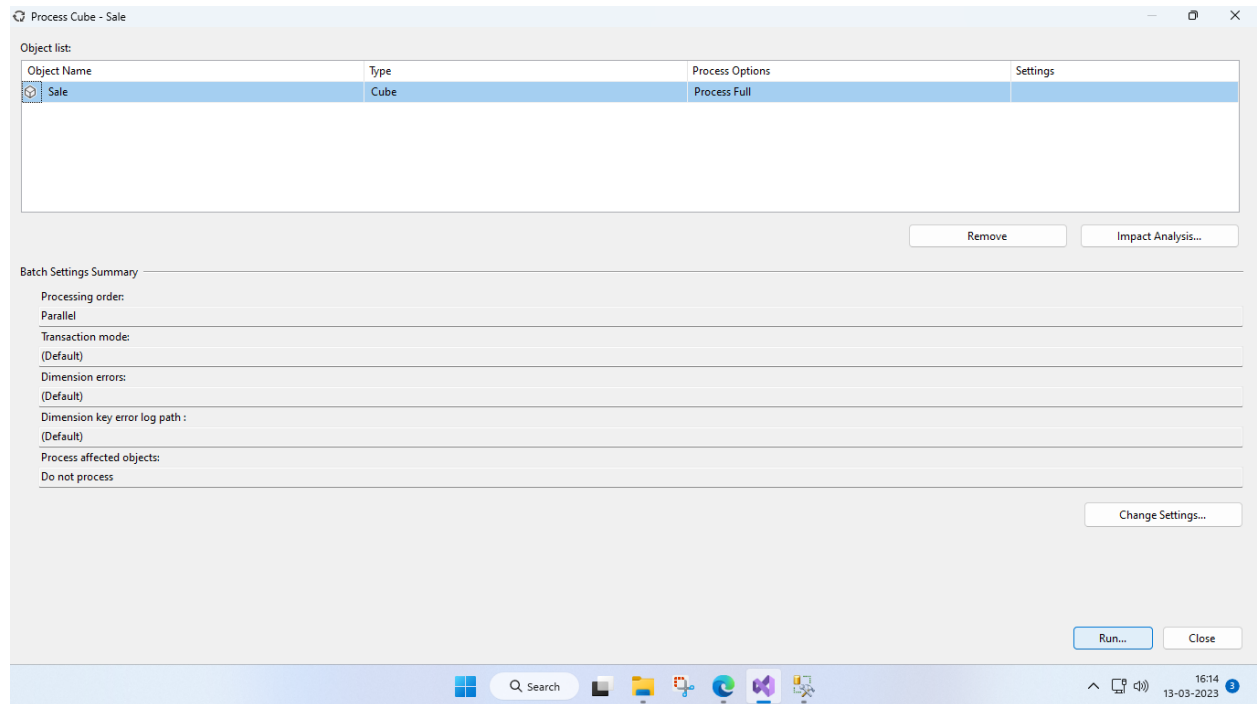
The screenshot shows the 'Process Cube - Sale' window. The 'Object list' table contains the following data:

Object Name	Type	Process Options	Settings
Sale	Cube	Process Full	

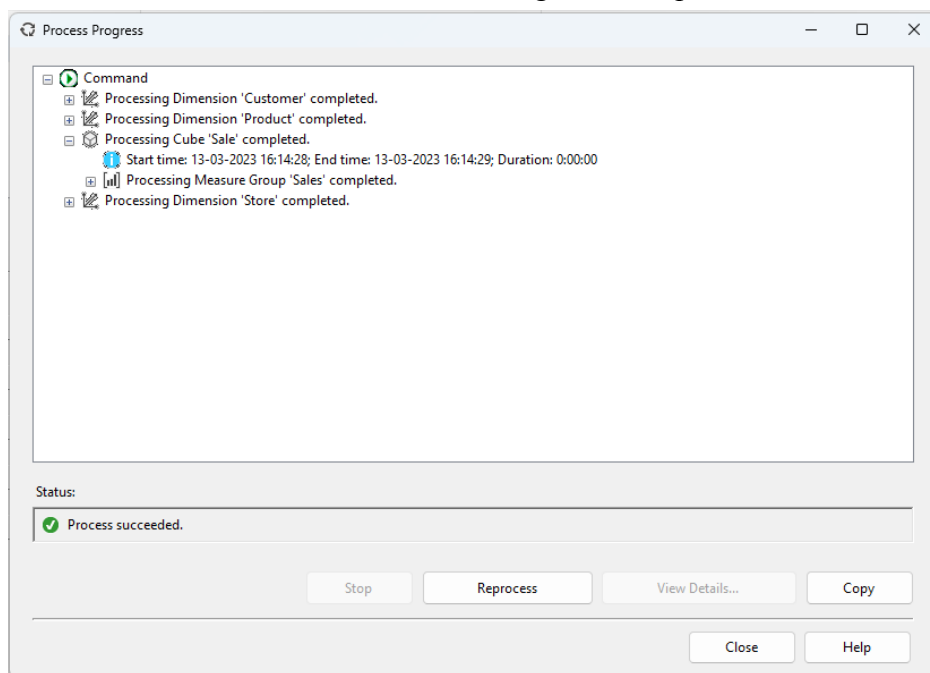
Below the table are buttons for 'Remove' and 'Impact Analysis...'. The 'Batch Settings Summary' section displays the following configuration:

- Processing order: Parallel
- Transaction mode: (Default)
- Dimension errors: (Default)
- Dimension key error log path: (Default)
- Process affected objects: Do not process

A 'Change Settings...' button is located at the bottom right of the settings section. In the background, a 'Deployment Progress' window shows 'Deployment Completed Successfully' with a green checkmark.



40: We can see that it was successful and now we drag different parameters to see the analysis.



The screenshot shows a BI tool interface with a top menu bar including 'Cube Structure', 'Dimension Usage', 'Calculations', 'KPIs', 'Actions', 'Partitions', 'Aggregations', 'Perspectives', 'Translations', and 'Browser'. Below the menu is a toolbar with icons for 'Edit as Text', 'Import...', and 'MDX'. The left pane shows a 'Metadata' tree with 'Sale' as the selected cube. The main area displays a table with the following data:

Customer ID	Quantity	Sales Count
1	81	2
2	55	3
3	60	2
4	50	3

The screenshot shows the same BI tool interface as above, but with a different data table displayed. The table has the following data:

Customer ID	Prod ID	Store ID	Quantity	Sales Count
1	1	2	2	1
1	4	3	79	1
2	3	1	9	1
2	3	4	12	1
2	4	1	34	1
3	1	2	57	1
3	3	2	3	1
4	2	1	2	1
4	2	3	13	1
4	2	4	35	1

Conclusion:

Thus we have successfully designed a Data Warehouse for a given case study and performed ETL and OLAP operations on it.