

# **Data Warehouse and Mining**

## **Experiment No.: 1**

**Write detail problem statement and  
identify source tables and populate  
sample data.**

# Experiment No. 1

1. **Aim:** Write detail problem statement and identify source tables and populate sample data.
2. **Objectives:** To create and understand the requirement of data warehouse in present digital and e business market for producing reports and making critical business decisions
3. **Course Outcomes:** The outcome is to establish and comprehend the need for a data warehouse in the current digital and e-business market, enabling the generation of reports and facilitating critical business decisions.
4. **Hardware / Software Required:** Any editor tool like excel or Power BI to create the tables and populate with relevant values.

**Theory:** The importance of a data warehouse lies in its ability to consolidate and organize large volumes of data from various sources into a single, unified repository. This provides several key benefits:

1. **Informed Decision-Making:** A data warehouse enables businesses to analyze historical data, identify trends, and gain insights that support strategic decision-making.
2. **Data Consistency and Accuracy:** By centralizing data from different sources, a data warehouse ensures that all departments work with the same, consistent information, reducing discrepancies and errors.
3. **Enhanced Business Intelligence:** Data warehouses support advanced analytics, allowing businesses to perform complex queries, generate detailed reports, and uncover deep insights that are not possible with traditional databases.
4. **Improved Performance:** With optimized data structures and indexing, data warehouses allow for faster query processing, enabling timely access to critical information.
5. **Scalability and Flexibility:** As businesses grow, data warehouses can scale to accommodate increasing volumes of data, making them a long-term solution for data management.
6. **Regulatory Compliance:** Data warehouses help businesses maintain data integrity and comply with industry regulations by providing a controlled environment for storing and managing data.

7. **Support for Data-Driven Culture:** By providing easy access to data and insights, data warehouses foster a data-driven culture within organizations, empowering employees to make decisions based on accurate information.

Consider any Business Situation to Design and implement a data warehouse and reporting structure to address this requirement for the client; a garment, Retail or any other e-business franchisees operating approximately in many locations. Write detail problem statement and identify source tables and populate sample data. Or Search and collect the transaction report of any real time business.

8. **Algorithm / Design / Procedure / Flowchart / Analysis:**

The Data Warehouse must provide strategic and tactical decision support to all levels of management. Below are the challenges to address

- The business hierarchies must be in synchronization with all stores and brands.
- A combined corporate view of actual, budgets and forecast scenarios must be supported.
- Reporting requirements are based on each sales, branch, categories and time.

9. **Results/Output Analysis:** Take a print of the table structure and sample values and explain the need for each identified field.

10. **Conclusions:** Discuss whether the experiment's aim was achieved and how the results relate to the theory.

11. **Viva Questions:** A list of potential questions related to the chosen business use case can be expected.

12. **References:**

<https://www.techshashank.com/data-warehousing/shipping-dimensional-modeling>

# **Data Warehouse and Mining**

## **Experiment No.: 2**

**Design dimensional data model - Star  
schema and Snowflake schema for a given  
problem statement**

# Experiment No. 2

13. **Aim:** Design dimensional data model i.e. Star schema and Snowflake schema for a given problem statement
14. **Objectives:** Building dimensional modeling to simplify complex data structures for efficient querying and reporting, enhancing data accessibility and performance in business intelligence processes.
15. **Course Outcomes:** The outcome is to Optimized database queries for faster data retrieval, improving the efficiency of reporting and analytics.
16. **Hardware / Software Required:** Any editor tool to model the schema or Power BI to data models
17. **Theory:** Dimensional modeling is a database design technique used in data warehousing to structure data into fact and dimension tables, optimizing it for easy retrieval, analysis, and reporting. The outcomes of dimensional modeling include:
  1. **Simplified Data Analysis:** Easy-to-understand data structures that facilitate quicker and more intuitive analysis by end users.
  2. **Enhanced Query Performance:** Optimized database queries for faster data retrieval, improving the efficiency of reporting and analytics.
  3. **Improved Business Insights:** Clear, actionable insights through well-organized dimensions and facts that align with business processes.
  4. **Data Consistency:** Consistent and accurate reporting across different departments due to standardized data representation.
  5. **Scalability:** A flexible model that can accommodate growing data volumes and evolving business needs without significant redesign.
  6. **Better Decision-Making:** Informed decision-making supported by a structured, logical, and comprehensive view of business data.

## Star Schema

The star schema is a type of database schema that organizes data into a central fact table connected to multiple dimension tables. The fact table contains quantitative data, such as sales or revenue, while the dimension tables store descriptive information, like time,

product, or customer details. The schema gets its name because the diagram of the model resembles a star, with the fact table at the center and the dimension tables radiating outward. The star schema is simple, intuitive, and efficient for querying and reporting, making it popular in data warehousing and business intelligence.

## Snowflake Schema

The snowflake schema is a more complex variant of the star schema, where dimension tables are normalized into multiple related tables. This normalization reduces data redundancy by splitting the dimension tables into additional tables that are linked together, resembling a snowflake shape in a diagram. While this schema can lead to more efficient storage and less data redundancy, it may result in more complex queries and slightly slower performance compared to the star schema. Snowflake schemas are useful when dealing with large, complex datasets where storage efficiency is a priority.

### 7. Algorithm / Design / Procedure / Flowchart / Analysis:

For the identified tables and values in experiment 1, draw the dimension model, star schema and snowflake schema using Power BI or any other tool. For example look at the diagram below

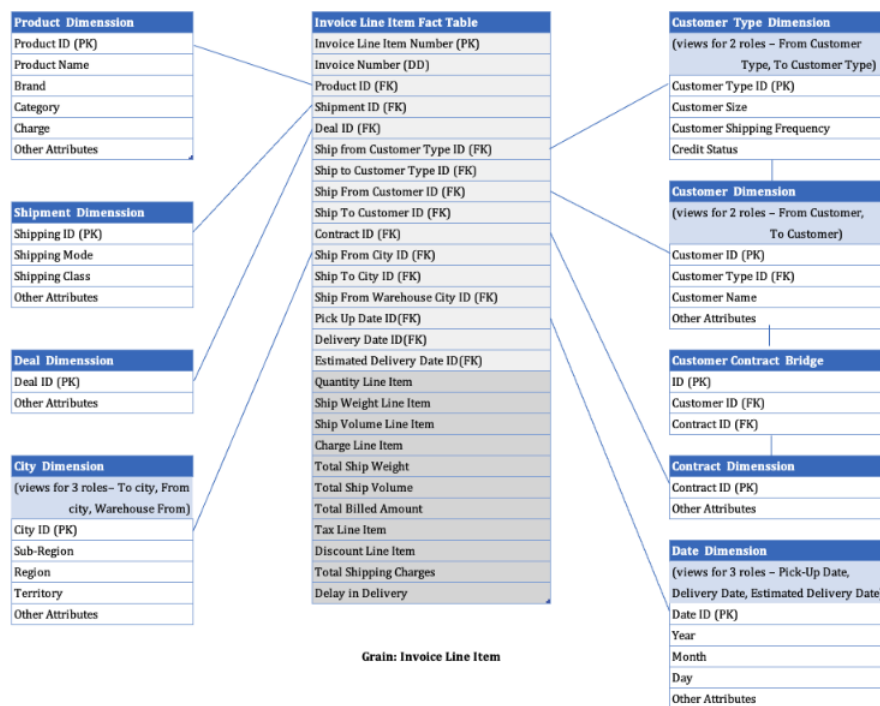


Figure : Star Schema

8. **Results/Output Analysis:** Take a print of the star schema and snowflake schema and justify the Dimension and Fact table with respect to the selected use case.
9. **Conclusions:** Discuss whether the experiment's aim was achieved and how the schema relate to your use case
10. **Viva Questions:** A list of potential questions related to the chosen business use case and dimension model can be expected.
11. **References:**  
<https://www.techshashank.com/data-warehousing/shipping-dimensional-modeling>

# **Data Warehouse and Mining**

## **Experiment No.: 3**

### **Build Data Warehouse/Data mart**



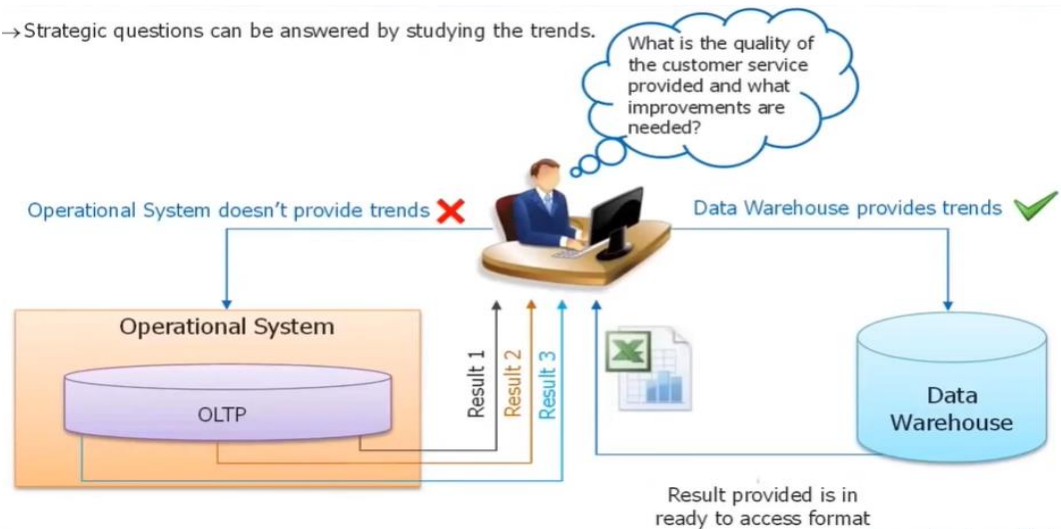
# Experiment No. 3

1. **Aim:** Build Data Warehouse/data mart using a modern tool
2. **Objectives:** To consolidate and store large volumes of data from various sources into a unified, optimized repository, enabling efficient analysis, reporting, and informed decision-making across an organization.
3. **Course Outcomes:** the creation of a centralized, consistent, and accessible data repository that enhances data analysis, supports accurate reporting, and facilitates data-driven decision-making across an organization.
4. **Hardware / Software Required:** Power BI tool to create the staging area of Data Warehouse
5. **Theory:** A data warehouse is a specialized type of database designed to support the efficient storage, retrieval, and analysis of large volumes of data. It consolidates data from various sources, such as operational databases, external data feeds, and other systems, into a unified repository. Key characteristics of a data warehouse include:
  - a. **Centralized Data Storage:** Aggregates data from multiple sources, providing a single source of truth for reporting and analysis.
  - b. **Data Integration:** Combines data from disparate systems, ensuring consistency and accuracy across the organization.
  - c. **Optimized for Query Performance:** Designed to handle complex queries and large datasets efficiently, supporting timely decision-making.
  - d. **Historical Data:** Maintains historical data to enable trend analysis and longitudinal studies.
  - e. **Support for Business Intelligence:** Facilitates advanced analytics, reporting, and data mining, empowering users to extract actionable insights.
  - f. **Scalability:** Capable of scaling to accommodate growing volumes of data and evolving business needs.
  - g. A data warehouse enhances an organization's ability to make informed decisions by providing a reliable, comprehensive, and efficient data infrastructure.

**Algorithm / Design / Procedure / Flowchart / Analysis:**

- h. For the identified tables and values in experiment 1, and for the data models designed in experiment 3, for the constructed data warehouse identify the possible queries/ strategic questions to answer.

→ Strategic questions can be answered by studying the trends.



i.

j. Figure: Data Warehouse providing data trends

6. **Results/Output Analysis:** List down atleast 10 strategic queries suitable and relevant to the dimension model created in experiment 2 and loaded in experiment 3.
7. **Conclusions:** Discuss about the Strategic queries in a data warehouse and how they provide essential insights for decision-making, performance monitoring, trend analysis, and resource optimization, supporting informed and data-driven business strategies.
8. **Viva Questions:** A list of potential questions related to the chosen business use case and dimension model and data warehouse/ data mart characteristics can be expected.
9. **References:**
  - Kimball Group: Kimball Group's Website offers articles and resources on dimensional modeling and data warehousing.
  - Coursera: Courses on Data Warehousing and Data Management, such as "Data Warehousing for Business Intelligence" by the University of Colorado.

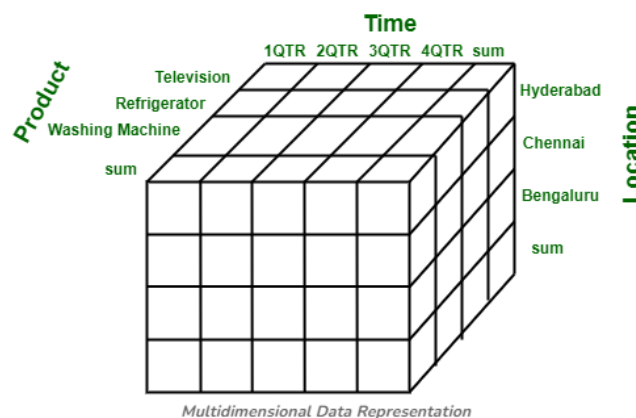
# **Data Warehouse and Mining**

## **Experiment No.: 4**

### **Creating and Visualizing a Cube Using any Modern Tool**

# Experiment No. 4

- 1 **Aim:** Creating and visualizing a Cube using any modern tool
- 2 **Objectives:** Visualizing a cube in a data warehouse to simplify complex data analysis by providing a multi-dimensional view of the data
- 3 **Course Outcomes:** Multi-dimensional views provide a clear understanding of relationships and patterns across different data dimensions, making the data more comprehensible.
- 4 **Hardware / Software Required:** Power BI tool to create the multi-dimensional view.
- 5 **Theory:** In data warehouse data modeling, a **cube** is a multi-dimensional data structure that allows for efficient querying and analysis of data across various dimensions. It organizes data into a structure that resembles a 3D cube, where each axis or dimension represents a different category of data, and the data within the cube represents metrics or facts such as sales, profit, or inventory. It represents data in the form of data cubes. Data cubes allow to model and view the data from many dimensions and perspectives. It is defined by dimensions and facts and is represented by a fact table. Facts are numerical measures and fact tables contain measures of the related dimensional tables or names of the facts. This is as represented in the below figure.



In the above given presentation, the factory's sales for Bangalore are, for the time dimension, which is organized into quarters and the dimension of items, which is sorted according to the kind of item which is sold. The facts here are represented in rupees (in thousands).

Location = "Bangalore"				
Time (quarter)	Type of item			
	Jam	Bread	Sugar	Milk
Q1	350	389	35	50
Q2	260	528	50	90
Q3	483	256	20	60
Q4	436	396	15	40

6

2D factory data

Now, if we desire to view the data of the sales in a three-dimensional **table**, then it is represented in the diagram given. Here the data of the sales is represented as a two **dimensional table**. Let us consider the data according to item, time and location (like Kolkata, Delhi, Mumbai).

	Location="Kolkata"			Location="Delhi"			Location="Mumbai"		
	item			item			item		
	Milk	Egg	Bread	Milk	Egg	Bread	Milk	Egg	Bread
Q1	340	604	38	335	365	35	336	484	80
Q2	680	583	10	684	490	48	595	594	39
Q3	535	490	50	389	385	15	366	385	20

7

3D data representation as 2D

Location

Mumbai

Delhi

Kolkata

Time(quarters)

Q1

Q2

Q3

Milk

Egg

Bread

item (types)

3D data representation

**Algorithm / Design / Procedure / Flowchart / Analysis:**

Load the data of your respective use case into power BI

Select the three relevant dimensions to visualize the data

Select the relevant visualization format from visualization pane

Analyse the 3D data loaded in the form of table

**8 Results/Output Analysis:**

The sample table of PoweBI for representing 3D data is shown below.

Country	Amarilla	Carretera	Montana	Paseo	Velo	VTT	Total
Canada	646,861.38	436,105.34	321,867.03	1,265,017.99	370,568.34	488,808.81	3,529,228.89
France	667,867.63	388,864.90	461,238.37	838,748.56	707,930.24	716,371.09	3,781,020.78
Germany	612,137.26	369,674.68	559,438.37	744,416.74	788,789.00	605,932.77	3,680,388.82
Mexico	498,611.39	393,668.42	337,689.31	928,651.39	173,303.89	575,598.71	2,907,523.11
United States of America	388,626.41	238,491.55	434,521.80	1,020,603.27	265,401.00	647,896.64	2,995,540.67
Total	2,814,104.06	1,826,804.89	2,114,754.88	4,797,437.95	2,305,992.47	3,034,608.02	16,893,702.26

- 9 **Conclusions:** By analyzing this 3D data , users can easily identify trends, correlations, and key performance indicators, turning raw data into actionable insights for business strategy and operations.

- 10 **Viva Questions:** A list of potential questions related to the OLAP operations can be expected.

**11 References:**

- Kimball Group: Kimball Group's Website offers articles and resources on dimensional modeling and data warehousing.
- "The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling" by Ralph Kimball and Margy Ross
- Building the Data Warehouse" by William H. Inmon

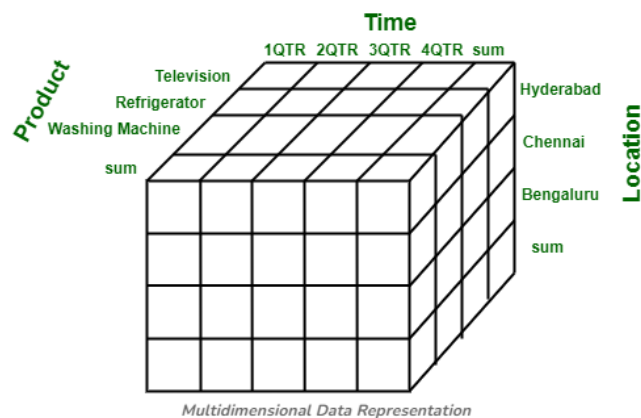
# **Data Warehouse and Mining**

## **Experiment No.: 5**

**To perform various OLAP operations  
such as slice, dice, drilldown, rollup, pivot  
using any open source tool**

# Experiment No. 5

- 1 **Aim:** To perform various OLAP operations such as slice, dice, drilldown, rollup, pivot using any open source tool
- 2 **Objectives:** to visualize and facilitate multidimensional data analysis and decision-making
- 3 **Course Outcomes:** Multi-dimensional views provide a clear understanding of relationships and patterns across different data dimensions, making the data more comprehensible.
- 4 **Hardware / Software Required:** Power BI tool to create the multi-dimensional view.
- 5 **Theory:** In data warehouse data modeling, a **cube** is a multi-dimensional data structure that allows for efficient querying and analysis of data across various dimensions. It organizes data into a structure that resembles a 3D cube, where each axis or dimension represents a different category of data, and the data within the cube represents metrics or facts such as sales, profit, or inventory. It represents data in the form of data cubes. Data cubes allow to model and view the data from many dimensions and perspectives. It is defined by dimensions and facts and is represented by a fact table. Facts are numerical measures and fact tables contain measures of the related dimensional tables or names of the facts. This is as represented in the below figure.



a.

b. Figure: Multidimensional Data Representation

## Example

Consider the following cubes illustrating temperature of certain days recorded weekly:



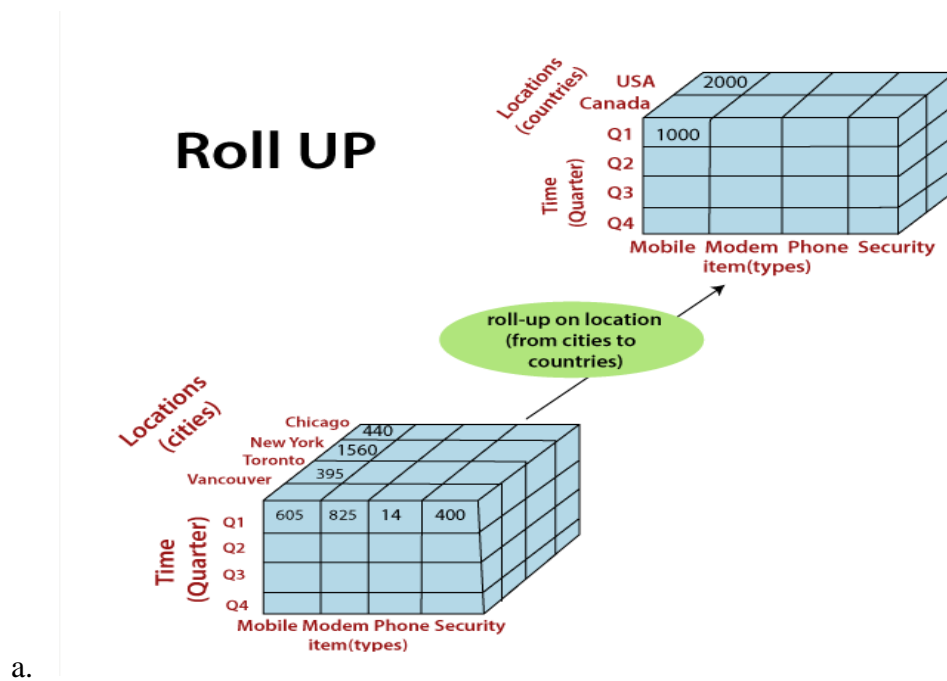
6 Temperature	7 64	8 65	9 68	10 69	11 70	12 71	13 72
19 Week1	20 1	21 0	22 1	23 0	24 1	25 0	26 0
32 Week2	33 0	34 0	35 0	36 1	37 0	38 0	39 0

Consider that we want to set up levels (hot (80-85), mild (70-75), cool (64-69)) in temperature from the above cubes. To do this, we have to group column and add up the value according to the concept hierarchies. This operation is known as a roll-up. By doing this, we contain the following cube:

45 Temperature	46 cool	47 mild	48 hot
49 Week1	50 2	51 1	52 1
53 Week2	54 2	55 1	56 1

The roll-up operation groups the information by levels of temperature.

The following diagram illustrates how roll-up works.



## Drill-Down

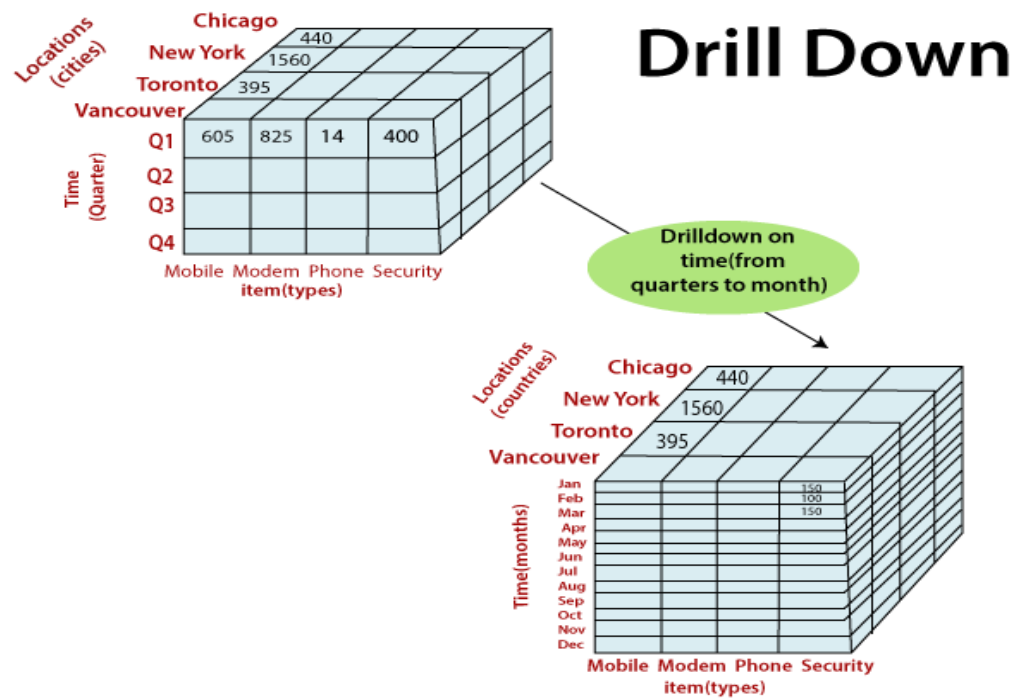
The drill-down operation (also called roll-down) is the reverse operation of roll-up. Drill-down is like zooming-in on the data cube. It navigates from less detailed record to more detailed data. Drill-down can be performed by either stepping down a concept hierarchy for a dimension or adding additional dimensions. Figure shows a drill-down operation performed on the dimension time by stepping down a concept hierarchy which is defined as day, month, quarter, and year. Drill-down appears by descending the time hierarchy from the level of the quarter to a more detailed level of the month. Because a drill-down adds more details to the given data, it can also be performed by adding a new dimension to a cube. For example, a drill-down on the central cubes of the figure can occur by introducing an additional dimension, such as a customer group.

### Example Drill-down adds more details to the given data

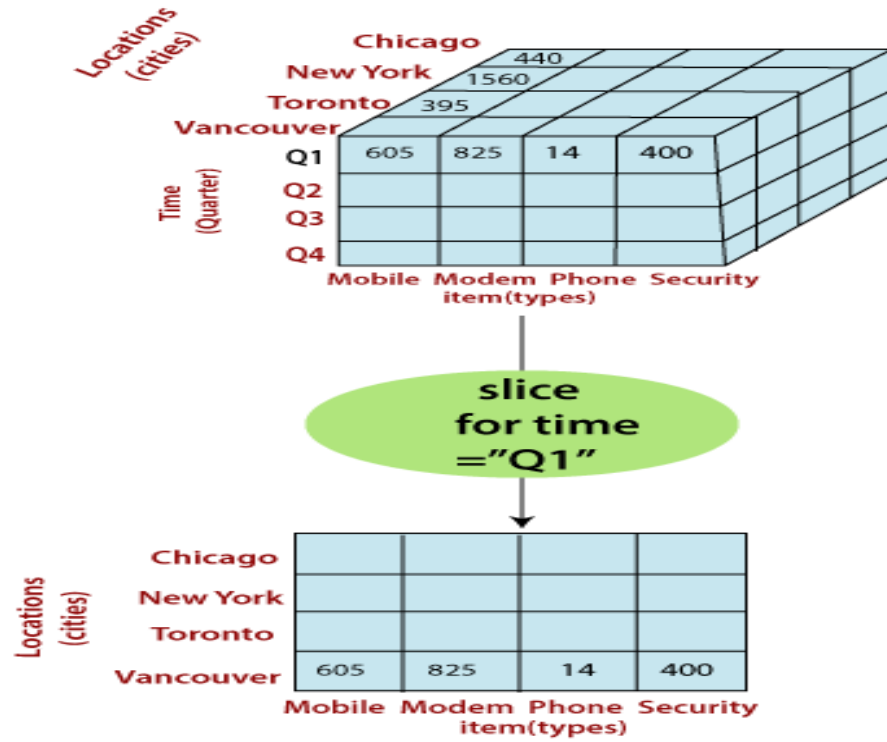
57 Temperature	58 cool	59 mild	60 hot
61 Day 1	62 0	63 0	64 0
65 Day 2	66 0	67 0	68 0
69 Day 3	70 0	71 0	72 1
73 Day 4	74 0	75 1	76 0
77 Day 5	78 1	79 0	80 0
81 Day 6	82 0	83 0	84 0
85 Day 7	86 1	87 0	88 0
89 Day 8	90 0	91 0	92 0

93 Day 9	94 1	95 0	96 0
97 Day 10	98 0	99 1	1000
101Day 11	1020	1031	1040
105Day 12	1060	1071	1080
109Day 13	1100	1110	1121
113Day 14	1140	1150	1160

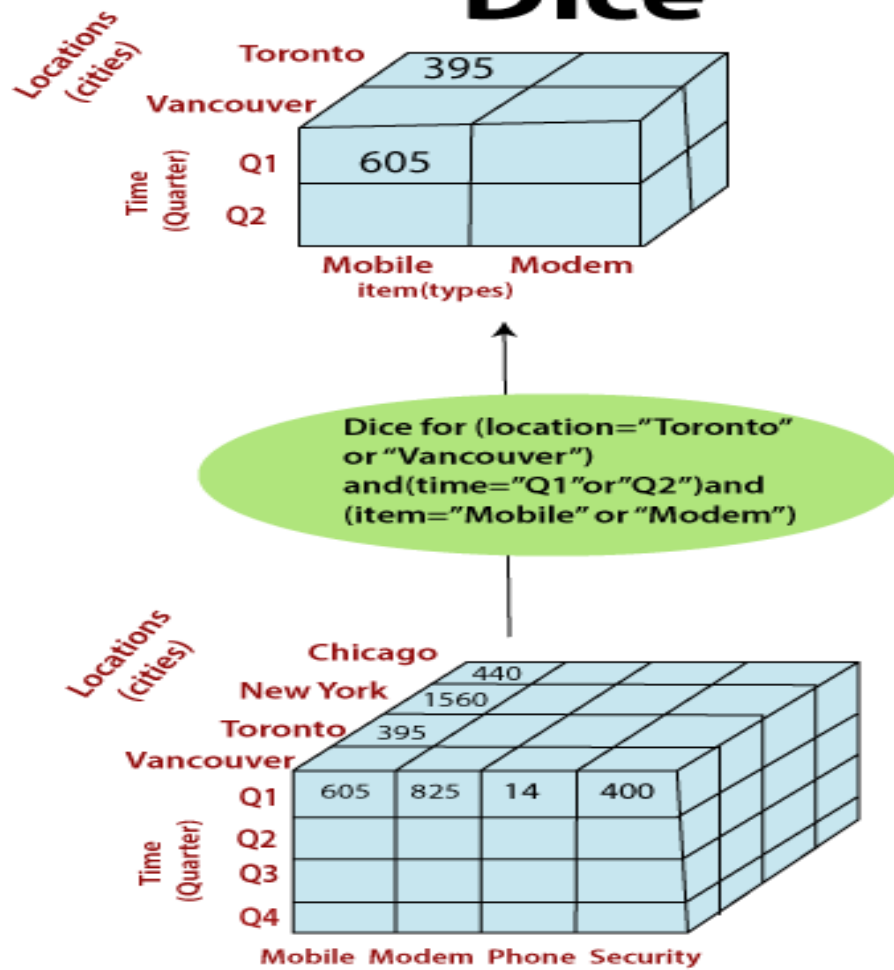
The following diagram illustrates how Drill-down works.



# Slice



# Dice

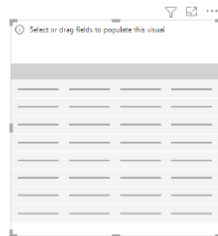


Algorithm / Design / Procedure / Flowchart / Analysis:

## First Step

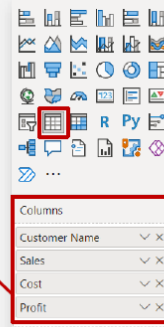
### Creating a Table

Click on the table visual to add it on your canvas. Once added, drag any field you want to visualize in the columns box.



Customer Name	Sales	Cost	Profit
Aaron Bootman	1635	618	817
Aaron Cunningham	820	150	670
Aaron Davey	1862	1119	743
Aaron Macrossan	81	56	25
Abbie Perry	619	128	491
Abby Colebe	676	500	176
Abby Mei	1504	393	1511
Abby Murawski	1010	381	629
Abigail Humfray	744	461	283
Ada Dalton	773	565	208
Adam Bentley	3458	1807	1651
Adam Gibbs	815	625	190
Adam Harris	2540	1216	1324
Adam Peacock	589	492	97
Total	1267544	640597	626947

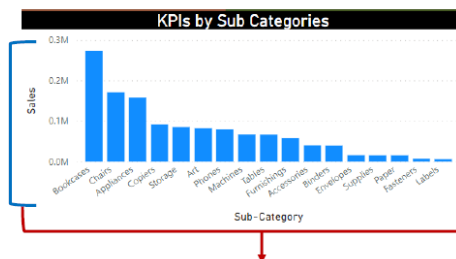
Add any field in columns as shown below.



## Second Step

### Creating a Column Chart – X-axis & Y-axis

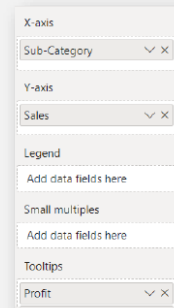
Sales is your Y-axis  
Adding numerical metric as Y-axis helps the user understand the extent of it by the column height.



Sub-Category is your X-axis

Add the text based field that you want to see in the X-axis. As we are using a column chart, numerical metrics like sales, profit etc will become Y-axis as we want the column HEIGHT to show the extent of that metric.

Add fields in X & Y Axis as shown below.



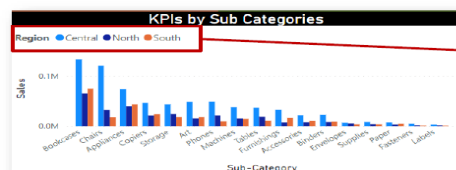
## Third Step

### Creating a Column Chart – Legend

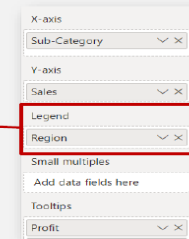
Legend further divides your visual into different colors by adding a second level of breakdown.

Adding Region to legend breaks all our subcategories from 1 column for each subcategory to 3 columns for each sub-category, all having different colors.

For ex – 1 Column of bookcases becomes 3 columns of Bookcases from Central, Bookcases from North & Bookcases from South.



Add fields in Legend as shown below.

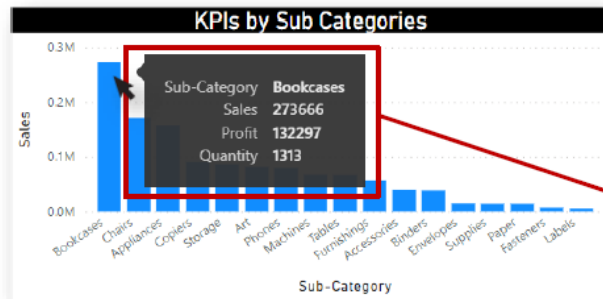


#### Fourth Step

### Creating a Column Chart – Tooltips

**Any metric (numerical field) that you add in Tooltips will be visible to you when you hover over different elements of the visual.**

For ex – After adding Profit to tooltips, whenever user hovers over ANY column of the column chart, they will see profit of that subcategory along with the Sales amount as well. Profit amount will only be visible as a value in the hover box and not as an additional column in the chart.



**Add fields in Tooltips as shown below.**

The configuration pane shows the following settings:

- X-axis: Sub-Category
- Y-axis: Sales
- Legend: Add data fields here
- Small multiples: Add data fields here
- Tooltips: Profit

#### 12. Results/Output Analysis:

The sample table of drill down and drill up operations using Power BI for representing 3D data is performed.

13. **Conclusions:** By analyzing this 3D data at various legends, users can easily identify trends, correlations, and key performance indicators, turning raw data into actionable insights for business strategy and operations by drilling down the data.

14. **Viva Questions:** A list of potential questions related to the OLAP operations can be expected.

#### References:

1. Kimball Group: Kimball Group's Website offers articles and resources on dimensional modeling and data warehousing.
2. "The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling" by Ralph Kimball and Margy Ross
3. Building the Data Warehouse" by William H. Inmon
4. <https://www.javatpoint.com/olap-operations>

# **Data Warehouse and Mining**

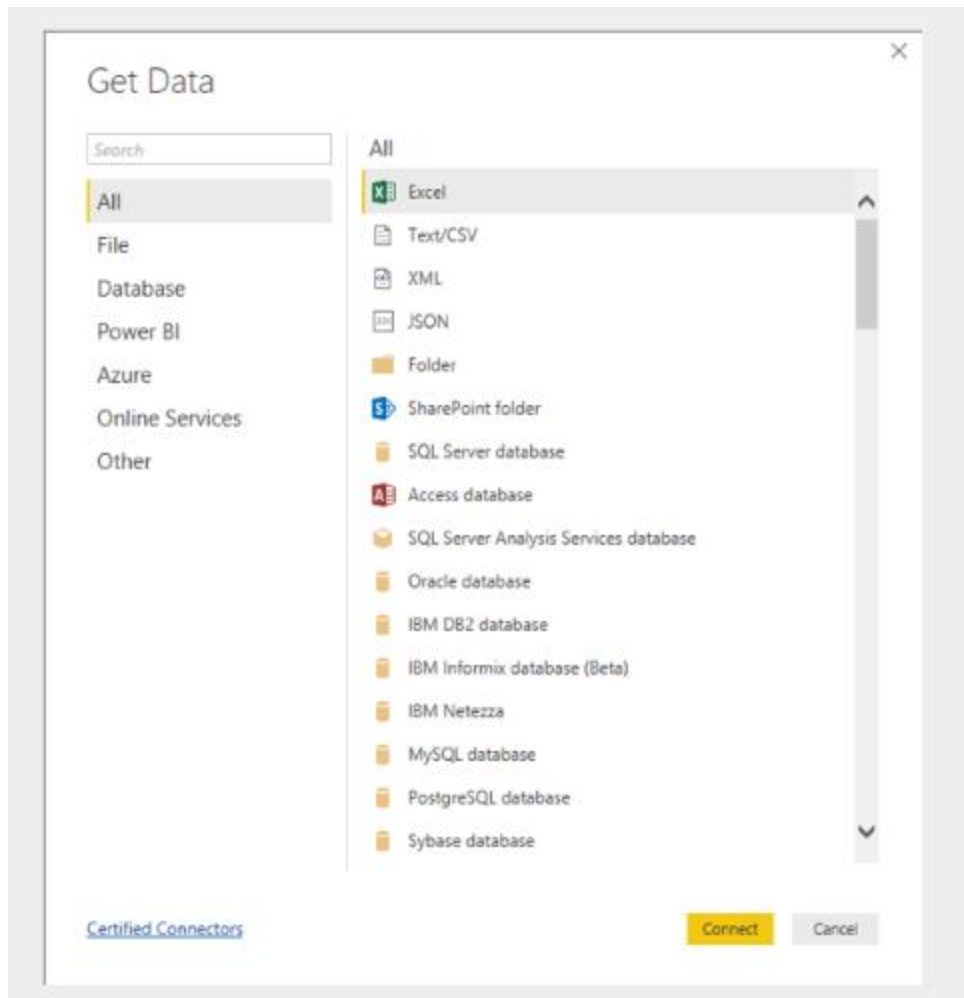
## **Experiment No.: 6**

**Perform the Extraction and  
Transformation process to load the data  
using Sqlserver / Power BI.**

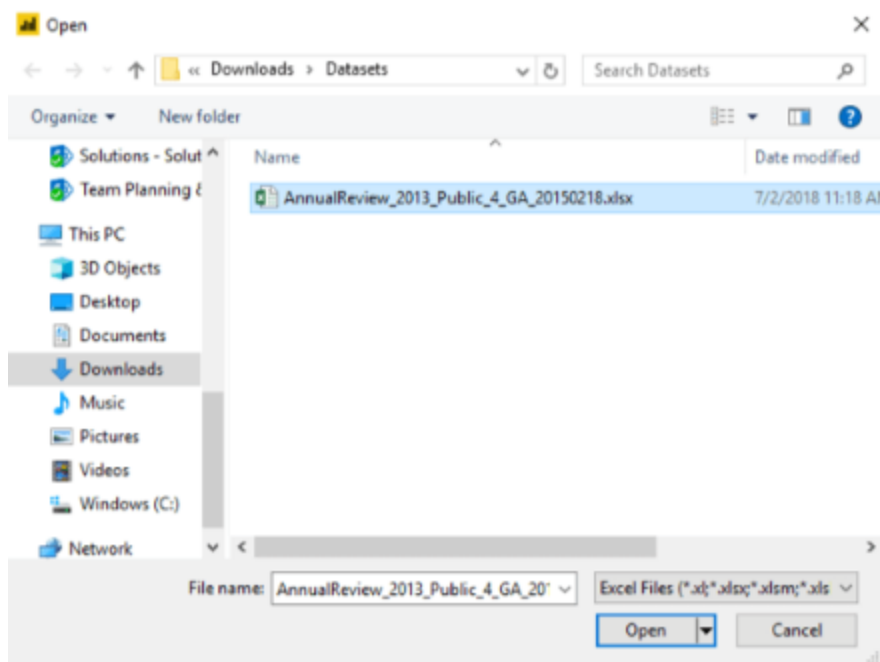


# Experiment No. 6

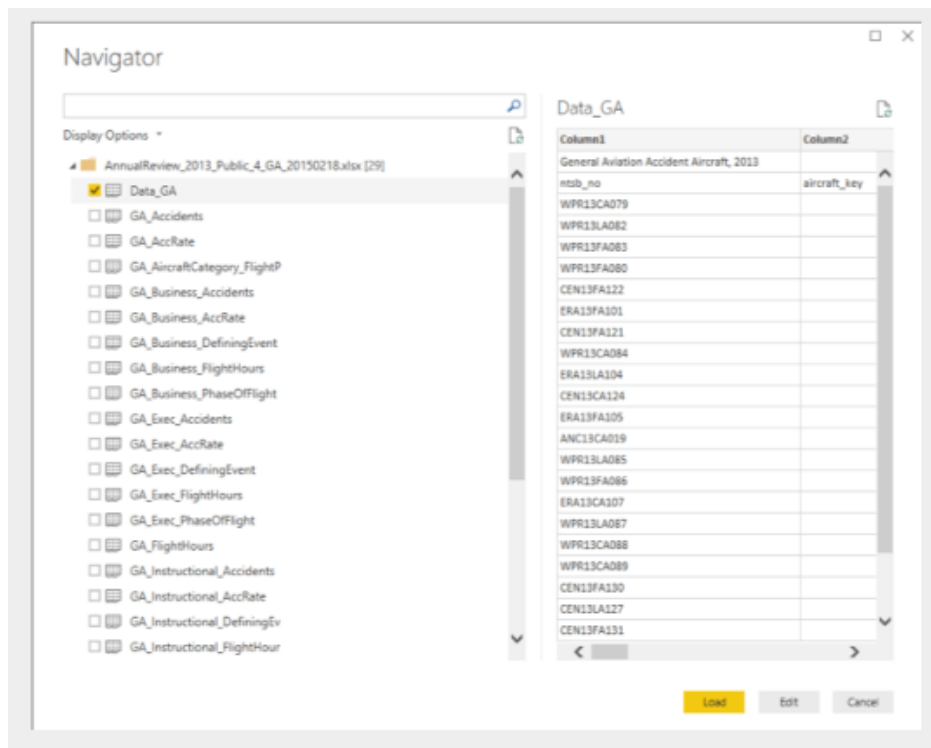
- 1 **Aim:** Perform the Extraction and Transformation process to load the data using Sqlserver / Power BI.
- 2 **Objectives:** To extract the data for transformation process.
- 3 **Course Outcomes:** Initiate the ETL process by connecting the data source and extract the data, which encompasses a wide range of options, including databases, Excel files, web services, and more.
- 4 **Hardware / Software Required:** Power BI tool to extract and load data into the staging area.
- 5 **Theory:** ETL is Extract, Transform, Load. It's a form of a data pipeline to integrate various data sources. Without it, your analytical reports and dashboards look old because of outdated data. ETL helps update them so your reports are current. Meanwhile, Microsoft defines Power BI: *Power BI is a collection of software services, apps, and connectors that work together to turn your unrelated sources of data into coherent, visually immersive, and interactive insights.*
  - a. **The first step is to launch Power BI Desktop, then follow these steps:**
  - b. 1 From the Power BI splash screen or toolbar, click on "Get Data," select the Excel connector and click "Connect."



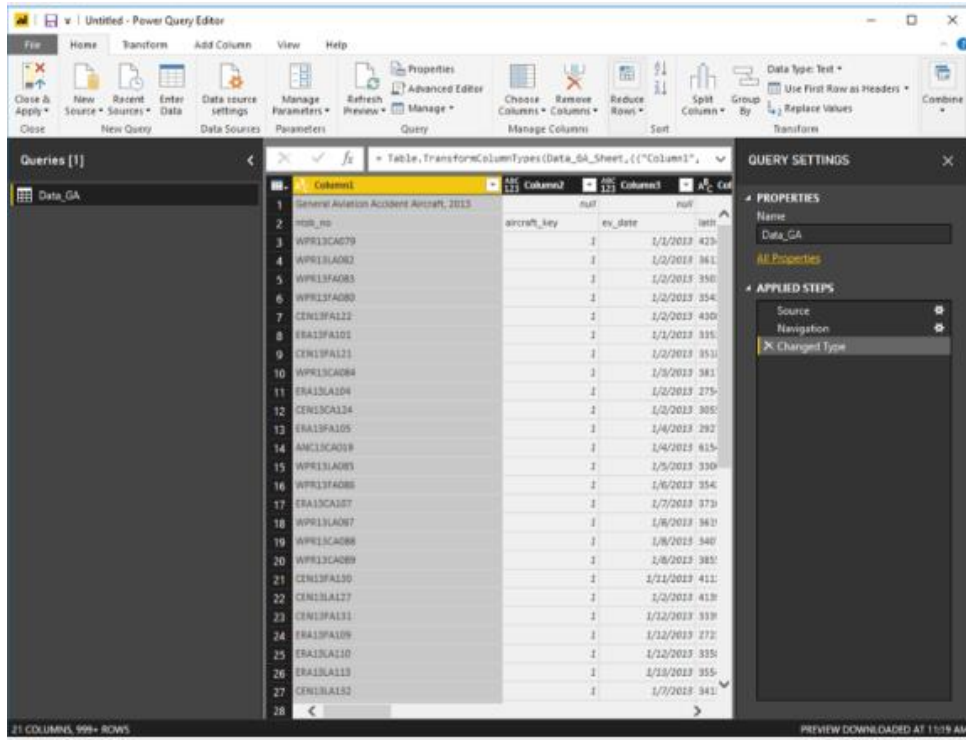
2. Browse for the Excel file, select it from the list and click “Open.”



3 On the Navigator dialog, select “Data\_GA” and click “Edit.”



The Power Query Editor opens to shape and transform our data.



You can see that, on the right, the Query Settings pane lists all of the Applied Steps taken so far. These steps were applied automatically to indicate the path of the source file (Source), the columns that were automatically discovered (Navigation) and the data types automatically detected (Changed Type).

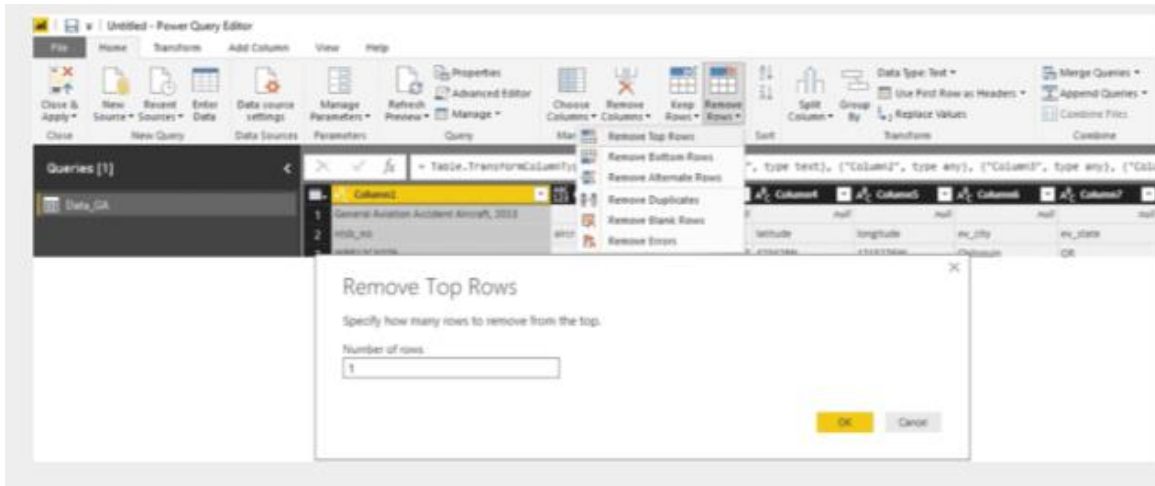
## 5. Results/Output Analysis:

The sample screenshots of extracting and loading data is to be demonstrated.

In cases where data is presented as a table with all of the appropriate headers and no empty rows, these steps will do the job for us accurately. In our case, the source excel file needs a little bit of clean-up—so we'll shape and transform the data as described below:

### 1. Set Appropriate Column Headers.

In this step, we'll need to remove the first row because our column headers are actually in the second row. To remove the first row, click on "Remove Rows" from the Power Query Editor toolbar, click on "Remove Top Rows" and type "1" in the Number of rows dialog box.

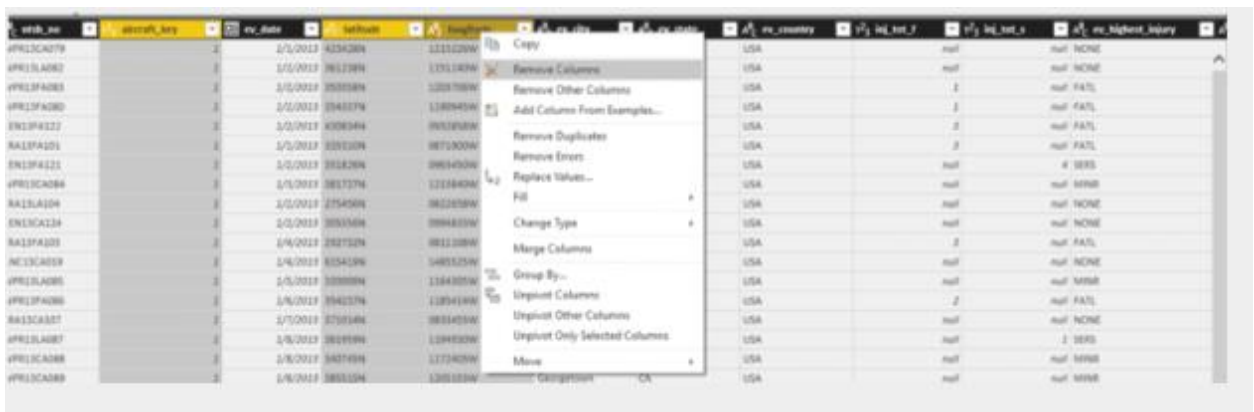


Now, we can indicate that the first row should be used as our column headers. To do this, click the “Use First Row as Headers” option on Power Query toolbar.



## ii Remove Unnecessary Columns

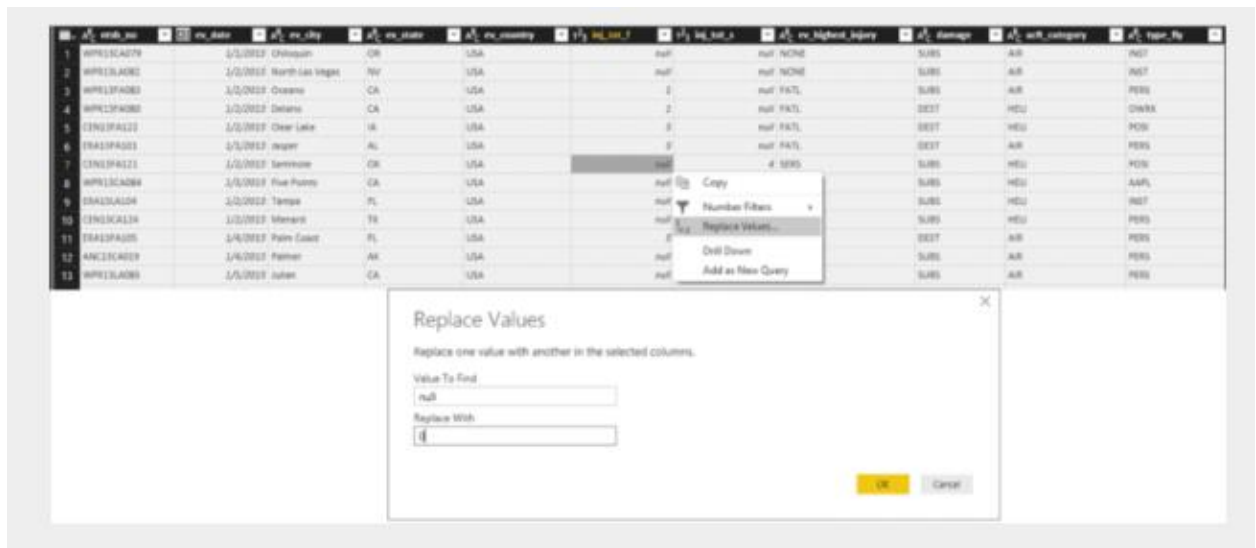
In this step, we need to remove all of the columns that we don’t need to analyze. To remove columns, select the columns while pressing the CTRL key, then right-click and select “Remove Columns.”



## iii Replace Null Values

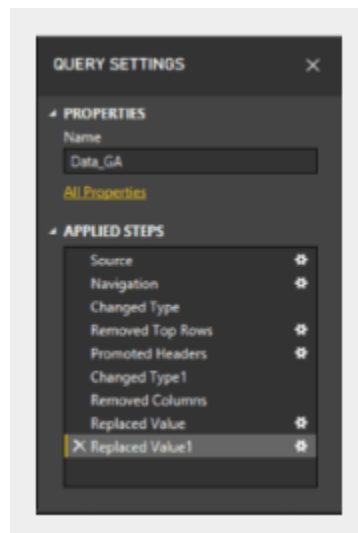
In this step, we need to replace null values with the number zero to ensure accurate analysis and to standardize values. In this example, we can replace all the null values for the “inj\_tot\_f” (Fatal Injuries)

and “inj\_tot\_s” (Serious Injuries) columns. To do this, right-click on any row with a null value and select “Replace Values...” then type the number “0” in the Replace With dialog box and click “OK.”



Repeat for any remaining column with null values. All null values will be now replaced with 0.

Each of these data shaping and transformation steps have been recorded in the Applied Steps pane as seen below. So, whenever data in the data source changes, all you need to do is refresh your Power BI file to reflect the changes. Power Query will then apply the same steps we applied, which will refresh the data in the internal tables as well as all visualizations that reference them.



6. **Conclusions:** ETL process of extraction by connecting the data source to extract the data, was performed to continue with the transformation stage of ETL process.

7. **Viva Questions:** A list of potential questions related to the ETL operations can be expected.

8. **References:**

- Kimball Group: Kimball Group's Website offers articles and resources on dimensional modeling and data warehousing.
- "The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling" by Ralph Kimball and Margy Ross
- Building the Data Warehouse" by William H. Inmon
- <https://agilethought.com/blogs/extract-transform-load-data-power-bi/>

# **Data Warehouse and Mining**

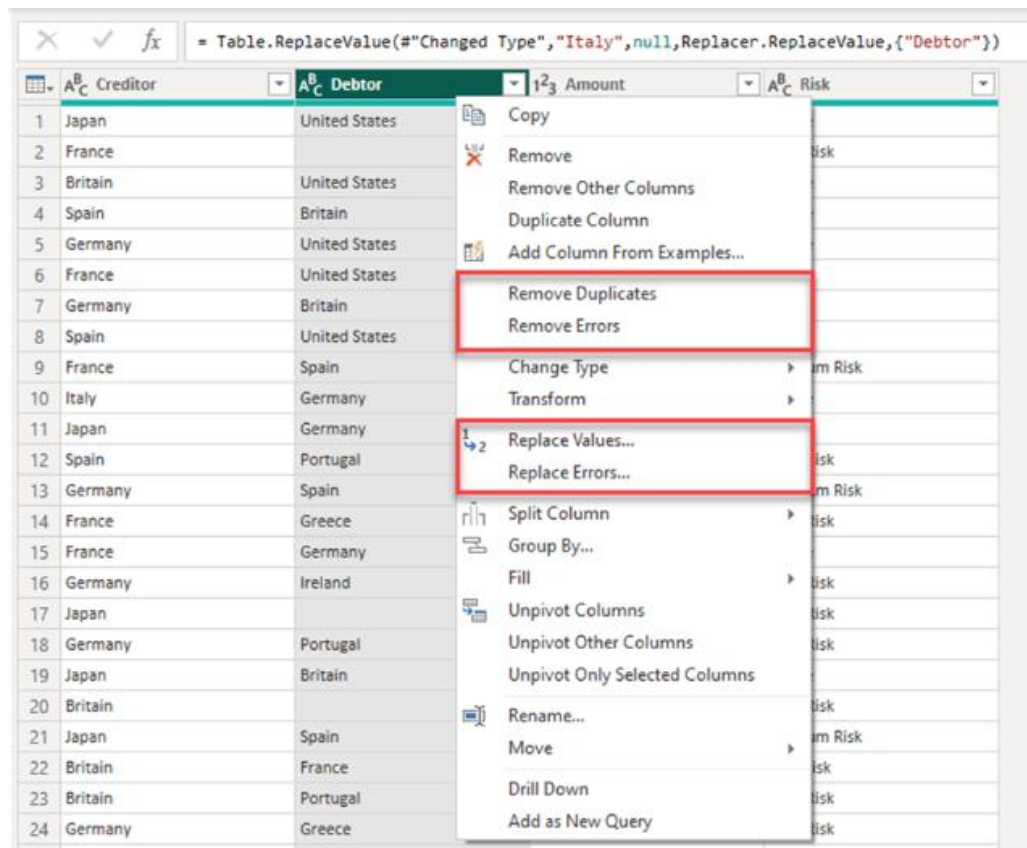
## **Experiment No.: 7**

**Perform Data Preprocessing on created  
database in Power BI.**



# Experiment No. 7

- 1 **Aim:** Perform Data Preprocessing on created database in Power BI.
- 2 **Objectives:** To perform the data transformation process to generate meaningful reports.
- 3 **Course Outcomes:** Initiate the transformation process by connecting the data source and transform the data, which encompasses a wide range of options, including data cleaning, column renaming and data type conversion.
- 4 **Hardware / Software Required:** Power BI tool to extract and load data into the staging area.
- 5 **Theory: Data Cleaning:** Data cleaning stands as an essential step in working with data, as it often arrives with inconsistencies, missing values, and errors. Some fundamental functionalities you should master in Power BI for data cleaning include removing duplicates, filling in missing values, and correcting data types.

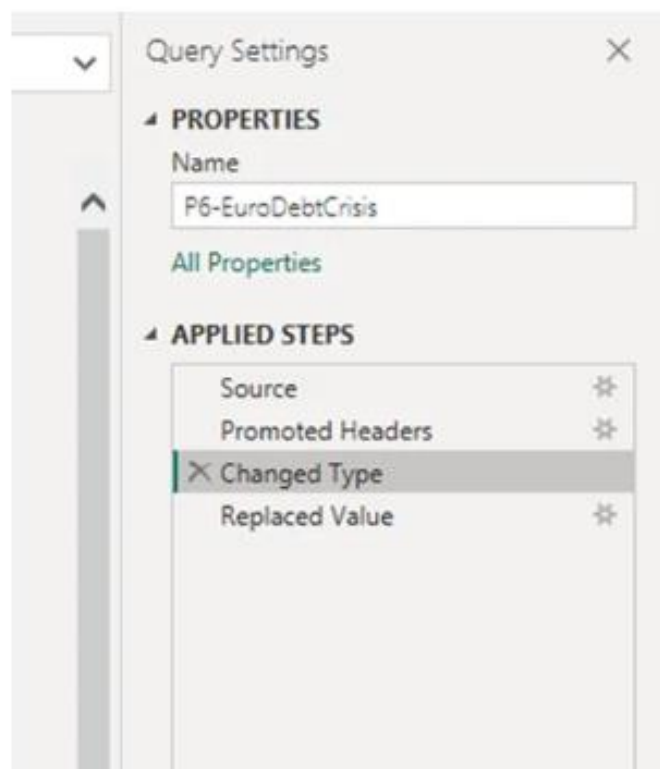


**Column Renaming:**

Column renaming can be a cumbersome task, but it can significantly enhance the clarity of your reports. In Power Query Editor, rename columns to make them more comprehensible. Simply right-click on a column header and select "Rename..." (Refer to the image above for guidance).

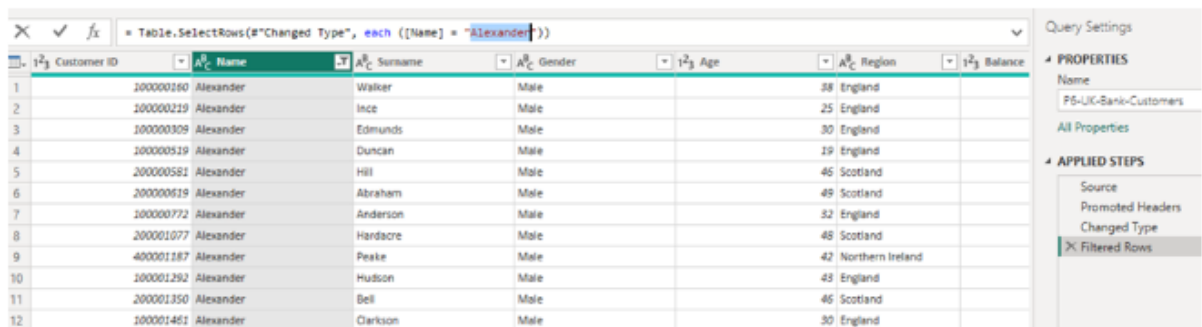
### **Data Type Conversion:**

Power BI generally does a commendable job of recognizing data types during the initial data source load. However, in cases where this information is incorrect, it is advisable to navigate to the "Change Type" option in your Change Pane, located on the rightmost side of the screen. Ensure you are at this step and update any incorrect data types. This is especially crucial if you have numeric fields that should be text-based, such as when you need to retain leading zeros. Use the "Change Type" feature in Power Query Editor to convert columns to the appropriate data types, like changing a text column to a date.



### **Filtering Data:**

Remove unnecessary rows or filter data based on specific criteria. Apply filters to select the relevant data for your analysis. This is particularly valuable when dealing with large datasets, where you may only be interested in specific data subsets. Additionally, when working with extensive datasets, you may not immediately see the desired filter, especially when filtering for specific values. In such cases, it can be advantageous to select a value first and then subsequently refine it to your specific criteria within the formula.

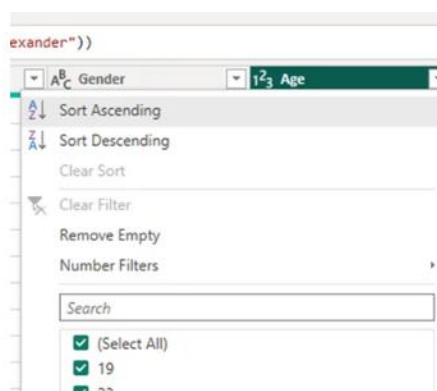


The screenshot shows a data table with columns: Customer ID, Name, Surname, Gender, Age, Region, and Balance. A formula bar at the top contains the formula: `=Table.SelectRows(#"Changed Type", each ([Name] = "Alexander"))`. The table displays 12 rows of data for customers named Alexander. On the right, a sidebar shows 'Query Settings' with 'Name' set to 'PS-UK-Bank-Customers', 'All Properties' expanded, and 'APPLIED STEPS' including 'Source', 'Promoted Headers', 'Changed Type', and 'Filtered Rows'.

	Customer ID	Name	Surname	Gender	Age	Region	Balance
1	100000180	Alexander	Walker	Male		38 England	
2	100000219	Alexander	Ince	Male		25 England	
3	100000309	Alexander	Edmunds	Male		30 England	
4	100000519	Alexander	Duncan	Male		19 England	
5	200000581	Alexander	Hill	Male		46 Scotland	
6	200000619	Alexander	Abraham	Male		49 Scotland	
7	100000772	Alexander	Anderson	Male		32 England	
8	200001077	Alexander	Hardacre	Male		48 Scotland	
9	400001187	Alexander	Peake	Male		42 Northern Ireland	
10	100001292	Alexander	Hudson	Male		43 England	
11	200001350	Alexander	Bell	Male		46 Scotland	
12	100001461	Alexander	Clarkson	Male		30 England	

## Sorting Data:

Arrange data in ascending or descending order to enhance readability and analysis. You can sort columns by clicking the dropdown menu for the specific column and choosing "Sort Ascending" or "Sort Descending."



## Adding Custom Columns:

Create new columns based on existing data using Power Query's "Add Column" feature. This is especially useful for calculated columns or for merging data from multiple columns. This feature will quickly become one of your most frequently used tools for data transformation. It's essential to consider when and where to use this; sometimes, creating measures instead of columns may be more advantageous. Below is an example that demonstrates how to segment your data into age buckets.

"Age Bucket", each "Age" &

Age	Region	Job Classification	Date Joined	Balance	Age Bucket
21	England	White Collar	05 Jan.15	11881015	Age 20-29
34	Northern Ireland	Blue Collar	06 Jan.15	3691973	Age 30-39
46	England	White Collar	07 Jan.15	10153683	Age 40-49
32	Wales	White Collar	08 Jan.15	142152	Age 30-39
				3563979	Age 30-39
				12244377	Age 30-39
				4287984	Age 30-39
				3668017	Age 40-49
				7428435	Age 30-39
				1091245	Age 40-49
				3966783	Age 40-49
				3228162	Age 30-39
				4078163	Age 20-29
				4879146	Age 40-49
				284603	Age 30-39
				211685	Age 40-49
				1035631	Age 30-39
				380169	Age 40-49
				6553469	Age 40-49
				1146264	Age 40-49
				317789	Age 30-39
				2125297	Age 50-59
				6678578	Age 40-49
				658081	Age 50-59
				2050532	Age 30-39
				4324926	Age 50-59
				30279	Age 30-39

Custom Column

Add a column that is computed from the other columns.

New column name

Age Bucket

Custom column formula

= "Age " &  
Number.ToText(Number.RoundDown([Age]/10)\*10) &  
"- " &  
Number.ToText(Number.RoundDown([Age]/10)+9)

Available columns

- Customer ID
- Name
- Surname
- Gender
- Age
- Region
- Job Classification

<< Insert

Learn about Power Query formulas

✓ No syntax errors have been detected.

OK Cancel

File Home Transform Add Column View Tools Help

Column From Examples Custom Column Invoke Custom Function General

Conditional Column Index Column Duplicate Column

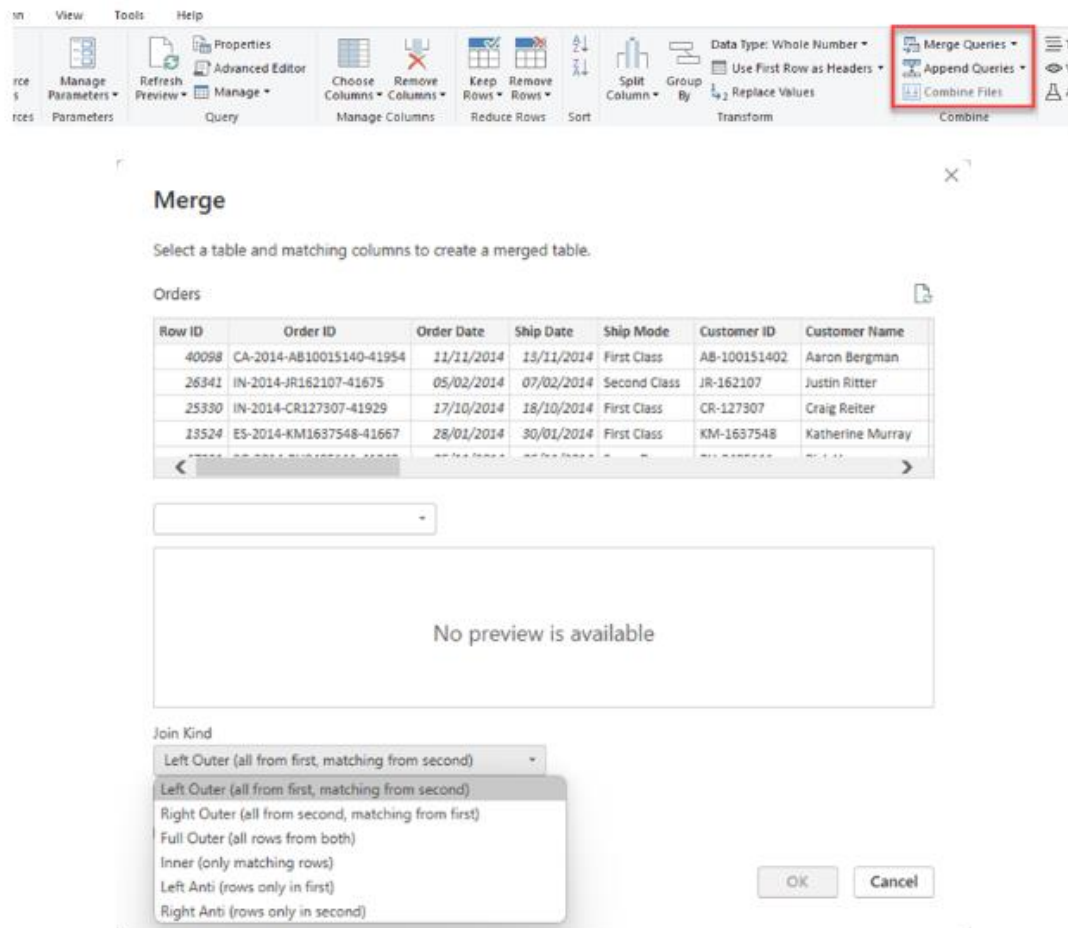
Format Merge Columns Extract Parse From Text

Statistics Standard Scientific

## Merging Queries:

Combine data from multiple sources or tables through query merging. Power BI offers various join types, such as inner join and left outer join, among others. It's crucial to assess whether your goal can be achieved through your data model, as this can help reduce the data load on your report. However, it's not always possible to achieve the

desired result solely through data modeling. Merging and appending data is a straightforward process. One thing to note is that sometimes you must create a new query for the merge to be executed correctly. Also, if you intend to append queries, ensure that the columns have the same naming conventions.

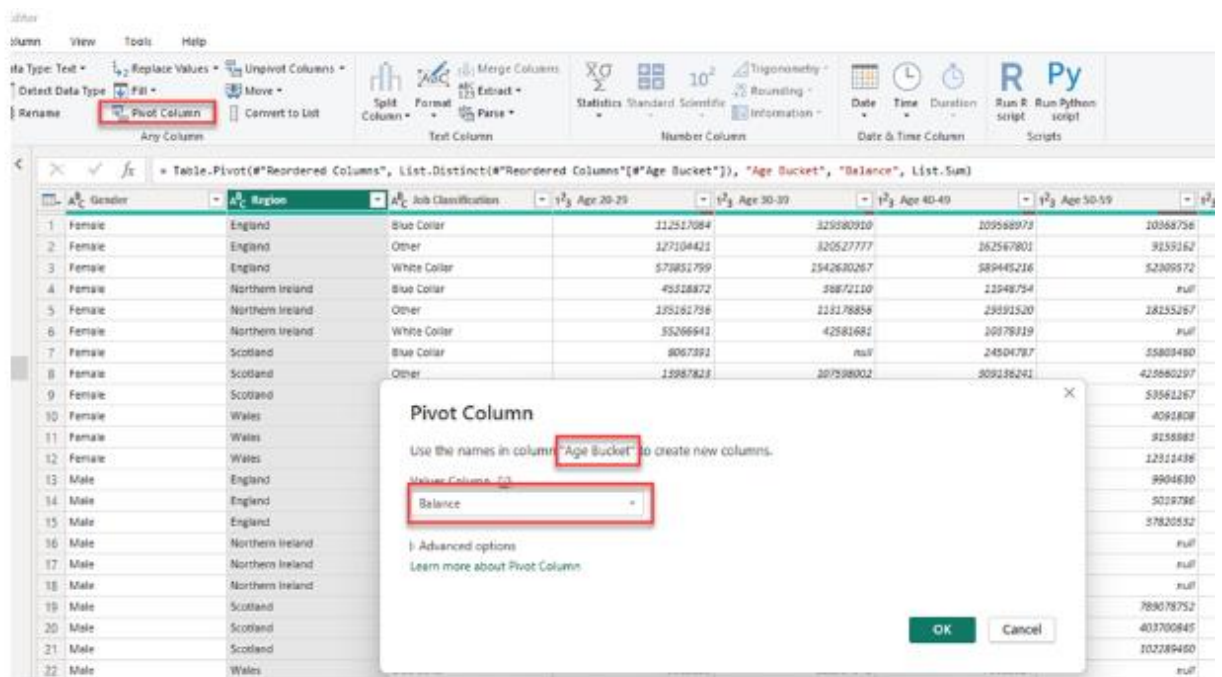


### Aggregating Data:

Group data by specific columns and perform aggregations like sum, average, or count using the "Group By" function. This is particularly useful for summarizing data. You can access the "Group By" feature either in the Ribbon or by right-clicking on a column. If you need to group by multiple columns, either highlight all of them or add them to your grouping in the pop-up window.

### Pivot and Unpivot:

Reshape your data using the "Pivot" and "Unpivot" transformations. Pivot allows you to convert columns into rows, while Unpivot accomplishes the reverse. To pivot, select the column you want to use for column headers and press the pivot option in the Ribbon under the Transform Tab. In the pop-up window, select the column you want to use for values. All other columns will remain as they are. To unpivot, select all the columns you want to transform from pivot to row structure. This process will create two columns: "Attribute" with the column header and "Value."



## 9. Results

Include the steps required to connect Power BI's capabilities to transform and visualize your data effectively. To create powerful and insightful reports and dashboards, you often need to clean and shape your data to make it suitable for analysis.

10. **Conclusions:** By performing the data transformation, we ensure that the data is clean and in uniform format to load and analyze the data.

11. **Viva Questions:** A list of potential questions related to the ETL operations can be expected.

12. **References:**

- Kimball Group: Kimball Group's Website offers articles and resources on dimensional modeling and data warehousing.
- "The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling" by Ralph Kimball and Margy Ross
- Building the Data Warehouse" by William H. Inmon
- <https://www.cittros.com/insights/basic-data-transformation-techniques-in-power-bi>

# **Data Warehouse and Mining**

## **Experiment No.: 8**

### **Implementation of Apriori Association Rule Mining Algorithm using Power BI**



# Experiment No. 8

- 1 **Aim:** Implementation of Apriori Association Rule Mining Algorithm using Power BI.
- 2 **Objectives:** To implement and visualize association rules using the Apriori algorithm in Power BI, based on transactional data, to identify patterns and correlations among items.
- 3 **Course Outcomes:** Students will be able to implement and visualize Apriori-based association rules using Power BI for pattern discovery.
- 4 **Hardware / Software Required:** Power BI tool.
- 5 **Theory:** Association Rule Mining is a data mining technique used to uncover hidden patterns and relationships between items in large transactional datasets. The Apriori algorithm is a popular method for finding frequent itemsets and generating association rules based on measures like support, confidence, and lift. It works by iteratively identifying item combinations that occur frequently and using them to build rules that predict the occurrence of an item based on the presence of others. In Power BI, this can be implemented by integrating Python or R scripts, allowing users to preprocess data, apply the Apriori algorithm, and visualize the resulting rules for actionable insights.

## a. Implementation Steps in Power BI:

### b. Step 1: Load Data

Open Power BI Desktop.

Use **Get Data** to load the transaction dataset from Excel or CSV.

Click **Transform Data** to open Power Query Editor if needed for preprocessing.

### c. Step 2: Preprocess the Data

Clean column names, ensure items are comma-separated or pivoted properly.

If each row represents one transaction with multiple items:

- d. Split by delimiter and unpivot to get each item per transaction in a separate row.

### e. Step 3: Enable R or Python Scripting (Optional)

Go to **File > Options > R/Python scripting**

Set up R or Python home directory (if needed)

f. **Step 4:** Use R/Python Script to Run Apriori Algorithm

g. **Step 5:** Load the Output

The output rules table can be loaded back to Power BI.

Use visualizations like:

h. Matrix Table for rule display

i. Bar/Column chart for support, confidence, and lift

j. Scatter plot (Lift vs Confidence)

## 6 Results

a. Apriori algorithm was successfully implemented in Power BI using Python/R scripting, and association rules were visualized based on itemset relationships.

7 **Conclusions:** Association Rule Mining is effectively used to extract actionable patterns from transaction datasets. Power BI with Python scripting allows integration of advanced machine learning algorithms like Apriori along with rich data visualization.

8 **Viva Questions:** What is the Apriori Algorithm used for?

1. Define Support, Confidence, and Lift.

## 9 References:

- Kimball Group: Kimball Group's Website offers articles and resources on dimensional modeling and data warehousing.
- "The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling" by Ralph Kimball and Margy Ross
- Building the Data Warehouse" by William H. Inmon
- <https://www.cittros.com/insights/basic-data-transformation-techniques-in-power-bi>

**Data Warehouse and Mining**

**Experiment No.: 9**

**Implementation of**

**Agglomerative/hierarchical Clustering  
method**

# Experiment No. 9

1. **Aim:** Implementation of Agglomerative/Hierarchical Clustering Method using Power BI.
2. **Objectives:** To implement and visualize Agglomerative (Hierarchical) Clustering on a dataset using Power BI with Python.
3. **Course Outcomes:** Students will be able to implement and visualize Agglomerative (Hierarchical) Clustering using Power BI for pattern discovery.
4. **Hardware / Software Required:** Power BI tool.
5. **Theory:** Hierarchical clustering is an unsupervised learning method that builds a hierarchy of clusters either through a bottom-up approach (Agglomerative) or a top-down approach (Divisive). Agglomerative clustering starts with each data point as an individual cluster and successively merges the closest pairs of clusters based on a distance metric, forming a tree-like structure called a dendrogram. It does not require the number of clusters to be specified in advance and is particularly useful for discovering nested data groupings. In Power BI, hierarchical clustering can be implemented using Python scripts, allowing data scientists to perform clustering analysis and visualize clusters using Power BI's rich visual tools.

## **Implementation Steps in Power BI:**

### **Step 1: Load Dataset**

Open Power BI → Click **Get Data** → Load from Excel/CSV.

### **Step 2: Preprocess the Data**

Clean missing values, remove unwanted columns, and normalize if needed.

Click **Transform Data** to clean data in Power Query Editor.

### **Step 3: Enable Python Scripting**

Go to **File > Options > Python scripting**

Set Python home path (e.g., Anaconda environment)

### **Step 4: Run Python Script for Clustering**

Click on **Transform Data > Run Python Script**

## **Step 5: Load Clustered Data into Power BI**

Load the output table with `Cluster` column back into Power BI.

### **Visualizations in Power BI:**

Use **Scatter Plots** to display clusters based on 2D features (e.g., Age vs Income, with Cluster as Legend).

Add **Table Visuals** to show records with their cluster labels.

Include **Dendrogram** as an image if rendered separately using Python.

6. **Results** Agglomerative clustering was successfully performed and visualized using Power BI by integrating Python for cluster generation and dendrogram construction.
7. **Conclusions:** Hierarchical clustering using Power BI enables users to analyze data groupings effectively by integrating Python scripts, and visually interpret cluster structures for informed decision-making.

### **8. Viva Questions:**

- a. What is the difference between agglomerative and divisive clustering?
- b. How does hierarchical clustering work?
- c. What is a dendrogram?
- d. Why do we normalize data before clustering?
- e. How can clustering results be visualized in Power BI?

### **13. References:**

- Kimball Group: Kimball Group's Website offers articles and resources on dimensional modeling and data warehousing.
- "The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling" by Ralph Kimball and Margy Ross
- Building the Data Warehouse" by William H. Inmon
- <https://www.cittros.com/insights/basic-data-transformation-techniques-in-power-bi>