

Power BI Assignment 1 – Data Transformation & Data Modeling

Data Transformation:

Restrict the "List of Orders" table to only the first 500 rows.

The screenshot shows the Power BI Desktop interface with the 'List of Orders' table loaded. The formula bar contains the expression `Table.FirstN(#"Changed Type",500)`. The table view displays columns: Order ID, Order Date, CustomerName, and State. The 'Query Settings' pane on the right shows the 'APPLIED STEPS' list with 'Kept First Rows' selected.

Ensure the "Order Date" column in the "List of Orders" table is set to data type 'Date'.

The screenshot shows the 'List of Orders' table with the 'Order Date' column highlighted. The 'Data Type: Date' dropdown is open, showing the 'Date' option selected. The formula bar contains the expression `Table.TransformColumnTypes(#"Kept First Rows",{{"Order Date", type date}})`. The 'Query Settings' pane on the right shows the 'APPLIED STEPS' list with 'Changed order date type' selected.

Change the data type of "Amount" and "Target" columns to 'Fixed Decimal Number'

Amount :

The screenshot shows the 'List of Orders' table with the 'Amount' and 'Target' columns highlighted. The 'Data Type: Fixed decimal number' dropdown is open, showing the 'Fixed decimal number' option selected. The formula bar contains the expression `Table.TransformColumnTypes(#"Changed Type",{{"Amount", Currency.Type}})`. The 'Query Settings' pane on the right shows the 'APPLIED STEPS' list with 'Changed Amount Datatype' selected.

Target :

The screenshot shows the Power BI Desktop interface with the 'Transform' tab selected. The 'Query Settings' pane on the right shows the 'APPLIED STEPS' list, where 'Changed Target Datatype' is the current step. The main data view displays a table with columns: Month of Order Date, Category, and Target. The 'Target' column is highlighted, and the formula bar shows the transformation: `= Table.TransformColumnTypes(#"Changed Type",{{"Target", Currency.Type}})`.

| Month of Order Date | Category | Target |
|---------------------|-----------|-----------|
| 01-04-2018 | Furniture | 10,400.00 |
| 01-05-2018 | Furniture | 10,500.00 |
| 01-06-2018 | Furniture | 10,600.00 |
| 01-07-2018 | Furniture | 10,800.00 |
| 01-08-2018 | Furniture | 10,900.00 |
| 01-09-2018 | Furniture | 11,000.00 |
| 01-10-2018 | Furniture | 11,100.00 |
| 01-11-2018 | Furniture | 11,300.00 |
| 01-12-2018 | Furniture | 11,400.00 |
| 01-01-2019 | Furniture | 11,500.00 |
| 01-02-2019 | Furniture | 11,600.00 |

Format the "CustomerName" column into proper case, ensuring consistent capitalization for each word.

The screenshot shows the Power BI Desktop interface with the 'Transform' tab selected. The 'Query Settings' pane on the right shows the 'APPLIED STEPS' list, where 'Capitalized CustomerName' is the current step. The main data view displays a table with columns: Order ID, Order Date, CustomerName, State, and City. The 'CustomerName' column is highlighted, and the formula bar shows the transformation: `= Table.TransformColumns(#"Changed order date type",{{"CustomerName", Text.Capitalize}})`.

| Order ID | Order Date | CustomerName | State | City |
|----------|------------|--------------|-------------------|--------------------|
| B-25601 | 01-04-2018 | Bharat | Gujarat | Ahmedabad |
| B-25602 | 01-04-2018 | Pearl | Maharashtra | Pune |
| B-25603 | 03-04-2018 | Jahan | Madhya Pradesh | Indore |
| B-25604 | 03-04-2018 | Divsha | Rajasthan | Jodhpur |
| B-25605 | 05-04-2018 | Kasheen | Karnataka | Bangalore |
| B-25606 | 06-04-2018 | Hazel | Kerala | Thiruvananthapuram |
| B-25607 | 06-04-2018 | Sonakshi | Jammu and Kashmir | Srinagar |
| B-25608 | 08-04-2018 | Aarushi | Tamil Nadu | Chennai |
| B-25609 | 09-04-2018 | Jitesh | Uttar Pradesh | Lucknow |
| B-25610 | 09-04-2018 | Yogesh | Bihar | Patna |
| B-25611 | 11-04-2018 | Anita | Kerala | Thiruvananthapuram |

Merge the "State" and "City" columns to create a new column named "Location" in the format 'City, State'.

The screenshot shows the Power BI Desktop interface with the 'Transform' tab selected. The 'Merge Columns' dialog box is open, showing the 'Separator' as a comma and the 'New column name (optional)' as 'Location'. The main data view displays a table with columns: Order ID, Order Date, CustomerName, State, and City. The 'State' and 'City' columns are highlighted.

| Order ID | Order Date | CustomerName | State | City |
|----------|------------|--------------|---------|-----------|
| B-25601 | 01-04-2018 | Bharat | Gujarat | Ahmedabad |
| B-25602 | | | | |
| B-25603 | | | | |
| B-25604 | | | | |
| B-25605 | | | | |
| B-25606 | | | | |
| B-25607 | | | | |
| B-25608 | | | | |
| B-25609 | | | | |
| B-25610 | | | | |
| B-25611 | | | | |
| B-25612 | | | | |
| B-25613 | | | | |
| B-25614 | | | | |
| B-25615 | | | | |
| B-25616 | | | | |

Create a new custom column named "Profit Margin" as the percentage of "Profit" divided by "Amount".

The screenshot shows the Power Query Editor interface. The 'Add Column' tab is selected. The 'Custom Column' dialog box is open, showing the formula `[Profit]/[Amount]` for the new column named 'Profit Margin'. The 'Available columns' list on the right includes Order ID, Amount, Profit, Quantity, Category, Sub-Category, and Profit Margin. The 'Query Settings' pane on the right shows the 'APPLIED STEPS' list with 'Added Custom' at the bottom.

Add a new conditional column named "Profit Status" based on the values in the "Profit" column. The conditions are as follows: if the profit is less than 0, the label should be "Loss"; if the profit equals 0, the label should be "Break-Even"; and if the profit is greater than 0, the label should be "Profit".

The screenshot shows the Power Query Editor interface. The 'Add Column' tab is selected. The 'Add Conditional Column' dialog box is open, showing the formula `if [Profit] < 0 then "Loss" else if [Profit] = 0 then "Break-Even" else "Profit"` for the new column named 'Profit Status'. The 'Column Name' is 'Profit Status', the 'Operator' is 'is less than', and the 'Value' is 0. The 'Output' is 'Loss'. The 'Query Settings' pane on the right shows the 'APPLIED STEPS' list with 'Added Conditional Column' at the bottom.

Merging Data (Joins):

Merge the "List of Orders" and "Order Details" tables into a new single table named "Orders Data" based on the "Order ID" relationship.

The screenshot shows the Power Query Editor interface. The 'Merge Queries' dialog box is open, showing the 'List of Orders' and 'Order Details' tables being merged on the 'Order ID' column. The resulting 'Orders Data' table is shown in the main view, with columns for Order ID, Order Date, Customer Name, Location, Amount, Profit, and Quantity. The 'Query Settings' pane on the right shows the 'APPLIED STEPS' list with 'Expanded Order Details' at the bottom.

| Order ID | Order Date | Customer Name | Location | Amount | Profit | Quantity |
|----------|------------|---------------|------------------------|----------|--------|----------|
| 1 | 01-04-2018 | Bharat | Ahmedabad, Gujarat | 1,275.00 | -1148 | -1148 |
| 2 | 01-04-2018 | Bharat | Ahmedabad, Gujarat | 66.00 | -12 | -12 |
| 3 | 01-04-2018 | Bharat | Ahmedabad, Gujarat | 8.00 | -2 | -2 |
| 4 | 01-04-2018 | Bharat | Ahmedabad, Gujarat | 80.00 | -56 | -56 |
| 5 | 01-04-2018 | Pearl | Pune, Maharashtra | 168.00 | -111 | -111 |
| 6 | 01-04-2018 | Pearl | Pune, Maharashtra | 424.00 | -272 | -272 |
| 7 | 01-04-2018 | Pearl | Pune, Maharashtra | 2,617.00 | 1151 | 1151 |
| 8 | 01-04-2018 | Pearl | Pune, Maharashtra | 561.00 | 212 | 212 |
| 9 | 01-04-2018 | Pearl | Pune, Maharashtra | 119.00 | -5 | -5 |
| 10 | 03-04-2018 | Jahan | Bhopal, Madhya Pradesh | 1,355.00 | -60 | -60 |
| 11 | 03-04-2018 | Jahan | Bhopal, Madhya Pradesh | 24.00 | -30 | -30 |
| 12 | 03-04-2018 | Jahan | Bhopal, Madhya Pradesh | 193.00 | -166 | -166 |

Handling Missing Data & Duplicate Data:

Identify missing values in the data and determine a strategy to address them. Check for duplicate rows and define a strategy to handle duplicates.

Home Transform Add Column View Tools Help

Query Settings

Properties

Name: Orders Data

Applied Steps

- Source
- Expanded Order Details
- Removed Duplicates

Sorting and Filtering Data:

In the 'Orders Data' table, utilize sorting and filtering techniques on columns like Order Date, State or Category to analyze data based on specific criteria:

◆ Sort the orders by Order Date in descending order to analyze recent trends.

Home Transform Add Column View Tools Help

Query Settings

Properties

Name: Orders Data

Applied Steps

- Source
- Expanded Order Details
- Removed Duplicates
- Sorted Order Date in Descending Order

◆ Filter the orders to focus only on a specific state (e.g., Tamil Nadu) for regional analysis.

Home Transform Add Column View Tools Help

Query Settings

Properties

Name: Orders Data

Applied Steps

- Source
- Expanded Order Details
- Removed Duplicates
- Sorted Order Date in Descending Order
- Filtered Rows

Grouping and Aggregating Data:

Duplicate the “Order Details” table and calculate the count of each Order ID, average profit by Category or total amount by Sub-Category.

Count of each Order ID

Home

Transform

Add Column

View

Tools

Help

Group By

Use First Row as Headers

Count Rows

Transpose

Reverse Rows

Count Rows

Data Type: Whole Number

Detect Data Type

Rename

Replace Values

Fill

Pivot Column

Unpivot Columns

Move

Convert to List

Queries [5]

List of Orders

Order Details

Sales target

Orders Data

Order Details (2)

✕

✓

fx

= Table.RenameColumns(#"Grouped Rows",{{"Count", "Order_Count"}})

| | AB _C Order ID | 1 ₂ ₃ Order_Count |
|----|--------------------------|---|
| 1 | B-25601 | 4 |
| 2 | B-25602 | 5 |
| 3 | B-25603 | 8 |
| 4 | B-25604 | 2 |
| 5 | B-25605 | 1 |
| 6 | B-25606 | 1 |
| 7 | B-25607 | 1 |
| 8 | B-25608 | 4 |
| 9 | B-25609 | 2 |
| 10 | B-25610 | 6 |
| 11 | B-25611 | 1 |
| 12 | B-25612 | 1 |
| 13 | B-25613 | 1 |
| 14 | B-25614 | 2 |
| 15 | B-25615 | 1 |
| 16 | B-25616 | 4 |
| 17 | B-25617 | 1 |
| 18 | B-25618 | 2 |
| 19 | B-25619 | 1 |
| 20 | B-25620 | 1 |
| 21 | B-25621 | 3 |

Average profit by each category

Home **Transform** Add Column View Tools Help

Group By as Headers Count Rows Table

Transpose Reverse Rows

Data Type: Text Replace Values Unpivot Columns

Detect Data Type Fill Move

Rename Pivot Column Convert to List

Split Column Format Merge Columns

Extract Parse

Text Column

Queries [5]

List of Orders

Order Details

Sales target

Orders Data

Order Details (2)

fx = Table.SelectRows("#Grouped Rows", each true)

| | Category | 1.2 Average_Profit |
|---|-------------|--------------------|
| 1 | Furniture | 9.456790123 |
| 2 | Clothing | 11.76290832 |
| 3 | Electronics | 34.07142857 |

Total amount by sub-category

Home **Transform** Add Column View Tools Help

Group By as Headers Count Rows Table

Transpose Reverse Rows

Data Type: Decimal Number Replace Values Unpivot Columns

Detect Data Type Fill Move

Rename Pivot Column Convert to List

Split Column Format Merge Columns

Extract Parse

Statistics Standard

Nur

Queries [5]

List of Orders

Order Details

Sales target

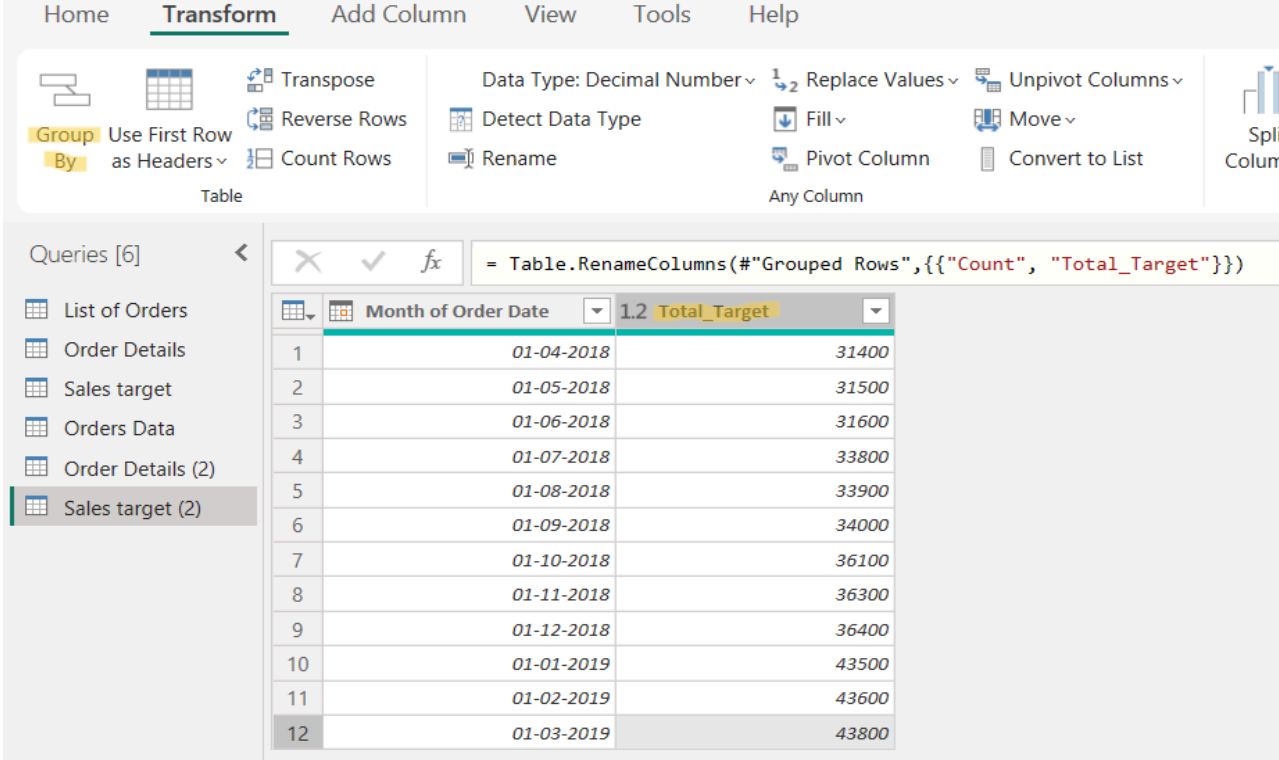
Orders Data

Order Details (2)

fx = Table.Group("#Filtered Rows", {"Sub-Category"}, {"Total_Amount", each List.Sum([Amount]), type nullable number})

| | Sub-Category | 1.2 Total_Amount |
|----|------------------|------------------|
| 1 | Bookcases | 56861 |
| 2 | Stole | 18546 |
| 3 | Hankerchief | 14608 |
| 4 | Electronic Games | 39168 |
| 5 | Phones | 46119 |
| 6 | Saree | 53511 |
| 7 | Trousers | 30039 |
| 8 | Chairs | 34222 |
| 9 | Kurti | 3361 |
| 10 | T-shirt | 7382 |
| 11 | Shirt | 7555 |
| 12 | Leggings | 2106 |
| 13 | Tables | 22614 |
| 14 | Printers | 58252 |
| 15 | Accessories | 21728 |
| 16 | Furnishings | 13484 |
| 17 | Skirt | 1946 |

Duplicate the “Sales Target” table and aggregate the total target amount by Month of Order Date.

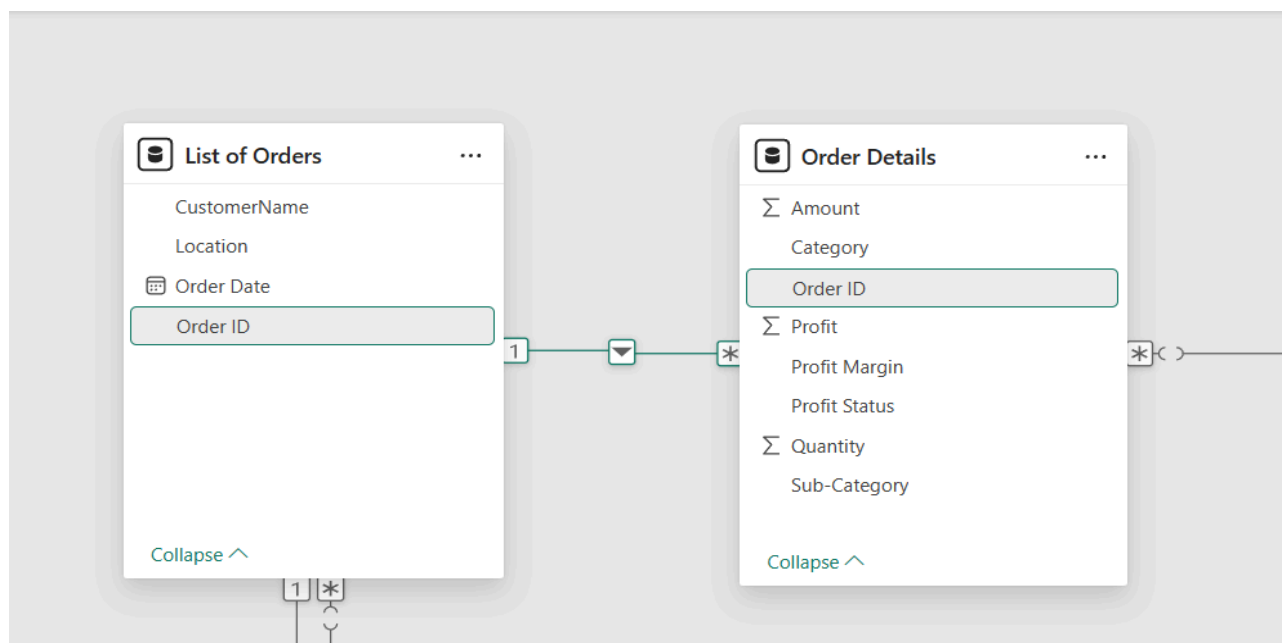


The screenshot shows the Power BI Transform ribbon with the 'Group By' operation applied. The formula bar displays the DAX formula: `= Table.RenameColumns(#"Grouped Rows",{{"Count", "Total_Target"}})`. The resulting table is as follows:

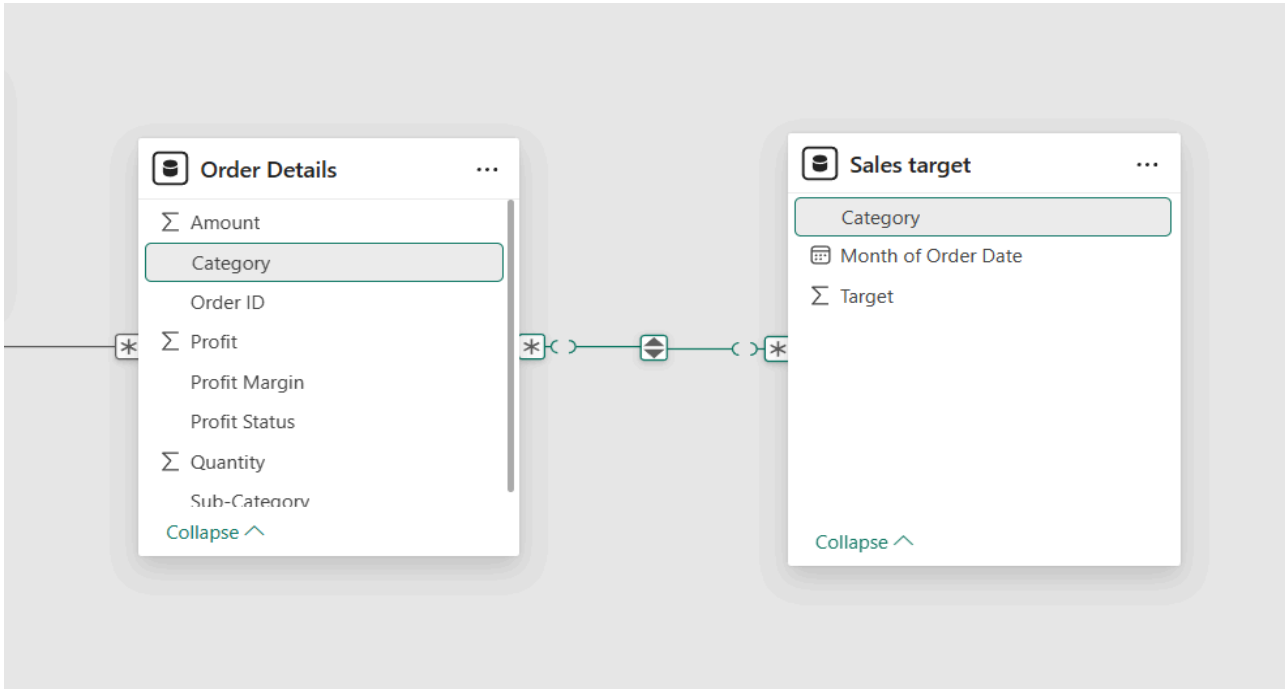
| | Month of Order Date | 1.2 Total_Target |
|----|---------------------|------------------|
| 1 | 01-04-2018 | 31400 |
| 2 | 01-05-2018 | 31500 |
| 3 | 01-06-2018 | 31600 |
| 4 | 01-07-2018 | 33800 |
| 5 | 01-08-2018 | 33900 |
| 6 | 01-09-2018 | 34000 |
| 7 | 01-10-2018 | 36100 |
| 8 | 01-11-2018 | 36300 |
| 9 | 01-12-2018 | 36400 |
| 10 | 01-01-2019 | 43500 |
| 11 | 01-02-2019 | 43600 |
| 12 | 01-03-2019 | 43800 |

Data Modeling:

Establish a relationship between the “List of Orders” and “Order Details” tables using the ‘Order ID’ column.



Build a relationship between the “Order Details” and “Sales Target” tables based on the ‘Category’ column.



Click "Manage relationships" and ensure this relationship is active.

Manage relationships ✕

+ New relationship

⚡ Autodetect

✎ Edit

🗑 Delete

⌵ Filter

| <input type="checkbox"/> From: table (column) ↑ | Relationship | To: table (column) | Status |
|---|--------------|-------------------------------|------------|
| <input type="checkbox"/> Order Details (Category) | | Sales target (Category) | Active ... |
| <input type="checkbox"/> Order Details (Order ID) | | List of Orders (Order ID) | Active ... |
| <input type="checkbox"/> Orders Data (CustomerName) | | List of Orders (CustomerName) | Active ... |
| <input type="checkbox"/> Orders Data (Order ID) | | List of Orders (Order ID) | Active ... |

