**How to manage aggregations step-by-step**

**Introduction**

In real-world data analytics, optimizing performance is of key importance, especially when dealing with large datasets. In the context of Power BI, aggregations are pre-calculated summaries of data that allow for quicker query execution and faster rendering of reports. When dealing with millions of rows of datasets, generating reports and dashboards in real-time can be resource-intensive and time-consuming. Aggregations solve this challenge by pre-computing common calculations and sorting them for quick retrieval, thereby significantly improving query performance.

In this activity, you must apply your knowledge to manage aggregations within a data model.

* You'll walk through the steps to manage aggregations in a data model using Power BI.
* The goal is to understand how Power BI facilitates aggregation management.

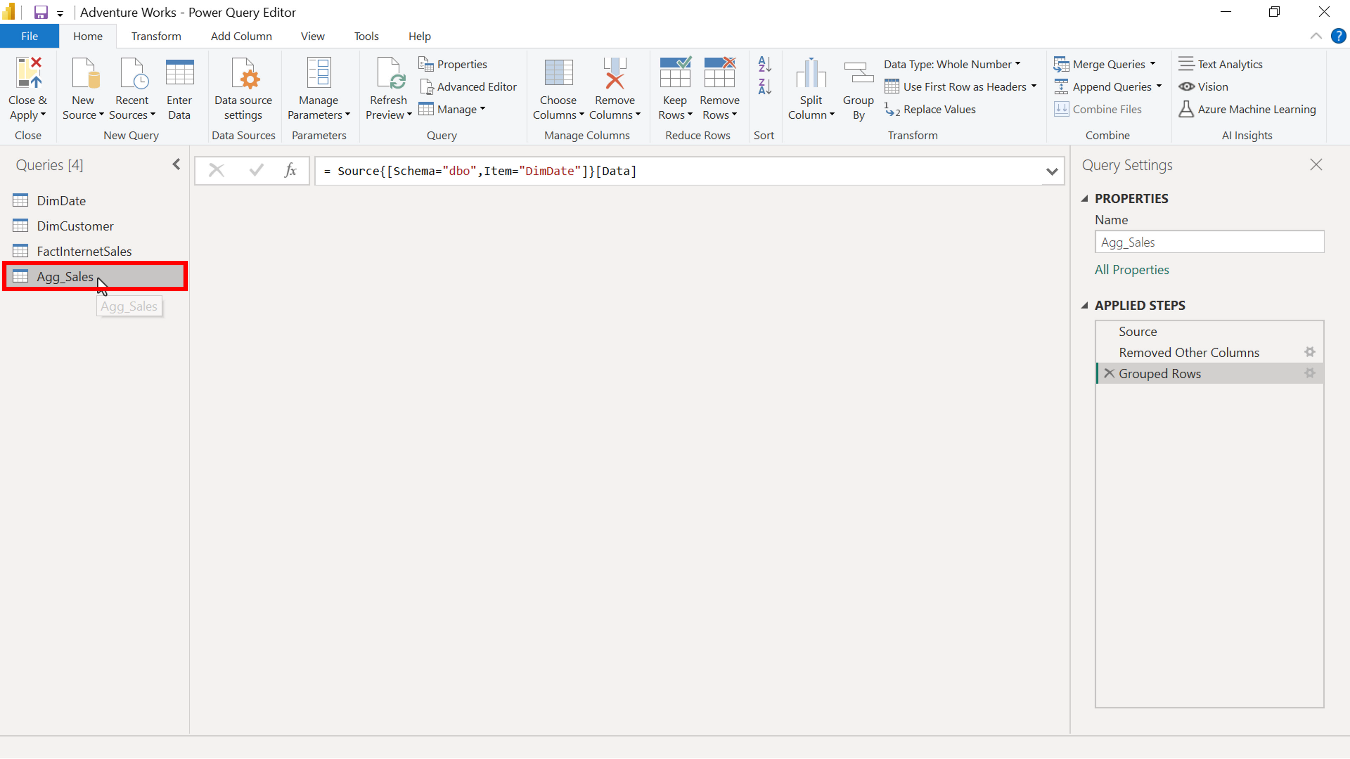
**Case study**

Adventure Works needs to use a data model with many different data types. However, Power BI is unaware of any aggregation. To address this, it is imperative to let Power BI know about the presence of aggregation. That is why it is necessary to configure and manage aggregations. Let’s walk through Power BI to configure and manage the aggregation we have created previously to use this in our reporting and visualization.

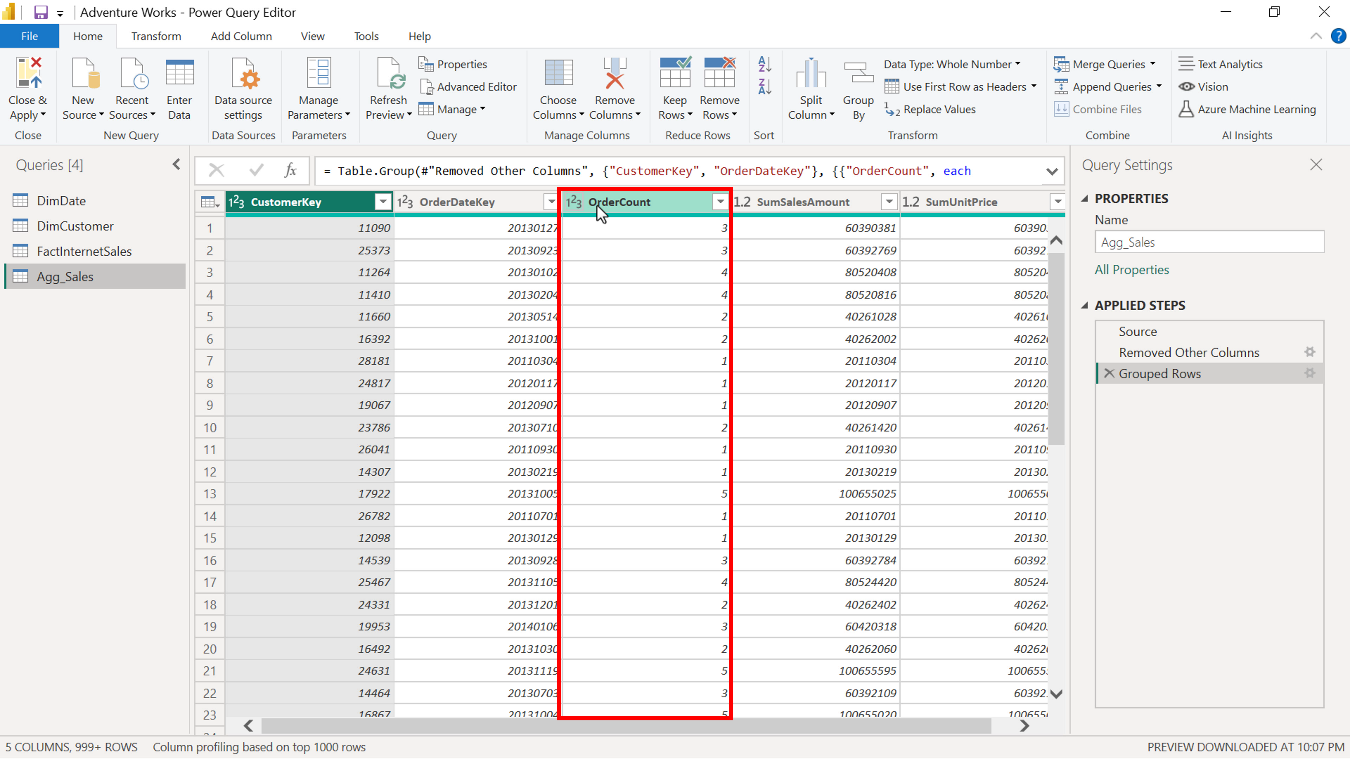
**Step 1: Match the data type of aggregate columns to source columns**

Before managing the aggregation in Power BI, but after configuring the storage mode, you’ll need to go back to the query editor and change the data type of aggregated columns.

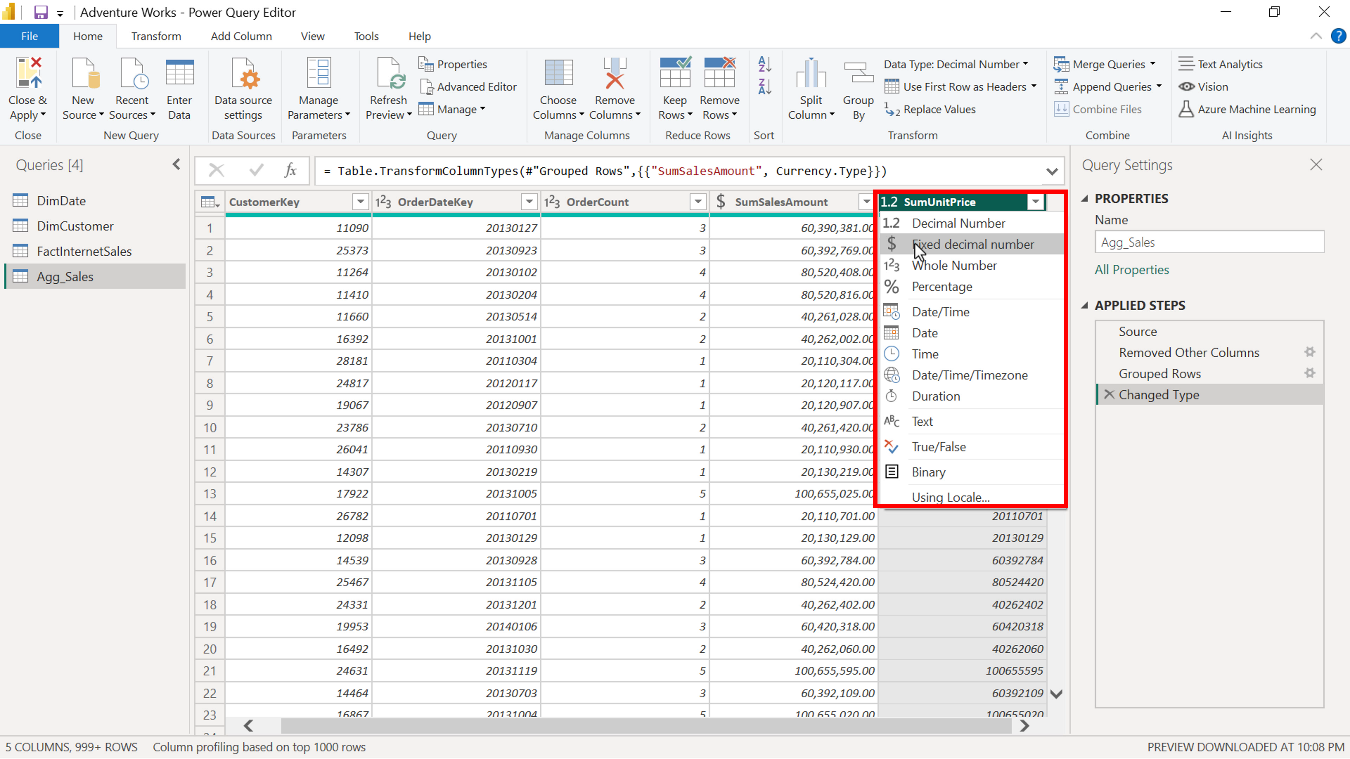
1. Navigate to the query editor and select the **Agg\_Sales** table from the **Queries** pane on the left side of the query editor.



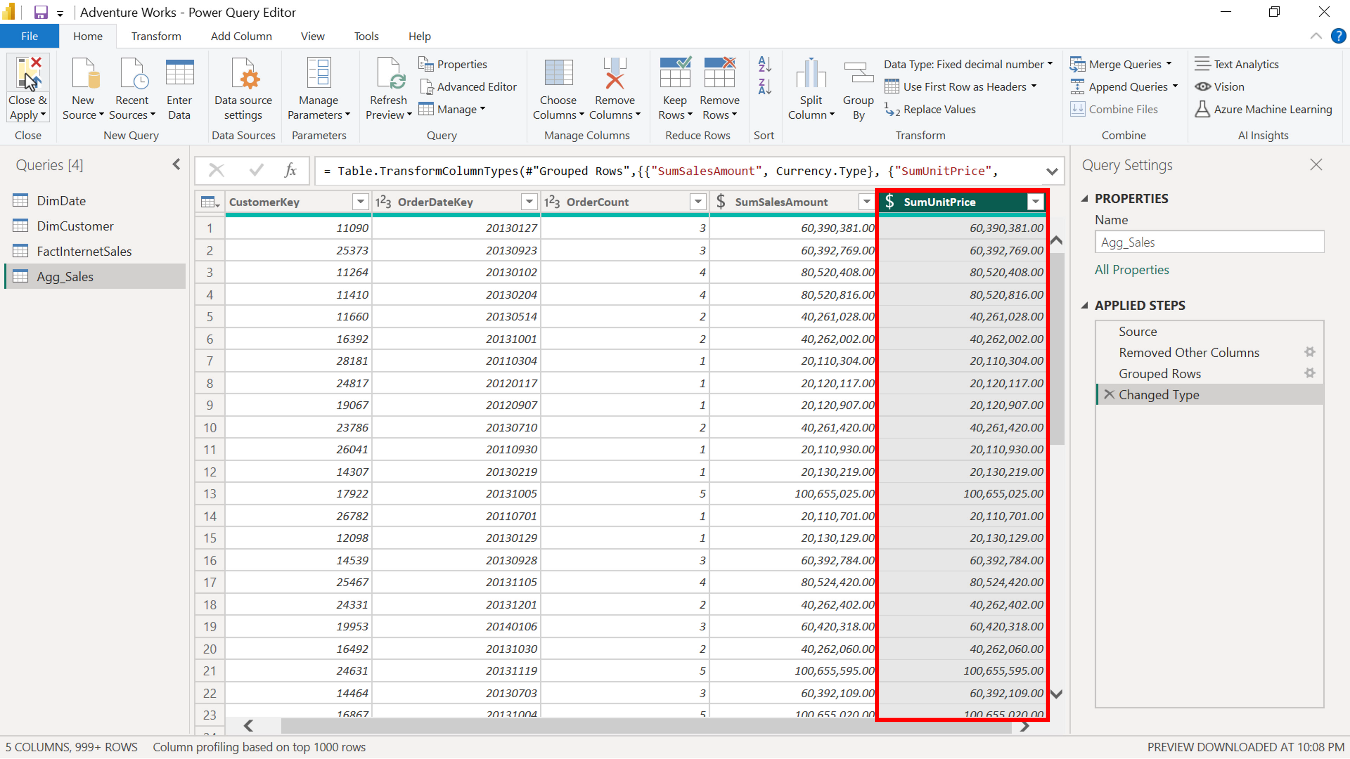
1. Select the **OrderCount** column and ensure the data types are set to **Whole number**. Any column that uses the **Count** function as an operation must have a **Whole number** data type.



1. Select the **SumSalesAmount** and **SumUnitPrice** columns and change the data type to a fixed decimal number, because the data type of the **Source** column is fixed decimal, and you need to match the data type.



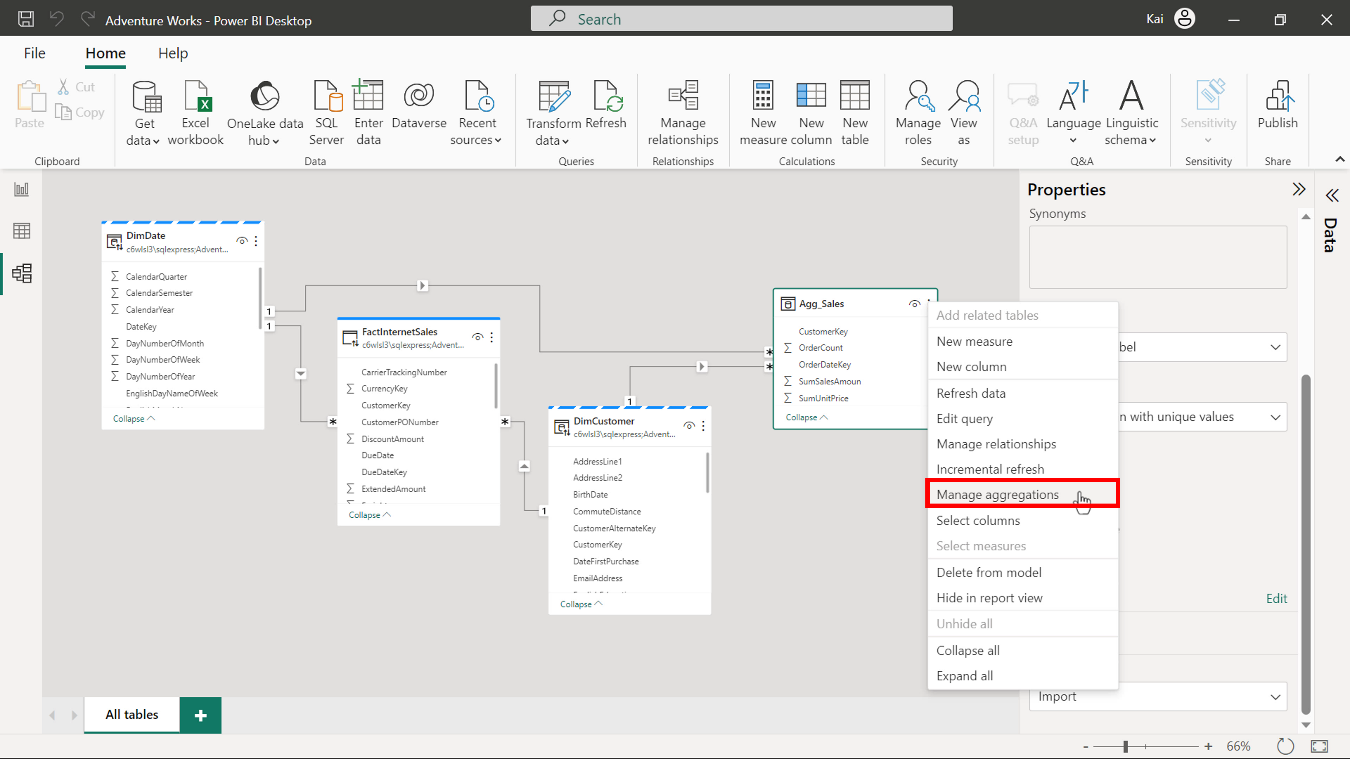
Once completed, check that the data is fixed decimal numbers. Then, close the query editor and return to the Power BI desktop interface.



**Step 2: Manage aggregations**

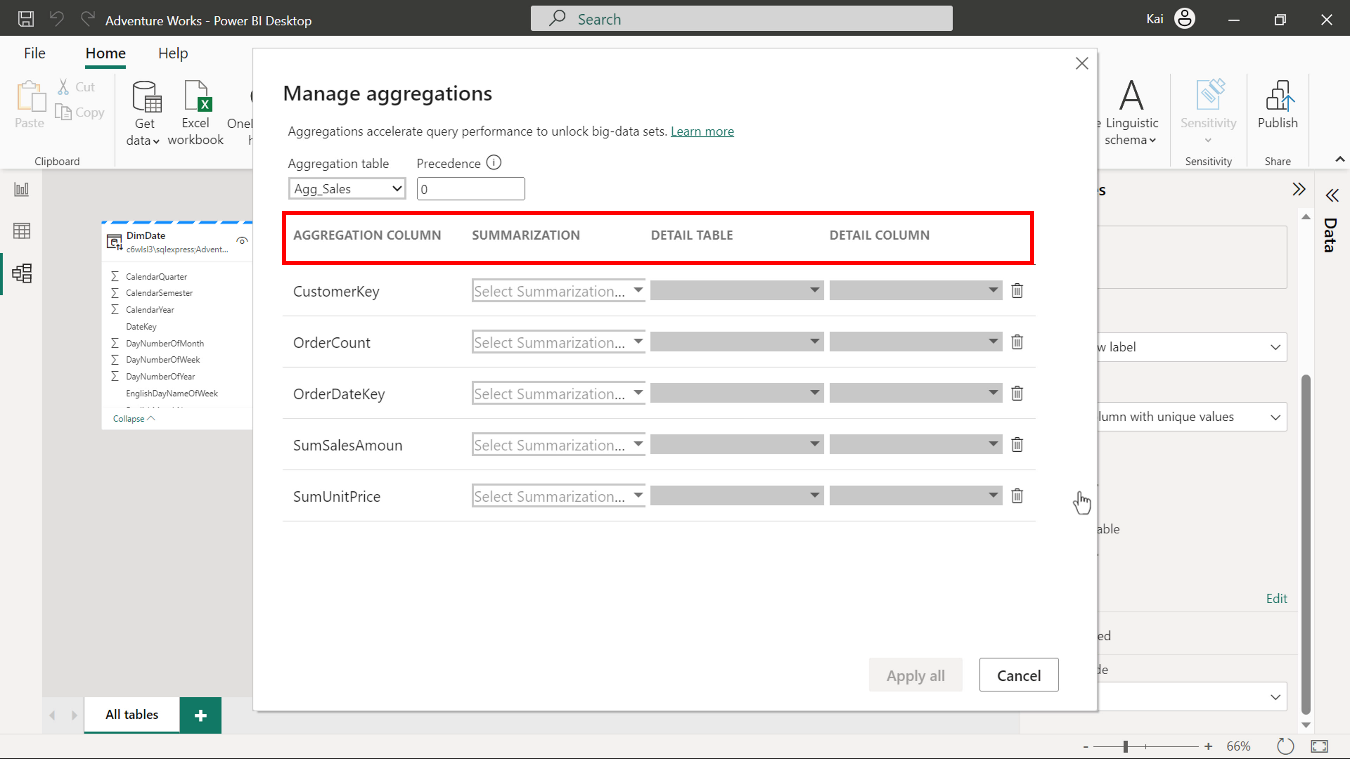
Now, you are ready to manage aggregations.

1. To navigate to the **Manage aggregation** window, select and right-click on the **Agg\_Sales** table from the data pane in the **Model view** of Power BI desktop. Select **Manage aggregations** from the drop-down menu.



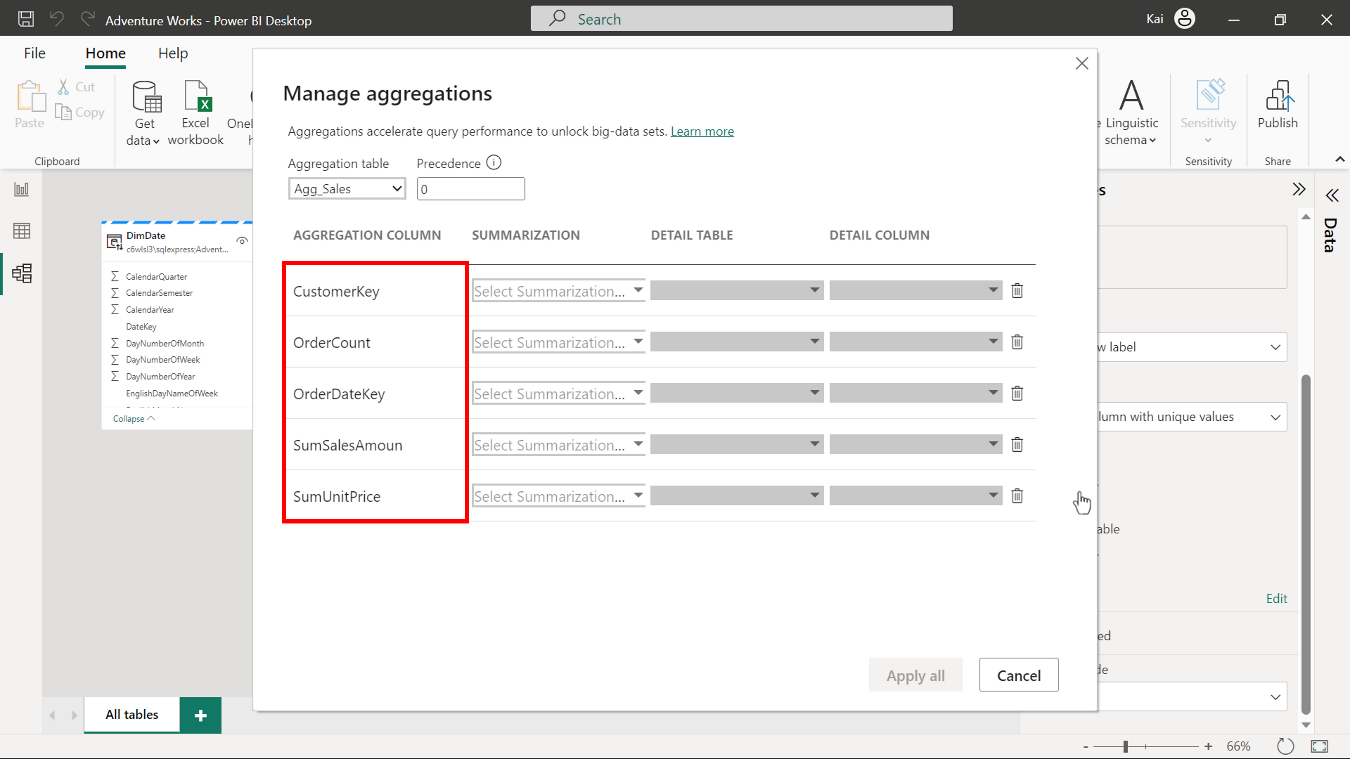
1. The **Manage aggregations** dialog box opens. The aggregation window has four sections:

* **Aggregation columns**
* **Summarization**
* **Detail table**
* And **Detail column**

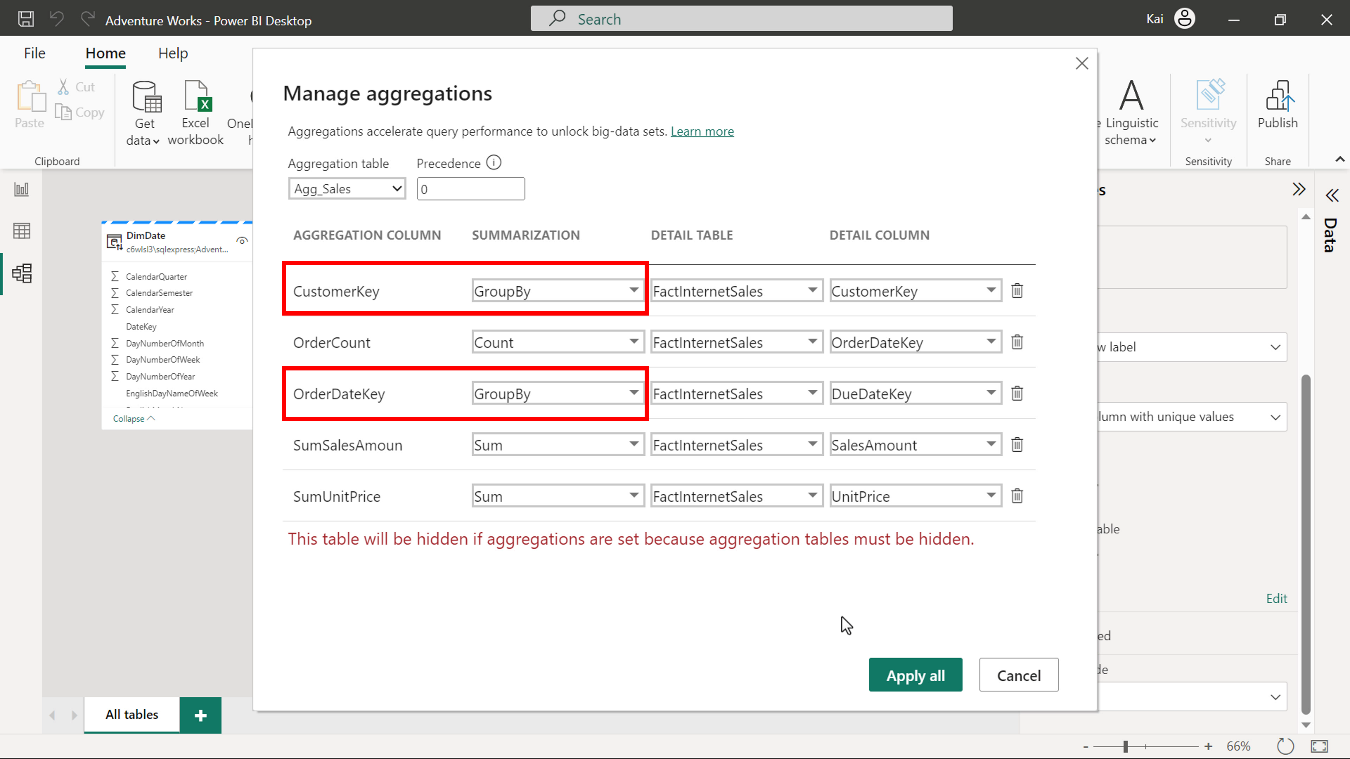


1. Check that the aggregation configuration matches the aggregated table you created during the **Group by** step in Power query editor. In the **Aggregation columns** section, the column names should be as follows:

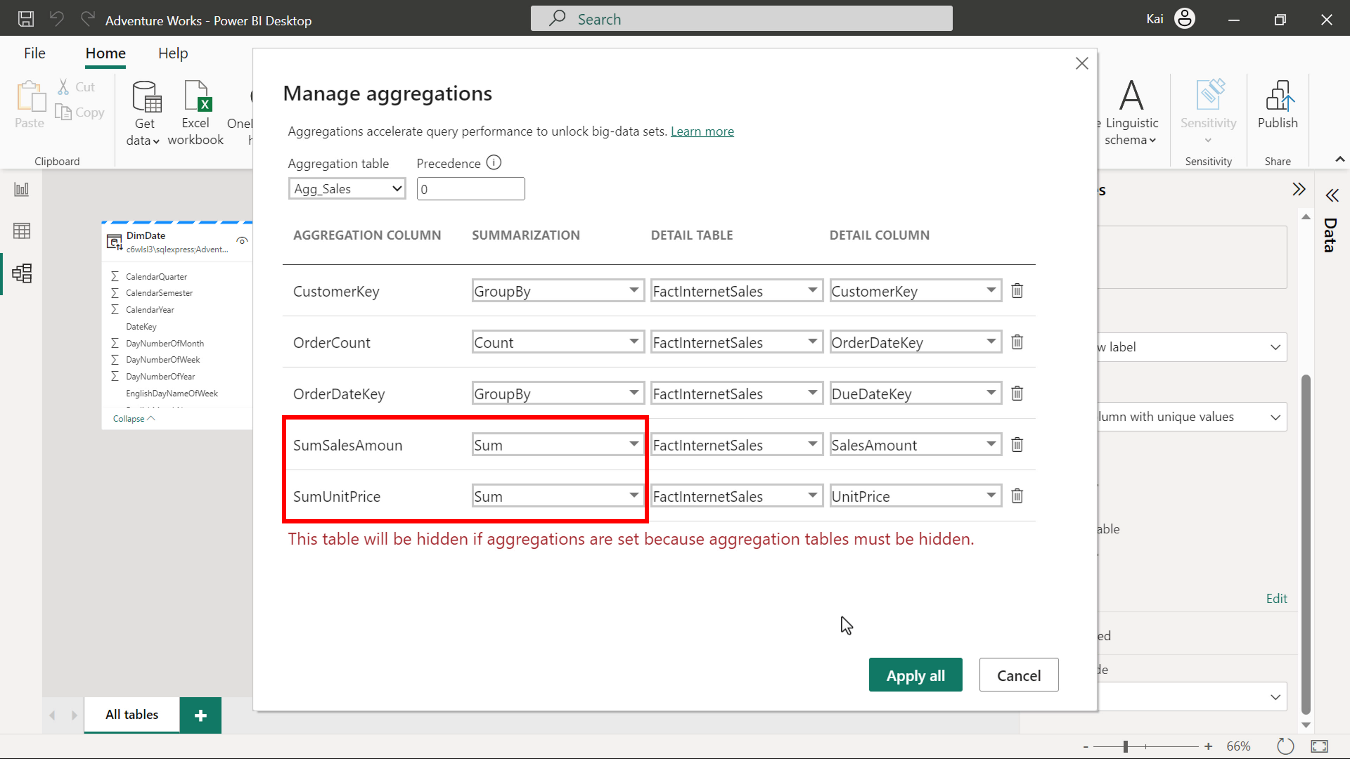
* **CustomerKey,**
* **OrderDateKey,**
* **OrderCount,**
* **SumSalesAmount**,
* and **SumUnitPrice.**



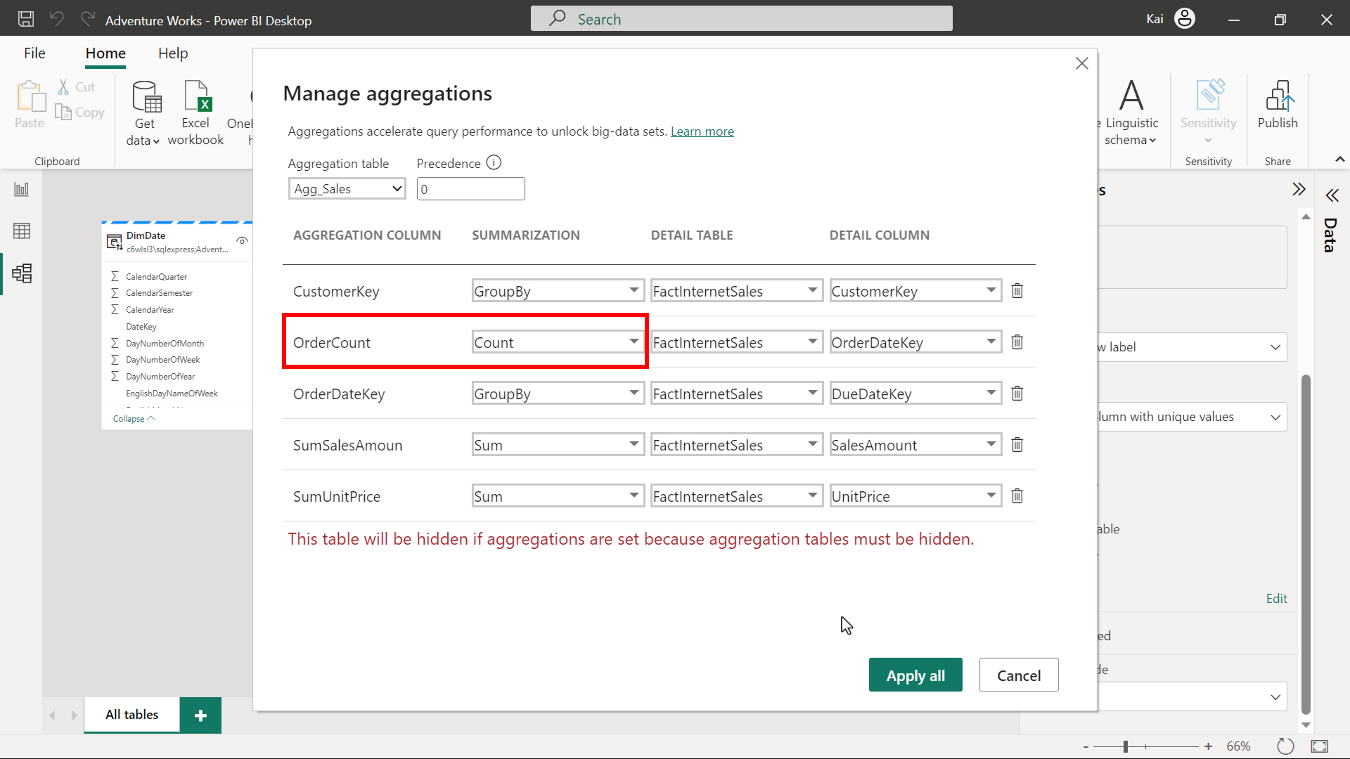
1. In the **Summarization** section, mark the **OrderDateKey** and **CustomerKey** fields as **Group by**, since they’re used to group data.



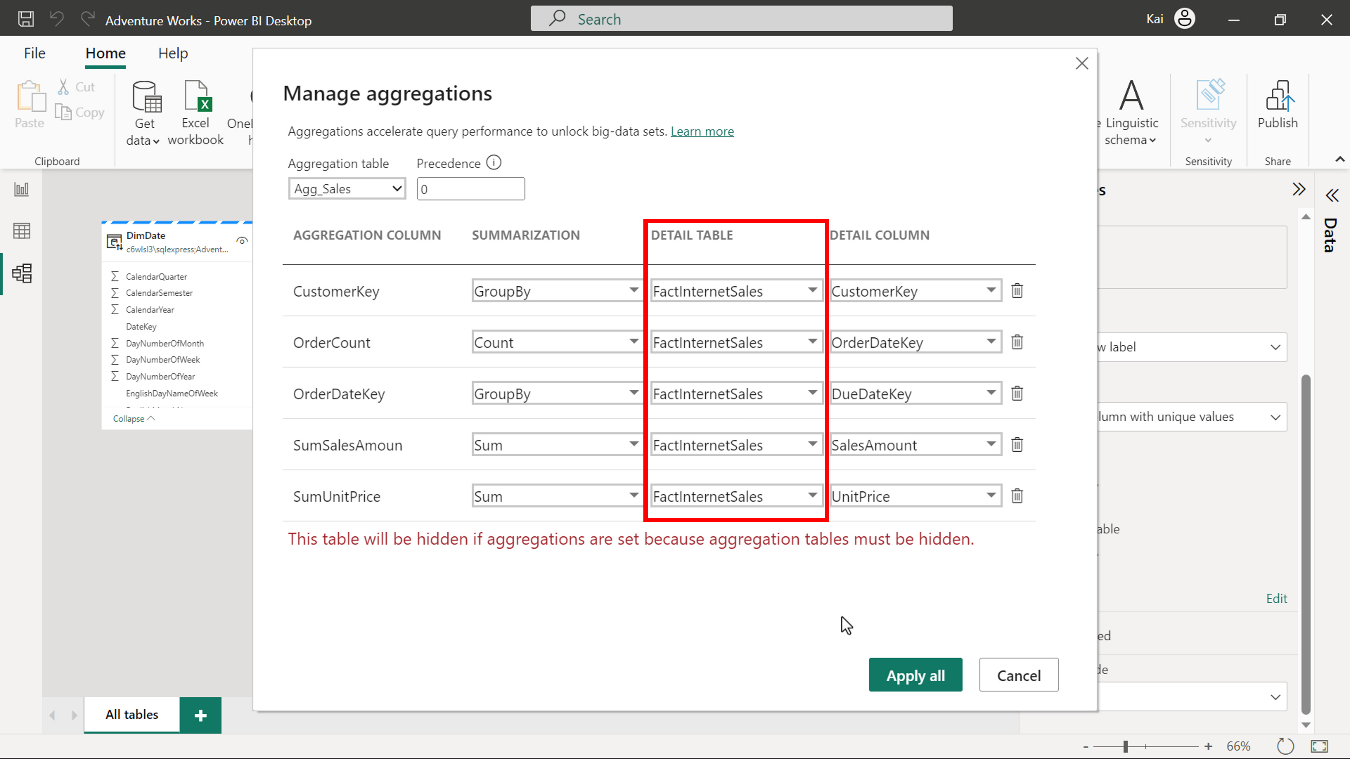
1. **SumSalesAmount** and **SumUnitPrice** are defined to aggregate values from the source table's **Total sales** and **Unit price** columns. Select **Sum** function from the drop-down list, to sum up the values.



1. Select the **Count** function for the **OrderCount** column from the drop-down list of options to define this column so that it counts the total number of orders.

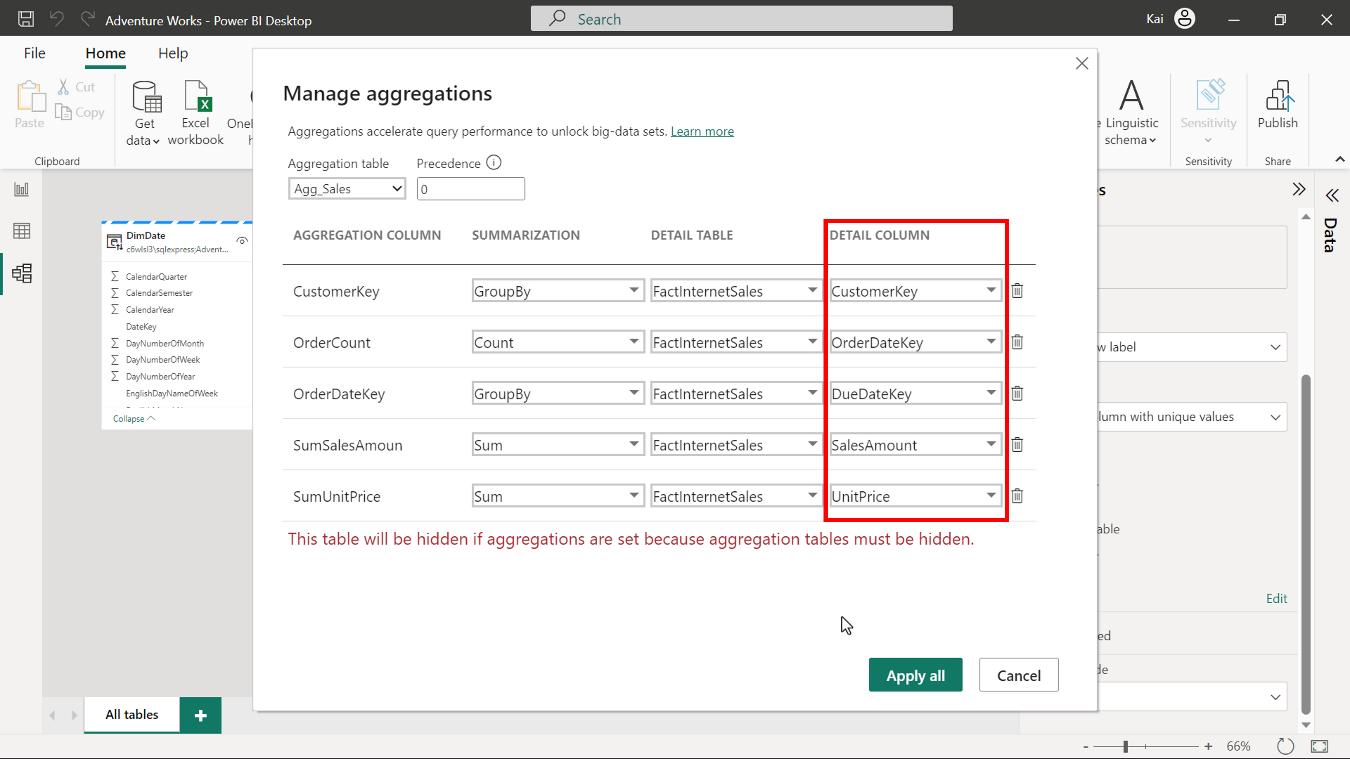


1. In the **Detail table** section, select the **FactInternetSales** table for all columns.

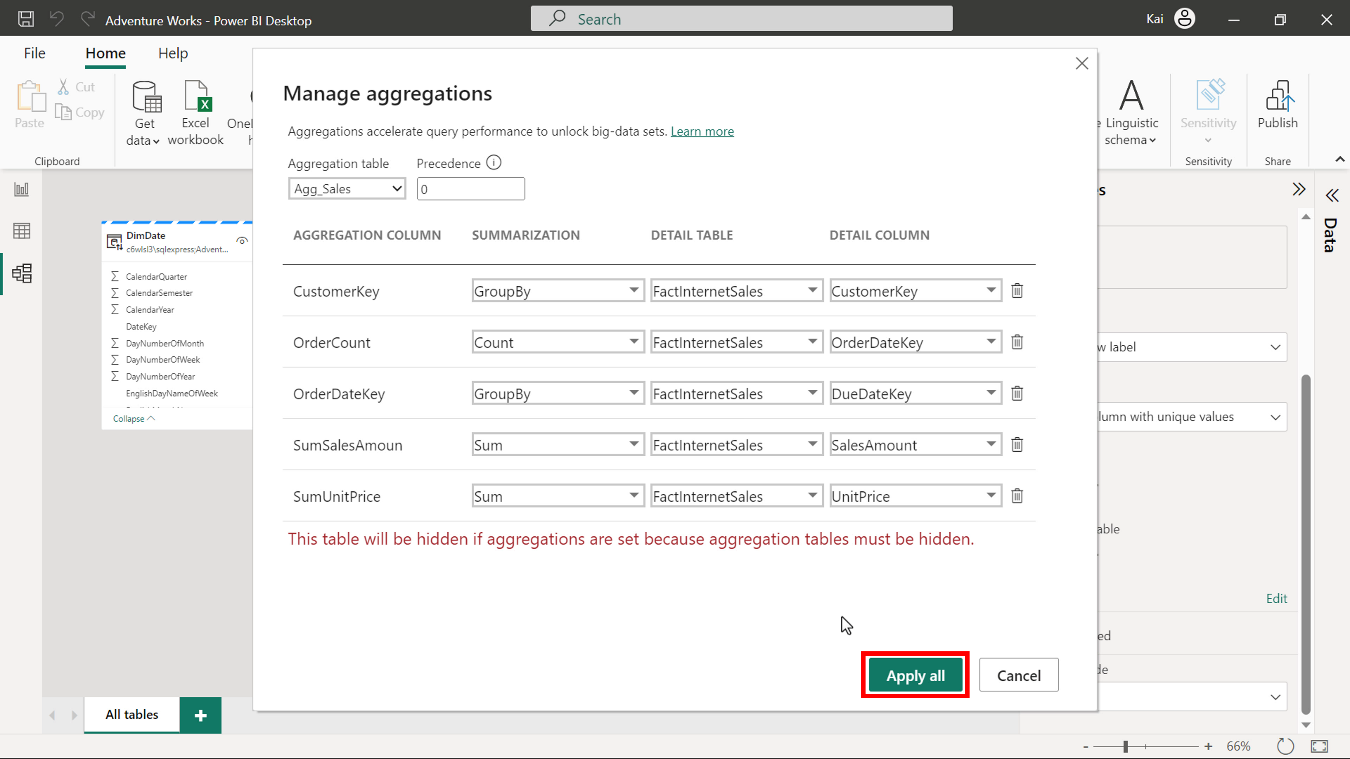


1. Finally, in the **Detail columns** section of the **Manage aggregation** window, map the corresponding column from the **Detail column** drop-down list to the aggregation columns as follows:

* **OrderDateKey** to **OrderDateKey**
* **CustomerKey** to **CustomerKey**
* **OrderCount** to **OrderDateKey**
* **SumSalesAmount** to **SalesAmount**
* **SumUnitPrice** to **UnitPrice**



1. Select **Apply all**. The aggregation tables are automatically hidden from the **Report view** in Power BI because users cannot query aggregation tables (this is to avoid security concerns.)



**Conclusion**

You have now created and managed aggregations that you can use to create visualizations and reports without sending queries to a larger Fact table. This is a valuable skill you’ll use frequently as a data analyst to enhance the performance of your data models.

Go to next item