**DAX Explanation in PowerBi**

1. **Processing Time**

**DAX:**

ProcessingTime = AVERAGEX(Orders, DATEDIFF(MIN(Orders[order\_date]), MIN(Orders[shipped\_date]), DAY))

**Purpose:** This formula calculates the average processing time for orders in a Power BI model. The processing time is defined as the difference between the order date and the shipped date, measured in days.

**Breakdown:**

1. **AVERAGEX(Orders, ...):**
   * This function calculates the average of an expression evaluated for each row in the specified table. In this case, the Orders table is used.
2. **DATEDIFF(MIN(Orders[order\_date]), MIN(Orders[shipped\_date]), DAY):**
   * This nested expression calculates the difference between the earliest order date and the earliest shipped date for all orders.
   * MIN(Orders[order\_date]) finds the minimum (earliest) order date among all orders.
   * MIN(Orders[shipped\_date]) finds the minimum (earliest) shipped date among all orders.
   * DATEDIFF(DAY, ...) calculates the difference between the two dates in days.

**Overall Calculation:**

* For each order, the formula calculates the difference between its order date and shipped date.
* The AVERAGEX function then calculates the average of these differences across all orders, providing the average processing time.

1. **Sales With Discount**

**DAX:**

Sales With Discount = SUMX(Order\_Items, Order\_Items[Quantity] \* Order\_Items[List\_Price] \* (1 - Order\_Items[Discount]))

**Purpose:** This formula calculates the total sales revenue for all order items, taking into account any discounts applied.

**Breakdown:**

1. **SUMX(Order\_Items, ...):**
   * This function calculates the sum of an expression evaluated for each row in the specified table. In this case, the Order\_Items table is used.
2. **Order\_Items[Quantity] \* Order\_Items[List\_Price] \* (1 - Order\_Items[Discount]):**
   * This expression calculates the sales revenue for each order item, considering the quantity, list price, and discount.
   * Order\_Items[Quantity] \* Order\_Items[List\_Price] calculates the total price before any discount.
   * (1 - Order\_Items[Discount]) calculates the discount factor, which is 1 minus the discount percentage. Multiplying the total price by this factor applies the discount.

**Overall Calculation:**

* For each order item, the formula calculates the discounted sales revenue by multiplying the quantity, list price, and discount factor.
* The SUMX function then sums up these discounted sales revenues for all order items to get the total sales with discounts.

1. **Sales by staff**

**DAX**: SalesByStaff = SUMX(Order\_Items, Order\_Items[quantity] \* Order\_Items[list\_price])

**Purpose:** This formula calculates the total sales for each staff member in a Power BI model.

**Breakdown:**

1. **SUMX(Order\_Items, ...):**
   * This function calculates the sum of an expression evaluated for each row in the specified table. In this case, the Order\_Items table is used.
2. **Order\_Items[quantity] \* Order\_Items[list\_price]:**
   * This expression calculates the total sales for each order item by multiplying the quantity and list price.

**Overall Calculation:**

* For each order item, the formula calculates the total sales based on its quantity and list price.
* The SUMX function then sums up these total sales for all order items associated with a particular staff member to get the total sales for that staff.

1. **Total Sales**

**DAX**: TotalSales = SUMX(Order\_Items, Order\_Items[quantity] \* Order\_Items[list\_price])

**Purpose:** This formula calculates the total sales revenue for all order items in a Power BI model.

**Breakdown:**

1. **SUMX(Order\_Items, ...):**
   * This function calculates the sum of an expression evaluated for each row in the specified table. In this case, the Order\_Items table is used.
2. **Order\_Items[quantity] \* Order\_Items[list\_price]:**
   * This expression calculates the total sales for each order item by multiplying the quantity and list price.

**Overall Calculation:**

* For each order item, the formula calculates the total sales based on its quantity and list price.
* The SUMX function then sums up these total sales for all order items to get the total sales revenue.