AdventureWorks BI Workflow Report

Scope: Data model design, relationships, key DAX measures, and calculated columns used to build the Executive, Map, Product Detail, Customer Detail, and Decomposition Tree dashboards.

---

1) Workflow (from raw data to dashboard)

1. **Ingest & Staging**
    - Sales, Products, Customers, Territories, Returns → imported/linked.
    - Created a dedicated **Calendar** table.
2. **Modeling**
    - Built a **star schema** with `Sales Data` and `Returns Data` as facts.
    - Conformed dimensions: Calendar, Product, Product Subcategory, Product Category, Customer, Territory.
    - Added **disconnected tables** for *Price Adjustment* and *Metric Selections*.
3. **Business Logic**
    - Created **calculated columns** (keys, names, flags).
    - Authored **measures** for core KPIs, period comps, rolling windows, targets/variances, and “what-if” pricing.
4. **Visualization**
    - KPI cards, trend lines, decomposition tree, geo map, top-N tables, category bars, target gauges.

---

2) Data Model (star schema)

Fact Tables

- **Sales Data**
  Keys: `CustomerKey`, `ProductKey`, `TerritoryKey`, `OrderDate`
  Metrics: `OrderQuantity`, `OrderLineItem`, `ProductCost`, `ProductPrice`.

- **Returns Data**
  Keys: `ProductKey`, `TerritoryKey`, `ReturnDate`
  Metrics: `ReturnQuantity`.

Dimension Tables

- **Calendar Lookup** – `Date` (active to Sales[OrderDate]), derived fields: `DayName`, `Month`, `Month Name`, `Month Num`, `Start of Month`, `Start of Quarter`, `Weekend`.

- **Product Lookup** – `ProductKey`, `ProductName`, `ProductColor`, `ModelName`, `ProductCost`, `ProductPrice`, `DiscountedPrice`.

- **Product Subcategories Lookup** – `ProductSubcategoryKey`, `SubcategoryName`.

- **Product Categories Lookup** – `ProductCategoryKey`, `CategoryName`.

- **Customer Lookup** – `CustomerKey`, `FullName`, `AnnualIncome`, `Occupation`, `Education`, etc.

- **Territory Lookup** – `TerritoryKey`, `Country`, `Region`, `Continent`.

Slicer / Helper (Disconnected) Tables

- **Measure Table** – housing all measures.

- **Price Adjustment** – parameter table for what-if analysis (`Price Adjustment`, `Price Adjustment Value`).

- **Product Metric Selection** – allows switching metrics (Orders / Revenue / Profit / Returns / Return%).

- **Customer Metric Selection** – toggles customer KPIs (e.g., Count / RPC).

Relationship Summary

- `Calendar[Date]` → _(1: active)*_ → `Sales Data[OrderDate]`

- `Calendar[Date]` → _(1: inactive)*_ → `Returns Data[ReturnDate]` _(activated in measures with USERELATIONSHIP)_

- `Product Lookup[ProductKey]` → **(1:*)** → `Sales Data[ProductKey]`, `Returns Data[ProductKey]`

- `Product Subcategories Lookup[ProductSubcategoryKey]` → **(1:*)** → `Product Lookup[ProductSubcategoryKey]`

- `Product Categories Lookup[ProductCategoryKey]` → **(1:*)** → `Product Subcategories Lookup[ProductCategoryKey]`

- `Customer Lookup[CustomerKey]` → **(1:*)** → `Sales Data[CustomerKey]`

- `Territory Lookup[TerritoryKey]` → **(1:*)** → `Sales Data[TerritoryKey]`, `Returns Data[TerritoryKey]`

- **Disconnected:** Price Adjustment, Metric Selection tables (no physical relationships).

---

3) Calculated Columns (examples)

```
-- Calendar: Weekend flag
Weekend = IF(WEEKDAY('Calendar Lookup'[Date],2) > 5, TRUE(), FALSE())

-- Customer: Full name
Full Name = 'Customer Lookup'[FirstName] & " " & 'Customer
Lookup'[LastName]
```

```
-- Product: Retail price (if not present)
Retail Price = COALESCE('Product Lookup'[ProductPrice],'Product
Lookup'[DiscountedPrice])
```

(Note: Most business logic stays in measures; columns are light and semantic.)

---

4) Core Measures (from Measure Table)

4.1 Base KPIs

```
Total Orders =
SUM('Sales Data'[OrderQuantity])

Total Revenue =
SUMX('Sales Data', 'Sales Data'[OrderQuantity] * 'Sales
Data'[ProductPrice])

Total Cost =
SUMX('Sales Data', 'Sales Data'[OrderQuantity] * 'Sales
Data'[ProductCost])

Total Profit =
[Total Revenue] - [Total Cost]

Quantity Returned =
CALCULATE(SUM('Returns Data'[ReturnQuantity]),
         USERELATIONSHIP('Calendar Lookup'[Date],'Returns
Data'[ReturnDate]))

Total Returns = [Quantity Returned]

Return Rate =
DIVIDE([Total Returns], [Total Orders])
```

4.2 Customer KPIs

```
Total Customers = DISTINCTCOUNT('Customer Lookup'[CustomerKey])
```

```
AvgRevenuePerCustomer =
DIVIDE([Total Revenue], [Total Customers])
```

4.3 Product-specific (bikes & accessories)

```
Bike Sales =
CALCULATE([Total Revenue], 'Product Categories Lookup'[CategoryName] =
"Bikes")

Bikes Returned =
CALCULATE([Total Returns], 'Product Categories Lookup'[CategoryName] =
"Bikes")

Bike Return rate = DIVIDE([Bikes Returned], CALCULATE([Total Orders],
'Product Categories Lookup'[CategoryName] = "Bikes"))
```

4.4 “What-If” Pricing (disconnected Price Adjustment table)

```
-- Parameter created via What-if: Price Adjustment Value (e.g.,
-10%..+10%)
Adjusted Price =
VAR Factor = 1 + 'Price Adjustment'[Price Adjustment Value]
RETURN AVERAGE('Product Lookup'[ProductPrice]) * Factor

Adjusted Revenue =
SUMX('Sales Data',
     'Sales Data'[OrderQuantity] * (RELATED('Product
Lookup'[ProductPrice]) * (1 + SELECTEDVALUE('Price Adjustment'[Price
Adjustment Value],0)))
)

Adjusted Profit =
[Adjusted Revenue] - [Total Cost]
```

4.5 Targets & Gaps (for gauges and cards)

```
Order Target = 213         -- or from a target table
Order Target Gap = [Order Target] - [Total Orders]
```

```
Revenue Target = 7180      -- scoped monthly; or LOOKUPVALUE from a
target table
Revenue Target Gap = [Revenue Target] - [Total Revenue]

Profit Target = 4613
Profit Target Gap = [Profit Target] - [Total Profit]
```

4.6 Time Intelligence & Rolling Windows

```
Previous Month Revenue =
CALCULATE([Total Revenue],
          DATEADD('Calendar Lookup'[Date], -1, MONTH))

Previous Month Profit =
CALCULATE([Total Profit],
          DATEADD('Calendar Lookup'[Date], -1, MONTH))

Previous Month Orders =
CALCULATE([Total Orders],
          DATEADD('Calendar Lookup'[Date], -1, MONTH))

10 Day Rolling Revenue =
VAR LastDate = MAX('Calendar Lookup'[Date])
RETURN CALCULATE([Total Revenue],
                 DATESINPERIOD('Calendar Lookup'[Date], LastDate, -10,
DAY))

90 Day Rolling profit =
VAR LastDate = MAX('Calendar Lookup'[Date])
RETURN CALCULATE([Total Profit],
                 DATESINPERIOD('Calendar Lookup'[Date], LastDate, -90,
DAY))

YTD Revenue =
TOTALYTD([Total Revenue], 'Calendar Lookup'[Date])
```

4.7 Share-of & Composition

```
All orders = CALCULATE([Total Orders], ALL('Product Lookup'))
% of All Orders = DIVIDE([Total Orders], [All orders])

All Returns = CALCULATE([Total Returns], ALL('Product Lookup'))
% of All Returns = DIVIDE([Total Returns], [All Returns])
```

4.8 Price & Ticket Size

```
Avg Retail Price = AVERAGE('Product Lookup'[ProductPrice])

High Ticket Orders =
CALCULATE([Total Orders],
         FILTER(VALUES('Product Lookup'[ProductKey]),
                'Product Lookup'[ProductPrice] > [Avg Retail Price]))

Over Average Price =
DIVIDE([High Ticket Orders], [Total Orders])
```

4.9 Weekend / Day-type metrics

```
Weekend Orders =
CALCULATE([Total Orders], 'Calendar Lookup'[Weekend] = TRUE())
```

4.10 Metric Switchers (disconnected selection tables)

Product Metric Selection (single-select slicer with values: “Orders”, “Revenue”, “Profit”, “Returns”, “Return %”)

```
Selected Product Metric :=
VAR Sel = SELECTEDVALUE('Product metric Selection'[Product metric
Selection], "Orders")
RETURN
SWITCH(
    TRUE(),
    Sel = "Orders",  [Total Orders],
    Sel = "Revenue", [Total Revenue],
    Sel = "Profit",  [Total Profit],
    Sel = "Returns", [Total Returns],
    Sel = "Return %",[Return Rate],
    [Total Orders]
)
```

Customer Metric Selection (e.g., “Total Customers”, “Revenue per Customer”)

```
Selected Customer Metric :=
VAR Sel = SELECTEDVALUE('Customer Metric Selection'[Customer Metric
Selection], "Total Customers")
RETURN
SWITCH(
    TRUE(),
    Sel = "Total Customers", [Total Customers],
    Sel = "Revenue per Customer", [AvgRevenuePerCustomer],
    [Total Customers]
)
```

---

5) Visuals → Measures Mapping

- **KPI Cards (Revenue, Orders, Profit, Return Rate):** `[Total Revenue]`, `[Total Orders]`, `[Total Profit]`, `[Return Rate]`.
- **Revenue Trending:** `[Total Revenue]` with trend line; comparison to `[Previous Month Revenue]` and `[YTD Revenue]`.
- **Orders by Category / Top 10 Products:** `[Total Orders]`, `% of All Orders`.
- **Return KPI bubbles in table:** `[Return Rate]`.
- **Product Detail Gauges:** `Order/Revenue/Profit Target` vs actual and `Gap` measures.
- **Price What-If slider:** drives `[Adjusted Price]`, `[Adjusted Revenue]`, `[Adjusted Profit]`.
- **Customer Detail:** `[Total Customers]`, `[AvgRevenuePerCustomer]`, and switcher `[Selected Customer Metric]`.

- **Map:** `[Total Orders]` aggregated by `Territory Lookup[Country]`.
- **Decomposition Tree:** base measure `[Total Orders]` with dimensions Category → Subcategory → Product.

---

6) Design Choices & Rationale

- **Star schema** minimizes ambiguity and optimizes DAX performance.
- **Inactive Calendar↔Returns** keeps a single canonical Date table; `USERELATIONSHIP` activates it only when needed.
- **Disconnected parameter/selection tables** enable what-if analysis and metric switching without polluting relationships.
- **Measures over columns** keeps the model lean and responsive; time intelligence stays fully filter-aware.
- **Targets** separated from actuals for clean variances and gauge visuals.

---

7) Implementation Tips / Gotchas

- Always mark **Calendar** as a **Date table**.
- Ensure **one active date relationship** (Sales) and **one inactive** (Returns).
- For performance, use **SUMX** only when row context is required; otherwise plain `SUM`.
- Use **format strings** (₹, %, K/M abbreviations) to match card visuals.
- For the Top-N product table, apply a **TOPN()** filter or visual “Top N” with `[Total Orders]`.