Comprehensive Power BI Model & Dashboard Validation Report

Subject: A complete analysis of the Power BI model's health, performance, and data accuracy, including detailed root cause analysis, all validation queries, and a definitive plan for correction.

1. Executive Summary

This report details the results of a multi-faceted investigation into our primary Power BI solution. The analysis included a deep technical audit of the model's internal structure, a comprehensive review of all DAX measures, and a rigorous visual-by-visual cross-validation of all report pages against the source SQL database.

My analysis uncovered both significant opportunities for optimization and a critical data accuracy issue that impacts all areas of the report.

Key Findings:

1. **Critical Accuracy Error:** The **"Volume of Orders" KPI is fundamentally incorrect and misleading**. The root cause is a flawed DAX measure (`COUNT(Sales[ProductKey])`) that counts every individual transaction line instead of unique orders. This systematically overstates this key metric by over 350%.
2. **Major Structural Flaws:** The model is severely bloated by widespread data duplication. The root cause is an error in the data loading process where core tables (`Sales`, `Product`, `Date`) were imported multiple times. This was technically proven by the presence of a `1-to-1` relationship between `Product` and `Product 2`.
3. **Critical Data Integrity Violation:** My analysis of the model's relationships revealed that **2,113 sales records have a blank or invalid `ShipDateKey`**. This is a major data quality issue that causes these records to be silently dropped from any analysis filtering by shipping date, leading to understated revenue figures.
4. **Performance Bottlenecks:** The `Customer` table is a major performance bottleneck due to the storage of unoptimized, high-cardinality text columns that are required for the report visuals.

I have validated that other key metrics on the dashboards, such as Total Revenue and Profit, are accurate. The action plan below provides a prioritized, step-by-step guide to remediate all identified issues.

Part 1: Overall Model Health & DAX Measure Audit

My initial analysis focused on the internal structure and logic of the Power BI model.

- **Finding 1.1: Widespread Data Duplication:** The model contains complete duplicates of our most important tables: `Sales` vs. `Sales 2`, `Product` vs. `Product 2`, and `Date` vs. `Date 2`.
- **Finding 1.2: Inefficient Customer Table:** The `Customer` table is the single largest consumer of memory. While fields like `Customer-CustomField` are necessary for the visuals, their inefficiency stems from their high cardinality (many unique values).
- **Finding 1.3: Flawed DAX Logic Identified:** The DAX metadata provided definitive proof of flawed logic:
  - **Volume of Orders:** The formula `COUNT(Sales[ProductKey])` is definitively the wrong logic for counting orders.
  - **Average Sales Price:** The formula `AVERAGE(Sales[Unit Price])` uses an unweighted average, which can be misleading.

Part 2: Data Model and Relationship Analysis

My analysis of the model's internal relationship structure provides definitive evidence of the structural and data quality issues.

Relationship Analysis Findings:

- **Finding 2.1: Confirmation of Duplicated Tables**
  - **Observation:** The relationship metadata explicitly shows a **`Product 2[ProductKey] 1↔1 Product[ProductKey]`** relationship.
  - **Root Cause Analysis:** A one-to-one relationship between two tables on their primary key is a technical proof that they are duplicates. This, along with the relationship found for `Sales 2` and `Date 2`, confirms that an error was made during the data import process in Power Query, where the same sources were loaded multiple times.
- **Finding 2.2: Critical Data Integrity Violation in `Sales[ShipDateKey]`**
  - **Observation:** The metadata for the relationship `Sales[ShipDateKey] ∞←1 Date[DateKey]` reports the following critical errors:
    - **`Invalid Rows: 2113`**
    - **`Missing Keys: 1`**
    - **`Sample Violations: (blank)`**
  - **Root Cause Analysis:** This technical data means: **2,113 transaction rows** in our `Sales` table have a shipping date that is blank or null. Because a blank value cannot find a match in our official `Date` table, this relationship is broken for all of those records. The root cause is poor data quality in the source system, which allows sales transactions to be recorded without a valid shipping date.
  - **Business Impact:** This is a critical data quality issue. Any report, measure, or slicer that uses the shipping date will **silently ignore these 2,113 sales records**, leading to understated revenue and profit figures. This directly impacts the accuracy of the `Revenue by shipdate` DAX measure, making it completely unreliable.

Part 3: Dashboard Visual Cross-Validation

I performed a detailed validation of each visual on all three report pages against our source SQL database.

Page 1: "Company Performance Overview in FY2020"

- **Total Revenue (51.88M), Profit (5.81M), and related charts were all validated as □ Accurate.**
- **Volume of Orders (84K)** was confirmed to be □ **Incorrect**.

- **SQL Finding:** The true number of distinct orders is **23,519**. The number of transaction lines is **84,285**. The DAX measure was counting the latter.

Page 2: "Customer Detail" (for customer `AW00011004`)

- **Financial KPIs (Total Revenue, Profit, etc.) and supporting charts were validated as □ Accurate.**
- **Volume of Orders (6)** was confirmed to be □ **Incorrect**.
  - **SQL Finding:** The true number of distinct orders for this customer is **3**. The visual was displaying the **6** transaction lines associated with the customer.

Page 3: "Reseller Detail" (for reseller `AW00000006`)

- **Financial KPIs (Total Revenue, Profit, etc.) and supporting charts were validated as □ Accurate.**
- **Volume of Orders (10)** was confirmed to be □ **Incorrect**.
  - **SQL Finding:** The true number of distinct orders for this reseller is **4**. The visual was displaying the **10** transaction lines associated with the reseller.

Part 4: Final Consolidated Action Plan (Correction Steps in Power BI)

Based on all findings, I have prioritized the following actions to correct errors and optimize the model.

1. **Priority 1 (Immediate Fix): Correct the Global "Volume of Orders" KPI**
   - **Action:** In Power BI Desktop, I will delete the current, flawed DAX measure `Volume of Orders`. I will replace it with a new measure that correctly performs a distinct count on the order number identifier.
   - **New DAX Formula:** `Volume of Orders = DISTINCTCOUNT(Sales[SalesOrderNumber])`
2. **Priority 2 (High Impact): Remediate Structural Model Flaws**
   - **Action:** In the Power Query Editor, I will delete the `Sales 2`, `Product 2`, and `Date 2` queries. I will then remove all obsolete DAX measures that referenced them.
3. **Priority 3 (Critical Data Quality): Fix `ShipDateKey` Integrity Violation**
   - **Action:** In Power Query, I will select the `Sales` query, target the `ShipDateKey` column, and use the "Replace Values" feature to find `null` values and replace them with a placeholder (e.g., `-1`). I will then ensure a corresponding "Unknown Date" row with a key of `-1` exists in the `Date` table.
   - **Impact:** This will fix the relationship for all 2,113 invalid rows, ensuring that measures like `Revenue by shipdate` are accurate and include all sales data.
4. **Priority 4 (Performance Tuning): Optimize the Customer Table**
   - **Action:** The following nuanced plan will be implemented to optimize the `Customer` table without breaking report visuals:
     1. **Isolate Component Columns:** In Power Query, I will ensure individual source columns that make up fields like `Customer-CustomField` (e.g., `Customer ID`, `First Name`, `Last Name`) are loaded.
     2. **Recreate Composite Fields in Power Query:** I will use the **"Merge Columns"** feature to create the exact combined fields needed for the visuals.
     3. **Remove Original High-Cardinality Columns:** I will remove the original, inefficient source columns that were pre-concatenated.
   - **Impact:** This "split and recombine" strategy will drastically reduce the model size while keeping the report visuals identical to the user.

Appendix: SQL Validation Queries Used

Here is the complete list of SQL queries I used to validate the Power BI report visuals against the source database.

A. Queries for "Company Performance Overview" Page

```
 -- A.1: Total Revenue for FY2020
SELECT SUM(s.SalesAmount) AS TotalRevenue_FY2020
FROM (
    SELECT OrderDateKey, SalesAmount FROM FactInternetSales
    UNION ALL
    SELECT OrderDateKey, SalesAmount FROM FactResellerSales
) s
JOIN DimDate d ON s.OrderDateKey = d.DateKey
WHERE d.FiscalYear = 2020;

-- A.2: Total Profit for FY2020
SELECT SUM(s.SalesAmount - s.TotalProductCost)
 AS TotalProfit_FY2020
FROM (
    SELECT OrderDateKey, SalesAmount, TotalProductCost
FROM FactInternetSales
    UNION ALL
    SELECT OrderDateKey, SalesAmount, TotalProductCost
FROM FactResellerSales
) s
JOIN DimDate d ON s.OrderDateKey = d.DateKey
WHERE d.FiscalYear = 2020;

-- A.3: Correct Distinct Order Count for FY2020
SELECT COUNT(DISTINCT SalesOrderNumber)
AS DistinctOrderCount_FY2020 FROM (
    SELECT fis.SalesOrderNumber
FROM FactInternetSales fis JOIN DimDate d
 ON fis.OrderDateKey = d.DateKey
WHERE d.FiscalYear = 2020
    UNION ALL
    SELECT frs.SalesOrderNumber
FROM FactResellerSales frs JOIN DimDate d
 ON frs.OrderDateKey = d.DateKey
 WHERE d.FiscalYear = 2020
) AS AllOrders;

-- A.4: Incorrect Order Line Count for FY2020 (to prove the error)
SELECT SUM(LineCount) AS TotalOrderLines_FY2020
FROM (
    SELECT COUNT(*) AS LineCount
 FROM FactInternetSales fis JOIN DimDate d
 ON fis.OrderDateKey = d.DateKey WHERE d.FiscalYear = 2020
    UNION ALL
    SELECT COUNT(*) AS LineCount
FROM FactResellerSales frs JOIN DimDate d
ON frs.OrderDateKey = d.DateKey WHERE d.FiscalYear = 2020
) AS OrderLines;
```

B. Queries for "Customer Detail" Page (Customer `AW00011004`)

```
-- B.1: Financial KPIs for a single customer
SELECT
    SUM(fis.SalesAmount) AS TotalRevenue,
    SUM(fis.SalesAmount - fis.TotalProductCost) AS TotalProfit,
    SUM(fis.SalesAmount - fis.TotalProductCost) / SUM(fis.SalesAmount)
 AS ProfitMargin
FROM FactInternetSales fis
JOIN DimCustomer dc ON fis.CustomerKey = dc.CustomerKey
WHERE dc.CustomerAlternateKey = 'AW00011004';

-- B.2: Incorrect Order Line Count for a single customer
SELECT COUNT(*) AS NumberOfOrderLines
FROM FactInternetSales fis
JOIN DimCustomer dc ON fis.CustomerKey = dc.CustomerKey
WHERE dc.CustomerAlternateKey = 'AW00011004';
```

C. Queries for "Reseller Detail" Page (Reseller `AW00000006`)

```
-- C.1: Financial KPIs for a single reseller
SELECT
    SUM(frs.SalesAmount) AS TotalRevenue,
    SUM(frs.SalesAmount - frs.TotalProductCost) AS TotalProfit,
    SUM(frs.SalesAmount - frs.TotalProductCost) / SUM(frs.SalesAmount)
 AS ProfitMargin
FROM FactResellerSales frs
JOIN DimReseller dr ON frs.ResellerKey = dr.ResellerKey
WHERE dr.ResellerAlternateKey = 'AW00000006';

-- C.2: Incorrect Order Line Count for a single reseller
SELECT COUNT(*) AS NumberOfOrderLines
FROM FactResellerSales frs
JOIN DimReseller dr ON frs.ResellerKey = dr.ResellerKey
WHERE dr.ResellerAlternateKey = 'AW00000006';
```