

# Technical Documentation - Bank Branch Profitability Analysis

---

## Overview

This technical documentation provides a detailed walkthrough of how the "Bank Branch Profitability Analysis" report was developed in Power BI. It includes data preparation steps, data modeling logic, DAX measures, and advanced interactive features.

The objective was to build a comprehensive tool to **assess branch health, simulate interest-rate shocks, and rank branches by risk-adjusted return (RAR)**. I was the sole contributor, responsible for handling the complete analytics pipeline from data ingestion to report design.

---

## Process Workflow

Excel Dataset → Power Query (Cleaning) → Data Modeling → DAX Calculations → Interactive Visual Design → Publishing

---

## Data Preparation (Power Query)

Data was loaded from the Excel source file. The following transformations were performed in the Power Query Editor to clean and model the data before loading it into Power BI.

### Key Steps:

#### 1. Created a Central Fact Table (Bank Details):

- The **Fact\_Deposits** table was used as the base.
- **Loans, Fees, and Loan\_Loss\_Provisions** tables were **merged** into **Fact\_Deposits** using a **Left Outer Join**.
- Initially, I used a Composite Key (**Branch\_ID + Month**) to merge the transactional tables. However, this resulted in significant performance lag during data refresh. To optimize the ETL pipeline, I implemented a **Surrogate Key** strategy. I generated a uniform Index Column across all sorted tables (ensuring identical sort order first)

and used this integer-based key for the merge. This reduced the computational load and significantly improved the model's load time.

- All financial columns (**Deposit\_Amount**, **Loan\_Amount**, etc.) were standardized to the **Currency** data type.

## 2. Created a Monthly Date Dimension (Date (Monthly)):

- A daily **Dim\_Date** sheet was loaded.
- A **Start of Month** column was inserted.
- The original **Date** column and other daily-grain columns (e.g., **Week of the Year**, **Is Weekend**) were removed.
- **Duplicates were removed** from the **Start of Month** column to create a unique row for each month, transforming the table from a daily grain to a monthly grain.
- Month names were shortened to 3 characters (e.g., "Jan"), and **Quarter** was formatted with a "Q" prefix (e.g., "Q1").

## 3. Loaded Dimension & Secondary Fact Tables:

- The **Dim\_Branch**, **Dim\_Rates**, and **Fact\_Operating\_Expenses** tables were loaded.
- For all tables, the first row was **promoted to headers**.
- All column data types were **checked and set correctly**.

## 4. Optimized Performance:

Query load was disabled for the original **Loans**, **Fees**, and **Loan\_Loss\_Provisions** tables, as they had already been merged into the **Bank Details** table. This reduces the data model's memory footprint and improves refresh speed.

---

## Data Modeling

The data model was designed as a **Star Schema** with **two central fact tables** (Bank Details and Operating\_Expenses) and **three primary dimension tables** (Date (Monthly), Branch, and Rates). This schema optimizes performance and simplifies DAX calculations.

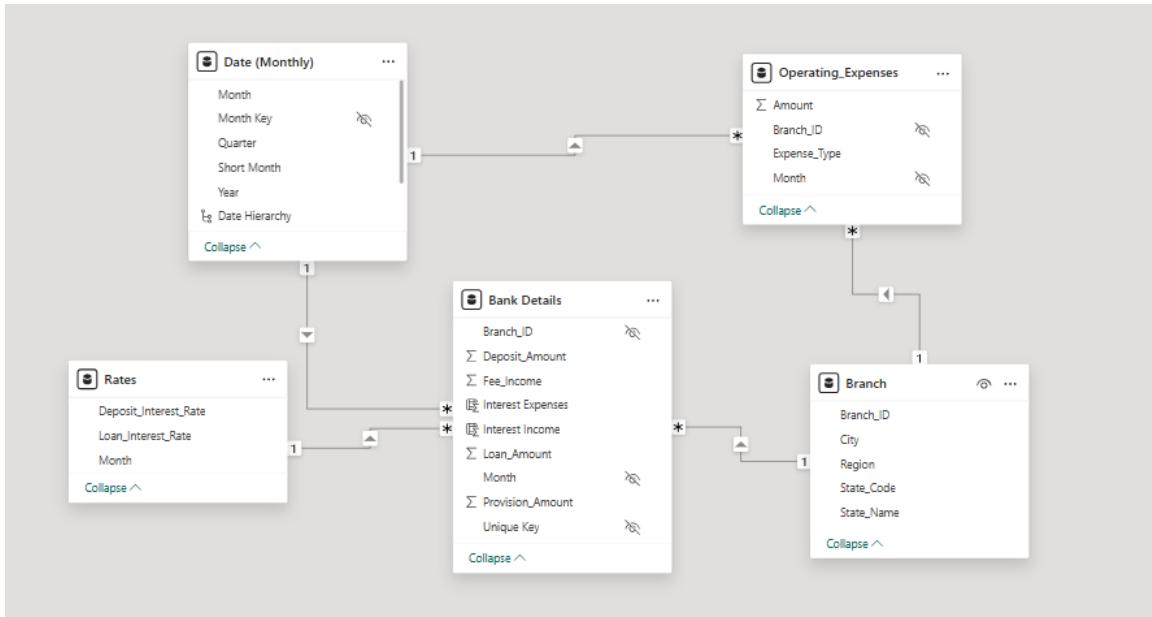
The tables are:

1. **Bank Details (Fact Table):** Contains the primary financial transactions (Deposits, Loans, Fees, Provisions) at a branch/month grain.

2. **Operating\_Expenses (Fact Table):** Contains the expense data (Salaries, Rent, Utilities) at a branch/month/expense\_type grain.
3. **Date (Monthly) (Dimension Table):** The primary date dimension, providing time context (Year, Quarter, Short Month, Month Key).
4. **Branch (Dimension Table):** Describes the branch hierarchy (Region, State, City).
5. **Rates (Dimension Table):** A supporting dimension table connected to the **Bank\_Details** table, providing the interest rates for each month.

#### **Additional Modeling Details:**

1. **Relationship Cardinality:** All relationships are **One-to-Many (1:\*)** from the dimension tables to the fact tables. All relationships are set to a **Single cross-filter direction**.
2. **What-if Parameter Table:** A disconnected table named **Interest Rate Shocks** was created as a **Parameter**. This table is not related to any other table and is used to capture user input from a slicer for the scenario analysis.
3. **Column Sorting:** To ensure correct chronological order in visuals, the **Short Month** column was **sorted by the Month Key column (1-12)**.
4. **Hierarchies Created:** A **Date** hierarchy (**Year > Quarter > Short Month**) was created to enable easy drill-down functionality on all time-series charts.
5. **Hidden Fields:** To simplify the report-building experience and prevent incorrect analysis, key columns (**Branch\_ID, Month Key**) and other redundant fields were **hidden from the report view**.



## DAX Measures

Here are the key measures used in the report:

### 1. Core KPIs

**Profit Margin % = DIVIDE([Net Profit], [Total Revenue], 0)**

**Cost to Income % = DIVIDE([Operating Expenses], [Total Revenue], 0)**

**Loan/Deposit Ratio % = DIVIDE([Total Loan], [Total Deposit], 0)**

**Fee Income % = DIVIDE([Total Fee Income], [Total Revenue], 0)**

### 2. Advanced Analytics (RAR Model)

These measures form the core of the project's "risk vs. reward" analysis.

**Profit Volatility Index** This measure calculates the "Risk" component of the RAR model. It uses the **STDEVX.P** function to find the population standard deviation of **Net Profit** over the entire time period, providing a single value for each branch's profit volatility.

**Profit Volatility Index = STDEVX.P('Date (Monthly)', [Net Profit])**

**Risk Adjusted Return** This is the "master" KPI used to rank all 40 branches. It calculates the "reward vs. risk" ratio by dividing a branch's average **Net Profit** by its **Profit Volatility Index**. A higher RAR indicates a better, more stable return.

**Risk Adjusted Return = DIVIDE(AVERAGEX('Date (Monthly)', [Net Profit], [Profit Volatility Index], 0)**

### 3. Time Intelligence & UX Measures

This group of measures calculates year-over-year growth and provides a better user experience for the growth chart.

**Revenue Growth % & Cost Growth %** These measures calculate the year-over-year (YoY) growth for Revenue and Cost.

- The logic uses **CALCULATE** with **SAMEPERIODLASTYEAR** to get the value from the prior year.
- The **IF(NOT ISBLANK())** check is critical. It prevents the measure from returning **0** or an error for the first year (2023), which has no prior-year data. This **BLANK()** output allows the line chart to simply start in 2024.
- **Cost Growth %** follows the identical logic.

**Revenue Growth % =**

**VAR Previous\_Year\_Revenue = CALCULATE([Total Revenue],  
SAMEPERIODLASTYEAR('Date (Monthly)'[Month]))**

**RETURN**

**IF(NOT ISBLANK(Previous\_Year\_Revenue),  
DIVIDE([Total Revenue] - Previous\_Year\_Revenue, Previous\_Year\_Revenue, 0))**

**Cost Growth % =**

**VAR Previous\_Year\_Cost = CALCULATE([Total Cost], SAMEPERIODLASTYEAR('Date (Monthly)'[Month]))**

**RETURN**

```
IF(NOT ISBLANK(Previous_Year_Cost),  
    DIVIDE([Total Cost] - Previous_Year_Cost, Previous_Year_Cost, 0))
```

**Growth Chart Message** This measure is used to provide a "smart message" to the user, creating a better user experience.

- **Context:** When a user selects *only* 2023, the YoY growth chart is blank (as intended by the DAX above), which can be confusing.
- This measure checks if the **SELECTEDVALUE** for the year is 2023. If it is, it returns a helpful text message. This measure is placed on a Card visual *on top of* the blank chart, so the message appears automatically, guiding the user.

**Growth Chart Message =**

```
VAR Selected_Year = SELECTEDVALUE('Date (Monthly)'[Year])
```

```
RETURN
```

```
IF(Selected_Year = 2023,
```

**"Year-over-year growth data begins in 2024. Please select 2024 or 2025 to see the trend.", " "**

#### 4. "What-if" Scenario Simulation

These measures power the "What-if Analysis" page, allowing users to simulate the impact of interest rate changes.

**Simulated Net Interest Income = [Simulated Interest Income] - [Total Interest Expenses]**

**NII Impact (\$) = [Simulated Net Interest Income] - [Net Interest Income]**

**NII Impact (%) = DIVIDE([NII Impact (\$)], [Net Interest Income], 0)**

#### 5. Dynamic Visual Measures

These measures are used for conditional formatting and creating dynamic titles to enhance the report's visual experience.

**Dynamic Drill-through Title** This measure creates the dynamic title for the "Branch Details" drill-through page. It checks if a single branch is selected and, if so, combines its ID, City, State, and Region into a clean title.

**Drillthrough =**

**VAR Branch\_Details =**

**"Branch " &**

**SELECTEDVALUE(Branch[Branch\_ID]) &**

**" - " &**

**SELECTEDVALUE(Branch[City]) &**

**"; " &**

**SELECTEDVALUE(Branch[State\_Name]) &**

**" (" &**

**SELECTEDVALUE(Branch[Region]) &**

**")"**

**RETURN**

**SWITCH(TRUE(),**

**HASONEVALUE(Branch[Branch\_ID]), Branch\_Details, "ALL")**

**Top10/Bottom10 Branches by RAR** This measure is used for the **conditional formatting icons** in the main branch ranking table, as required by the project brief. It ranks all branches by RAR and assigns a text value used to display up-arrows (Top 10), down-arrows (Bottom 10), or yellow dashes (Others).

**Top10/Bottom10 Branches by RAR =**

**VAR Ranking\_by\_RAR = RANKX(ALL(Branch), [Risk Adjusted Return], , DESC, Dense)**

**RETURN**

**SWITCH(TRUE(),**

**Ranking\_by\_RAR <= 10, "Top 10",**

**Ranking\_by\_RAR > 30, "Bottom 10", "Others")**

**Quantile Color Formatting** This measure is used for the **conditional formatting** of the "Profit Margin % by State" map visual. It compares each state's Profit Margin % to pre-calculated quantile measures ([Profit Margin % 25Q], etc.) to assign the correct color (Red, Yellow, etc.).

**Quantile Color Formatting =**

**SWITCH(TRUE(),**

**[Profit Margin %] <= [Profit Margin % 25Q], "Red",**

**[Profit Margin %] <= [Profit Margin % 50Q], "Yellow",**

**[Profit Margin %] <= [Profit Margin % 75Q], "Light Green", "Green")**

**Regional Benchmark Comparison** To contextualize a branch's performance, I created a measure that dynamically calculates the median profit margin of the selected region. This prevents unfair comparisons (e.g., comparing a rural branch to a metro branch).

**Regional\_P/M\_Benchmark =**

**VAR Current\_Region = SELECTEDVALUE/Branch[Region])**

**RETURN**

**CALCULATE(MEDIANX/Branch, [Profit Margin %]),**

**FILTER(ALL/Branch), Branch[Region] = Current\_Region))**

**Dynamic Peer Quartile Status** This measure segments branches into performance quartiles (1st to 4th) relative to their region. It uses **PERCENTILEX.INC** to determine thresholds and **SWITCH** to assign a text status. This is used in the drill-through header to give an instant "Health Check" status.

**Profit\_Margin\_Peer\_Quartile\_Status =**

**VAR Current\_Branch = "Branch " & SELECTEDVALUE/Branch[Branch\_ID]) & ":"**

**VAR Current\_Region = SELECTEDVALUE/Branch[Region])**

**VAR Percentile\_75th = CALCULATE(PERCENTILEX.INC/Branch, [Profit Margin %], 0.75), FILTER(ALL/Branch), Branch[Region] = Current\_Region))**

```

VAR Percentile_50th = CALCULATE(PERCENTILEX.INC/Branch, [Profit Margin %],
0.50), FILTER(ALL/Branch), Branch[Region] = Current_Region))

VAR Quartile_Status = SWITCH(TRUE(),

[Profit Margin %] >= Percentile_75th, Current_Branch & " 1st Quartile
(High Margin)",

[Profit Margin %] >= Percentile_50th, Current_Branch & " 2nd Quartile
(Above Avg)",

[Profit Margin %] >= Percentile_25th, Current_Branch & " 3rd Quartile
(Below Avg)", Current_Branch & " 4th Quartile (Low Margin)")

RETURN Quartile_Status

```

---

## Interactive Features & Visual Experience

The report was designed to be a fully interactive tool.

- **Page Navigator:** A main navigation pane for easy movement between the three core pages.
  - **Bookmark Navigator:** Used as **clickable tabs** to switch between different analytical views on the 'Performance Analysis' and 'What-if Analysis' pages.
  - **Drill-through:** Users can **right-click any branch** on a table to jump to a **dedicated, full-page report** ('Branch Details') with its specific performance breakdown.
  - **Sync-Slicers:** The **Year, Region, and State** slicers are synced across all pages for a consistent filtering experience.
  - **Drill-down:** The Date hierarchy is enabled on all time-series charts, allowing users to **drill-up/down** to expand from a yearly view down to a quarterly or monthly view or vice versa.
  - **Information Icons (Toolips):** Information icons are placed next to key visuals to provide extra context to the user.
-

## Testing and Quality Assurance

- Verified all core KPIs (**Profit Margin %**, **C/I %**, etc.) against manual calculations in Excel.
- Tested all slicers, bookmarks, and **Sync-Slicer** functionality to ensure correct filter context.
- Validated the **Drill-through** feature to ensure it passed the correct filter and the dynamic title worked as expected.
- Tested the **What-if** slicer to confirm that the **NII Impact (\$)** and **Simulated NII measures** were calculating correctly.