Week 5 Task: Performance Optimization and Report Enhancement

Internship: Virtual Power BI Data Insights Internship

Prepared by: Mohammad Saniya

Date: July 12, 2025

1. Objective

The objective of this task is to optimize Power BI data models and reports by applying performance best practices. The focus is on reducing report load times, enhancing visual responsiveness, streamlining queries, and improving overall user experience and efficiency.

2. Performance Bottlenecks Identified

- Redundant columns or unnecessary calculated columns.
- Use of too many slicers or visuals on a single page.
- High-cardinality columns not optimized.
- Unused relationships or tables present in the model.

3. Optimization Techniques Applied

- **Query Folding**

  • What it does: Allows Power BI to push transformations back to the data source. This reduces in-memory processing and speeds up data load.
  • Where applied: Applied in Power Query Editor for Attendance and Performance tables.

- **Incremental Refresh**

  • What it does: Enables refreshing only new or changed data instead of the entire dataset, improving performance in large datasets.
  • Where applied: Applied to the Attendance table with a date field as partition key.

- **Aggregation Tables**

• What it does: Stores pre-aggregated data to reduce on-the-fly calculations, especially in large datasets.
• Where applied: Used for class-level average scores and attendance summaries.

- **Disabling Auto Date/Time**

  • What it does: Reduces memory usage and speeds up model loading by preventing automatic date hierarchy creation.
  • Where applied: Disabled in report settings for all date fields.

- **Using Variables in DAX**

  • What it does: Improves DAX readability and reduces repeated calculations, optimizing query execution.
  • Where applied: Used in ScoreZScore and AttendanceTrend calculations.

- **Optimizing Relationships (1:many)**

  • What it does: Prevents relationship ambiguity and improves model performance and filtering.
  • Where applied: Ensured all fact tables link correctly to dimension tables using 1:many relationships.

- **Removing Unnecessary Columns**

  • What it does: Reduces dataset size and memory usage.
  • Where applied: Removed columns like middle name, full address, raw source columns not used in visuals.

4. Visual & Interactivity Enhancements

- Removed excess visuals that were slowing down rendering.
- Used bookmarks to toggle visibility between sections instead of loading all visuals at once.
- Limited slicers to only those necessary for filtering specific visuals.
- Used tooltips to display detailed information on hover, reducing dashboard clutter.

5. Optimization Roadmap

1. Step 1: Remove unused columns.
2. Step 2: Apply query folding in Power Query.
3. Step 3: Replace calculated columns with DAX measures.
4. Step 4: Apply incremental refresh to large fact tables.
5. Step 5: Create aggregation table for class-level KPIs.
6. Step 6: Organize visuals across report pages and reduce slicers.
7. Step 7: Use Performance Analyzer to measure impact.

6. Monitoring and Validation

- - Use the Performance Analyzer in Power BI Desktop to track visual load times.
- - Compare report refresh duration before and after optimizations.
- - Use DAX Studio to analyze heavy queries and optimize them.
- - Maintain a version control log to document every change applied during optimization.

7. Summary Table of Improvements

Area	Problem Identified	Optimization Applied	Expected Benefit
Data Model	Redundant columns	Removed unnecessary fields	Reduced memory load
Query Load	Slow refresh time	Enabled query folding	Faster data refresh
Visual Responsiveness	Cluttered visuals/slicers	Reduced visuals, used bookmarks	Faster report interaction
DAX Measures	Repeated logic	Used VAR to reduce calculation	Improved query performance

8. Before vs After Optimization Summary

The following table compares the report's performance metrics before and after optimization. These are estimated based on observed refresh durations and visual load times in Power BI Desktop.

Metric	Before Optimization	After Optimization
Data Refresh Time	2 minutes	45 seconds
Visual Load Time	5–6 seconds	1–2 seconds
Memory Usage	High	Optimized
DAX Efficiency	Repeated calculations	Optimized with variables

9. Testing and User Feedback Simulation

Post-optimization, the dashboard was tested on multiple screen sizes and devices (laptop and desktop) to ensure visual responsiveness. Visual load times were evaluated using Power BI's Performance Analyzer. Filters and slicers were stress-tested with simultaneous selections to confirm stable behavior.

10. Documentation and Version Control

All optimization changes were logged in a version-controlled document with time stamps and expected impact. Rollback notes were maintained to allow safe experimentation with model and DAX changes. Performance screenshots were saved for internal tracking.

11. Key Learnings

- Learned how to identify and eliminate bottlenecks in Power BI reports.
- Gained hands-on experience with query folding, incremental refresh, and aggregation.
- Developed a better understanding of balancing visual richness with performance.
- Applied Power BI's analyzer tools and DAX Studio to validate improvements.
- Practiced documenting technical workflows in a professional and organized manner.