

Weely Sales DB

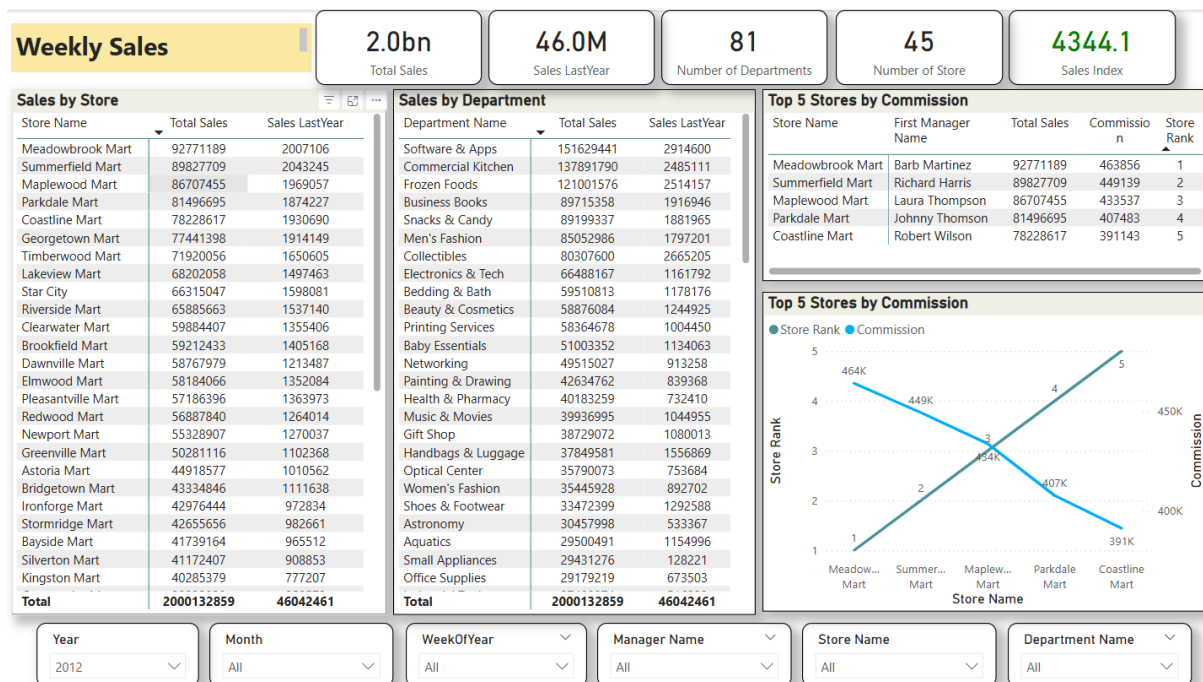
This Power BI dashboard project provides a comprehensive view of weekly sales performance across multiple stores and departments. The primary objective is to enable stakeholders to monitor trends, track top/bottom performers, and make data-driven decisions to improve sales operations.

Key highlights of the report include:

- Weekly trends for total sales by store and department.
- Ranking of top and bottom 10 stores/departments based on sales performance.
- Comparison with last year's sales to understand growth or decline.
- Dynamic filters to explore data by Week, Month, Year, and Manager.
- Sales Index to benchmark performance across different entities.

This dashboard simplifies complex sales data into actionable insights, helping business leaders identify opportunities, address challenges, and optimize sales strategies effectively.

Weekly Sales Page



Metrics at the Top:

- Total Sales
- Sales Last Year
- Number of Departments
- Number of Stores
- Sales Index - A calculated KPI - comparing sales trend or performance index.

Tables/Metrics:

- Sales by Store - Shows total and last year's sales for each store. Helps compare current vs past performance store-wise.
- Sales by Department - Displays department-wise total sales and last year's figures.
- Top 5 Stores by Commission - Focuses on commission for each store and their respective managers. Helps track manager-wise performance impact.
- Similarly Top 5 Store Sales with commission is showing in line chart.
- Filters/Slicers: Store Name, Department Name, Manager Name, Year, Month, WeekOfYear

Top Sales by Department and Store – Page 2



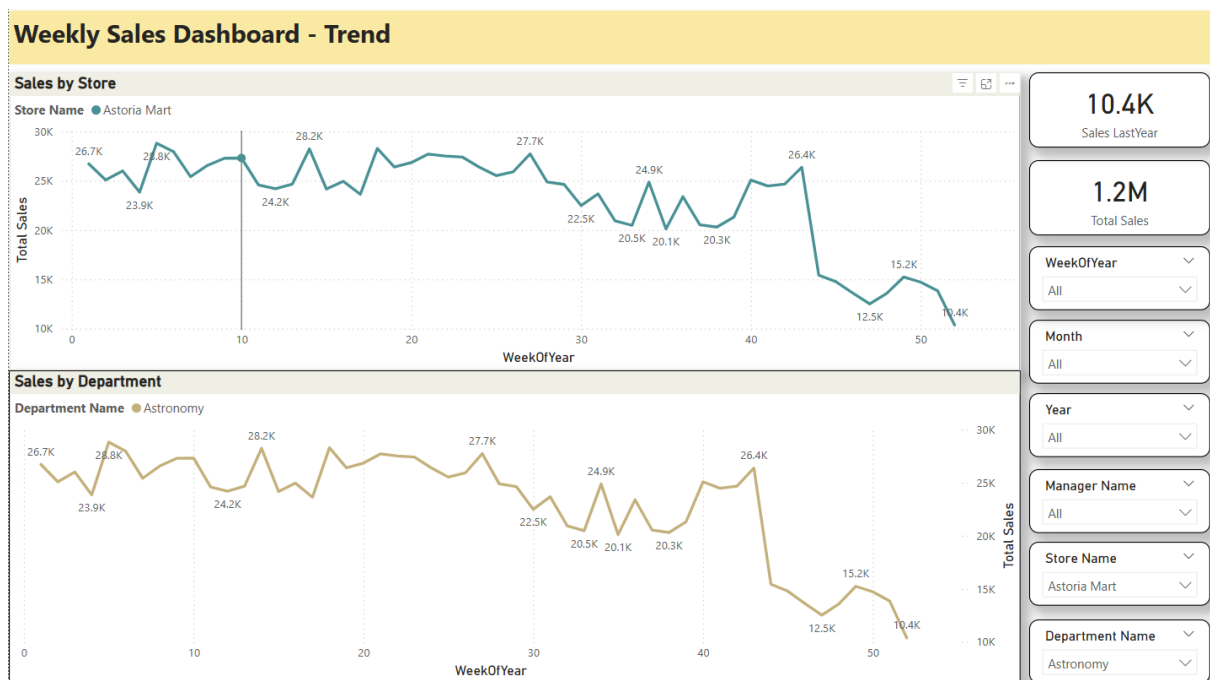
Metrics at the Top:

- Total Sales
- Sales Last Year
- Number of Departments
- Number of Stores
- Sales Index - A calculated KPI - comparing sales trend or performance index.

Tables/Metrics:

- Top & Bottom Performers
 - Top 10 Stores - Lists the top 10 stores based on total sales.
 - Bottom 10 Stores - Opposite of above — stores with lowest sales totals.
 - Top 10 Departments - Top-selling departments ranked by total sales.
 - Bottom 10 Departments - Departments with the least sales activity.
- Filters/Slicers: Store Name, Department Name, Manager Name, Year, Month, WeekOfYear

Weekly Sales Trend



This page shows how sales performance changed week by week throughout the year:

- **Sales by Store**
 - Each line represents one store's total weekly sales. Helps identify seasonal trends, dips, or spikes in store performance.
 - Use slicers (WeekOfYear, Month, Manager, Store Name, Department Name, Year) to narrow down specific periods or people.
- **Sales by Department**
 - Visualizes total weekly sales for each department.
 - Useful for spotting high-performing departments and sudden changes.
- **Key Metrics**
 - Total Sales current and last year cards.
 - Filters/Slicers: Store Name, Department Name, Manager Name, Year, Month, WeekOfYear

Recommendations to Sumera (The CEO)

Implement a simple data pipeline or partial “warehouse”:

- Use SharePoint/OneDrive + Incremental Refresh in Power BI Desktop
 - Store weekly files in a OneDrive folder synchronized with Power BI.
 - In Power Query, use parameters to only import data where DimDate[Date] > LatestDateInFact.
 - Turn on Incremental Refresh in Power BI (requires at least Power BI Pro).
 - Benefit: only new rows load each week. But you’ll still need to “Publish” and wait for the service to refresh.

Implement Incremental Refresh

- In Power BI Desktop, go to the Fact_WeeklySales query → Properties → Enable “Incremental refresh” → set policy (e.g. store last 3 years, refresh only last 4 weeks).
- This requires you to have Date in your fact to partition by. Once set up, the service will only refresh the latest week (rather than reprocessing all 100+ weeks).

Standardize File Naming & Storage

- Ask Suleman to name weekly files consistently (e.g. Sales_YYYY_WW.xlsx).
- Store them in a cloud folder (OneDrive or SharePoint) so that Power BI Service can automatically pick them up.
- Archive older files to a subfolder like Archive\2012\Week01.xlsx (this prevents clutter).

Pros & Cons for the CEO

- Manual Folder Approach (Today):
 - Zero extra cost (no new tools), easy to demonstrate.
 - Slow refresh, more error-prone, no incremental load, difficult to scale beyond ~200 files.
- Power BI Dataflows / Incremental Refresh:
 - Automates data ingestion weekly, faster refresh times, all transformations happen in cloud.
 - Requires Premium (or premium-per-user) licensing, some upfront setup.
- SQL-based Pipeline:
 - Future-proof (if you ever add online stores or monthly data), central data source for multiple reports/users, security at row/column level possible.
 - Requires database admin work, possibly an ETL tool, and some scripting.
- Recommendation: If the chain of departmental stores is fairly large and going to keep generating weekly extracts for years, begin with a minimal Power BI Dataflow (cloud-based ETL) so that your refresh time stays under 5 minutes once you exceed ~150 weekly files.

Convincing Sumera: The Case for an Automated Data Pipeline

Current Pain Points

- Each refresh reprocesses all 94 (soon 95, 96...) files → longer refresh times (already 2–4 minutes; after 200 weeks maybe 8–10 minutes).
- Risk of human error: forgetting to move the new file, misnaming it, or accidentally overwriting a historical file.
- Scalability: as the company opens new stores/departments or moves to daily data, flat files become unsustainable.

Benefits of Starting a Minimal Data Pipeline

- Automated ingestion: a small ETL (e.g. PowerShell) picks up new weekly files and appends them into a central SQL table.
- Incremental refresh in Power BI, so PBIX only needs to recalc latest week instead of all 200+ rows.
- Single source of truth: if other teams (e.g. Finance, Operations) want to build their own reports, they can connect to the same SQL database.
- Error detection: ETL can log invalid files or schema mismatches and alert IT or Suleman immediately.
- Performance: a centralized SQL With proper indexing can serve queries in seconds, whereas flat Excel files may time out.
- Security: define row-level permissions in SQL (e.g. store managers only see their store), whereas Excel has no controlled security.

Suggested Roadmap to Sumera

- Phase 1 (Now – 1 month):
 - Build and finalize the Power BI model using “From Folder” (as described above).
 - Document the manual refresh steps.
 - Need to show her a prototype dashboard with Week 19 (19 Oct 2012–25 Oct 2012) live data.
- Phase 2 (Next 2 months):
 - Spin up a small Azure SQL database.
 - Create a table [dbo].[WeeklySales] with schema matching your fact file (StoreID, DeptID, WeekStartDate, Sales, UnitsSold, etc.).
 - Write a lightweight PowerShell package that:
 1. Scans the “All data” folder for any new weekly file not yet loaded (use a UTC timestamp or a control table).
 2. Loads it into [dbo].[WeeklySales].
 3. Archives the file to /Archive/Year/Week.xlsx.
 - Update your Power BI model to point to [dbo].[WeeklySales] instead of folder. Build the same DAX measures, so your visuals don’t change.
 - Enable Incremental Refresh on the SQL-based table so that only the latest week is refreshed nightly.
- Phase 3 (Ongoing – Quarterly Reviews):
 - Monitor performance. After 1 year, you’ll have ~52 records × (# stores × # depts). If that grows substantially, consider partitioning by YearWeek in SQL or using Azure Synapse.
 - Enhance the model: add “Promotion” flags, “Holiday” flags, or integrate with HR data (so you can see if a manager change affected sales).

Estimate Gains

- Manual Model: 3–4 minutes to refresh now → 8–10 minutes in 6 months → 15+ minutes in 1 year.
- SQL + Incremental: Nightly refresh runs in, 1–2 minutes (only the new week).
- ROI: If you spend 10 minutes every Monday morning refreshing and validating, that's 40 hours per year (assuming 52 weeks). With an automated pipeline, you reduce that to 5 minutes/week → 4.3 hours per year. You "buy back" ~35–36 hours of staff time (worth thousands in salaries).