

Replacing Dashboards with Natural Language

MGMT 675: Generative AI for Finance

Kerry Back

The Dashboard Trap

Organizations spend millions building dashboards that answer **yesterday's questions**. When a new question arises, the cycle restarts: requirements gathering, design, development, QA, deployment.

The Typical Dashboard Lifecycle

1. Business user requests a report
2. Analyst translates to requirements
3. Engineer builds SQL queries
4. Designer creates visualizations
5. IT deploys to Tableau/Power BI
6. User asks a follow-up question
7. **Back to step 1**

The Cost

- **Time:** Weeks to months per dashboard
- **Money:** Tableau/Power BI licenses, engineering hours, maintenance
- **Rigidity:** Fixed views of fixed data
- **Low adoption:** Fewer than half of D&A teams effectively provide value (**Gartner, 2023**)
- **Staleness:** Questions evolve; dashboards don't

Dashboard Fatigue is Real

- Gartner (2019): “Through 2022, only **20%** of analytic insights will deliver business outcomes” ([Designing for Analytics summary](#))
- Most dashboards are built, viewed a few times, and abandoned
- The bottleneck isn’t data — it’s the translation layer between *questions* and *answers*

The fundamental problem: dashboards answer **pre-defined questions**. But the most valuable analysis comes from **ad-hoc questions** that arise in the moment.

- “What happened to margins in the Southeast last quarter?”
- “Show me our top 10 customers by growth rate, excluding one-time orders”
- “Compare Q3 headcount vs. budget by department, and flag anyone over 110%”

These are simple questions. Getting answers shouldn’t require a development cycle.

Natural Language as the Query Interface

The Shift: From Dashboards to Conversations

	Traditional Dashboard	Natural Language AI
Query method	Click filters, select dates	Ask in plain English
Time to answer	Minutes to weeks	Seconds
Follow-up questions	New dashboard request	Next sentence
Visualization	Pre-built charts	Generated on demand
Data sources	Pre-configured	Connected via MCP/API
Who can use it	Trained users	Anyone
Maintenance	Ongoing engineering	Prompt updates
Cost per question	High (amortized)	Near zero (marginal)

The dashboard was a *workaround* for the fact that databases don't speak English.
Now they do.

How It Works

AI replaces the entire stack between a business question and a visual answer.

1. **User asks a question** in natural language
2. **AI writes a query** (SQL, Python, API call) based on the data schema
3. **AI executes the query** and retrieves results
4. **AI generates a visualization** (chart, table, or both)
5. **AI explains the results** in plain English
6. **User asks a follow-up** — AI refines, drills down, or pivots

No SQL knowledge. No chart configuration. No waiting for IT. The user *describes what they want to see*, and AI produces it.

What This Looks Like in Practice

The Conversation

- *"Show me monthly revenue by product line for 2025"*
- AI: writes SQL, runs it, produces a grouped bar chart with a summary
- *"Now break out the Enterprise segment by region"*
- AI: refines the query, adds a region dimension, updates the chart
- *"Which region had the biggest drop from Q2 to Q3?"*
- AI: calculates quarter-over-quarter changes, highlights the answer
- *"Put this in a slide deck"*
- AI: generates a PowerPoint with charts and narrative

What AI Did Behind the Scenes

- Read the database schema
- Wrote 4 different SQL queries
- Produced 3 visualizations (matplotlib/plotly)
- Calculated derived metrics
- Generated a narrative summary
- Created a PPTX file

What the User Needed to Know

- What questions to ask
- Whether the answers make sense
- **Nothing else**

Connecting AI to Your Data

Three Ways to Connect AI to Data

Upload Files

- Drag CSV, Excel, PDF into chat
- Works in Chat, Cowork, Code
- No setup required
- Best for: ad-hoc analysis of exported data

"Here's our Q3 P&L. Analyze margin trends."

MCP Connectors

- Connect to databases, APIs, services directly
- Snowflake, BigQuery, Postgres, etc.
- AI queries live data
- Best for: recurring analysis on production data

"Query the sales database for YTD revenue by region."

API Calls

- Financial data providers (FMP, Alpha Vantage, Bloomberg)
- Web scraping via browser
- Internal REST APIs
- Best for: market data, public filings, external sources

"Pull Apple's last 4 quarters and chart revenue growth."

The Database Agent Pattern

The most powerful dashboard replacement is an **AI agent connected to your database**. The agent knows the schema, writes SQL, and presents results visually.

How to Build It

1. Connect database via MCP (or provide a connection string)
2. Give AI the schema: table names, columns, relationships
3. Describe the business context: "revenue is in the orders table, dates are fiscal quarters"
4. Start asking questions

What the Agent Can Do

- Write and execute SQL queries
- Join tables, aggregate, filter
- Compute derived metrics (growth rates, ratios, moving averages)
- Generate charts (bar, line, scatter, waterfall)
- Export to Excel or PowerPoint
- Explain what it found and why it matters

Example: From SQL to Insight in Seconds

User Prompt

“Compare revenue and gross margin by business unit for Q3 2025 vs. Q3 2024. Highlight any unit where margin declined more than 200 basis points. Show it as a table and a chart.”

AI Generates

```
SELECT business_unit,  
SUM(CASE WHEN fiscal_qtr='Q3-2025'  
THEN revenue END) as rev_25,  
SUM(CASE WHEN fiscal_qtr='Q3-2024'  
THEN revenue END) as rev_24, ...  
FROM financials  
GROUP BY business_unit  
ORDER BY margin_change ASC
```

AI Delivers

- Formatted comparison table
- Grouped bar chart (revenue by unit, side-by-side years)
- Margin waterfall chart
- Red flags on units with >200bp decline
- Written summary: “Two units saw margin compression ...”

Follow-up

“Drill into the Hardware unit — is the margin decline driven by pricing or COGS?”

AI-Generated Visualizations

Describe the Chart, Not the Config

Traditional BI tools require you to configure chart type, axes, filters, colors, and labels manually. With AI, you **describe what you want to see** and the visualization is generated.

Natural Language Prompts

- “Bar chart of revenue by quarter”
- “Scatter plot of P/E vs. growth rate for our peer group”
- “Waterfall chart from budget to actual operating income”
- “Heat map of correlations across our portfolio”
- “Time series of daily returns with a 20-day moving average”

AI Handles the Details

- Chooses appropriate chart libraries (matplotlib, plotly, seaborn)
- Sets axis labels, titles, legends
- Applies color schemes
- Handles data transformations
- Adds annotations where relevant
- Iterates if you say “make it cleaner” or “add a trend line”

Interactive Artifacts

Claude's **artifacts** are interactive, publishable applications generated from a single prompt. They go far beyond static charts.

What Artifacts Can Be

- Interactive dashboards with filters and dropdowns
- Calculators (loan amortization, DCF, option pricing)
- Data explorers with sorting and search
- Scenario analysis tools with sliders
- Formatted reports with charts and tables

Key Advantage

- Generated in seconds from a description
- Publishable via a shareable link
- Fully interactive (React under the hood)
- **Disposable**: Generate a new one when needs change instead of maintaining the old one
- No engineering team required

An artifact is a *single-serving dashboard* — built for the question at hand, not for

Artifacts Across Platforms

Every major AI chatbot now generates interactive web applications from a prompt. The feature goes by different names, but the concept is the same.

	Claude Artifacts	ChatGPT Canvas	Gemini Canvas
Technology	React (HTML/CSS/JS)	React (HTML/CSS/JS)	HTML/CSS/JS
Side panel	Yes	Yes	Yes
Publish / share	One-click shareable link	Shareable link	Export or copy
Viewer needs account	No	No	Yes (Google)
Iterative editing	Chat to refine	Inline edits + chat	Chat to refine
Python execution	Separate (code tool)	Built into canvas	Via Colab link
Best for	Interactive apps, calculators, dashboards	Collaborative writing + code editing	Google ecosystem, Sheets/Slides integration

All three can produce the same kinds of interactive finance tools — DCF calculators, portfolio dashboards, scenario analyzers. Choose based on your ecosystem and subscription.

Finance Applications

Replacing Finance Dashboards

The Weekly Operations Review — Before and After

Before: The Dashboard Era

1. Data team pulls exports (2 hours)
2. Analyst builds slides in Excel/PPT (4 hours)
3. Manager reviews and requests changes (1 hour)
4. Analyst revises (2 hours)
5. VP asks a question not on the slide
6. “We’ll get back to you next week”

Total: 9+ hours per week

After: Natural Language AI

1. AI agent connected to database
2. Analyst: *“Generate the weekly ops review”*
3. AI: pulls data, creates charts, writes narrative (3 minutes)
4. Manager reviews, asks for changes in chat
5. VP asks a question not on the slide
6. **AI answers it in 10 seconds**

Total: 15 minutes + live Q&A

This is what HPE’s “Alfred” did. The 90% reduction in prep time wasn’t efficiency — it was **eliminating the translation layer** between data and decisions

Making It Work

When Dashboards Still Win

Natural language AI doesn't replace *all* dashboards. Some use cases still favor traditional BI tools.

Keep the Dashboard

- **Real-time monitoring:** Trading floors, server health, live KPIs on a wall screen
- **Regulatory reporting:** Fixed formats required by regulators (SEC, Basel)
- **Self-service for non-AI users:** Some users prefer clicking to typing
- **Embedded analytics:** Dashboards inside products for customers

Replace with AI

- **Ad-hoc analysis:** One-off questions that arise in meetings
- **Executive reporting:** Weekly/monthly decks assembled from data
- **Variance analysis:** "Why did we miss?" questions
- **Exploratory analysis:** Data exploration, hypothesis testing
- **Client reporting:** Customized reports per client

Ensuring Accuracy

The biggest objection: “How do I know the AI queried the right data?” This is solvable.

- **Show the code:** AI can display the SQL or Python it wrote — analysts can review it
- **Spot-check results:** Compare AI output to a known number from an existing report
- **Schema descriptions:** Tell AI exactly what each table and column means (via system prompts or instructions)
- **Read-only access:** Connect AI to a read replica — it can query but never modify
- **Guardrails:** Restrict which tables AI can access; require approval for queries touching sensitive data
- **Audit trail:** Log every query AI runs for compliance review

The same analysts who currently build dashboards become **AI reviewers** —

Getting Started: A Practical Roadmap

1. **Start with uploaded files:** Export a CSV or Excel from your existing system. Ask AI to analyze it in chat. No integration needed.
2. **Build a schema document:** List your key tables, columns, and business definitions in a markdown file. This becomes AI's "data dictionary."
3. **Connect a read-only database:** Use MCP to connect AI to a read replica or data warehouse. Start with one department's data.
4. **Save your best prompts:** Write down the instructions that produce good results (weekly review, variance analysis, portfolio report). Reuse them each time.
5. **Automate and share:** Later in the course, we'll see how to package these instructions as *skills* and *plugins* that your whole team can invoke with a single command.

Step 1 takes **5 minutes**. Steps 1–3 can be done in **a single afternoon**. You don't need to replace your BI platform — you just stop building new dashboards.

Summary

The Problem

- Dashboards answer fixed questions
- Building them is slow and expensive
- Most are underused
- Follow-ups require a new build cycle

The Solution

- Ask questions in plain English
- AI writes queries and charts
- Follow-ups are instant
- Artifacts for interactive views

How to Start

- Upload a file and ask questions
- Build a schema description
- Connect a database via MCP
- Save and reuse your best prompts

The best dashboard is no dashboard
It's a conversation with your data.