AI-powered recommendation tool: Supplementing ISE 519 (Database Applications in Industrial and Systems Engineering)

by

Akshatha Bopanna Cheppudira

A project submitted to the Integrated Manufacturing

Systems Engineering Department of

North Carolina State University

Raleigh, North Carolina

2025

ABSTRACT

This project aims to develop a proactive, OpenAI-powered recommendation tool to enhance the practical learning experience of ISE 519: Database Applications in Industrial and Systems Engineering, with a focus on SQL and Power BI. Unlike traditional reactive models, the custom GPT will serve as a guided learning companion, generating realistic, industry-relevant datasets that incorporate common data challenges such as fragmentation, incompleteness, and inconsistency. The model will coach students step-by-step through cleaning, integrating, and analyzing these datasets, offering tailored guidance, clarifying questions, and actionable recommendations. By bridging the gap between theoretical coursework and real-world industrial applications particularly in supply chain analytics, manufacturing operations, and process improvement, this tool enables students to gain hands-on experience, apply course concepts effectively, and build career-ready skills.

TABLE OF CONTENTS

1. Overview

In learning technical subjects, the challenge often lies not in finding answers, but in knowing which questions to ask. Until students gain exposure to industry, it is difficult for them to anticipate the types of scenarios and data challenges they might face in real-world applications. This project addresses that gap by developing a proactive, OpenAI-powered custom GPT model to guide students of ISE 519: Database Applications in Industrial and Systems Engineering. Unlike traditional reactive tools that only provide answers, this model actively prompts clarifying questions, suggests areas to explore, and generates realistic, industry-relevant datasets containing common challenges such as fragmentation, incompleteness, and inconsistencies. By providing structured, step-by-step guidance in SQL and Power BI, the tool helps students learn how to approach complex scenarios, practice problem-solving, and apply course concepts effectively. In addition to enhancing guided learning, the tool saves significant time for students by reducing trial-and-error in dataset exploration and analysis, allowing them to focus on building practical, career-ready skills.

The project will be carried out in three phases. Phase I focuses on developing a curated knowledge base of industry-relevant content, including typical use cases, KPIs, SQL queries, and Power BI dashboards. Phase II involves designing and fine-tuning the custom GPT model using both no-code and code-based approaches, ensuring that it can interactively guide students and provide stepwise solutions. Phase III will pilot the tool with students, gather feedback, refine its behavior, and expand the knowledge base to ensure continued relevance and depth in practical learning scenarios.

2. Phase I

Phase I focuses on developing a comprehensive, curated knowledge base that will serve as the foundation for the custom GPT model. This involves collecting industry-relevant content, including typical use cases of SQL and Power BI, as well as commonly used key performance indicators, with an emphasis on supply chain analytics, manufacturing operations, and process improvement. The knowledge base will also feature practice datasets designed to reflect real-world challenges, such as fragmented, incomplete, and inconsistent data. By organizing and structuring this content, Phase I ensures that the GPT model is grounded in accurate, practical, and domain-specific information, enabling it to provide targeted guidance and actionable recommendations to students in subsequent phases.

I. Industry Applications

SQL and Power BI are widely used in industry to extract, integrate, and visualize operational and business data across supply chain, manufacturing, and process improvement domains. SQL handles large, fragmented datasets such as inventory records, production logs, and supplier metrics, while Power BI provides dashboards to monitor KPIs, trends, and anomalies. For this AI project, these real-world use cases will form the foundation of the knowledge base, enabling the custom GPT model to generate realistic datasets and coach students in applying SQL and Power BI to practical scenarios.

i. Supply Chain Management

- Reorder Point Analysis: SQL retrieves sales, lead times, and current stock to calculate when new orders should be placed. Power BI automates alerts and dashboards for planners.

- Safety Stock Optimization: SQL models variability in demand and supply. Power BI visualizes safety stock trade-offs (cost vs. service levels).
- Forecast Accuracy Tracking: SQL compares forecasted demand with actual consumption. Power BI visualizes Mean Absolute Percentage Error (MAPE) trends across SKUs, plants, or regions.
- Obsolescence & Excess Stock: SQL flags low-demand SKUs, while Power BI highlights aging inventory and carrying cost implications for finance teams.
- Order Fulfillment Tracking: SQL queries retrieve order statuses from ERP systems, helping identify late shipments, backorders, and pending approvals. Power BI dashboards visualize fulfillment rates across regions or customer segments.
- Transportation & Logistics: SQL integrates shipment data (carrier performance, freight costs, delivery times). Power BI maps highlight route delays, carrier reliability, and opportunities for consolidation.
- Supplier Management: SQL enables monitoring of supplier lead times, defects, and pricing changes. Power BI dashboards generate supplier scorecards for vendor comparison and contract negotiations.
- Warehouse Operations: SQL queries on warehouse transactions (put-aways, picks, cycle counts) provide granular inventory visibility. Power BI highlights slow-moving vs. fast-moving SKUs, space utilization, and picking efficiency.

ii. Manufacturing Operations

- Machine Utilization & Downtime: SQL queries from MES logs identify machine idle times, setup times, and breakdown frequencies. Power BI visualizes uptime vs. downtime by machine or line.

- Production Throughput: SQL aggregates production volumes per shift, operator, or machine. Power BI dashboards highlight throughput variance against targets.
- Quality Control: SQL queries extract defect logs, rework orders, and scrap counts. Power BI trends defect rates by process, product family, or operator.
- Predictive Maintenance: SQL stores IoT sensor data (temperature, vibration). Power BI visualizes anomalies, enabling proactive maintenance scheduling.
- Workforce Productivity: SQL consolidates labor hours per job or shift. Power BI visualizes operator efficiency, overtime trends, and labor cost contribution per unit.

iii. Process Improvement

- Bottleneck Identification: SQL merges process times from ERP. Power BI highlights which steps delay throughput, enabling line balancing.
- Cycle Time Reduction: SQL captures time stamps for production stages. Power BI visualizes process cycle times against takt time.
- Cost Analysis: SQL integrates raw material costs, scrap, and labor inputs. Power BI highlights cost overruns by department or product line.
- Six Sigma Projects: SQL supports data extraction for DMAIC phases. Power BI creates real-time dashboards to monitor defect reduction progress.
- Kaizen Tracking: SQL maintains continuous improvement logs. Power BI visualizes savings achieved, project timelines, and employee participation.

II. Scenarios and Typical KPIs

A central component of the AI-powered recommendation tool will be a curated repository containing realistic industry scenarios and typical KPI metrics from supply chain, manufacturing operations, and logistics. Often, students know the type of analysis they want to perform but are unsure which SQL queries to write or which KPIs to track in Power BI. The AI tool addresses this gap by providing targeted guidance for each scenario, suggesting the appropriate SQL tasks and highlighting relevant KPIs. By offering step-by-step recommendations, the tool helps students translate their learning objectives into concrete data tasks and actionable insights, enabling hands-on practice with real-world, industry-relevant challenges.

Some examples include:

Inventory Turnover Analysis

SQL Task: Query historical inventory and sales data to calculate turnover ratios.

Power BI KPI Metrics: Inventory Turns, Days of Inventory on Hand, Stockout Frequency, Safety Stock Levels.

On-Time Delivery Tracking

SQL Task: Extract shipment and order data, calculate delivery delays.

Power BI KPI Metrics: On-Time Delivery Rate (OTD), Average Transit Time, Late Shipment Count, Carrier Performance Score.

Demand vs. Supply Analysis

SQL Task: Combine historical demand data with production or supplier data.

Power BI KPI Metrics: Forecast Accuracy, Fill Rate, Backorder Levels, Lead Time Variability.

Overall Equipment Effectiveness (OEE) Monitoring

SQL Task: Aggregate machine uptime, downtime, and production output data.

Power BI KPI Metrics: OEE %, Machine Utilization, Downtime by Machine/Shift, Scrap Rate.

Defect and Quality Tracking

SQL Task: Query production logs to count defects, categorize by type or supplier.

Power BI KPI Metrics: Defect Rate, Parts Per Million (PPM), Supplier Defect Contribution, Rework Percentage.

Production Throughput Analysis

SQL Task: Analyze completed units over time, by production line or shift.

Power BI KPI Metrics: Units Produced per Hour, Cycle Time, Bottleneck Identification, Capacity Utilization.

Freight Cost Optimization

SQL Task: Consolidate carrier invoices, calculate cost-per-mile, identify outliers.

Power BI KPI Metrics: Average Freight Cost per Shipment, Cost Variance by Carrier, Accessorial Charge Frequency, Route Efficiency.

Warehouse Operations and Space Utilization

SQL Task: Extract WMS data on stock location, pick/pack/ship times.

Power BI KPI Metrics: Warehouse Utilization %, Order Picking Accuracy, Cycle Time, Inventory Accuracy.

Supplier Performance Scorecards

SQL Task: Combine supplier delivery, quality, and pricing data.

Power BI KPI Metrics: Supplier On-Time Delivery %, Defect Rate, Cost Variance, Lead Time Compliance.

III. Pain Points

To ensure the AI-powered recommendation tool is not only academically useful but also industry-relevant, research was conducted into common challenges and pain points faced by professionals and alumni when applying SQL and Power BI in real-world settings. This exploration helps bridge the gap between classroom learning and industry practice by identifying where students typically struggle once they enter the workforce.

By translating these pain points into targeted tool features such as providing practice datasets with formatting errors, duplicate entries, and missing values, offering optimized SQL query examples to process and clean the data efficiently, and providing ready-to-use KPI dashboard templates to visualize insights effectively, the AI tutor equips students to tackle the full data workflow encountered in professional environments. In this way, the tool transforms academic exercises into practical, hands-on learning experiences, helping students develop the skills and confidence necessary to analyze, visualize, and draw actionable insights from real-world industrial data.

After researching SQL and Power BI applications in industry settings and gathering feedback from alumni working in supply chain, manufacturing, logistics, and consulting, several clear themes emerged. These insights highlight not only where these tools are applied but also the recurring challenges that limit their effectiveness in practice.

Below are the common pain points with using SQL in industry:

i. Time-Consuming Data Cleaning and Standardization

What the model does: Provides datasets with inconsistent formats (e.g., mixed date formats, irregular naming conventions, missing units).

Student experience: Learns to identify inconsistencies, clean, and standardize data before analysis.

Custom GPT support: If a student gets stuck, the model can suggest cleaning strategies, SQL functions, or stepwise guidance for standardization.

ii. Incomplete or Messy Data

What the model does: Generates datasets with missing values, nulls, or partial history.

Student experience: Learns to handle missing data, impute values, and make forecasting or planning decisions despite gaps.

Custom GPT support: Guides on best practices for dealing with incomplete data, shows examples of imputations, and asks clarifying questions to help students decide on approaches.

iii. Complex Data Integration

What the model does: Provides multiple, interrelated tables with realistic joins, some with mismatched keys or extra irrelevant columns.

Student experience: Learns advanced SQL joins, query optimization, and integration logic.

Custom GPT support: Offers hints, examples, or stepwise walkthroughs for integrating tables and generating meaningful metrics.

iv. Fragmented Systems

What the model does: Supplies data from multiple "systems" (simulated ERP, WMS, MES) with differing formats or KPIs.

Student experience: Practices consolidating fragmented data into a single view for analysis.

Custom GPT support: Advises on merging strategies, consistent KPIs, and ETL best practices.

3. Phase II

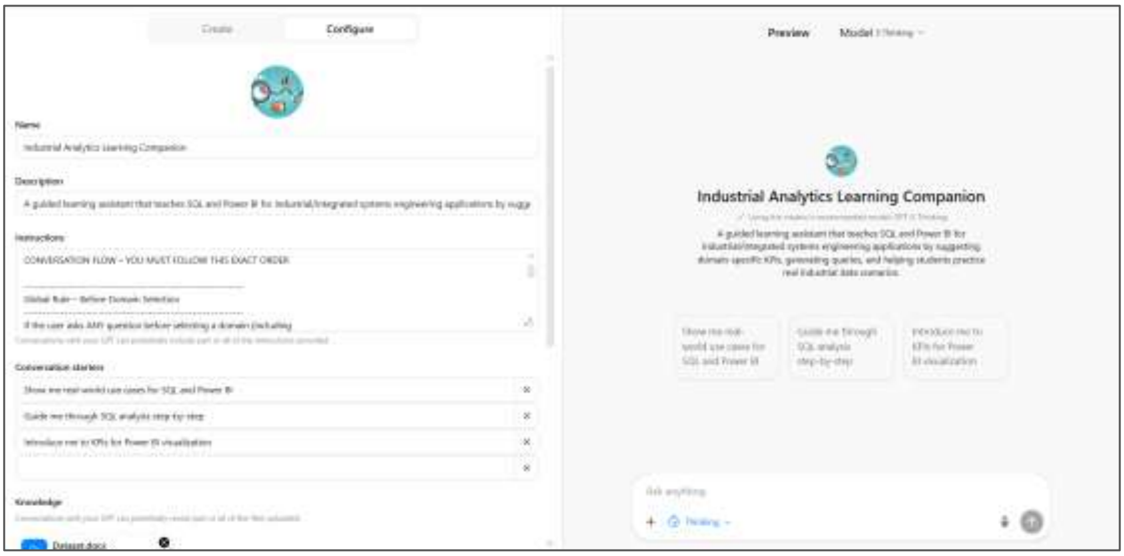
In Phase II, the project focused on building and deploying the interactive AI tool, while continuing to enhance the existing dataset with additional industry scenarios and KPIs. The dataset was further expanded by incorporating examples from multiple functions within each domain such as planning, procurement, inventory, manufacturing, warehousing, and logistics to ensure broader coverage and more realistic, end-to-end analytical use cases. The Custom GPT model titled "**Industrial Analytics Learning Companion**" was developed using OpenAI's framework. The model is designed to guide students through real-world analytical problem-solving by recognizing the type of analysis they intend to perform, identifying relevant KPIs, and suggesting appropriate SQL queries and Power BI approaches. The model can be accessed using the link - https://chatgpt.com/g/g-69325ec27d7481919c698564d8d58cbb-industrial-analytics-learning-companion. The complete project has been uploaded to a dedicated GitHub repository for review, testing, and future development. The repository, accessible at https://github.com/ab03c/Industrial-Analytics-Learning-Companion-/, contains all relevant documentation, configuration files, dataset descriptions, model instructions, and supporting materials required to recreate and evaluate the AI tool.

I. Custom GPT Configuration

The AI tool was configured using OpenAI's Custom GPT framework. The model was built through a guided configuration interface that allowed definition of its instructional role, reasoning behavior, and expected outputs. A sample dataset was uploaded to establish a comprehensive knowledge base, and structured instructions were embedded to ensure accurate interpretation of student prompts. These configurations particularly the instructional logic was iterated multiple times to refine the model's ability to guide students effectively, resulting in a more natural, coherent, and supportive learning experience.

The resulting architecture consisted of three integrated layers:

- **Instruction Layer:** Defines how the model teaches, thinks, and communicates so that students receive clear and consistent guidance.
- **Knowledge Layer:** All datasets and SQL examples were organized as accessible resources that the model could reference.
- **Interaction Layer:** A user-facing interface was enabled, allowing students to explore domains, choose functions and receive SQL queries and KPI recommendations.



II. Analytical Intent Recognition

The model was trained to interpret student queries by identifying the underlying analytical intent rather than relying solely on keywords. The system was configured to recognize common analysis categories such as inventory planning, forecasting, procurement, quality, logistics, warehousing, and manufacturing operations based on the structure and context of the question. Once the intent was detected, the model mapped it to the appropriate KPIs using predefined relationships embedded within the instructional framework.

This analytical intent engine allowed the model to:

- Classify the user's goal.
- Match the goal to the correct domain (supply chain, manufacturing, process improvement etc.).
- Select relevant KPIs and provide a structured, context-aware response.
- Guide the student toward industry-standard performance measures.

III. SQL Query Generation

The AI tool was configured to generate SQL queries that align with the student's analytical intent, chosen difficulty level, and dataset structure. Once the model identified the type of analysis and corresponding KPIs, it translated those requirements into SQL logic by referencing uploaded example queries and adjusting the complexity based on whether the student selected beginner, intermediate, or advanced-level SQL.

Query suggestions were context-specific: for example, an analysis involving inventory turnover prompted calculations using sales and inventory tables, whereas a supplier performance request led to queries involving lead times, defect counts, and delivery records.

The model was also designed to explain the reasoning behind each query, helping students understand why certain clauses, filters, or joins were required. This instructional layer allowed learners not only to use the queries but also to grasp the underlying logic.

IV. Sample Workflow

Example 1

Show me real-world use cases for SQL and Power BI

Hello there! 👋

Let's explore how **SQL and Power BI** are used in real-world business scenarios.

Here's a summary of key **use cases** from the document you uploaded, grouped by domain:

---

1. Supply Chain Management

SQL handles data extraction and logic; **Power BI** turns it into actionable dashboards.

1. **Forecast Accuracy Tracking** – SQL compares forecast vs. actual demand; Power BI visualizes **MAPE** and **bias trends** by region or SKU.
2. **Lead Time Variability Analysis** – SQL calculates differences between expected and actual supplier lead times; Power BI shows variability and risk zones.
3. **Procurement Spend Analytics** – SQL aggregates PO spend by category or vendor; Power BI reveals **maverick spend** and **price trend** insights.
4. **Inventory Health Monitoring** – SQL detects negative stock or high DOH; Power BI creates **inventory risk heatmaps**.
5. **Supplier Scorecards** – SQL merges defect, OTD, and cost data; Power BI ranks suppliers by **scorecards and performance KPIs.**

---

2. Manufacturing Operations

SQL pulls data from MES/ERP; Power BI measures efficiency and throughput.

1. **OEE (Overall Equipment Effectiveness)** – SQL computes availability, performance, and quality; Power BI visualizes **OEE by shift or machine.**
2. **Predictive Maintenance** – SQL stores IoT sensor data; Power BI identifies **anomalies and MTBF trends.**
3. **Cycle Time & Takt Time Analysis** – SQL calculat ↓ :tual vs. standard times; Power BI highlights **bottlenecks and imbalance.**

Would you like me to **focus deeper on one domain** (e.g., Supply Chain, Manufacturing, or Lean Initiatives) — or would you like examples of **SQL queries and Power BI KPIs** for each?

focus on supply chain

Perfect! 🌎 Let's dive into **Supply Chain Management** — one of the richest areas for SQL and Power BI applications.

Before we go deeper — could you tell me what you'd like to explore next?

👉 Would you like **SQL queries**, **Power BI KPIs/dashboards**, or **both** for Supply Chain Management?

(Once you choose, I'll tailor examples like forecast accuracy tracking, supplier scorecards, and inventory optimization accordingly.)

both

Excellent choice! 🚀
Let's explore **Supply Chain Management** with **both SQL queries and Power BI KPIs** — showing how they work together to create insight-driven decisions.

Here are **five high-impact use cases** where SQL + Power BI deliver strong value:

---

1. Forecast Accuracy Tracking

Goal: Measure how well demand forecasts match actual sales.

SQL Logic:

sql Copy code

```sql
SELECT
    SKU,
    Region,
    Forecast_Month,
    SUM(Forecast_Qty) AS Forecast_Qty,
    SUM(Actual_Qty) AS Actual_Qty,
    ABS(SUM(Forecast_Qty) - SUM(Actual_Qty)) / NULLIF(SUM(Forecast_Qty), 0) * 100 AS MAPE
FROM Forecast_vs_Actual
```

+ Ask anything

Power BI KPIs:

- 📊 **MAPE (Mean Absolute Percentage Error)** by SKU/region
- 📈 **Bias %** (over/under forecasting)
- 🕒 **Trend chart** of forecast accuracy over time

Insight: Identifies SKUs or regions with consistent forecasting bias, helping improve S&OP accuracy.

---

2. Lead Time Variability Analysis

Goal: Evaluate how consistent suppliers are with delivery lead times.

SQL Logic:

sql Copy code

```sql
SELECT
    Supplier_ID,
    AVG(Actual_Lead_Time) AS Avg_Lead_Time,
    STDDEV(Actual_Lead_Time) AS Lead_Time_Variability,
    COUNT(*) AS Shipments
FROM Supplier_Deliveries
GROUP BY Supplier_ID;
```

Power BI KPIs:

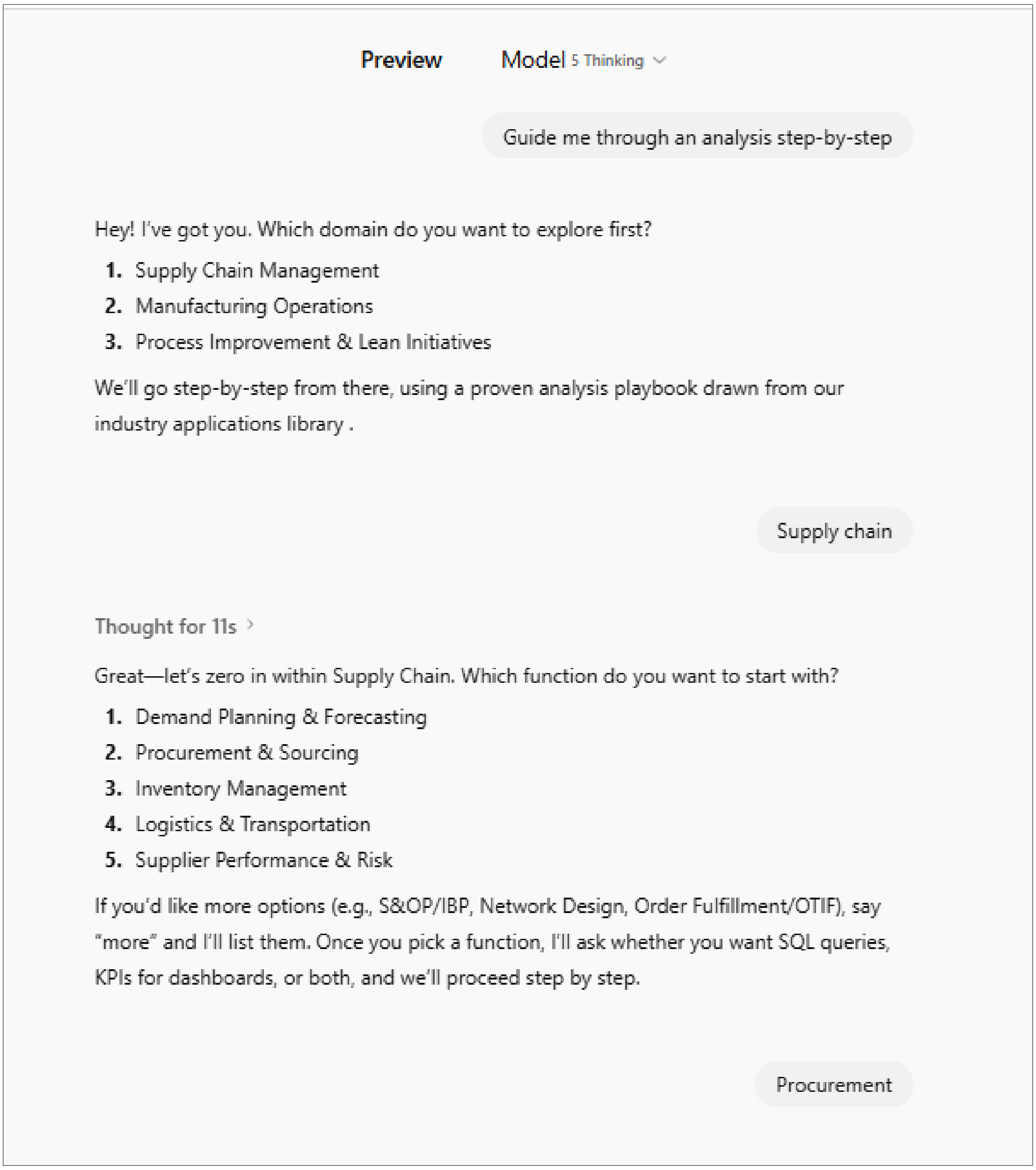
- ⏱ **Average vs. Quoted Lead Time**
- 🔄 **Lead Time Variability %**
- 🚦 **Reliability heatmap** by supplier

Insight: Highlights unreliable suppliers and routes causing production or replenishment delays.

Ask anything

New version of GPT available - Continue chatting to use the old version, or start a new chat for the latest version.

Example 2


Preview
Model 5 Thinking
Guide me through an analysis step-by-step
Hey! I've got you. Which domain do you want to explore first?
1. Supply Chain Management
2. Manufacturing Operations
3. Process Improvement & Lean Initiatives
We'll go step-by-step from there, using a proven analysis playbook drawn from our industry applications library .
Supply chain
Thought for 11s
Great—let's zero in within Supply Chain. Which function do you want to start with?
1. Demand Planning & Forecasting
2. Procurement & Sourcing
3. Inventory Management
4. Logistics & Transportation
5. Supplier Performance & Risk
If you'd like more options (e.g., S&OP/IBP, Network Design, Order Fulfillment/OTIF), say "more" and I'll list them. Once you pick a function, I'll ask whether you want SQL queries, KPIs for dashboards, or both, and we'll proceed step by step.
Procurement

Preview Model 5 Thinking

Awesome—Procurement it is!

Next step: do you want **SQL queries**, **KPIs for dashboards**, or **both**?

Once you choose, I'll walk you through a clear, step-by-step plan tailored to procurement (e.g., spend analysis, supplier performance, PO cycle time), and we'll use your dataset as needed.

SQL

Perfect 👍 Let's walk through Procurement **SQL analysis step-by-step**.

We'll start simple and build up.

🔹 Step 1: Define Your Procurement Analysis Objective

Typical SQL-driven goals include:

1. **Spend Analysis** – understand total spend by supplier, category, or region.
2. **Supplier Performance** – measure delivery timeliness, quality, and cost.
3. **Purchase Order Efficiency** – analyze PO cycle times and approval delays.
4. **Price Variance Tracking** – identify cost differences across suppliers or time.
5. **Compliance & Maverick Spend** – detect off-contract or unauthorized purchases.

👉 Which one would you like to start with?
(You can choose one, or say "suggest the top priority for most companies" and I'll recommend where to begin.)

3

Preview   Model 5 Thinking

Excellent choice — **Purchase Order Efficiency** is one of the most valuable procurement analyses because it reveals process bottlenecks, approval delays, and inefficiencies.

Let's move through this step-by-step 👇

🔹 Step 2: Define Key Metrics for PO Efficiency

We'll focus on metrics that can be derived from your dataset:

1. **Average PO Cycle Time** = `PO_Approved_Date - PO_Creation_Date`
2. **Average Supplier Lead Time** = `PO_Received_Date - PO_Approved_Date`
3. **PO Approval Delay** = `PO_Approved_Date - PO_Submission_Date`
4. **% of Delayed POs** = POs where cycle time > standard threshold
5. **Top Delaying Approvers / Departments** (if approval logs exist)

🔹 Step 3: SQL Query Examples

Assuming a table called `purchase_orders` with fields like:
`po_id`, `supplier_id`, `created_date`, `submitted_date`, `approved_date`, `received_date`, `department`, `status`

Here are some example SQLs 👇

1. Average PO Cycle Time

sql   Copy code

```sql
SELECT
    AVG(DATEDIFF(day, created_date, approved_date)) AS avg_po_cycle_time
FROM purchase_orders
WHERE approved_date IS NOT NULL;
```

Reasoning: Measures how long it typically takes for a PO to go from creation to approval — key for efficiency.

---

2. Average Supplier Lead Time

sql   Copy code

```sql
SELECT
    supplier_id,
    AVG(DATEDIFF(day, approved_date, received_date)) AS avg_supplier_lead_tim
FROM purchase_orders
```

4. Next Steps

Phase II: Model Enhancement

As next steps for Phase II, the model will be enhanced to create an even better learning experience for students. It will simulate **real-world data challenges** such as missing values, fragmented tables, duplicate entries, and inconsistent formats, guiding students step-by-step on how to clean and prepare this flawed data.

The tool will also introduce **quick SQL quizzes** to help learners practice concepts and check their understanding. These interactive elements will reinforce learning and ensure students can apply concepts in practical scenarios.

Additionally, the model will be **fine-tuned by testing it with edge-case scenarios** to evaluate how it responds under unusual or challenging conditions. These improvements will make the tool more practical, reliable, and effective for students learning analytics.

Example:

User: Hi, I'd like a practice dataset.

Custom GPT: Great! Which scenario would you like to practice: Inventory Turnover, On-Time Delivery, Order Fulfillment Tracking, or another Supply Chain scenario?

User: Order Fulfillment Tracking.

Custom GPT: Excellent choice! I've prepared a dataset for Order Fulfillment Tracking. Just a heads-up, this dataset includes some intentional challenges, like missing values in delivery dates, inconsistent order ID formats, and duplicate entries. These are designed to simulate real-world scenarios and help you practice cleaning and analyzing data.

User: Okay, I'll try working on it.

User: I've solved some of it, but I'm stuck on handling the duplicate entries and missing delivery dates.

Custom GPT: No worries! Here's a hint: start by identifying duplicate order IDs using a SQL query with `GROUP BY` or `COUNT`. For the missing delivery dates, consider using conditional logic or imputation techniques depending on what you're trying to measure.

For example, you could run a query like:

sql Copy code

```sql
SELECT OrderID, COUNT(*)
FROM Orders
GROUP BY OrderID
HAVING COUNT(*) > 1;
```

This will highlight duplicates.

For missing delivery dates, you could create a flag column in SQL to mark nulls, or calculate an average delivery time by region to fill in missing values temporarily for analysis.

Once you clean the duplicates and handle missing values, you can calculate KPIs like Order Fulfillment Rate and Average Transit Time, and then visualize them in Power BI.

Phase III: Pilot Testing & Integration

Phase III will focus on pilot testing and integrating the AI tool into the ISE 519 course. The tool will initially be evaluated through the GitHub repository, where students and reviewers can access the model's documentation. This setup allows for transparent testing, issue tracking, and collaborative feedback. A selected group of students will then use the tool in real course activities to assess its usability, clarity, and practical effectiveness in supporting SQL and Power BI learning. Their feedback combined with insights gathered from GitHub-based testing will guide further refinement of the model's recommendations, improve conversation flow, and expand the library of analytical scenarios to ensure the tool aligns closely with course objectives and student needs.