# Interactive Data Visualization with DBMS

Discover how PI2 and NL2INTERFACE are revolutionizing data visualization, making database interactions more accessible and efficient for both technical and non-technical users.

Exploring Retail Sales / E- commerce Data

## Presentation Details

**Presented By:**
Niharika Mysore Gowda, Aamandra Thakur, Mohammed Anwar Sayeed (Group 2)

**Date:** December 5th
**Time:** 4:10 pm - 4:25 pm

Presented by:

# The Power of Data Visualization

## Raw Data to Visual Insights

Data visualization transforms complex datasets into intuitive charts and graphs, enabling quick comprehension and analysis.

## Simplified Database Interactions

PI2 and NL2INTERFACE bridge the gap between technical and non-technical users, making data exploration accessible to all.

# Automation: The Game Changer


Automation in data visualization streamlines the entire process, from query to insight.

# Breaking Down Traditional Barriers

## SQL Expertise No Longer Required

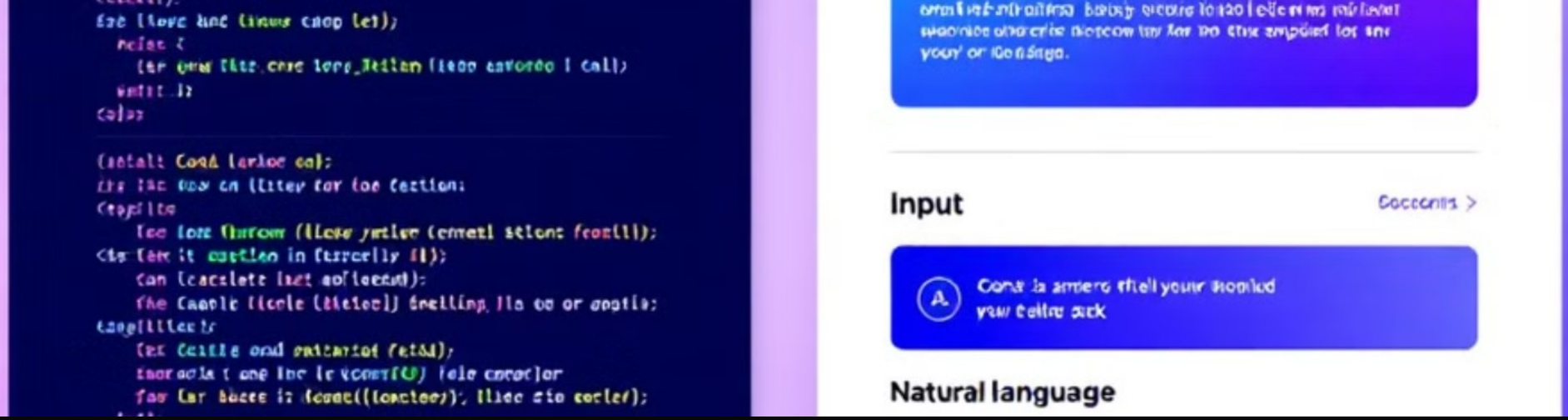
Non-technical users can now access and analyze data without extensive SQL knowledge.

## Minimized Time and Errors

Automated processes reduce the time-intensive nature and potential for errors in manual querying.

## Democratized Data Access

Broader access to data insights empowers decision-making across all organizational levels.




# Introducing PI2 and NL2INTERFACE

## PI2: SQL to Visuals

Transforms SQL queries into detailed visualizations for technical users.

## NL2INTERFACE: Natural Language to Visuals

Converts plain language requests into simple visuals for everyone.

## PI2 (Programming Interface 2)

PI2 is designed for technical users who need precise and detailed visualizations derived from structured SQL queries. It enables users to directly interact with databases and transform query results into complex visual outputs like bar charts, line graphs, and scatter plots.

**How It Works:**

**Input**:
Users write SQL queries to fetch data.

Example:

```
SELECT Region, SUM(SalesAmount) FROM sales GROUP BY Region;
```

**Processing**:
The query is executed using **SQLAlchemy**, which retrieves results directly from the database.

**Output**:
Visualizations are created using libraries like **Matplotlib** or **Plotly**, such as bar charts with regions on the x-axis and sales amounts on the y-axis

**Use Case**:

- A retail analyst queries sales data by region to identify high-performing areas, aiding in inventory optimization and marketing strategies.

**Technical Detail**:

- PI2 relies on SQL commands and generates visualizations in real-time based on the exact query result set.

# NL2INTERFACE (Natural Language to Interface)

NL2INTERFACE caters to non-technical users, allowing them to interact with databases using natural language queries. It removes the need for SQL knowledge by converting plain language inputs into database queries.

**How It Works:**

**Input**:
Users type queries in natural language. Example:
*"What are the total sales by region for the last quarter?"*

**Processing**:
The system uses an NLP engine (e.g., **spaCy**) to convert the natural language input into SQL.
Example SQL Query:

```
SELECT Region, SUM(SalesAmount) FROM sales WHERE date >= '2024-07-01' GROUP BY Region;
```

**Output**:
The SQL query is executed, and the results are visualized using libraries like **Plotly** or **Matplotlib**.

**Use Case**:

- A manager without technical expertise asks for sales data by region to assess performance, receiving a clear bar chart without needing to understand SQL.

**Technical Detail**:

- The NLP engine identifies keywords and maps them to SQL syntax, bridging the gap between plain language and structured data queries.


# PI2 vs NL2INTERFACE: A Closer Look

| Feature | PI2 | NL2INTERFACE |
| --- | --- | --- |
| Input | SQL | Natural Language |
| Output | Detailed Visuals | Simple Visuals |
| Target Users | Technical | Non-Technical |

# Challenges of Traditional Database Querying

**Complexity**: Requires technical expertise in SQL, making it inaccessible for non-technical users.

**Time-Consuming**: Manual query writing and validation extend analysis timelines.

**Error-Prone**: Increased chances of mistakes in query formulation.

**Accessibility Issues**: Significant barriers for users without technical backgrounds.

# Purpose of Automation in Data Visualization

**Enhances Efficiency**: Automates the transformation of database queries into actionable visual insights.

**Reduces Complexity**: Minimizes technical expertise required for data interactions.

**Focus on Analysis**: Shifts the focus from managing technical processes to discovering valuable insights.

# Real-World Applications

## Retail

Analyze sales performance by region and product category to optimize inventory and marketing strategies.

## Marketing

Measure and visualize the effectiveness of ad spend across different channels and campaigns.

## Finance

Track and forecast profit trends, identifying areas for cost reduction and revenue growth.

## Healthcare

Identify patterns in patient outcomes to improve treatment protocols and resource allocation.

# Real-World Applications

**Retail**: Analyze sales performance by region or product category to optimize inventory and marketing strategies.

**Marketing**: Correlate advertising spend with sales growth to evaluate campaign effectiveness.

**Finance**: Visualize trends in revenue, profit, and expenses to identify growth opportunities.

**Healthcare**: Analyze patient data for trends and improve treatment outcomes.

# Comparison of PI2 and NL2INTERFACE

PI2 and NL2INTERFACE serve distinct user groups and purposes, yet both contribute significantly to simplifying data interaction and visualization.

**Target Users**:
PI2 is designed for technical users with expertise in SQL, offering precise control over queries and data visualization. In contrast, NL2INTERFACE caters to non-technical users, allowing them to interact with databases using natural language queries without requiring SQL knowledge.

**Input Style**:
PI2 requires users to write structured SQL queries to interact with the database, which demands familiarity with database schema and SQL syntax. NL2INTERFACE simplifies this by allowing users to provide natural language inputs, which the system translates into equivalent SQL queries.

**Processing**:
PI2 directly processes SQL queries through a framework like SQLAlchemy, ensuring accurate and efficient execution. On the other hand, NL2INTERFACE uses an NLP engine to interpret natural language and generate SQL queries, adding a layer of abstraction.

**Output and Visualizations**:
Both tools generate visual outputs like bar charts, line graphs, and scatter plots using libraries such as Matplotlib or Plotly. However, PI2 provides more customization and precision in its visualizations, whereas NL2INTERFACE focuses on ease of use and accessibility.

**Complexity and Use Cases**:
PI2 excels in handling complex, customized queries and is suited for detailed analytical tasks where precision is key. NL2INTERFACE, while less suited for highly complex queries, is perfect for straightforward data exploration, making data insights accessible to users with no technical background.

**Accessibility**:
PI2 appeals to users who are comfortable with technical tools and SQL, while NL2INTERFACE democratizes data access by enabling non-technical users to engage with data effortlessly.

# Literature Review

Our literature review focused on advancements in interactive data visualization. PI2 automates the generation of visualizations from structured queries, streamlining the data analysis process. Meanwhile, NL2INTERFACE allows users to query databases using natural language, enabling a broader range of users to interact with data without needing to write complex queries. Both systems address the need for more intuitive data access and provide tools for simplifying complex database interactions. PI2 offers a technical solution with its automated query-to-visualization features, while NL2INTERFACE facilitates visualization creation using natural language inputs. Together, these systems make data visualization more accessible, even to non-experts, democratizing data insights across various domains

## References


Shneiderman, B., & Plaisant, C. (2009). *Designing the User Interface: Strategies for Effective Human-Computer Interaction.*

Heer, J., & Shneiderman, B. (2012). *Interactive Dynamics for Visual Analysis.* Communications of the ACM.

Keim, D. A., Mansmann, F., & Schneidewind, J. (2008). *Visual Analytics: Methods and Tools.* Proceedings of the International Conference on Data Mining

# Importance for Technical and Non-Technical Users

**Technical Users**:

- Tools like PI2 accelerate the data analysis process by converting complex SQL queries into instant, customizable visualizations.

- Enables data scientists to focus on advanced problem-solving rather than routine tasks.

**Non-Technical Users**:

- NL2INTERFACE allows natural language querying, empowering users to explore data independently without requiring SQL expertise.

**Goal**:

- Promote organization-wide data democratization.
- Foster a data-driven culture by enabling accessible and intuitive data interaction.

# System Architecture and Installation

**Database Layer**:

- Relies on PostgreSQL or MySQL for reliable data organization and retrieval.

**Backend Processing**:

- Python with SQLAlchemy ORM for efficient query execution and database abstraction.

**NLP Processing**:

- Combines spaCy with Hugging Face Transformers to transform natural language queries into SQL.

**Visualization Layer**:

- Uses Matplotlib and Seaborn for creating interactive charts and dashboards.

**API Interface**:

- Flask or Django serves as the communication bridge between visualization components and backend systems.

**Installation Requirements**:

- Python 3.7+, PostgreSQL/MySQL, Flask/Django, spaCy, and Hugging Face Transformers.
- Industry-standard libraries like Matplotlib and Seaborn for visualization.

**Setup Instructions**:

- Create a Python virtual environment.

Use a requirements.txt file for package installation via pip.

- Update configuration files with database credentials and API keys.

# Usability Testing and Findings

**Experiment Plan Overview**:

- Evaluates PI2 and NL2INTERFACE against traditional SQL querying methods based on task completion time, accuracy, and user satisfaction.

**Metrics for Success**:

**Efficiency**: Reduction of time from query to visualization by 50% or more.

**Accuracy**: Achieving 95% or higher precision in query interpretation.

**User Satisfaction**: Aiming for 4.5/5 or higher in usability surveys.

**Adaptability**: Successfully processing at least 90% of queries across diverse database schemas.

**Initial Results**:

- Non-technical users completed tasks 45% faster with NL2INTERFACE compared to traditional SQL methods.
- High user satisfaction with intuitive interfaces and rapid visualizations.

# Conclusion and Future Directions

**Expanding NLP Capabilities**:

- Handle more complex query structures with greater precision.

**Advanced Visualization Options**:

- Include predictive analytics and dynamic dashboards.

**Performance Enhancements**:

- Leverage machine learning for query optimization and faster processing.

By continuing to bridge the gap between technical and non-technical users, PI2 and NL2INTERFACE will drive a robust data-driven culture in organizations.

# System Architecture: The Four Layers

1

## Database Layer

Manages data in PostgreSQL/MySQL

2

## Backend Layer

Processes queries with Python

3

## NLP Layer

Converts language to SQL using spaCy

4

## Visualization Layer

Creates graphs using Matplotlib

# Bringing Data to Life: The Visualization Layer

1

## Data Input

Raw data from database queries

2

## Processing

Preparation

3

## Visualization

Creation of charts using Matplotlib/Pyplot

4

## Output

Clear, actionable visual insights

# Seeing is Believing: A Live Demo

1. **User Input**

   User asks: "What are last month's sales?"

2. **Query Conversion**

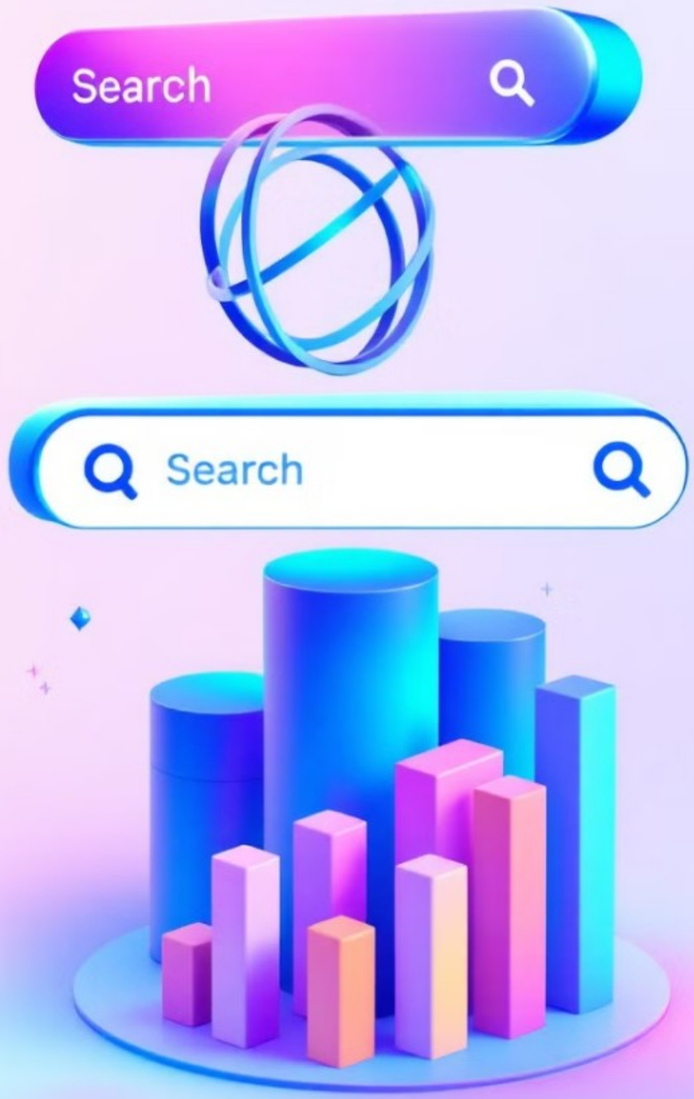
   NL2INTERFACE translates the question into a SQL query.

3. **Data Retrieval**

   The system fetches relevant data from the database.

4. **Visualization**

   An interactive sales trend graph is automatically gen



# Demonstration of Use Cases

**Scenario 1**:

*"Show total sales by region for the last quarter."*

**PI2**:

**SQL Input**: SELECT Region, SUM(SalesAmount) FROM sales WHERE date >= '2024-07-01' GROUP BY Region;

**Output**: A bar chart with regions on the x-axis and total sales on the y-axis

**NL2INTERFACE**:

**Natural Language Input**:

*"Show total sales by region for the last quarter."*

**Processing**: Translated to the SQL query above

**Output**: The same bar chart as PI2, generated from natural language input.

## Scenario 2:

*"Show the correlation between advertising spend and sales growth over the past year."*

**PI2**:

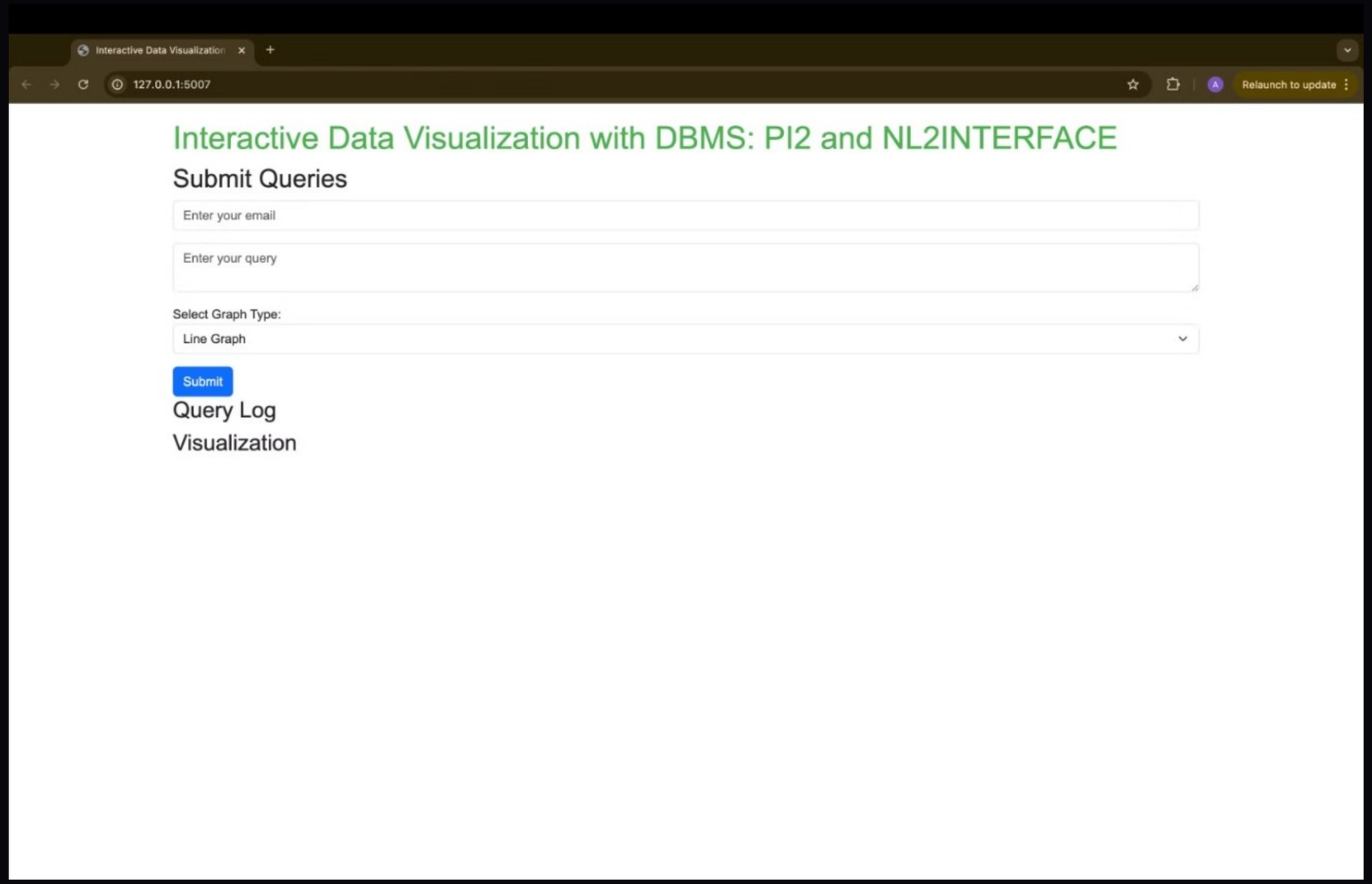
**SQL Input -** SELECT AdvertisingSpend, SUM(SalesAmount) FROM sales WHERE date >= '2023-01-01' GROUP BY AdvertisingSpend;

**Output**: A scatter plot with advertising spend on the x-axis and total sales on the y-axis

**NL2INTERFACE**:

**Natural Language Input**:
*"Show the correlation between advertising spend and sales growth over the past year."*

**Processing**: Recognizes keywords like "correlation," "advertising spend," and "sales growth," translating the input into the SQL query above.



Interactive Data Visualization

127.0.0.1:5007

Relaunch to update

abc@gmail.com

Total sales by region

Select Graph Type:

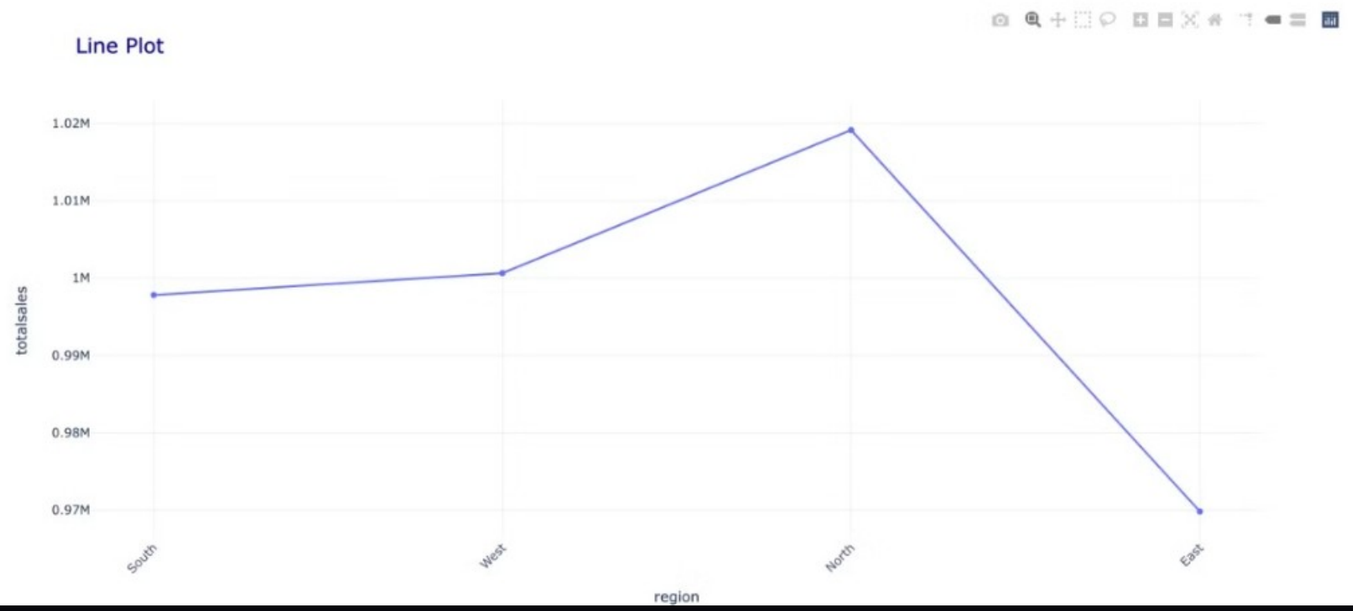
Line Graph

Submit

## Query Log

Email: abc@gmail.com, Query: Total sales by region

## Visualization



Interactive Data Visualization

127.0.0.1:5007

Relaunch to update

abc@gmail.com

Total sales by region

Select Graph Type:

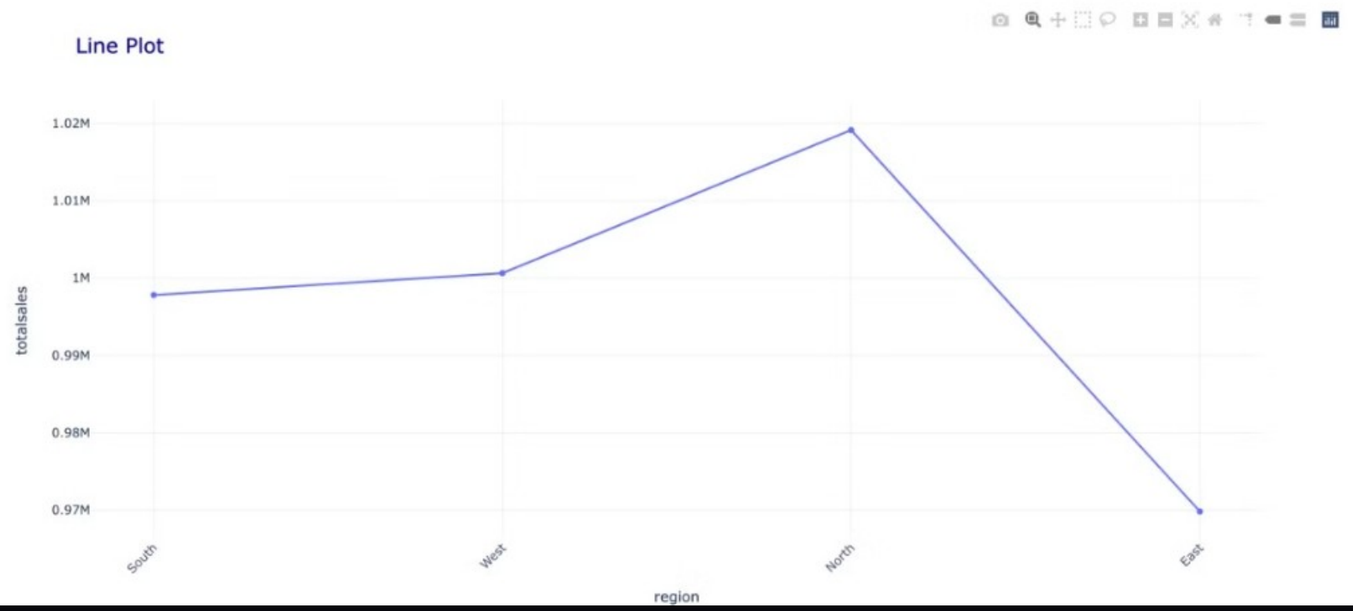
Line Graph

Submit

## Query Log

Email: abc@gmail.com, Query: Total sales by region

## Visualization



Interactive Data Visualization

127.0.0.1:5007

Relaunch to update

abc@gmail.com

quarterly sales analysis for the past year

Select Graph Type:

Line Graph

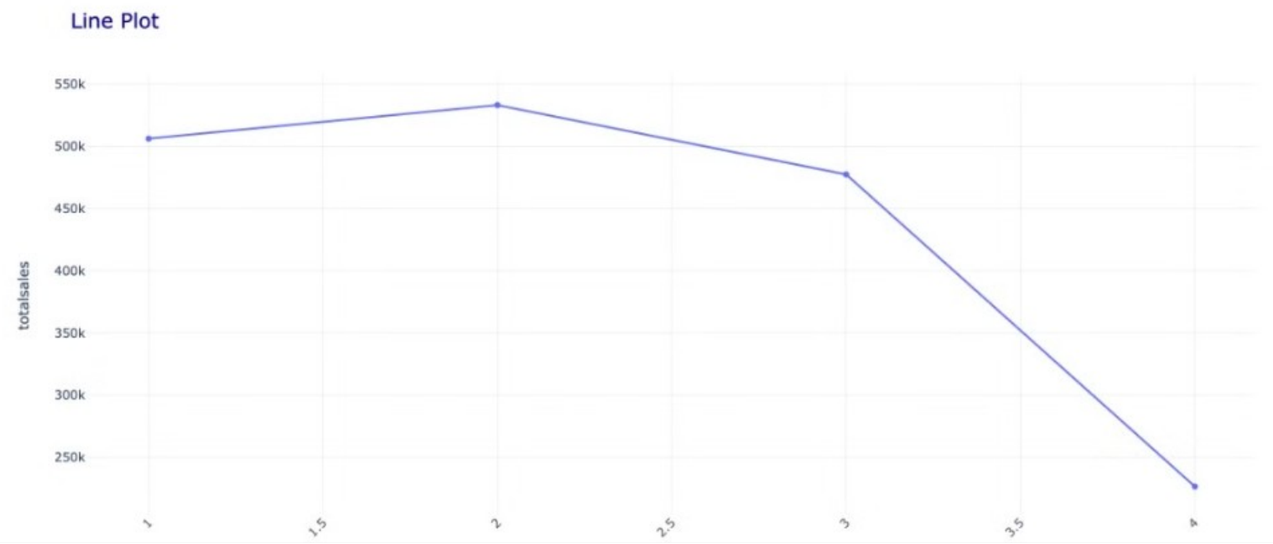
Submit

## Query Log

Email: abc@gmail.com, Query: Total sales by region

Email: abc@gmail.com, Query: quarterly sales analysis for the past year

## Visualization



# That's a Wrap!

Thanks for joining us on this journey through data visualization

We'd love to hear your thoughts and answer any questions you may have