SQL_Tracker: Real-Time Data Lineage Tracker - Report

Overview

SQL_Tracker is a Python-based tool designed to assist data engineers by dynamically tracking and visualizing SQL query transformations. It monitors data lineage, capturing and displaying how data evolves at both row and column levels. This tool provides real-time insights into the effects of operations like filtering and aggregation, empowering data professionals to better understand and manage data flows.

Features

1. Intercept SQL Queries

  - Identifies key SQL clauses (SELECT, WHERE, GROUP BY) in user input, ensuring systematic query analysis.

2. Track Table Transformations

  - Logs source-to-target table transformations:

   - WHERE clauses are logged as filtered tables.

   - GROUP BY clauses are logged as aggregated tables.

3. PrettyTable-Based Visualization

  - Uses PrettyTable to show data transformations before and after operations, allowing quick visualization of query impacts.

4. Dynamic Query Execution

  - Operates in a feedback loop, capturing new SQL queries, updating lineage maps, and offering real-time feedback on table transformations.

How SQL_Tracker Works

SQL_Tracker: Real-Time Data Lineage Tracker - Report

1. Query Interception

   - Parses and formats SQL queries with sqlparse to ensure readability and systematic analysis.

2. Lineage Tracking

   - Extracts source tables and columns, logs transformations, and dynamically creates new tables (e.g., filtered or aggregated tables).

3. Data Visualization

   - Displays data transformations in table format using PrettyTable, enabling users to compare data before and after SQL operations.

4. Real-Time Query Input

   - Supports continuous input of new SQL queries, ideal for exploratory data engineering tasks requiring real-time feedback.

Technology Dependencies

1. Python

   - Provides the flexibility needed for SQL processing and data visualization.

2. sqlparse

   - Used for parsing, formatting, and analyzing SQL queries, ensuring statements are clear and easy to interpret.

3. PrettyTable

   - Facilitates clear tabular display of data transformations, making the effect of SQL operations easy to understand.

4. time (Standard Library)

   - Adds small delays between query executions to simulate real-time feedback, supporting smooth, sequential query processing.

Use Cases for Data Engineers

1. Monitor Data Lineage in Real-Time

   - Tracks how raw data transforms through various stages, enhancing data quality and supporting troubleshooting.

2. Understand Table Dependencies

   - Offers insights into the transformation linkages between tables, essential for managing complex data pipelines.

3. Quick Data Visualization

   - Allows rapid visualization of SQL transformations using PrettyTable, reducing the need to manually execute queries.

4. Support for Dynamic Workflows

   - Real-time input loop supports experimenting with multiple transformations and instantly viewing their impact.

Conclusion

SQL_Tracker is a powerful tool for tracking SQL query transformations and visualizing data lineage. It provides valuable insights into table dependencies and query impacts, making it easier for data engineers to manage pipelines, maintain data quality, and troubleshoot issues. Its real-time query processing and clear data visualization make it an essential tool for anyone working with SQL-based data transformations.