Project Report: Social Media Analytics Backend

Introduction

The primary goal of this project was to design and build a functional SQL backend for a social media analytics platform. The system needed to effectively store and manage data related to users, their posts, and engagement metrics such as likes and comments. The objective was to create a structured database capable of tracking user interactions, analyzing content trends through features like hashtag tracking, and generating insightful analytical reports on post performance.

Abstract

This project successfully implemented a social media analytics database using PostgreSQL. A normalized schema was designed with core tables for `Users`, `Posts`, `Comments`, and `Likes`. The database was populated with realistic sample data to simulate user activity. Key features include an automated trigger that updates post like counts in real-time and a hashtag tracking system built on a many-to-many relationship, allowing for content categorization and trend analysis. Finally, SQL views were created to simplify complex queries, and window functions were used to analyze the data and generate a ranked leaderboard of posts based on engagement.

Tools Used

- **Database Management System:** PostgreSQL
- **Database Administration Tool:** pgAdmin 4

Steps Involved

The project was executed in a series of logical steps to ensure a robust final product:

1. **Schema Design**: The initial phase involved designing a relational database schema. Four tables (`Users`, `Posts`, `Likes`, `Comments`) were modeled to store essential information and their relationships were defined using primary and foreign keys. A `UNIQUE` constraint was added to the `Likes` table to prevent duplicate entries.
2. **Data Population**: After creating the table structures, the database was populated with sample data to simulate a live environment. This included creating users, posts, and various interactions like comments and likes to provide a foundation for testing and analysis.
3. **Automation with a Trigger**: To ensure the `like_count` on posts was always accurate, a trigger was implemented. This PostgreSQL trigger automatically executes a function after every `INSERT` on the `Likes` table, recounting the likes for the relevant post and updating the `Posts` table. This eliminated the need for manual updates and ensured data consistency.

4. **Implementing Hashtag Tracking**: To add a layer of content analysis, a hashtag tracking system was implemented. This required creating two new tables, `Hashtags` and `Post_Hashtags`, to establish a many-to-many relationship between posts and tags. This feature allows for querying posts by specific hashtags, enabling trend analysis.
5. **Analysis with Views and Window Functions**: To simplify reporting, a SQL `VIEW` named `PostEngagement` was created to provide a consolidated look at posts and their engagement metrics. Using this view, a final analytical query was written using the `RANK()` window function to create a ranked leaderboard of posts, ordered by the highest engagement.

Conclusion

This project successfully demonstrates the creation of a complete backend system for social media analytics using SQL. Through structured design, automation, and advanced querying, a scalable and efficient database was developed. Key takeaways include a practical understanding of database normalization, the power of triggers in maintaining real-time data accuracy, the implementation of many-to-many relationships for complex features, and the utility of views and window functions for performing insightful data analysis. The resulting system can effectively track and report on user engagement and content trends, fulfilling all initial project objectives.