Database Project

Project Name: Social Media Database with Sentiment Analysis

Team Members (AID SEC 3)

Ahmed Mohamed	320230110
Shahd Ibrahim	320230100
Monay Mohamed	320230097
Marc Wassim	320230095
Abdelrahman Essam	320230120

Doctor : Mohamed Essa

Teaching Assistant : Sama Osama

1. Introduction

The rapid growth of social media platforms has resulted in massive amounts of user-generated content being created and shared every second. While this growth enables communication and community building, it also introduces significant challenges related to content moderation, user behavior analysis, and the detection of harmful or toxic interactions.

This project presents the design and implementation of a Social Media Database with Sentiment Analysis, aimed at efficiently managing social media data while providing intelligent insights into user engagement and emotional trends. The system stores and organizes user profiles, posts, comments, reactions, and interactions in a structured relational database.

In addition to basic social media functionalities, the platform integrates sentiment analysis techniques to automatically evaluate the emotional tone of user-generated content. This enables administrators and moderators to identify toxic content, monitor community health, track trending discussions, and support data-driven moderation and decision-making processes. The system is designed with scalability, performance, and data integrity in mind, making it suitable for large-scale social networking environments

2. Business Understanding

2.1 Problem Statement

Modern social media platforms face increasing challenges in managing large volumes of user-generated content. Manual moderation is often inefficient, time-consuming, and prone to human bias, especially with millions of posts and comments generated daily.

Additionally, the lack of structured analytical tools makes it difficult for platform administrators to understand user engagement patterns, detect harmful or toxic behavior, and identify emerging trends in public sentiment. Without automated sentiment analysis and centralized data management, maintaining a safe and healthy online environment becomes a complex task.

2.2 Business Goals

The primary goal of this system is to provide a robust and scalable database solution that supports both social media operations and advanced sentiment-based analytics. The system aims to:

• Enable efficient storage and management of users, posts, comments, and interactions.
• Automatically analyze user-generated content to classify sentiment and detect toxic behavior.
• Support data-driven content moderation by flagging harmful content for administrative review.
• Provide analytical insights into user engagement, trending topics, and overall community sentiment.
• Assist platform administrators in monitoring platform growth, activity patterns, and community health.

3. Stakeholders

3.1 Primary Stakeholders

End Users:
Regular users who create posts, comments, and interact with content. They benefit from a safe platform with improved content moderation and personalized engagement.

Platform Administrators:
Responsible for managing the platform, monitoring content, analyzing trends, and making strategic decisions based on system analytics.

Moderators:
Review flagged or toxic content identified by sentiment analysis tools and take appropriate moderation actions to maintain platform safety.

3.2 Secondary Stakeholders

Data Analysts:
Utilize engagement metrics and sentiment statistics to analyze user behavior and platform performance.

Compliance and Governance Teams:
Ensure that platform operations comply with legal regulations, ethical standards, and data protection policies.

Marketing and Growth Teams:
Analyze trends, popular content, and influencer activity to support platform growth and user engagement strategies

4. Functional Requirements

The system shall provide the following functional capabilities:

4.1 User Management

- The system shall allow users to register, authenticate, and manage their accounts.
- The system shall store user profile information including username, bio, profile picture, and location.
- The system shall allow users to update personal details and privacy preferences.
- The system shall support cross-platform profiles for users across different social media platforms.
- The system shall allow administrators to suspend, deactivate, or delete user accounts when necessary.

4.2 Post Management

- The system shall allow users to create, edit, and delete posts containing text, images, or videos.
- The system shall store post metadata such as creation timestamp, visibility, and engagement counters.
- The system shall support tagging users and adding hashtags to posts.
- The system shall associate posts with topics, locations, or campaigns when applicable.
- The system shall automatically trigger sentiment analysis upon post creation.

4.3 Comment and Reply Management

- The system shall allow users to comment on posts.
- The system shall support nested replies through recursive comment relationships.
- The system shall allow users to edit or delete their comments.
- The system shall apply sentiment analysis to comments for toxicity detection.

4.4 Reaction and Interaction Management

- The system shall support multiple reaction types (like, love, angry, sad, etc.) for posts and comments.
- The system shall prevent duplicate reactions from the same user on the same target.
- The system shall track total reactions and interaction metrics for analytical purposes.
- The system shall record interactions such as likes, mentions, shares, follows, and blocks.

4.5 Relationship Management

- The system shall allow users to follow and unfollow other users.
- The system shall prevent duplicate follow or block relationships.
- The system shall allow users to block other users to restrict interactions.
- The system shall store follow and block relationships for analytics and moderation.

4.6 Sentiment Analysis and Content Moderation

- The system shall analyze the sentiment of posts and comments using NLP techniques.
- The system shall classify content into positive, negative, neutral, or toxic categories.
- The system shall automatically flag potentially harmful content.
- The system shall allow moderators to review flagged content and take moderation actions.
- The system shall maintain a history of sentiment analysis results for trend tracking.

4.7 Notifications and Reporting

- The system shall notify users of new comments, reactions, mentions, or followers.
- The system shall alert moderators about flagged or toxic content.
- The system shall generate analytical and moderation reports for administrators.
- The system shall support exporting reports for compliance or auditing purposes.

4.8 Analytics and Trend Detection

- The system shall track engagement metrics over time (posts/day, reactions/hour, etc.).
- The system shall identify trending hashtags, posts, and users.

- The system shall aggregate sentiment statistics by user, topic, hashtag, or campaign.
- The system shall provide dashboards and analytical views for administrators.

5. Non-Functional Requirements

The system shall satisfy the following non-functional requirements:

5.1 Scalability

- The system shall support millions of users, posts, and interactions.
- The database design shall allow horizontal scaling to handle high traffic volumes.
- The system shall efficiently manage time-series analytical data.

5.2 Performance

- The system shall respond to user actions (posting, commenting, reacting) within acceptable response times.
- Sentiment analysis and moderation tasks shall be executed asynchronously to avoid performance degradation.
- Frequently accessed attributes (user_id, post_id, hashtag, timestamps) shall be indexed for fast querying.

5.3 Security

- The system shall securely store encrypted passwords and sensitive user data.
- The system shall enforce role-based access control for administrators and moderators.
- The system shall sanitize user inputs to prevent SQL injection and XSS attacks.
- The system shall log critical system and moderation activities for auditing purposes.

5.4 Reliability and Data Integrity

- The system shall maintain consistency across all user interactions and sentiment results.
- The system shall prevent duplicate or invalid relationships (follows, reactions, blocks).
- The system shall support backup and recovery mechanisms to prevent data loss.

5.5 Usability

- The system shall provide an intuitive interface for posting, browsing, and interactions.
- Analytical dashboards shall be clear and easy to interpret.
- Moderation tools shall allow efficient review and decision-making.

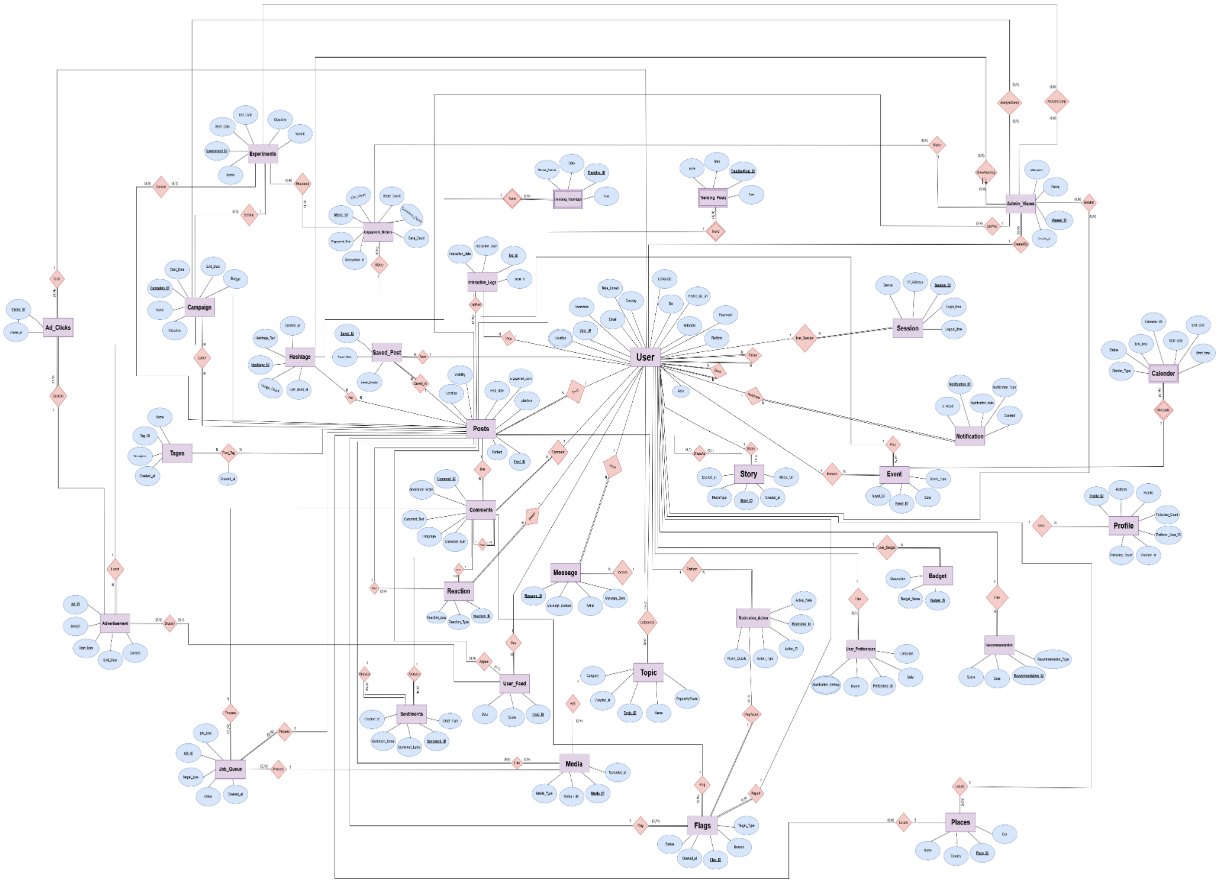
5.6 Maintainability

- The system shall be modular to support future feature extensions.
- The sentiment analysis engine shall be replaceable or upgradable without affecting core functionality.
- The system shall use APIs for integration with external NLP models.

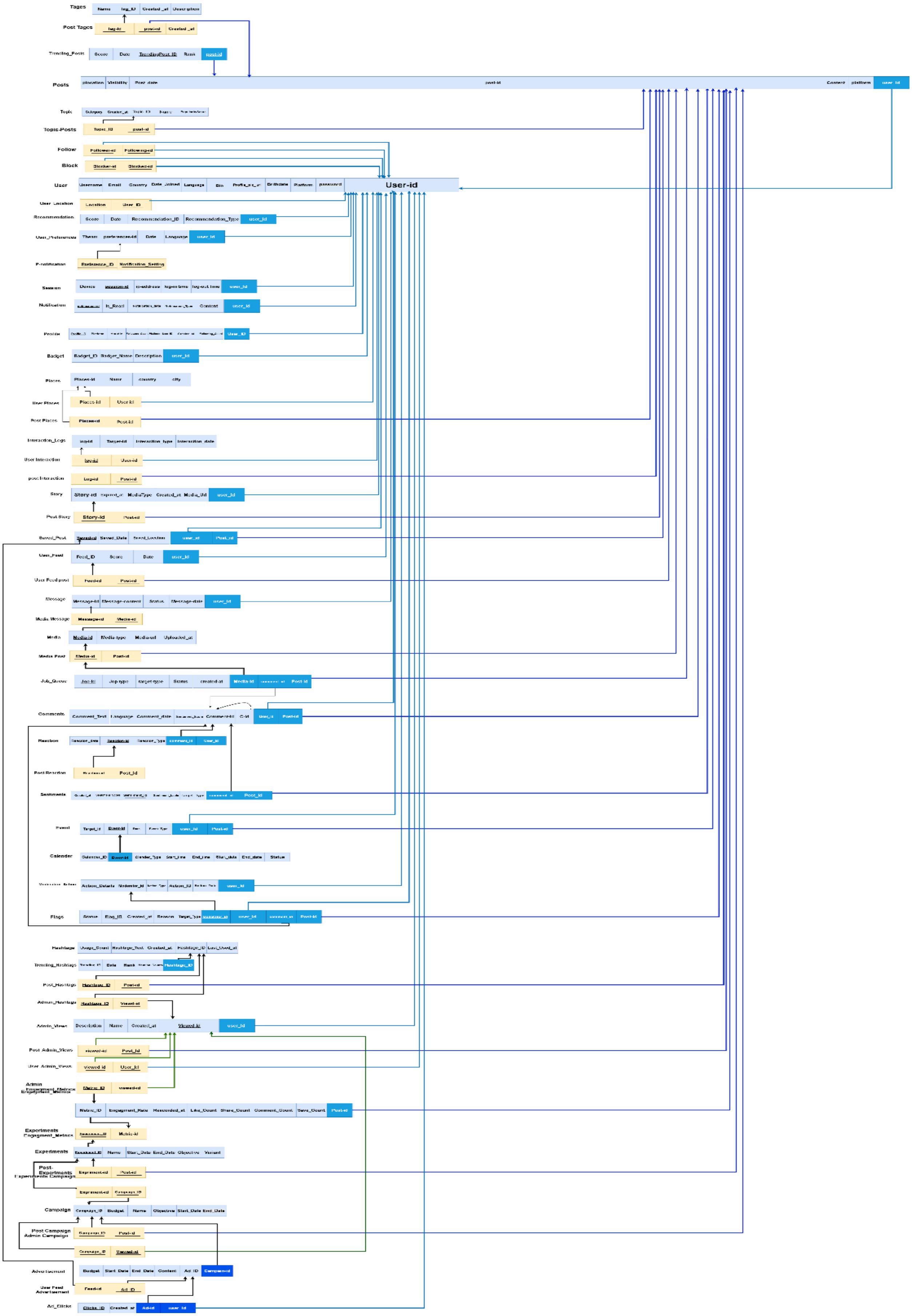
5.7 Ethical and Privacy Compliance

- The system shall respect user privacy and data protection regulations.
- Users shall have control over their personal data and privacy settings.
- Automated moderation decisions shall be reviewable by human moderators.

6. Conceptual Design (ERD)



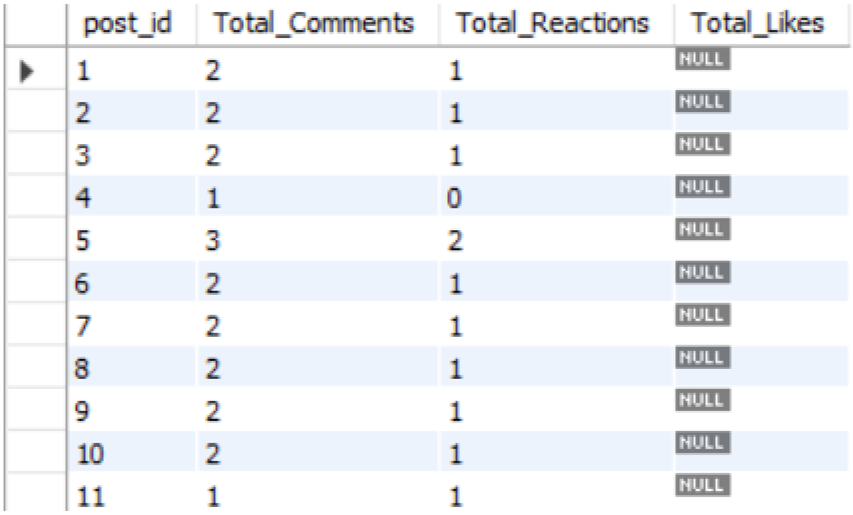
7. Logical Design & Mapping


Tages
Name
tag_ID
Created _at
Description
Post Tages
tag-id
post-id
Created _at
Trending_Posts
Score
Date
TrendingPost_ID
Rank
post-id
Posts
plocation
Visibility
Post_date
post-id
Content
platform
user_id
Topic
Category
Created_at
Topic_ID
Name
Topic-Posts
Topic_ID
post-id
Follow
Follower-id
Following-id
Block
Blocker-id
Blocked-id
User
Username
Email
Country
Date_Joined
Language
Bio
Profile_pic_url
Birthdate
Platform
password
User-id
User_Location
Location
User_ID
Recommendation
Score
Date
Recommendation_ID
Recommendation_Type
user_id
User_Preferences
Theam
preferences-id
Date
Language
user_id
P-notification
Preference_ID
Notification_Setting
Session
Device
session-id
ip-address
log-in time
log-out time
user_id
Notification
Is_Read
Content
user_id
Profile
User_ID
Badget
Badget_ID
Badget_Name
Description
user_id
Places
Places-id
Name
country
city
User Places
Places-id
User-id
Post Places
Places-id
Post-id
Interaction_Logs
log-id
Target-id
Interaction_type
Interaction_date
User Interaction
log-id
User-id
post Interaction
Log-id
Post-id
Story
Story-id
Expired_at
MediaType
Created_at
Media_Url
user_id
Post Story
Story-id
Post-id
Saved_Post
Saved-id
Saved_Date
Saved_Location
user_id
Post_id
User_Feed
Feed_ID
Score
Date
user_id
User Feed post
Feed-id
Post-id
Message
Message-id
Message-content
Status
Message-date
user_id
Media Message
Message-id
Media-id
Media
Media-id
Media-type
Media-url
Uploaded_at
Media Post
Media-id
Post-id
Job_Queue
Job-id
Job-type
target-type
Status
created-at
Media-id
Post-id
Comments
Comment_Text
Language
Comment_date
Comment-id
C-id
User_id
Post-id
Reaction
Reaction_date
Reaction-id
Reaction_Type
comment_id
User_id
Post Reaction
Post_Id
Sentiments
Sentiment_ID
Post_Id
Event
Target_id
Event-id
Event Type
user_id
Post-id
Calender
Calender_ID
Event-id
Clender_Type
Start_time
End_time
Start_date
End_date
Status
Action_Details
Moderator_Id
Action_ID
user_id
Flags
Status
Flag_ID
Created_at
Reason
Target_Type
user_id
comment_id
Post-id
Hashtags
Usage_Count
Hashtags_Text
Created_at
Hashtags_ID
Last_Used_at
Trending_Hashtags
Date
Rank
Hashtags_ID
Post_Hashtags
Hashtags_ID
Post-id
Admin_Hashtags
Hashtags_ID
Viewed-id
Admin_Views
Description
Name
Created_at
Viewed-id
user_id
Post_Admin_Views
viewed-id
Post_Id
User_Admin_Views
viewed id
User_Id
Admin
Engagment_Metrics
Metric_ID
viewed-id
Metric_ID
Engagment_Rate
Recorded_at
Like_Count
Share_Count
Comment_Count
Save_Count
Post-id
Experiments
Engagment_Metrics
Metric-id
Experiments
Name
Start_Date
End_Date
Objective
Variant
Post-
Experiments
Experiments Campaign
Expriment-id
Post-id
Expriment-id
Campaign_ID
Campaign
Campaign_ID
Budget
Name
Objective
Start_Date
End_Date
Post Campaign
Admin Campaign
Campaign_ID
Post-id
Campaign_ID
Viewed-id
Advertisement
Budget
Start_Date
End_Date
Content
Ad_ID
Campaign-id
User Feed
Advertisement
Feed-id
Ad_ID
Ad_Clicks
Clicks_ID
Created_at
Ad-id
user_id

8. Queries & Relational Algebra

8.1 -- Summary of All Posts and Their Engagement Metrics

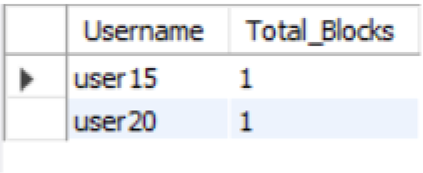
```
SELECT p.post_id,
       COUNT(DISTINCT c.Comment_id) AS Total_Comments,
       COUNT(DISTINCT r.Reaction_id) AS Total_Reactions,
       SUM(em.Like_Count) AS Total_Likes
FROM Posts p
LEFT JOIN Comments c ON p.post_id = c.Post_id
LEFT JOIN Post_Reaction r ON p.post_id = r.Post_id
LEFT JOIN Engagement_Metrics em ON p.post_id = em.Post_ID
GROUP BY p.post_id;
```



post_id	Total_Comments	Total_Reactions	Total_Likes
1	2	1	NULL
2	2	1	NULL
3	2	1	NULL
4	1	0	NULL
5	3	2	NULL
6	2	1	NULL
7	2	1	NULL
8	2	1	NULL
9	2	1	NULL
10	2	1	NULL
11	1	1	NULL

8.2 -- Users Who Have Blocked Others

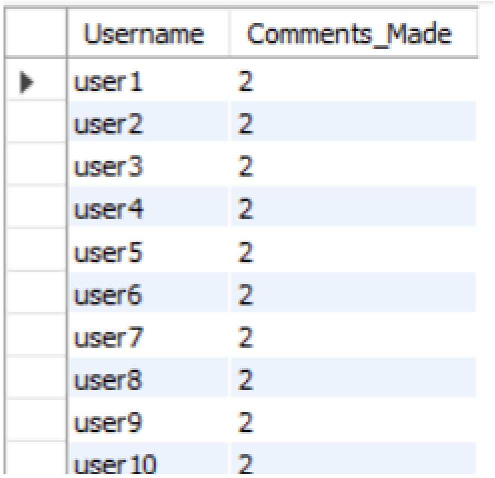
```
SELECT u.Username, COUNT(b.Blocked_id) AS Total_Blocks
FROM Users u
JOIN Block b ON u.User_id = b.Blocker_id
GROUP BY u.Username
ORDER BY Total_Blocks DESC;
```



Username	Total_Blocks
user15	1
user20	1

8.3 -- Top Commenters

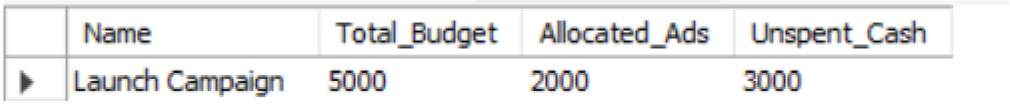
```
SELECT U.Username, COUNT(C.Comment_id) AS Comments_Made
FROM Users U
JOIN Comments C ON U.User_id = C.user_id
GROUP BY U.Username
ORDER BY Comments_Made DESC
LIMIT 10;
```



Username	Comments_Made
user1	2
user2	2
user3	2
user4	2
user5	2
user6	2
user7	2
user8	2
user9	2
user10	2

8.4 -- Unspent Money by Campaign

```
SELECT
   C.Name,
   C.Budget AS Total_Budget,
   SUM(A.Budget) AS Allocated_Ads,
   (C.Budget - SUM(A.Budget)) AS Unspent_Cash
FROM Campaign C
LEFT JOIN Advertisement A ON C.Campaign_ID = A.Campaign_ID
GROUP BY C.Name, C.Budget;
```



Name	Total_Budget	Allocated_Ads	Unspent_Cash
Launch Campaign	5000	2000	3000

8.5 Get Posts with Specific Topic

$\sigma_{\text{Topic\_Name='Databases}}(\text{Topic}) \longrightarrow \text{Selected\_Topic}$

$\pi_{\text{Topic\_ID}}(\text{Selected\_Topic}) \longrightarrow \text{Topic\_ID\_Filtered}$

$\text{Topic\_Posts} * \text{Topic\_ID\_Filtered} \longrightarrow \text{TP\_Filtered}$

$\text{Posts} * \text{TP\_Filtered} \longrightarrow \text{Posts\_With\_Topic}$

8.6 Find Users Followed by a Specific User

$\sigma_{\text{Username='user2'}}(\text{Users}) \longrightarrow \text{Filtered\_User}$

$\pi_{\text{User\_id}}(\text{Filtered\_User}) \longrightarrow \text{User\_ID\_Filtered}$

$\text{Users} \bowtie_{\text{User\_id = Following\_id}} \text{Follow} \longrightarrow \text{Users\_Follow\_Join}$

$\sigma_{\text{Follower\_id = User\_ID\_Filtered.User\_id}}(\text{Users\_Follow\_Join}) \longrightarrow \text{Followed\_Users}$

8.7 Count Reactions and Comments on Each Post

$\text{Posts} * \text{Comments} \longrightarrow \text{Posts\_Comments\_Join}$

Posts_Comments_Join ⟕ Post_Reaction

Posts_Comments_Join.post_id = Post_Reaction.Post_id

→ Posts_Comments_Reactions

γ_post_id ; (Posts_Comments_Reactions)

COUNT(Comment_id) → Total_Comments

,COUNT(Reaction_id) → Total_Reactions

8.8 Find Posts by Specific User

σ (Users) → Filtered_User

Username=′user8′

π Filtered_User → User_ID_Filtered

User_id

Posts * Filtered_User → Result Post

8.9 Posts with the Highest Average Sentiment Score

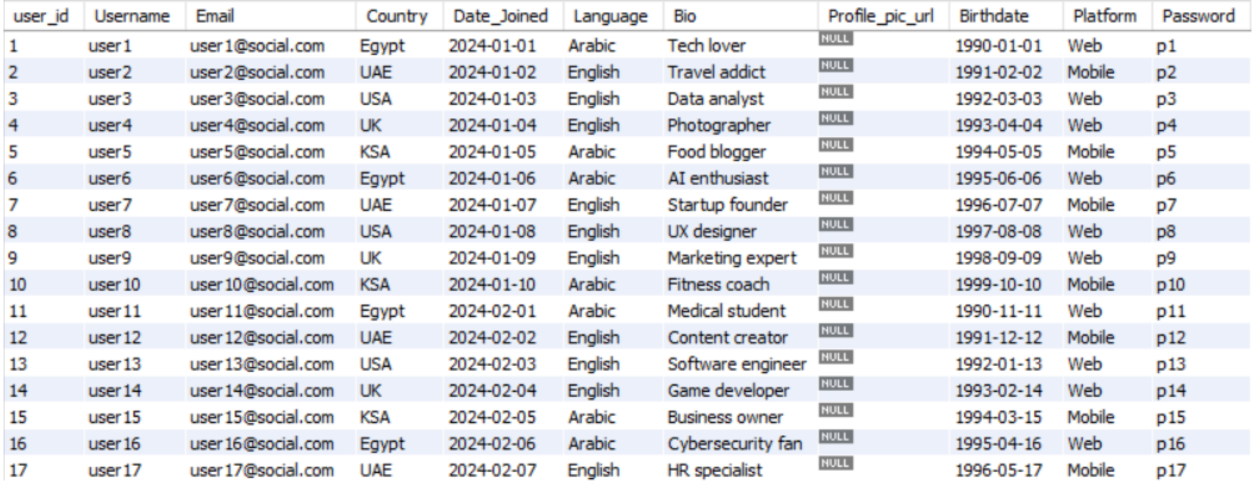
Posts * Sentiments → Posts_Sentiments_Join

γ_post_id; (Posts_Sentiments_Join)

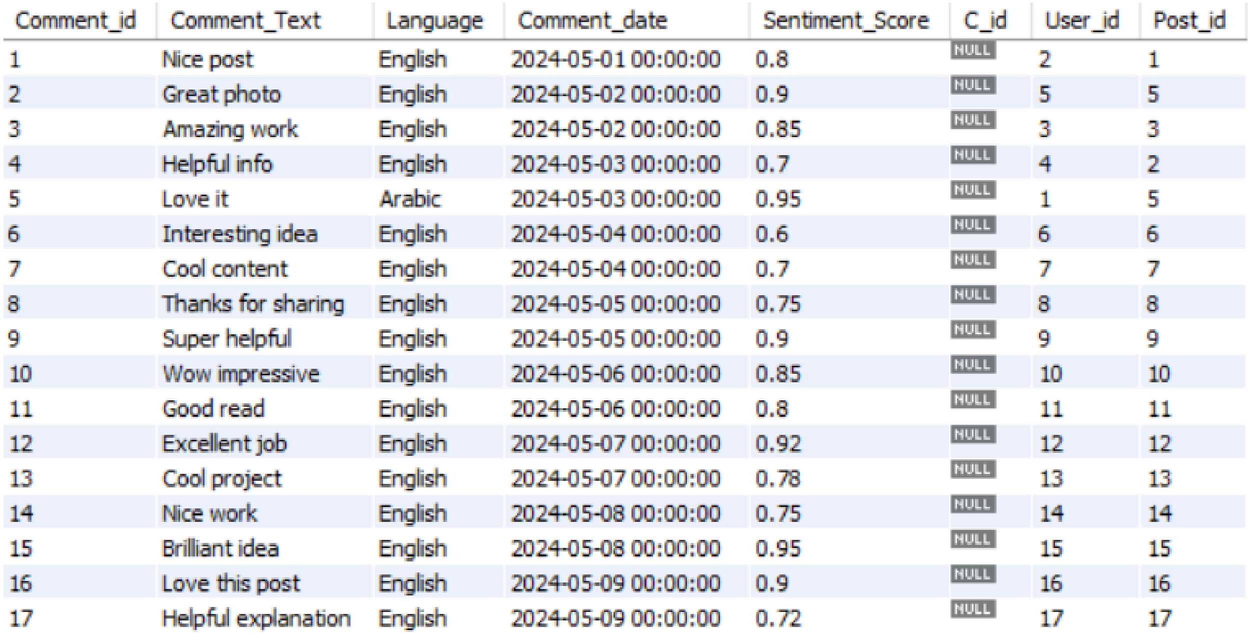
AVG(Sentiment_Score)→Avg_Sentiment

→ Posts_Avg_Sentiment

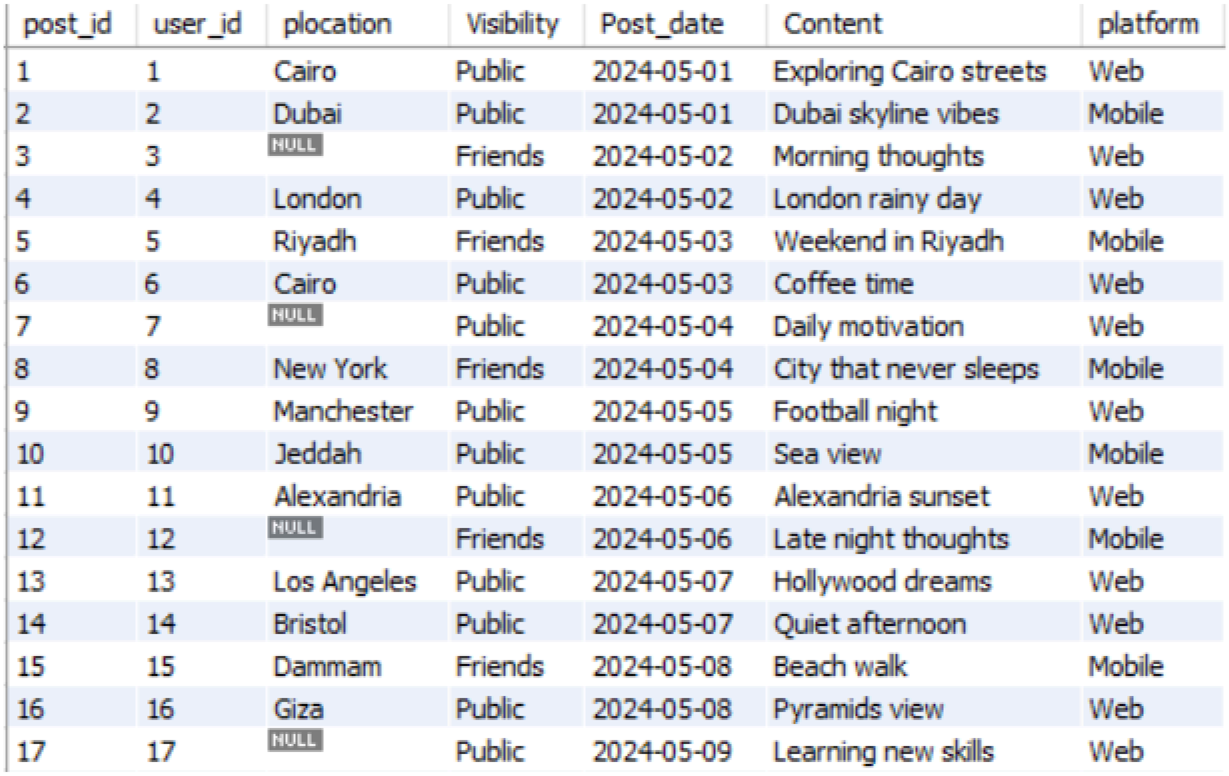
9. Data dictionary



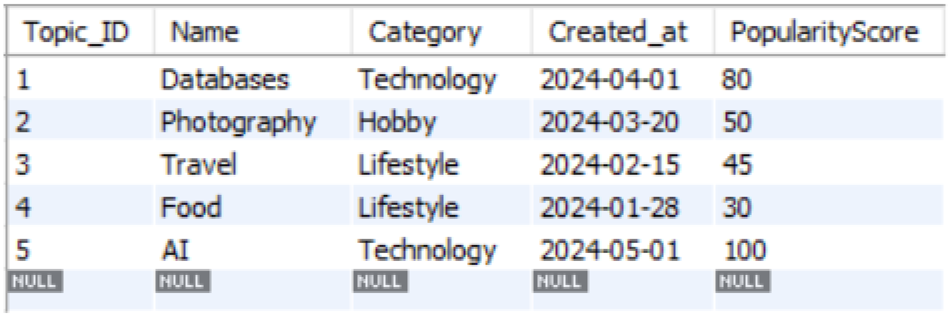
user_id	Username	Email	Country	Date_Joined	Language	Bio	Profile_pic_url	Birthdate	Platform	Password
1	user1	user1@social.com	Egypt	2024-01-01	Arabic	Tech lover	NULL	1990-01-01	Web	p1
2	user2	user2@social.com	UAE	2024-01-02	English	Travel addict	NULL	1991-02-02	Mobile	p2
3	user3	user3@social.com	USA	2024-01-03	English	Data analyst	NULL	1992-03-03	Web	p3
4	user4	user4@social.com	UK	2024-01-04	English	Photographer	NULL	1993-04-04	Web	p4
5	user5	user5@social.com	KSA	2024-01-05	Arabic	Food blogger	NULL	1994-05-05	Mobile	p5
6	user6	user6@social.com	Egypt	2024-01-06	Arabic	AI enthusiast	NULL	1995-06-06	Web	p6
7	user7	user7@social.com	UAE	2024-01-07	English	Startup founder	NULL	1996-07-07	Mobile	p7
8	user8	user8@social.com	USA	2024-01-08	English	UX designer	NULL	1997-08-08	Web	p8
9	user9	user9@social.com	UK	2024-01-09	English	Marketing expert	NULL	1998-09-09	Web	p9
10	user10	user10@social.com	KSA	2024-01-10	Arabic	Fitness coach	NULL	1999-10-10	Mobile	p10
11	user11	user11@social.com	Egypt	2024-02-01	Arabic	Medical student	NULL	1990-11-11	Web	p11
12	user12	user12@social.com	UAE	2024-02-02	English	Content creator	NULL	1991-12-12	Mobile	p12
13	user13	user13@social.com	USA	2024-02-03	English	Software engineer	NULL	1992-01-13	Web	p13
14	user14	user14@social.com	UK	2024-02-04	English	Game developer	NULL	1993-02-14	Web	p14
15	user15	user15@social.com	KSA	2024-02-05	Arabic	Business owner	NULL	1994-03-15	Mobile	p15
16	user16	user16@social.com	Egypt	2024-02-06	Arabic	Cybersecurity fan	NULL	1995-04-16	Web	p16
17	user17	user17@social.com	UAE	2024-02-07	English	HR specialist	NULL	1996-05-17	Mobile	p17



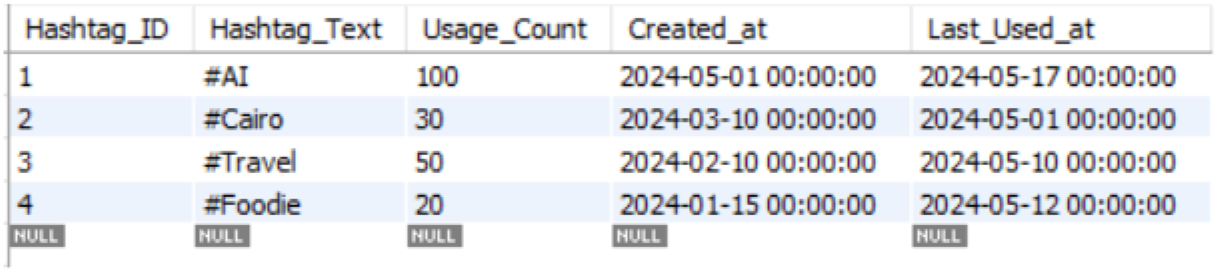
Comment_id	Comment_Text	Language	Comment_date	Sentiment_Score	C_id	User_id	Post_id
1	Nice post	English	2024-05-01 00:00:00	0.8	NULL	2	1
2	Great photo	English	2024-05-02 00:00:00	0.9	NULL	5	5
3	Amazing work	English	2024-05-02 00:00:00	0.85	NULL	3	3
4	Helpful info	English	2024-05-03 00:00:00	0.7	NULL	4	2
5	Love it	Arabic	2024-05-03 00:00:00	0.95	NULL	1	5
6	Interesting idea	English	2024-05-04 00:00:00	0.6	NULL	6	6
7	Cool content	English	2024-05-04 00:00:00	0.7	NULL	7	7
8	Thanks for sharing	English	2024-05-05 00:00:00	0.75	NULL	8	8
9	Super helpful	English	2024-05-05 00:00:00	0.9	NULL	9	9
10	Wow impressive	English	2024-05-06 00:00:00	0.85	NULL	10	10
11	Good read	English	2024-05-06 00:00:00	0.8	NULL	11	11
12	Excellent job	English	2024-05-07 00:00:00	0.92	NULL	12	12
13	Cool project	English	2024-05-07 00:00:00	0.78	NULL	13	13
14	Nice work	English	2024-05-08 00:00:00	0.75	NULL	14	14
15	Brilliant idea	English	2024-05-08 00:00:00	0.95	NULL	15	15
16	Love this post	English	2024-05-09 00:00:00	0.9	NULL	16	16
17	Helpful explanation	English	2024-05-09 00:00:00	0.72	NULL	17	17



post_id	user_id	plocation	Visibility	Post_date	Content	platform
1	1	Cairo	Public	2024-05-01	Exploring Cairo streets	Web
2	2	Dubai	Public	2024-05-01	Dubai skyline vibes	Mobile
3	3	NULL	Friends	2024-05-02	Morning thoughts	Web
4	4	London	Public	2024-05-02	London rainy day	Web
5	5	Riyadh	Friends	2024-05-03	Weekend in Riyadh	Mobile
6	6	Cairo	Public	2024-05-03	Coffee time	Web
7	7	NULL	Public	2024-05-04	Daily motivation	Web
8	8	New York	Friends	2024-05-04	City that never sleeps	Mobile
9	9	Manchester	Public	2024-05-05	Football night	Web
10	10	Jeddah	Public	2024-05-05	Sea view	Mobile
11	11	Alexandria	Public	2024-05-06	Alexandria sunset	Web
12	12	NULL	Friends	2024-05-06	Late night thoughts	Mobile
13	13	Los Angeles	Public	2024-05-07	Hollywood dreams	Web
14	14	Bristol	Public	2024-05-07	Quiet afternoon	Web
15	15	Dammam	Friends	2024-05-08	Beach walk	Mobile
16	16	Giza	Public	2024-05-08	Pyramids view	Web
17	17	NULL	Public	2024-05-09	Learning new skills	Web

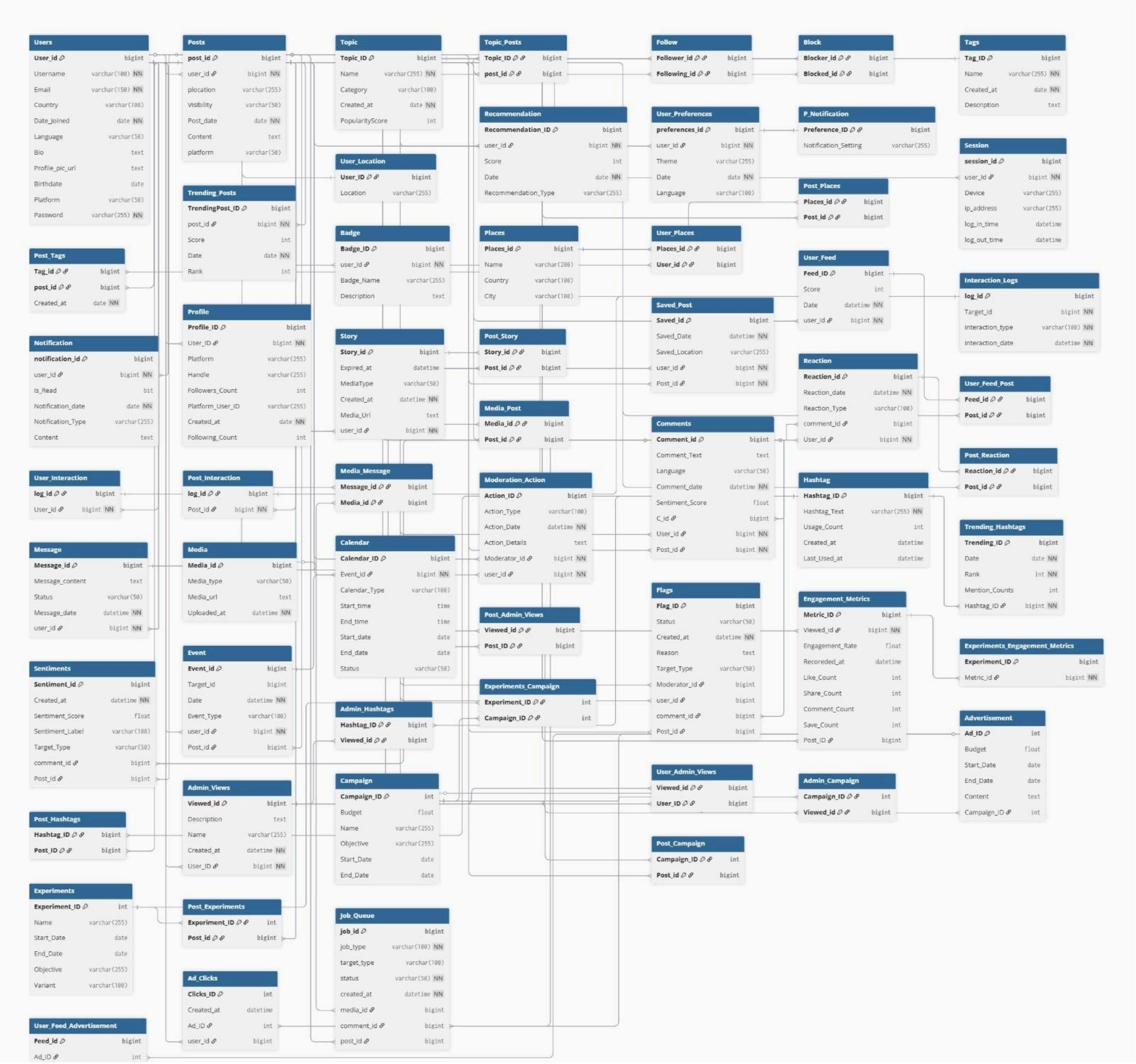


Topic_ID	Name	Category	Created_at	PopularityScore
1	Databases	Technology	2024-04-01	80
2	Photography	Hobby	2024-03-20	50
3	Travel	Lifestyle	2024-02-15	45
4	Food	Lifestyle	2024-01-28	30
5	AI	Technology	2024-05-01	100
NULL	NULL	NULL	NULL	NULL



Hashtag_ID	Hashtag_Text	Usage_Count	Created_at	Last_Used_at
1	#AI	100	2024-05-01 00:00:00	2024-05-17 00:00:00
2	#Cairo	30	2024-03-10 00:00:00	2024-05-01 00:00:00
3	#Travel	50	2024-02-10 00:00:00	2024-05-10 00:00:00
4	#Foodie	20	2024-01-15 00:00:00	2024-05-12 00:00:00
NULL	NULL	NULL	NULL	NULL

10. DB schema


Users
User_id bigint
Username varchar(100) NN
Email varchar(150) NN
Country varchar(100)
Date_Joined date NN
Language varchar(50)
Bio text
Profile_pic_url text
Birthdate date
Platform varchar(50)
Password varchar(255) NN
Post_Tags
Tag_id bigint
post_id bigint
Created_at date NN
Notification
notification_id bigint
user_id bigint NN
Is_Read bit
Notification_date date NN
Notification_Type varchar(255)
Content text
User_Interaction
log_id bigint
User_id bigint NN
Message
Message_id bigint
Message_content text
Status varchar(50)
Message_date datetime NN
user_id bigint NN
Sentiments
Sentiment_id bigint
Created_at datetime NN
Sentiment_Score float
Sentiment_Label varchar(100)
Target_Type varchar(50)
comment_id bigint
Post_id bigint
Post_Hashtags
Hashtag_ID bigint
Post_ID bigint
Experiments
Experiment_ID int
Name varchar(255)
Start_Date date
End_Date date
Objective varchar(255)
Variant varchar(100)
User_Feed_Advertisement
Feed_id bigint
Ad_ID int
Posts
post_id bigint
user_id bigint NN
plocation varchar(255)
Visibility varchar(50)
Post_date date NN
Content text
platform varchar(50)
Trending_Posts
TrendingPost_ID bigint
post_id bigint NN
Score int
Date date NN
Rank int
Profile
Profile_ID bigint
User_ID bigint NN
Platform varchar(255)
Handle varchar(255)
Followers_Count int
Platform_User_ID varchar(255)
Created_at date NN
Following_Count int
Post_Interaction
log_id bigint
Post_id bigint NN
Media
Media_id bigint
Media_type varchar(50)
Media_url text
Uploaded_at datetime NN
Event
Event_id bigint
Target_id bigint
Date datetime NN
Event_Type varchar(100)
user_id bigint NN
Post_id bigint
Admin_Views
Viewed_id bigint
Description text
Name varchar(255)
Created_at datetime NN
User_ID bigint NN
Post_Experiments
Experiment_ID int
Post_id bigint
Ad_Clicks
Clicks_ID int
Created_at datetime
Ad_ID int
user_id bigint
Topic
Topic_ID bigint
Name varchar(255) NN
Category varchar(100)
Created_at date NN
PopularityScore int
User_Location
User_ID bigint
Location varchar(255)
Badge
Badge_ID bigint
user_id bigint NN
Badge_Name varchar(255)
Description text
Story
Story_id bigint
Expired_at datetime
MediaType varchar(50)
Created_at datetime NN
Media_Url text
user_id bigint NN
Media_Message
Message_id bigint
Media_id bigint
Calendar
Calendar_ID bigint
Event_id bigint NN
Calendar_Type varchar(100)
Start_time time
End_time time
Start_date date
End_date date
Status varchar(50)
Admin_Hashtags
Hashtag_ID bigint
Viewed_id bigint
Campaign
Campaign_ID int
Budget float
Name varchar(255)
Objective varchar(255)
Start_Date date
End_Date date
Job_Queue
job_id bigint
job_type varchar(100) NN
target_type varchar(100)
status varchar(50) NN
created_at datetime NN
media_id bigint
comment_id bigint
post_id bigint
Topic_Posts
Topic_ID bigint
post_id bigint
Recommendation
Recommendation_ID bigint
user_id bigint NN
Score int
Date date NN
Recommendation_Type varchar(255)
Places
Places_id bigint
Name varchar(200)
Country varchar(100)
City varchar(100)
Post_Story
Story_id bigint
Post_id bigint
Media_Post
Media_id bigint
Post_id bigint
Moderation_Action
Action_ID bigint
Action_Type varchar(100)
Action_Date datetime NN
Action_Details text
Moderator_id bigint NN
user_id bigint NN
Post_Admin_Views
Viewed_id bigint
Post_ID bigint
Experiments_Campaign
Experiment_ID int
Campaign_ID int
Follow
Follower_id bigint
Following_id bigint
User_Preferences
preferences_id bigint
user_id bigint NN
Theme varchar(255)
Date date NN
Language varchar(100)
User_Places
Places_id bigint
User_id bigint
Saved_Post
Saved_id bigint
Saved_Date datetime NN
Saved_Location varchar(255)
user_id bigint NN
Post_id bigint NN
Comments
Comment_id bigint
Comment_Text text
Language varchar(50)
Comment_date datetime NN
Sentiment_Score float
C_id bigint
User_id bigint NN
Post_id bigint NN
Flags
Flag_ID bigint
Status varchar(50)
Created_at datetime NN
Reason text
Target_Type varchar(50)
Moderator_id bigint
user_id bigint
comment_id bigint
Post_id bigint
User_Admin_Views
Viewed_id bigint
User_ID bigint
Post_Campaign
Campaign_ID int
Post_id bigint
Block
Blocker_id bigint
Blocked_id bigint
P_Notification
Preference_ID bigint
Notification_Setting varchar(255)
Post_Places
Places_id bigint
Post_id bigint
User_Feed
Feed_ID bigint
Score int
Date datetime NN
user_id bigint NN
Reaction
Reaction_id bigint
Reaction_date datetime NN
Reaction_Type varchar(100)
comment_id bigint
User_id bigint NN
Hashtag
Hashtag_ID bigint
Hashtag_Text varchar(255) NN
Usage_Count int
Created_at datetime
Last_Used_at datetime
Engagement_Metrics
Metric_ID bigint
Viewed_id bigint NN
Engagement_Rate float
Recoreded_at datetime
Like_Count int
Share_Count int
Comment_Count int
Save_Count int
Post_ID bigint
Admin_Campaign
Campaign_ID int
Viewed_id bigint
Tags
Tag_ID bigint
Name varchar(255) NN
Created_at date NN
Description text
Session
session_id bigint
user_id bigint NN
Device varchar(255)
ip_address varchar(255)
log_in_time datetime
log_out_time datetime
Interaction_Logs
log_id bigint
Target_id bigint NN
interaction_type varchar(100) NN
interaction_date datetime NN
User_Feed_Post
Feed_id bigint
Post_id bigint
Post_Reaction
Reaction_id bigint
Post_id bigint
Trending_Hashtags
Trending_ID bigint
Date date NN
Rank int NN
Mention_Counts int
Hashtag_ID bigint NN
Experiments_Engagement_Metrics
Experiment_ID bigint
Metric_id bigint NN
Advertisement
Ad_ID int
Budget float
Start_Date date
End_Date date
Content text
Campaign_ID int

11. Backend Architecture

11.1 Overview

The application provides users with an interactive web-based platform to access the system's features.

In this project, several frameworks and tools were used. The frontend was developed using HTML, CSS, and JavaScript, while the backend was implemented using FastAPI, utilizing Swagger to create and test API endpoints. These endpoints enable the GUI to communicate with the database, which was created using PostgreSQL.

The system uses RESTful APIs to allow communication between the GUI and the backend. The APIs are organized under versioned routes, which ensures scalability, maintainability, and ease of future enhancements.

11.2 Backend Structure

The backend follows a modular architecture, where each module is responsible for a specific functionality. The backend includes:

- API endpoints accessed by the GUI
- Database models representing PostgreSQL tables
- Request and response validation using schemas
- Authentication, security, and sentiment analysis logic
- Database connection and configuration management

12. GUI Overview

GUI Pages and Corresponding Backend APIs

Authentication Interface

- Login Page
- Registration Page

User Profile Interface

- View Profile
- Edit Profile
- User Preferences

Posts and Feed Interface

- Create Post
- View Feed
- View Post Details

Comments and Reactions

- Comment Section
- Like / Reaction Buttons

Follow System

- Follow / Unfollow Users
- Followers List

Notifications Interface

- Notifications Panel

Messaging Interface

- Inbox

Stories Interface

- View Stories
- Add Story

Search Interface

- Search Bar
- Search Results Page

Bookmarks Interface

- Saved Posts Page

Media Upload Interface

- Upload Images / Videos

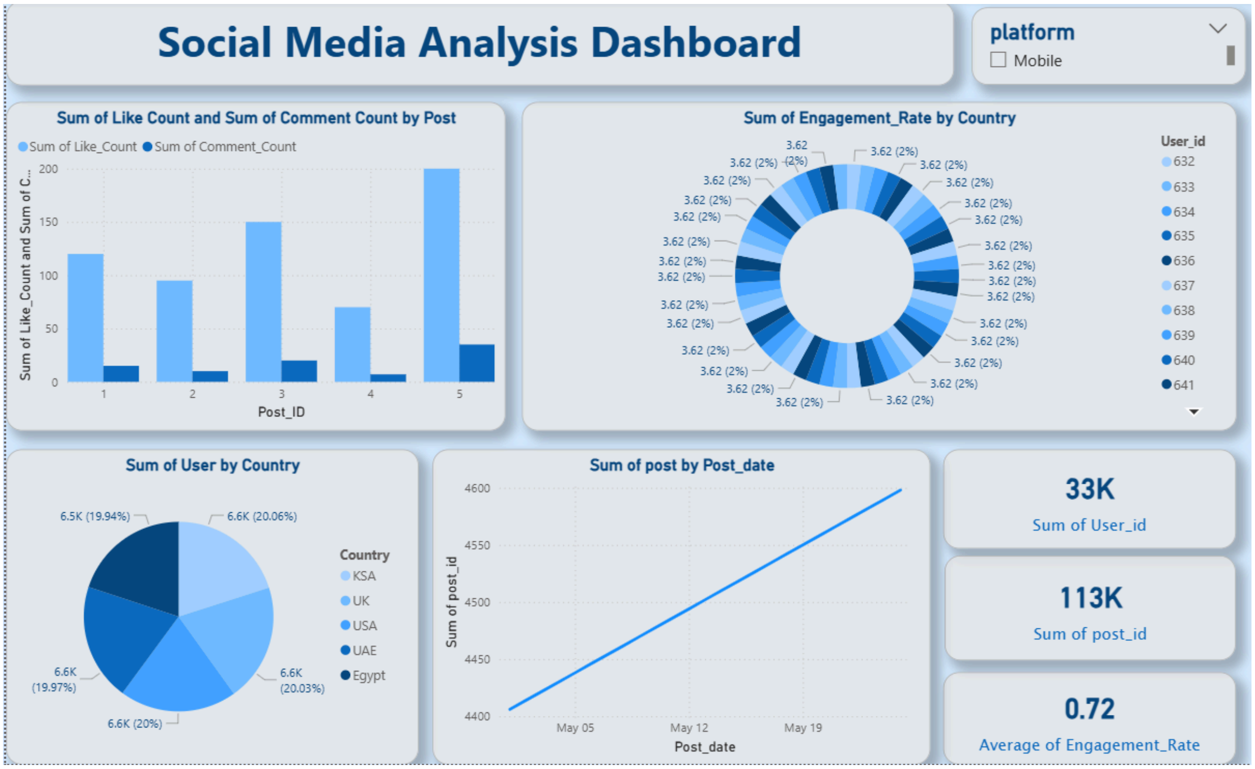
Error Handling and Feedback

Backend errors are returned in JSON format and handled by JavaScript on the client side.

Security in GUI

- JWT-based authentication
- Protected routes using FastAPI dependencies
- No direct database access from the frontend
- Secure environment variables

13. Dashboard Analysis



14. Conclusion & Future Enhancements

14.1 Conclusion

This project presents a comprehensive database design for a modern social media platform with integrated sentiment analysis. The system efficiently manages users, posts, comments, reactions, and interactions, while enabling real-time content analysis and moderation.

The ERD clearly models the relationships between all core and advanced entities, ensuring data consistency, scalability, and support for analytics-driven decisions. Overall, the design provides a solid foundation for building a safe, intelligent, and highly interactive social media platform.

14.2 Future Enhancements

- Integrate more advanced AI models for sentiment analysis and content recommendations.
- Extend analytics dashboards with real-time insights and predictive trends.
- Support additional social features such as groups, live content, and polls.
- Improve performance through distributed databases and caching mechanisms.
- Enhance moderation automation while maintaining human review capabilities

  .