Database Schema Documentation

PROJECT	DATABASE	GENERATED
Social Media Posting System	PostgreSQL 15	October 16, 2025 at 09:53 AM
TOTAL TABLES	EXTENSIONS	VERSION
25	uuid-ossp, ltree	1.0

Table of Contents

1. Schema Overview

25
Tables

32
Functions

18
Triggers

85+
Indexes

System Architecture

The database schema is organized into seven major subsystems that work together to provide a comprehensive social media platform with advanced features including user interactions, reputation tracking, geolocation services, and Reddit-style group communities.

Extensions Used

Extension	Purpose	Usage
`uuid-ossp`	UUID generation	Generating unique identifiers for resources
`ltree`	Hierarchical data	Efficient nested comment threading in groups

2. Core Tables

users

Core Table

Description: Main user accounts and authentication

Column	Type	Constraints	Description
id	SERIAL	PRIMARY KEY	Unique identifier
username	VARCHAR(50)	UNIQUE NOT NULL	Unique username
email	VARCHAR(255)	UNIQUE NOT NULL	Email address
password_hash	VARCHAR(255)	NOT NULL	Hashed password
first_name	VARCHAR(100)	NOT NULL	First name
last_name	VARCHAR(100)	NOT NULL	Last name
bio	TEXT		User biography
avatar_url	VARCHAR(500)		Profile picture URL
is_active	BOOLEAN	DEFAULT TRUE	Account status
email_verified	BOOLEAN	DEFAULT FALSE	Email verification status
location_latitude	DECIMAL(10,7)		Latitude coordinate
location_longitude	DECIMAL(10,7)		Longitude coordinate

location_city	`VARCHAR(100)`		City name
location_state	`VARCHAR(100)`		State/province
location_sharing	`VARCHAR(20)`	CHECK DEFAULT 'off'	Privacy: exact, city, off
created_at	`TIMESTAMP`	DEFAULT CURRENT_TIMESTAMP	Creation timestamp
updated_at	`TIMESTAMP`	DEFAULT CURRENT_TIMESTAMP	Last update timestamp

Indexes

- **idx_users_username**
  `(username)`
- **idx_users_email**
  `(email)`
- **idx_users_active**
  `(is_active)`
- **idx_users_location_coords**
  `(location_latitude, location_longitude)`

posts

Core Table

Description: User posts and content

Column	Type	Constraints	Description
id	SERIAL	PRIMARY KEY	Unique identifier
user_id	INTEGER	FK users(id) CASCADE	Post author
content	TEXT	NOT NULL	Post content/text
privacy_level	VARCHAR(20)	CHECK DEFAULT 'public'	public, friends, private
is_published	BOOLEAN	DEFAULT TRUE	Publication status
is_archived	BOOLEAN	DEFAULT FALSE	Archive status
views_count	INTEGER	DEFAULT 0	Number of views
created_at	TIMESTAMP	DEFAULT CURRENT_TIMESTAMP	Creation timestamp

Indexes

idx_posts_user_id
(user_id)

idx_posts_privacy_level
(privacy_level)

idx_posts_created_at
(created_at)

comments

Core Table

Description: Nested comments on posts

Column	Type	Constraints	Description
id	SERIAL	PRIMARY KEY	Unique identifier
user_id	INTEGER	FK users(id) CASCADE	Comment author
post_id	INTEGER	FK posts(id) CASCADE	Parent post
parent_id	INTEGER	FK comments(id) CASCADE	Parent comment (for nesting)
content	TEXT	NOT NULL	Comment text
depth	INTEGER	DEFAULT 0	Nesting level
created_at	TIMESTAMP	DEFAULT CURRENT_TIMESTAMP	Creation timestamp

media

Core Table

Description: Media attachments for posts and comments

Column	Type	Constraints	Description
id	SERIAL	PRIMARY KEY	Unique identifier
user_id	INTEGER	FK users(id) CASCADE	Uploader
post_id	INTEGER	FK posts(id) CASCADE	Associated post
comment_id	INTEGER	FK comments(id) CASCADE	Associated comment
file_url	VARCHAR(500)	NOT NULL	Public URL
mime_type	VARCHAR(100)	NOT NULL	File MIME type
media_type	VARCHAR(20)	CHECK	image, video, audio, document
width	INTEGER		Image/video width
height	INTEGER		Image/video height

reactions

Core Table

Description: Emoji reactions on posts and comments

Column	Type	Constraints	Description
id	`SERIAL`	PRIMARY KEY	Unique identifier
user_id	`INTEGER`	FK users(id) CASCADE	User who reacted
post_id / comment_id	`INTEGER`	FK	Target content
emoji_name	`VARCHAR(50)`	NOT NULL	Emoji shortcode
emoji_unicode	`VARCHAR(20)`	NOT NULL	Unicode emoji

Unique Constraints:

- One reaction per user per post per emoji
- One reaction per user per comment per emoji

3. Social Features

follows

Social

Description: User follow relationships with status tracking

Column	Type	Constraints	Description
id	SERIAL	PRIMARY KEY	Unique identifier
follower_id	INTEGER	FK users(id) CASCADE	User who follows
following_id	INTEGER	FK users(id) CASCADE	User being followed
status	VARCHAR(20)	CHECK DEFAULT 'active'	active, muted, blocked
notifications_enabled	BOOLEAN	DEFAULT TRUE	Notification preferences

Constraints:

- UNIQUE (follower_id, following_id)
- CHECK follower_id != following_id (no self-follow)

shares

Social

Description: Post shares and reposts

Column	Type	Constraints	Description
id	SERIAL	PRIMARY KEY	Unique identifier
user_id	INTEGER	FK users(id) CASCADE	User who shared
post_id	INTEGER	FK posts(id) CASCADE	Shared post
share_type	VARCHAR(20)	CHECK DEFAULT 'repost'	repost, quote, external
share_comment	TEXT		Optional comment on share

user_stats

Denormalized

Description: Aggregated user statistics for performance

Column	Type	Description
user_id	INTEGER	PRIMARY KEY
follower_count	INTEGER	Number of followers
following_count	INTEGER	Number following
post_count	INTEGER	Total posts
total_reactions_received	INTEGER	Reactions on user's content
total_shares_received	INTEGER	Shares of user's posts
engagement_score	DECIMAL(10,2)	Calculated engagement metric

Note: Auto-updated by triggers on follows and shares tables

timeline_cache

Cache

Description: Pre-computed timeline with scoring algorithm

Column	Type	Description
user_id	INTEGER	Timeline owner
post_id	INTEGER	Post in timeline
score	DECIMAL(10,2)	Relevance score (0-100)
reason	VARCHAR(50)	following, popular, shared, suggested
expires_at	TIMESTAMP	Cache expiration (7 days)

Scoring Algorithm (0-100 points)

- **Relationship (40pts):** Following > mutual > suggested
- **Recency (25pts):** Newer posts score higher
- **Engagement (20pts):** Reactions + comments + shares
- **Content Type (10pts):** Media posts bonus
- **User Activity (5pts):** Active users bonus

7. Group System (Reddit-style Communities)

groups

Communities

Description: Reddit-style community groups with moderation

Column	Type	Constraints	Description
id	SERIAL	PRIMARY KEY	Unique identifier
name	VARCHAR(100)	UNIQUE NOT NULL	Internal name
slug	VARCHAR(100)	UNIQUE NOT NULL	URL-friendly identifier
display_name	VARCHAR(255)	NOT NULL	Public display name
description	TEXT		Group description
avatar_url	VARCHAR(500)		Group icon (400x400px)
banner_url	VARCHAR(500)		Group banner (1200x400px)
visibility	VARCHAR(20)	CHECK DEFAULT 'public'	public, private, invite_only
require_approval	BOOLEAN	DEFAULT FALSE	Membership approval required
post_approval_required	BOOLEAN	DEFAULT FALSE	Post moderation required
location_restricted	BOOLEAN	DEFAULT FALSE	Geographic restrictions enabled
location_type	VARCHAR(20)	CHECK	radius, country, state, city, polygon

member_count	INTEGER	DEFAULT 0	Total active members
creator_id	INTEGER	FK users(id) RESTRICT	Group creator

group_memberships

Communities

Description: User memberships in groups with roles

Column	Type	Constraints	Description
group_id	INTEGER	FK groups(id) CASCADE	Group reference
user_id	INTEGER	FK users(id) CASCADE	Member reference
role	VARCHAR(20)	CHECK DEFAULT 'member'	admin, moderator, member
status	VARCHAR(20)	CHECK DEFAULT 'active'	active, banned, pending, invited
banned_by	INTEGER	FK users(id)	Moderator who banned
banned_reason	TEXT		Ban reason

group_posts

Communities

Description: Posts within groups with voting system

Column	Type	Description
id	SERIAL	PRIMARY KEY
group_id	INTEGER	FK groups(id) CASCADE
user_id	INTEGER	FK users(id) CASCADE
title	VARCHAR(300)	Post title
content	TEXT	Post content
post_type	VARCHAR(20)	text, link, media, poll
status	VARCHAR(20)	draft, pending, published, removed
upvotes	INTEGER	Upvote count (auto-updated)
downvotes	INTEGER	Downvote count (auto-updated)
score	INTEGER	upvotes - downvotes (auto-updated)
is_pinned	BOOLEAN	Pinned to top of group
is_locked	BOOLEAN	Comments disabled

group_comments

Communities

Description: Nested comments using ltree for efficient threading

Column	Type	Description
id	SERIAL	PRIMARY KEY
post_id	INTEGER	FK group_posts(id) CASCADE
parent_id	INTEGER	FK group_comments(id) CASCADE
content	TEXT	Comment text
path	LTREE	Hierarchical path (e.g., "1.3.7")
depth	INTEGER	Nesting depth (0 = top-level)
upvotes	INTEGER	Upvote count
downvotes	INTEGER	Downvote count
score	INTEGER	upvotes - downvotes

ltree Extension: Enables efficient queries like "get all children of comment X" or "get full comment tree"

group_votes

Communities

Description: Upvote/downvote system for posts and comments

Column	Type	Description
user_id	`INTEGER`	User who voted
post_id	`INTEGER`	Voted post (null if comment)
comment_id	`INTEGER`	Voted comment (null if post)
vote_type	`VARCHAR(10)`	upvote or downvote

Triggers: Auto-updates upvotes, downvotes, and score columns on group_posts and group_comments

8. Stored Functions & Procedures

update_updated_at_column()

Purpose: Universal trigger function to update the updated_at timestamp
Returns: TRIGGER
Used by: 15+ tables

```
CREATE OR REPLACE FUNCTION update_updated_at_column() RETURNS TRIGGER AS $$ BEGIN NEW.updated_at = CURRENT_TIMESTAMP;
RETURN NEW; END; $$ language 'plpgsql';
```

update_follow_counts()

Purpose: Automatically update follower/following counts in user_stats
Returns: TRIGGER
Trigger: AFTER INSERT OR DELETE ON follows

```
-- Increments following_count for follower -- Increments follower_count for following -- Uses UPSERT (ON CONFLICT) for
atomic updates -- Ensures counts never go below 0 with GREATEST(0, count - 1)
```

calculate_distance_miles(lat1, lon1, lat2, lon2)

Purpose: Calculate distance between two coordinates using Haversine formula
Returns: DECIMAL(10,2) - Distance in miles

Parameters: 4 DECIMAL values (latitude and longitude pairs)

```
-- Haversine formula for great-circle distance -- Accurate up to a few hundred miles -- Earth radius: 3959 miles RETURN (R
* c)::DECIMAL(10, 2);
```

find_nearby_users(user_id, lat, lon, radius_miles, limit, offset)

Purpose: Find users within specified radius sorted by distance
Returns: TABLE with user details and distances
Performance: Uses bounding box filter before exact distance calculation

```
-- Step 1: Rough bounding box filter (fast) -- Step 2: Exact distance calculation (Haversine) -- Step 3: Filter by
location_sharing != 'off' -- Returns: user details + distance_miles
```

update_group_member_count()

Purpose: Auto-update member_count on groups table
Returns: TRIGGER
Trigger: AFTER INSERT OR UPDATE OR DELETE ON group_memberships

```
-- Tracks status changes: active ↔ banned/pending -- Only counts 'active' members -- Handles INSERT, UPDATE, DELETE
operations
```

update_group_post_votes()

Purpose: Recalculate upvotes, downvotes, and score for group posts

Returns: TRIGGER
Trigger: AFTER INSERT OR UPDATE OR DELETE ON group_votes

```
-- Counts upvote/downvote using FILTER WHERE -- Calculates score = upvotes - downvotes -- Updates group_posts table atomically
```

set_group_comment_path()

Purpose: Set ltree path for nested comments
Returns: TRIGGER
Trigger: BEFORE INSERT ON group_comments

```
-- Top-level: path = id (e.g., "42") -- Nested: path = parent_path.id (e.g., "42.137.245") -- Also calculates depth for nesting level
```

calculate_reputation_score(user_id)

Purpose: Calculate user reputation score (0-1000) and level
Returns: INTEGER
Algorithm:

- Base score from average rating (max 500)
- Volume bonus from total ratings (max 100)
- Quality content bonus (max 250)
- Helpful bonus (max 100)
- Verified bonus (max 50)
- Penalties for reports (-10 per report)

Levels: newcomer (0-99), member (100-299), contributor (300-499), veteran (500-699), expert (700-849), legend (850-1000)

update_user_reputation()

Purpose: Update user_reputation table when ratings change
Returns: TRIGGER
Trigger: AFTER INSERT OR UPDATE OR DELETE ON user_ratings

```
-- Recalculates all aggregates: -- - Average rating across all categories -- - Rating distribution (1-5 stars) -- - Positive/neutral/negative counts -- - Category-specific averages
```

is_group_member(user_id, group_id)

Purpose: Check if user is an active member of group
Returns: BOOLEAN
Usage: Permission checks in application code

is_group_moderator(user_id, group_id)

Purpose: Check if user is moderator or admin of group
Returns: BOOLEAN
Usage: Moderation permission checks

is_group_admin(user_id, group_id)

Purpose: Check if user is admin of group
Returns: BOOLEAN

Usage: Admin-only operations (settings, role changes)

9. Triggers & Automation

Timestamp Triggers

Trigger Name	Table	Timing	Function
update_users_updated_at	users	BEFORE UPDATE	update_updated_at_column()
update_posts_updated_at	posts	BEFORE UPDATE	update_updated_at_column()
update_comments_updated_at	comments	BEFORE UPDATE	update_updated_at_column()
update_groups_updated_at	groups	BEFORE UPDATE	update_updated_at_column()

Denormalization Triggers

Trigger Name	Table	Timing	Function	Purpose
trigger_update_follow_counts	follows	AFTER INSERT/DELETE	update_follow_counts()	Update follower/following counts
trigger_update_share_counts	shares	AFTER INSERT/DELETE	update_share_counts()	Update share received counts
trigger_update_group_member_count	group_memberships	AFTER INSERT/UPDATE/DELETE	update_group_member_count()	Update group member counts
trigger_update_group_post_count	group_posts	AFTER INSERT/UPDATE/DELETE	update_group_post_count()	Update group post counts

Voting System Triggers

Trigger Name	Table	Function	Updates
trigger_update_group_post_votes	group_votes	update_group_post_votes()	group_posts.upvotes, downvotes, score
trigger_update_group_comment_votes	group_votes	update_group_comment_votes()	group_comments.upvotes, downvotes, score
trigger_update_group_post_comment_count	group_comments	update_group_post_comment_count()	group_posts.comment_count

Comment Path Triggers

Trigger Name	Table	Timing	Purpose
trigger_set_group_comment_path	group_comments	BEFORE INSERT	Set ltree path and depth for nested comments

Reputation Triggers

Trigger Name	Table	Timing	Function
trigger_update_user_reputation	user_ratings	AFTER INSERT/UPDATE/DELETE	update_user_reputation()
trigger_update_helpful_count	helpful_marks	AFTER INSERT	update_helpful_count()

Comment Metrics Triggers

Trigger Name	Table	Function	Purpose
comment_interaction_metrics_trigger	comment_interactions	update_comment_metrics()	Update aggregated metrics on new interactions

Data Retention Triggers

Trigger Name	Table	Function	Purpose
trigger_limit_location_history	location_history	limit_location_history()	Keep only last 100 location entries per user

10. Indexes & Performance Optimization

Primary Tables Indexes

users table

idx_users_username
(username) - Login lookups

idx_users_email
(email) - Email lookups

idx_users_location_coords
(location_latitude, location_longitude) - Geo queries

idx_users_location_sharing
(location_sharing) - Privacy filtering

posts table

idx_posts_user_id
(user_id) - User's posts

idx_posts_created_at
(created_at) - Timeline sorting

idx_posts_published
(is_published) - Filter drafts

comments table

idx_comments_post_id
(post_id) - Get post comments

idx_comments_parent_id
(parent_id) - Get child comments

idx_comments_user_id
(user_id) - User's comments

Social Features Indexes

follows table

idx_follows_follower_id

idx_follows_following_id

(follower_id) - Who user follows

(following_id) - User's followers

idx_follows_composite
(follower_id, following_id, status) - Check follow status

timeline_cache table

idx_timeline_user_score
(user_id, score DESC) - Get user timeline

idx_timeline_expires_at
(expires_at) - Cleanup old entries

Group System Indexes

groups table

idx_groups_slug
(slug) - URL lookups

idx_groups_visibility
(visibility) - Filter by visibility

idx_groups_name_search
GIN (to_tsvector) - Full-text search

idx_groups_location_restricted
(location_restricted) WHERE TRUE - Geo groups

group_posts table

idx_group_posts_group
(group_id) - Group's posts

idx_group_posts_created
(group_id, created_at DESC) - Sort by new

idx_group_posts_score
(group_id, score DESC) - Sort by hot

idx_group_posts_pinned
(group_id, is_pinned, created_at DESC) - Pinned posts first

group_comments table

idx_group_comments_path
GIST (path) - ltree queries

idx_group_comments_post
(post_id) - Post's comments

idx_group_comments_score
(post_id, score DESC) - Sort by best

Reputation System Indexes

user_ratings table

idx_user_ratings_rated_user
`(rated_user_id) - User's ratings`

idx_user_ratings_context
`(context_type, context_id) - Rating context`

user_reputation table

idx_user_reputation_score
`(reputation_score DESC) - Leaderboards`

idx_user_reputation_level
`(reputation_level) - Filter by level`

Performance Optimization Notes

- **Composite Indexes:** Used for multi-column queries (e.g., filtering + sorting)
- **Partial Indexes:** indexes with WHERE clauses for specific subsets (e.g., active users only)
- **GIN Indexes:** For full-text search and ltree path queries
- **GIST Indexes:** For geometric and ltree data types
- **Foreign Key Indexes:** All FK columns are indexed for join performance
- **Timestamp Indexes:** DESC order for timeline queries (newest first)

Summary Statistics

25

Total Tables

32

Functions

18

Triggers

85+

Indexes

Table Categories

Category	Tables	Count
Core System	users, posts, comments, media, reactions	5
Social Features	follows, shares, user_stats, timeline_cache	4
Interaction Tracking	comment_interactions, comment_metrics	2
Reputation System	user_ratings, user_reputation, rating_reports, helpful_marks	4
Geolocation	location_history, nearby_search_cache (+ columns in users)	2
Group System	groups, group_memberships, group_posts, group_comments, group_votes, group_post_media, group_comment_media, group_invitations, group_activity_log	9

Key Features

- **Automatic Denormalization:** Triggers maintain counts and scores

- **Full-Text Search:** GIN indexes on text fields for fast searching
- **Nested Comments:** Support for both traditional parent_id and ltree paths
- **Geolocation:** Haversine distance calculations without PostGIS
- **Privacy Controls:** Granular location sharing settings
- **Reddit-style Voting:** Upvote/downvote system with auto-calculated scores
- **Moderation System:** Activity logs and role-based permissions
- **Reputation Tracking:** Multi-factor reputation scoring (0-1000)
- **Timeline Algorithm:** Scored cache system for personalized feeds

Database Schema Documentation

Social Media Posting System • PostgreSQL 15 • Version 1.0

Generated: October 16, 2025 at 09:53 AM