

Developer Summary — Updated SQL Patch-ChatBot

1. Users & Authentication

- Table: `users`
- Purpose: Stores user accounts with secure password hashes (`password`), roles, and status.
- Features:
 - `role` column allows role-based access (Admin/User).
 - `status` indicates active/inactive users.
 - `deleted_at` supports soft delete for GDPR compliance.
 - `created_at` and `updated_at` track record creation and modification.
 - Unique constraint on `email` ensures no duplicate accounts.
- Seed Data: Includes Admin, normal user, and demo user accounts.

2. Password Reset Management

- Table: `password_resets`
- Purpose: Handles password reset requests securely.
- Features:
 - `token` stores unique reset token.
 - `expires_at` defines validity of reset token.
 - Foreign key links email to `users` table for integrity.
 - Indexes on `email`, `token`, and `expires_at` optimize search.

3. Remember Me / Persistent Login

- Table: `remember_tokens`
- Purpose: Implements "remember me" functionality for persistent sessions.
- Features:
 - `token` and `expires_at` track session persistence.
 - Soft delete via `deleted_at`.
 - Indexed by `user_id` and `expires_at` for fast validation.
 - Foreign key ensures token belongs to a valid user.

4. Chat Sessions

- Table: `chat_sessions`
- Purpose: Tracks chat sessions per user or anonymous visitor.
- Features:
 - `session_uid` ensures unique session identification.
 - `status` indicates active/completed sessions.
 - Tracks start/end timestamps.
 - Foreign key links sessions to registered users (`user_id`).
 - Indexed for fast session lookups by `user_id` and `status`.

5. Chat Messages

- Table: `chat_messages`
- Purpose: Stores individual messages for each session.
- Features:
 - `sender` indicates user vs bot messages.
 - Soft delete via `deleted_at` and Full-text search on `message` supports content-based queries.
 - Indexed by `session_id` and `timestamp` for chronological retrieval.
 - Foreign key ensures message belongs to a valid session.

6. Chat History

- Table: `chat_history`
- Purpose: Aggregates chat data with user, IP, location, and agent info.
- Features:
 - `user_message` & `bot_response` capture conversation.
 - Supports geo-tracking (`latitude`, `longitude`, `location`) for analytics.
 - Soft delete via `deleted_at`.
 - Foreign key links to `users` table.
 - Indexed for quick retrieval by `created_at`, `user_id`, and `ip_address`.

7. Chatbot Suggestions

- Table: `chatbot_suggestion_logs`
- Purpose: Logs suggested bot responses and user selection.
- Features:
 - `selected` indicates if suggestion was accepted.
 - `deleted_at` supports soft delete.
 - Foreign keys link suggestions to `chat_messages` and `users`.
 - Indexed for analytics and A/B testing via `created_at`, `user_id`, `message_id`.

8. Visitor Tracking

- Table: `chatbot_visitors`
- Purpose: Tracks site visitors including anonymous users.
- Features:
 - Stores IP, hashed IP, user agent, and geo-location (`city`, `region`, `country`, `latitude`, `longitude`).
 - Soft delete via `deleted_at`.
 - Indexed by IP, timestamp, and geo coordinates for analytics and geo-mapping.

9. FAQ / Knowledge Base

- Table: `faq`
- Purpose: Stores bot knowledge base with hierarchical structure.
- Features:
 - `parent_id` allows nested FAQ relationships.
 - `is_suggestion` identifies suggested answers.
 - `is_welcome` flags welcome messages.
 - Soft delete via `deleted_at`.
 - Full-text search on `user_question` supports quick matching.
 - Foreign key ensures hierarchical integrity (`parent_id`).

10. User Activity Logging

- Table: `user_activity_log`
- Purpose: Tracks actions performed by users for auditing and security.
- Features:
 - `action` codes represent specific activities.
 - Logs IP and user agent for traceability.
 - Soft delete via `deleted_at`.
 - Indexed by `user_id` for fast queries.
 - Foreign key ensures activities are linked to valid users.

11. Site Settings

- Table: `site_settings`
 - Purpose: Stores key-value configuration for the system.
 - Features:
 - `name` is primary key ensuring uniqueness.
- Allows dynamic configuration of system behavior.

12. Data Integrity & Referential Constraints

- Foreign keys enforce relationships:
 - `Messages` → `Sessions`
 - `Suggestions` → `Messages & Users`
 - `History` → `Users`
 - `Sessions` → `Users`
 - `FAQ` → `Parent FAQ`
 - `Password resets` → `Users`
 - `Remember tokens` → `Users`
 - `User activity` → `Users`
- Cascading updates/deletes ensure consistent data.

13. Indexing & Performance

- Proper indexes added for:
 - Fast search (`FULLTEXT` on messages and FAQs)
 - Quick retrieval by timestamps, `user_id`, IP, `session_id`, geo coordinates
- Supports analytics, A/B testing, and real-time queries efficiently.

14. Soft Delete Support

- All main tables (`users`, `chat_messages`, `chat_sessions`, `chat_history`, `faq`, `chatbot_suggestion_logs`, `remember_tokens`, `user_activity_log`) include `deleted_at` for compliance and reversible deletions.

15. Seed Data

- Preloaded Admin, User, and Demo accounts for quick setup.

Admin user assigned role `2` and active status.

Summary of Key Developer Capabilities Provided

- Full-featured chat system with sessions, messages, and conversation history.
- Suggestion logs, visitor tracking, and FAQ hierarchy for analytics & AI response optimization.
- Secure authentication, password reset, and persistent login support.
- Soft deletes and auditing for compliance and tracking.
- Indexed and optimized for high performance with `FULLTEXT` search and geo queries.
- Referential integrity ensures data consistency across all relationships.

ER diagram and relationships,

