Creating a database for a messaging app like WhatsApp requires a well-structured schema to manage users, messages, groups, permissions, and chat features. Below is the **database design** along with **table structures and explanations**.

**Database Name: messaging\_app**

**1. Users Table**

Stores user details.

**Table: users**

| **Column Name** | **Data Type** | **Constraints** | **Description** |
| --- | --- | --- | --- |
| user\_id | INT (PK) | AUTO\_INCREMENT | Unique user ID |
| username | VARCHAR(50) | UNIQUE, NOT NULL | Username of the user |
| email | VARCHAR(100) | UNIQUE, NOT NULL | Email of the user |
| phone | VARCHAR(15) | UNIQUE, NOT NULL | Phone number |
| password | VARCHAR(255) | NOT NULL | Hashed password |
| status | VARCHAR(255) | NULL | Status message |
| profile\_pic | TEXT | NULL | Profile picture URL |
| created\_at | TIMESTAMP | DEFAULT CURRENT\_TIMESTAMP | Account creation date |

**2. Contacts Table**

Stores user contacts.

**Table: contacts**

| **Column Name** | **Data Type** | **Constraints** | **Description** |
| --- | --- | --- | --- |
| contact\_id | INT (PK) | AUTO\_INCREMENT | Unique contact ID |
| user\_id | INT (FK) | NOT NULL | Reference to users table |
| contact\_user\_id | INT (FK) | NOT NULL | Reference to users table (contact person) |
| created\_at | TIMESTAMP | DEFAULT CURRENT\_TIMESTAMP | Added date |

**Constraints:**

* user\_id and contact\_user\_id together should be unique to prevent duplicate contacts.

**3. Messages Table**

Stores user messages (one-to-one and group).

**Table: messages**

| **Column Name** | **Data Type** | **Constraints** | **Description** |
| --- | --- | --- | --- |
| message\_id | INT (PK) | AUTO\_INCREMENT | Unique message ID |
| sender\_id | INT (FK) | NOT NULL | User sending the message |
| receiver\_id | INT (FK) | NULL | User receiving the message (NULL if group) |
| group\_id | INT (FK) | NULL | Reference to group chat (NULL for one-to-one) |
| message\_text | TEXT | NULL | Message text |
| message\_type | ENUM | ('text','image','video','audio','file') | Type of message |
| media\_url | TEXT | NULL | URL of media file (if applicable) |
| is\_read | BOOLEAN | DEFAULT FALSE | Read status |
| created\_at | TIMESTAMP | DEFAULT CURRENT\_TIMESTAMP | Sent time |

**Constraints:**

* Either receiver\_id (for one-to-one chat) or group\_id (for group chat) must be provided.
* Foreign keys reference the users and groups tables.

**4. Groups Table**

Stores group details.

**Table: groups**

| **Column Name** | **Data Type** | **Constraints** | **Description** |
| --- | --- | --- | --- |
| group\_id | INT (PK) | AUTO\_INCREMENT | Unique group ID |
| group\_name | VARCHAR(255) | NOT NULL | Name of the group |
| admin\_id | INT (FK) | NOT NULL | User ID of the group admin |
| created\_at | TIMESTAMP | DEFAULT CURRENT\_TIMESTAMP | Creation date |

**5. Group Members Table**

Stores group participants.

**Table: group\_members**

| **Column Name** | **Data Type** | **Constraints** | **Description** |
| --- | --- | --- | --- |
| member\_id | INT (PK) | AUTO\_INCREMENT | Unique member ID |
| group\_id | INT (FK) | NOT NULL | Reference to groups table |
| user\_id | INT (FK) | NOT NULL | User ID of the participant |
| role | ENUM | ('admin','member') | User role in the group |
| joined\_at | TIMESTAMP | DEFAULT CURRENT\_TIMESTAMP | Joined time |

**Constraints:**

* A user can be part of multiple groups.
* Each group must have at least one admin.

**6. Chat Settings Table**

Stores chat-specific settings for users.

**Table: chat\_settings**

| **Column Name** | **Data Type** | **Constraints** | **Description** |
| --- | --- | --- | --- |
| setting\_id | INT (PK) | AUTO\_INCREMENT | Unique setting ID |
| user\_id | INT (FK) | NOT NULL | Reference to users table |
| chat\_type | ENUM | ('one\_to\_one', 'group', 'secret') | Type of chat |
| chat\_id | INT (FK) | NOT NULL | Chat reference (group\_id or user\_id) |
| muted | BOOLEAN | DEFAULT FALSE | Chat muted status |
| archived | BOOLEAN | DEFAULT FALSE | Chat archived status |

**Constraints:**

* Helps in managing chat visibility, mute settings, etc.

**7. Secret Chats Table**

Stores secret chats with encryption.

**Table: secret\_chats**

| **Column Name** | **Data Type** | **Constraints** | **Description** |
| --- | --- | --- | --- |
| chat\_id | INT (PK) | AUTO\_INCREMENT | Unique secret chat ID |
| user\_one | INT (FK) | NOT NULL | First user ID |
| user\_two | INT (FK) | NOT NULL | Second user ID |
| encryption\_key | TEXT | NOT NULL | Encryption key for chat |

**8. Blocked Users Table**

Stores blocked contacts.

**Table: blocked\_users**

| **Column Name** | **Data Type** | **Constraints** | **Description** |
| --- | --- | --- | --- |
| block\_id | INT (PK) | AUTO\_INCREMENT | Unique block ID |
| user\_id | INT (FK) | NOT NULL | User who blocked someone |
| blocked\_user\_id | INT (FK) | NOT NULL | Blocked user ID |
| blocked\_at | TIMESTAMP | DEFAULT CURRENT\_TIMESTAMP | Blocked time |

**Database Relationships**

1. **Users Table** → Connected with messages, contacts, blocked\_users, groups, and chat\_settings.
2. **Messages Table** → Connected with users (sender, receiver) and groups.
3. **Groups Table** → Connected with group\_members, messages.
4. **Secret Chats** → Uses encryption for private messaging.
5. **Blocked Users** → Stores blocked contacts.
6. **Chat Settings** → Stores muted/archived chats.

**Summary**

This database structure supports: ✅ User management  
✅ One-to-one messaging  
✅ Group messaging  
✅ Secret chats with encryption  
✅ Message media support (images, videos, files)  
✅ Mute, archive, and disable chat options  
✅ User blocking system

Yes! These are the **core database models** needed to build a messaging app like WhatsApp. However, depending on additional features you may want, we can extend this schema. Below are some **additional models** that could enhance the app:

**Additional Database Models (Optional Enhancements)**

**9. Message Reactions Table**

Stores reactions (like 👍❤️😂) to messages.

**Table: message\_reactions**

| **Column Name** | **Data Type** | **Constraints** | **Description** |
| --- | --- | --- | --- |
| reaction\_id | INT (PK) | AUTO\_INCREMENT | Unique reaction ID |
| message\_id | INT (FK) | NOT NULL | Reference to messages table |
| user\_id | INT (FK) | NOT NULL | User who reacted |
| reaction | ENUM | ('like','love','haha','wow','sad','angry') | Type of reaction |
| created\_at | TIMESTAMP | DEFAULT CURRENT\_TIMESTAMP | Time of reaction |

**10. Message Read Receipts Table**

Stores read/delivered status of messages.

**Table: message\_read\_receipts**

| **Column Name** | **Data Type** | **Constraints** | **Description** |
| --- | --- | --- | --- |
| receipt\_id | INT (PK) | AUTO\_INCREMENT | Unique ID |
| message\_id | INT (FK) | NOT NULL | Reference to messages table |
| user\_id | INT (FK) | NOT NULL | User who read the message |
| status | ENUM | ('delivered', 'read') | Read or delivered status |
| timestamp | TIMESTAMP | DEFAULT CURRENT\_TIMESTAMP | Time of status update |

**11. Calls Table**

Stores audio/video call logs.

**Table: calls**

| **Column Name** | **Data Type** | **Constraints** | **Description** |
| --- | --- | --- | --- |
| call\_id | INT (PK) | AUTO\_INCREMENT | Unique call ID |
| caller\_id | INT (FK) | NOT NULL | Caller user ID |
| receiver\_id | INT (FK) | NOT NULL | Receiver user ID |
| call\_type | ENUM | ('audio', 'video') | Type of call |
| call\_status | ENUM | ('missed', 'completed', 'declined') | Status of the call |
| duration | INT | DEFAULT 0 | Duration in seconds |
| started\_at | TIMESTAMP | DEFAULT CURRENT\_TIMESTAMP | Call start time |

**12. Call Participants Table**

Stores participants for group calls.

**Table: call\_participants**

| **Column Name** | **Data Type** | **Constraints** | **Description** |
| --- | --- | --- | --- |
| participant\_id | INT (PK) | AUTO\_INCREMENT | Unique participant ID |
| call\_id | INT (FK) | NOT NULL | Reference to calls table |
| user\_id | INT (FK) | NOT NULL | User ID of participant |
| joined\_at | TIMESTAMP | DEFAULT CURRENT\_TIMESTAMP | When they joined |

**13. Notifications Table**

Stores push notifications for messages, calls, etc.

**Table: notifications**

| **Column Name** | **Data Type** | **Constraints** | **Description** |
| --- | --- | --- | --- |
| notification\_id | INT (PK) | AUTO\_INCREMENT | Unique notification ID |
| user\_id | INT (FK) | NOT NULL | User receiving the notification |
| type | ENUM | ('message', 'call', 'group\_invite') | Type of notification |
| message | TEXT | NOT NULL | Notification content |
| is\_read | BOOLEAN | DEFAULT FALSE | Read status |
| created\_at | TIMESTAMP | DEFAULT CURRENT\_TIMESTAMP | Time of notification |

**14. Group Invites Table**

Stores pending group invitations.

**Table: group\_invites**

| **Column Name** | **Data Type** | **Constraints** | **Description** |
| --- | --- | --- | --- |
| invite\_id | INT (PK) | AUTO\_INCREMENT | Unique invite ID |
| group\_id | INT (FK) | NOT NULL | Reference to groups table |
| inviter\_id | INT (FK) | NOT NULL | User who sent the invite |
| invitee\_id | INT (FK) | NOT NULL | User receiving the invite |
| status | ENUM | ('pending', 'accepted', 'declined') | Invite status |
| created\_at | TIMESTAMP | DEFAULT CURRENT\_TIMESTAMP | Time of invite |

**15. Status Updates Table (Like WhatsApp Stories)**

Stores status updates (text, images, videos).

**Table: status\_updates**

| **Column Name** | **Data Type** | **Constraints** | **Description** |
| --- | --- | --- | --- |
| status\_id | INT (PK) | AUTO\_INCREMENT | Unique status ID |
| user\_id | INT (FK) | NOT NULL | User who posted the status |
| status\_text | TEXT | NULL | Text status (optional) |
| media\_url | TEXT | NULL | URL of image/video |
| visibility | ENUM | ('public', 'contacts', 'private') | Who can see it |
| created\_at | TIMESTAMP | DEFAULT CURRENT\_TIMESTAMP | Time of posting |
| expires\_at | TIMESTAMP | NULL | Expiry time (24 hours default) |

**16. Status Views Table**

Stores who viewed a status.

**Table: status\_views**

| **Column Name** | **Data Type** | **Constraints** | **Description** |
| --- | --- | --- | --- |
| view\_id | INT (PK) | AUTO\_INCREMENT | Unique view ID |
| status\_id | INT (FK) | NOT NULL | Reference to status\_updates table |
| viewer\_id | INT (FK) | NOT NULL | User who viewed the status |
| viewed\_at | TIMESTAMP | DEFAULT CURRENT\_TIMESTAMP | Time of viewing |

**17. Admin Reports Table**

Stores reports about users, groups, or messages.

**Table: reports**

| **Column Name** | **Data Type** | **Constraints** | **Description** |
| --- | --- | --- | --- |
| report\_id | INT (PK) | AUTO\_INCREMENT | Unique report ID |
| reported\_by | INT (FK) | NOT NULL | User who filed the report |
| reported\_user | INT (FK) | NULL | Reported user (if applicable) |
| reported\_group | INT (FK) | NULL | Reported group (if applicable) |
| reported\_message | INT (FK) | NULL | Reported message (if applicable) |
| reason | TEXT | NOT NULL | Reason for the report |
| status | ENUM | ('pending', 'resolved', 'dismissed') | Current status |
| created\_at | TIMESTAMP | DEFAULT CURRENT\_TIMESTAMP | Time of report |

**Final Thoughts**

* The **original core models** cover essential **messaging features**.
* The **additional models** handle **reactions, calls, notifications, statuses, and admin reports**.
* This design is **scalable**, meaning you can **add more features** in the future without breaking existing functionality.