

Smart Contact Manager Table Structure

Tables:-

1.User:

2.Contact:

Relationship:-

One User → Many Contacts(OneToMany):

A single user can have multiple contacts associated with them. For instance, if a user is managing their address book, they can store details of several contacts, each uniquely identified in the contact table.

Many Contacts → One User(ManyToOne):

Each contact is associated with only one user. This ensures that every contact record in the contact table belongs to a specific user, preventing ambiguity.

User:-

	Field	Type	Null	Key	Default	Extra
►	id	int	NO	PRI	NULL	
	about	varchar(500)	YES		NULL	
	email	varchar(255)	YES	UNI	NULL	
	image_url	varchar(255)	YES		NULL	
	is_enable	bit(1)	NO		NULL	
	name	varchar(20)	NO		NULL	
	password	varchar(255)	YES		NULL	
	role	varchar(255)	YES		NULL	

1. id

- Type: int
- Nullability: NO (Required)
- Key: Primary Key (PRI)

This field uniquely identifies each user in the database. It is an integer value and serves as the primary key, ensuring each

record has a distinct identity. Typically, this field is auto-incremented, meaning the database automatically assigns a new, unique value when a user is created.

2. about

- Type: varchar(500)
- Nullability: YES (Optional)

The about field stores a brief description or biography of the user, allowing them to share personal or additional information. It can hold up to 500 characters and is optional, so users may leave it blank if they choose not to provide any details.

3. email

- Type: varchar(255)
- Nullability: YES (Optional)
- Key: Unique Constraint (UNI)

The email field is used to store the user's email address and must be unique for each record. It plays a crucial role in user identification and authentication processes, as well as for communication purposes, such as sending notifications or password recovery emails.

4. image_url

- Type: varchar(255)
- Nullability: YES (Optional)

This field stores the URL of the user's profile picture. The application can fetch and display the image on the user's profile page. It is optional, so users who do not upload a profile picture will have this field left empty.

5. is_enable

- Type: bit(1)
- Nullability: NO (Required)

The is_enable field is a boolean flag indicating whether the user's account is active. A value of 1 (true) means the account is enabled, while a value of 0 (false) indicates that the account is disabled or inactive. This is often used for access control or to manage banned or deactivated users.

6. name

- Type: varchar(20)
- Nullability: NO (Required)

The name field stores the user's name, with a maximum length of 20 characters. This is a required field and is often used for displaying the user's identity in the application, such as on profiles, dashboards, or contact lists.

7. password

- Type: varchar(255)
- Nullability: YES (Optional)

This field stores the user's password in a hashed format for security purposes. While it is marked as optional in the schema, it is typically mandatory during user registration. The hashed password ensures that sensitive user data is not stored in plain text.

8. role

- Type: varchar(255)
- Nullability: YES (Optional)

The role field determines the user's role or permission level within the system (e.g., ADMIN, USER, MODERATOR). This is essential for implementing role-based access control, allowing the system to differentiate user privileges based on their roles.

Contact:-

	Field	Type	Null	Key	Default	Extra
►	cid	int	NO	PRI	NULL	
	cemail	varchar(255)	YES		NULL	
	cimage	varchar(255)	YES		NULL	
	cname	varchar(255)	YES		NULL	
	decription	varchar(10000)	YES		NULL	
	phone	varchar(255)	YES		NULL	
	second_name	varchar(255)	YES		NULL	
	work	varchar(255)	YES		NULL	
	user_id	int	YES	MUL	NULL	

1. cid

- Type: int
- Nullability: NO (Required)
- Key: Primary Key (PRI)

This field serves as the primary key for the contact table. It uniquely identifies each contact in the database. Typically, this field is auto-incremented, meaning the database assigns a unique integer value for every new contact.

2. cemail

- Type: varchar(255)
- Nullability: YES (Optional)

The `cemail` field stores the contact's email address. This is an optional field, allowing users to leave it blank if they do not have or wish to store an email address for a contact. When provided, it can be used for communication or notifications related to the contact.

3. `cimage`

- Type: `varchar(255)`
- Nullability: YES (Optional)

This field stores the URL of the contact's profile picture. Like the `cemail` field, it is optional and allows users to associate a visual representation (image) with a contact, making it easier to identify them visually in the application.

4. `cname`

- Type: `varchar(255)`
- Nullability: YES (Optional)

The `cname` field holds the primary name of the contact. This is typically the first name or the full name of the contact. It is optional, but it is often a critical piece of information to identify and manage contacts effectively.

5. `description`

- Type: `varchar(10000)`
- Nullability: YES (Optional)

This field provides a space for detailed notes or descriptions about the contact. With a maximum length of 10,000 characters, it can be used to store extended information, such as personal or professional details, special instructions, or other relevant context.

6. phone

- Type: `varchar(255)`
- Nullability: YES (Optional)

The `phone` field stores the contact's phone number. It is optional but commonly used for direct communication with the contact via calls or messages. Storing it as `varchar` allows flexibility for various phone number formats, including international dialing codes.

7. second_name

- Type: `varchar(255)`
- Nullability: YES (Optional)

The `second_name` field stores an additional name for the contact, such as a last name, middle name, or nickname. This enhances the ability to store full names or other identifying details about the contact.

8. work

- Type: `varchar(255)`
- Nullability: YES (Optional)

The `work` field describes the contact's occupation, profession, or workplace. This information is useful for categorizing or contextualizing contacts based on their professional background.

9. `user_id`

- Type: `int`
- Nullability: YES (Optional)
- Key: Foreign Key (MUL)

This field serves as a foreign key linking the contact to a specific user in the `user` table. It establishes a one-to-many relationship, where a single user can have multiple contacts. This association ensures that contacts are tied to the user who created or owns them.
