# Users Table

Fields:

* **id**: Unique identifier for the user.
* **first\_name**: User's first name.
* **last\_name**: User's last name.
* **last\_name2**: User's second last name (optional).
* **mobile\_phone**: User's mobile phone number.
* **email**: User's email address (must be unique).
* **token**: Token for email or SMS verification (optional).
* **email\_code**: Code sent to the user's email for verification (optional).
* **sms\_code**: Code sent to the user's mobile phone for verification (optional).
* **password**: User's password (stored securely).
* **token\_date**: Date when the token was generated (optional).
* **active**: Indicates if the user's account is active (default is 1).
* **validated\_email**: Indicates if the user's email has been validated (default is 0).
* **email\_validation\_attempts**: Number of attempts made to validate the email (default is 0).
* **password\_change\_date**: Date when the password was last changed (optional).
* **password\_change\_request**: Indicates if there has been a request to change the password (default is 0).
* **login\_attempts**: Number of login attempts made by the user (default is 0).
* **password\_change\_attempts**: Number of attempts to change the password (default is 0).
* **password\_change\_request\_date**: Date when the password change request was made (optional).
* **admin**: Indicates if the user has administrative privileges (default is 0).

CREATE TABLE users (

id INT(11) PRIMARY KEY AUTO\_INCREMENT,

first\_name VARCHAR(50) NOT NULL,

last\_name VARCHAR(50) NOT NULL,

last\_name2 VARCHAR(50),

mobile\_phone VARCHAR(20) NOT NULL,

email VARCHAR(100) NOT NULL UNIQUE,

token VARCHAR(255),

email\_code VARCHAR(10),

sms\_code VARCHAR(10),

password VARCHAR(255) NOT NULL,

token\_date DATETIME,

active TINYINT(1) DEFAULT 1, -- Indicates if the user is active (1 = active, 0 = inactive)

validated\_email TINYINT(1) DEFAULT 0,

email\_validation\_attempts INT(11) DEFAULT 0,

password\_change\_date DATETIME,

password\_change\_request TINYINT(1) DEFAULT 0,

login\_attempts INT(11) DEFAULT 0,

password\_change\_attempts INT(11) DEFAULT 0,

password\_change\_request\_date DATETIME,

admin TINYINT(1) DEFAULT 0 -- Indicates if the user has admin privileges (1 = admin, 0 = not admin)

);

# Password History Table

Fields:

* **id**: Unique identifier for the password history entry.
* **user\_id**: Foreign key referencing the users table to associate the password with a specific user.
* **password**: The password stored (should be stored securely using hashing).
* **change\_date**: The date when the password was changed.

CREATE TABLE password\_history (

id INT(11) PRIMARY KEY AUTO\_INCREMENT,

user\_id INT(11) NOT NULL, -- Foreign key referencing users

password VARCHAR(255) NOT NULL,

change\_date DATETIME NOT NULL,

FOREIGN KEY (user\_id) REFERENCES users(id) ON DELETE CASCADE -- Ensures that history is deleted if the user is deleted

);

# Roles Table

Fields:

* **id**: Unique identifier for the role.
* **role\_name**: The name of the role (must be unique).
* **description**: A description of the role's responsibilities and permissions.

CREATE TABLE roles (

id INT(11) PRIMARY KEY AUTO\_INCREMENT,

role\_name VARCHAR(50) NOT NULL UNIQUE, -- Name of the role

description VARCHAR(255) -- Description of the role

);

# User Roles Table

Fields:

* **id**: Unique identifier for the user-role association.
* **user\_id**: Foreign key referencing the users table.
* **role\_id**: Foreign key referencing the roles table.

CREATE TABLE user\_roles (

id INT(11) PRIMARY KEY AUTO\_INCREMENT,

user\_id INT(11) NOT NULL, -- Foreign key referencing users

role\_id INT(11) NOT NULL, -- Foreign key referencing roles

FOREIGN KEY (user\_id) REFERENCES users(id) ON DELETE CASCADE,

FOREIGN KEY (role\_id) REFERENCES roles(id) ON DELETE CASCADE

);

# Permissions Table

Fields:

* **id**: Unique identifier for the permission.
* **permission\_name**: Name of the permission (must be unique).
* **description**: Description of what the permission allows the user to do.

CREATE TABLE permissions (

id INT(11) PRIMARY KEY AUTO\_INCREMENT,

permission\_name VARCHAR(50) NOT NULL UNIQUE, -- Name of the permission

description VARCHAR(255) -- Description of the permission

);

# Roles Permissions Table

Fields:

* **id**: Unique identifier for the role-permission association.
* **role\_id**: Foreign key referencing the roles table, indicating which role the permission is associated with.
* **permission\_id**: Foreign key referencing the permissions table, indicating which permission is granted.
* **UNIQUE KEY (role\_id, permission\_id)**: This ensures that a specific permission can only be assigned to a role once.

CREATE TABLE roles\_permissions (

id INT(11) PRIMARY KEY AUTO\_INCREMENT,

role\_id INT(11) NOT NULL, -- Foreign key referencing roles

permission\_id INT(11) NOT NULL, -- Foreign key referencing permissions

FOREIGN KEY (role\_id) REFERENCES roles(id) ON DELETE CASCADE,

FOREIGN KEY (permission\_id) REFERENCES permissions(id) ON DELETE CASCADE,

UNIQUE KEY (role\_id, permission\_id) -- Prevent duplicate permission assignments for the same role

);