Organizational Networking Platform

~ Tushar Kolte

1. Overview

This database design models an **Organizational Networking Platform** where users can:

- Register and manage profiles
- Join multiple organizations with different membership types
- Connect socially with other users (leader/follower)
- Participate in an **Organizational Tree (Org Tree)** built from positions
- Move within the organization while maintaining history

2. Entity Responsibilities

2.1 User

The `user` table stores all registered users.

- Each user has a unique username and email
- Stores login credentials and profile information
- A user can:
    - Join multiple organizations
    - Hold positions
    - Follow other users
    - Request password recovery

2.2 Organization

The `organization` table represents independent organizations.

- Each organization:
    - Has its own members
    - Has its own Organizational Tree

- Organizations are not dependent on users and exist independently.

2.3 Organization Membership

The `user_org_membership` table connects users and organizations.

- Defines **how** a user is connected to an organization
- Membership types:
    - **Full-Time** → can occupy positions
    - **Part-Time** → can occupy positions
    - **Follower** → receives updates only
- Memberships require approval using a status field:
    - Pending / Approved / Rejected

This table enforces the rule that **followers do not appear in the Org Tree**.

2.4 Organizational Tree (Positions)

The `position` table represents the Org Tree.

- Each position:
    - Belongs to exactly one organization
    - May reference another position as its parent (hierarchy)
    - May be vacant or assigned to one user
- Founder positions are marked using a boolean flag
- Founders always appear at the top of the tree

2.5 Position Assignment History

The `position_history` table tracks changes over time.

- Records which user held which position and when
- Preserves historical movement within the organization
- Ensures no data is lost when users move or leave

2.6 User Connections (Leader / Follower)

The `user_user_connection` table models social relationships.

- Represents leader–follower connections between users
- Completely independent of organizations
- Uses status to track request approval

This cleanly separates **social networking** from **organizational structure**.

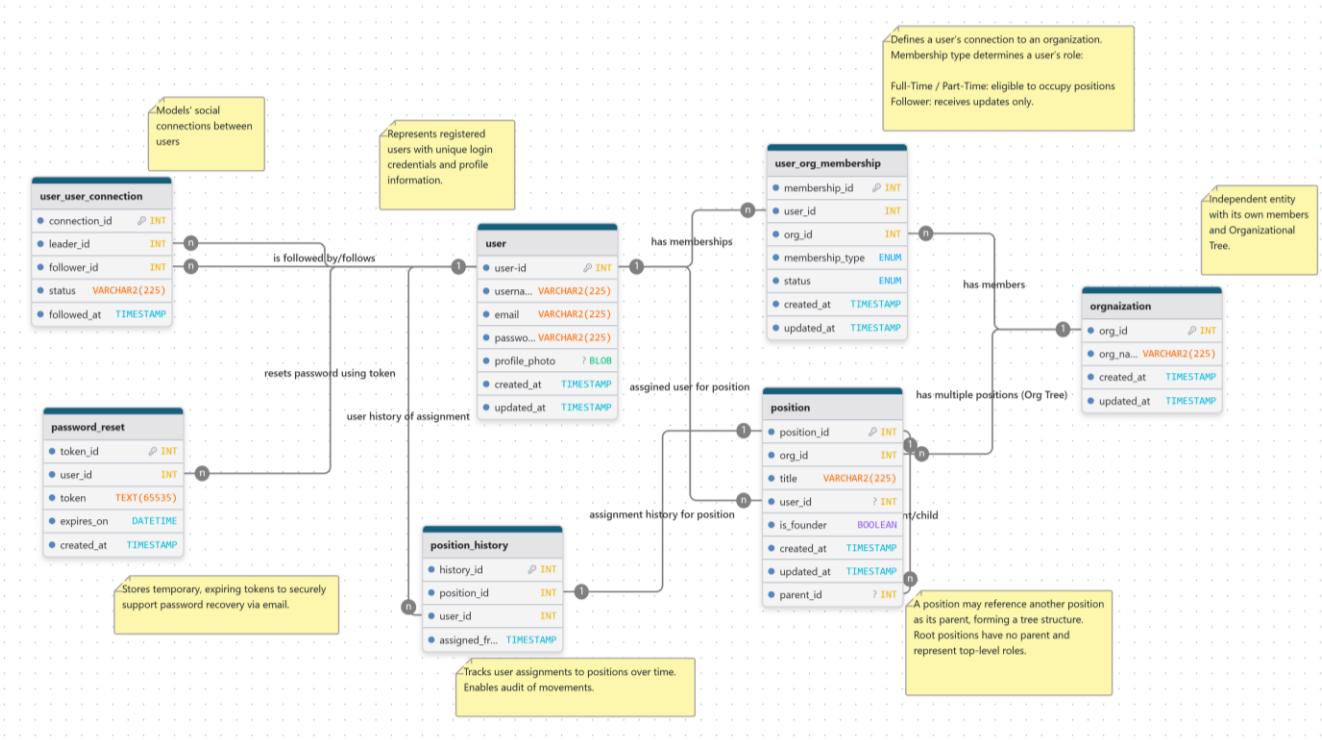
2.7 Password Recovery

The `password_reset` table supports password recovery.

- Stores temporary, expiring reset tokens
- Tokens are linked to users
- Enables secure password reset via email
- Tokens are single-use and time-limited

3. ER Diagram

This ER diagram cleanly models organizational hierarchy, membership types, social connections, and history tracking while remaining simple, flexible, and aligned with all given requirements.


Defines a user's connection to an organization. Membership type determines a user's role:
Full-Time / Part-Time: eligible to occupy positions
Follower: receives updates only.
Models' social connections between users
Represents registered users with unique login credentials and profile information.
Independent entity with its own members and Organizational Tree.
user_user_connection
connection_id INT
leader_id INT
follower_id INT
status VARCHAR2(225)
followed_at TIMESTAMP
is followed by/follows
user
user-id INT
userna... VARCHAR2(225)
email VARCHAR2(225)
passwo... VARCHAR2(225)
profile_photo ? BLOB
created_at TIMESTAMP
updated_at TIMESTAMP
has memberships
user_org_membership
membership_id INT
user_id INT
org_id INT
membership_type ENUM
status ENUM
created_at TIMESTAMP
updated_at TIMESTAMP
has members
orgnaization
org_id INT
org_na... VARCHAR2(225)
created_at TIMESTAMP
updated_at TIMESTAMP
resets password using token
user history of assignment
asssgined user for position
has multiple positions (Org Tree)
password_reset
token_id INT
user_id INT
token TEXT(65535)
expires_on DATETIME
created_at TIMESTAMP
position
position_id INT
org_id INT
title VARCHAR2(225)
user_id ? INT
is_founder BOOLEAN
created_at TIMESTAMP
updated_at TIMESTAMP
parent_id ? INT
assignment history for position
nt/child
position_history
history_id INT
position_id INT
user_id INT
assigned_fr... TIMESTAMP
Stores temporary, expiring tokens to securely support password recovery via email.
Tracks user assignments to positions over time. Enables audit of movements.
A position may reference another position as its parent, forming a tree structure. Root positions have no parent and represent top-level roles.