Organizational Networking Platform Relational Database Design Documentation

1. Introduction

This document presents the relational database design for an **Organizational Networking Platform**.
The system allows users to join organizations, participate in hierarchical organizational structures, follow other users socially, and submit various workflow requests (join, movement, position creation, etc.).

The database is designed with normalization, clarity, scalability, and future extensibility in mind.

2. Objectives of the System

- Manage users and their authentication details
- Allow users to join multiple organizations
- Maintain role-based access and membership types
- Construct hierarchical organizational trees using positions
- Track user assignments to positions with history
- Enable social following between users
- Handle connection and movement requests with approval workflow

4. Entity Descriptions

4.1 USER

Stores login credentials and profile details for every user.

4.2 ORG

Represents an organization; each organization has its own Org Tree.

4.3 USER-ORGANIZATION

Junction table linking users with organizations while storing membership type.

4.4 Position

Represents a node in the organizational tree.
Each position belongs to one organization and may have a parent position (hierarchy).

4.5 position_assign

Tracks assignment of users to positions including start and end date (history tracking).

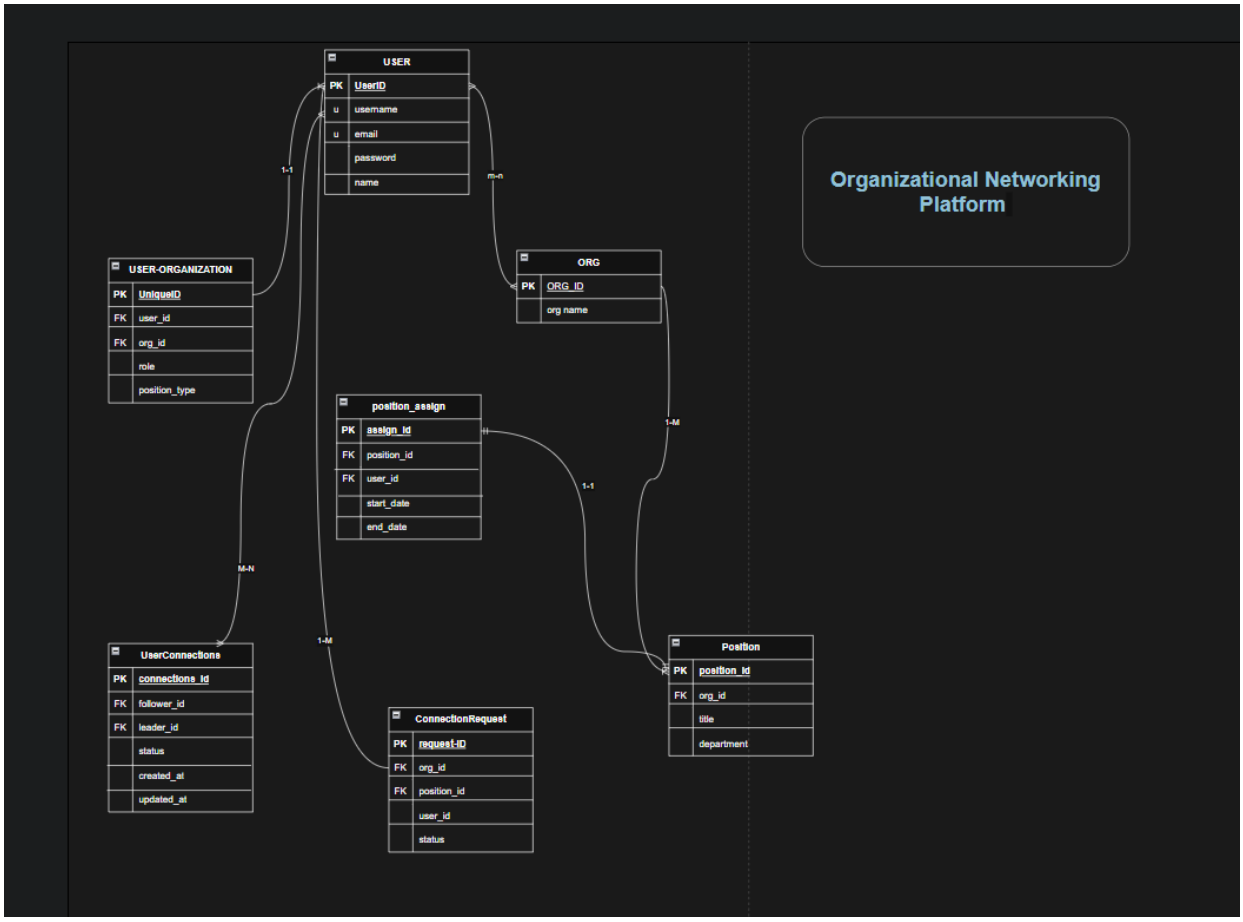
4.6 UserConnections

Represents social follower-leader relationships between users.

4.7 ConnectionRequest

Stores join/movement/position/follow requests with lifecycle states.

5. ER Diagram



6. Relationship Explanation

1. USER ↔ USER-ORGANIZATION (Many-to-Many)

A single user may join multiple organizations.
An organization may have many users.
Handled using a junction table USER-ORGANIZATION.

2. ORG → Position (One-to-Many)

Each organization can have multiple positions in its hierarchy.

3. Position → Position (Recursive One-to-Many)

Positions form a hierarchical tree using parent_position_id.

4. Position ↔ position_assign (One-to-One / One-to-Zero)

A position can be:

- Vacant (no assign row)
- Assigned to exactly one user (unique constraint)

5. USER → position_assign (One-to-Many)

A user can be assigned to multiple positions over time (movement, promotion).

6. USER ↔ USER via UserConnections (Many-to-Many)

A user can follow multiple users and can be followed by many.

7. USER → ConnectionRequest (One-to-Many)

A user can raise multiple requests.

8. ORG → ConnectionRequest (Optional One-to-Many)

Only applicable for join/movement/position-related requests.

9. Position → ConnectionRequest (Optional One-to-Many)

Used for movement and position update requests.

7. Table Definitions (Summary)

Table: USER

- user_id (PK)
- username
- email
- password
- name

Table: ORG

- org_id (PK)
- org_name

Table: USER-ORGANIZATION

- userOrg_id (PK)
- user_id (FK)
- org_id (FK)
- role
- position_type

Table: Position

- position_id (PK)
- org_id (FK)
- parent_position_id (FK, NULL)
- title
- department

Table: position_assign

- assign_id (PK)
- position_id (FK, UNIQUE active)
- user_id (FK)

- start_date
- end_date (NULL = active)

Table: UserConnections

- connection_id (PK)
- follower_id (FK)
- leader_id (FK)
- status
- created_at
- updated_at

Table: ConnectionRequest

- request_id (PK)
- org_id (FK, optional)
- position_id (FK, optional)
- requester_id (FK)
- status
- created_at
- updated_at

10. Conclusion

The presented relational database schema fulfills all functional requirements of the Organizational Networking Platform, including user management, organization membership, hierarchical structures, position assignment, social relationships, and request tracking.

It supports scalability, maintains data integrity, and ensures clean separation of concerns through well-defined relationships and normalization.