# Task Management System - Database Structure

## 1. users Table

CREATE TABLE users (  
 id SERIAL PRIMARY KEY,  
 name VARCHAR NOT NULL,  
 email VARCHAR UNIQUE NOT NULL,  
 hashed\_password VARCHAR NOT NULL,  
 department VARCHAR NOT NULL,  
 role VARCHAR NOT NULL,  
 manager\_id INTEGER REFERENCES users(id), -- hierarchy link (self-referencing FK)  
 created\_at TIMESTAMP WITH TIME ZONE DEFAULT NOW() NOT NULL  
);

🔹 `manager\_id` lets you build the tree hierarchy (CEO → Manager → Team Lead → Executive).

🔹 Recursive CTEs (`WITH RECURSIVE`) can fetch subordinates easily.

## 2. tasks Table

CREATE TABLE tasks (  
 id SERIAL PRIMARY KEY,  
 title VARCHAR NOT NULL,  
 description TEXT,  
 priority VARCHAR NOT NULL CHECK (priority IN ('Low', 'Medium', 'High', 'Critical')),  
 status VARCHAR DEFAULT 'NEW' NOT NULL CHECK (status IN ('NEW', 'IN\_PROGRESS', 'PENDING', 'FINISHED', 'STOPPED', 'CANCELLED')),  
 start\_date TIMESTAMP WITH TIME ZONE NOT NULL,  
 due\_date TIMESTAMP WITH TIME ZONE NOT NULL,  
 follow\_up\_date TIMESTAMP WITH TIME ZONE,  
 tags TEXT[] DEFAULT '{}',  
 created\_by INTEGER REFERENCES users(id),  
 assigned\_to INTEGER REFERENCES users(id),  
 escalated\_to INTEGER REFERENCES users(id), -- escalation target (optional)  
 created\_at TIMESTAMP WITH TIME ZONE DEFAULT NOW() NOT NULL,  
 updated\_at TIMESTAMP WITH TIME ZONE DEFAULT NOW() NOT NULL  
);

🔹 Added `escalated\_to` for escalation tracking.

🔹 Used `CHECK` constraints to enforce valid `priority` and `status`.

## 3. task\_observers Table

CREATE TABLE task\_observers (  
 id SERIAL PRIMARY KEY,  
 task\_id INTEGER REFERENCES tasks(id) ON DELETE CASCADE,  
 user\_id INTEGER REFERENCES users(id) ON DELETE CASCADE,  
 created\_at TIMESTAMP WITH TIME ZONE DEFAULT NOW() NOT NULL,  
 UNIQUE (task\_id, user\_id)  
);

🔹 Allows multiple people to 'watch' a task.

🔹 Observers get reminders & notifications.

## 4. task\_logs Table

CREATE TABLE task\_logs (  
 id SERIAL PRIMARY KEY,  
 task\_id INTEGER REFERENCES tasks(id) NOT NULL ON DELETE CASCADE,  
 title VARCHAR NOT NULL,  
 description TEXT,  
 start\_time TIMESTAMP WITH TIME ZONE NOT NULL,  
 end\_time TIMESTAMP WITH TIME ZONE NOT NULL  
);

## 5. notifications Table

CREATE TABLE notifications (  
 id SERIAL PRIMARY KEY,  
 user\_id INTEGER REFERENCES users(id) NOT NULL,  
 task\_id INTEGER REFERENCES tasks(id),  
 type VARCHAR NOT NULL CHECK (type IN ('REMINDER', 'ALERT', 'ESCALATION', 'INFO')),  
 title VARCHAR NOT NULL,  
 message TEXT NOT NULL,  
 is\_read BOOLEAN DEFAULT FALSE NOT NULL,  
 created\_at TIMESTAMP WITH TIME ZONE DEFAULT NOW() NOT NULL,  
 read\_at TIMESTAMP WITH TIME ZONE  
);

🔹 Added stricter `CHECK` for `type`.

## 6. escalations Table

CREATE TABLE escalations (  
 id SERIAL PRIMARY KEY,  
 task\_id INTEGER REFERENCES tasks(id) NOT NULL,  
 escalated\_by INTEGER REFERENCES users(id) NOT NULL,  
 escalated\_to INTEGER REFERENCES users(id) NOT NULL,  
 reason TEXT,  
 created\_at TIMESTAMP WITH TIME ZONE DEFAULT NOW() NOT NULL  
);

🔹 Keeps history of who escalated tasks, to whom, and why.

🔹 Useful for accountability reports.

## 7. reports Table

CREATE TABLE reports (  
 id SERIAL PRIMARY KEY,  
 type VARCHAR NOT NULL CHECK (type IN ('INDIVIDUAL', 'TEAM', 'MONTHLY', 'WEEKLY')),  
 generated\_for INTEGER REFERENCES users(id),  
 period\_start DATE NOT NULL,  
 period\_end DATE NOT NULL,  
 file\_url TEXT,  
 created\_at TIMESTAMP WITH TIME ZONE DEFAULT NOW() NOT NULL  
);

🔹 Stores report metadata for history & downloads.