DataBase

AUCA

PROJECT MANAGEMENT SYSTEM

Kaiyrkul Kanatbekova

Introduction

Purpose of the project

WHY DO WE NEED DBMS FOR PROJECT MANAGEMENT?

Develop a relational database for centralized storage and management of all project data (Tasks, Employees, Deadlines, Statuses).

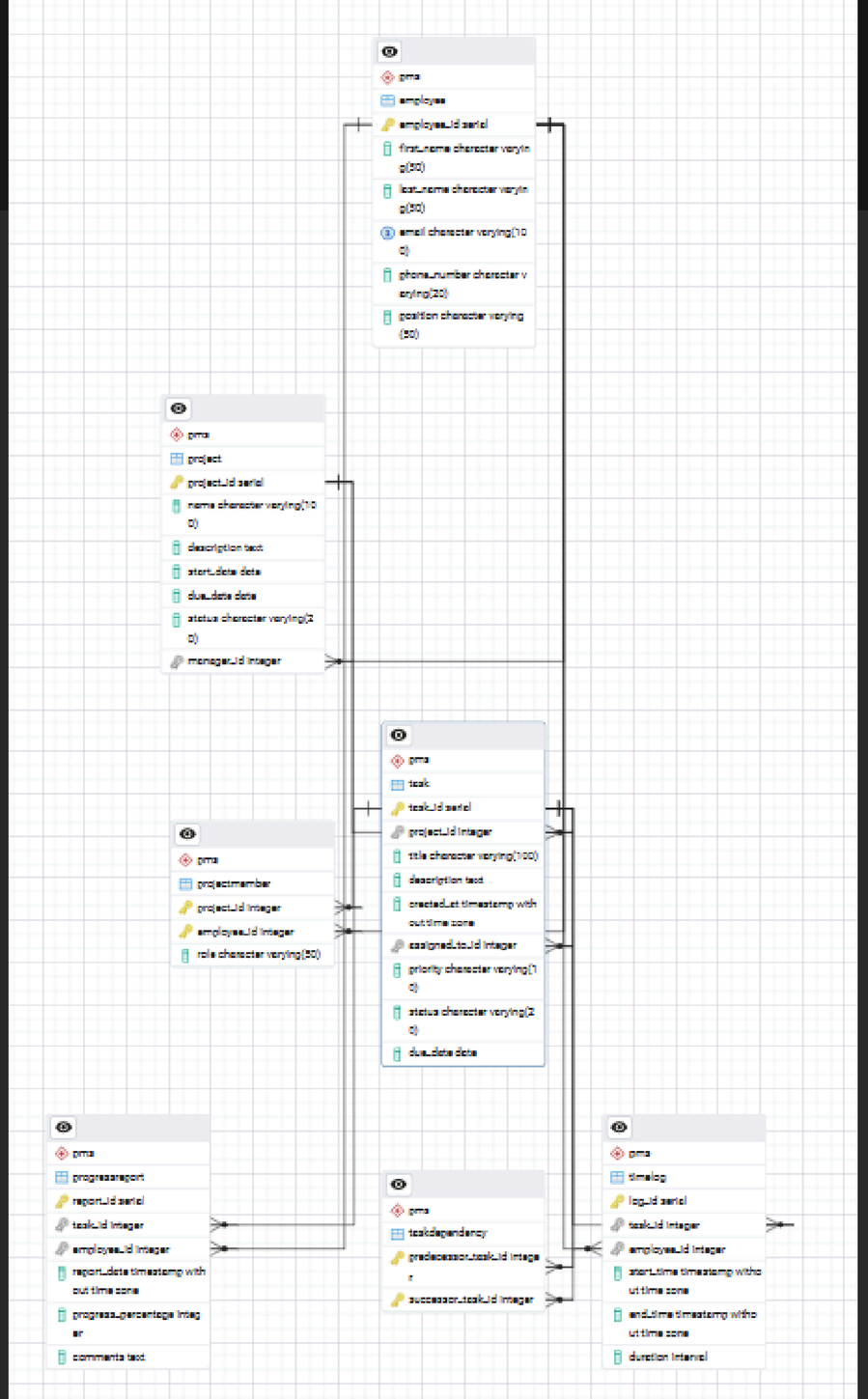
SYSTEM REQUIREMENTS

/1

Entities: We have modeled 6 main business objects:

- Employee
- Project (Projects, including manager_id)
- Task (Tasks with priority and status fields)
- TimeLog (Accounting of working hours)
- Progress Report
- Binders: ProjectMember (M:N relationship) and TaskDependency (Task Dependency).

Table name
employee
progressreport
project
projectmember
task
taskdependency
timelog



DATABASE SCHEMA (E-R MODEL)

- Project to Task relationship: 1:M (ON DELETE CASCADE).
- Employee-Task Relationship: 1:M (each task is assigned to one assigned_to_id).
- Using CHECK limits for statuses (Active, InProgress, Done) and priorities (Low, Medium, High, Urgent).
- Date verification: CHECK (due_date IS NULL OR due_date >= start_date).

BASIC FUNCTIONALITY (CRUD)

Purpose: To demonstrate basic data management operations.

- CREATE (INSERT): Adding a new employee (Bermet Dadibayeva).
- READ (SELECT): Getting active projects or high priority tasks.
- UPDATE: Transferring the task to the Done (Migration of the database schema...) status.
- DELETE: Deleting progress reports on a specific comment

```
CREATE TABLE IF NOT EXISTS pms.projectmember
(
    project_id integer NOT NULL,
    employee_id integer NOT NULL,
    role character varying(50) COLLATE pg_catalog."default" NOT NULL,
    CONSTRAINT projectmember_pkey PRIMARY KEY (project_id, employee_id),
    CONSTRAINT projectmember_employee_id_fkey FOREIGN KEY (employee_id)
        REFERENCES pms.employee (employee_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE CASCADE,
    CONSTRAINT projectmember_project_id_fkey FOREIGN KEY (project_id)
        REFERENCES pms.project (project_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE CASCADE
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS pms.projectmember
    OWNER to postgres;
```

TRANSACTIONS AND OPTIMIZATION

1. Transactions: Demonstration of BEGIN/COMMIT (to atomically transfer a task to a completed project) and ROLLBACK (to cancel a test operation).
2. Indexes: Create multi-column indexes (idx_task_status_priority) to speed up queries.
3. Performance Analysis: Using EXPLAIN ANALYSIS to evaluate query performance.