Executive Summary – Bakhtiyar Yesbolsyn

What Was Accomplished

- Complete normalized database schema with proper relationships
- Docker containerization for local development with PostgreSQL
- Cloud deployment on PythonAnywhere with Supabase PostgreSQL
- All 14 SQL queries functioning correctly (updates, deletes, joins, aggregations, views)
- Full-stack web application with complete CRUD operations for all entities
- Professional responsive UI with Bootstrap 5

Technology Stack

PostgreSQL 15 (Docker/Supabase), Python 3, SQLAlchemy 2.0, Flask 3.0, Bootstrap 5, PythonAnywhere

Part 1: Database Design

Designed and implemented a normalized relational schema with 7 tables: `users`, `caregiver`, `member`, `address`, `job`, `job_application`, and `appointment`. The schema uses one-to-one relationships (users to caregivers/members) and many-to-many relationships (jobs to caregivers via applications). Foreign key constraints with CASCADE deletes ensure referential integrity. Populated with 10+ realistic records per table, including required test data (Arman Armanov, Amina Aminova, Kabanbay Batyr addresses, "soft-spoken" job requirements).

Part 2: Query Operations

Implemented all 14 required database operations using SQLAlchemy. All queries execute successfully without errors and return expected results.

Part 3: Web Application

Built a Flask-based CRUD application with complete data management for all 5 main entities. Each entity has modal forms for creation/editing, dropdown selections for foreign keys, confirmation prompts for deletions, and responsive Bootstrap 5 interface with navigation between all pages.

Deployment

Application was deployed to PythonAnywhere cloud platform with Supabase PostgreSQL database for production use. Environment variables used for secure database connection management. Accessible via web interface from any device.