Bazy danych projekt

Veronika Sazanovich
Group: 21IiE-SP

Date: June 21, 2024

1 Introduction

This program implements a management system that integrates functionalities related to user registration, task management, company representation, and interactive messaging within a GUI framework. Users can register using their username, password, and associated company ID, which also links them as representatives in the Representative table. The system supports login functionalities with role-based access to different interfaces. Our logical and relational model of the database looks like:

1.1 Logical Model

The logical model diagram visually represents the data and their relationships within our system. It serves as a blueprint for the database structure, showing various entities such as users, tasks, representatives, and companies, along with their attributes. This model helps in understanding the data flow and the essential fields that are necessary for the functioning of our system.

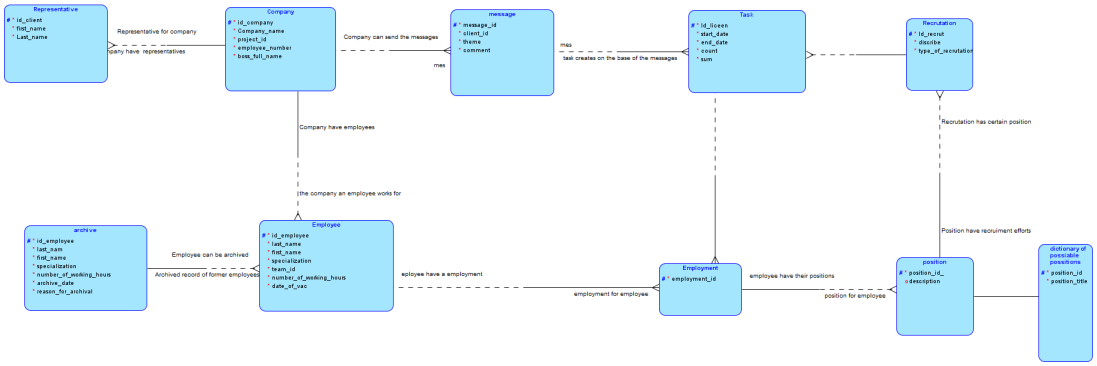

Figure 1: Logical model

1.2 Relational Model

The relational model further refines the logical model by outlining the specific relationships between tables. It defines the primary keys, foreign keys, and constraints that ensure data integrity and relational links between different data points.

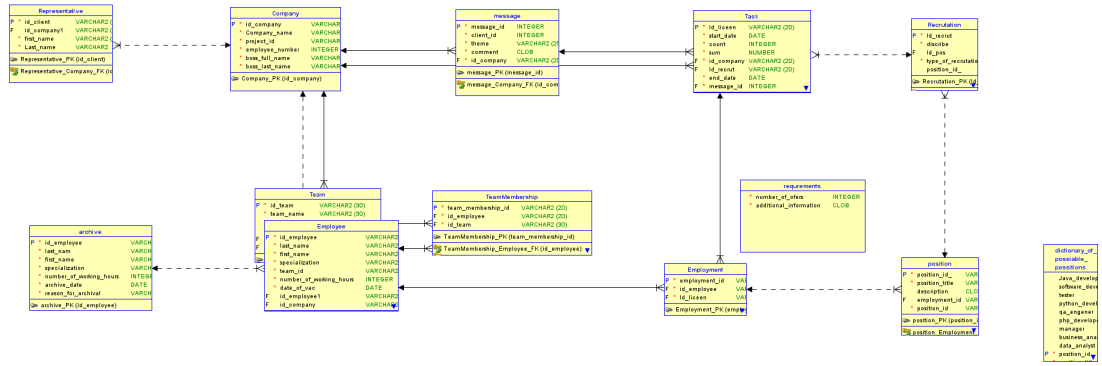

Figure 2: Relational model

First, we connect our program written in the Python language to the SQL database.

```
def database_connection():
    return mysql.connector.connect(
        host='127.0.0.1',
        port='3306',
        database='dbproject',
        user='root',
        password=''
    )
```

The entire program is implemented in Python using the Tkinter GUI. Below are a few initial steps for creating the application in Tkinter. This program is a comprehensive employee management system developed using Python and the Tkinter library. It interfaces with a MySQL database to manage users, employees, tasks, and messages. The primary functionalities are divided across four main pages: Login, Registration, Client Interface, and Admin Interface.

1.3 Login Page

The login page allows existing users to log in by entering their username and password. If the credentials are correct and the user's status is approved, they are directed to the appropriate interface based on their role (admin or client).

1.4 Registration Page

The registration page enables new users to register by providing necessary details such as username, password, company ID, first name, and last name. The registration request is stored in the database with a pending status until an admin approves it.

1.5 Client Interface

Clients can use this interface to view their user ID, send messages to the manager, view their task history, and access company statistics. The client can also contact the manager through a form, and the message is stored in the database for the admin to review.

1.6 Admin Interface

The admin interface offers extensive functionalities to manage employees and tasks. Admins can:

- **Add Employee:** Fill out a form to add new employee details to the database.
- **Modify Employee:** Select an employee from a list, modify their details, and update the database.
- **Delete Employee:** Select an employee to delete. The employee's data is archived before deletion from the main table.
- **View Applications:** Review and approve or reject user registration requests.
- **Add Task:** Create new tasks by providing details such as company ID, start and end dates, count, and sum.

1.7 Database Operations

The program uses MySQL to store and retrieve data. Functions are defined to connect to the database and execute SQL queries for various operations, including fetching specializations, adding and modifying employees, sending messages, and calculating task costs.

1.8 Statistics and Visualization

Admins and clients can view company statistics, displayed as bar charts using Matplotlib. The statistics show the sum of tasks by company.

1.9 Interface and User Experience

The user interface is built with Tkinter, providing a responsive and interactive experience. Users can navigate between different pages, fill out forms, view data in tree views, and interact with various widgets like buttons and entry fields.

2 Conclusion

This employee management system efficiently integrates various functionalities essential for managing users, tasks, and communications within a company. By leveraging a logical and relational model, the system ensures data integrity and effective relationships between entities. The use of Tkinter for the GUI and MySQL for database operations provides a robust and user-friendly interface for both administrators and clients. This system not only streamlines employee management but also enhances the overall operational efficiency of the organization.