

# Mentorship & Research Collaboration Matching System

Name: Brandon Ismalej

Course: COMP440 - Database Design

## Overview

This mentorship and collaboration database project is backed by a MySQL relational database which has been designed to manage academic profiles, support flexible search and matching features, and maintain clean, normalized relationships between users and their stated research interests. The database schema, core functions, query logic, and data management strategies for the system are described below:

## Database Schema & Normalization

The schema consists of three main tables, normalized to third normal form, to ensure that redundant data is eliminated and updates remain consistent:

1. users - Stores the primary profile information:
  - a. id: Primary key
  - b. name, email, academic\_position
  - c. institution, department, bio
  - d. interested\_in: Comma-separated roles
  - e. headshot\_path: Path to uploaded or default profile image
2. Research\_interests - A lookup table of all possible interests:
  - a. id: Primary key
  - b. name: Unique interest name
3. user\_research\_interests - A join table to handle the many-to-many relationship:
  - a. user\_id: Foreign key to users
  - b. interest\_id: Foreign key to research\_interests

## Insertion & Creation Features

When a new profile is created and submitted through the web GUI

- User data is inserted into the users table
- Research interests are selected by ID, or added through user input. New interests are inserted with:
  - IGNORE INTO research\_interests (name) VALUES (...);
- Selected interests are added into user\_research\_interests with:
  - INSERT INTO user\_research\_interests (user\_id, interest\_id);
- Optional headshot files are saved with secure\_filename, and users who skip this image upload are assigned to a default image

## Search Functionality

- Keyword-based searches on name, institution, and department use SQL LIKE queries to allow partial string matches:
  - `SELECT * FROM users WHERE department LIKE "%computer science%"`
- Filter-based searches use checkboxes for position and role. These are processed into SQL IN clauses to find profiles with matching categories:
  - `SELECT * FROM users WHERE academic_position IN ("Faculty", "Postdoc")`
- For matching comma-separated interested\_in fields, FIND\_IN\_SET is used to match exact roles within a single column:
  - `SELECT * FROM users WHERE FIND_IN_SET('Providing Mentorship', interested_in)`

## Matching Functionality

The matching logic compares the intent of the querying user (the user wanting mentorship or collaboration, for example) with others who can offer the corresponding role. This logic works by:

- Checking the both users share at least one research interests, through joins on user\_research\_interests
- Verifying role compatibility with FIND\_IN\_SET

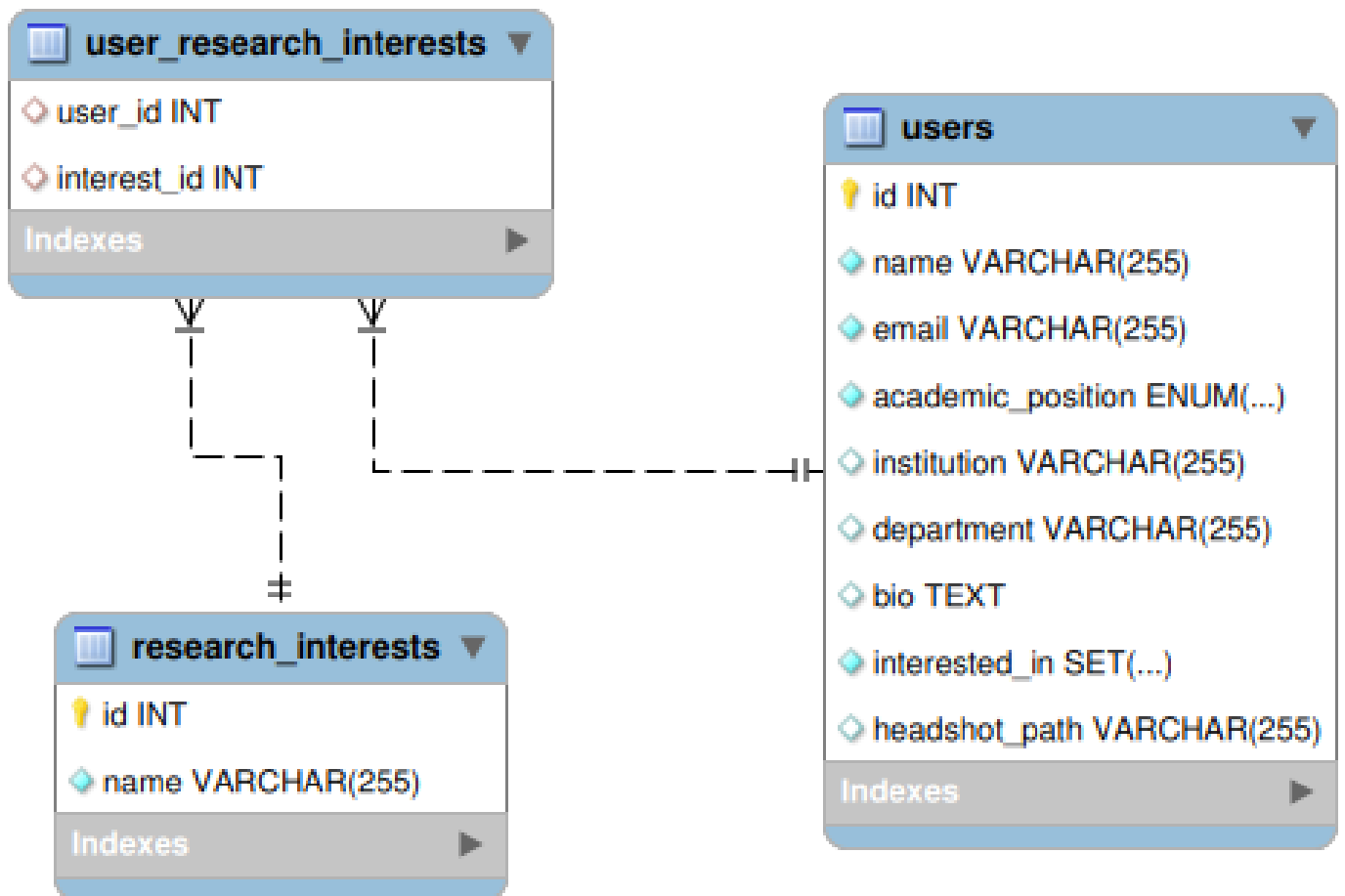
## Updating and Deleting Profiles

- Users update their profiles by providing the same email used during creation. The system fetches the user by email and allows changes to all editable fields.
- Deleting profiles requires confirmation and removes the user from users, along with cascading deletions from user\_research\_interests.
- All updates and deletions are executed through SQL UPDATE and DELETE statements within the Flask routes.

## Backup

- To support the reproducibility of live demonstration, the system includes a backup SQL dump file containing the full schema and a set of dummy profiles. The file is created using:
  - `mysqldump -u root -p mentorship_db > mentorship_db_backup.sql`

## EM Diagram



## How to Run Project:

---

### Requirements:

- Python 3.10+
- MySQL server
- Flask + MySQL bindings

### 1. Install Python Dependencies:

- `$ pip install -r requirements.txt`

### 2. Create the MySQL Database

- `$ mysql -u root -p`

### Then inside the MySQL shell:

- `$ CREATE DATABASE mentorship_db;`
- `$ EXIT;`

### 3. Import Schema and Data

- `$ mysql -u root -p mentorship_db < sql/mentorship_db_backup.sql`

### 4. Run the App

- `$ python app.py`

Visit <http://127.0.0.1:5000> in your browser.