The Islamia University of Bahawalpur Pakistan

Project Database Systems

Spring-2025 Department of Computer Science

03- June-2025

Submitted By:

Hassan Raza
Roll No. F24BDOCS1M01442
Subject: Database
BSCS (2^{nd} Semester)
Section 7M

Submitted To:

Professor Muhammad Usman Ghani.

Title: University Professors and Organizations Management System.

Introduction:

1.1.Background:

In universities, many professors work with different organizations. But managing all this data manually is hard and confusing. So, we created a database that stores information about **universities**, **professors**, and the **organizations** they are connected to. This helps keep everything in one place, neat and easy to access.

1.2. Goal:

The goal of this project is to:

- Save data about universities and professors.
- Show which professor is connected to which organization.
- Make it easy to **search**, **add**, **update**, or **delete** any data.

1.3. Requirements:

- What users need from this system:
- Save basic info about universities (like name and city).
- Store professor details (like their name and which university they belong to).
- Keep records of organizations (like name and sector).
- Connect professors to the organizations they work with.
- Make sure all the links between tables are correct and no data is lost.

2. Functional Description:

2.1. Method of use:-

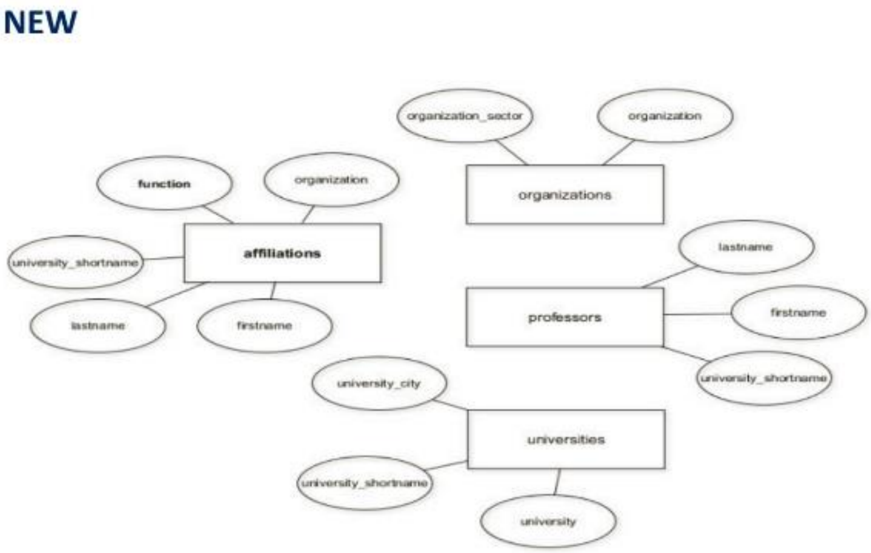
This system will be used by:

- **University staff** to manage professors.
- **Students** to see where professors work or do research.
- **Database admins** to maintain clean and correct data.

3. Entity Data Model:-

We have 4 main tables:

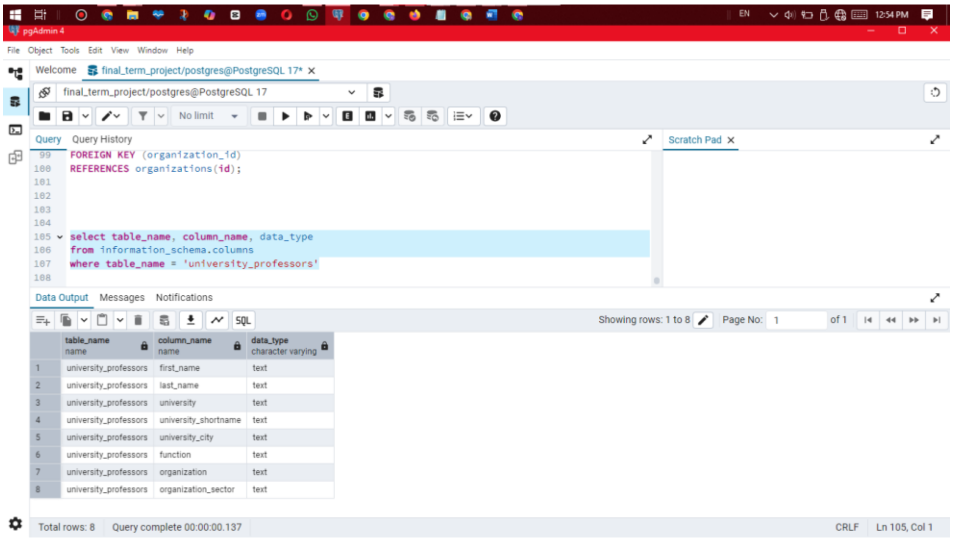
1. **Universities** – Stores university names and cities.
2. **Professors** – Stores professors' first and last names, and their university.
3. **Organizations** – Stores details about companies or research centers.
4. **Affiliations** – Connects professors with the organizations they work with.



Relationships:

- Professors are connected to one university.
- Professors can work with one or more organizations.
- Each connection between a professor and an organization is saved in the "affiliations" table.

Table Design (Schema) Screenshots


pgAdmin 4
File Object Tools Edit View Window Help
Welcome
final_term_project/postgres@PostgreSQL 17*
final_term_project/postgres@PostgreSQL 17
No limit
Query Query History
Scratch Pad
Data Output Messages Notifications
SQL

```
FOREIGN KEY (organization_id)
REFERENCES organizations(id);




select table_name, column_name, data_type
from information_schema.columns
where table_name = 'university_professors'
```

Showing rows: 1 to 8 Page No: 1 of 1

	table_name name	column_name name	data_type character varying
1	university_professors	first_name	text
2	university_professors	last_name	text
3	university_professors	university	text
4	university_professors	university_shortname	text
5	university_professors	university_city	text
6	university_professors	function	text
7	university_professors	organization	text
8	university_professors	organization_sector	text

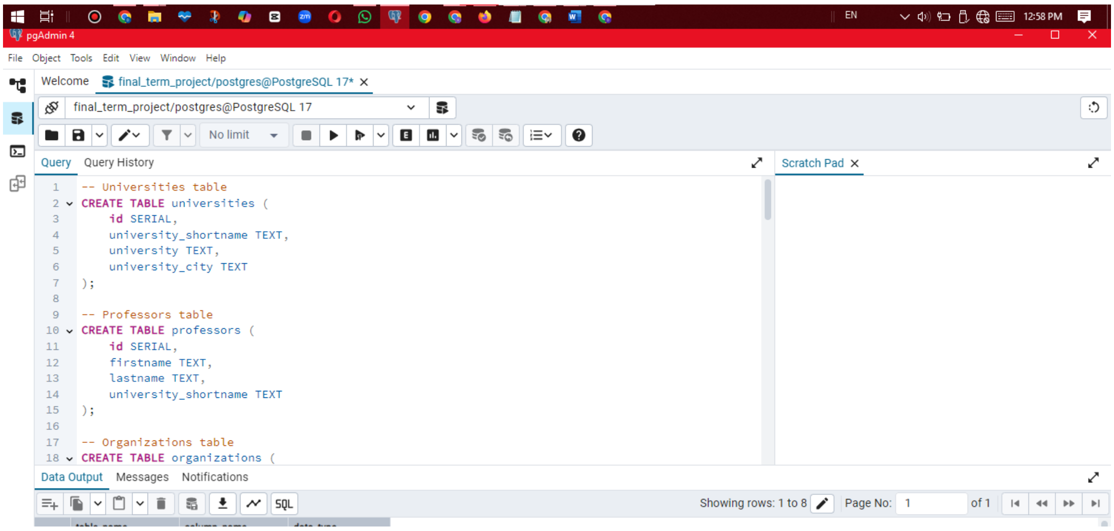
Total rows: 8 Query complete 00:00:00.137 CRLF Ln 105, Col 1

Summary of Normalized Schema:

- **professors**: Basic identity, linked to university.
- **universities**: Info about the university.
- **organizations**: External bodies professors are affiliated with.
- **affiliations**: Links a professor to an organization with a role.

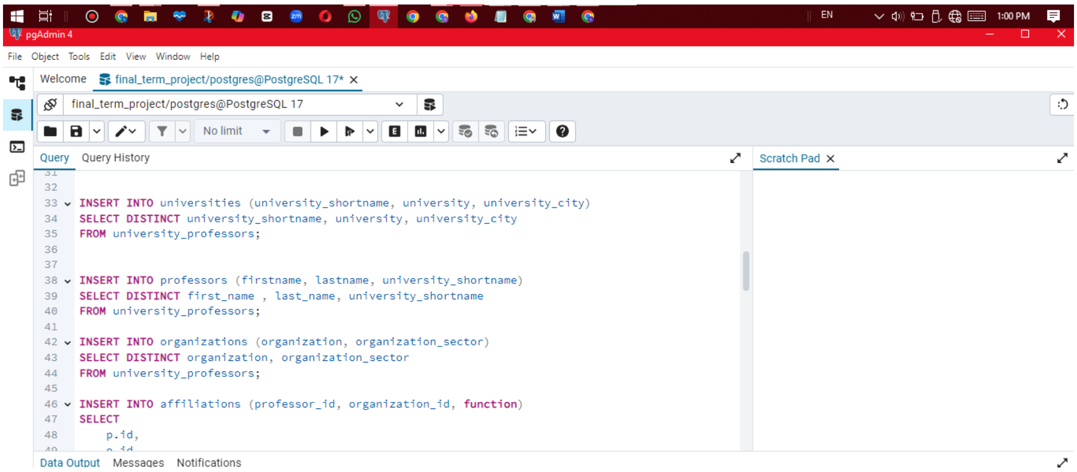
5. Frontend Screenshots

Creation of data:



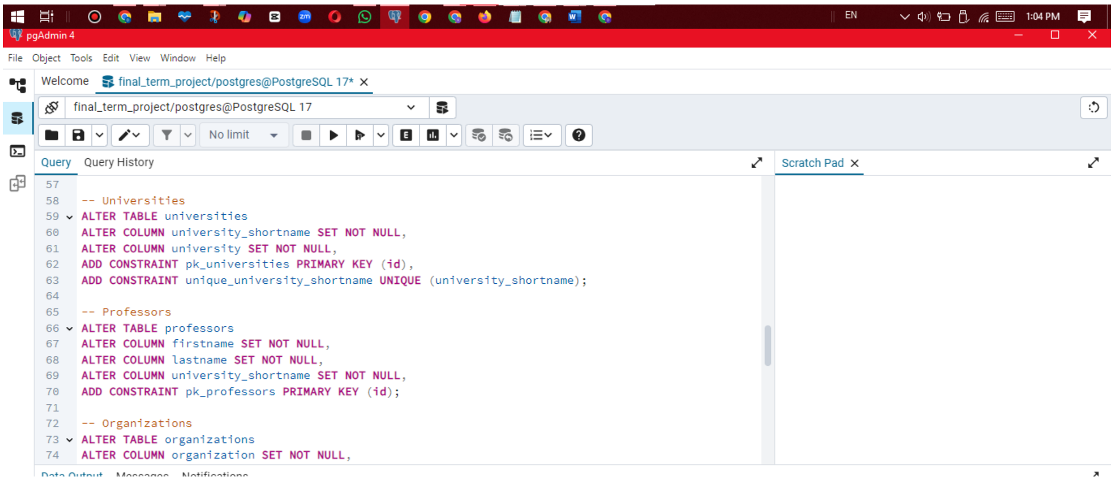
```sql
-- Universities table
CREATE TABLE universities (
    id SERIAL,
    university_shortname TEXT,
    university TEXT,
    university_city TEXT
);

-- Professors table
CREATE TABLE professors (
    id SERIAL,
    firstname TEXT,
    lastname TEXT,
    university_shortname TEXT
);

-- Organizations table
CREATE TABLE organizations (
```

Insertion of data:


pgAdmin 4
File Object Tools Edit View Window Help
Welcome
final_term_project/postgres@PostgreSQL 17*
final_term_project/postgres@PostgreSQL 17
No limit
Query
Query History
Scratch Pad
Data Output
Messages
Notifications
1:00 PM
EN

```sql


INSERT INTO universities (university_shortname, university, university_city)
SELECT DISTINCT university_shortname, university, university_city
FROM university_professors;


INSERT INTO professors (firstname, lastname, university_shortname)
SELECT DISTINCT first_name , last_name, university_shortname
FROM university_professors;

INSERT INTO organizations (organization, organization_sector)
SELECT DISTINCT organization, organization_sector
FROM university_professors;

INSERT INTO affiliations (professor_id, organization_id, function)
SELECT
    p.id,
    o.id
```

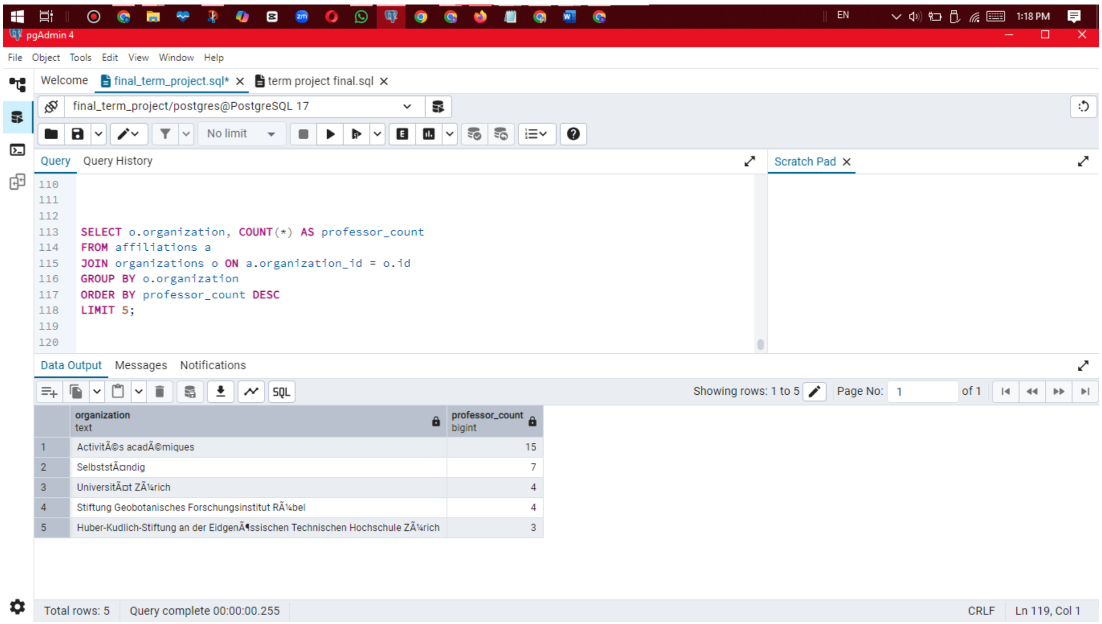
Build a Relationship:



```
-- Universities
ALTER TABLE universities
ALTER COLUMN university_shortname SET NOT NULL,
ALTER COLUMN university SET NOT NULL,
ADD CONSTRAINT pk_universities PRIMARY KEY (id),
ADD CONSTRAINT unique_university_shortname UNIQUE (university_shortname);

-- Professors
ALTER TABLE professors
ALTER COLUMN firstname SET NOT NULL,
ALTER COLUMN lastname SET NOT NULL,
ALTER COLUMN university_shortname SET NOT NULL,
ADD CONSTRAINT pk_professors PRIMARY KEY (id);

-- Organizations
ALTER TABLE organizations
ALTER COLUMN organization SET NOT NULL,
```

6. Example query:

- **Count how many professors are in each organization**



```
SELECT o.organization, COUNT(*) AS professor_count
FROM affiliations a
JOIN organizations o ON a.organization_id = o.id
GROUP BY o.organization
ORDER BY professor_count DESC
LIMIT 5;
```

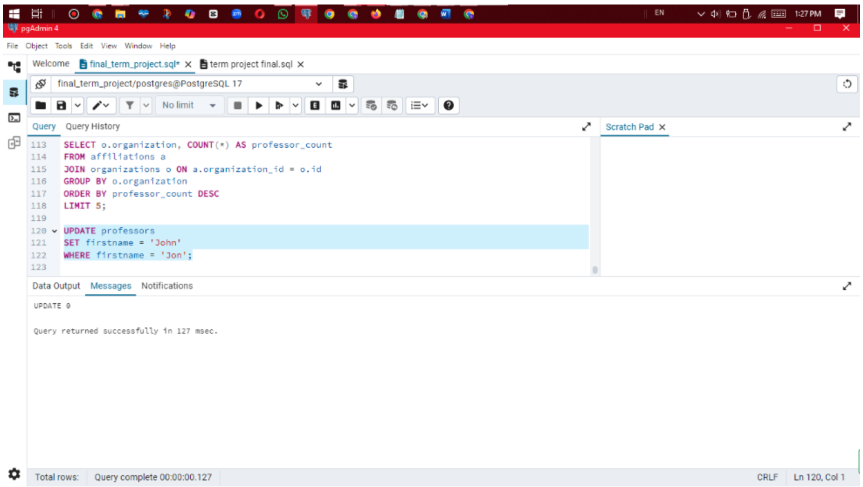
	organization text	professor_count bigint
1	ActivitÃ©s acadÃ©miques	15
2	SelbststÃ¤ndig	7
3	UniversitÃ¤t ZÃ¼rich	4
4	Stiftung Geobotanisches Forschungsinstitut RÃ¼bel	4
5	Huber-Kudlich-Stiftung an der EidgenÃ¶ssischen Technischen Hochschule ZÃ¼rich	3

Total rows: 5 Query complete 00:00:00.255

- **Update table:**

UPDATE professors
SET firstname = 'John'

WHERE firstname = 'Jon';

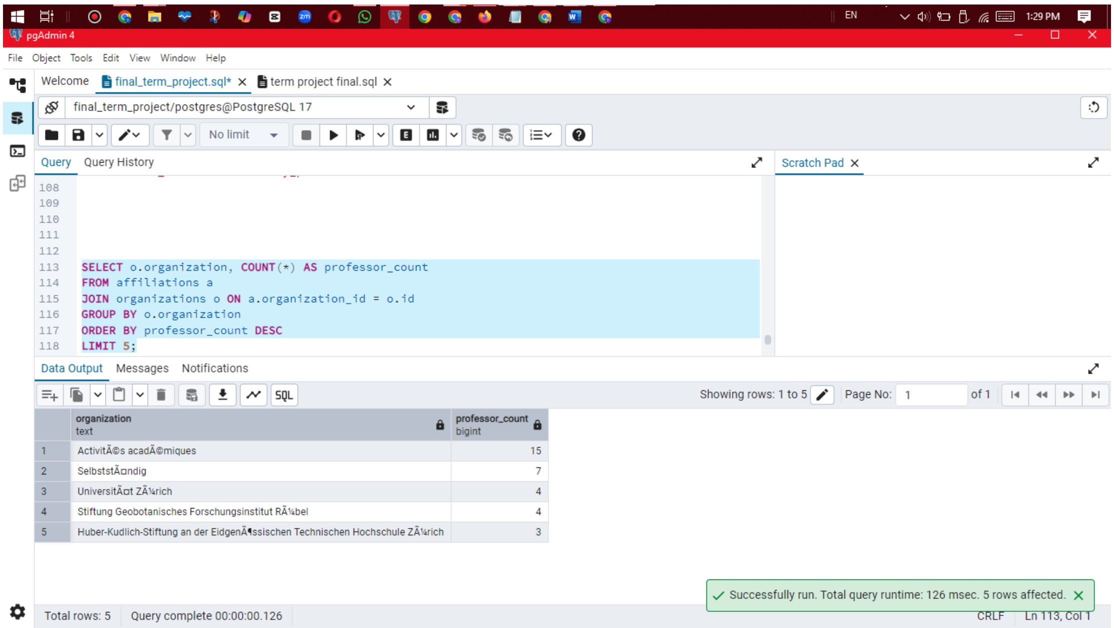


```
SELECT o.organization, COUNT(*) AS professor_count
FROM affiliations a
JOIN organizations o ON a.organization_id = o.id
GROUP BY o.organization
ORDER BY professor_count DESC
LIMIT 5;

UPDATE professors
SET firstname = 'John'
WHERE firstname = 'Jon';
```

```
UPDATE 0

Query returned successfully in 127 msec.
```

Total rows: Query complete 00:00:00.127

JOIN Example:



```
SELECT o.organization, COUNT(*) AS professor_count
FROM affiliations a
JOIN organizations o ON a.organization_id = o.id
GROUP BY o.organization
ORDER BY professor_count DESC
LIMIT 5;
```

Showing rows: 1 to 5 Page No: 1 of 1

	organization text	professor_count bigint
1	ActivitÃ©s acadÃ©miques	15
2	SelbststÃ¤ndig	7
3	UniversitÃ¤t ZÃ¼rich	4
4	Stiftung Geobotanisches Forschungsinstitut RÃ¼bel	4
5	Huber-Kudlich-Stiftung an der EidgenÃ¶ssischen Technischen Hochschule ZÃ¼rich	3

Successfully run. Total query runtime: 126 msec. 5 rows affected.

Total rows: 5 Query complete 00:00:00.126

SELECT o.organization, COUNT(*) AS professor_count
FROM affiliations a
JOIN organizations o ON a.organization_id = o.id
GROUP BY o.organization
ORDER BY professor_count DESC
LIMIT 5;

7. Referential Integrity:

We added **foreign keys** to make sure:

- Professors must belong to a valid university.
- Affiliations must link to real professors and organizations.

This keeps the data **correct** and **connected**.

8. Project Files

(Provide github link):

https://github.com/hassanraza1442/Database-final-term-project-