

**PROJECT**

**SUBMITTED BY** M Belal

**SUBMITTED TO** Prof. M. Usman Ghani

**BS CS 7M, 2nd**

A black and white text

AI-generated content may be incorrect.

\*PROJECT\*

# *1- INTRODUCTION*

* 1. **Background (REASON):**

The reason behind all this that ,many professors work with different organizations in universities 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.

**2.2 GOAL :**

The Main objectives of this project are shown and explained below;

1-To save data about universities and professors.

2-To show which professor is connected to which organization.

3-To make it easy for;

* SEARCH
* ADD
* UPDATE
* DELETE

**2.3 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 USE:**

Its uses are defines below;

• University staff to manage professors.

• Students to see where professors work or do research.

• Database admins to maintain clean and correct data.

**2.2 Entity Data Model:**

It consists of 4 main tables ;

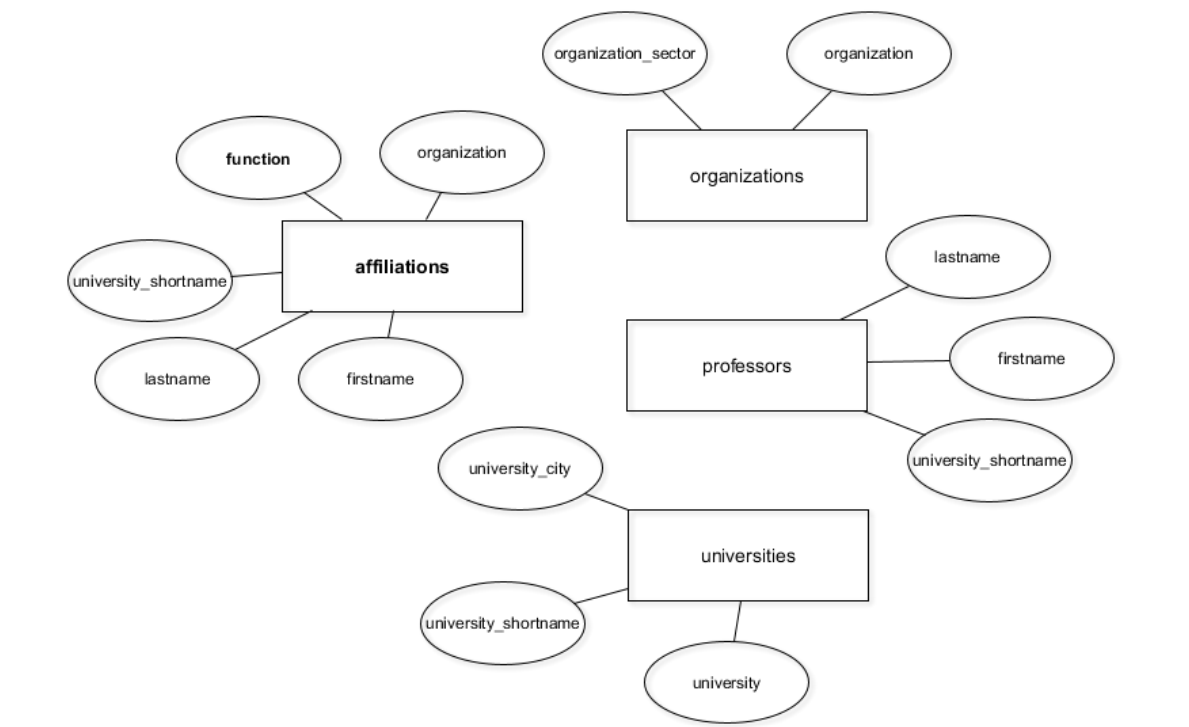
1. ***Universities –*** Stores university names and cities.

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.

It’s shown below;



**Fig;** Data Entity Model

**Relationship:**

The relationship between the upper tables of data entity model is given below;

• 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.

# *4-TABLE DESIGN*

**Schema Screenshots**

**A screenshot of a computer

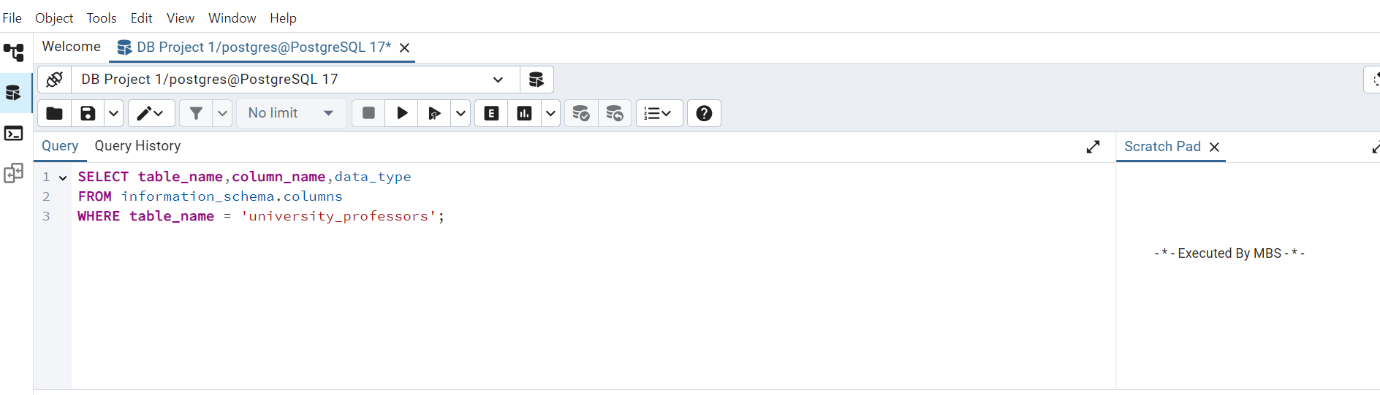
AI-generated content may be incorrect.**

**OUTPUT,**

**A screenshot of a computer

AI-generated content may be incorrect.**

Normalized schema;



**OUTPUT,**

**A white rectangular frame with a black border

AI-generated content may be incorrect.**

**QUERY SUMMARY**

The summary normalized schema is defined below;

• **professors**: Basic identity, linked to university.

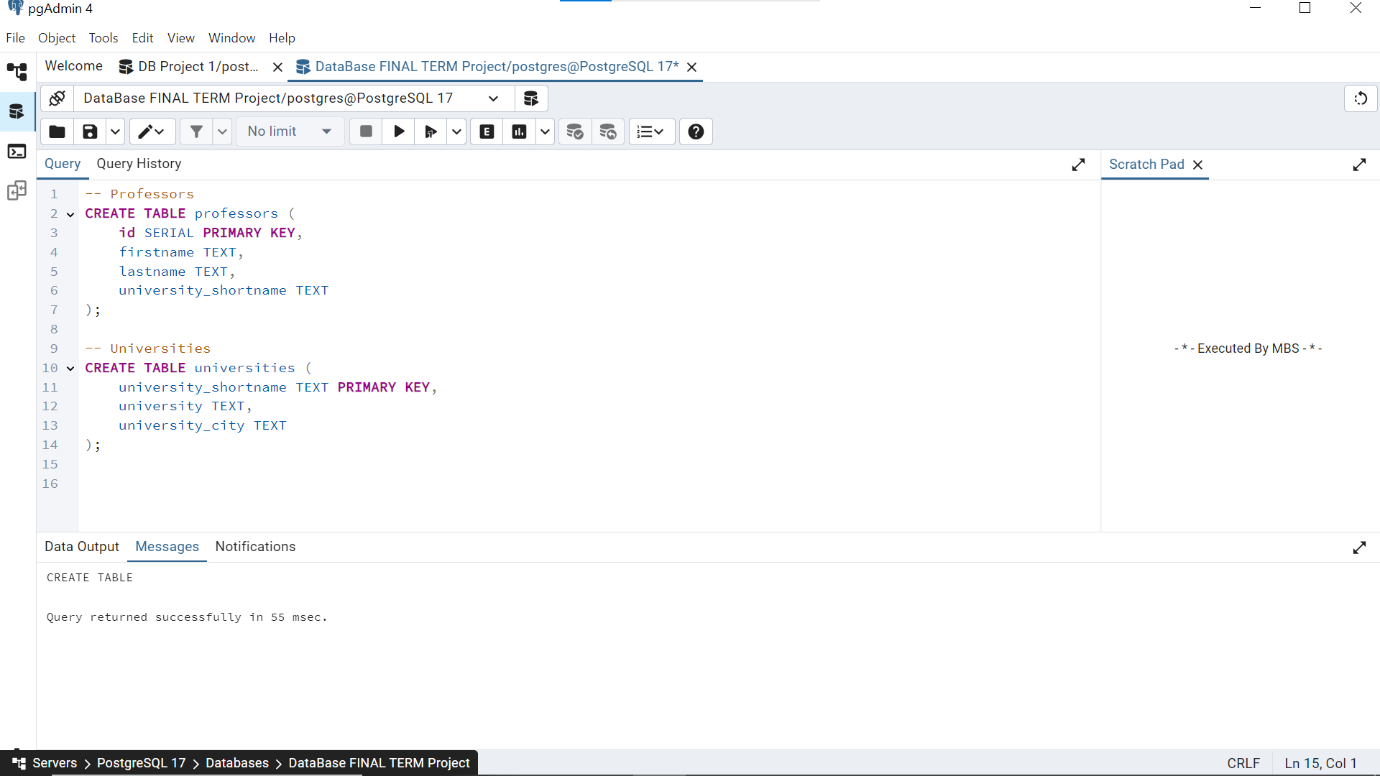
• **universities**: Info about the university.

• **organizations:** External bodies professors are affiliated with.

• **affiliations:** Links a professor to an organization with a role.

# 5-FRONT END (Screenshots)

**CREATION OF DATA**



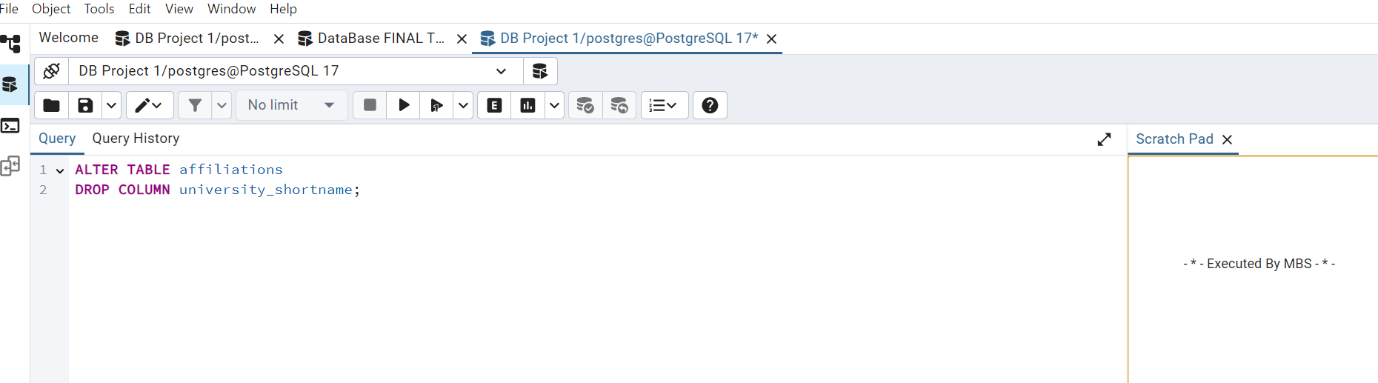
The tables universities, professors, organizations, affiliations are created !

**INSERTION OF DATA**

Insert data into four connected tables: **professors, universities**, **organizations**, and **affiliations**

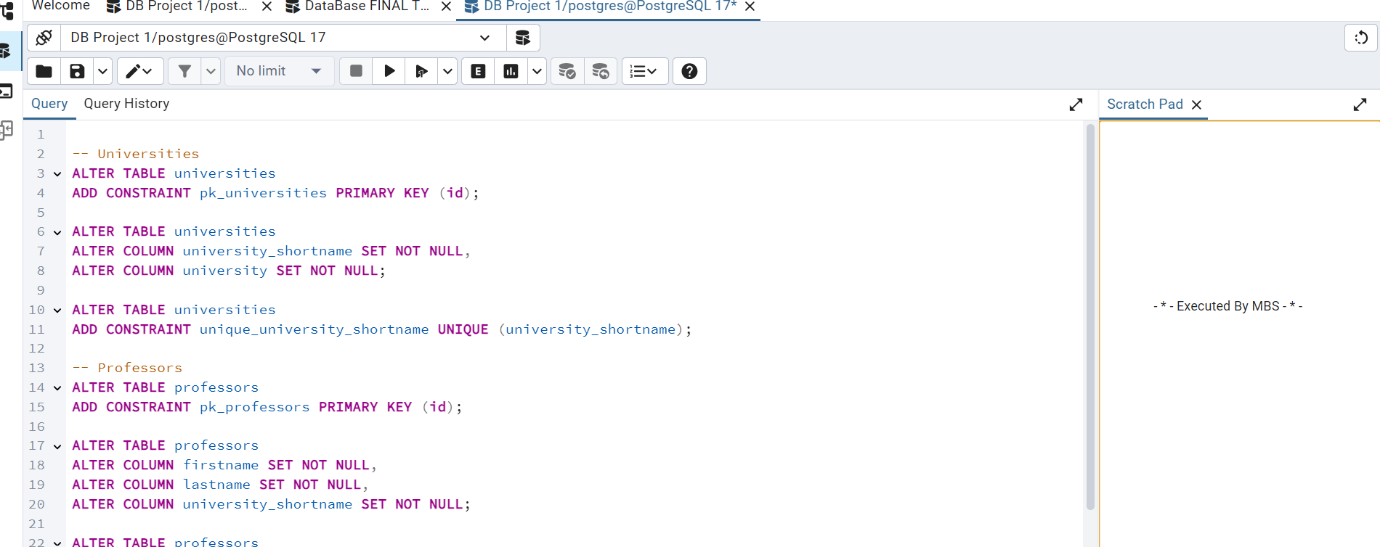


Here a professor is uniquely identified by **firstname** and **lastname** columns , so there is no need of **univerty\_shortname** column in the **affiliations** table .



We updated the professors table by setting the firstname and lastname columns to NOT NULL.

**Using ALTER (Key);**

****

This query first renames the column university\_shortname to university\_id in the professors table to better reflect its purpose. Then, it adds a foreign key constraint to ensure that the university\_id in the professors table must match an existing id in the universities table. This helps maintain a valid link between professors and their universities.

# 6-EXAMPLE QUERY

**For(Example);**

Count how many professors are in each organization;

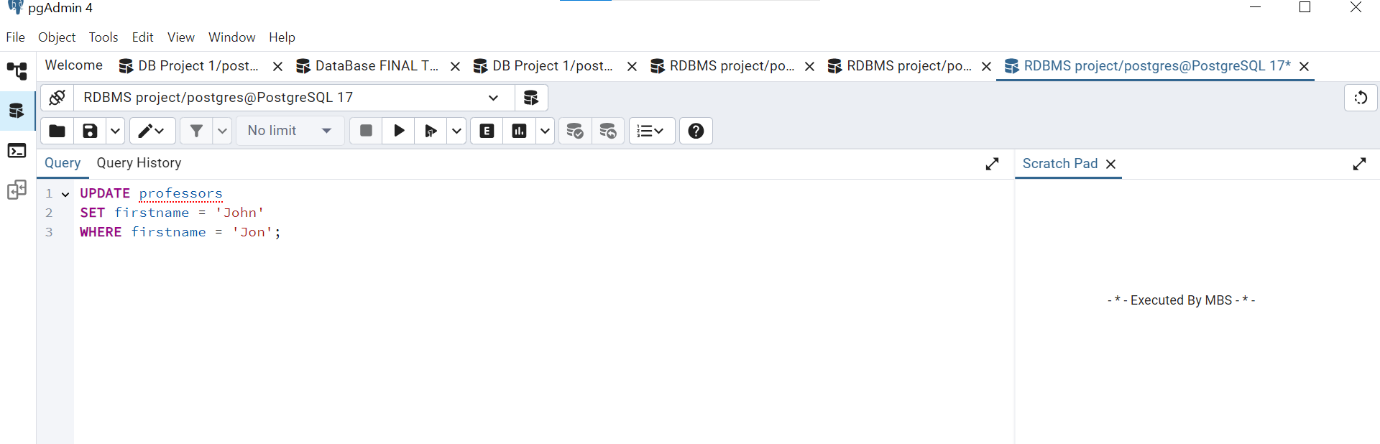
Its query id defined and explained below;

A screenshot of a computer

AI-generated content may be incorrect.

Here we counted that how many professors are there in each organization in relational databases using query defined and shown upside;

NOW, Update the exixting table using queries shown below;

****

In the upper query we updated the data of an exixtisting table names (professors), in which we rename the data inside it.

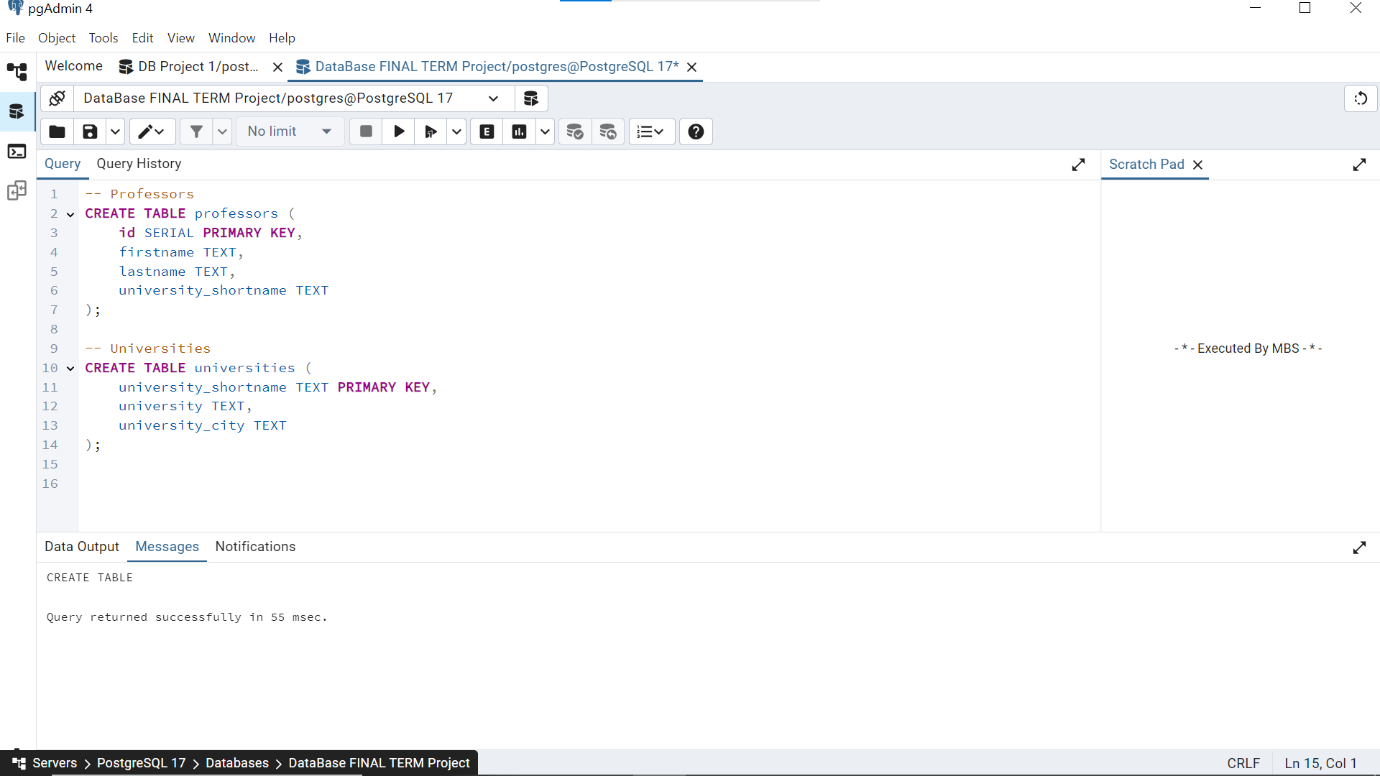
**JOIN EXAMPLE**

**A screenshot of a computer

AI-generated content may be incorrect.**

Here we are joining the data of two tables which we need for **!**

**AND..,**

****

Here we are joining the data of both of the two tables

**OUTPUT**

**A screenshot of a computer

AI-generated content may be incorrect.**

# **7-REFRENTIAL INTEGRITY**

Here we added foreign keys to make sure:

* The Professors should belong to a valid university.
* Affiliations must linked to professors and organizations.

Which makes the data connected.

# FINAL ER-DIAGRAM

A diagram of a university

AI-generated content may be incorrect.