Project: Database Systems  
Fall-2024 Department of Computer Science

**The Islamia University Bahawalpur**

**Submitted By:**

Muhammad Shahzaib Iqbal  
S23BDOCS1M01192

**Submitted To:**

Muhammad Usman Ghani



**The Islamia University Bahawalpur**

**1. Introduction:**

**1.1 Background:** In this project, you will design a relational database using a provided dataset. The task involves creating a parent table to store the main data and linking it to child tables to organize and structure the information more efficiently.

### Problem Context:

Databases are essential for storing and managing large amounts of structured data. In this project, the goal is to convert a dataset into a well-organized database. The dataset may contain multiple types of data that need to be separated into different tables, with clear relationships between them. By using parent and child tables, the database will minimize redundancy, improve data integrity, and make it easier to query and manage the data.

### Task:

**Create a parent table** that holds the main data.

**Design child tables** that store related information.

**Establish relationships** between the parent and child tables using foreign keys.

**Import the dataset** into these tables while ensuring the data is organized and consistent.

**1.2 Goal:**

1. **Create a Structured Database:**

Design and implement a parent table to store the primary data from the given dataset.

Create child tables to store related data, ensuring proper organization.

1. **Establish Relationships**:

Link the parent table to child tables using foreign keys, ensuring data integrity and clear relationships.

1. **Normalize the Database**:

Apply normalization techniques to reduce redundancy and improve efficiency in data storage and retrieval.

1. **Data Import and Organization**:

Import data from the provided dataset into the parent and child tables while maintaining consistency and structure.

1. **Enable Efficient Querying**:

Develop queries to retrieve, update, and manage data across parent and child tables, demonstrating the relational structure and integrity of the database.

**1.2 Requirements:**

1. **Database Management System (DBMS)**:

* Use a relational database platform such as MySQL, PostgreSQL, SQLite, or Oracle.

1. **Dataset Format**:

* Acceptable formats for the dataset: CSV, JSON, or other structured formats compatible with the chosen DBMS.

1. **Tools and Software**:

* Use a database IDE or tool (e.g., MySQL Workbench, pgAdmin, or DBeaver) for schema design and queries.

1. **Programming Language (Optional)**:

* If required, use a language like Python, Java, or PHP to automate data import or interact with the database.

1. **Documentation**:

* Provide clear documentation of the database schema, relationships, and key queries.

**2. Functional Description:**

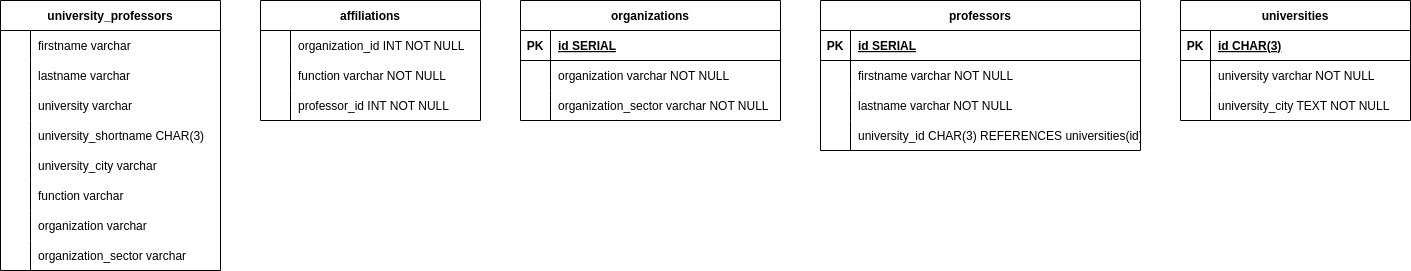
**2.1 Method of Use**

This database system is a small-scale project developed for educational purposes to demonstrate the organization and management of relational data in **PostgreSQL**.

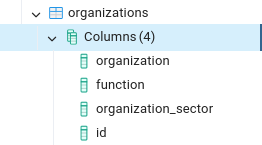
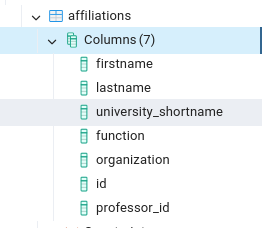
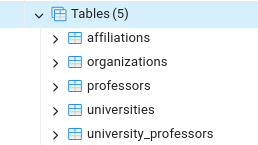
#### ****Intended Users****:

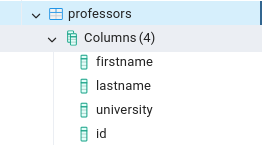
1. **Students**:
   * Use the database to learn about relational database concepts, normalization, and SQL queries.
   * Interact with the database to practice CRUD (Create, Read, Update, Delete) operations.
2. **Instructors**:
   * Evaluate the database structure, design, and functionality as part of the project assessment.
   * Provide feedback and suggestions for improvement.
3. **Developers (Yourself)**:
   * Gain hands-on experience in designing, implementing, and optimizing databases using PostgreSQL.
   * Explore how to import datasets, establish relationships, and write queries to retrieve and manage data effectively.

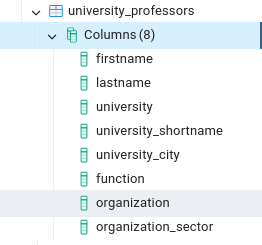
**3. Entity Data Model:**

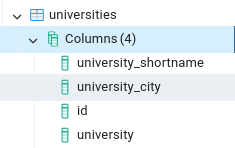
****

**4. Table Design (Schema) Screenshots:**

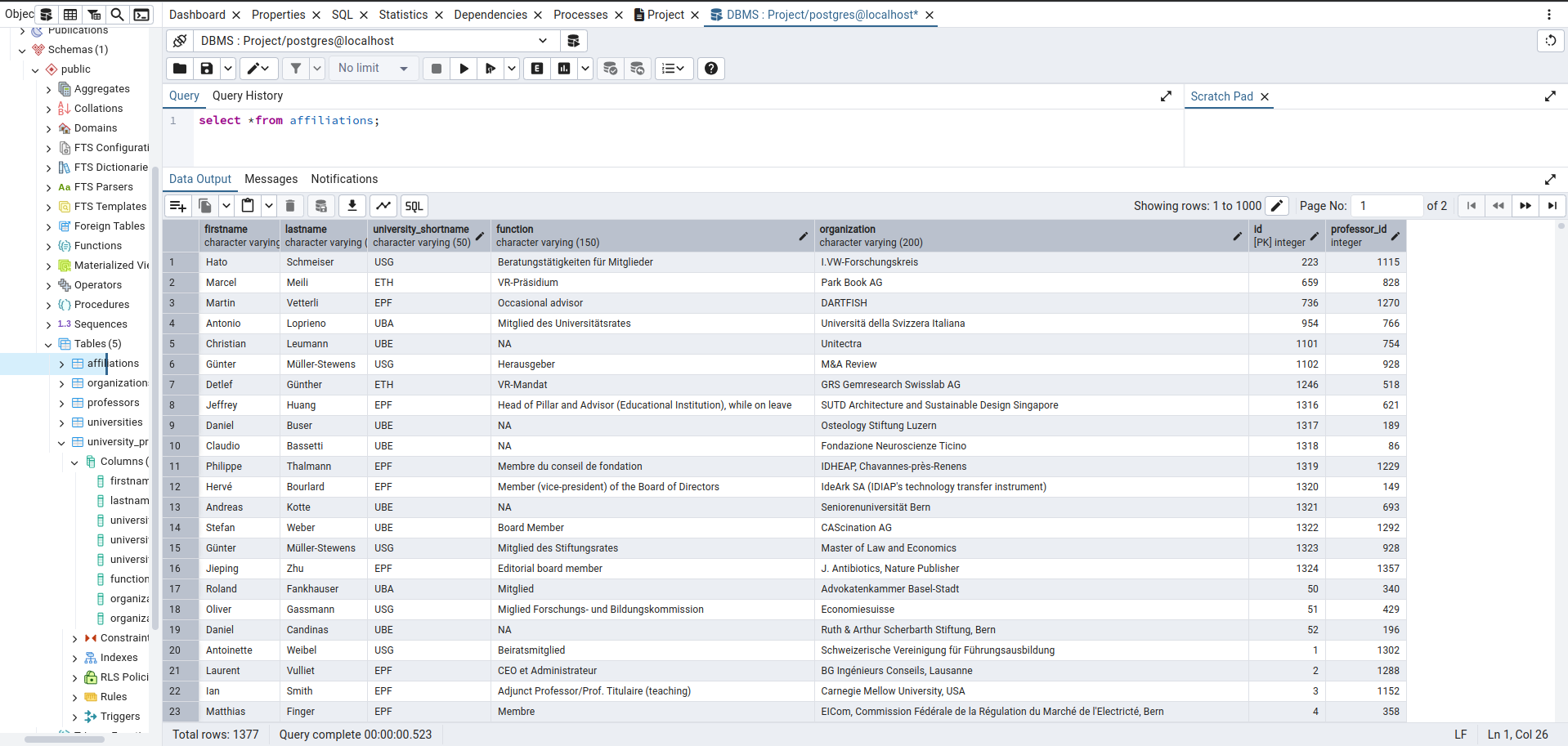




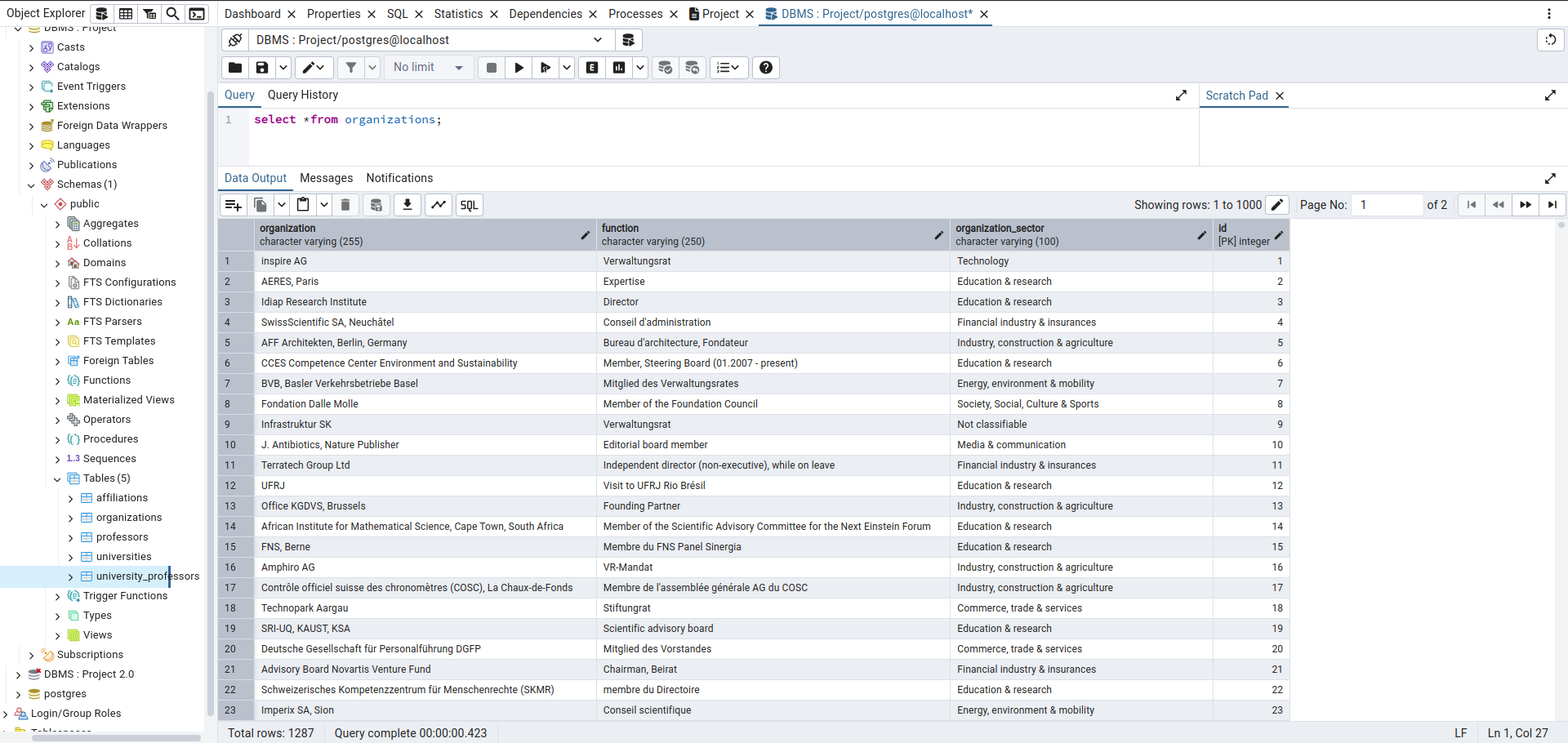




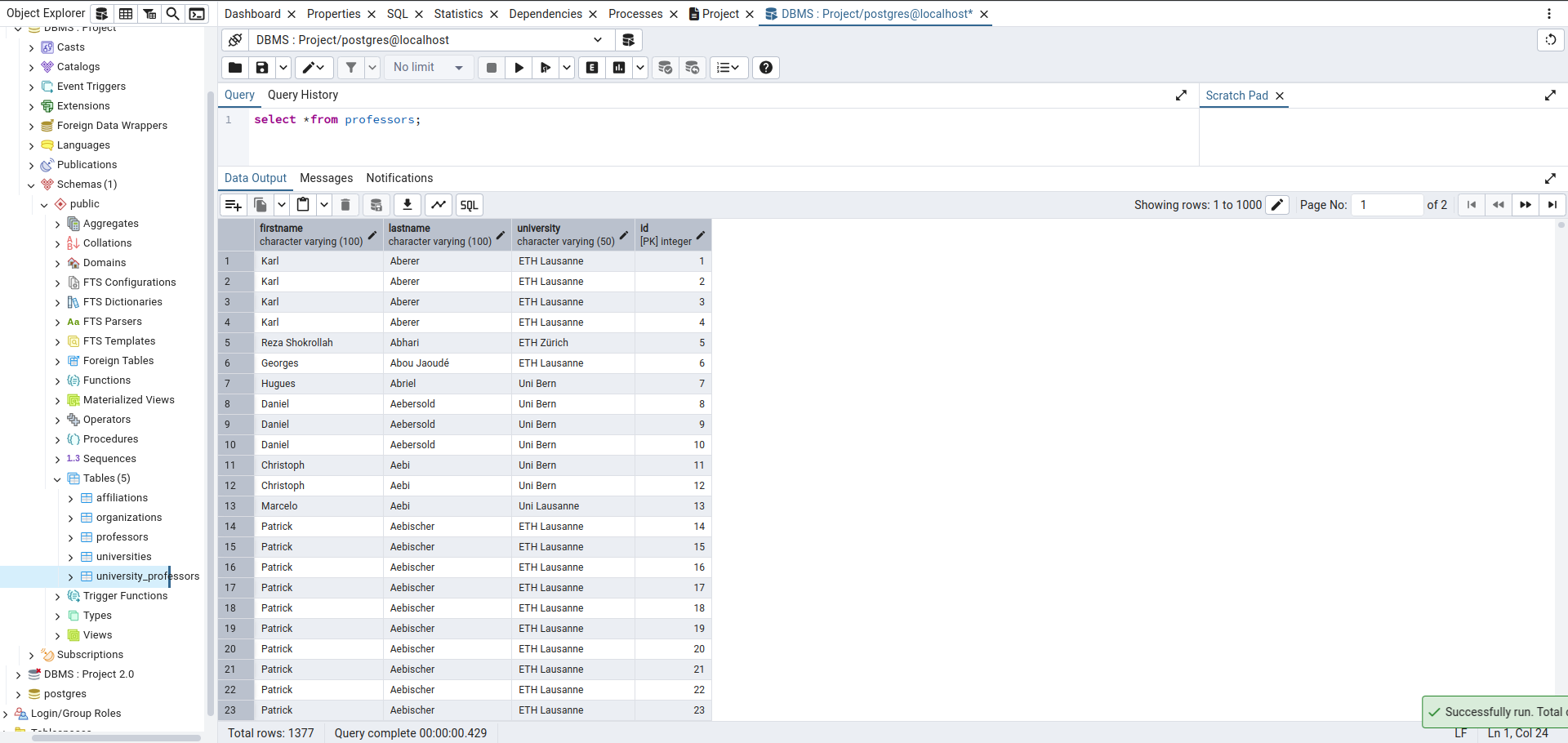
**5. Frontend Screenshots:  
  
 5.1: Affiliations :**



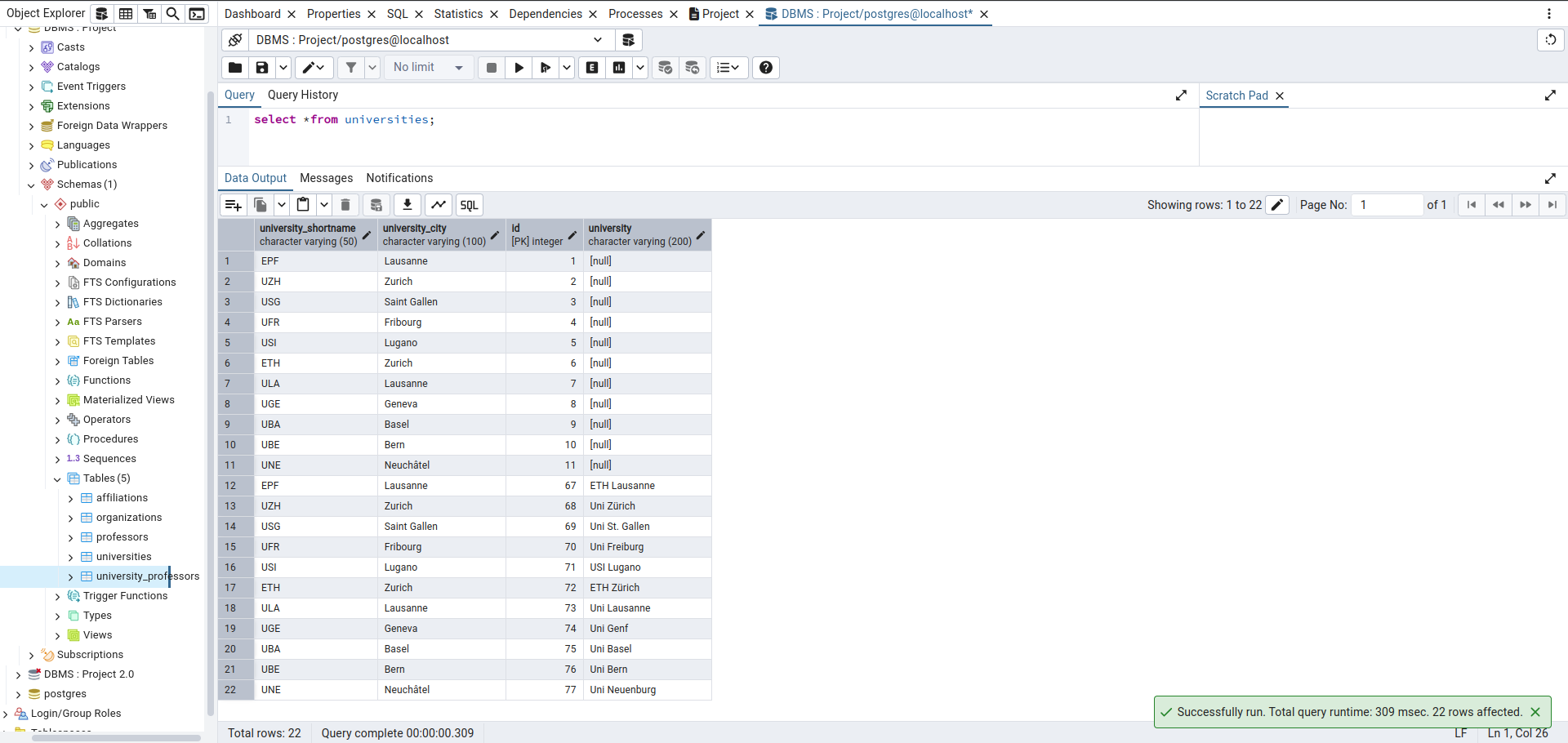
**5.2: Organizations :**



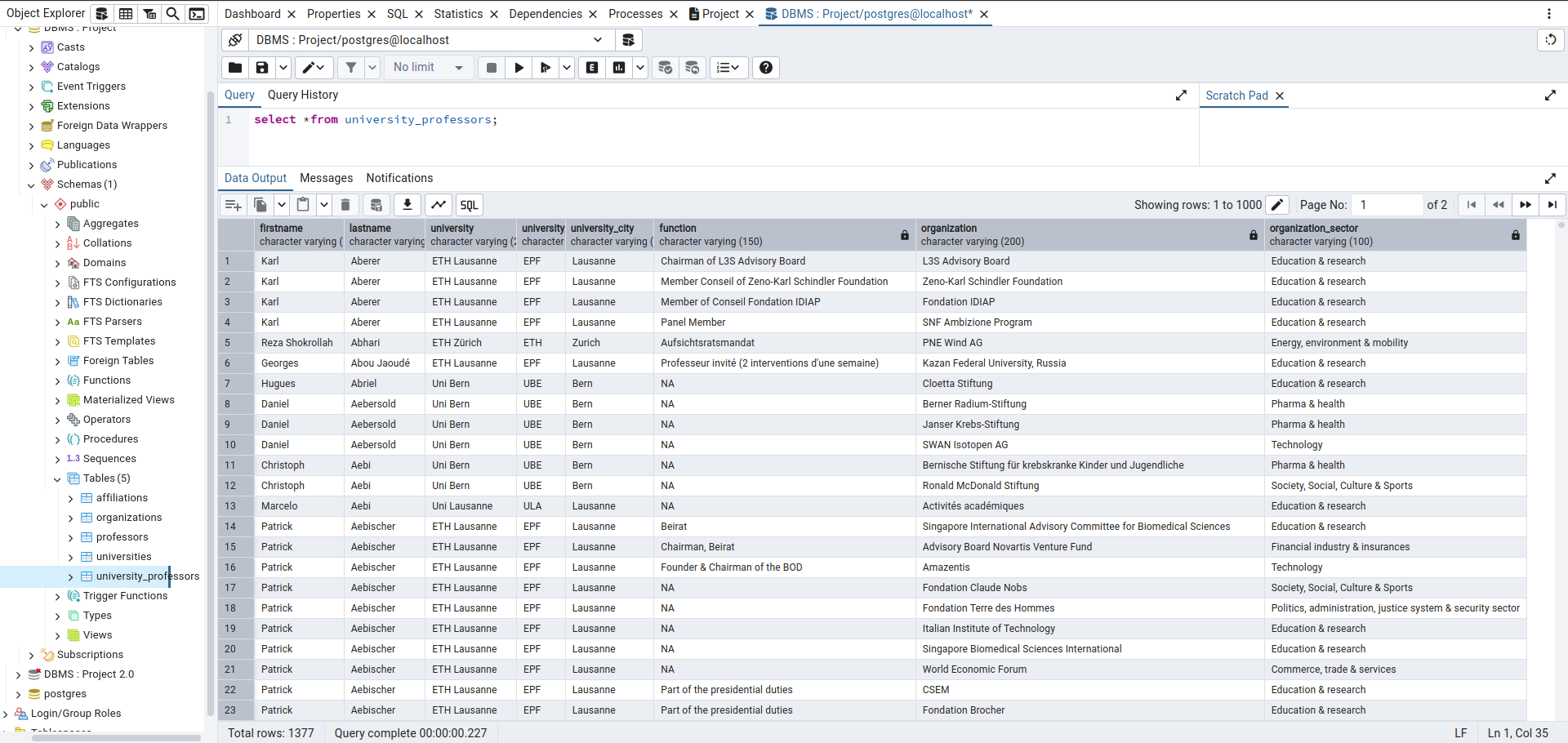
**5.3: Professors :**



**5.4: Universities :**



**5.5: University\_professors:**



**6. Examples :  
 -- CREATE TABLE :**

CREATE TABLE universities (

university\_id INT PRIMARY KEY,

university\_name VARCHAR(255)

);

CREATE TABLE professors (

id INT PRIMARY KEY,

firstname VARCHAR(255),

lastname VARCHAR(255),

university\_id INT,

FOREIGN KEY (university\_id) REFERENCES universities(university\_id)

);  
 **-- INSERT :**INSERT INTO universities (university\_id, university\_name)

VALUES

(1, 'Harvard University'),

(2, 'MIT'),

(3, 'Stanford University');

INSERT INTO professors (id, firstname, lastname, university\_id)

VALUES

(101, 'John', 'Doe', 1),

(102, 'Jane', 'Smith', 2),

(103, 'Emily', 'White', 3); **-- UPDATE :**UPDATE professors

SET lastname = 'Brown'

WHERE id = 101; **-- DELETE :**DELETE FROM professors

WHERE id = 103;

**-- JOINS :**SELECT professors.firstname, professors.lastname, universities.university\_name

FROM professors

INNER JOIN universities ON professors.university\_id = universities.university\_id; **-- Referential Integrity :**DELETE FROM universities

WHERE university\_id = 1;

**--------------------------------------------------------------------------**