Project Report :

Background:

The project focused on improving the structure of a database initially stored in a single table, **"university\_professor"**, which included professors' details, universities, organizations, and affiliations. This single table caused redundancy, inefficiency, and difficulty in managing data. To solve this, the data was split into four related tables: **Professors** (personal details), **Universities** (university info), **Organizations** (organization details), and **Affiliations** (linking professors to organizations with their roles). This approach reduced redundancy, improved data consistency, made queries easier, and ensured the database could handle growth efficiently.

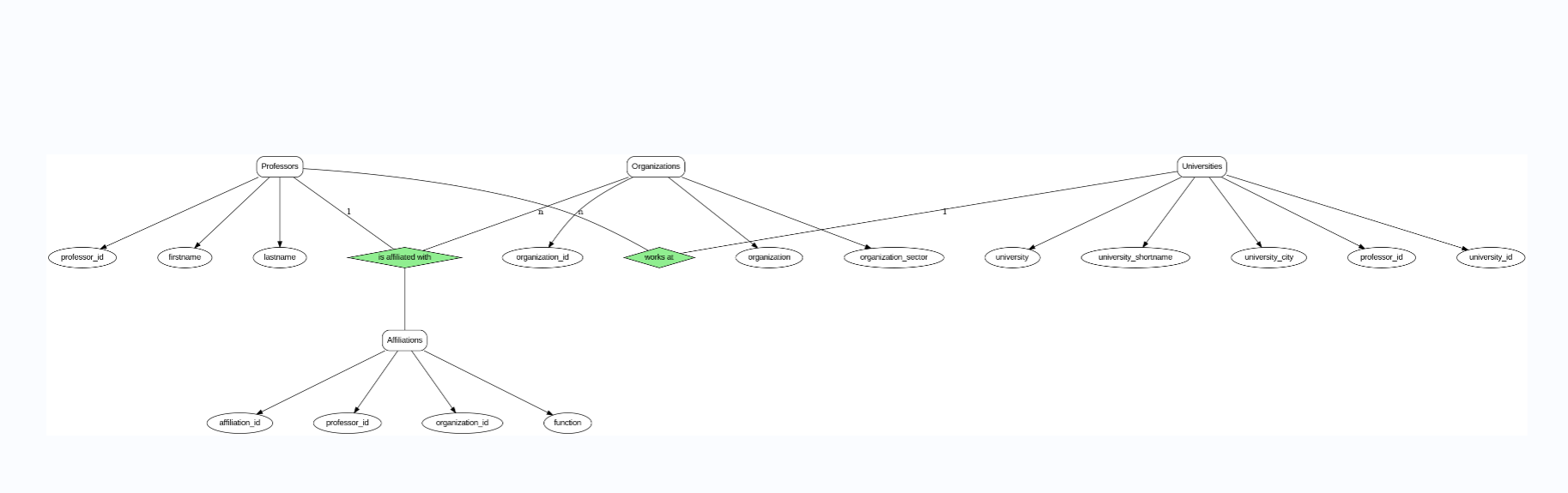
Goal:

The main goal of this project is to analyze, manipulate, and efficiently retrieve data by organizing it into a well-structured database.

Method of use :

This system is for administrators, university staff, professors, and organizations to manage and view relevant data.

E-R diagram:



University\_id

**Table Design (Schema):**

select \*

from university\_professor;

CREATE TABLE professors (

professor\_id SERIAL PRIMARY KEY ,

firstname VARCHAR(255),

lastname VARCHAR(255)

);

CREATE TABLE universities (

university\_id SERIAL PRIMARY KEY,

university VARCHAR(255),

university\_shortname VARCHAR(50),

university\_city VARCHAR(255),

professor\_id INT REFERENCES professors( professor\_id)

);

CREATE TABLE organizations (

organization\_id SERIAL PRIMARY KEY,

organization VARCHAR(255),

organization\_sector VARCHAR(255)

);

CREATE TABLE affiliations (

affiliation\_id SERIAL PRIMARY KEY,

professor\_id INT REFERENCES professors (professor\_id),

organization\_id INT REFERENCES organizations(organization\_id),

function VARCHAR (255)

);

INSERT INTO professors ( firstname, lastname )

SELECT DISTINCT firstname, lastname

FROM university\_professor;

INSERT INTO universities ( university , university\_shortname, university\_city )

SELECT DISTINCT university, university\_shortname, university\_city

FROM university\_professor;

INSERT INTO organizations (organization, organization\_sector)

SELECT DISTINCT organization, organization\_sector

FROM university\_professor;

INSERT INTO affiliations (professor\_id, organization\_id, function)

SELECT

p.professor\_id,

o.organization\_id,

mt.function

FROM university\_professor mt

JOIN professors p

ON mt.firstname = p.firstname AND mt.lastname = p.lastname

JOIN organizations o

ON mt.organization = o.organization AND mt.organization\_sector = o.organization\_sector;

ALTER TABLE university\_professor

RENAME TO university\_professors;

ALTER TABLE universities

DROP CONSTRAINT universities\_professor\_id\_fkey;

ALTER TABLE universities

DROP COLUMN professor\_id ;

ALTER TABLE professors

ADD COLUMN university\_id INTEGER;

ALTER TABLE professors

ADD CONSTRAINT fk\_university FOREIGN KEY (university\_id) REFERENCES universities (university\_id);

ALTER TABLE professors

ALTER COLUMN firstname SET NOT NULL,

ALTER COLUMN lastname SET NOT NULL ;

ALTER TABLE universities

ALTER COLUMN university SET NOT NULL,

ALTER COLUMN university\_shortname SET NOT NULL,

ALTER COLUMN university\_city SET NOT NULL;

ALTER TABLE organizations

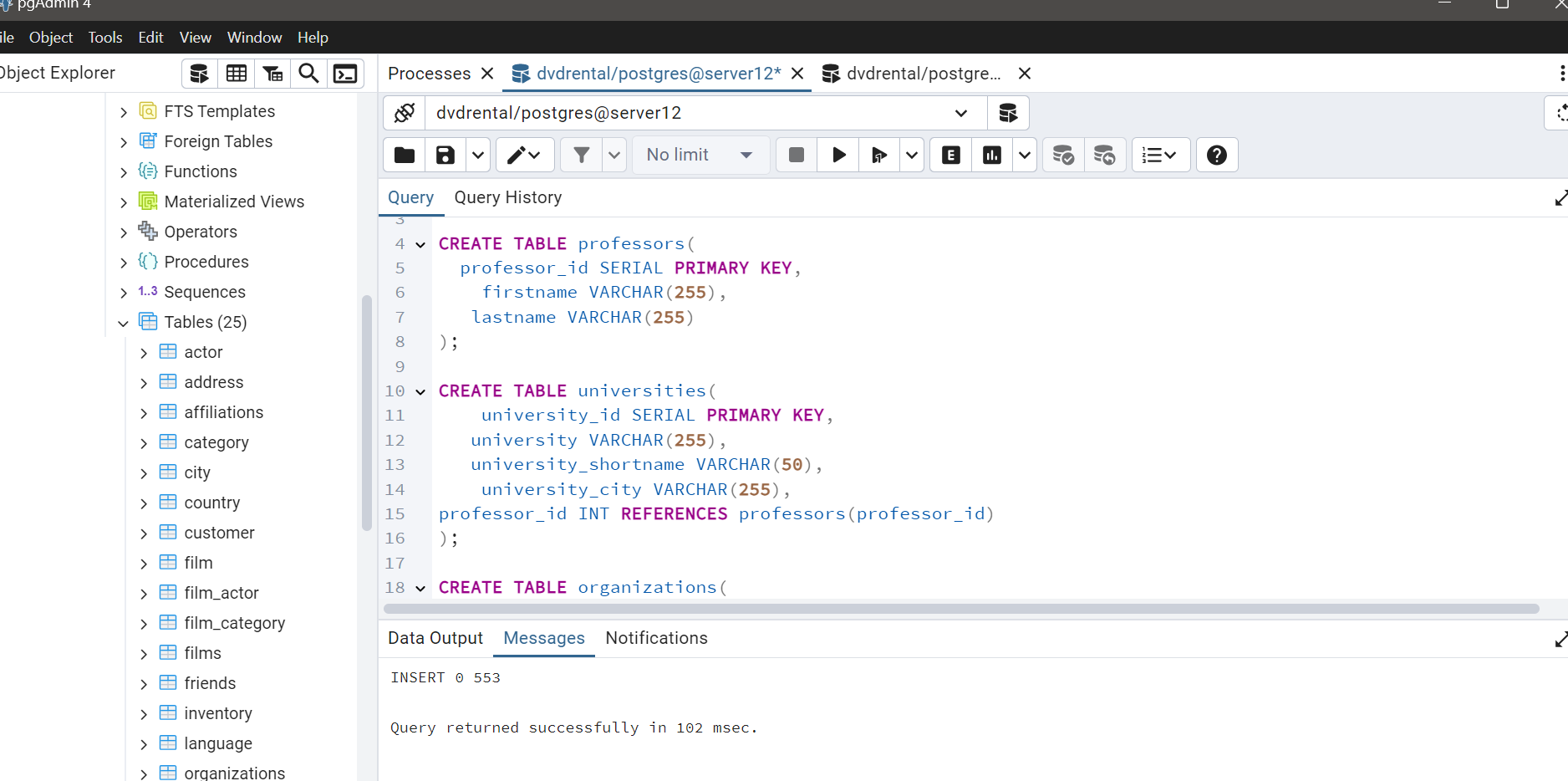
ALTER COLUMN organization SET NOT NULL,

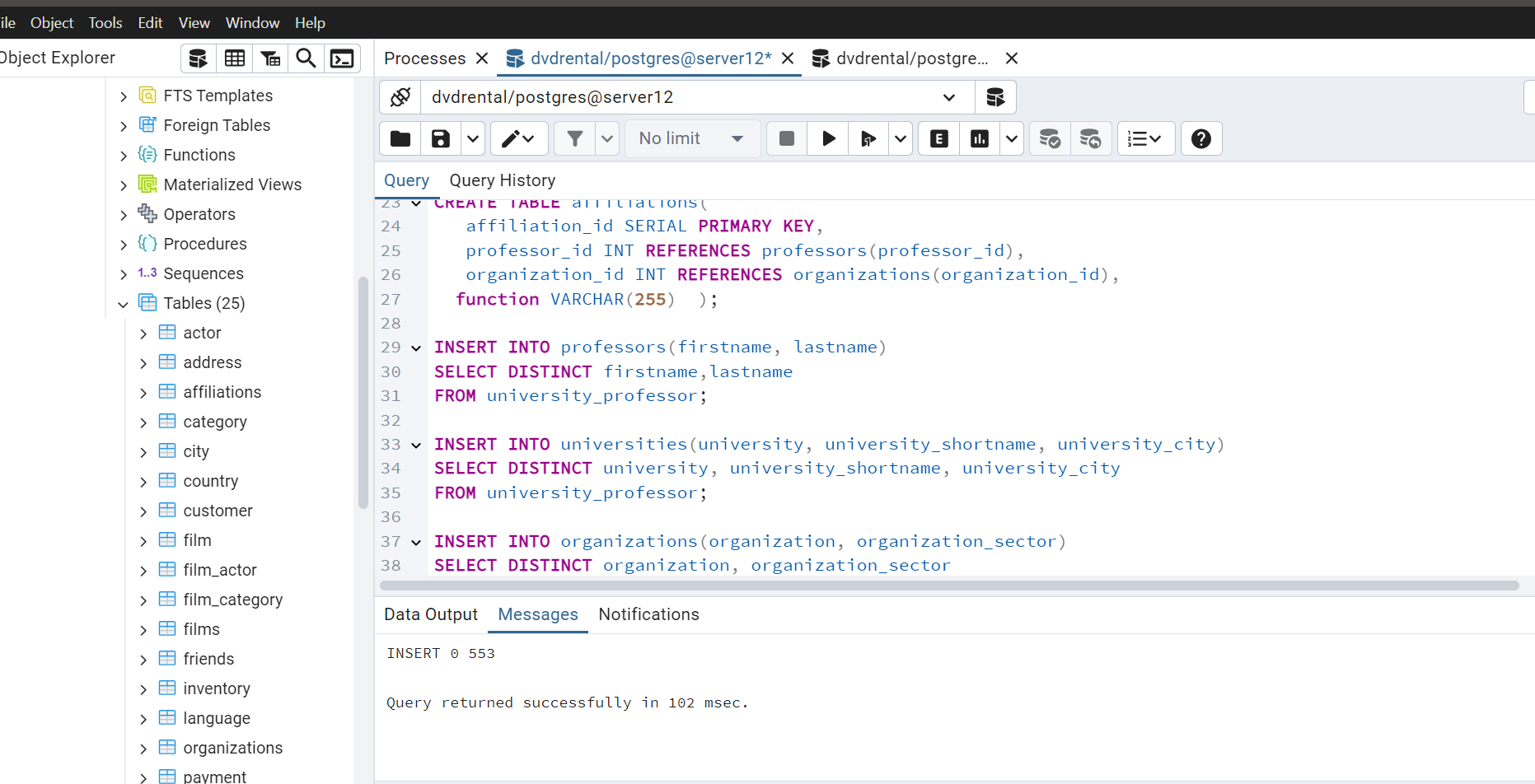
ALTER COLUMN organization\_sector SET NOT NULL ;

ALTER TABLE affiliations

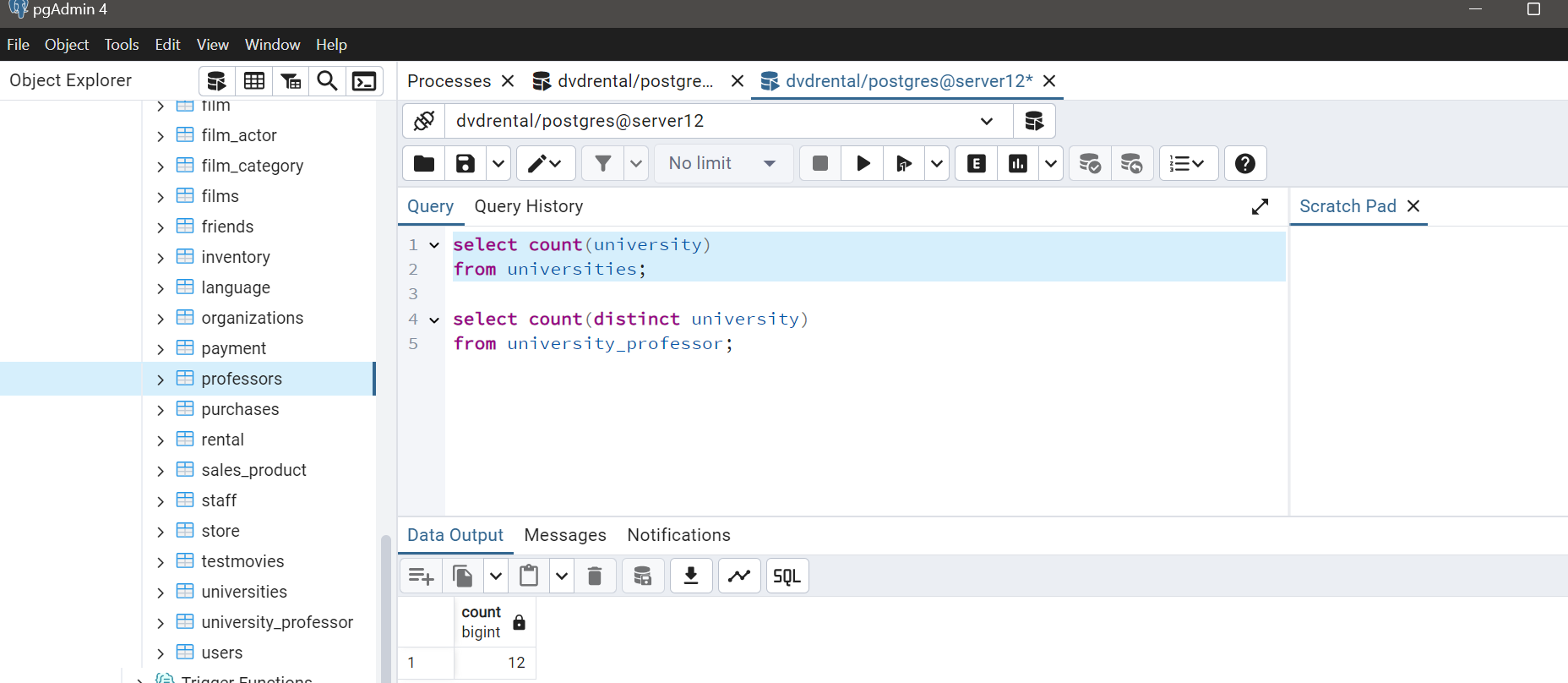
ALTER COLUMN professor\_id SET NOT NULL,

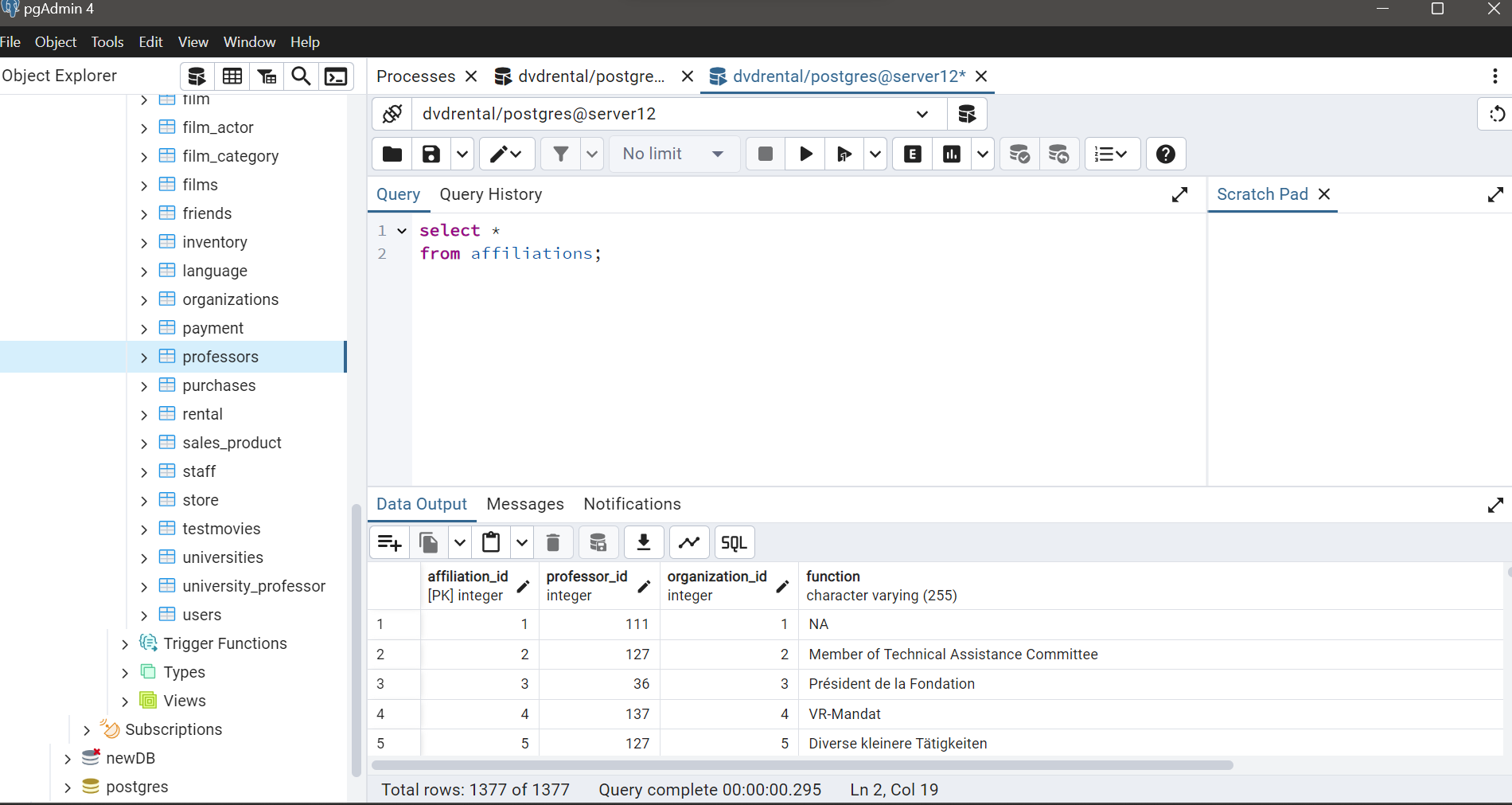
ALTER COLUMN organization\_id SET NOT NULL ;

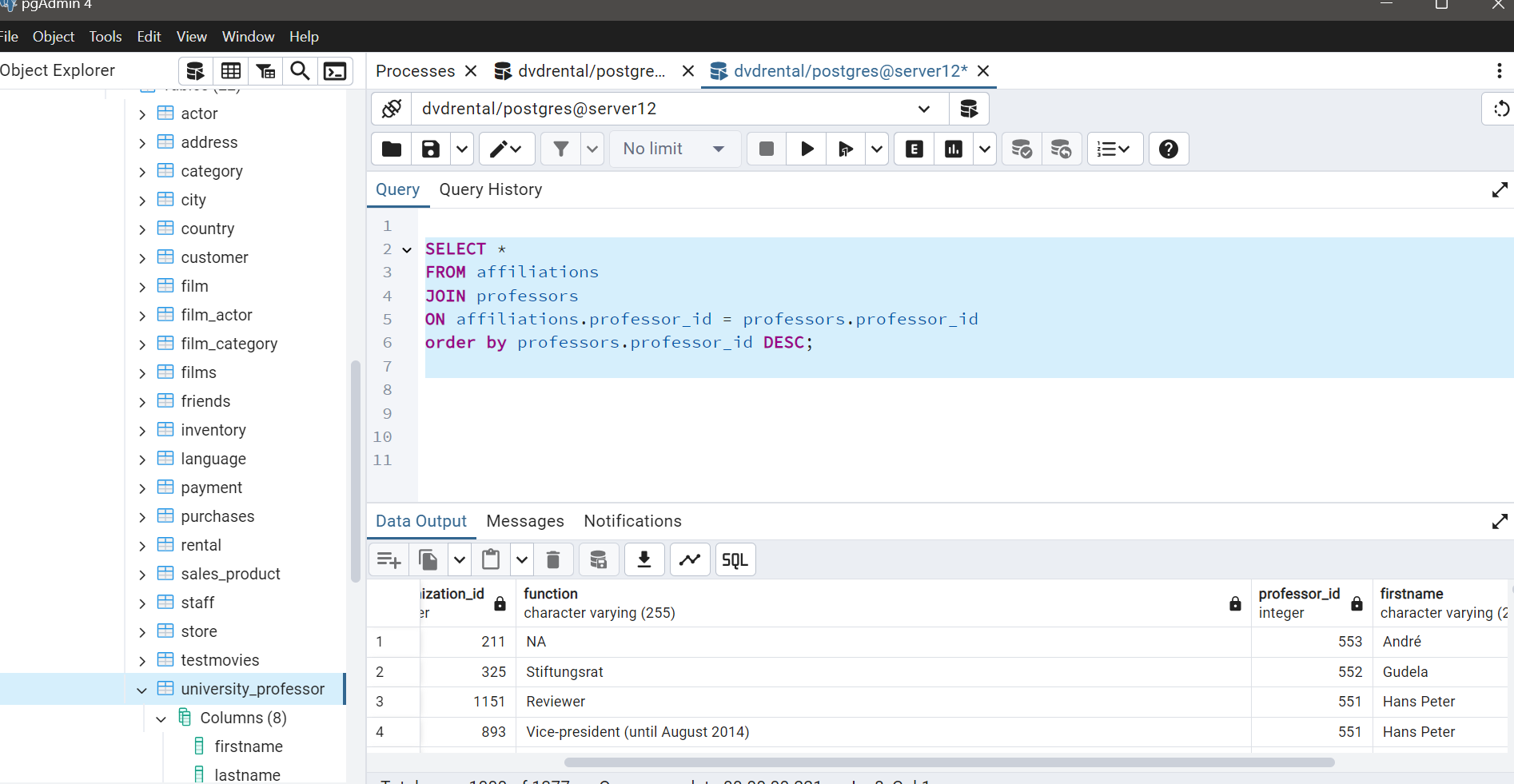


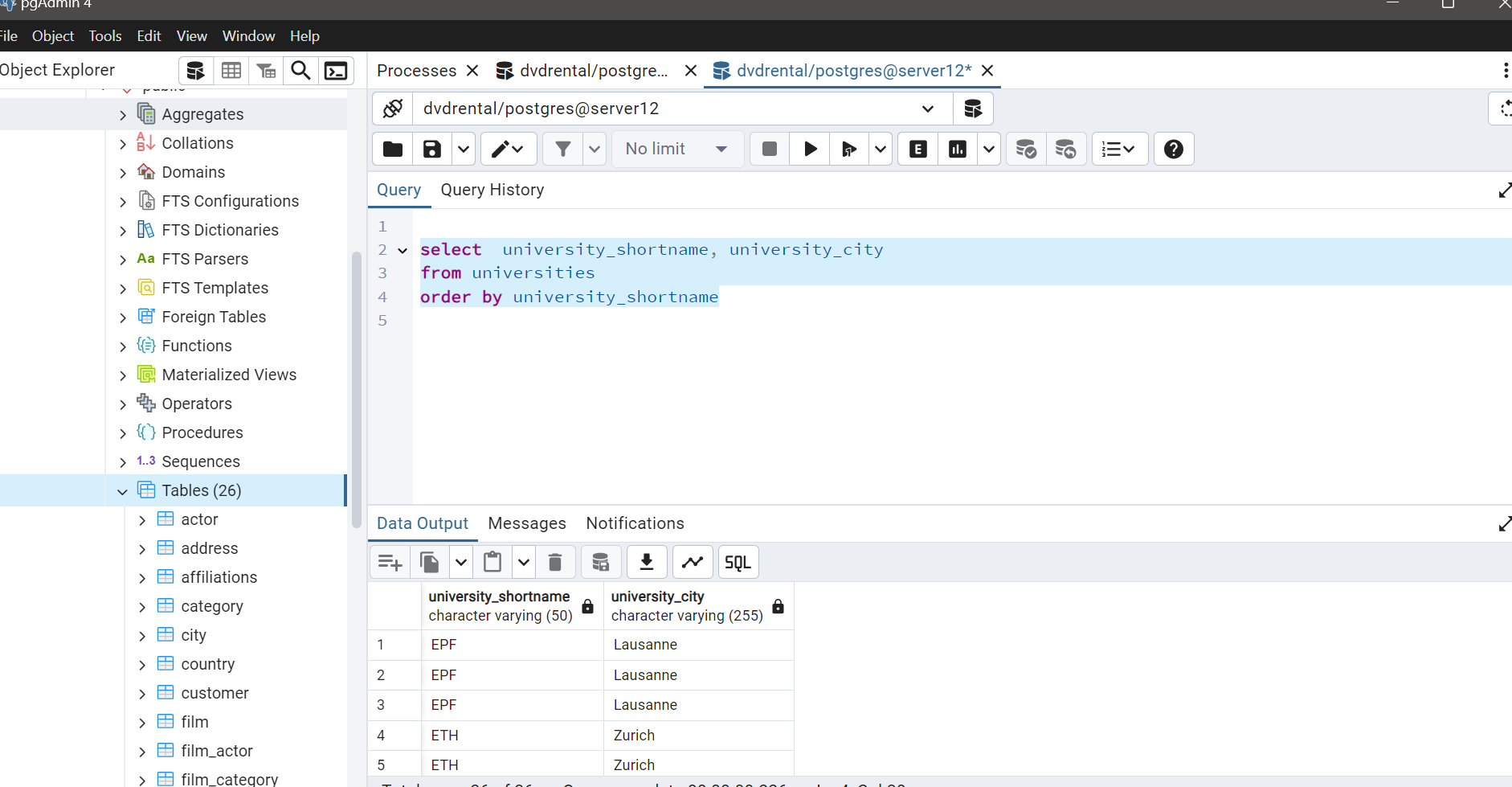


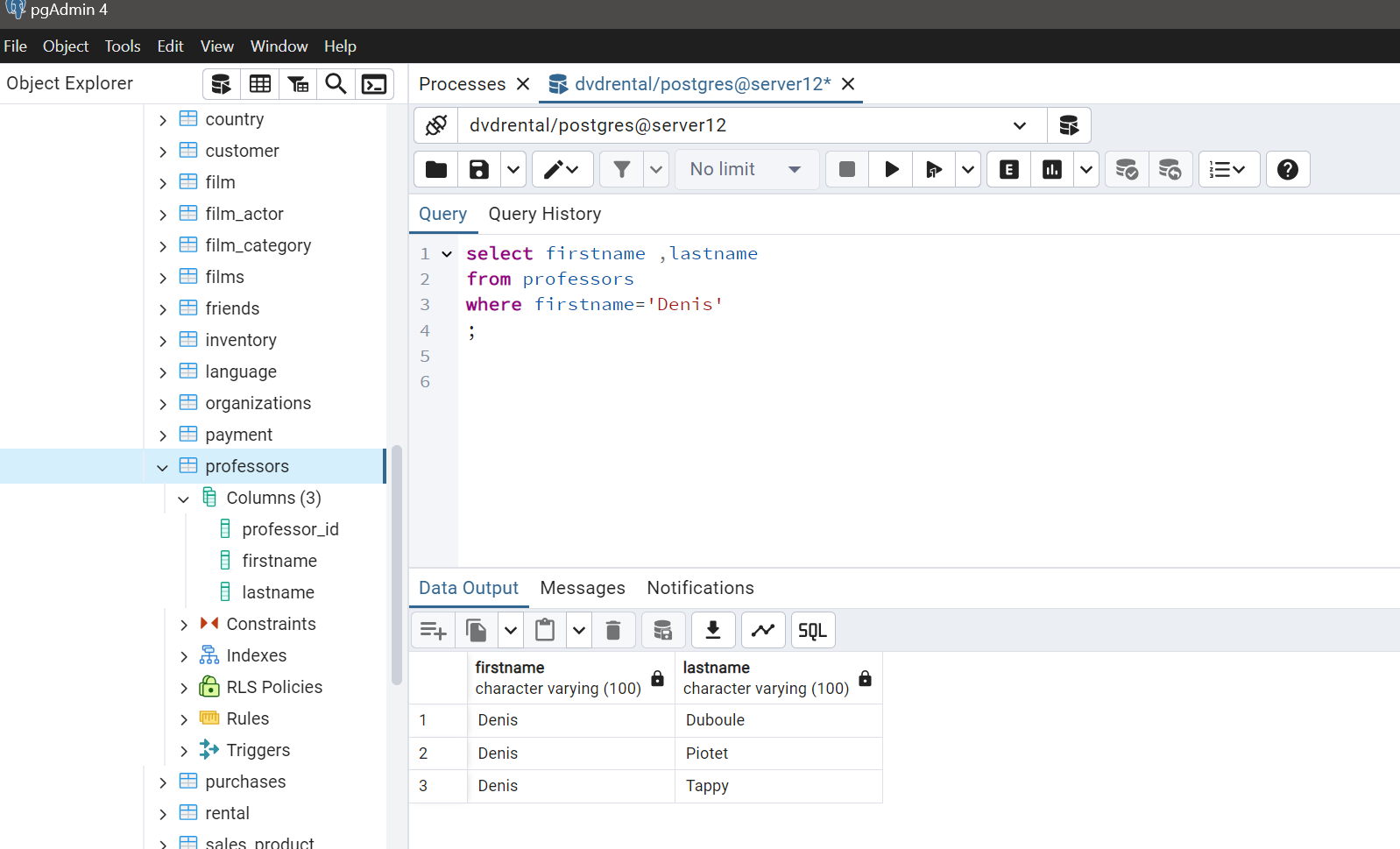
**Example Queries and the outputs:**

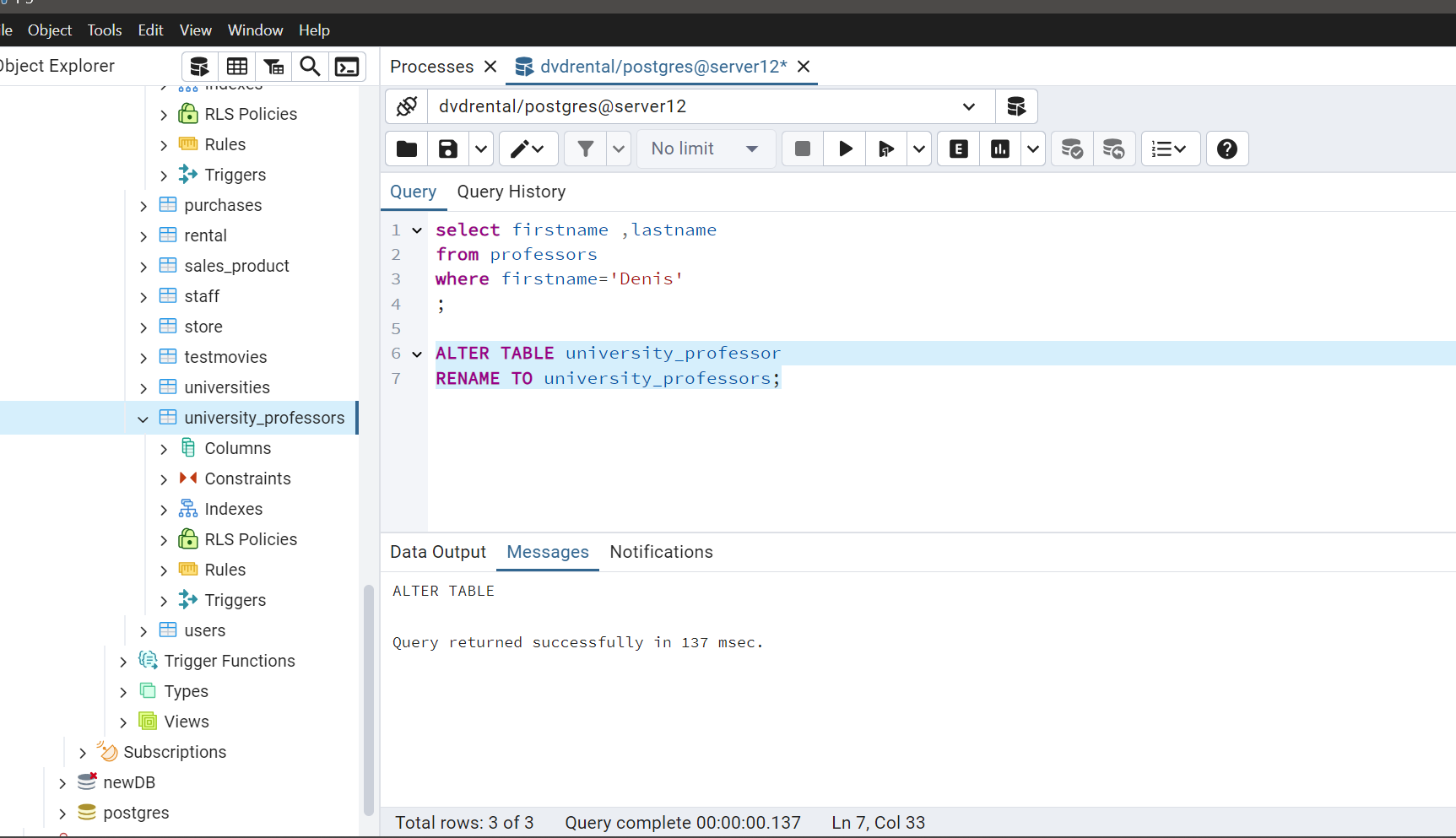


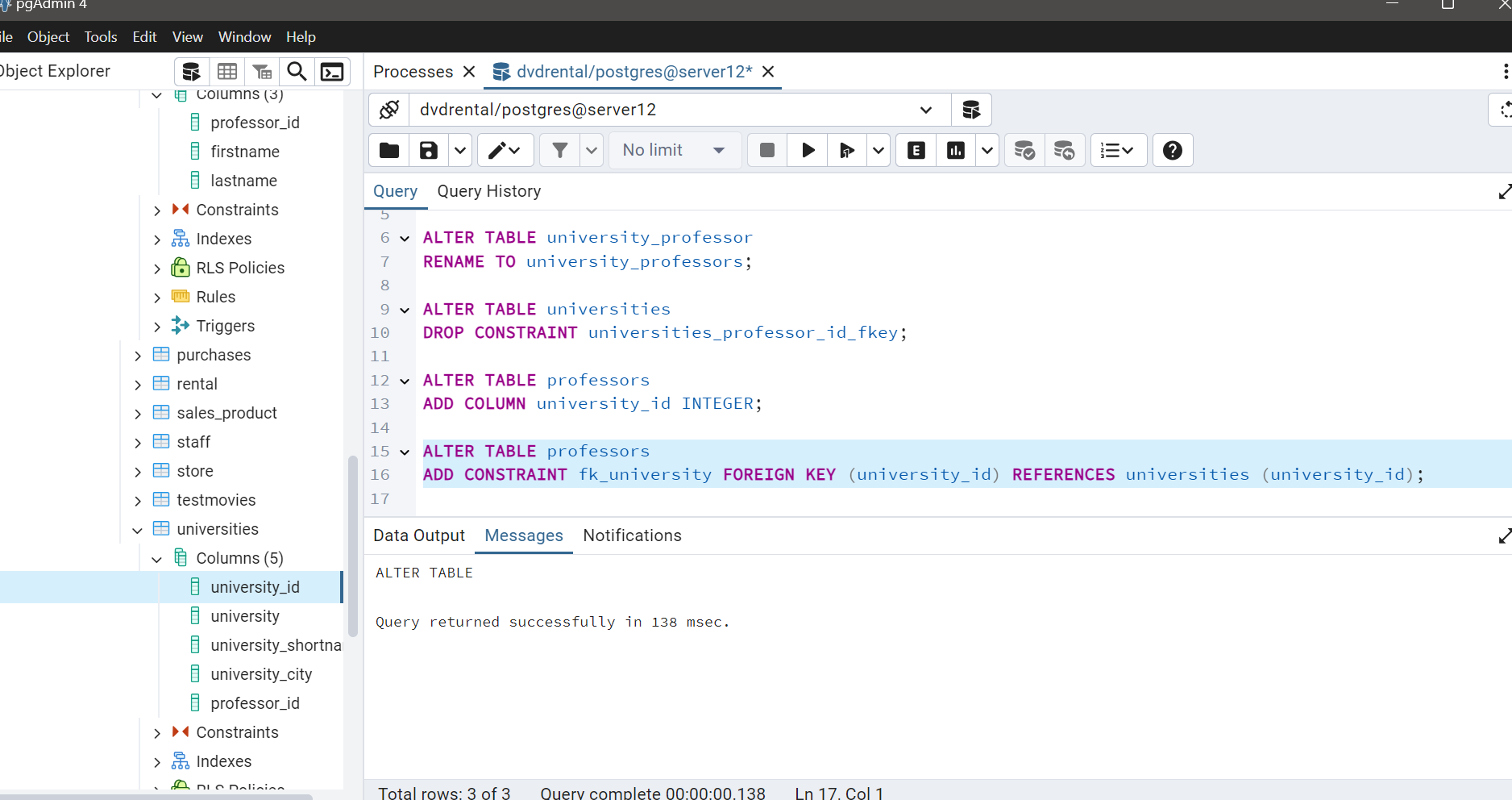


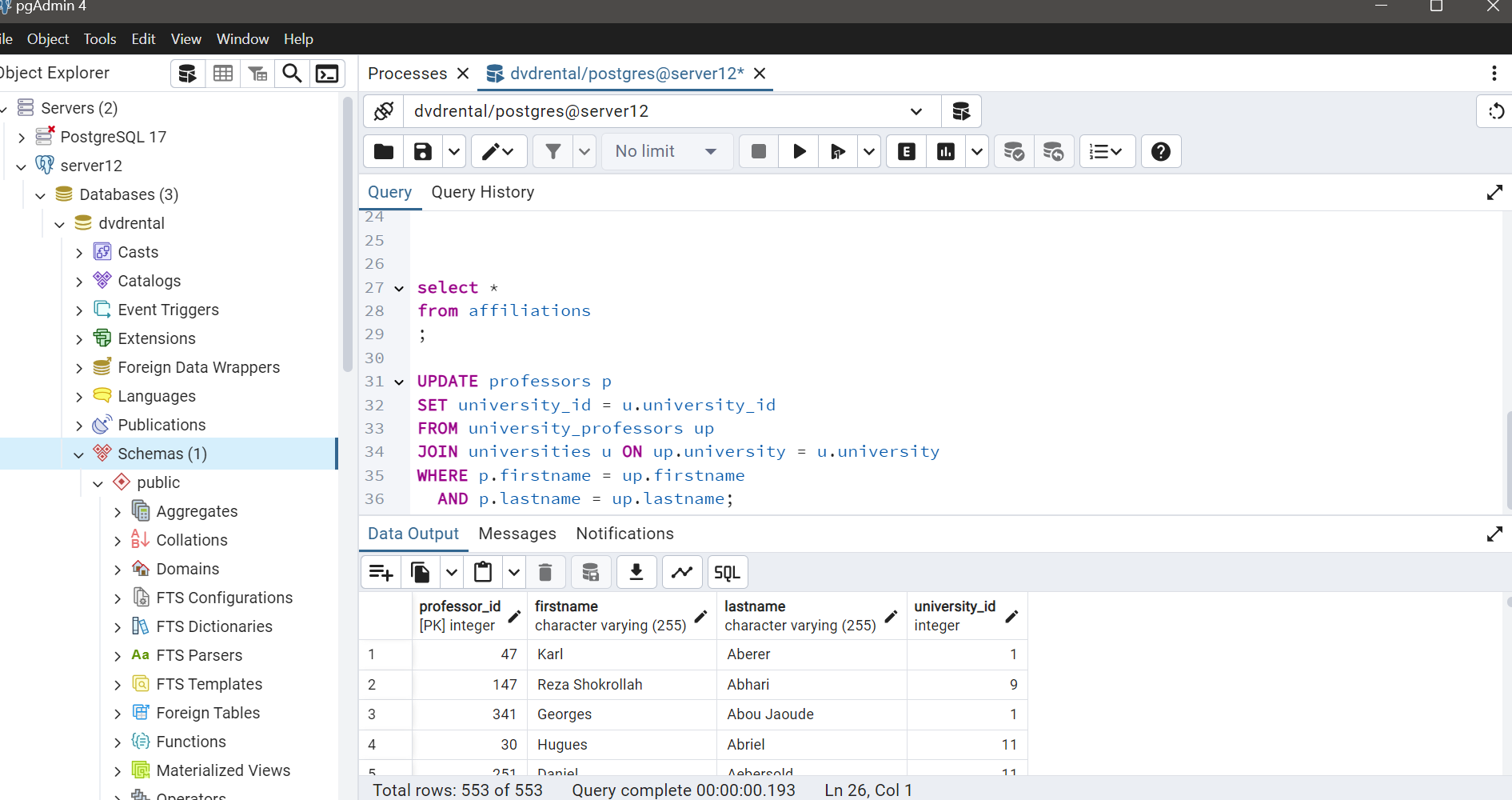












GITHUB LINK: