

Mastering SQL Bootcamp

with Genggam Data

Hands On :
Case Study 1
Using
Postgresql

Created by : Muhammad Pajrul Palah



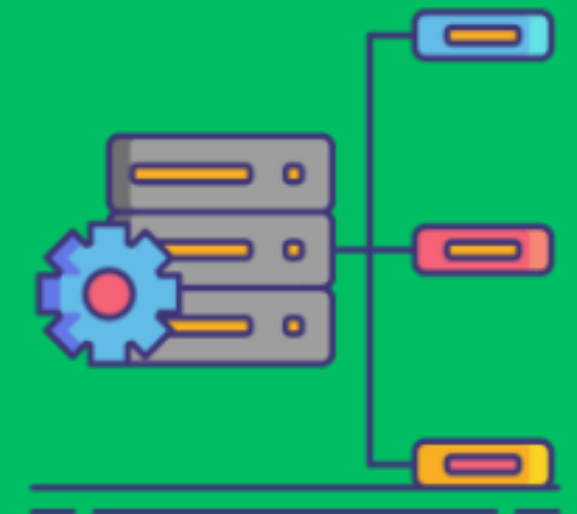


Problem Statement

Perusahaan Teknologi

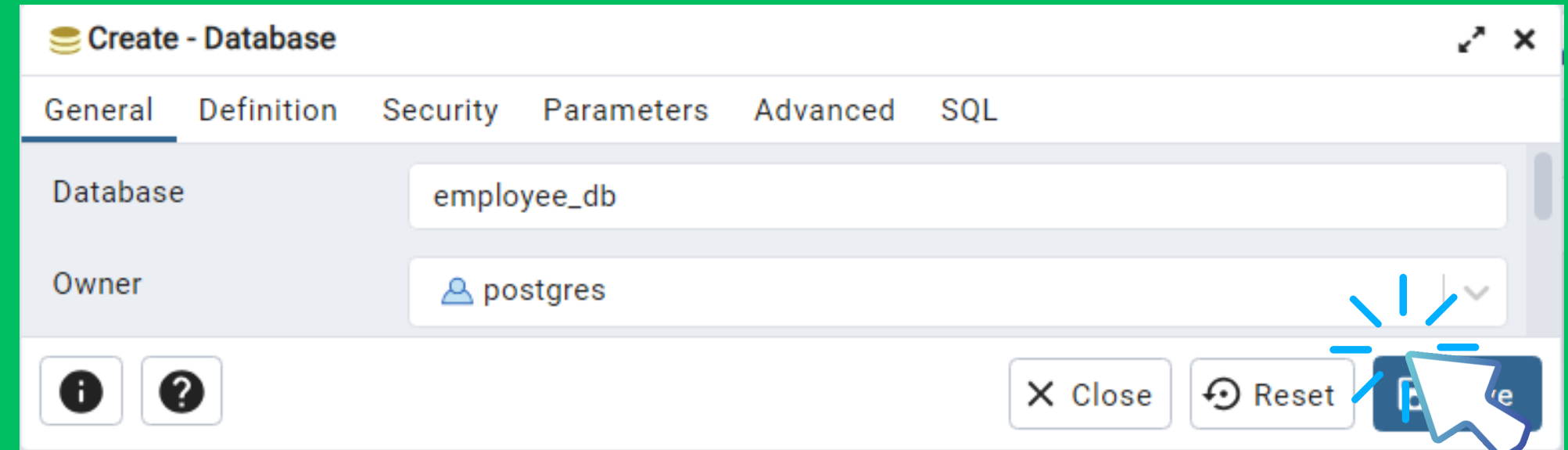
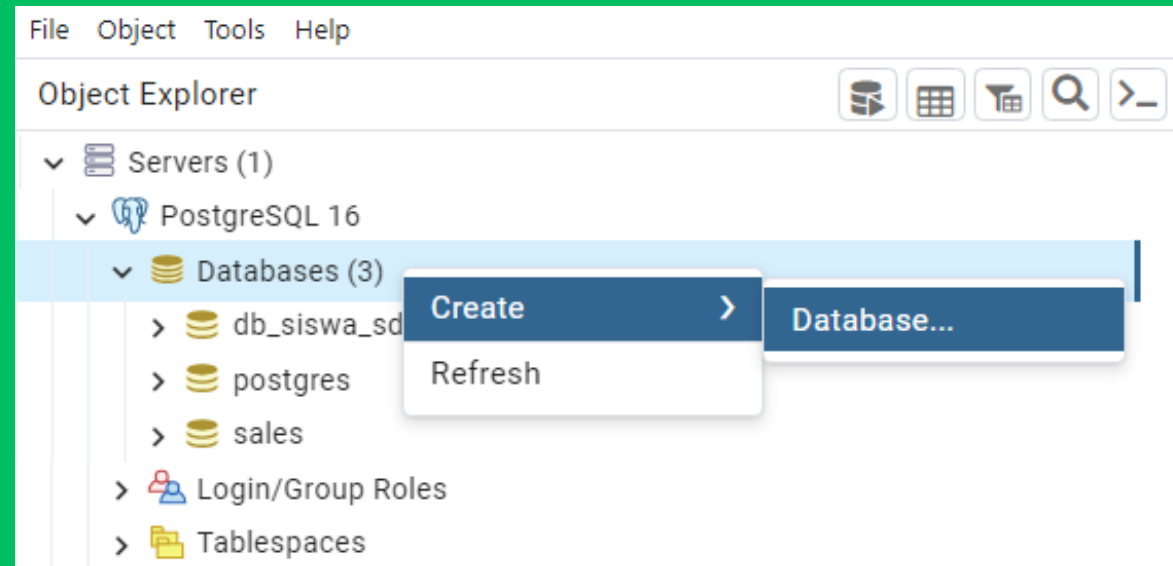
PT. Teknologi Maju adalah sebuah perusahaan teknologi yang sedang berkembang pesat. Perusahaan ini memiliki ratusan pegawai yang tersebar di berbagai wilayah di Indonesia.

Untuk mengelola data pegawai, PT. Teknologi Maju perlu menggunakan sebuah database. Database ini akan digunakan untuk menyimpan informasi tentang pegawai, seperti nama, usia, gaji, dan alamat. Dengan demikian kamu sebagai Data Analyst diminta untuk :



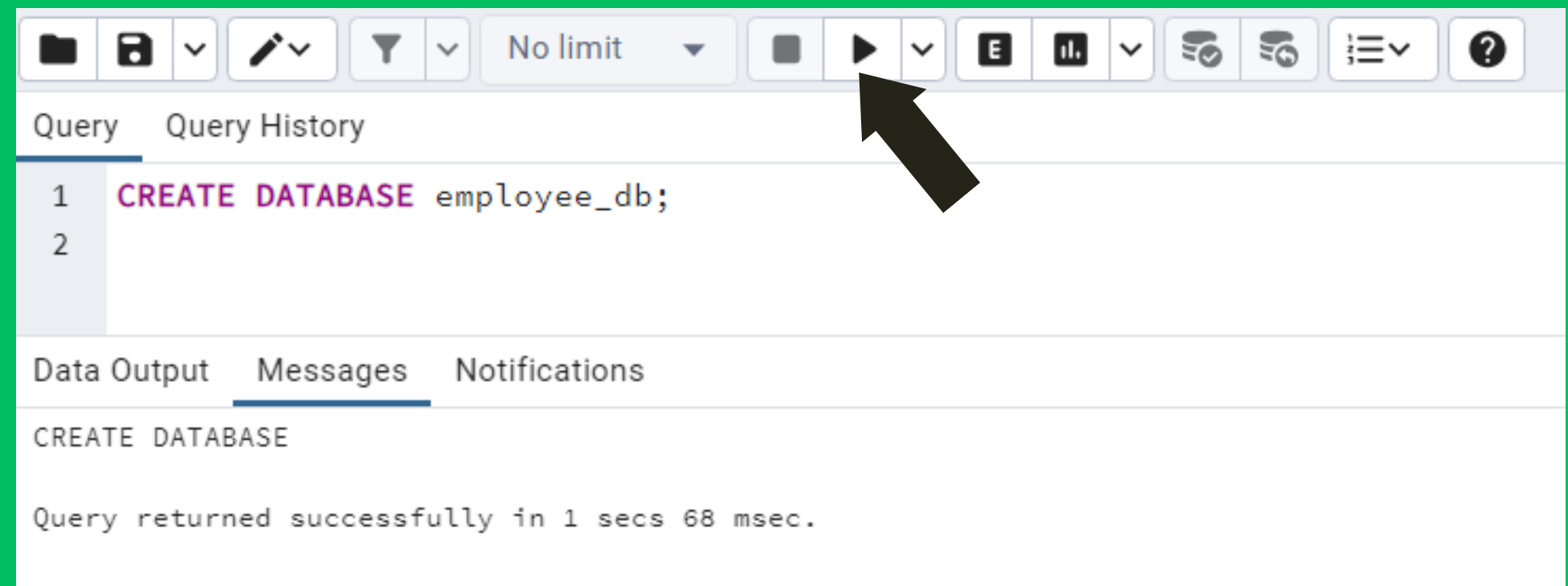
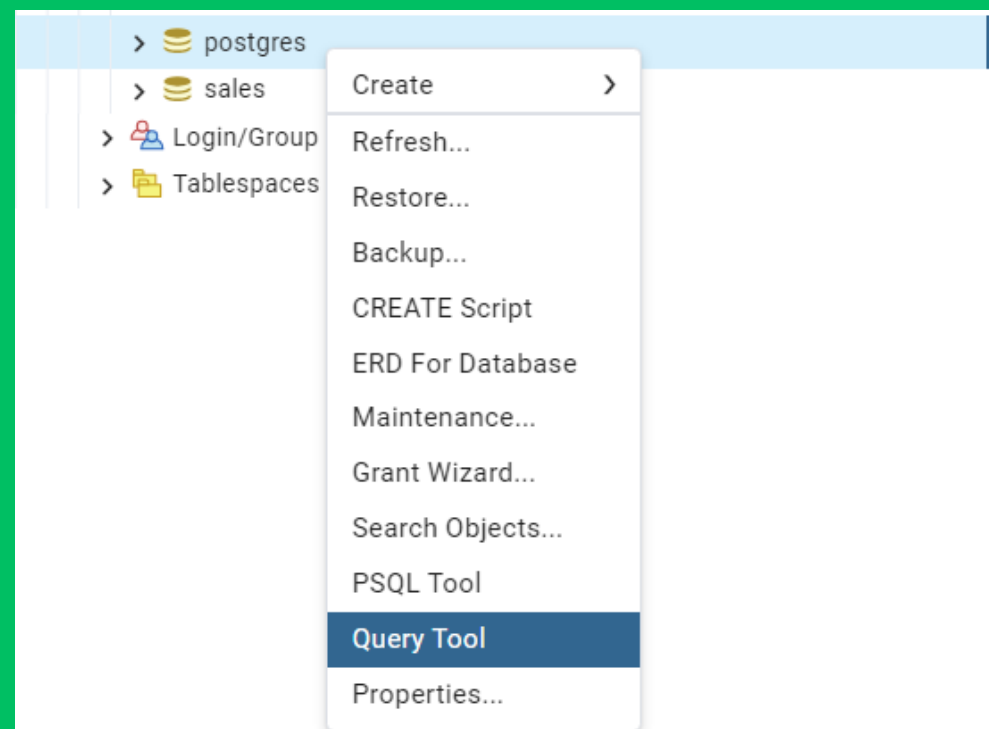
1. Buatlah sebuah database baru dengan nama "employee_db".

A. Use Toobar Menu



B. With SQL Query

CREATE DATABASE employee_db;



2. Di dalam database "employee_db", buatlah tabel "pegawai"
dengan kolom-kolom berikut:
id (integer, primary key)
nama (varchar)
nama_panggilan (varchar)
usia (integer)
salary (int)

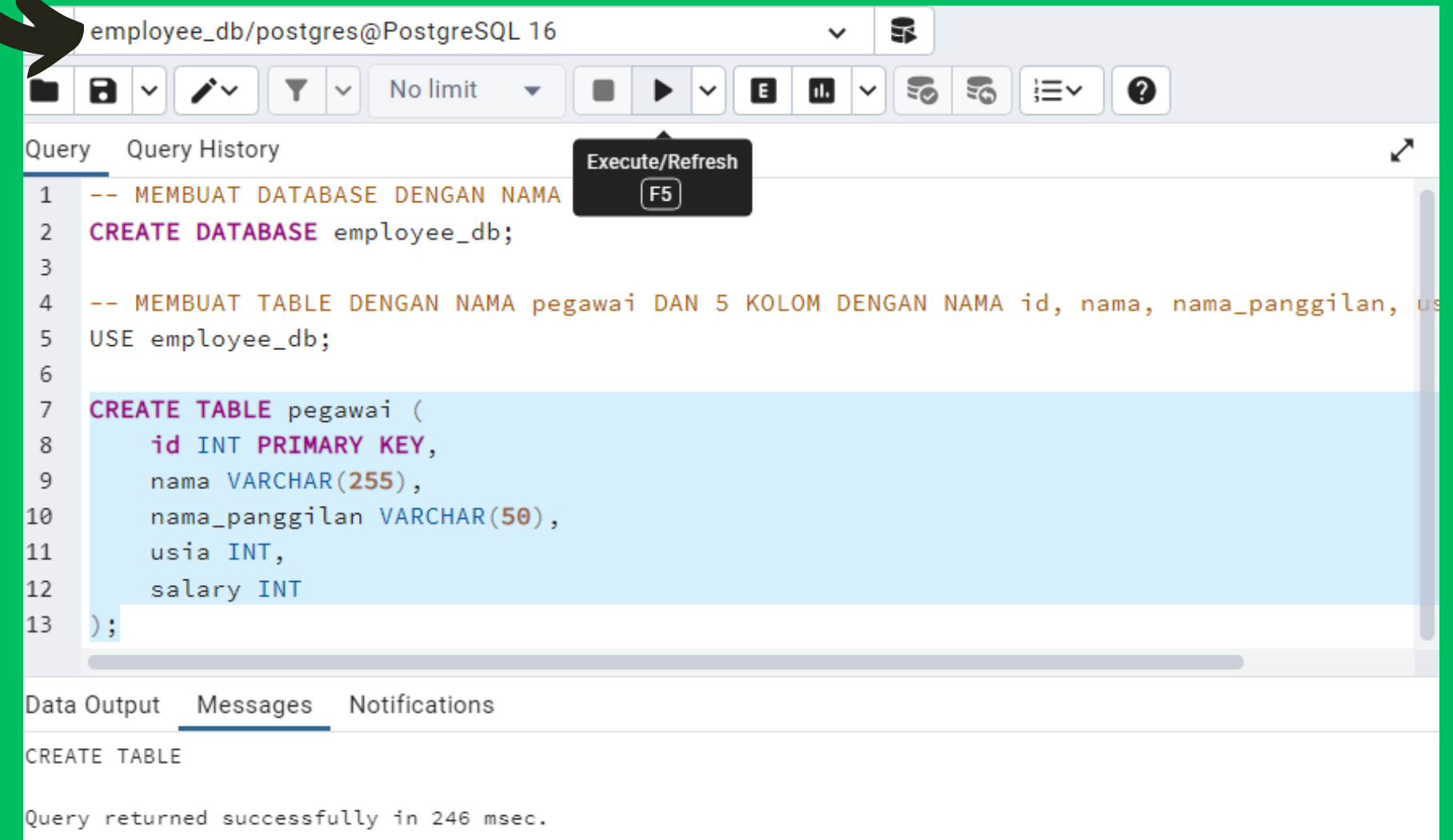
A. With SQL Query

```
CREATE TABLE pegawai (  
    id INT PRIMARY KEY,  
    nama VARCHAR(255),  
    nama_panggilan VARCHAR(50),  
    usia INT,  
    salary INT  
);
```

```
-- MEMUNCULKAN TABLE pegawai  
SELECT *  
FROM pegawai ;
```

Note:

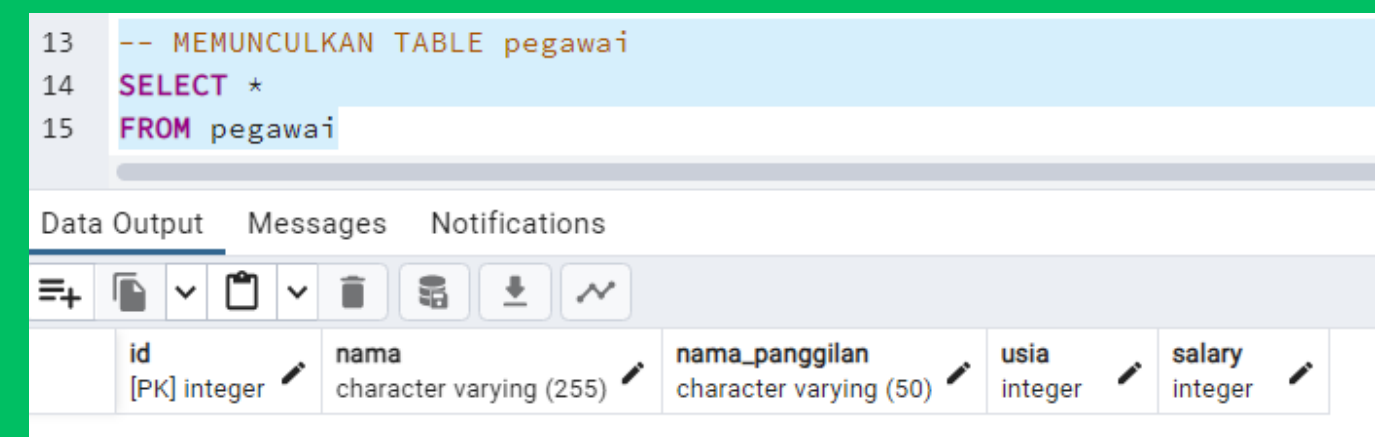
Ketika Ingin mengeksekusi query
pastikan berada pada nama lokasi
database yang tepat



The screenshot shows a PostgreSQL client window titled "employee_db/postgres@PostgreSQL 16". The "Query" tab is active, displaying the following SQL code:

```
1  -- MEMBUAT DATABASE DENGAN NAMA  
2  CREATE DATABASE employee_db;  
3  
4  -- MEMBUAT TABLE DENGAN NAMA pegawai DAN 5 KOLOM DENGAN NAMA id, nama, nama_panggilan, us  
5  USE employee_db;  
6  
7  CREATE TABLE pegawai (  
8      id INT PRIMARY KEY,  
9      nama VARCHAR(255),  
10     nama_panggilan VARCHAR(50),  
11     usia INT,  
12     salary INT  
13 );
```

A tooltip "Execute/Refresh (F5)" is visible over the "Execute" button in the toolbar. Below the query editor, the "Messages" tab is selected, showing the message: "CREATE TABLE Query returned successfully in 246 msec."



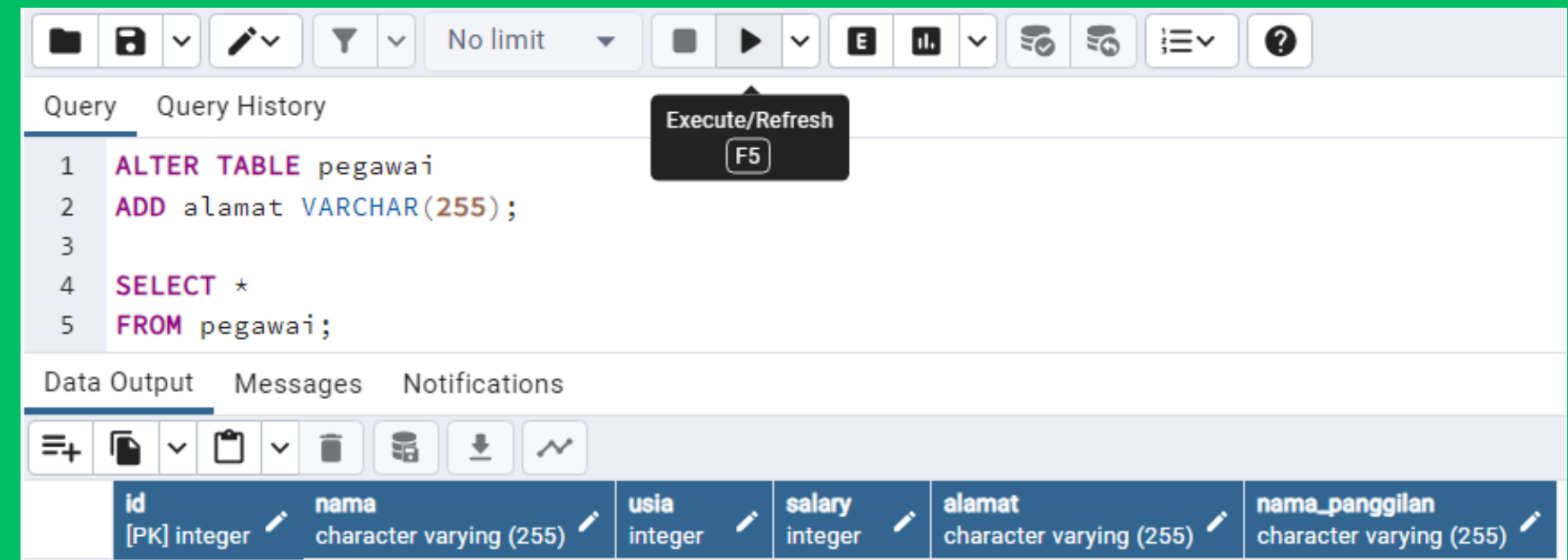
The screenshot shows the same PostgreSQL client window, but now the "Data Output" tab is active. It displays the result of the "SELECT *" query, showing the structure of the "pegawai" table.

	id	nama	nama_panggilan	usia	salary
	[PK] integer	character varying (255)	character varying (50)	integer	integer

3. Tambahkan kolom alamat dengan tipe data varchar ke dalam tabel "pegawai"

A. With SQL Query

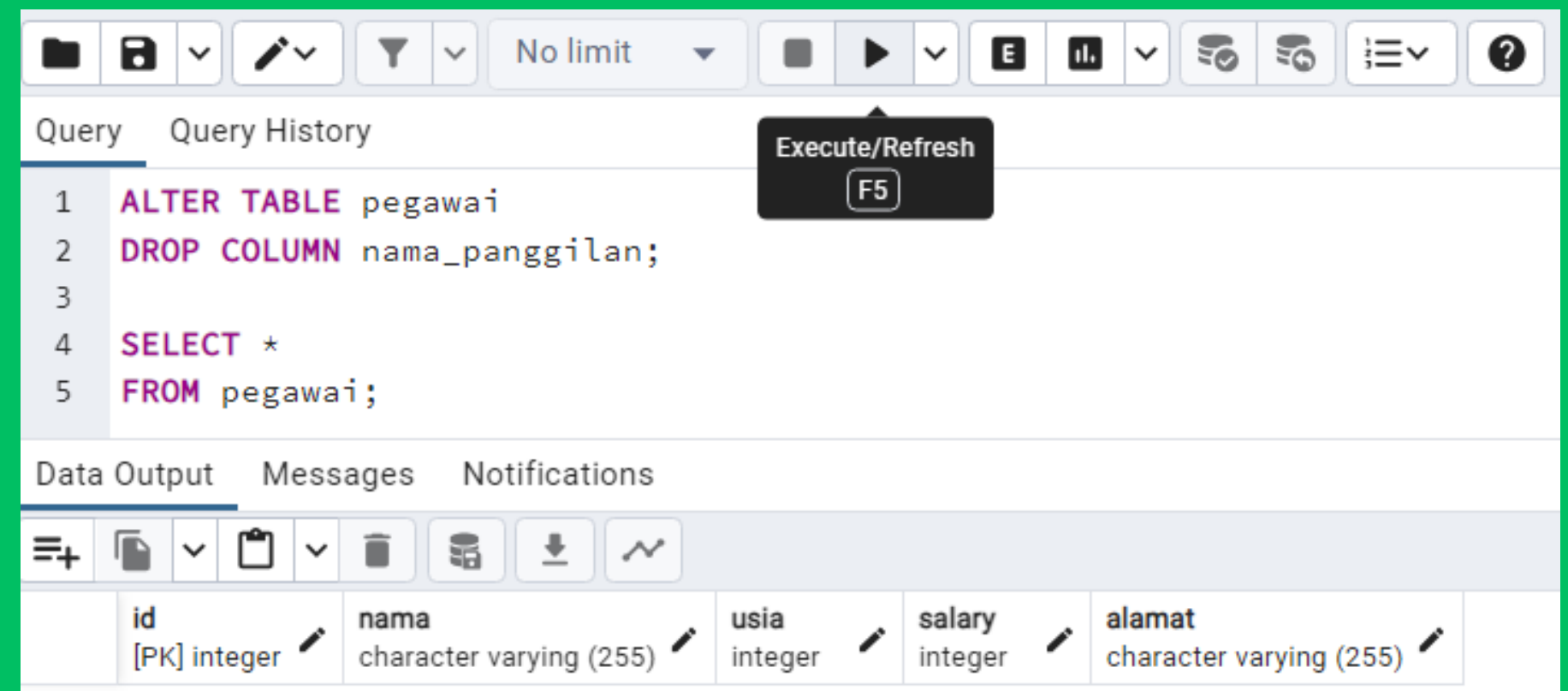
```
ALTER TABLE pegawai  
ADD alamat VARCHAR(255);
```



4. Hapus kolom nama_panggilan dari tabel "pegawai"

A. With SQL Query

```
ALTER TABLE pegawai  
DROP COLUMN nama_panggilan;
```

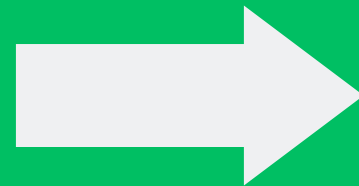


5. Ubah tipe data kolom salary menjadi bigint di tabel

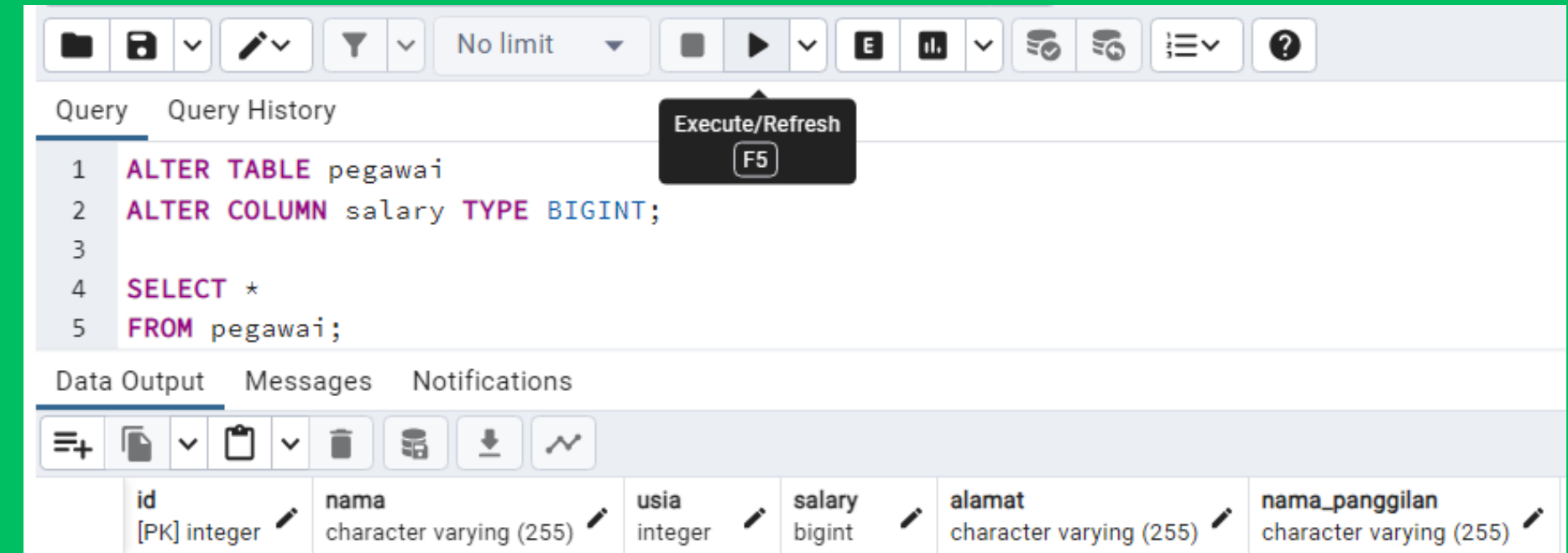
"pegawai" pegawai

A. With SQL Query

ALTER TABLE pegawai
ALTER COLUMN salary TYPE BIGINT ;



SELECT *
FROM pegawai;



6. Sisipkan beberapa data baru ke dalam tabel "pegawai" dengan informasi berikut:

'Ramzan', Usia: 30, Gaji: 5000000, Alamat: 'Jl. ABC No. 123'.

'Boodey', Usia: 35, Gaji: 6000000, Alamat: 'Jl. CDE No. 123'.

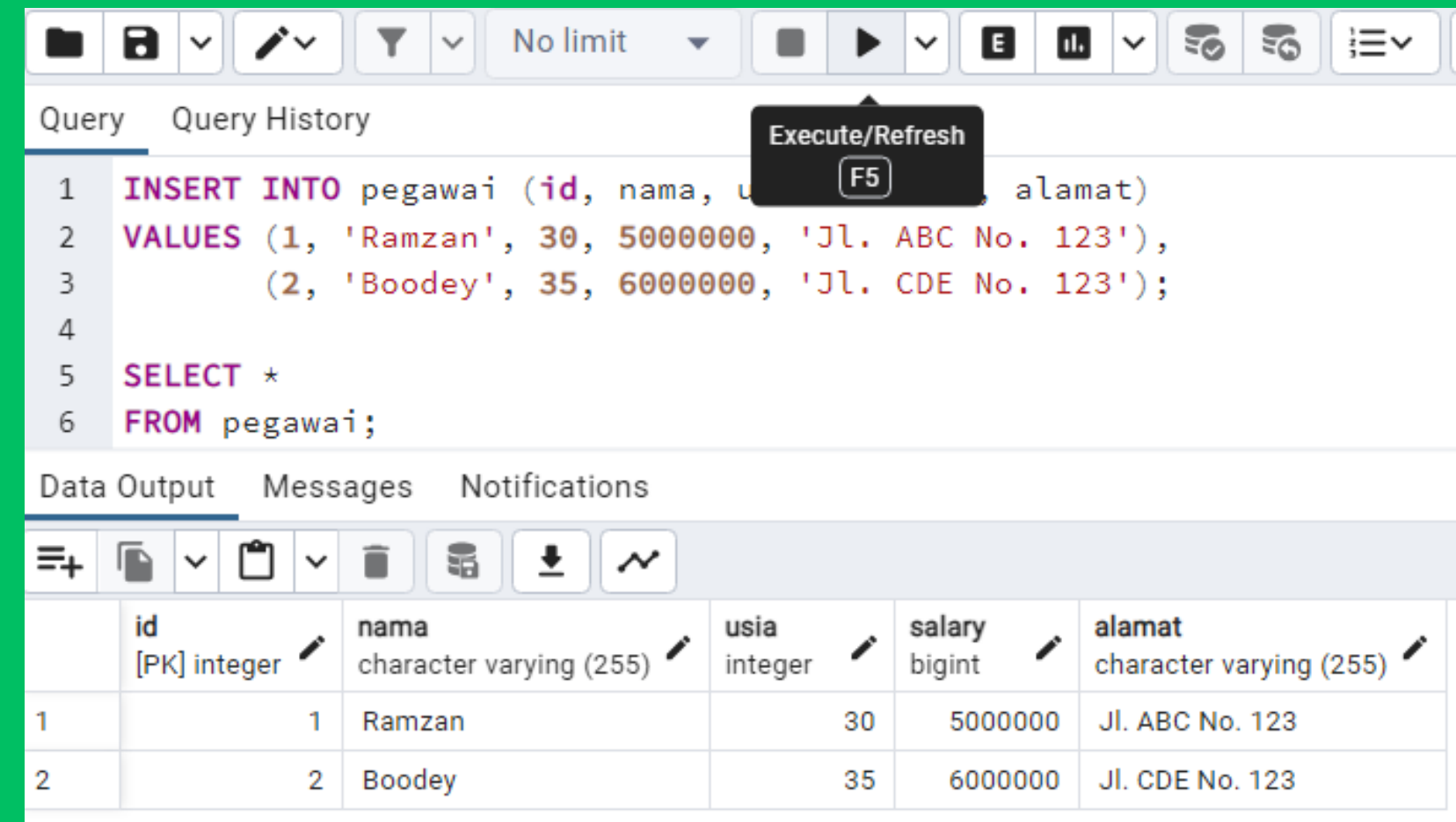


A. With SQL Query

INSERT INTO pegawai (id, nama, usia, salary, alamat)

VALUES

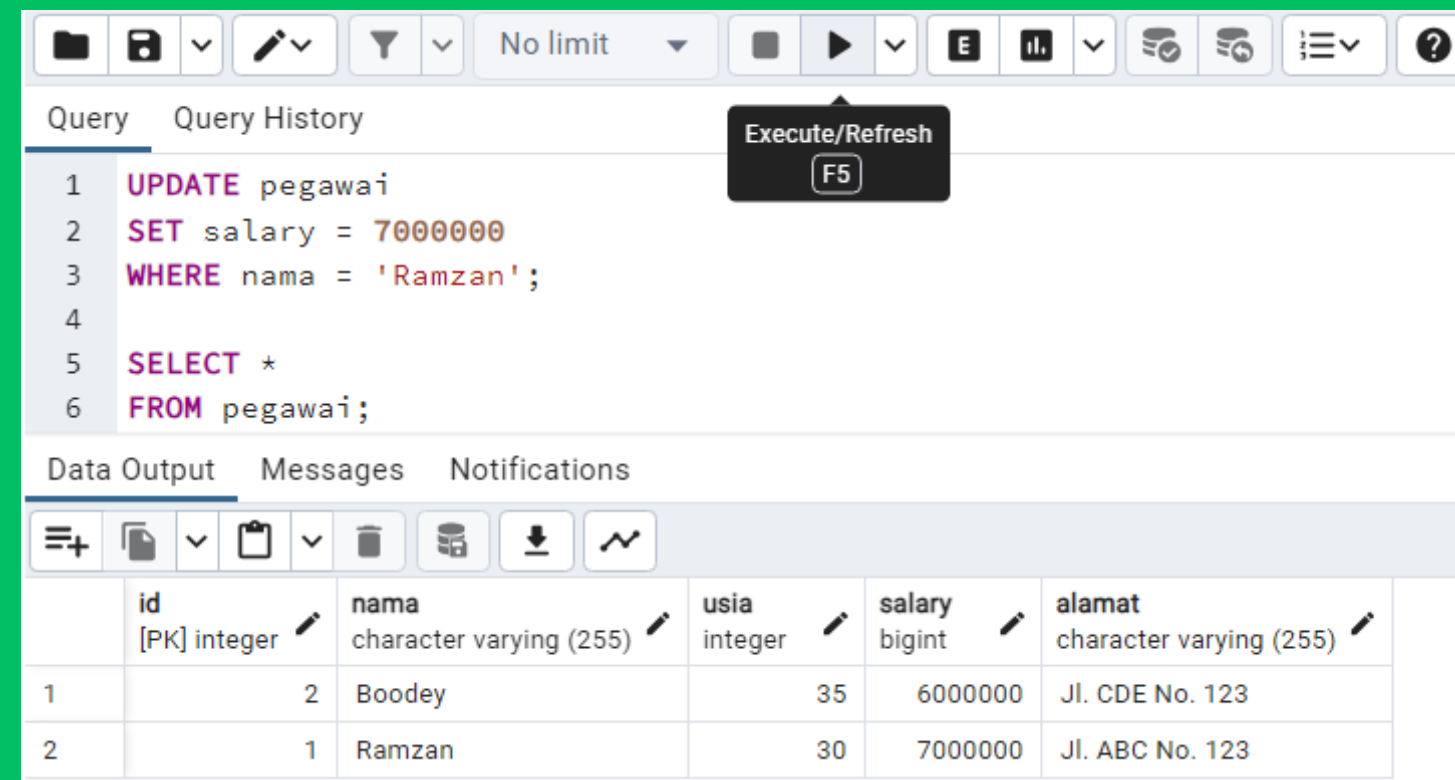
(1, 'Ramzan', 30, 5000000, 'Jl. ABC No. 123'),
(2, 'Boodey', 35, 6000000, 'Jl. CDE No. 123');



7. Perbarui gaji pegawai dengan nama 'Ramzan' menjadi 7000000.

A. With SQL Query

```
UPDATE pegawai  
SET salary = 7000000  
WHERE nama = 'Ramzan';
```



The screenshot shows a database management tool interface. The top toolbar includes icons for file operations, query execution, and settings. The 'Query' tab is active, displaying the following SQL query:

```
1 UPDATE pegawai  
2 SET salary = 7000000  
3 WHERE nama = 'Ramzan';  
4  
5 SELECT *  
6 FROM pegawai;
```

Below the query editor, the 'Data Output' tab is active, showing a table with the following data:

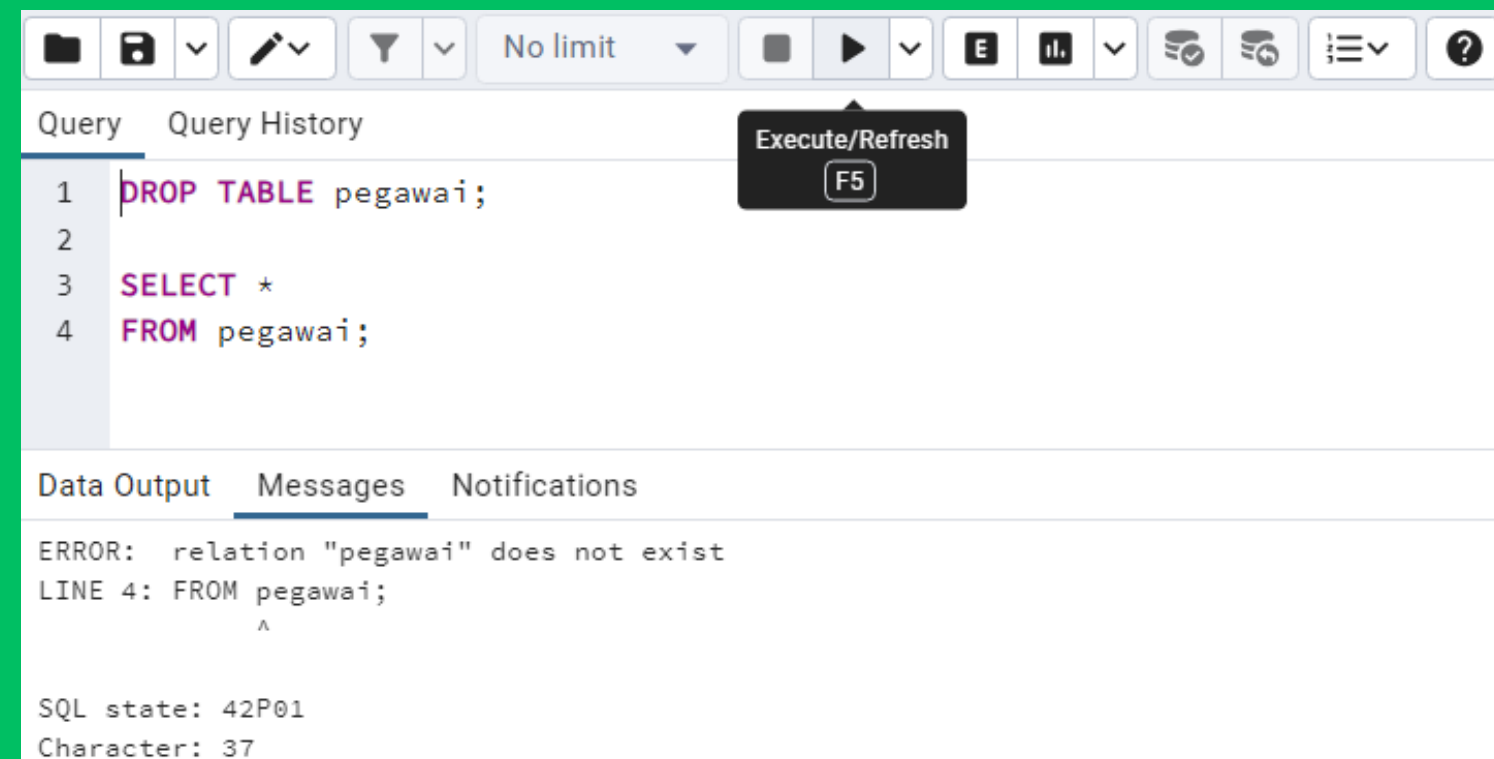
	id [PK] integer	nama character varying (255)	usia integer	salary bigint	alamat character varying (255)
1	2	Boodey	35	6000000	Jl. CDE No. 123
2	1	Ramzan	30	7000000	Jl. ABC No. 123

8. Hapus tabel "pegawai" dari database.

A. With SQL Query

```
DROP TABLE pegawai;
```

```
SELECT *  
FROM pegawai;
```



The screenshot shows a database management tool interface. The top toolbar is identical to the previous screenshot. The 'Query' tab is active, displaying the following SQL query:

```
1 DROP TABLE pegawai;  
2  
3 SELECT *  
4 FROM pegawai;
```

Below the query editor, the 'Data Output' tab is active, showing an error message:

```
ERROR: relation "pegawai" does not exist  
LINE 4: FROM pegawai;  
      ^  
  
SQL state: 42P01  
Character: 37
```

9. Hapus database "employee_db".

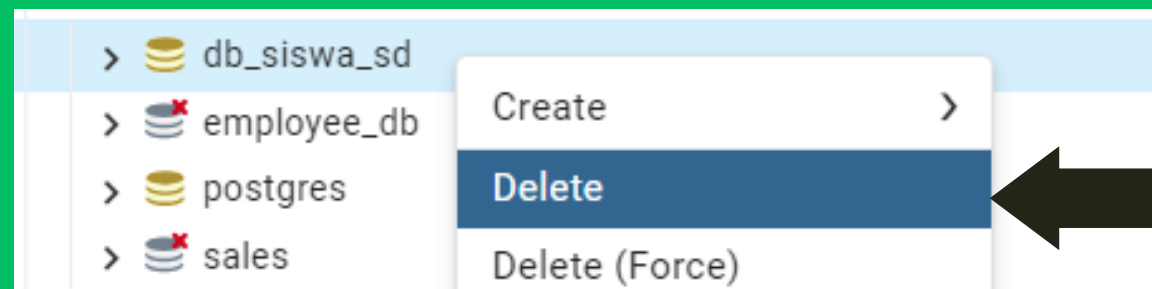
A. With SQL Query

DROP DATABASE employee_db;

atau Menggunakan

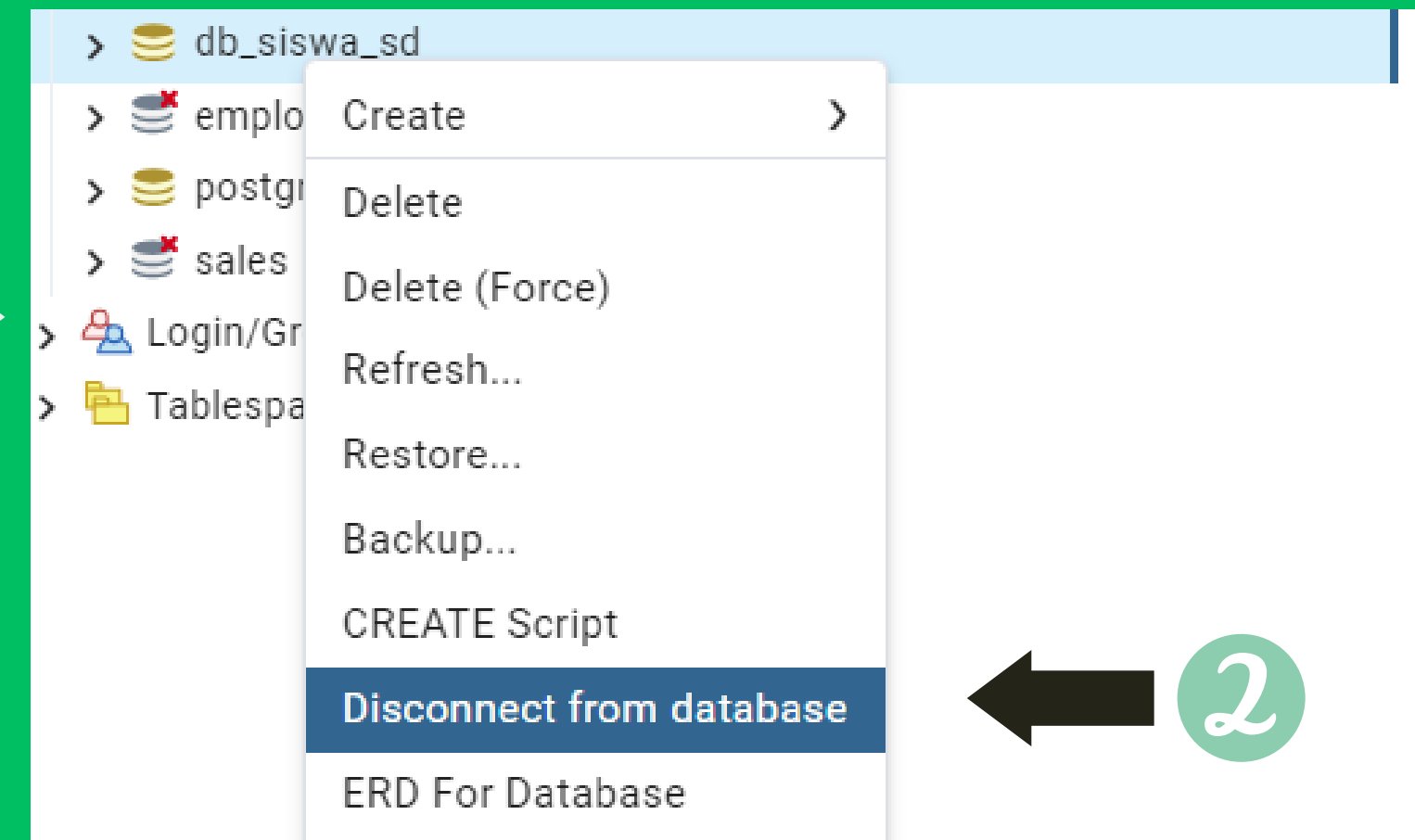
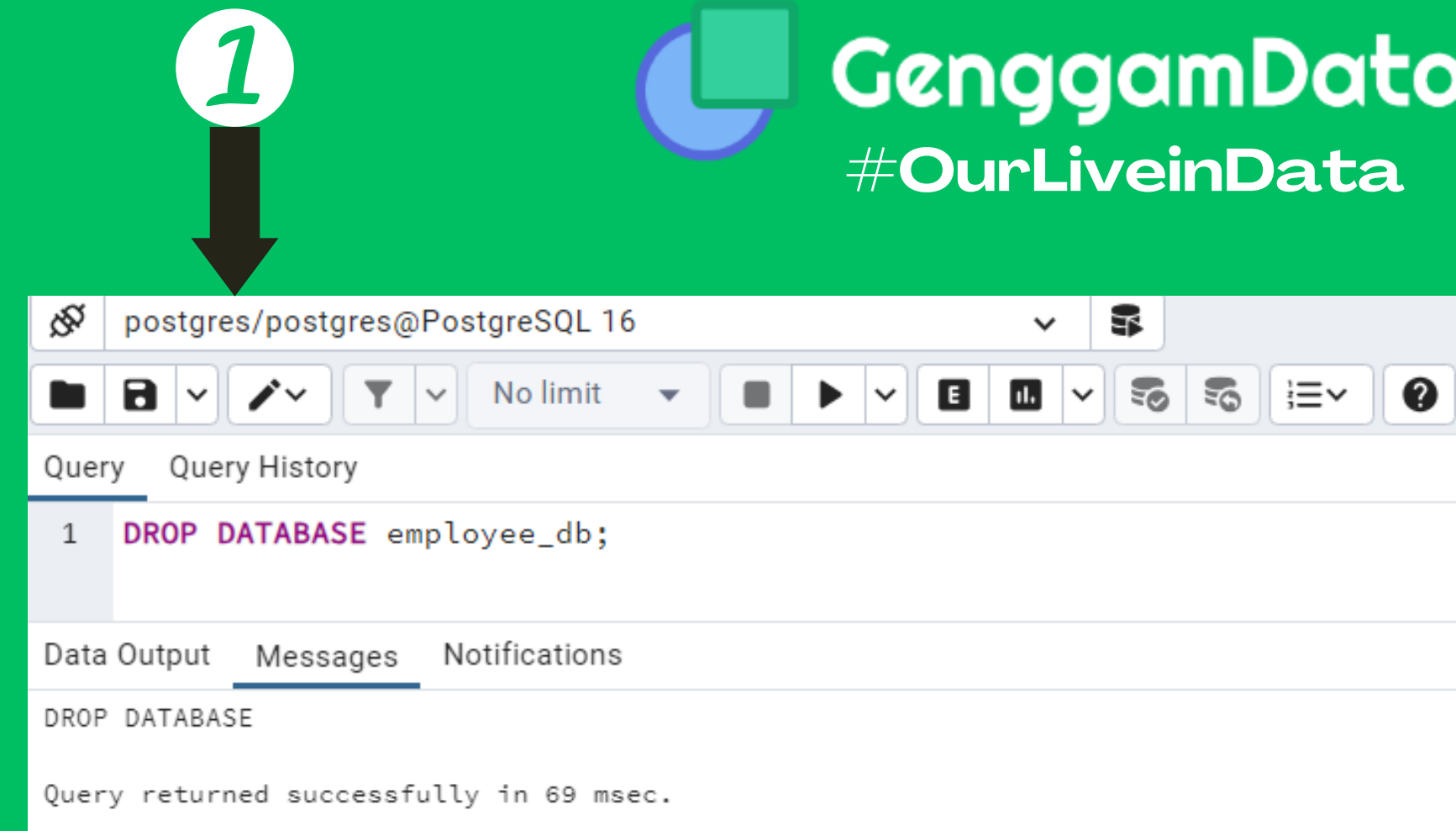
DROP DATABASE employee_db WITH (FORCE);

B. Toolbar Menu



Note:

1. Pastikan panel query database yang ingin dihapus diclose (employee_db/postgres@PostgreSQL) dan menggunakan panel query database lain.
2. Mendisconnect database yang akan dihapus seperti pada gambar disamping
3. Jika masih tidak bisa maka cara terakhir adalah menghapus dengan fitur delete mengklik kanan (mouse) pada database yang akan dihapus dan pilih opsi delete





GenggamData
#OurLiveinData

Thank
you!