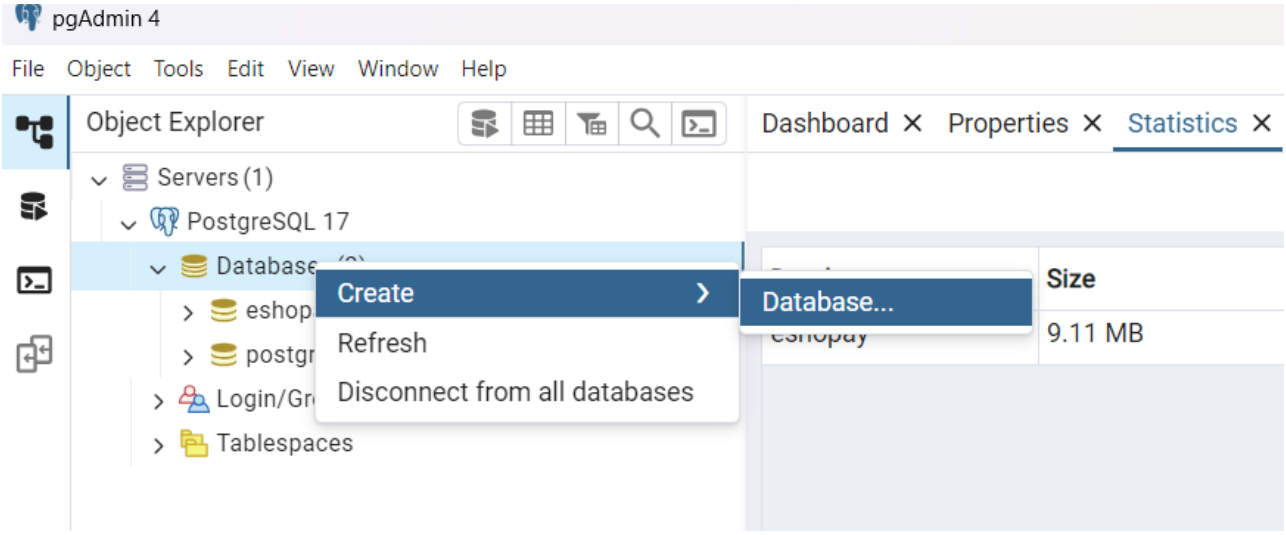
Database

Create Database

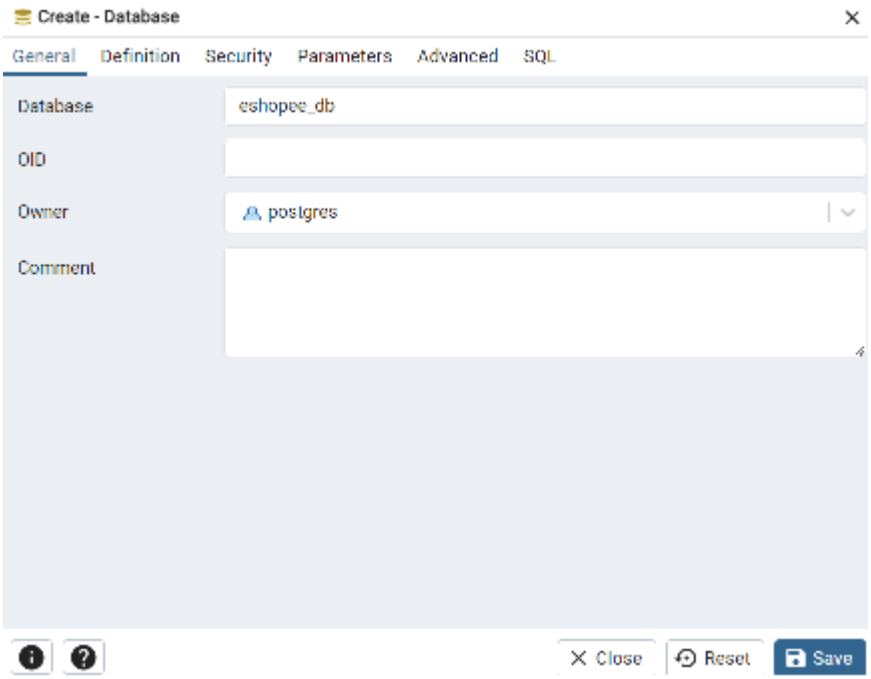
Ada dua alternatif untuk create database, kita bisa gunakan tool PgAdmin atau console PgSQL.

- PGAdmin

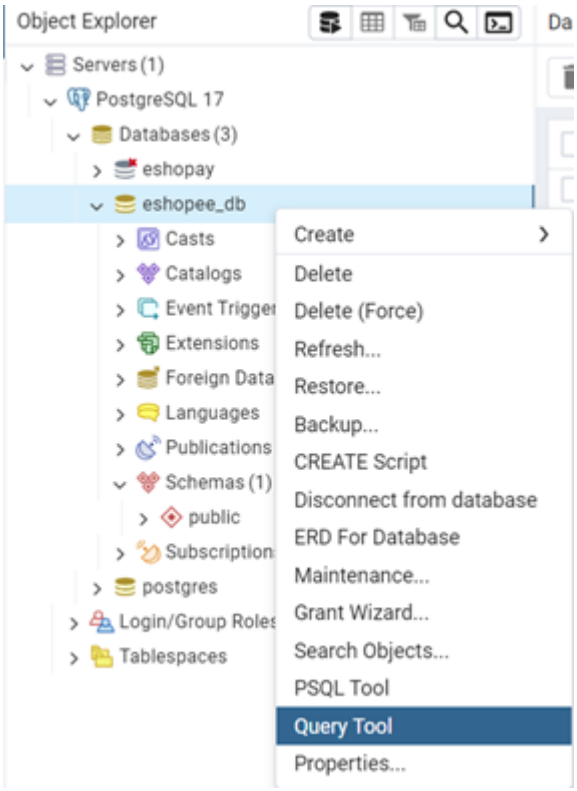
1. Buka PGAdmin di start menu window, connect ke database lalu klik kanan menu Database dan pilih menu Create > Datatabase.



2. Di open dialog create nama dabatase : eshopee_db. Lalu klik tombol Save.

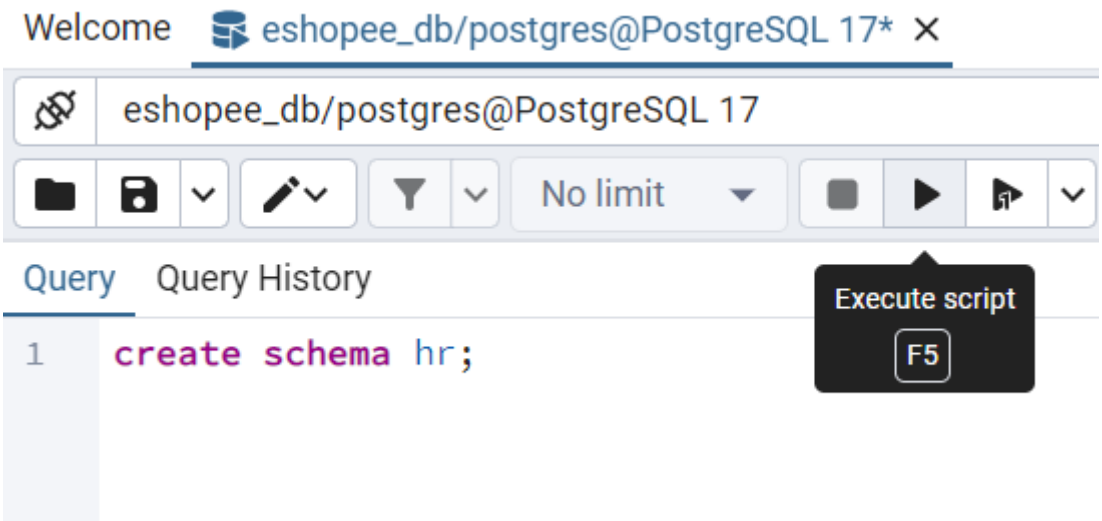


3. Klik kanan database eshopee_db, lalu pilih Query Tool.



4. Create schema hr di query tool. Block dua script dibawah lalu execute.

```
create schema hr;
```



5. PostgreSQL hanya mencari table di schema public, karena kita akan meng-akses schema lain, kita tambahkan schema hr ke search_path.

```
SET search_path TO hr;
```

atau jika ingin set permanent :

```
ALTER ROLE postgres SET search_path TO hr;
```

6. Buka DDL & DML untuk create & insert table di schema hr.

- *PgSQL Command*

1. Kita bisa gunakan command sql di PgSQL dengan cara :

```
CREATE DATABASE eshopee_db
    WITH
    OWNER = postgres
    ENCODING = 'UTF8'
    LOCALE_PROVIDER = 'libc'
    CONNECTION LIMIT = -1
    IS_TEMPLATE = False;
```

2. Create schema hr di query tool. Block dua script dibawah lalu execute.

```
create schema hr;
```

3. Execute sintak berikut di Query Tool, agar kita bisa akses semua object database untuk dua schema diatas tanpa menambahkan prefix hr.

```
SET search_path TO hr;
```

4. Execute sintak sql dibawah ini di Query Tool database *eshopee_db*.

```
CREATE TABLE hr.regions (
    region_id SERIAL PRIMARY KEY,
    region_name VARCHAR(25)
);

CREATE TABLE hr.countries (
    country_id CHARACTER (2) PRIMARY KEY,
    country_name VARCHAR(40),
    region_id INTEGER NOT NULL,
     FOREIGN KEY (region_id) REFERENCES hr.regions (region_id) ON UPDATE
CASCADE ON DELETE CASCADE
);

CREATE TABLE hr.locations (
    location_id SERIAL PRIMARY KEY,
    street_address VARCHAR(40),
    postal_code VARCHAR(12),
    city VARCHAR(30) NOT NULL,
```

```
    state_province VARCHAR(25),
    country_id CHARACTER (2) NOT NULL,
    FOREIGN KEY (country_id) REFERENCES hr.countries (country_id) ON UPDATE
CASCADE ON DELETE CASCADE
);

CREATE TABLE hr.departments (
    department_id SERIAL PRIMARY KEY,
    department_name VARCHAR(30) NOT NULL,
    location_id INTEGER,
     FOREIGN KEY (location_id) REFERENCES hr.locations (location_id) ON
UPDATE CASCADE ON DELETE CASCADE
);

CREATE TABLE hr.jobs (
    job_id SERIAL PRIMARY KEY,
    job_title VARCHAR(35) NOT NULL,
    min_salary NUMERIC (8, 2),
    max_salary NUMERIC (8, 2)
);

CREATE TABLE hr.employees (
    employee_id SERIAL PRIMARY KEY,
    first_name VARCHAR(20),
    last_name VARCHAR(25) NOT NULL,
    email VARCHAR(100) NOT NULL,
    phone_number VARCHAR(20),
    hire_date DATE NOT NULL,
    job_id INTEGER NOT NULL,
    salary NUMERIC (8, 2) NOT NULL,
    manager_id INTEGER,
    department_id INTEGER,
     FOREIGN KEY (job_id) REFERENCES hr.jobs (job_id) ON UPDATE CASCADE ON
DELETE CASCADE,
    FOREIGN KEY (department_id) REFERENCES hr.departments (department_id) ON
UPDATE CASCADE ON DELETE CASCADE,
    FOREIGN KEY (manager_id) REFERENCES hr.employees (employee_id) ON UPDATE
CASCADE ON DELETE CASCADE
);

CREATE TABLE hr.dependents (
    dependent_id SERIAL PRIMARY KEY,
    first_name VARCHAR(50) NOT NULL,
    last_name VARCHAR(50) NOT NULL,
    relationship VARCHAR(25) NOT NULL,
    employee_id INTEGER NOT NULL,
     FOREIGN KEY (employee_id) REFERENCES hr.employees (employee_id) ON
DELETE CASCADE ON UPDATE CASCADE
);
```