

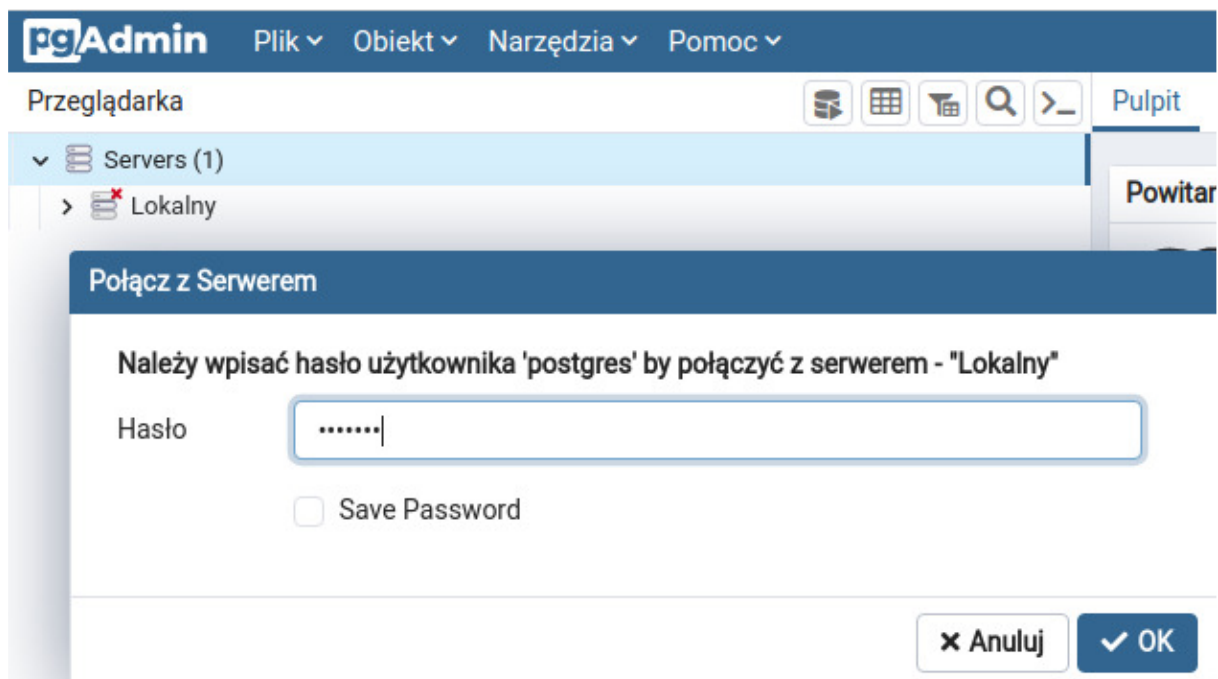
Zarządzanie bazą danych PostgreSQL za pomocą pgAdmin i języka SQL

Opracował Dawid Warchoł
Politechnika Rzeszowska,
Katedra Informatyki i Automatyki.

1. Logowanie i rozpoczęcie pracy

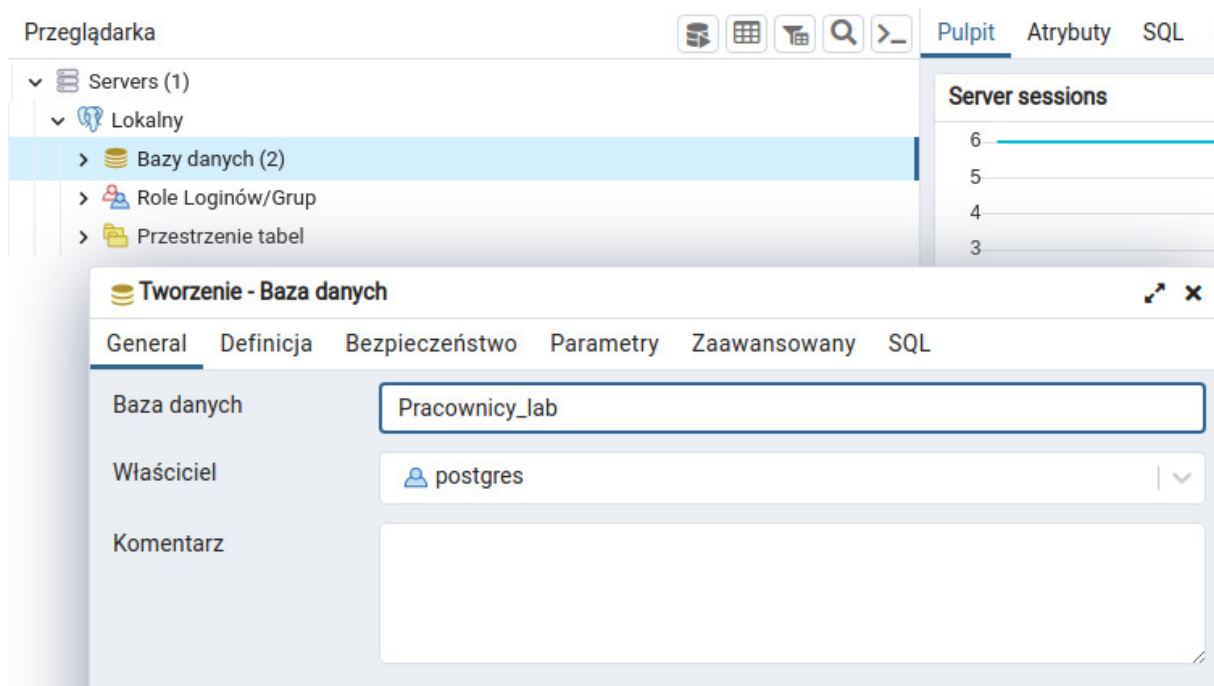
Uruchomić system operacyjny o nazwie PostgreSQL oraz aplikację pgAdmin 4. Jest to popularny program do zarządzania bazą danych PostgreSQL.

Kliknąć podwójnie w „Servers” w zakładce „Przeglądarka” z lewej strony okna i zalogować się wpisując hasło „student”.

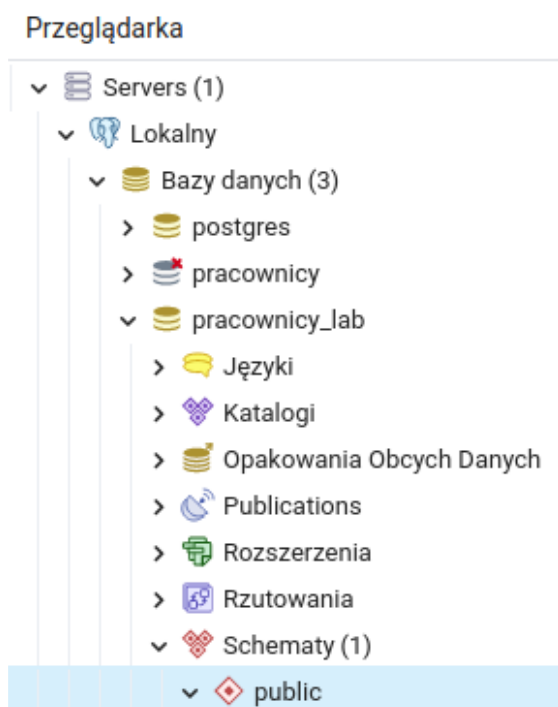


2. Tworzenie nowej bazy danych

Przejsz do „Lokalny” i kliknąć prawym klawiszem w „Bazy danych”, a następnie „Utwórz” -> „Baza danych...”. Pojawi się okno tworzenia bazy, w którym należy podać jej nazwę „Pracownicy_lab”.



Przejsć do nowo utworzonej bazy i wybrać schemat „public”. Będą w nim przechowywane nasze tabele.



3. Tworzenie tabel

Zapoznaj się z pojęciami klucza unikatowego, klucza głównego i klucza obcego.

- Klucz unikatowy (unikalny, jednoznaczny) – atrybut, które wartości nie mogą się powtarzać w różnych wpisach w bazie (wierszach).
- Klucz główny (podstawowy) – atrybut będący identyfikatorem tabeli. Klucz główny jest zawsze unikatowy.

- Klucz obcy – atrybut, który odnosi się do klucza głównego znajdującego się w tej samej lub innej tabeli. Jeśli tabela zawiera klucz obcy odnoszący się do innej tabeli, to oznacza, że obie tabele są ze sobą powiązane.

Jako pierwszą należy utworzyć tabelę ETAT. Kliknąć prawym klawiszem w „Tabele”, a następnie „Utwórz” -> „Tabela...”. W zakładce „General” uzupełnić pole „Nazwa”. Opcjonalnie można wpisać komentarz.

Tworzenie - Tabela

General Kolumny Zaawansowany Ograniczenia Partycje Parametry Bezpieczeństwo SQL

Nazwa: ETAT

Właściciel: postgres

Schemat: public

Przestrzeń Tabel: Select an item...

Partitioned table? ☐

Komentarz: Zawiera etaty

[i] [?] X Zamknij ↺ Reset Zapisz

W zakładce „Kolumny” wpisać dane wszystkich kolumn zgodnie z poniższym rysunkiem i tabelą. Nową kolumnę można wstawić klikając w znak „+” w prawym górnym rogu okna.

General **Kolumny** Zaawansowany Ograniczenia Partycje Parametry Bezpieczeństwo SQL

Dziedziczone z tabel(i): Wybór dziedziczenia z...

Kolumny +

	Nazwa	Typ danych	Length/Precision	Skala	Nie NULL?	Klucz główny?	Domyślnie
	ID_ETAT	integer			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
	NAZWA	character varying	20		<input checked="" type="checkbox"/>	<input type="checkbox"/>	
	PLACA_MIN	real			<input type="checkbox"/>	<input type="checkbox"/>	
	PLACA_MAX	real			<input type="checkbox"/>	<input type="checkbox"/>	

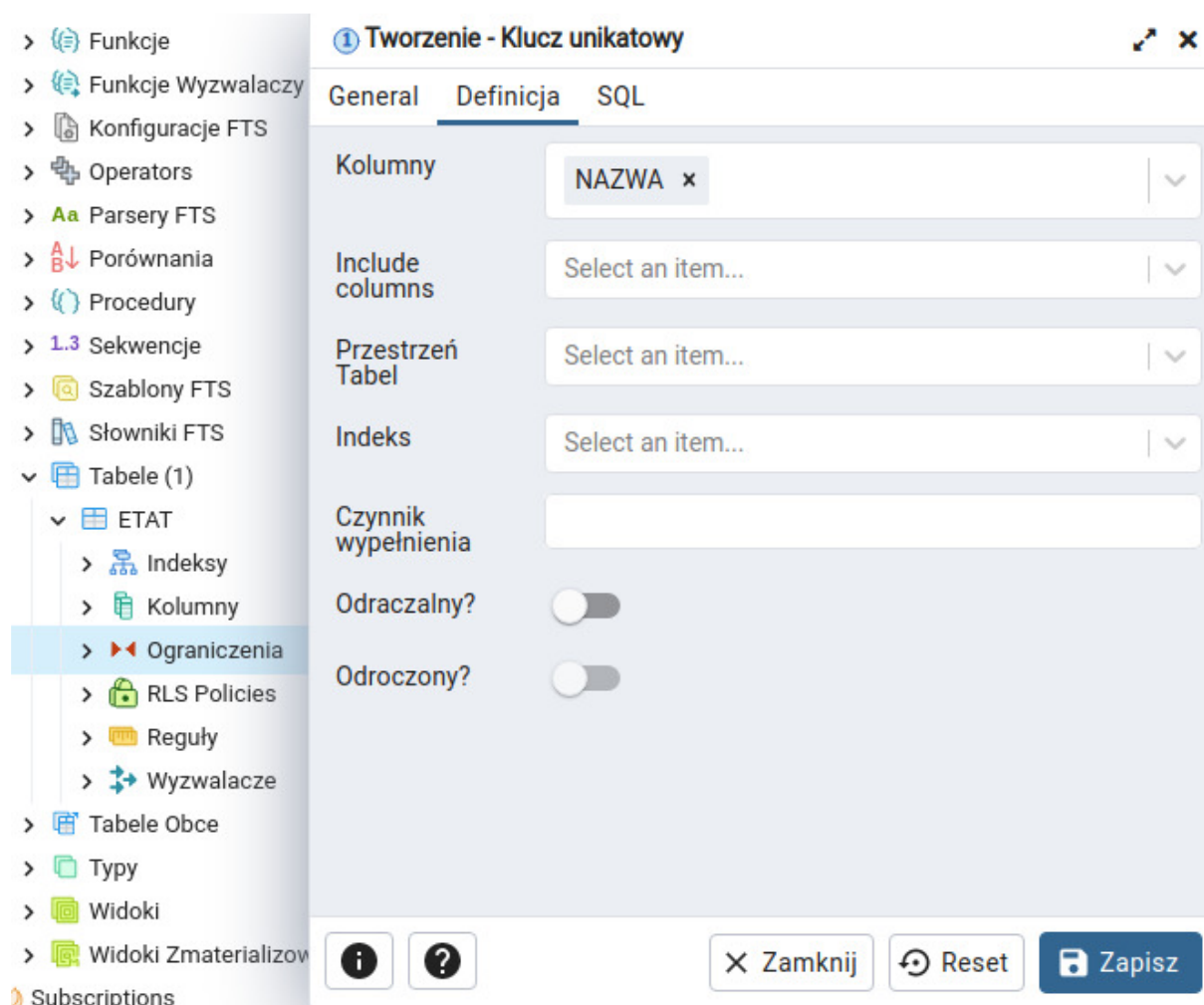
[i] [?] X Zamknij ↺ Reset Zapisz

ETAT

Kolumna	Typ	Nie NULL	Domyślnie	Więzy integralności
ID_ETAT	integer	TAK		klucz główny
NAZWA	character varying (20)	TAK		klucz unikatowy
PLACA_MIN	real			
PLACA_MAX	real			

Po wypełnieniu pól kliknąć w przycisk „Zapisz”. Tabela powinna się pojawić na liście tabel schematu „public”.

Aby utworzyć klucz unikatowy należy rozwinąć tabelę ETAT, a następnie kliknąć prawym klawiszem w „Ograniczenia” -> „Utwórz” -> „Klucz unikatowy...”. W zakładce „Definicja”, w polu „Kolumny” należy wybrać z listy kolumnę NAZWA i kliknąć w przycisk „Zapisz”.



The screenshot shows the 'Tworzenie - Klucz unikatowy' (Creating - Unique Key) dialog box in a database management tool. The 'Definicja' (Definition) tab is selected. The 'Kolumny' (Columns) field contains 'NAZWA'. The 'Include columns' field is empty. The 'Przestrzeń Tabel' (Table Space) field is empty. The 'Indeks' (Index) field is empty. The 'Czynnik wypełnienia' (Fill Factor) field is empty. There are two toggle switches: 'Odraczalny?' (Deferred) and 'Odroczony?' (Deferred). At the bottom are buttons for 'Zamknij' (Close), 'Reset', and 'Zapisz' (Save).

Zad. 1. Samodzielnie należy utworzyć pozostałe tabele ZESPOL oraz PRACOWNIK. Opis tworzenia kluczy obcych znajduje się w punkcie 4 instrukcji.

ZESPOL

Kolumna	Typ	Nie NULL	Domyślnie	Więzy integralności
ID_ZESP	Integer	TAK		klucz główny
NAZWA	character varying (20)	TAK		
ADRES	character varying (20)			

PRACOWNIK

Kolumna	Typ	Nie NULL	Domyślnie	Więzy integralności
NUMER	integer	TAK		klucz główny
NAZWISKO	character varying (30)	TAK		
ETAT	integer			Klucz obcy do tabeli ETAT i kolumny ID_ETAT
SZEF	integer			Klucz obcy do tabeli PRACOWNIK i kolumny NUMER
PRACUJE_OD	date		<i>now()</i>	
PLACA_OD	real			
PLACA_DOD	real			
ID_ZESP	integer			Klucz obcy do tabeli ZESPOL i kolumny ID_ZESP

4. Tworzenie kluczy obcych

Aby utworzyć klucze należy rozwinąć tabelę (w naszym przypadku PRACOWNIK), a następnie kliknąć prawym klawiszem w „Ograniczenia” -> „Utwórz” -> „Klucz obcy...”.

Aby określić, że kolumna ETAT tabeli PRACOWNIK jest kluczem obcym do kolumny ID_ETAT tabeli ETAT należy wpisać nazwę „PRACOWNIK_ETAT_fkey” w polu „Nazwa” w zakładce General.

Tworzenie - Klucz obcy

General
Definicja
Kolumny
Akcja
SQL

Nazwa

PRACOWNIK_ETAT_fkey

Komentarz

//

Następnie należy uzupełnić dane w zakładce „Kolumny” zgodnie z poniższym rysunkiem, kliknąć w przycisk „Add” oraz „Zapisz”.

Tworzenie - Klucz obcy

General
Definicja
Kolumny
Akcja
SQL

Kolumny

Lokalna kolumna

ETAT

Odniesienia

public."ETAT"

Referencja

ID_ETAT

Add

Lokalny	Wskazywane	Referenced Table
---------	------------	------------------

i
?

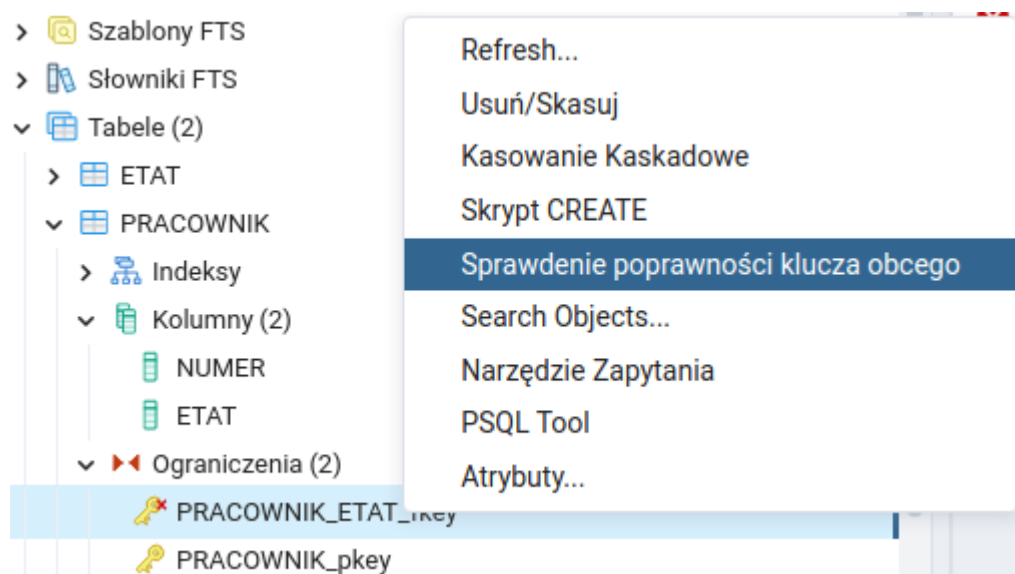
X Zamknij

Reset

Zapisz

Jeśli utworzony klucz obcy nie pojawi się na liście ograniczeń, to należy kliknąć prawym klawiszem w Ograniczenia i wybrać opcję „Odśwież”.

W nowo utworzony klucz obcy należy kliknąć prawym klawiszem i wybrać opcję „Sprawdzenie poprawności klucza obcego”. Jeśli klucz utworzy się poprawnie, to zmieni się jego ikona ze złotej na srebrną.

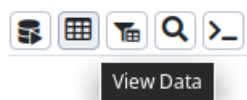


W analogiczny sposób należy dodać klucze obce ID_ZESP oraz SZEFE.

Uwaga! Atrybut SZEFE powinien odnosić się do klucza głównego NUMER znajdującego się w tej samej tabeli (PRACOWNIK). Z tego powodu w polu „Referencja” należy wpisać NUMER, a nie SZEFE. Idea tego klucza jest taka, że pracownik może być szefem innego pracownika (identyfikowanego przez atrybut NUMER).

5. Wypełnianie tabeli danymi

Aby wypełnić tabelę danymi, tzn. wstawić nowe wiersze, należy zaznaczyć nazwę tabeli ETAT i kliknąć w przycisk „View Data” („Wszystkie Wiersze”) znajdujący się w górnej części okna.



W dolnej części okna pojawi się tabela, w której należy wpisać dane zgodnie z poniższą tabelą.

Wyjście Danych		Plan zapytania	Komunikaty	Notifications
	ID_ETAT [PK] integer	NAZWA character varying (20)	PLACA_MIN real	PLACA_MAX real
1	1	Stażysta	800	1000
2	2	Sekretarka	900	1200
3	3	Asystent	1000	1600
4	4	Adiunkt	1600	2000
5	5	Profesor	2000	2500
6	6	Dyrektor	2500	3200

Po wpisaniu wszystkich danych należy zapisać zmiany (przycisk „Save Data Changes” powyżej tabelki).

ETAT

ID_ETAT	NAZWA	PLACA_MIN	PLACA_MAX
1	Stażysta	800	1000
2	Sekretarka	900	1200
3	Asystent	1000	1600
4	Adiunkt	1600	2000
5	Profesor	2000	2500
6	Dyrektor	2500	3200

Zad. 2. W analogiczny sposób wpisać dane do tabeli ZESPOL i PRACOWNIK.

ZESPOL

ID_ZESP	NAZWA	ADRES
10	administracja	Piotrowo 3a
20	bazy danych	Wieżowa 75
30	sieci komputerowe	Garbary 3
40	systemy operacyjne	Piotrowo 3a
50	translatory	Mansfelda 4

PRACOWNIK

NUMER	NAZWISKO	ETAT	SZEF	PRACUJE_OD	PLACA_POD	PLACA_DOD	ID_ZESP
1000	Lech	6	NULL	1971-01-01	3160	570	10
1080	Koliberek	2	1000	1983-02-20	1150	0	10
1010	Podgajny	5	1000	1975-05-01	2180	420	20
1040	Rus	4	1010	1979-08-15	1750	0	20
1070	Muszyński	4	1010	1985-05-01	1600	0	20
1060	Misiecki	3	1010	1985-03-01	1400	0	20
1090	Palusz	3	1040	1989-08-15	1200	0	20
1020	Delcki	5	1000	1977-08-01	2050	270	30
1030	Maleja	4	1020	1968-07-01	1750	0	30
1100	Warski	3	1030	1987-07-15	1350	0	30
1110	Rajski	1	1030	1990-07-01	900	0	30
1050	Lubicz	4	1000	1983-08-01	1780	0	40
1120	Orka	3	1050	1988-04-01	1350	0	40
1130	Kolski	1	1050	1991-08-01	900	0	40

6. Sortowanie wyświetlonych danych

Aby posortować wyświetlone dane tabeli należy w oknie wstawiania danych kliknąć w przycisk „Filter” znajdujący się w górnej części okna lub wcisnąć kombinację klawiszy Alt+F.

Pojawi się okno, które umożliwi posortowanie danych według wybranej kolumny (kolumn). ASC oznacza sortowanie w kolejności rosnącej, natomiast DESC – w kolejności malejącej.

Sort/Filter options

Filter (accesskey + F)

SQL Filter 1

Data Sorting

Kolumna	Order
ID_ETAT	ASC

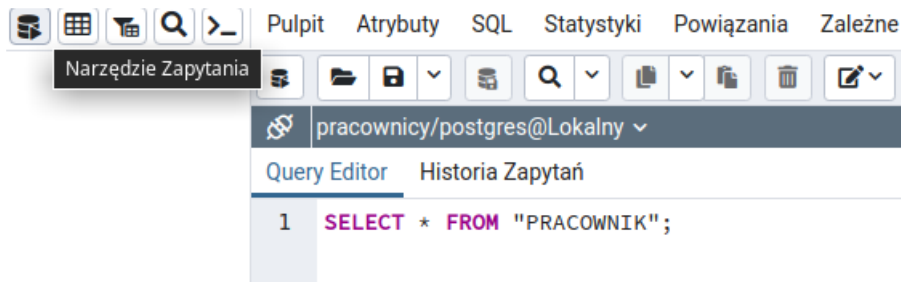
? Anuluj OK

Zad. 3. Wyświetlić dane z tabeli PRACOWNIK posortowane po numerze etatu i (w drugiej kolejności) nazwisku.

7. Zaawansowane wybieranie danych za pomocą języka SQL

Uwaga! Baza PostgreSQL jest czuła na wielkość liter używanych w identyfikatorach (nazwy tabel, kolumn, itp.). W przypadku gdy nazwa została zapisana z dużymi literami w zapytaniu SQL musi być objęta znakami cudzysłowów.

Edytor języka SQL można uruchomić klikając w przycisk „Narzędzie zapytania” znajdujący się w górnej części okna.

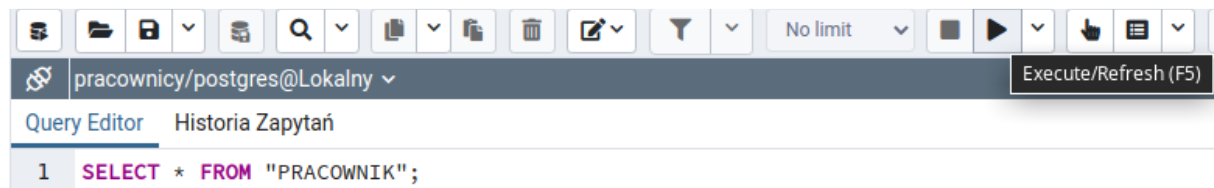


Zad. 4. Wyświetl wszystkie atrybuty i wszystkie wiersze tabeli PRACOWNIK.

Zapytanie SQL rozwiązujące zadany problem jest następujące:

```
SELECT * FROM "PRACOWNIK";
```

Po wpisaniu zapytania należy kliknąć w przycisk „Execute/Refresh” znajdujący się w górnej części ekranu lub wcisnąć klawisz F5. Wynik zapytania pojawi się w formie tabeli w dolnej części okna.



W przypadku baz danych typu PostgreSQL nazwy tabel i atrybutów w zapytaniach zapisujemy w cudzysłowie. Gdybyśmy pominęli cudzysłów, to duże litery zostałyby automatycznie zamienione na małe. Byłoby to problemem, ponieważ SQL rozróżnia wielkość liter w nazwach.

Zad. 5. Wyświetl wszystkie atrybuty i wszystkie wiersze tabeli ETAT.

Do samodzielnego wykonania.

Zad. 6. Wyświetl wszystkie atrybuty i wszystkie wiersze tabeli ZESPOL.

Do samodzielnego wykonania.

Zad. 7. Wyświetl wszystkie atrybuty pracowników, których pensja podstawowa jest większa od 1400.

```
SELECT * FROM "PRACOWNIK"  
WHERE "PRACOWNIK"."PLACA_POD" > 1400;
```

Zad. 8. Wyświetlić wszystkie atrybuty pracowników, których pensja podstawowa mieści się w zamkniętym przedziale [1000;2000].

```
SELECT * FROM "PRACOWNIK"  
WHERE "PRACOWNIK"."PLACA_POD" >= 1000  
AND "PRACOWNIK"."PLACA_POD" <= 2000;
```

lub

```
SELECT * FROM "PRACOWNIK"  
WHERE "PRACOWNIK"."PLACA_POD" BETWEEN 1000 AND 2000;
```

Operator BETWEEN używamy do przedziałów zamkniętych. Gdyby przedział był otwarty (1000;2000), to zapytanie wyglądałoby tak:

```
SELECT * FROM "PRACOWNIK"  
WHERE "PRACOWNIK"."PLACA_POD" > 1000  
AND "PRACOWNIK"."PLACA_POD" < 2000;
```

Zad. 9. Wyświetlić nazwisko i nazwę etatu każdego z pracowników.

```
SELECT "PRACOWNIK"."NAZWISKO", "ETAT"."NAZWA"  
FROM "PRACOWNIK", "ETAT"  
WHERE "PRACOWNIK"."ETAT" = "ETAT"."ID_ETAT";
```

Warunek w tym zadaniu polega na wskazaniu w jaki sposób tabele (z których wybieramy dane) są połączone. Taki warunek ma zawsze następującą formę: <klucz obcy> = <odpowiadający mu klucz główny>. W naszym zapytaniu wskazujemy, że klucz obcy ETAT z tabeli PRACOWNIK odnosi się do klucza głównego ID_ETAT z tabeli ETAT. Jeśli wybieramy dane z więcej niż dwóch tabel, to należy określić w ten sposób wszystkie połączenia między nimi.

Zad. 10. Wyświetlić nazwisko, nazwę etatu, nazwę zespołu i płacę podstawową pracowników.

Do samodzielnego wykonania.

Zad. 11. Dodać do tabeli PRACOWNIK nową osobę opisaną następującymi atrybutami:

NUMER	NAZWISKO	ETAT	SZEF	PRACUJE_OD	PLACA_POD	PLACA_DOD	ID_ZESP
1210	Lubomirski	1	1030	now()	1200	NULL	30

```
INSERT INTO "PRACOWNIK" ("NUMER", "NAZWISKO", "ETAT", "SZEF",  
"PRACUJE_OD", "PLACA_POD", "ID_ZESP")  
VALUES (1210, 'Lubomirski', 1, 1030, now(), 1200, 30);
```

Zapytania wstawiające, modyfikujące i usuwające dane nie pokazują wyników w formie zaktualizowanych tabel. Aby sprawdzić, czy takie zapytanie zakończyło się sukcesem należy wyświetlić zmienioną tabelę za pomocą zapytania SELECT lub ręcznie (przycisk „View data”/„Wszystkie Wiersze”).

Zad. 12. Obniż pracownikowi o numerze 1210 płacę podstawową o 10%.

```
UPDATE "PRACOWNIK"  
SET "PLACA_POD" = "PLACA_POD" - 0.1*"PLACA_POD"  
WHERE "NUMER" = 1210;
```

Zad. 13. Usuń pracownika(-ów) o nazwisku 'Kolski'.

```
DELETE FROM "PRACOWNIK"  
WHERE "NAZWISKO" = 'Kolski';
```

Zad. 14. Wyświetl nazwiska wszystkich pracowników uszeregowane rosnąco według kolejności alfabetycznej.

```
SELECT "PRACOWNIK"."NAZWISKO"  
FROM "PRACOWNIK"  
ORDER BY "PRACOWNIK"."NAZWISKO" ASC;
```

Tak samo można szeregować według kolejności liczbowej i dat.

Aby uszeregować w kolejności malejącej należy wpisać na końcu DESC zamiast ASC. Domyślna jest kolejność rosnąca. Oznacza to, że możemy pominąć słowo ASC, jeśli chcemy sortować rosnąco.

Zad. 15. Wyświetl nazwisko, nazwę etatu, płacę podstawową i datę zatrudnienia wszystkich adiunktów i profesorów uszeregowane po nazwie etatu i dacie zatrudnienia malejąco, których płaca podstawowa jest większa od 1600.

Do samodzielnego wykonania.

Podpowiedź 1: Nazwy etatów są zapisane w tabeli ETAT z dużej litery (Adiunkt, Profesor itp.).

Podpowiedź 2: Warunki dotyczące nazw etatów należy zapisać w nawiasie (okrągłym), aby zostały sprawdzone w pierwszej kolejności.

Zad. 16. Wyświetl średnią płacę dodatkową pracowników.

```
SELECT avg("PLACA_DOD") FROM "PRACOWNIK";
```

avg jest funkcją obliczającą średnią arytmetyczną wartości podanego atrybutu w wybranych wierszach.

Zad. 17. Wypróbuj wymienione poniżej funkcje grupujące wpisując je zamiast avg w zapytaniu z zad. 16.

min – najmniejsza wartość;

max – największa wartość;

sum – suma wszystkich wartości;

count – liczba wszystkich wartości (nie licząc wierszy, w których wybrany atrybut przyjmuje wartość NULL).

Do samodzielnego wykonania.

Zad. 18. Bazując na zadaniu 11, dodaj do tabeli ZESPOL nowy zespół opisany następującymi atrybutami:

ID_ZESP	NAZWA	ADRES
60	Uczenie maszynowe	Wincentego Pola 2

Do samodzielnego wykonania.

Zad. 19. Bazując na zadaniu 12, zmień płacę maksymalną etatu Adiunkt tak, aby była równa 130% płacy minimalnej.

Do samodzielnego wykonania.

Zad. 20. Bazując na zadaniu 13, usuń pracowników których płaca podstawowa jest mniejsza od 1500 oraz płaca dodatkowa jest równa 0.

Do samodzielnego wykonania.

Zad. 21. Wyświetl średnią płacę podstawową dla każdego etatu z osobna (wraz z nazwami etatów).

```
SELECT "ETAT"."NAZWA", avg("PRACOWNIK"."PLACA_POD")
FROM "PRACOWNIK", "ETAT"
WHERE "PRACOWNIK"."ETAT" = "ETAT"."ID_ETAT"
GROUP BY "ETAT"."NAZWA";
```

Klauzulę GROUP BY należy użyć zawsze wtedy, gdy wybieramy jakąś kolumnę danych wraz z funkcją grupującą. Musimy określić względem której kolumny należy pogrupować wyświetlane informacje.

Zad. 22. Bazując na zad. 21 wyświetl liczbę pracowników pracujących na każdym etacie z osobna (wraz z nazwami etatów). Wyniki powinny być posortowane według liczby zatrudnionych rosnąco.

Do samodzielnego wykonania.

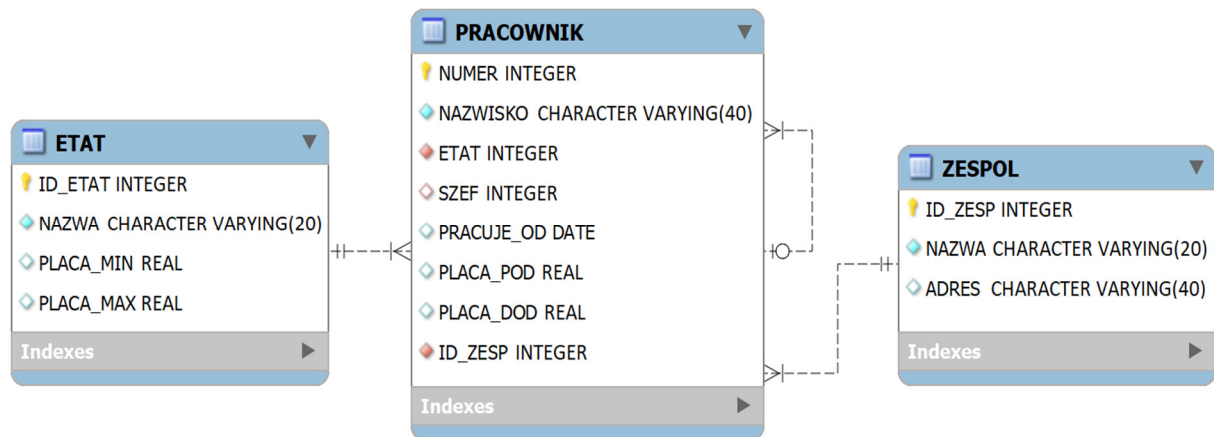
Zad. 23. Bazując na zad. 21 wyświetl całkowitą pensję (łącznie podstawową i dodatkową) pracowników każdego zespołu z osobna (wraz z nazwami zespołu).

Do samodzielnego wykonania.

8. Materiały pomocnicze

Poniżej znajduje się **Diagram związków encji** (ang. entity relationship diagram, ERD) bazy danych tworzonej w niniejszej instrukcji.

Klucze główne są zaznaczone symbolem złotego klucza, klucze obce są zaznaczone czerwonym kwadratem, natomiast pozostałe atrybuty niebieskim kwadratem. Kwadraty wypełnione oznaczają atrybut obowiązkowy (taki, który musi mieć wpisaną konkretną wartość; nie może być NULL), natomiast kwadraty niewypełnione oznaczają atrybuty opcjonalne.



Operatory języka SQL

Porównania:

- > większe
- < mniejsze
- >= większe lub równe
- <= mniejsze lub równe
- = równe
- != różne
- <> różne
- BETWEEN przynależność do przedziału

Arytmetyczne:

- + suma
- - różnica
- * iloczyn
- / iloraz
- % reszta z dzielenia

Logiczne:

- AND „i”
- OR „lub”
- NOT „nieprawda, że”