

# ZAPYTANIA REKURENCYJNE Z WYKORZYSTANIEM WYRAŻEŃ CTE W SQL SERVER

*Andrzej Ptasznik*

Warszawska Wyższa Szkoła Informatyki

aptaszni@wwsi.edu.pl

## 1. Zagadnienia programowe

Rekurencja pojawia się na lekcjach matematyki i informatyki (patrz rozdz. w tym opracowaniu), a bazy danych i język SQL – na lekcjach informatyki, zgodnie z poniższymi zapisami w podstawie programowej.

### **Informatyka, IV etap edukacyjny, zakres podstawowy**

2. Wyszukiwanie, gromadzenie, selekcjonowanie, przetwarzanie i wykorzystywanie informacji, współtworzenie zasobów w sieci, korzystanie z różnych źródeł i sposobów zdobywania informacji. Uczeń:
  - 1) znajduje dokumenty i informacje w udostępnianych w Internecie bazach danych (np. bibliotecznych, statystycznych, w sklepach internetowych), ocenia ich przydatność i wiarygodność i gromadzi je na potrzeby realizowanych projektów z różnych dziedzin;
4. Opracowywanie informacji za pomocą komputera, w tym rysunków, tekstów, danych liczbowych, animacji, prezentacji multimedialnych i filmów. Uczeń:
  - 6) tworzy bazę danych, posługuje się formularzami, porządkuje dane, wyszukuje informacje stosując filtrowanie;
  - 7) wykonuje podstawowe operacje modyfikowania, i wyszukiwania informacji na relacyjnej bazie danych;

### **Informatyka, IV etap edukacyjny, zakres rozszerzony**

2. Wyszukiwanie, gromadzenie, selekcjonowanie, przetwarzanie i wykorzystywanie informacji, współtworzenie zasobów w sieci, korzystanie z różnych źródeł i sposobów zdobywania informacji. Uczeń:
  - 1) projektuje relacyjną bazę danych z zapewnieniem integralności danych;
  - 2) stosuje metody wyszukiwania i przetwarzania informacji w relacyjnej bazie danych (język SQL);
  - 3) tworzy aplikację bazodanową, w tym sieciową, wykorzystującą język zapytań, kwerendy, raporty; zapewnia integralność danych na poziomie pól, tabel, relacji;
  - 5) opisuje mechanizmy związane z bezpieczeństwem danych: szyfrowanie, klucz, certyfikat, zaporą ogniową.

*Uwaga.* Przedstawiony scenariusz zajęć wybiega poza zagadnienia programowe i powinien być adresowany do uczniów z lepszym przygotowaniem informatycznym i wykazujących większe zainteresowania pogłębieniem wiedzy z zakresu baz danych.

## 2. Temat zajęć

### Zapytania rekurencyjne z wykorzystaniem wyrażeń CTE w SQL Server

Temat jest wprowadzeniem do stosowania wyrażeń CTE (ang. *Common Table Expressions*) w poleceniach języka SQL oraz użycia tych wyrażeń do zapytań rekurencyjnych.

*Komentarz.* Jest to trudny temat, ponieważ łączy w sobie problemy wyszukiwania danych w relacyjnej bazie danych z myśleniem rekurencyjnym.

## 3. Cele zajęć

W wyniku tych zajęć uczeń powinien umieć:

- tworzyć zapytania w języku SQL z wykorzystaniem wyrażeń CTE,
- pisać zapytania rekurencyjne.

*Komentarz.* Cele tych zajęć wykraczają poza podstawę programową.

## 4. Przygotowanie uczniów

Uczniowie przystępujący do tych zajęć powinni:

- wcześniej poznać metodę rozwiązywania problemów z pomocą komputera, składającą się z sześciu etapów;
- znać podstawowe pojęcia z zakresu informatyki i baz danych;
- umieć pisać proste zapytania w języku SQL;
- znać pojęcie rekurencji.

## 5. Metody pracy

W zajęciach są stosowane następujące metody pracy:

- generalnie, rozwiązywanie każdego rozważanego zagadnienia (problemu) składa się z sześciu etapów, które składają się na metodę rozwiązywania problemów z pomocą komputera; te etapy, to:
  - opis, dyskusja i zrozumienie sytuacji problemowej,
  - podanie specyfikacji problemów do rozwiązania,
  - zaprojektowanie rozwiązania,
  - implementacja (realizacja) rozwiązania w postaci polecenia w języku SQL,
  - testowanie i ewaluacja rozwiązania,
  - prezentacja sposobu otrzymania rozwiązania i samego rozwiązania.
- przygotowanie przez uczniów listy problemów związanych z realizacją zadania;
- samodzielne sporządzenie przez uczniów opisów sposobów rozwiązania poszczególnych problemów;
- zapisanie odpowiednich poleceń w języku SQL i ich testowanie;
- samodzielne testowanie poprawności zdefiniowanych zapisanych poleceń;
- prezentacja otrzymanych rozwiązań.



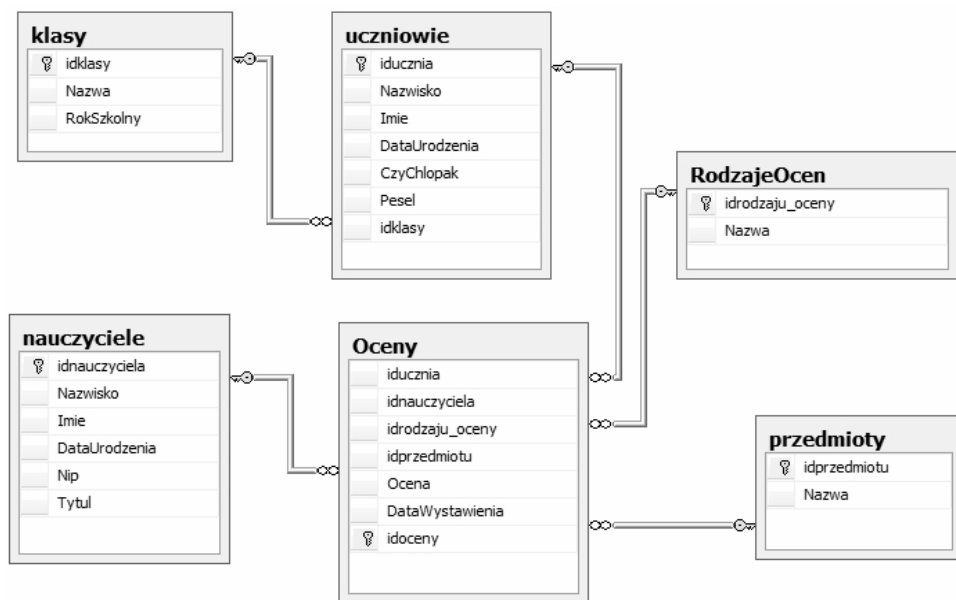
## 6. Formy pracy

Założone cele są realizowane za pomocą następujących form pracy:

- podczas burzy mózgów prowadzonej przez całą klasę lub w grupach uczniów – ma to doprowadzić do sformułowania sposobu rozwiązania problemu;
- przygotowywanie opisów rozwiązania może odbywać się indywidualnie lub w parach uczniów;
- polecenia SQL uczniowie piszą samodzielnie;
- testowanie poleceń może odbywać się w grupach uczniów lub samodzielnie (w zależności od stopnia zaawansowania);
- końcowym efektem pracy nad danym problemem jest prezentacja napisanego polecenia oraz wyników jego działania.

## 7. Materiały pomocnicze

Do przeprowadzenia zajęć niezbędne jest zainstalowanie oprogramowania SQL Server 2008 R2 Express Edition (darmowe oprogramowanie możliwe do pobrania ze stron firmy Microsoft). Po zainstalowaniu oprogramowania serwera baz danych należy utworzyć przykładową bazę danych a w niej zdefiniować tabele według schematu pokazanego na rysunkach 1 i 2.



Rysunek 1. Schemat bazy danych ElektronicznyDziennikOcen

## 8. Środki dydaktyczne

Uczniowie wykorzystują w czasie tych zajęć:

- podręcznik, np.: E. Gurbiel, G. Hardt-Olejniczak, E. Kołczyk, H. Krupicka, M. M. Sysło, *Informatyka, zakres rozszerzony, WSiP, Warszawa 2004.*

- materiały dotyczące baz danych, opracowane w projekcie Informatyka +;
- komputer i jego podstawowe oprogramowanie, w tym oprogramowanie MS SQL2008 R2 Express Edition oraz MS SQL Server Management Studio.

*Komentarz.* Oprogramowanie MS SQL Server 2008 R2 Express Edition jest darmowe; można pobrać ze strony <http://www.microsoft.com/express/Database>.



Rysunek 2. Schemat bazy danych Pracownicy

## 9. Przebiegu zajęć (kolejnych lekcji)

Zajęcia, których celem jest wykorzystanie wyrażeń CTE do realizacji zapytań rekurencyjnych mogą być rozłożone na kilka lekcji, niekoniecznie kolejne. W niniejszej propozycji przyjęliśmy następujący tryb realizacji omawianego tematu:

- na początku, nauczyciel objaśnia składnię i zastosowanie wyrażeń CTE (Lekcja 1);
- na kolejnym etapie są realizowane złożone zapytania z wykorzystaniem wyrażeń CTE (Lekcja 2);
- następnie uczniowie zajmują się problemami, które mają naturę rekurencyjną w tym sensie, że ich sformułowania i wynikające z nich algorytmy są rekurencyjne (Lekcja 3);
- rozwiązywanie każdego z zagadnień składa się z sześciu etapów, wyżej wymienionych (w Metodach pracy).

### Lekcja 1. Wyrażenia CTE w języku SQL. Czas: 30 min.

Wyrażenia **CTE** (ang. *Common Table Expression*) są nowym elementem języka SQL. Podstawowym celem wprowadzenia wyrażeń CTE było umożliwienie pisania zapytań rekurencyjnych. Dodatkowym zastosowaniem tych wyrażeń jest tworzenie nazwanych zbiorów danych, które mogą być wykorzystywane do realizacji jednego z poleceń manipulacji danymi (SELECT, INSERT, UPDATE, DELETE,

MERGE). Nazwane zbiory są tworzone dla konkretnego polecenia i przestają istnieć po jego wykonaniu. Podstawowa składnia tworzenia wyrażeń CTE jest następująca:

```
WITH Nazwa_zbioru AS
(
  Polecenie_Select
)
Polecenie SQL
```

Wykorzystanie wyrażeń CTE można wyjaśnić na prostych przykładach.

**Pierwszy przykład:** Na podstawie bazy danych ElektronocnyDziennikOcen wybrać uczniów, którzy w rankingu według średniej ocen zajmują pozycje od 3 do 8.

*Uwaga.* W rozwiązaniu tego przykładowego zadania należy wykorzystać funkcję szeregującą Row\_Number(). Problem można rozwiązać za pomocą następującego polecenia:

```
WITH RankingSrednich AS
(
  SELECT ROW_NUMBER() OVER (ORDER BY AVG(ocena) DESC ) as Lp,
     Nazwisko, Imie, Pesel, AVG(ocena) as Srednia
  FROM Uczniowie JOIN Oceny
     ON Uczniowie.iducznia=Oceny.iducznia
  GROUP BY Nazwisko, Imie, Pesel
)
SELECT Nazwisko, Imie, Pesel, Srednia
FROM RankingSrednich
WHERE Lp BETWEEN 3 and 8
```

Wyjaśniając istotę wyrażeń CTE należy w pierwszym kroku pokazać wynik zapytania wykorzystanego do tworzenia nazwanego zbioru danych (RankingSrednich).

**Drugi przykład:** Na podstawie bazy danych ElektronicznyDziennikOcen wybrać tych uczniów, których średnia ocena jest wyższa od średniej oceny jego klasy.

W rozwiązaniu tego problemu utworzymy dwa zbiory (średnie uczniów i średnie klas), a następnie połączymy te zbiory i wybierzemy te wiersze, dla których spełniony jest założony warunek. Problem można rozwiązać za pomocą następującego polecenia;

```
WITH SrednieKlas AS
(
  SELECT idklasy, AVG(ocena) AS SredniaKlasy
  FROM Uczniowie JOIN Oceny ON Uczniowie.iducznia=Oceny.iducznia
  GROUP BY idklasy
), SrednieUczniow AS
(
  SELECT Nazwisko, Imie, Pesel, idklasy, AVG(ocena) AS SredniaUcznia
  FROM Uczniowie JOIN Oceny ON Uczniowie.iducznia=Oceny.iducznia
  GROUP BY Nazwisko, Imie, Pesel, idklasy
)
```

```
SELECT Nazwisko, Imie, Pesel
FROM SrednieKlas JOIN SrednieUczniow ON SrednieKlas join SrednieUczniow
WHERE SredniaUcznia>SredniaKlasy
```

W omawianym przykładzie zdefiniowaliśmy dwa nazwane zbiory a następnie umieściliśmy zapytanie, które z tych zbiorów korzysta. Na zakończenie zajęć należy omówić z uczniami sposoby wykorzystania wyrażeń CTE zwracając uwagę na fakt, że dla jednego polecenia w języku SQL możemy zdefiniować wiele nazwanych zbiorów, dzięki czemu ostateczna postać polecenia jest prosta i oddaje istotę realizowanego zadania.

## Lekcja 2. Wykorzystanie wyrażeń CTE do złożonych zapytań. Czas: 60 min.

Celem tej lekcji jest zilustrowanie, w jaki sposób wyrażenia CTE można wykorzystać do pisania poleceń realizujących złożone wybieranie danych. Definiowanie nazwanych zbiorów danych powinno ułatwić pisanie takich poleceń. W trakcie lekcji należy zwracać szczególną uwagę na prostą postać ostatecznego polecenia, uzyskaną dzięki wcześniejszemu zdefiniowaniu nazwanych zbiorów, które są częścią rozwiązania.

**Pierwszy przykład:** Na podstawie bazy danych ElektronicznyDziennikOcen wybrać tych uczniów, których średnia ocena z fizyki jest wyższa od ich średniej oceny z matematyki.

Przykład można rozwiązać za pomocą następującego polecenia;

```
WITH SrednieZFizyki AS
(
    SELECT Nazwisko, Imie, Pesel, Iducznia, AVG(ocena) AS SredniaFizyka
    FROM Uczniowie JOIN Oceny ON Uczniowie.iducznia=Oceny.iducznia
        JOIN Przedmioty ON Oceny.idprzedmiotu=Przedmioty.idprzedmiotu
    WHERE Przedmioty.Nazwa='Fizyka'
    GROUP BY iducznia Nazwisko, Imie, Pesel,iducznia
),
SrednieZMatematyki AS
(
    SELECT Nazwisko, Imie, Pesel, iducznia, AVG(ocena) AS SredniaFizyka
    FROM Uczniowie JOIN Oceny ON Uczniowie.iducznia=Oceny.iducznia
        JOIN Przedmioty ON Oceny.idprzedmiotu=Przedmioty.idprzedmiotu
    WHERE Przedmioty.Nazwa='Matematyka'
    GROUP BY Nazwisko, Imie, Pesel, iducznia
)
SELECT F.Nazwisko, F.Imie, F.Pesel
FROM SrednieZFizyki as F JOIN SrednieZMatematyki as M
    ON F.iducznia=M.iducznia
WHERE SredniaFizyka>SredniaMatematyka
```

Przy omawianiu tego przykładu należy zwrócić uwagę na fakt, że ostateczna postać polecenia jest prosta, natomiast wcześniej zdefiniowano nazwane zbiory, do których się odwołujemy.

Dla utrwalenia zasad stosowania wyrażeń CTE można formułować kolejne problemy, w rozwiązaniach których będzie można wykorzystać mechanizm wyrażeń CTE.



**Drugi przykład:** Na podstawie bazy danych ElektronicznyDziennikOcen wybrać uczniów, których średnia ocena z fizyki jest większa od 4.00 za wyjątkiem tych, których średnia ocena z matematyki jest niższa od 3.50

```
WITH SrednieZFizyki AS
(
  SELECT Nazwisko, Imie, Pesel, Iducznia, AVG(ocena) AS SredniaFizyka
  FROM Uczniowie JOIN Oceny ON Uczniowie.iducznia=Oceny.iducznia
      JOIN Przedmioty ON Oceny.idprzedmiotu=Przedmioty.idprzedmiotu
  WHERE Przedmioty.Nazwa='Fizyka'
  GROUP BY iducznia Nazwisko, Imie, Pesel,iducznia
  HAVING AVG(ocena)>4.00
),
SrednieZMatematyki AS
(
  SELECT Nazwisko, Imie, Pesel, iducznia, AVG(ocena) AS SredniaFizyka
  FROM Uczniowie JOIN Oceny ON Uczniowie.iducznia=Oceny.iducznia
      JOIN Przedmioty ON Oceny.idprzedmiotu=Przedmioty.idprzedmiotu
  WHERE Przedmioty.Nazwa='Matematyka'
  GROUP BY Nazwisko, Imie, Pesel, iducznia
  HAVING AVG(ocena) <3.50
)
SELECT Nazwisko, Imie, Pesel
FROM SrednieZFizyki
EXCEPT
SELECT Nazwisko, Imie, Pesel
FROM SrednieZMatematyki
```

W omawianym przykładzie dodatkowo należy zwrócić uwagę na wykorzystanie klauzuli HAVING do filtrowania zapytania w stosunku do wyniku działania funkcji agregującej oraz wykorzystanie operatora EXCEPT, który zwraca różnicę dwóch zbiorów (zbiory przygotowane zapytaniem). Zmieniając treść zadania można pokazać działanie operatora UNION (suma zbiorów) oraz INTERSECT (część wspólna zbiorów).

### Lekcja 3. Zapytania rekurencyjne z wykorzystaniem wyrażeń CTE.

**Czas: 90 min.**

Dla odwzorowania hierarchii w relacyjnych bazach danych stosuje się klucze obce, które odwołują się do innego wiersza w tej samej tabeli. Przykład takiego rozwiązania jest pokazany na rysunku 2. W tabeli Pracownicy jest umieszczony klucz obcy o nazwie idprzełożonego, który wskazuje na inny wiersz w tej samej tabeli (przełożony jest też pracownikiem i ma swojego przełożonego). Takie konstrukcje tabel stwarzają problemy przy zapytaniach, których istotą jest wybranie danych powiązanych z konkretnych wierszem na wszystkich poziomach hierarchii (wszyscy podwładni konkretnej osoby na wszystkich poziomach podległości). Ponieważ problem przeglądania hierarchii jest w swej istocie problemem rekurencyjnym, to jego rozwiązanie wymaga również rekurencyjnego podejścia. Klasyczne polecenie SELECT języka SQL nie umożliwi realizacji zapytań rekurencyjnych. Wyrażenia

CTE mogą być użyte do zdefiniowania takiego zbioru nazwanego, który posłuży do rozwiązania zadań rekurencyjnych.

**Przykład pierwszy:** wygenerowanie tabeli z kolumną będącą sekwencją kolejnych liczb całkowitych w zakresie od 0 do 10.

Polecenie realizujące ten przykład może mieć następującą postać;

```
WITH Sekwencja AS
(
SELECT 0 AS liczba
UNION ALL
SELECT Liczba+1
FROM Sekwencja
WHERE liczba<10
)
SELECT Liczba FROM Sekwencja
```

Liczba
0
1
2
3
4
5
6
7
8
9
10

Rysunek 3. Wynik zapytania generującego sekwencję liczb

Na tym prostym przykładzie można omówić istotę wyrażania rekurencji w zapytaniach wykorzystujących wyrażenia CTE. Polecenie tworzące sekwencję liczb można opisać jako następujące czynności:

- definicja nazwanego zbioru Sekwencja(wyrażenie CTE) – WITH Sekwencja AS;
- zapytanie tworzące ten zbiór składa się z dwóch zapytań połączonych operatorem UNION ALL (dodaje do siebie zbiory bez eliminowania powtarzających się wartości):
  - pierwsze zapytanie (element wyjściowy) pobiera dane, od których rozpoczynamy rekurencję – SELECT 0 AS liczba – tworzy jeden wiersz w tabeli zawierającej jedną kolumnę z wartością 0 (zero);
  - drugie zapytanie – SELECT Liczba+1 FROM Sekwencja WHERE liczba<10 – (element rekurencyjny) pobiera dane ze zbioru Sekwencja zwiększając odczytaną kolumnę Liczba o jeden a operator UNION ALL dodaje ten wynik do wcześniej wybranych wierszy w zbiorze Sekwencja (w zbiorze Sekwencja znajdują się wiersze ostatnio wybrane);
  - po zdefiniowaniu wyrażenia CTE tworzymy zapytanie pobierające wyniki – SELECT Liczba FROM Sekwencja.

Na tym etapie należy poinformować uczniów, że SQL Server ma domyślne ograniczenie dla liczby wywołań rekurencyjnych wynoszące 100, ale można tę ilość zmienić używając opcji MAXRECURSION.



W naszym przykładzie, gdybyśmy chcieli wygenerować sekwencję 500 kolejnych liczb całkowitych – polecenie miałooby następującą postać:

```
WITH Sekwencja AS
(
SELECT 0 AS Liczba
UNION ALL
SELECT Liczba+1
FROM Sekwencja
WHERE Liczba<10
)
SELECT Liczba FROM Sekwencja
OPTION (MAXRECURSION 500)
```

**Przykład drugi:** wybranie wszystkich podwładnych (na dowolnym poziomie) konkretnego pracownika.

Odwołując się do tabel pokazanych na rysunku 2 możemy przystąpić do realizacji zadania. Dla lepszego zrozumienia przykładu przyjmijmy, że tabele Pracownicy i Funkcje mają zawartość jak na rysunku 4 i 5.

IdPracownika	Nazwisko	Imie	Pesel	IdPrzełożonego	IdFunkcji
1	Kot	Jan	87090812321	1	1
2	Pies	Janina	81110987654	1	2
3	Okoń	Stanisław	76120785609	1	2
4	Płotka	Zofia	59122188723	2	3
5	Leszcz	Piotr	76120976781	2	3
6	Szczupak	Maria	82061209834	4	4
7	Lin	Karol	87062187654	4	4
8	Myszka	Szara	44091187654	6	5
9	Szczurek	Paweł	69121189876	6	5
10	Królik	Violetta	90111298723	8	6

Rysunek 4. Przykładowa zawartość tabeli Pracownicy

idfunkcji	Nazwa
1	Prezes
2	Dyrektor Departamentu
3	Dyrektor Działu
4	Kierownik Zespołu
5	Referent
6	Asystent

Rysunek 5. Przykładowa zawartość tabeli Funkcje

Polecenie, które dla przykładowych danych zwróci dane wszystkich podwładnych (na wszystkich poziomach) pracownika o IdPracownika=1 może mieć następującą postać;

```
WITH Podwładni AS
(
SELECT Imie+' '+Nazwisko as Pracownik,
```

```

0 as Poziom,
idpracownika
FROM Pracownicy
WHERE IdPracownika=1
UNION ALL
SELECT Pr.Imie+''+Pr.Nazwisko as Pracownik ,
       Po.Poziom+1 as Poziom,
       Pr.idpracownika
FROM Pracownicy Pr JOIN Podwladni Po
ON Pr.IdPrzelozonego=Po.idpracownika
WHERE Pr.idPrzelozonego !=Pr.IdPracownika
)
SELECT * FROM Podwladni
    
```

Wynik tego zapytania dla przykładowych danych pokazanych na rysunkach 4 i 5 jest przedstawiony na rysunku 6.

Pracownik	Poziom	idpracownika
Jan Kot	0	1
Janina Pies	1	2
Stanisław Okoń	1	3
Zofia Płotka	2	4
Piotr Leszcz	2	5
Maria Szczupak	3	6
Karol Lin	3	7
Szara Myszka	4	8
Paweł Szczurek	4	9
Violetta Królik	5	10

Rysunek 6. Wynik zapytania

W omawianym zapytaniu utworzyliśmy kolumnę o nazwie Poziom, która określa „odległość” w hierarchii danego pracownika od tego, który jest początkowym elementem hierarchii.

Uczniom można zaproponować modyfikację tego zapytania, aby otrzymać wyniki pokazane na rysunku 7.

Pracownik	Funkcja	Poziom	idpracownika	ścieżka
Jan Kot	Prezes	0	1	
Janina Pies	Dyrektor Departamentu	1	2	Prezes-Jan Kot/
Stanisław Okoń	Dyrektor Departamentu	1	3	Prezes-Jan Kot/
Zofia Płotka	Dyrektor Działu	2	4	Prezes-Jan Kot./Dyrektor Departamentu-Janina Pies/
Piotr Leszcz	Dyrektor Działu	2	5	Prezes-Jan Kot./Dyrektor Departamentu-Janina Pies/
Maria Szczupak	Kierownik Zespołu	3	6	Prezes-Jan Kot./Dyrektor Departamentu-Janina Pies./Dyrektor Działu-Zofia Płotka/
Karol Lin	Kierownik Zespołu	3	7	Prezes-Jan Kot./Dyrektor Departamentu-Janina Pies./Dyrektor Działu-Zofia Płotka/
Szara Myszka	Referent	4	8	Prezes-Jan Kot./Dyrektor Departamentu-Janina Pies./Dyrektor Działu-Zofia Płotka./Kierownik Zespołu-Maria Szczupak/
Paweł Szczurek	Referent	4	9	Prezes-Jan Kot./Dyrektor Departamentu-Janina Pies./Dyrektor Działu-Zofia Płotka./Kierownik Zespołu-Maria Szczupak/
Violetta Królik	Asystent	5	10	Prezes-Jan Kot./Dyrektor Departamentu-Janina Pies./Dyrektor Działu-Zofia Płotka./Kierownik Zespołu-Maria Szczupak./Referent-Szara Myszka/

Rysunek 7. Wyniki zapytania

W wynikach zapytania na rysunku 7 jest utworzona kolumna, która opisuje hierarchie podwładności dla każdego pracownika.