

Jednostka 8 - Paradygmat deklaratywny

Joanna Dagil

Grupa TCH-1

26 listopada 2025

1 Cel ćwiczenia

Celem ćwiczenia jest zapoznanie się z paradygmatem deklaratywnym i porównanie go z programowaniem imperatywnym i funkcyjnym. Wykorzystujemy do tego język SQL i Prolog.

2 Zadania

2.1 Zadanie 1

Tworzenie struktury tabeli.

```
1 CREATE DATABASE IF NOT EXISTS usos;
2 USE usos;
3
4 CREATE TABLE IF NOT EXISTS Studenci (
5     id INT PRIMARY KEY AUTO_INCREMENT,
6     imie VARCHAR(100),
7     nazwisko VARCHAR(100),
8     rok INT,
9     kierunek VARCHAR(100)
10 );
11
12 CREATE TABLE IF NOT EXISTS Przedmioty (
13     id INT PRIMARY KEY AUTO_INCREMENT,
14     nazwa VARCHAR(100) NOT NULL
15 );
16
17 CREATE TABLE IF NOT EXISTS Oceny (
18     id INT PRIMARY KEY AUTO_INCREMENT,
19     student_id INT NOT NULL,
20     przedmiot_id INT NOT NULL,
21     FOREIGN KEY (student_id) REFERENCES Studenci(id),
22     FOREIGN KEY (przedmiot_id) REFERENCES Przedmioty(id),
23     ocena DECIMAL(2,1)
24 );
```

Wgrywanie przykładowych danych do tabeli.

```
1 INSERT INTO Studenci (imie, nazwisko, rok, kierunek) VALUES
2 ('Anna', 'Kowalska', 1, 'Informatyka'),
3 ('Jan', 'Nowak', 3, 'Informatyka'),
4 ('Maria', 'Wiśniewska', 2, 'Matematyka'),
5 ('Krzysztof', 'Malinowski', 3, 'Informatyka');
6
7 INSERT INTO Przedmioty (nazwa) VALUES
8 ('Analiza'),
9 ('Algebra'),
10 ('Programowanie');
11
12 INSERT INTO Oceny (student_id, przedmiot_id, ocena) VALUES
13 (1, 1, 4.5),
14 (1, 2, 5.0),
```

```

15 (2, 1, 3.5),
16 (2, 3, 4.0),
17 (3, 2, 4.0),
18 (4, 1, 5.0),
19 (4, 3, 4.5);

```

Zapytania.

```

1 SELECT *
2 FROM Studenci
3 WHERE rok = 3;
4
5 SELECT s.imie,
6       s.nazwisko,
7       AVG(o.ocena) AS srednia_ocen
8 FROM Studenci s JOIN Oceny o ON s.id = o.student_id
9 GROUP BY s.id;
10
11 SELECT s.imie,
12       s.nazwisko,
13       AVG(o.ocena) AS srednia_ocen
14 FROM Studenci s JOIN Oceny o ON s.id = o.student_id
15 GROUP BY s.id
16 HAVING AVG(o.ocena) > 4.0;

```

Wyniki zapytań.

```

1 +-----+-----+-----+-----+
2 | id | imie      | nazwisko   | rok   | kierunek   |
3 +-----+-----+-----+-----+
4 | 2 | Jan        | Nowak      |     3 | Informatyka |
5 | 4 | Krzysztof  | Malinowski |     3 | Informatyka |
6 +-----+-----+-----+-----+
7 2 rows in set (0.00 sec)
8
9 +-----+-----+-----+
10 | imie     | nazwisko   | srednia_ocen |
11 +-----+-----+-----+
12 | Anna     | Kowalska   |      4.75000 |
13 | Jan      | Nowak      |      3.75000 |
14 | Maria    | Wisniewska |      4.00000 |
15 | Krzysztof | Malinowski |      4.75000 |
16 +-----+-----+-----+
17 4 rows in set (0.00 sec)
18
19 +-----+-----+-----+
20 | imie     | nazwisko   | srednia_ocen |
21 +-----+-----+-----+
22 | Anna     | Kowalska   |      4.75000 |
23 | Krzysztof | Malinowski |      4.75000 |
24 +-----+-----+-----+
25 2 rows in set (0.00 sec)

```

2.2 Zadanie 2

Tworzenie faktów rodzic(dorosły, dziecko)

```

1 rodzic(jan, anna).
2 rodzic(jan, piotr).
3 rodzic(anna, kasia).
4 rodzic(anna, bartek).
5 rodzic(piotr, ola).
6 rodzic(piotr, Tomek).
7 rodzic(maria, jan).
8 rodzic(maria, ewa).

```

Tworzenie reguły dziecko/2

```

1 dziecko(Dziecko, Rodzic) :-
2     rodzic(Rodzic, Dziecko).

```

Tworzenie reguł przodek/2

Przodek może być albo bezpośrednio rodzicem potomka, albo rekurencyjnie rodzicem przodka potomka.

```
1 przodek(Przodek, Potomek) :-  
2     rodzic(Przodek, Potomek).  
3  
4 przodek(Przodek, Potomek) :-  
5     rodzic(Przodek, Posrednik),  
6     przodek(Posrednik, Potomek).
```

Przykłady zapytań prezentujące funkcjonowanie relacji rodzinnych

Kto jest rodzicemanny?

```
1 ?- rodzic(Kto, anna).  
2 Kto = jan.
```

Kto jest dzieckiemanny?

```
1 ?- dziecko(Kto, anna).  
2 Kto = kasia ;  
3 Kto = bartek.
```

Czy jan jest przodkiem tomka?

```
1 ?- przodek(jan, tomek).  
2 true.
```

Czy tomek jest przodkiem jana?

```
1 ?- przodek(tomek, jan).  
2 false.
```

Kto jest przodkiem tomka?

```
1 ?- przodek(Kto, tomek).  
2 Kto = piotr ;  
3 Kto = jan ;  
4 Kto = maria.
```

2.3 Zadanie 3

Tworzenie faktów studiuje(osoba, kierunek)

```
1 studiuje(hela, informatyka).  
2 studiuje(asia, informatyka).  
3 studiuje(fela, informatyka).  
4 studiuje(olaf, informatyka).  
5 studiuje(anna, matematyka).  
6 studiuje(paul, matematyka).  
7 studiuje(alek, matematyka).
```

Tworzenie reguły studiowany/2

```
1 studiowany(Kierunek, Osoba) :-  
2     studiuje(Osoba, Kierunek).
```

Tworzenie faktów prowadzi(osoba, kierunek)

```
1 prowadzi(adam, informatyka).  
2 prowadzi(asia, matematyka).
```

Tworzenie reguły prowadzony/2

```
1 prowadzony(Kierunek, Osoba) :-  
2     prowadzi(Osoba, Kierunek).
```

Tworzenie reguły prowadzacy_studenta/2

```
1 prowadzacy_studenta(Prowadzacy, Student) :-  
2     prowadzi(Prowadzacy, Kierunek),  
3     studiuje(Student, Kierunek).
```

Zapytanie "Kto jest prowadzącym studenta alek?"

```
1 ?- prowadzacy_studenta(Kto, alek).  
2 Kto = asia.
```

3 Wnioski

Porównanie rozwiązań deklaratywnych i imperatywnych/funkcyjnych:

Kryterium	Deklaratywne	Imperatywne/funkcyjne
Sposób myślenia	Myślimy w kategoriach faktów, reguł i własności, które mają być spełnione (<i>co ma być prawdą?</i>)	Myślimy w kategoriach kroków algorytmu i przekształceń danych (<i>jak to policzyć?</i>)
Sposób opisu	Fakty, reguły, zapytania; program to baza wiedzy i aparat wnioskowania	Funkcje, procedury, metody; program to zbiór instrukcji i definicji operujących na danych
Przepływ	Kolejność wykonania jest w dużej mierze ukryta – mechanizm wnioskowania steruje obliczeniami	Kolejność jest jawnie kontrolowana przez programistę (instrukcje, wywołania funkcji, pętle, rekursja)
Wiele rozwiązań	Naturalnie wspiera szukanie wszystkich wartości spełniających warunki jednego zapytania (wiele odpowiedzi na to samo pytanie)	Zwykle zwraca jedną wartość dla danego zestawu argumentów; wiele rozwiązań wymaga jawnego kodu (np. dodatkowych struktur danych)
Powiązania między danymi	Relacje między obiektami są centralne (predykaty typu <i>rodzic/2, student/3</i>); łatwo wyrażać złożone zależności i zapytania	Relacje trzeba zaszyć w strukturach danych i logice funkcji; zapytania są implementowane jako zwykły kod
Zastosowania	Systemy eksperckie, logika, wnioskowanie, zapytania do baz danych, zadania typu „znajdź wszystkie obiekty spełniające warunki”	Algorytmy numeryczne, przetwarzanie sygnałów, aplikacje interaktywne, systemy wbudowane, typowe programy użytkowe