

Wyrażenia CTE i zapytania rekurencyjne

Czas: 2 × 45 min

praca indywidualna lub w parach

Cel: Nauczenie się zapisywania złożonych zapytań jako czytelnych bloków WITH oraz przechodzenia przez hierarchie (pracownicy, drzewo kategorii) i generowania ciągów dat zapytaniem rekurencyjnym.

Po zajęciach potrafisz:

- zastępuje zagnieźdzone podzapytania jednym lub kilkoma wyrażeniami CTE
- pisze zapytanie WITH RECURSIVE z częścią początkową, częścią rekurencyjną i warunkiem stopu
- generuje kalendarz dat i łączy go z danymi, żeby pokazać okresy bez sprzedaży
- wypisuje hierarchię z poziomem i ścieżką, w dół i w górę drzewa

Sprzęt i materiały

- komputer w pracowni z XAMPP (MariaDB 10.4) lub MySQL 8.4
- baza sklep zaimportowana z pliku `sklep.sql` (folder Pliki_do_zadan)
- plik `r2_05_katalog_drzewo.sql` z folderu Pliki_do_zadan (tworzy bazę katalog)

Oprogramowanie

- **XAMPP (Apache + MariaDB + phpMyAdmin)** — panel sterowania XAMPP, konsola: przycisk Shell albo `C:\xampp\mysql\bin\mysql.exe -u root`
apachefriends.org/download.html
- **phpMyAdmin** — zakładka SQL do uruchamiania zapytań; wchodzi w skład XAMPP
localhost/phpmyadmin
- **VS Code lub Notepad++** — do pisania pliku .sql z zapytaniami
code.visualstudio.com

Bezpieczeństwo

- Każde zapytanie rekurencyjne musi mieć warunek stopu. MySQL przerywa rekurencję po 1000 krokach (`cte_max_recursion_depth`), MariaDB od 10.6 też ma limit 1000 (`max_recursive_iterations`), ale **w MariaDB 10.4 z XAMPP limit jest praktycznie nieograniczony** — błędne zapytanie może obciążyć serwer na długo. Gdy zapytanie nie kończy się po kilku sekundach, przerwij je (Ctrl+C w konsoli) i popraw warunek.
- Zmianę danych w Z14 wykonuj tylko w transakcji zakończonej ROLLBACK.

Przebieg zadania

1 CTE zamiast podzapytań

Składnia: `WITH nazwa AS (SELECT ...), druga AS (SELECT ... FROM nazwa ...) SELECT ... FROM druga;`. Kolejne CTE mogą korzystać z wcześniejszych.

1. Zaimportuj `sklep.sql` i `r2_05_katalog_drzewo.sql`. Utwórz plik `r2_05_nazwisko.sql`. Pracuj w bazie `sklep`.
2. Z1. Za pomocą CTE `srednia` (średnia cena produktu) wypisz produkty droższe od średniej i o ile są droższe (zaokrąglaj do groszy).
3. Z2. Dwa CTE: `wartosci` (wartość każdego zamówienia nieanulowanego z `id_klienta`) i `klienci_sumy` (suma i liczba zamówień klienta, korzysta z `wartosci`). Wypisz imię, nazwisko, liczbę zamówień i sumę klientów, których suma jest większa od średniej sumy wszystkich klientów.
4. Z3. CTE `sprzedaz` (wartość sprzedaży w kategoriach, bez anulowanych) i `calosc` (suma wszystkiego). Wypisz kategorie z wartością i udziałem procentowym (1 miejsce po przecinku).

2 Rekurencja — generowanie ciągów

```
WITH RECURSIVE liczby AS (  
    SELECT 1 AS n                                -- część początkowa  
    UNION ALL  
    SELECT n + 1 FROM liczby WHERE n < 10        -- część rekurencyjna z warunkiem stopu  
)  
SELECT * FROM liczby;
```

1. Z4. Uruchom zapytanie z ramki. Zmień je tak, żeby jednym wierszem (`GROUP_CONCAT`) wypisało liczby 1–10 po przecinku.
2. Z5. Wygeneruj rekurencyjnie wszystkie miesiące 2026 roku (pierwszy dzień miesiąca, krok `+ INTERVAL 1 MONTH`) i połącz `LEFT JOIN`-em z liczbą i wartością zamówień nieanulowanych w miesiącu. Miesiące bez sprzedaży mają mieć 0. Wynik: 12 wierszy.
3. Z6. Wykonaj `SET lc_time_names = 'pl_PL';`. Wygeneruj wszystkie dni września 2026 i policz: soboty i niedziele (z polskimi nazwami dni — `DAYNAME`) oraz dni robocze (bez świąt). Użyj `DAYOFWEEK` (1 = niedziela, 7 = sobota).

3 Hierarchia pracowników

Tabela `pracownicy` ma kolumnę `id_przełożonego` — to drzewo zapisane w jednej tabeli.

1. Z7. Wypisz całą strukturę firmy od dyrektora: pracownik z wcięciem zależnym od poziomu (`REPEAT`), stanowisko, poziom i ścieżka nazwisk „Majewska > Olszewski > Król”. Posortuj po ścieżce. Uwaga: kolumnę ścieżki w części początkowej rzutuj na dłuższy tekst: `CAST(nazwisko AS CHAR(300))`.
2. Z8. Policz wszystkich podwładnych (na każdym poziomie) Marka Olszewskiego (`id` 2) i sumę ich pensji.
3. Z9. Wypisz drogę „w górę” od Igora Grabowskiego (`id` 7) do dyrektora z numerem kroku (0 = Igor).

4 Drzewo kategorii (baza katalog)

Przejdź do bazy katalog: `USE katalog;`. Kategoria główna ma `id_nadrzednej = NULL`.

1. Z10. Wypisz całe drzewo kategorii z wcięciami, poziomem i ścieżką „Komputery / Podzespoły / Dyski”, posortowane po ścieżce (18 wierszy).
2. Z11. Znajdź wszystkie kategorie w poddrzewie „Podzespoły” (razem z nią samą) i policz produkty w tym poddrzewie oraz ich najniższą i najwyższą cenę.
3. Z12. Zbuduj „okruszki” (breadcrumbs) dla produktu „Dysk SSD NVMe 2 TB”: jeden wiersz w postaci „Komputery > Podzespoły > Dyski > Dyski SSD”.
4. Z13. Dla każdej kategorii głównej policz produkty z całego jej poddrzewa (wskazówka: w CTE przenoś identyfikator korzenia).
5. Z14. W transakcji ustaw kategorii Podzespoły (id 6) kategorię nadrzędną 9 (Dyski SSD) — powstaje cykl. Napisz zapytanie o poddrzewo kategorii 6 z kolumną poziom i warunkiem stopu `WHERE p.poziom < 10` w części rekurencyjnej. Ile wierszy zwraca? Wykonaj ROLLBACK.

W sprawozdaniu umieść

- ☐ plik `.sql` z zapytaniami Z1–Z14
- ☐ zrzuty ekranu wyników Z5, Z7 i Z10
- ☐ wynik Z14 i wyjaśnienie, co by się stało bez warunku stopu
- ☐ odpowiedzi na pytania kontrolne

Pytania kontrolne

1. Podaj dwie zalety CTE w porównaniu z podzapytaniem w FROM.
2. Z jakich dwóch części składa się CTE rekurencyjne i dlaczego łączy je UNION ALL?
3. Dlaczego w Z7 trzeba rzutować kolumnę ścieżki przez CAST w części początkowej?
4. Jak w bazie z cyklem (Z14) można wykryć, że wróciliśmy do już odwiedzonego węzła, bez ograniczania liczby poziomów?

Kryteria oceny

Kryterium	Punkty
CTE nierekurencyjne (Z1–Z3)	4
Generowanie ciągów i kalendarza (Z4–Z6)	4
Hierarchia pracowników (Z7–Z9)	4
Drzewo kategorii i cykl (Z10–Z14)	5
Odpowiedzi na pytania kontrolne	3
Razem	20

Sprawozdanie przygotuj na wzorze dołączonym do zadania (plik .docx o tym samym numerze).