

Hierarchie i modele danych w bazie relacyjnej

Czas: 2 × 45 min

praca indywidualna lub w parach

Cel: Porównanie w praktyce sposobów zapisu hierarchii (lista sąsiedztwa, ścieżka materializowana, dokument JSON) i grafu (tabela krawędzi) w bazie relacyjnej oraz dobór modelu danych do zastosowania.

Po zajęciach potrafisz:

- odczytuje poddrzewo, przodków i liście hierarchii zapisanej jako lista sąsiedztwa (WITH RECURSIVE)
- wyszukuje w ścieżce materializowanej i porównuje koszt przeniesienia gałęzi w obu modelach
- odnajduje węzły w drzewie zapisanym jako dokument JSON
- wykonuje zapytania grafowe: znajomi, propozycje znajomych, stopień oddalenia i najkrótsza ścieżka
- uzasadnia wybór modelu danych dla konkretnego zastosowania

Sprzęt i materiały

- komputer z XAMPP (MariaDB 10.4) lub MySQL 8.4
- plik `r2_27_modele.sql` z folderu `Pliki_do_zadan`

Oprogramowanie

- **XAMPP (MariaDB, phpMyAdmin)** — zapytania WITH RECURSIVE wygodniej pisać w konsoli lub w zakładce SQL [apachefriends.org](https://www.apachefriends.org)
- **draw.io (diagrams.net)** — szkic drzewa kategorii i grafu znajomości app.diagrams.net

Przygotowanie przed zajęciami

- Przypomnij sobie z lekcji 2.03 budowę zapytania WITH RECURSIVE: część początkowa, UNION ALL, część rekurencyjna z warunkiem zakończenia.

Bezpieczeństwo

- Pracuj w bazie `modele`. Polecenia zmieniające drzewo (Z8) wykonuj w transakcji i zakończ ROLLBACK, żeby dalsze zapytania dawały wyniki zgodne z treścią.
- Zapytanie rekurencyjne bez warunku zakończenia może działać bardzo długo przy cyklu w danych — zawsze ogranicz głębokość i sprawdzaj, czy węzeł nie był już odwiedzony.

Przebieg zadania

1 Lista sąsiedztwa i WITH RECURSIVE

Zaimportuj plik `r2_27_modele.sql`. Narysuj w draw.io drzewo z tabeli `kategorie` (16 węzłów, korzeń „Sklep”). Każdy wiersz zna tylko swojego rodzica (`id_rodzica`).

```
-- W MySQL kolumna tekstowa CTE ma długość z części początkowej --
-- dlatego ścieżkę zaczynaj od CAST(... AS CHAR(255)).
WITH RECURSIVE poddrzewo AS (
  SELECT id_kategorii, nazwa, 0 AS poziom, CAST(nazwa AS CHAR(255)) AS sciezka
  FROM kategorie WHERE nazwa = 'Peryferia'
  UNION ALL
  SELECT ...    -- uzupełnij
)
SELECT ...
```

1. Z1. Wyświetl bezpośrednie podkategorie kategorii „Peryferia” (złączenie tabeli z samą sobą).
2. Z2. Uzupełnij zapytanie z ramki: wyświetl całe poddrzewo „Peryferia” z poziomem i ścieżką „Peryferia > Myszy > ...”. Nazwy wcięte według poziomu (REPEAT), sortowanie według ścieżki.
3. Z3. Zbuduj „okruszki” (breadcrumb) dla kategorii „Myszy pionowe”: jedna wartość w postaci „Sklep > Peryferia > ...” (rekurencja w górę drzewa i GROUP_CONCAT).
4. Z4. Wyświetl liście drzewa — kategorie bez podkategorii (NOT EXISTS).
5. Z5. Dla każdej kategorii policz wszystkich jej potomków na wszystkich poziomach; pomір kategorii bez potomków. Wskazówka: rekurencyjnie zbuduj pary (przodek, potomek).

2 Ścieżka materializowana i dokument JSON

Tabela `kategorie_sciezka` przechowuje to samo drzewo jako tekst ścieżki, np. „/1/3/10/” (Sklep → Peryferia → Myszy). Tabela `katalog_json` zawiera gałąź Peryferia jako jeden zagnieżdżony dokument.

1. Z6. Wyświetl poddrzewo „Peryferia” z tabeli `kategorie_sciezka` jednym zapytaniem bez rekurencji (LIKE). Oblicz poziom węzła z liczby znaków „/” w ścieżce.
2. Z7. Wyświetl przodków kategorii „Myszy pionowe” z tabeli `kategorie_sciezka` (złączenie tabeli z samą sobą: ścieżka potomka zaczyna się od ścieżki przodka).
3. Z8. W transakcji przenieś kategorię „Myszy biurowe” (z jej podkategorią) bezpośrednio pod „Peryferia” w obu modelach. Zapisz, ile wierszy zmieniło każde polecenie UPDATE (ROW_COUNT()), sprawdź wynik i wykonaj ROLLBACK.
4. Z9. Z dokumentu w `katalog_json` odczytaj nazwę drugiej podkategorii, liczbę podkategorii pierwszego poziomu oraz nazwę najgłębiej położonego węzła. Funkcją JSON_SEARCH znajdź ścieżkę do „Myszy pionowe” i wszystkie ścieżki nazw zaczynających się od „Myszy”.

3 Graf znajomości w tabeli krawędzi

Tabela znajomosci zapisuje każdą znajomość raz ($id_a < id_b$). Narysuj graf w draw.io: 10 osób, 10 krawędzi.

1. Z10. Utwórz widok znajomi (osoba, znajomy, od_kiedy) zawierający każdą znajomość w obu kierunkach (UNION ALL).
2. Z11. Wyświetl znajomych Celiny oraz liczbę znajomych każdej osoby, od największej.
3. Z12. Zaproponuj Ali nowych znajomych: znajomi jej znajomych, którzy nie są jej znajomymi ani nią samą, z liczbą wspólnych znajomych. Osobno wyświetl wspólnych znajomych Ali i Darka.
4. Z13. Zapytaniem WITH RECURSIVE oblicz stopień oddalenia każdej osoby od Ali (Ala = 0, jej znajomi = 1 itd.). Zapobiegaj cyklom: przechowuj listę odwiedzonych id i sprawdzaj ją funkcją FIND_IN_SET; ogranicz głębokość do 6.
5. Z14. Wyświetl najkrótszą ścieżkę znajomości od Ali do Henryka w postaci „Ala → ... → Henryk”. Wyjaśnij, dlaczego w wyniku Z13 nie ma Igi i Jana.

4 Wybór modelu danych

Na podstawie wyników i lekcji 2.17 dobierz model do zastosowania.

1. Uzupełnij w sprawozdaniu tabelę „Wybór modelu”: dla każdego zastosowania wskaż model (relacyjny, hierarchiczny — lista sąsiedztwa lub ścieżka, dokumentowy, grafowy, klucz–wartość) i jedno zdanie uzasadnienia.

W sprawozdaniu umieść

- ☐ plik `zadanie27.sql` z zapytaniami Z1–Z14 opisanymi komentarzami
- ☐ rysunki drzewa kategorii i grafu znajomości z draw.io
- ☐ zrzuty ekranu wyników Z2, Z8 (przed ROLLBACK), Z13 i Z14
- ☐ tabela porównania modeli hierarchii i tabela „Wybór modelu”
- ☐ odpowiedzi na pytania kontrolne

Pytania kontrolne

1. Który zapis hierarchii (lista sąsiedztwa czy ścieżka materializowana) jest lepszy, gdy drzewo często się zmienia, a który — gdy częściej się je odczytuje? Odwołaj się do Z8.
2. Dlaczego znajomość zapisano raz z warunkiem $id_a < id_b$ i po co powstał widok znajomi?
3. Jak zapytanie z Z13 unika nieskończonej pętli w grafie z cyklem (Ala–Bartek–Celina)?
4. Kiedy warto użyć osobnej bazy grafowej (np. Neo4j) zamiast tabeli krawędzi w MySQL?

Kryteria oceny

Kryterium	Punkty
Z1–Z5: zapytania rekurencyjne na liście sąsiedztwa	5
Z6–Z9: ścieżka materializowana, przeniesienie gałęzi, dokument JSON	4
Z10–Z14: widok i zapytania grafowe	5
Rysunki, tabele porównania i wyboru modelu	3
Odpowiedzi na pytania kontrolne	3
Razem	20

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