

Widoki — tworzenie i użycie

Czas: 2 × 45 min

praca indywidualna lub w parach

Cel: Nauczenie się tworzenia widoków, które upraszczają złożone zapytania i ukrywają niepotrzebne kolumny, oraz sprawdzenie, że widok nie przechowuje danych.

Po zajęciach potrafisz:

- tworzy widoki poleceniem CREATE VIEW i zmienia je przez CREATE OR REPLACE VIEW
- korzysta z widoku jak z tabeli w zapytaniach z WHERE, GROUP BY i ORDER BY
- buduje widok na podstawie innego widoku i wyjaśnia skutki usunięcia widoku bazowego
- odczytuje definicję widoku (SHOW CREATE VIEW, information_schema.VIEWS)
- projektuje widok ukrywający dane osobowe

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)

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 pisanie pliku .sql z zapytaniami
code.visualstudio.com

Bezpieczeństwo

- Pracuj na świeżo zaimportowanej bazie sklep na swoim komputerze. Widoki usuwaj tylko te, które sam utworzyłeś.
- Polecenie z Z12 wykonuj w transakcji zakończonej ROLLBACK — dane mają wrócić do stanu wyjściowego.
- Widok ukrywający kolumny (Z5) chroni dane tylko wtedy, gdy użytkownik nie ma dostępu do samej tabeli — to temat zadania 3.

Przebieg zadania

1 Pierwszy widok

Widok to zapisane zapytanie: serwer wykonuje je przy każdym odczycie.

1. Zaimportuj ponownie `sklep.sql`, żeby zacząć od danych wyjściowych. Utwórz plik `r2_02_nazwisko.sql`.
2. Z1. Utwórz widok `v_katalog` z kolumnami: `id_produktu`, `nazwa`, `kategoria` (nazwa kategorii), `cena`, `stan`. Produkty bez kategorii też mają się w nim znaleźć.
3. Z2. Korzystając tylko z widoku `v_katalog`, wypisz monitory, a potem policz wszystkie wiersze widoku (powinno być 15).
4. Z3. Wyświetl listę widoków w bazie: `SHOW FULL TABLES WHERE Table_type = 'VIEW'`. Sprawdź, jak widok jest oznaczony w phpMyAdmin.

2 Widok z grupowaniem i widok bez danych osobowych

Widok `v_wartosc_zamowien` będzie potrzebny w kolejnych punktach — utwórz go dokładnie według poniższej definicji.

```
CREATE VIEW v_wartosc_zamowien AS
SELECT z.id_zamowienia, z.data_zamowienia, z.id_klienta,
       CONCAT(k.imie, ' ', k.nazwisko) AS klient, z.status,
       SUM(pz.ilosc * pz.cena_jedn) AS wartosc
FROM zamowienia z
JOIN klienci k ON k.id_klienta = z.id_klienta
JOIN pozycje_zamowienia pz ON pz.id_zamowienia = z.id_zamowienia
GROUP BY z.id_zamowienia, z.data_zamowienia, z.id_klienta,
         k.imie, k.nazwisko, z.status;
```

1. Z4. Utwórz widok `v_wartosc_zamowien` (kod powyżej). Napisz zapytanie do widoku, które pokaże 3 najdroższe zamówienia nieanulowane.
2. Z5. Utwórz widok `v_klienci_publiczni` z kolumnami: `id_klienta`, imie, nazwisko skrócone do pierwszej litery z kropką (np. „N.”), miasto. Bez adresu e-mail i daty rejestracji.
3. Z6. Spróbuj odczytać kolumnę `email` z widoku `v_klienci_publiczni`. Zapisz numer i treść błędu.

3 Zmiana widoku i widok na widoku

1. Z7. Zmień `v_katalog` poleceniem `CREATE OR REPLACE VIEW`: dodaj kolumnę `wartosc_magazynu` (`cena × stan`). Następnie oblicz z widoku wartość magazynu w każdej kategorii, od największej.
2. Z8. Utwórz widok `v_sprzedaz_klientow` na podstawie widoku `v_wartosc_zamowien`: `id_klienta`, klient, liczba_zamowien, razem — bez zamówień anulowanych. Wyświetl go posortowany malejąco według kwoty.
3. Z9. Utwórz widok `v_miesiace` z listą nazw kolumn w nagłówku `CREATE VIEW v_miesiace (miesiac, liczba, wartosc) AS ...`: miesiąc w formacie RRRR-MM, liczba i wartość zamówień nieanulowanych. Wyświetl go według miesięcy.

4 Jak widok działa w środku

1. Z10. Wykonaj `SHOW CREATE VIEW v_klienci_publiczni;` Odszukaj w wyniku `ALGORITHM`, `DEFINER` i `SQL SECURITY` i zapisz ich wartości.
2. Z11. Z tabeli `information_schema.VIEWS` wypisz nazwy widoków bazy sklep i kolumnę `IS_UPDATABLE`. Zastanów się, dlaczego część widoków ma wartość `NO`.
3. Z12. W konsoli rozpocznij transakcję, dodaj do zamówienia 8 pozycję (produkt 7, ilość 3, cena 129.00), odczytaj wartość zamówienia 8 z widoku, wykonaj `ROLLBACK` i odczytaj ją jeszcze raz. Co to mówi o przechowywaniu danych w widoku?
4. Z13. Usuń widok `v_wartosc_zamowien` i spróbuj odczytać `v_sprzedaz_klientow`. Zapisz błąd. Następnie odtwórz `v_wartosc_zamowien` i sprawdź, że `v_sprzedaz_klientow` znów działa.
5. Z14. Usuń widok `v_miesiace` poleceniem `DROP VIEW IF EXISTS`. Sprawdź, że tabele i dane bazy sklep zostały bez zmian.

W sprawozdaniu umieść

- ☐ plik `.sql` z poleceniami Z1–Z14
- ☐ zrzuty ekranu wyników Z4, Z7 i Z8 oraz komunikatów błędów z Z6 i Z13
- ☐ wartości `ALGORITHM`, `DEFINER` i `SQL SECURITY` z Z10
- ☐ wnioski z Z12 i Z13
- ☐ odpowiedzi na pytania kontrolne

Pytania kontrolne

1. Czy widok zajmuje miejsce na dysku tak jak tabela? Uzasadnij na podstawie Z12.
2. Podaj dwa powody, dla których programista aplikacji woląłby pisać `SELECT * FROM v_wartosc_zamowien` zamiast pełnego złączenia.
3. Dlaczego usunięcie widoku bazowego nie zostało zablokowane, choć zależy od niego inny widok? Jakie to niesie ryzyko?
4. Dlaczego widok `v_wartosc_zamowien` jest nieaktualizowalny (`IS_UPDATABLE = NO`)?

Kryteria oceny

Kryterium	Punkty
Widoki Z1–Z5 utworzone poprawnie i użyte w zapytaniach	6
Zmiana widoku i widoki na widokach (Z7–Z9)	5
Analiza działania widoków (Z10–Z14) z wnioskami	5
Odpowiedzi na pytania kontrolne	4
Razem	20

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