

Warunki w SQL: CASE, IF, IFNULL, COALESCE, NULLIF

Czas: 2 × 45 min

praca indywidualna lub w parach

Cel: Nauczenie się wybierania wartości według warunków w zapytaniach, budowania raportów przestawnych jednym zapytaniem i bezpiecznego obchodzenia się z NULL i dzieleniem przez zero.

Po zajęciach potrafisz:

- stosuje CASE w obu postaciach oraz funkcję IF w SELECT, ORDER BY i UPDATE
- kategoryzuje dane (przedziały cen, segmenty klientów) i grupuje po kategorii
- buduje raport przestawny (pivot) z SUM(CASE ...) i COUNT(CASE ...)
- zastępuje NULL wartością domyślną (IFNULL, COALESCE) i zapobiega dzieleniu przez zero (NULLIF)

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

- Polecenia UPDATE z Z12 i Z13 wykonuj tylko w transakcji (START TRANSACTION ... ROLLBACK) — ceny i stany mają wrócić do stanu wyjściowego.
- Przed każdym UPDATE z CASE sprawdź wynik tym samym wyrażeniem w SELECT. Brak gałęzi ELSE w CASE ustawia NULL — w kolumnie NOT NULL to błąd, a w innej utrata danych.

Przebieg zadania

1 CASE i IF

CASE ma dwie postacie: z warunkami (CASE WHEN warunek THEN ...) i z porównaniem do wartości (CASE kolumna WHEN wartość THEN ...).

1. Zaimportuj `sklep.sql`. Utwórz plik `r2_04_nazwisko.sql`.
2. Z1. Każdemu produktowi przypisz przedział cenowy: „tani” (poniżej 300 zł), „średni” (300–1499,99 zł), „drogi” (od 1500 zł). Policz produkty w każdym przedziale i posortuj przedziały od najtańszego.
3. Z2. Dla zamówień 4, 8, 9 i 10 wypisz opis statusu (CASE z porównaniem): nowe → „czeka na płatność”, opłacone → „do wysłania”, wysłane → „zrealizowane”, pozostałe → „anulowane”.
4. Z3. Funkcją IF oznacz produkty o stanie poniżej 5 sztuk jako „dostępny” albo „brak”.
5. Z4. Wypisz zamówienia w kolejności obsługi: najpierw nowe, potem opłacone, wysłane i anulowane, a w obrębie statusu od najstarszego (CASE w ORDER BY). Pokaż 5 pierwszych.

2 NULL i wartości domyślne

1. Z5. Wypisz klientów bez miasta, pokazując zamiast NULL tekst „nie podano” (IFNULL). Wypisz produkt 1 i produkty bez kategorii z nazwą kategorii albo „bez kategorii” (COALESCE i LEFT JOIN).
2. Z6. Jednym SELECT-em sprawdź wartości: `COALESCE(NULL, NULL, 'trzeci', 'czwarty')`, `IFNULL(NULL, 'zastępstwo')`, `NULLIF(5, 5)`, `NULLIF(5, 3)`, `IF(NULL = NULL, 'tak', 'nie')`, `IFNULL(NULL = NULL, 'nieznane')`. Wyjaśnij każdy wynik.
3. Z7. Podziel klientów na segmenty według łącznej wartości zamówień bez anulowanych: VIP (od 5000 zł), stały (od 1000 zł), okazjonalny (powyżej 0), bez zakupów. Klienci bez zamówień też mają się pojawić z kwotą 0.00. Wskazówka: warunek `status <> 'anulowane'` umieść w ON złączenia, nie w WHERE.

3 Raporty przestawne

Wiersze grupujemy jak zwykle, a każda kolumna to osobny `SUM(CASE ...)` dla jednej wartości.

1. Z8. Raport: wiersze — miasta klientów, kolumny — liczba zamówień w statusach nowe, opłacone, wysłane, anulowane i razem. Posortuj według kolumny razem malejąco.
2. Z9. Raport sprzedaży (bez anulowanych): wiersze — kategorie (produkty bez kategorii jako „bez kategorii”), kolumny — wartość sprzedaży w kwartałach Q1, Q2, Q3 2026 i razem. Użyj funkcji `QUARTER()`.
3. Z10. Dla każdego klienta, który złożył zamówienie, podaj: liczbę wszystkich zamówień, liczbę anulowanych (`COUNT(CASE ...)`) oraz liczbę opłaconych lub wysłanych (`SUM(IF(...))`).

4 Dzielenie przez zero i CASE w UPDATE

1. Z11. Wykonaj `SELECT 10/0;` i `SHOW WARNINGS;`. Następnie dla każdego klienta oblicz średnią wartość zamówienia nieanulowanego: suma wartości nieanulowanych / liczba nieanulowanych (zbuduj najpierw tabelę pochodną z wartością każdego zamówienia). Klientka z samym anulowanym zamówieniem daje dzielenie przez zero — zabezpiecz mianownik funkcją `NULLIF` i porównaj liczbę ostrzeżeń przed i po.
2. Z12. Odczytaj `SELECT @@sql_mode;`. W transakcji spróbuj `UPDATE produkty SET stan = stan / 0 WHERE id_produkta = 1;` i zakończ `ROLLBACK`. Wyjaśnij, dlaczego w SELECT było tylko ostrzeżenie, a tu (przy `ERROR_FOR_DIVISION_BY_ZERO` i trybie ścisłym) jest błąd.
3. Z13. W transakcji jednym poleceniem `UPDATE` z CASE obniż ceny laptopów (kategoria 1) o 5% i podnieś ceny akcesoriów (kategoria 3) o 10%, z zaokrągleniem do groszy; pozostałe ceny bez zmian. Sprawdź liczbę zmienionych wierszy i nowe ceny, potem `ROLLBACK`.

W sprawozdaniu umieść

- ☐ plik `.sql` z zapytaniami Z1–Z13
- ☐ zrzuty ekranu wyników Z7, Z8 i Z9
- ☐ wyjaśnienie wyników Z6 i Z11–Z12
- ☐ odpowiedzi na pytania kontrolne

Pytania kontrolne

1. Czym różni się CASE z warunkami od CASE z porównaniem do wartości? Kiedy nie da się użyć tej drugiej postaci?
2. Co zwróci CASE bez ELSE, gdy żaden warunek nie jest spełniony? Jakie to ma skutki w UPDATE?
3. Czym różni się COALESCE od IFNULL?
4. Dlaczego w Z7 warunek na status trzeba umieścić w ON, a nie w WHERE?

Kryteria oceny

Kryterium	Punkty
CASE i IF w SELECT i ORDER BY (Z1–Z4)	4
Obsługa NULL i segmentacja klientów (Z5–Z7)	4
Raporty przestawne (Z8–Z10)	5
Dzielenie przez zero i UPDATE z CASE (Z11–Z13) z wyjaśnieniem	4
Odpowiedzi na pytania kontrolne	3
Razem	20

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