

SELECT — wybieranie kolumn i wierszy

Czas: 2 × 45 min

praca indywidualna

Cel: Pisanie prostych zapytań SELECT: wybieranie kolumn, aliasy, DISTINCT, filtrowanie wierszy warunkami WHERE i obliczenia w zapytaniu.

Po zajęciach potrafisz:

- wybiera z tabeli potrzebne kolumny i nadaje im aliasy
- usuwa powtórzenia wyników słowem DISTINCT
- filtruje wiersze operatorami porównania oraz AND, OR i NOT, stosując nawiasy
- oblicza w zapytaniu wartości wyliczane (np. wartość magazynu, cena netto)
- zapisuje zapytania w pliku .sql z komentarzami i dokumentuje wyniki

Sprzęt i materiały

- komputer z Windows 11 i zainstalowanym XAMPP (Apache + MariaDB/MySQL)
- plik `sklep.sql` — baza ćwiczeniowa z prezentacji (folder Pliki_do_zadan)

Oprogramowanie

- **XAMPP** — phpMyAdmin (<http://localhost/phpmyadmin>) i konsola `mysql` apachefriends.org/download.html
- **Visual Studio Code** lub **Notepad++** — plik z zapytaniem; Notepad++: <https://notepad-plus-plus.org/code.visualstudio.com>

Przygotowanie przed zajęciami

- Przypomnij sobie z prezentacji 1.11 budowę zapytania SELECT ... FROM ... WHERE.
- Sprawdź, czy XAMPP uruchamia moduł MySQL bez błędów (zadanie 1).

Bezpieczeństwo

- Pracuj wyłącznie na swojej bazie `sklep` na własnym komputerze; nie usuwaj innych baz.
- W tym zadaniu wykonujesz tylko zapytania SELECT — nie uruchamiaj poleceń UPDATE, DELETE ani DROP.

Przebieg zadania

1 Przygotowanie bazy i pliku z zapytaniami

Każde zapytanie wpisuj do pliku `zad14_nazwisko.sql`, poprzedzając je komentarzem z numerem, a następnie uruchamiaj w konsoli lub na karcie SQL w phpMyAdmin.

```
-- Zadanie 14, Jan Kowalski, klasa 1T
USE sklep;

-- Z1. Wszystkie dane z tabeli produkty
SELECT ... ;
```

1. W XAMPP Control Panel uruchom moduły Apache i MySQL (przycisk Start).
2. Jeśli bazy `sklep` nie ma albo była zmieniana na wcześniejszych zajęciach, zaimportuj ponownie plik `sklep.sql` (phpMyAdmin → Import). Skrypt sam usuwa i tworzy bazę od nowa.
3. Otwórz konsolę (przycisk Shell w XAMPP, polecenie `mysql -u root`) albo phpMyAdmin → baza `sklep` → karta SQL. Obejrzyj strukturę tabel poleceniem `DESCRIBE produkty;` i `DESCRIBE klienci;`.

2 Kolumny, aliasy, DISTINCT

1. Z1. Wyświetl wszystkie kolumny i wiersze tabeli `produkty`. Zapisz liczbę wierszy.
2. Z2. Wyświetl tylko nazwę i cenę wszystkich produktów.
3. Z3. Zmień Z2 tak, aby kolumny miały nagłówki `produkt` i `cena brutto` (alias ze spacją ujmij w odwrotne apostrofy — klawisz pod Esc — albo w cudzysłów).
4. Z4. Wyświetl listę miast, z których pochodzą klienci, bez powtórzeń. Ile jest wartości i czy wśród nich jest NULL?

3 WHERE i operatory porównania

Operatory: `=`, `<>` (lub `!=`), `<`, `>`, `<=`, `>=`. Teksty i daty zapisuj w apostrofach, np. `'2026-01-01'`.

1. Z5. Nazwy i ceny produktów droższych niż 1000 zł.
2. Z6. Produkty, których nie ma w magazynie (stan równy 0).
3. Z7. Produkty z kategorii „Akcesoria” — najpierw odczytaj jej identyfikator zapytaniem `SELECT * FROM kategorie;`, a potem użyj go w warunku.
4. Z8. Imię, nazwisko i datę rejestracji klientów zarejestrowanych 1 stycznia 2026 r. lub później.
5. Z9. Zamówienia o statusie innym niż `anulowane`.

4 AND, OR, NOT i nawiasy

AND ma wyższy priorytet niż OR — tak jak mnożenie przed dodawaniem.

1. Z10. Produkty w cenie od 500 do 1500 zł włącznie — użyj operatorów `>=` i `<=` połączonych AND.
2. Z11. Produkty z kategorii 1 lub 2, których stan jest mniejszy niż 5. Napisz dwie wersje: Z11a bez nawiasów (`id_kategorii = 1 OR id_kategorii = 2 AND stan < 5`) i Z11b z nawiasami wokół warunku OR. Porównaj liczbę wierszy.
3. Z12. Klienci, którzy nie mieszkają w Krakowie (użyj NOT). Dodaj liczbę wierszy Z12 do liczby klientów z Krakowa — czy wychodzi 10? Wyjaśnij.
4. Z13. Imię, nazwisko, stanowisko i pensja pracowników zarabiających ponad 9000 zł, z wyjątkiem dyrektora.

5 Wyrażenia w SELECT

W liście kolumn możesz liczyć: `+` `-` `*` `/`, a wynik zaokrąglić funkcją `ROUND(wartość, 2)`.

1. Z14. Nazwa, cena, stan oraz wartość towaru w magazynie ($\text{cena} \times \text{stan}$) jako kolumna `wartosc_magazynu`.
2. Z15. Nazwa, cena brutto i cena netto ($\text{brutto} / 1,23$, zaokrąglona do 2 miejsc) dla produktów z kategorii „Podzespoły”.
3. Z16. Dla produktów droższych niż 3000 zł: nazwa, cena, cena po rabacie 10% (`cena_po_rabacie`) i kwota rabatu (`rabat`).
4. Sprawdź, czy alias z Z14 można użyć w WHERE: dopisz do Z14 warunek `WHERE wartosc_magazynu > 10000` i zanotuj komunikat. Następnie popraw warunek tak, by działał.
5. Zapisz plik `zad14_nazwisko.sql` i zrób zrzuty ekranu wyników Z4, Z11a, Z11b, Z12 i Z14.

W sprawozdaniu umieść

- ☐ tabela Z1–Z16 z treścią zapytań i liczbą zwróconych wierszy
- ☐ zrzuty ekranu wyników Z4, Z11a, Z11b, Z12 i Z14
- ☐ wyjaśnienie różnicy między Z11a i Z11b oraz wyniku Z12
- ☐ komunikat błędu przy aliasie w WHERE i poprawione zapytanie
- ☐ plik `zad14_nazwisko.sql`
- ☐ odpowiedzi na pytania kontrolne

Pytania kontrolne

1. Dlaczego w programach lepiej wypisywać potrzebne kolumny zamiast używać `SELECT *`?
2. W jakiej kolejności serwer wykonuje AND i OR? Jak to zmienić?
3. Dlaczego warunek `miasto <> 'Kraków'` nie zwraca klientów, którzy nie podali miasta?
4. Dlaczego alias z listy SELECT nie działa w WHERE tego samego zapytania?

Kryteria oceny

Kryterium	Punkty
Z1–Z9 — poprawne zapytania i wyniki	6
Z10–Z13 — poprawne warunki złożone, wyjaśnienie Z11 i Z12	5
Z14–Z16 i test aliasu w WHERE	3
Plik .sql z komentarzami i zrzuty ekranu	3
Odpowiedzi na pytania kontrolne	3
Razem	20

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