

PHP i MySQLi — strona z danymi z bazy

Czas: 2 × 45 min

praca indywidualna (warunki jak na egzaminie: bez internetu poza dokumentacją)

Cel: Zbudowanie strony internetowej, która pobiera dane z bazy `sklep` funkcjami `mysqli`, wyświetla je w tabeli HTML i filtruje po kategorii wybranej w formularzu.

Po zajęciach potrafisz:

- łączy się z bazą funkcją `mysqli_connect` i ustawia kodowanie `utf8mb4`
- wykonuje zapytanie (`mysqli_query`) i wypisuje wiersze w pętli `mysqli_fetch_assoc` / `mysqli_fetch_row`
- buduje listę rozwijaną formularza z danymi z bazy
- filtruje dane według wartości z formularza zapytaniem przygotowanym `mysqli_prepare`
- tworzy układ strony z plikiem CSS zgodnie z opisem

Sprzęt i materiały

- komputer z XAMPP (Apache, PHP 8, MariaDB 10.4)
- świeżo zaimportowana baza `sklep` (plik `sklep.sql`)

Oprogramowanie

- **XAMPP** — pliki strony w `C:\xampp\htdocs\sklep_www`
apachefriends.org/download.html
- **phpMyAdmin** — testowanie zapytań Z1–Z6 i zrzuty
localhost/phpmyadmin
- **VS Code lub Notepad++** — edytor HTML, CSS i PHP
code.visualstudio.com
- **Dokumentacja PHP — MySQLi** — dozwolona w trakcie pracy
php.net/manual/pl/book.mysqli.php

Przygotowanie przed zajęciami

- Przypomnij sobie z prezentacji 2.13 kolejność: połączenie → zapytanie → pętla po wierszach → zamknięcie połączenia.

Bezpieczeństwo

- Pracuj na koncie root tylko lokalnie w XAMPP — na prawdziwym serwerze aplikacja używa osobnego konta z minimalnymi uprawnieniami (zadanie 19).
- Wartość z formularza (`$_GET['kat']`) przekazuj do bazy tylko jako parametr zapytania przygotowanego. Nazwy produktów wypisuj przez `htmlspecialchars` — zawierają cudzysłowy (np. Monitor 27").
- Nie zmieniaj danych w bazie — zadanie wymaga tylko odczytu.

Przebieg zadania

1 Zapytania w phpMyAdmin

Każde zapytanie wykonaj w zakładce SQL bazy `sklep`, zrób zrzut ekranu wyniku (nazwy plików `z1.png` ... `z6.png`) i zapisz treść w pliku `kwerendy.txt`.

1. Z1. Lista kategorii: identyfikator i nazwa, posortowane alfabetycznie po nazwie.
2. Z2. Wszystkie produkty: nazwa produktu, nazwa kategorii, cena i stan — także produkty bez kategorii; od najdroższego.
3. Z3. Produkty kategorii o identyfikatorze 2: nazwa, kategoria, cena, stan; od najdroższego.
4. Z4. Dla każdej kategorii: nazwa, liczba produktów i wartość magazynu (suma cena × stan); także kategorie bez produktów.
5. Z5. Produkty, których nie ma w magazynie (stan 0).
6. Z6. Liczba produktów i średnia cena (zaokrąglona do 2 miejsc) w każdej kategorii, tylko kategorie z co najmniej 3 produktami.

2 Układ strony i CSS

Utwórz folder `C:\xampp\htdocs\sklep_www` z plikami `produkty.php` i `styl.css`. Stronę oglądaj pod adresem `http://localhost/sklep_www/produkty.php`.

1. Strona ma tytuł karty „Sklep komputerowy — produkty”, język polski i kodowanie UTF-8.
2. Baner (header): tło `#1F4E79`, biały tekst, wyśrodkowany, nagłówek `h1` „Sklep komputerowy”.
3. Panel lewy (aside): 25% szerokości, wysokość 600 px, tło `#DDEBF7`; panel prawy (main): 75% szerokości, wysokość 600 px. Panele obok siebie.
4. Stopka: jak baner, z tekstem „Stronę wykonał: ” i Twoim imieniem i nazwiskiem.
5. Tabela: obramowanie 1 px ciągłe `#1F4E79`, scalone krawędzie, nagłówki z tłem `#9BC2E6`, podświetlenie wiersza po najechaniu kursorem (`#F2F2F2`). Klasa `brak` — czerwony tekst.

3 Skrypty PHP z mysqli

Używaj funkcji w stylu proceduralnym: `mysqli_connect`, `mysqli_set_charset`, `mysqli_query`, `mysqli_fetch_assoc` / `mysqli_fetch_row`, `mysqli_num_rows`, `mysqli_prepare`, `mysqli_close`.

1. Skrypt 1 (panel lewy): formularz metodą GET z listą rozwijaną `kat`. Pierwsza opcja „wszystkie” (wartość 0), kolejne — kategorie z zapytania Z1 (wartość = id, tekst = nazwa) i przycisk „Pokaż”.
2. Skrypt 2 (panel lewy, pod formularzem): nagłówek „Podsumowanie” i lista punktowana z wyników zapytania Z4 w formacie „Laptopy: produktów 3, wartość 61 584,00 zł” (`number_format`).
3. Skrypt 3 (panel prawy): jeśli nie wybrano kategorii — wynik zapytania Z2, jeśli wybrano — Z3 z parametrem z formularza (`mysqli_prepare`, `mysqli_stmt_bind_param` z typem `i`, `mysqli_stmt_execute`, `mysqli_stmt_get_result`). Nad tabelą wypisz „Liczba produktów: N” (`mysqli_num_rows`).
4. W tabeli: produkt, kategoria (dla NULL — „bez kategorii”), cena w formacie „1 149,00 zł”, stan. Produkty ze stanem 0 dostają klasę `brak`, a w kolumnie stan słowo „brak”.
5. Na końcu skryptu zamknij połączenie. Sprawdź stronę dla „wszystkie”, „Monitory” i „Drukarki” oraz po ręcznym wpisaniu w adresie `?kat=abc` (strona ma pokazać wszystkie produkty bez błędu).

4 Oddanie pracy

1. Zrób zrzut ekranu całej strony dla kategorii „Monitory” (`strona.png`) i sprawdź poprawność HTML w walidatorze `https://validator.w3.org/` (wklej kod źródłowy z przeglądarki).
2. Spakuj folder `sklep_www` razem z plikami `kwerendy.txt` i zrzutami do archiwum `nazwisko_r2_20.zip`.

W sprawozdaniu umieść

- ☐ plik `kwerendy.txt` z zapytaniami Z1–Z6 i zrzuty wyników
- ☐ pliki `produkty.php` i `styl.css`
- ☐ zrzut ekranu strony dla kategorii „Monitory”
- ☐ wynik walidacji HTML (zrzut lub informacja o błędach)
- ☐ odpowiedzi na pytania kontrolne

Pytania kontrolne

1. Czym różnią się `mysqli_fetch_assoc`, `mysqli_fetch_row` i `mysqli_fetch_array`?
2. Dlaczego w zapytaniu Z2 trzeba użyć LEFT JOIN, a w Z3 wystarczy JOIN?
3. Po co rzutowanie `(int)$GET['kat']` i zapytanie przygotowane, skoro w formularzu są tylko liczby?
4. Co się stanie, jeśli pominiesz `mysqli_set_charset($db, 'utf8mb4')`?

Kryteria oceny

Kryterium	Punkty
Zapytania Z1–Z6 ze zrzutami	6
Układ strony i CSS zgodne z opisem	4
Skrypt 1 i 2 (formularz z bazy, podsumowanie)	4
Skrypt 3 (tabela, filtr przez <code>mysqli_prepare</code> , klasa brak)	4
Odpowiedzi na pytania kontrolne	2
Razem	20

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