

ETAP 1 - Baza danych

Za pomocą narzędzia phpMyAdmin wykonaj następujące operacje na bazie danych:

- Utwórz bazę danych o nazwie sklep_internetowy, z zestawem polskich znaków (np. utf8_unicode_ci) – Do utworzonej bazy zainportuj tabele z pliku baza.sql

◆ Baza danych:

sklep_internetowy

◆ Tabele:

- uzytkownicy - kategorie - zakupy

◆ Relacje:

- zakupy.uzytkownik_id → uzytkownicy.id

- zakupy.kategoria_id → kategorie.id

Wykonaj zapytania SQL z punktu A i B. Wyniki zapisz w pliku **zapytania_SQL.sql**. Będą one dotyczyć między innymi Twoich tabel:

- **uzytkownicy** (id, imie, nazwisko, email, data_rejestracji)
- **kategorie** (id, nazwa, opis)
- **zakupy** (id, id_uzytkownika, id_kategorii, data_zakupu, cena, ilosc, opis)

CZĘŚĆ I - Funkcje daty i czasu

Zadanie 1:

Wyświetl wszystkie zakupy, które odbyły się dzisiaj.

Zadanie 2:

Wyświetl bieżącą godzinę.

Opis: Zwraca aktualną godzinę w formacie HH:MM:SS.

Zadanie 3:

Wyświetl aktualną datę i czas w jednym polu.

Opis: Zwraca bieżącą datę oraz godzinę.

Zadanie 4:

Wyświetl datę, która nastąpi 7 dni po dzisiejszej dacie.

Opis: Dodaje 7 dni do bieżącej daty.

Zadanie 5:

Wyświetl datę zakupu sprzed 3 dni dla każdego rekordu.

Opis: Pokazuje, jaka była data 3 dni przed faktycznym zakupem.

Zadanie 6:

Policz, ile dni minęło od daty zakupu każdego użytkownika do dzisiaj.

Opis: Oblicza różnicę w dniach między datą zakupu a dzisiejszą datą.

Zadanie 7:

Wyświetl zakupy w formacie daty: „dzień.miesiąc.rok”.

Opis: Formatuje datę do postaci z kropkami.

Zadanie 8:

Wyświetl tylko miesiąc (liczbę) z daty zakupu.

Opis: Zwraca numer miesiąca (1–12) z daty zakupu.

Zadanie 9:

Wyświetl ostatni dzień miesiąca dla każdej daty zakupu.

Opis: Pokazuje ostatni dzień miesiąca przypisanego do danej daty.

Zadanie 10:

Oblicz różnicę w godzinach między dwoma datami zakupu (przykładowo 1 i 2).

Opis: Zwraca liczbę godzin pomiędzy dwoma datami zakupów.

CZĘŚĆ II - Funkcje tekstowe

Zadanie 11:

Połącz imię i nazwisko użytkownika w jedno pole.

Opis: Łączy imię i nazwisko w jeden ciąg znaków.

Zadanie 12:

Połącz nazwę kategorii i opis oddzielając myślnikiem.

Opis: Łączy pola, dodając separator „-” pomiędzy nimi.

Zadanie 13:

Wyświetl długość imienia każdego użytkownika.

Opis: Zwraca liczbę znaków w kolumnie imie.

Zadanie 14:

Wyświetl wszystkie opisy produktów (z tabeli zakupy) małymi literami.

Opis: Zamienia wszystkie litery w kolumnie opis na małe.

Zadanie 15:

Wyświetl pierwsze 5 znaków z opisu zakupu.

Opis: Zwraca fragment tekstu od 1 do 5 znaku. Zadania raportowe (tabela zakupy)

Zadanie 16:

Wyświetl wszystkich użytkowników, którzy dokonali zakupów o łącznej wartości powyżej 500 zł.

Opis: Grupuje zakupy według użytkownika i pokazuje tylko tych, którzy wydali więcej niż 500 zł.

Zadanie 17:

Wyświetl średnią cenę produktu w każdej kategorii.

Opis: Oblicza średnią cenę z zakupów dla każdej kategorii.

Zadanie 18:

Pokaż kategorie, w których liczba zakupów przekracza 3 transakcje.

Opis: Liczy liczbę rekordów (zakupów) w każdej kategorii i filtryuje te, które mają więcej niż 3.

Zadanie 19:

Wyświetl najdroższy zakup w każdej kategorii, ale tylko jeśli kosztował więcej niż 100 zł.

Opis: Szuka największej ceny w każdej kategorii i zwraca tylko te, które przekraczają 100 zł.

Zadanie 20:

Wyświetl sumę zakupów w podziale na miesiące, tylko dla zakupów z bieżącego roku.

Opis: Tworzy miesięczny raport sprzedaży dla aktualnego roku kalendarzowego.

ETAP 2 - CECHY WITRYNY

- Składa się ze strony o nazwie sklep.php
- Zapisana w języku HTML5
- Zadeklarowany polski język zawartości witryny
- Jawnie zastosowany właściwy standard kodowania polskich znaków (UTF-8)
- Tytuł strony „SKLEP INTERNETOWY – PANEL ANALITYCZNY”
- Arkusz stylów w pliku styles.css prawidłowo połączony ze stroną
- Strona podzielona za pomocą semantycznych znaczników HTML5

Blok nagłówkowy <header></header> składa się:

Z dwóch wierszy (div, nav).

W pierwszym wierszu (div) jest przycisk. Poniżej opis działania funkcjonalności przycisku:



Przycisk posiada funkcje:



- Przełączenie na wygląd przycisku:
- Wyłączenie widoczności menu lewego
- Włączenie widoczności navbar znajdujący się pod spodem pierwszego wiersza w nagłówku



Przycisk posiada funkcje:



- Przełączenie na wygląd przycisku:
- Wyłączenie widoczności navbar znajdujący się pod spodem pierwszego wiersza w nagłówku
- Włączenie widoczności menu lewego

W pierwszym wierszu (div) jest także nagłówek H2: „SKLEP INTERNETOWY – ANALIZA DANYCH SQL” . Jest wycentrowany w pierwszym wierszu.

W pierwszym wierszu (div) jest także zegar + aktualna data i czas w UTF (JavaScript). Aktualny czas i data jest zawsze prezentowana - [zadanie na ocenę 6](#)

W drugim wierszu nav jest pętla z linkiem przekierowującym do odpowiedniego zadania SQL (ilustracja 1):

```
<a href="?zadanie=1"> Zadanie 1 </a>
```

W części głównej (znacznik semantyczny main) znajduje się

- menu lewe
- sekcja bloku głównego

Menu lewe zawiera listę linków. Linki są ułożone jeden pod drugim. W jednej kolumnie (ilustracja 1)

SKLEP INTERNETOWY – ANALIZA DANYCH SQL

12.11.2025 09:30:59

The screenshot shows a web application interface. On the left, there is a vertical sidebar containing a list of items from "Zadanie 1" to "Zadanie 20". The main content area has a header "Wynik zapytania: Zadanie 1" and a message "Brak wyników do wyświetlenia." Below the content area, there is a footer bar with the text "Autor: Imię Nazwisko – klasa X grupa X – przedmiot: Projekt SQL i PHP".

Ilustracja 1

SKLEP INTERNETOWY – ANALIZA DANYCH SQL

12.11.2025 09:36:59

The screenshot shows a web application interface. At the top, there is a horizontal navigation bar with items: "Zadanie 1", "Zadanie 2", "Zadanie 3", "Zadanie 4", "Zadanie 5", "Zadanie 6", "Zadanie 7", "Zadanie 8", "Zadanie 9", "Zadanie 10", "Zadanie 11", "Zadanie 12", "Zadanie 13", "Zadanie 14", "Zadanie 15", "Zadanie 16", "Zadanie 17", "Zadanie 18", "Zadanie 19", and "Zadanie 20". The main content area has a header "Wynik zapytania: Zadanie 1" and a message "Brak wyników do wyświetlenia." Below the content area, there is a footer bar with the text "Autor: Imię Nazwisko – klasa X grupa X – przedmiot: Projekt SQL i PHP".

Ilustracja 2

Menu lewe to 30% szerokości bloku głównego (main): Zawiera listę zadań od 1 do 20.
Ułożonych pionowo. Zadanie 1 Zadanie 2 ... Zadanie 20

Po kliknięciu w link prezentowany jest wynik zapytania w tabeli zgodnie z treścią zadania. Czyli link "Zadanie 1" prezentuje wynik zapytania SQL: "*Wyświetl wszystkie zakupy, które odbyły się dzisiaj.*" w bloku głównym.

Blok główny:

Wyświetla wyniki zapytań SQL w tabeli.

Do połączenia z bazą danych należy zastosować metodę:

```
mysqli_connect(host, username, password, database);
```

Najczęściej wartość dla host:

- "localhost" – jeśli baza działa lokalnie
- "127.0.0.1" – to samo, ale omija DNS
- "nazwa_serwera" – jeśli łączysz się zdalnie
- "adres_ip" – np. "192.168.1.10"

Dodać obsługę błędu połączenia z bazą danych używając metody **mysqli_error**

Sprawdzamy czy w adresie URL istnieje parametr **zadanie** za pomocą metody **isset**.

Następnie pobraną wartość z adresu URL konwertujemy do liczby całkowitej za pomocą metody **intval(\$_GET['zadanie'])**. Dla określonej wartości wykonujemy odpowiednie zadanie SQL (**mysqli_query(\$polaczenie, \$sql)**);

Korzystając z metody **mysqli_num_rows** sprawdzamy czy wynik jest większy od 0, jeśli tak wyświetlamy nagłówki tabeli:

```
$liczbaKolumn = mysqli_num_fields($wynik);
$naglowki = mysqli_fetch_fields($wynik);
foreach ($naglowki as $naglowek) {
    echo "<th>" . $naglowek->name . "</th>";
}
```

następnie pobieramy wiersze z tabeli (rekordy tabeli):

```
while ($wiersz = mysqli_fetch_array($wynik)) {
    echo "<tr>";
    foreach ($wiersz as $klucz => $wartosc) {
        // Pomijamy duplikaty w mysqli_fetch_array (numeryczne)
        if (is_numeric($klucz)) continue;

        echo "<td>$wartosc</td>";
    }
    echo "</tr>";
}
```

Stopka:

Wyśrodkowany tekst „Autor: Imię Nazwisko – klasa ... przedmiot ...”

ETAP 3 – CSS

- Styl CSS zdefiniowany jest w całości w zewnętrznym pliku o nazwie styles.css.
- Domyślnie dla wszystkich selektorów: krój czcionki: Roboto, w przypadku jego braku Arial, sans-serif,
- **Dla bloku nagłówkowego** tło ma korzystać z opcji:
 - o **background-image: linear-gradient(to right, #13cdcd, #17310a);**
- przyciski << lub >> w bloku nagłówkowym z lewej strony mają być na wysokości tytułu strony "SKLEP INTERNETOWY – ANALIZA DANYCH SQL".
- Z prawej strony na wysokości tytułu na być zegar z datą
- Menu lewe dla listy z i przycisków w nagłówku [*Zadanie 20*](#) ma mieć tło **background-color: rgb(53, 90, 53);** Czcionka koloru białego. Bez obramowania (**border: none;**) Dla [*Zadanie 20*](#) w menu lewym i menu głównym ma wynosić: **padding** ma wynosić **10px**; Kursor w właściwością **pointer (cursor: pointer)** Rozmiar czcionki ma wynosić 14 pikseli (font-size: 14px;) a obramowanie na listy linków w menu lewym i w nagłówku ma wynosić: border-radius: 5px;
- **Dla selektora ciała strony (bloku głównego):** kolor tła jednolitego koloru (dowolna propozycja)
- **Dla selektora stopki:** kolor tła jednolitego koloru (dowolna propozycja), biały kolor czcionki, marginesy wewnętrzne 10 px
- **Dla selektora tabeli:** wyśrodkowanie tabeli, szerokość 60%, obramowanie linią ciągłą o dowolnym kolorze i szerokości 1 px
- W momencie gdy kurSOR myszy znajdzie się na wierszu tabeli, jego kolor tła zmienia się na dowolny kolor, a kolor czcionki na biały
- Menu lewe ma mieć 30% szerokości strony.
- Blok główny ma mieć 70% szerokości strony w przypadku jak jest pokazane menu lewe. W przypadku ukrycia menu lewego blok główny ma mieć 100% szerokości
- Dane w bloku głównym mają się wyświetlać wyniki zapytań w tabeli

Dla szerokości ekranu 700px i mniej. W menu lewym linki mają się prezentować pod sobą (pionowo). Tak jak pokazuje **ilustracja 3**. Przy zamkniętym menu lewym ma się prezentować tak jak pokazuje **ilustracja 4** Jak Stopka na samym dole zawsze

700px × 897p

SKLEP INTERNETOWY – ANALIZA DANYCH SQL
13.11.2025 10:41:55

Zadanie 1

Zadanie 2

Zadanie 3

Zadanie 4

Zadanie 5

Zadanie 6

Zadanie 7

Zadanie 8

Zadanie 9

Zadanie 10

Zadanie 11

Zadanie 12

Zadanie 13

Zadanie 14

Zadanie 15

Zadanie 16

Zadanie 17

Zadanie 18

Zadanie 19

Zadanie 20

Wynik zapytania: Zadanie 4

komunikat
Brak zapytania dla tego numeru

Autor: Imię Nazwisko – klasa X grupa X – przedmiot: Projekt SQL i PHP

Ilustracja 3

SKLEP INTERNETOWY – ANALIZA DANYCH SQL

13.11.2025 10:48:14

Zadanie 1 Zadanie 2 Zadanie 3 Zadanie 4 Zadanie 5 Zadanie 6
Zadanie 7 Zadanie 8 Zadanie 9 Zadanie 10 Zadanie 11 Zadanie 12
Zadanie 13 Zadanie 14 Zadanie 15 Zadanie 16 Zadanie 17 Zadanie 18
Zadanie 19 Zadanie 20

Wynik zapytania: Zadanie 4

komunikat
Brak zapytania dla tego numeru

Autor: Imię Nazwisko – klasa X grupa X – przedmiot: Projekt SQL i PHP

Ilustracja 4

ETAP 4 – PHP

Skrypt połączenia z bazą

W tabeli 1 zamieszczono wybrane funkcje PHP do obsługi bazy danych.

Wymagania dotyczące skryptu:

Napisany w języku PHP

- Należy stosować znaczące nazewnictwo zmiennych i funkcji w języku polskim
- Łączy się z serwerem bazodanowym na *localhost*, użytkownik *root* bez hasła, baza danych o nazwie *sklep_internetowy*

- Wysyła do bazy danych zapytania od zadania nr 1 do 20.
- Zgodnie z klikniętym linkiem i mu przyporządkowanym zapytaniem SQL
- W każdym wierszu tabeli, w odpowiednich komórkach, wyświetlane są zwrocone zapytaniem kolejne wiersze z bazy – Na końcu działania skrypt zamknie połączenie z serwerem

Będziesz miała:

- połączenie
- wykonanie wybranego zapytania
- wyświetlenie wyników w tabeli
- zamknięcie połączenia

Tabela 1. Wybór funkcji języka PHP do obsługi bazy MySQL i MariaDB

Funkcje biblioteki mysql Zwracana wartość

mysqli_connect(serwer, użytkownik,hasło, nazwa_bazy) - id połączenia lub FALSE, gdy niepowodzenie

mysqli_select_db(id_polaczenia,nazwa_bazy) - TRUE/FALSE w zależności od stanu operacji

mysqli_error(id_polaczenia) - Tekst komunikatu błędu

mysqli_close(id_polaczenia) - TRUE/FALSE w zależności od stanu operacji

mysqli_query(id_polaczenia, zapytanie) - Wynik zapytania

mysqli_fetch_row(wynik_zapytania) - Tablica numeryczna odpowiadająca wierszowi zapytania

mysqli_fetch_array(wynik_zapytania) - Tablica zawierająca kolejny wiersz z podanych w wyniku zapytania lub FALSE, jeżeli nie ma więcej wierszy w wyniku zapytania

mysqli_num_rows(wynik_zapytania) - Liczba wierszy w podanym zapytaniu

mysqli_num_fields(wynik_zapytania) - Liczba kolumn w podanym zapytaniu

isset(\$zmienna) - Sprawdzenie, czy \$zmienna istnieje

Lista najczęściej używanych znaczników semantycznych w HTML5:

1. **<header>** – Definiuje nagłówek strony, sekcji lub artykułu, zazwyczaj zawierający logo, menu nawigacyjne lub tytuły.
2. **<nav>** – Określa sekcję nawigacyjną, zawierającą linki do innych stron lub części dokumentu.
3. **<main>** – Reprezentuje główną treść dokumentu, unikalną dla danej strony (powinna występować tylko raz).
4. **<article>** – Oznacza niezależną, samodzielnią treść, taką jak wpis na blogu, artykuł czy post.
5. **<section>** – Grupuje powiązane tematycznie treści, zwykle z nagłówkiem (np. `<h2>`).
6. **<aside>** – Zawiera treści poboczne, takie jak panele boczne, reklamy czy dodatkowe informacje.
7. **<footer>** – Definiuje stopkę strony lub sekcji, zawierającą np. informacje kontaktowe, prawa autorskie.
8. **<figure>** – Służy do grupowania multimedialnych (np. obrazów, diagramów) z opcjonalnym podpisem.
9. **<figcaption>** – Podpis dla elementu **<figure>**, opisujący zawartość multimedialną.
10. **<details>** – Tworzy interaktywny element, który można rozwinąć/zwinąć, aby pokazać dodatkowe informacje.
11. **<summary>** – Definiuje nagłówek dla elementu **<details>**, widoczny przed rozwinięciem.
12. **<mark>** – Wyróżnia tekst, który jest istotny w danym kontekście (np. wyniki wyszukiwania).
13. **<time>** – Oznacza datę, godzinę lub zakres czasowy, z opcjonalnym atrybutem `datetime` .
14. **<address>** – Służy do oznaczania informacji kontaktowych, np. adresu e-mail, telefonu czy lokalizacji.
15. **<progress>** – Reprezentuje pasek postępu, np. dla ładowania lub wypełnienia formularza.
16. **<meter>** – Wskazuje wartość w określonym zakresie, np. poziom naładowania baterii.
17. **<dialog>** – Definiuje okno dialogowe lub modalne, np. do wyświetlania alertów.
18. **<picture>** – Umożliwia definiowanie różnych źródeł obrazów dla różnych urządzeń lub rozdzielczości.
19. **<template>** – Przechowuje treść, która nie jest wyświetlana od razu, ale może być użyta przez JavaScript.

Znaczniki pomagają w lepszej organizacji kodu, poprawiają dostępność (accessibility) i optymalizację dla wyszukiwarek (SEO).