

Próbnny egzamin INF.03 — baza danych i strona PHP (kino „Kadr”)

Czas: 2 × 45 min

praca samodzielna, warunki egzaminacyjne

Cel: Wykonanie w 90 minut kompletnego zadania egzaminacyjnego: utworzenie bazy z pliku, zapisanie czterech zapytań ze zrzutami oraz zbudowanie strony internetowej z arkuszem stylów i skryptem PHP pobierającym dane z bazy przez mysqli.

Po zajęciach potrafisz:

- importuje bazę danych i wykonuje zapytania SELECT (z WHERE, ORDER BY, JOIN, GROUP BY) oraz UPDATE
- dokumentuje zapytania zrzutami ekranu i plikiem kwerendy.txt
- buduje stronę z sekcjami header, main (dwa panele) i footer zgodnie z opisem wyglądu
- pisze skrypt PHP: mysqli_connect, mysqli_query, pętla mysqli_fetch_assoc, mysqli_close
- sprawdza pracę z kryteriami oceny w stylu egzaminacyjnym

Sprzęt i materiały

- komputer z XAMPP (Apache, MariaDB, PHP 8) i przeglądarką
- pliki `r2_32_kino.sql` i `r2_32_kadr.png` z folderu Pliki_do_zadan
- bez dostępu do internetu i materiałów (jak na egzaminie)

Oprogramowanie

- **XAMPP** — uruchom moduły Apache i MySQL; pliki strony w `C:\xampp\htdocs\kino`
[apachefriends.org](http://www.apachefriends.org)
- **phpMyAdmin** — `http://localhost/phpmyadmin`
- **Edytor kodu (VS Code, Notepad++)** — bez wtyczek generujących kod
code.visualstudio.com

Przygotowanie przed zajęciami

- Powtórz z lekcji 1.18 i 2.13 budowę skryptu PHP z mysqli oraz typowe układy stron z egzaminu (baner, panele, stopka).

Bezpieczeństwo

- Konto root bez hasła stosujemy tylko na lokalnym serwerze ćwiczeniowym, tak jak na egzaminie. W prawdziwej aplikacji używa się osobnego konta z hasłem (zadanie 2.19).
- Przed zapytaniem UPDATE (kw4) upewnij się, że ma poprawny warunek WHERE — po pomyłce zaimportuj plik `r2_32_kino.sql` ponownie.

Przebieg zadania

1 Przygotowanie stanowiska (5 min)

Utwórz na pulpicie folder `nazwisko_imie` (bez polskich znaków). Zapisuj w nim wszystkie wyniki pracy.

1. Uruchom Apache i MySQL w XAMPP. W phpMyAdmin zaimportuj plik `r2_32_kino.sql` — sam tworzy bazę `kino` z tabelami `filmy` i `seanse` (relacja: `seanse.id_filmu` → `filmy.id`).
2. Skopiuj plik `r2_32_kadr.png` do folderu `C:\xampp\htdocs\kino` pod nazwą `kadr.png`.

2 Operacje na bazie danych (25 min)

Każde zapytanie wykonaj w phpMyAdmin, zrób zrzut ekranu wyniku (cały wynik i treść zapytania) i zapisz jako `kw1.png` ... `kw4.png`. Treść zapytań zapisz w pliku `kwerendy.txt`, każde zakończone średnikiem.

1. Z1 (kw1). Wybierz tytuł, rok produkcji i ocenę filmów z gatunku komedia. Posortuj malejąco według roku.
2. Z2 (kw2). Wybierz godzinę seansu, tytuł filmu i numer sali dla seansów z 14 listopada 2026 r. Posortuj rosnąco według godziny.
3. Z3 (kw3). Dla każdego gatunku wybierz jego nazwę i liczbę filmów. Kolumnę z liczbą nazwij `liczba_filmow`.
4. Z4 (kw4). Obniż o 4 zł cenę wszystkich seansów rozpoczynających się przed godziną 14:00. Na zrzucie ma być widoczna liczba zmienionych wierszy.

3 Strona internetowa (50 min)

Utwórz w folderze `C:\xampp\htdocs\kino` pliki `kino.php` i `styl.css`. Strona ma być zgodna ze standardem HTML5, z językiem polskim i kodowaniem UTF-8; tytuł karty: „Kino Kadr — repertuar”.

1. Z5. Struktura: baner (header) z nagłówkiem pierwszego stopnia „Kino studyjne KADR”; część główna z dwoma panelami — lewym i prawym; stopka (footer) z akapitem „Stronę wykonał: ” i twoim imieniem i nazwiskiem.
2. Z6. Panel lewy: obraz `kadr.png` z tekstem alternatywnym „Logo kina Kadr”, nagłówek drugiego stopnia „Cennik” i lista nienumerowana z trzema punktami: „bilet normalny — według seansu”, „bilet ulgowy — 20% taniej”, „seanse przed 14:00 — taniej o 4 zł”.
3. Z7. Panel prawy: nagłówek drugiego stopnia „Repertuar na 14 listopada 2026” i skrypt PHP, który łączy się z bazą `kino` (serwer `localhost`, użytkownik `root`, bez hasła), wysyła zapytanie z `kw2` rozszerzone o cenę biletu i dla każdego seansu wypisuje akapit w postaci: godzina pogrubiona (tylko gg:mm), spacja, tytuł, „ — sala ”, numer sali, „, „, cena, „ zł”. Na końcu skrypt zamyka połączenie.
4. Z8. Styl CSS: wszystkie elementy krojem Verdana; tło strony `#F4F1EA`; baner i stopka — tło `#1B2A41`, biały kolor czcionki, wyrównanie baneru do środka, stopki do prawej, marginesy wewnętrzne 10 px; panel lewy — szerokość 30%, wysokość 500 px, tło `#E9C46A`; panel prawy — szerokość 70%, wysokość 500 px, tło białe (panele obok siebie, margines wewnętrzny 10 px nie może psuć układu); obraz — szerokość 90%, wyśrodkowany, zaokrąglone rogi 10 px; nagłówki drugiego stopnia — kolor `#1B2A41` i dolne obramowanie 2 px ciągłe w tym kolorze; akapit w panelu prawym po najechaniu kursorem — tło `#E9C46A`.
5. Z9. Zrób zrzut całej strony w przeglądarce (`http://localhost/kino/kino.php`) i zapisz jako `strona.png`. Skopiuj `kino.php`, `styl.css` i `kadr.png` do folderu `nazwisko_imie`.

4 Samoocena (10 min)

1. Porównaj pracę z kryteriami w sprawozdaniu (nazwy plików, średniki w kwerendy.txt, kompletność zrzutów, poprawność kodu w walidatorze — jeśli nauczyciel na to pozwoli po zakończeniu części egzaminacyjnej). Wypełnij tabelę samooceny.

W sprawozdaniu umieść

- ☐ folder `nazwisko_imie` z plikami kw1.png–kw4.png, kwerendy.txt, kino.php, styl.css, kadr.png i strona.png
- ☐ tabela samooceny z punktami za każdy element
- ☐ lista błędów, które popełniłeś, i jak je poprawisz na egzaminie

Pytania kontrolne

1. Dlaczego w zapytaniu kw2 trzeba złączyć tabele i jak zapisać warunek złączenia?
2. Co zwraca `mysqli_fetch_assoc` i dlaczego pętla `while` kończy się po ostatnim wierszu?
3. Jak w CSS ustawić dwa panele obok siebie, aby stopka znalazła się pod nimi?
4. Jak w PHP wyświetlić tylko gg:mm z wartości 11:00:00 i jak to samo zrobić w SQL?

Kryteria oceny

Kryterium	Punkty
Zapytania kw1–kw4: poprawne wyniki, zrzuty, plik kwerendy.txt	6
Struktura strony HTML (Z5–Z6): sekcje, nagłówki, obraz z alt, lista, stopka	4
Skrypt PHP (Z7): połączenie, zapytanie, pętla z formatem wyniku, zamknięcie połączenia	5
Styl CSS (Z8) zgodny z opisem	4
Samoocena i komplet plików w folderze	1
Razem	20

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