

Środowisko pracy: XAMPP, phpMyAdmin i konsola

Czas: 2 × 45 min

praca indywidualna

Cel: Uruchomienie serwera bazy danych z pakietu XAMPP, zaimportowanie bazy ćwiczeniowej sklep oraz poznanie dwóch narzędzi pracy: phpMyAdmin i konsoli `mysql`.

Po zajęciach potrafisz:

- uruchamia i zatrzymuje Apache oraz MySQL/MariaDB w panelu sterowania XAMPP
- importuje bazę z pliku `.sql` w phpMyAdmin i w konsoli
- przegląda bazy, tabele i ich strukturę poleceniami `SHOW`, `DESCRIBE` i `SELECT`
- odczytuje wersję serwera, port i położenie plików konfiguracyjnych
- wyjaśnia, dlaczego puste hasło konta root jest dopuszczalne tylko na komputerze lokalnym

Sprzęt i materiały

- komputer z systemem Windows 11 i zainstalowanym pakietem XAMPP (lub uprawnieniami do instalacji)
- plik `sklep.sql` z folderu Pliki_do_zadan (od nauczyciela)

Oprogramowanie

- **XAMPP dla Windows** — Apache, MariaDB 10.4, PHP i phpMyAdmin w jednym pakiecie; domyślny katalog `C:\xampp` apachefriends.org/download.html
- **phpMyAdmin** — w XAMPP dostępny pod adresem `http://localhost/phpmyadmin` phpmyadmin.net
- **Konsola mysql (klient MariaDB)** — plik `C:\xampp\mysql\bin\mysql.exe` albo przycisk Shell w panelu XAMPP mariadb.com/kb/en/mariadb-command-line-client

Przygotowanie przed zajęciami

- Skopiuj plik `sklep.sql` do folderu bez spacji i polskich znaków w ścieżce, np. `C:\Pliki_do_zadan\sklep.sql`.
- Utwórz plik tekstowy `zad01_nazwisko.sql` (Notatnik lub VS Code) — będziesz w nim zapisywać wszystkie polecenia Z1–Z14.

Bezpieczeństwo

- Konto root w XAMPP domyślnie nie ma hasła — to wygoda na komputerze uczniowskim, ale nigdy nie udostępniaj takiego serwera w sieci (nie otwieraj portu 3306 w zaporze, nie zmieniaj adresu nasłuchu).
- Nie zmieniaj hasła root na komputerze szkolnym bez polecenia nauczyciela — phpMyAdmin w XAMPP loguje się pustym hasłem i przestanie działać.
- Import pliku `sklep.sql` usuwa i tworzy od nowa bazę sklep — nie importuj go, jeśli w tej bazie masz zmiany, które chcesz zachować.

Przebieg zadania

1 Uruchomienie serwerów

Panel sterowania XAMPP (xampp-control.exe).

1. Uruchom panel sterowania XAMPP i kliknij Start przy modułach Apache i MySQL. Oba wiersze powinny podświetlić się na zielono, a w kolumnie Port(s) pojawić się numery portów. Zrób zrzut ekranu.
2. Jeśli MySQL się nie uruchamia, odczytaj komunikat w dolnym oknie dziennika (najczęściej port 3306 zajęty przez inny serwer) i zgłoś go nauczycielowi.
3. Kliknij przycisk Config przy module MySQL i wybierz `my.ini`. Odszukaj w pliku linie `port=` i `datadir=` i zapisz ich wartości. Zamknij plik bez zmian.

2 Import bazy w phpMyAdmin

Przeglądarka: <http://localhost/phpmyadmin>

1. Otwórz phpMyAdmin i wybierz zakładkę Import. Wskaż plik `sklep.sql` i kliknij przycisk importu na dole strony.
2. Po komunikacie o powodzeniu rozwiń w lewym panelu bazę sklep. Zapisz nazwy tabel i liczbę wierszy w każdej (zakładka Struktura bazy pokazuje kolumnę Wiersze).
3. Kliknij tabelę produkty i obejrzyj zakładki Przeglądaj i Struktura. Zrób zrzut ekranu struktury tabeli.
4. Otwórz zakładkę SQL i wykonaj Z1: `SELECT VERSION();` — zapisz wersję serwera.

3 Konsola mysql

W panelu XAMPP kliknij Shell albo uruchom Wiersz polecenia (cmd) i przejdź do folderu klienta.

```
cd C:\xampp\mysql\bin
mysql -u root --default-character-set=utf8mb4

MariaDB [(none)]> \s
MariaDB [(none)]> exit
```

1. Zaloguj się do konsoli jak w ramce. Wpisz `\s` (status) i zapisz: wersję serwera, użytkownika, port i kodowanie połączenia. Zwróć uwagę, że nie podałeś hasła.
2. Wykonaj kolejno i zapisz w pliku .sql: Z2 — wyświetl listę baz danych; Z3 — wybierz bazę sklep (`USE`) i wyświetl listę jej tabel; Z4 — wyświetl strukturę tabeli produkty; Z5 — wyświetl strukturę tabeli klienci.
3. Z6 — wyświetl całą zawartość tabeli kategorie. Z7 — wyświetl całą zawartość tabeli produkty. Z8 — wyświetl całą zawartość tabeli klienci. Sprawdź, czy polskie znaki (np. „Wiśniewska”) wyświetlają się poprawnie.
4. Z9 — policz wiersze tabeli zamowienia poleceniem `SELECT COUNT(*) FROM zamowienia;`. Porównaj wynik z phpMyAdmin.
5. Każde polecenie SQL kończ średnikiem. Jeśli konsola pokazuje znak zachęty `->`, oznacza to, że czeka na dokończenie polecenia — dopisz `;` i naciśnij Enter.

4 Konfiguracja i konta

Odczyt ustawień serwera poleceniami SQL (konsola lub zakładka SQL w phpMyAdmin).

1. Z10 — `SHOW VARIABLES LIKE 'port';` oraz Z11 — `SHOW VARIABLES LIKE 'datadir';`. Porównaj z wartościami odczytanymi z `my.ini`.
2. Z12 — sprawdź domyślne kodowanie i porównywanie znaków serwera: `SELECT @@character_set_server, @@collation_server;`. Porównaj z kodowaniem bazy sklep (phpMyAdmin → sklep → Operacje, pole Metoda porównywania napisów).
3. Z13 — wyświetl listę kont serwera: `SELECT User, Host FROM mysql.user;`. Zapisz, ile jest kont o nazwie root i z jakich hostów mogą się logować.
4. Z14 — zaimportuj bazę ponownie, tym razem w konsoli: `SOURCE C:/Pliki_do_zadan/sklep.sql` (ukośniki zwykłe). Sprawdź poleceniem z Z9, że liczba zamówień się nie zmieniła.
5. Zapisz plik `zad01_nazwisko.sql` ze wszystkimi poleceniami Z1–Z14 i oddaj go razem ze sprawozdaniem.

W sprawozdaniu umieść

- ☐ zrzut ekranu panelu XAMPP z uruchomionymi modułami Apache i MySQL
- ☐ zrzut struktury tabeli produkty w phpMyAdmin i zrzut wyniku Z4 w konsoli
- ☐ tabela zapytań Z1–Z14 z wynikami
- ☐ wersja serwera, port, katalog danych i ścieżka do pliku `my.ini`
- ☐ odpowiedzi na pytania kontrolne
- ☐ plik `zad01_nazwisko.sql`

Pytania kontrolne

1. Czym różni się serwer bazy danych (MariaDB/MySQL) od narzędzi phpMyAdmin i konsoli mysql?
2. Dlaczego do działania phpMyAdmin potrzebny jest uruchomiony Apache, a do działania konsoli — nie?
3. Co oznacza zapis root@localhost i dlaczego konto root bez hasła jest groźne na serwerze dostępnym z sieci?
4. Do czego służy polecenie DESCRIBE, a do czego SELECT * FROM? Podaj przykład z bazy sklep.
5. Dlaczego na serwerze produkcyjnym nie importuje się plików .sql zawierających DROP DATABASE bez wcześniejszej kopii?

Kryteria oceny

Kryterium	Punkty
Uruchomione środowisko i poprawny import bazy sklep (zrzuty)	4
Poprawne polecenia Z1–Z9 w konsoli i phpMyAdmin z wynikami	6
Odczyt konfiguracji: port, katalog danych, kodowanie, konta (Z10–Z13)	4
Import w konsoli (Z14) i kompletny plik .sql	2
Odpowiedzi na pytania kontrolne	4
Razem	20

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

Analiza wymagań i projekt bazy szkolnej biblioteki

Czas: 2 × 45 min

praca w parach (sprawozdanie każdy oddaje osobno)

Cel: Przełożenie opisu potrzeb klienta na listę encji, atrybutów i reguł biznesowych oraz pierwszą propozycję tabel — jeszcze bez SQL.

Po zajęciach potrafisz:

- wyodrębnia z opisu słownego encje i ich atrybuty
- formułuje reguły biznesowe i wskazuje, jak wpływają na projekt
- przygotowuje pytania doprecyzowujące wymagania klienta
- tworzy słownik danych z typem ogólnym, wymagalnością i przykładem
- proponuje zestaw tabel z kolumnami

Sprzęt i materiały

- komputer z edytorem tekstu lub arkuszem kalkulacyjnym
- kartki i kolorowe pisaki do szkicowania (opcjonalnie)

Oprogramowanie

- **LibreOffice Calc lub Microsoft Excel** — słownik danych w arkuszu (opcjonalnie — można pisać w tabelach sprawozdania) libreoffice.org/download/download-libreoffice

Przygotowanie przed zajęciami

- Przypomnij sobie z prezentacji 1.02 pojęcia: encja, atrybut, reguła biznesowa, słownik danych.

Bezpieczeństwo

- Projektując dane osobowe, zbieraj tylko to, co jest potrzebne do działania biblioteki (zasada minimalizacji danych z RODO) — np. numer PESEL nie jest bibliotece potrzebny.

Przebieg zadania

1 Wywiad z klientem — notatka

Przeczytaj notatkę z rozmowy z bibliotekarką szkolną. To jedyne źródło wymagań; tam, gdzie czegoś brakuje, zapytasz klienta (krok 3).

Biblioteka ma ok. 6000 książek podzielonych na działy (lektury, informatyka, beletrystyka...).

Każdy tytuł ma ISBN, tytuł, rok wydania, wydawnictwo i jednego lub kilku autorów. Tego samego tytułu mamy często kilka egzemplarzy. Każdy egzemplarz ma własny numer inwentarzowy naklejony na grzbiecie i stan: dobry, zniszczony albo wycofany. Czytelnikami są uczniowie (z klasą, np. 2TI) i nauczyciele. Każdy dostaje kartę z numerem. Uczeń może mieć naraz najwyżej 3 egzemplarze, nauczyciel 10. Wypożyczenie trwa 30 dni, można je raz przedłużyć o 14 dni. Zapisuję datę wypożyczenia, termin zwrotu i datę faktycznego zwrotu. Chcę szybko widzieć, kto przetrzymuje książki, i znać historię wypożyczeń każdego egzemplarza.

Gdy wszystkie egzemplarze tytułu są wypożyczone, czytelnik może zarezerwować tytuł. Przed końcem roku absolwenci muszą oddać wszystkie książki.

1. Podkreśl w notatce (lub wypisz) wszystkie rzeczowniki. Zastanów się, które z nich to osobne obiekty z własnymi cechami (encje), a które są tylko cechami innych obiektów (atrybuty).
2. Zwróć uwagę na różnicę między tytułem książki a egzemplarzem — to dwie różne encje. Zapisz, po czym to poznałeś.

2 Encje, atrybuty i reguły

Wypełniaj table we wzorze sprawozdania.

1. Wypisz 6–9 encji. Dla każdej podaj krótki opis i jeden przykładowy obiekt (np. Egzemplarz — nr inw. 004512, „Pan Tadeusz”, stan dobry).
2. Dla każdej encji wypisz atrybuty. Zaznacz atrybut (lub atrybuty), który jednoznacznie identyfikuje obiekt, albo zapisz, że trzeba będzie dodać sztuczny identyfikator.
3. Wypisz co najmniej 6 reguł biznesowych z notatki (np. „Uczeń może mieć naraz najwyżej 3 egzemplarze”). Przy każdej zapisz, czy da się ją zapisać w strukturze bazy (klucz, wymagane pole, lista wartości), czy będzie ją pilnować aplikacja.
4. Zapisz, które informacje da się wyliczyć i nie trzeba ich przechowywać (np. czy wypożyczenie jest przetrzymane — z terminu zwrotu i dzisiejszej daty).

3 Pytania do klienta

Notatka nie odpowiada na wszystko. Nauczyciel odgrywa rolę bibliotekarki i odpowiada na pytania.

1. Przygotujcie w parze co najmniej 4 pytania doprecyzowujące (np. o dane kontaktowe czytelników, o to, co dzieje się z klasą ucznia po roku, o czas ważności rezerwacji).
2. Zadajcie pytania nauczycielowi i zapiszcie odpowiedzi. Uzupełnijcie listę encji, atrybutów i reguł o nowe informacje.

4 Słownik danych i propozycja tabel

1. Uzupełnij słownik danych dla co najmniej 14 atrybutów: encja, atrybut, typ ogólny (tekst, liczba całkowita, kwota, data, wybór z listy, tak/nie), czy wymagany, czy unikalny, przykład. Typy SQL dobierzesz w zadaniu 8.
2. Zaproponuj table: nazwa tabeli i lista kolumn. Jeśli jakiś atrybut przyjmuje wiele wartości (np. kilku autorów), zastanów się, czy nie potrzebna jest osobna tabela, i zapisz pomysł.
3. Sprawdź propozycję na trzech scenariuszach: (a) uczeń 2TI wypożycza drugi egzemplarz „Lalki”, (b) bibliotekarka szuka przetrzymanych książek, (c) egzemplarz zostaje wycofany. Czy każda potrzebna informacja ma swoje miejsce w tabelach?
4. Zachowaj wynik — w zadaniu 3 narysujesz z niego diagram ERD.

W sprawozdaniu umieść

- ☐ tabela encji z opisami i przykładami
- ☐ tabela reguł biznesowych z oceną, gdzie będą pilnowane
- ☐ pytania do klienta i uzyskane odpowiedzi
- ☐ słownik danych (min. 14 atrybutów)
- ☐ propozycja tabel i sprawdzenie na trzech scenariuszach
- ☐ odpowiedzi na pytania kontrolne

Pytania kontrolne

1. Czym różni się encja od atrybutu? Podaj przykład z biblioteki, w którym ta sama informacja mogłaby być jednym albo drugim.
2. Dlaczego tytuł i egzemplarz książki trzeba przechowywać osobno? Co by się stało, gdyby była jedna tabela?
3. Podaj przykład reguły biznesowej, której nie da się wymusić samą strukturą tabel.
4. Dlaczego nie warto przechowywać informacji, którą da się wyliczyć (np. liczby dni przetrzymania)?

Kryteria oceny

Kryterium	Punkty
Trafnie wyodrębnione encje (w tym rozróżnienie tytuł / egzemplarz)	4
Atrybuty i identyfikatory encji	3
Reguły biznesowe z oceną sposobu wymuszenia	3
Pytania do klienta i uwzględnienie odpowiedzi	2
Słownik danych	3
Propozycja tabel sprawdzona na scenariuszach	2
Odpowiedzi na pytania kontrolne	3
Razem	20

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

Diagram ERD — encje, atrybuty i związki 1:N

Czas: 2 × 45 min

praca indywidualna

Cel: Narysowanie diagramów ERD w notacji kurzej stopki z poprawnymi krotnościami i uczestnictwem — najpierw dla istniejącej bazy sklep, potem dla własnego projektu biblioteki.

Po zajęciach potrafisz:

- rysuje encje z atrybutami i kluczami w programie draw.io
- oznacza związki 1:N w notacji kurzej stopki z krotnością i uczestnictwem (obowiązkowe / opcjonalne)
- odczytuje związki istniejącej bazy z jej struktury i z information_schema
- uzasadnia uczestnictwo na podstawie reguł biznesowych i danych

Sprzęt i materiały

- komputer z XAMPP (baza sklep zaimportowana w zadaniu 1)
- wyniki zadania 2 (encje i reguły biblioteki)

Oprogramowanie

- **draw.io (diagrams.net)** — wersja w przeglądarce app.diagrams.net albo desktopowa; biblioteka kształtów Entity Relation drawio.com
- **phpMyAdmin lub konsola mysql** — odczyt struktury bazy sklep phpmyadmin.net

Przygotowanie przed zajęciami

- Przynieś (lub otwórz) swoje wyniki z zadania 2: listę encji, atrybutów i reguł biznesowych.

Przebieg zadania

1 Notacja kurzej stopki — przypomnienie

Każdy koniec linii związku ma dwa znaki: bliżej encji — krotność maksymalną (kreska = jeden, kurza stopka = wiele), dalej — minimalną (kreska = co najmniej jeden, kółko = zero, czyli uczestnictwo opcjonalne).

```
Klient ||-----o< Zamówienie
każde zamówienie ma dokładnie jednego klienta (|| przy Kliencie)
klient może mieć zero lub wiele zamówień (o< przy Zamówieniu)
```

1. W draw.io utwórz nowy pusty diagram. Kliknij „+ Więcej kształtów” (More Shapes) na dole lewego panelu i włącz bibliotekę Entity Relation.
2. Przeciągnij kształt tabeli (Table) i sprawdź, jak dodać wiersz z atrybutem. Klucz główny oznacz PK, klucz obcy FK.
3. Narysuj linię między dwiema tabelami i w panelu Format (Styl) ustaw zakończenia linii na znaki ERmandOne, ERzeroToMany itp. Znajdź wszystkie cztery warianty: ||, o|, |<, o<.

2 Część A — odczyt struktury bazy sklep

Wykonuj zapytania w phpMyAdmin (zakładka SQL bazy sklep) albo w konsoli. Zapisuj je w pliku `zad03_nazwisko.sql`.

```
SELECT TABLE_NAME, COLUMN_NAME, REFERENCED_TABLE_NAME, REFERENCED_COLUMN_NAME
FROM information_schema.KEY_COLUMN_USAGE
WHERE TABLE_SCHEMA = 'sklep' AND REFERENCED_TABLE_NAME IS NOT NULL;
```

1. Z1 — wyświetl strukturę tabel kategorii, produkty, klienci, zamówienia i pozycje_zamowienia poleceniem DESCRIBE (pięć poleceń). Zapisz klucze główne (PRI) i kolumny oznaczone MUL.
2. Z2 — wykonaj zapytanie z ramki. Pokazuje ono wszystkie klucze obce bazy: która kolumna w której tabeli wskazuje na którą tabelę. Zapisz wynik w tabeli „Związki” sprawozdania (pomiń na razie tabelę pracownicy).
3. Z3 — sprawdź, które kolumny kluczy obcych dopuszczają NULL: wyświetl z `information_schema.COLUMNS` kolumny TABLE_NAME, COLUMN_NAME, IS_NULLABLE dla bazy sklep i kolumn id_kategorii oraz id_klienta, które nie są kluczem głównym (warunek `COLUMN_KEY <> 'PRI'`). Wzoruj się na zapytaniu z ramki.
4. Z4 — policz, ilu różnych klientów złożyło zamówienia: `SELECT COUNT(DISTINCT id_klienta) FROM zamowienia;` i porównaj z liczbą klientów. Z5 — w ten sam sposób policz różne produkty w tabeli pozycje_zamowienia i porównaj z liczbą produktów.

3 Część A — diagram sklep

Na podstawie Z1–Z5 narysuj diagram pięciu tabel: kategorie, produkty, klienci, zamówienia, pozycje_zamowienia.

1. Narysuj encje z wszystkimi atrybutami; oznacz PK i FK.
2. Narysuj cztery związki 1:N. Uczestnictwo ustal tak: kolumna FK dopuszczająca NULL oznacza, że po stronie „jeden” jest kółko (o|); wynik Z4 i Z5 pokazuje, czy po stronie „wiele” powinno być kółko.
3. Przy każdym związku dopisz czasownik w obu kierunkach (np. „składa” / „jest złożone przez”).
4. Zapisz diagram jako `zad03_sklep_nazwisko.drawio` i wyeksportuj do PNG (Plik → Eksportuj jako → PNG).

4 Część B — diagram biblioteki

Na podstawie zadania 2 narysuj diagram biblioteki szkolnej z samymi związkami 1:N. Związek książki–autorzy zaznacz na razie zwykłą linią z podpisem „N:M — zad. 4”.

1. Narysuj encje co najmniej: Dział, Wydawnictwo, Książka, Egzemplarz, Czytelnik, Wypożyczenie, Rezerwacja — z atrybutami i kluczami.
2. Ustal krotności i uczestnictwo każdego związku na podstawie reguł z zadania 2. Zapisz uzasadnienie dla trzech wybranych związków (np. „Książka ma zero lub wiele egzemplarzy, bo nowy tytuł można dodać przed dostawą”).
3. Sprawdź, czy Wypożyczenie łączy się z Egzemplarzem, a Rezerwacja z Książką (tytułem) — i dlaczego tak.
4. Zapisz plik `zad03_biblioteka_nazwisko.drawio` i wyeksportuj PNG. Zachowaj go do zadania 4.

W sprawozdaniu umieść

- ☐ tabela związków bazy sklep odczytana z Z2 z krotnościami i uczestnictwem
- ☐ wyniki Z3–Z5 i wnioski o uczestnictwie
- ☐ diagram ERD bazy sklep (PNG)
- ☐ diagram ERD biblioteki (PNG) z uzasadnieniem trzech związków
- ☐ odpowiedzi na pytania kontrolne

□ pliki .drawio i plik zad03_nazwisko.sql

Pytania kontrolne

1. Co oznacza zapis o< przy encji Zamówienie, a co || przy encji Klient?
2. Dlaczego klucz obcy umieszcza się zawsze po stronie „wiele” związku 1:N?
3. Po czym w strukturze tabeli poznasz, że uczestnictwo po stronie „jeden” jest opcjonalne?
4. Baza nie wymusza, żeby zamówienie miało co najmniej jedną pozycję. Jak to zaznaczysz na diagramie i kto powinien pilnować tej reguły?

Kryteria oceny

Kryterium	Punkty
Zapytania Z1–Z5 i poprawna tabela związków bazy sklep	4
Diagram sklep: encje, atrybuty, PK/FK	3
Diagram sklep: poprawne krotności i uczestnictwo	3
Diagram biblioteki z uzasadnieniem związków	6
Odpowiedzi na pytania kontrolne	4
Razem	20

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

ERD — związki N:M, tabela pośrednicząca i związek rekurencyjny

Czas: 2 × 45 min

praca indywidualna

Cel: Rozbijanie związków wiele-do-wielu na tabelę pośredniczącą, rozpoznawanie encji słabych i związków rekurencyjnych oraz samodzielne zaprojektowanie diagramu dla nowej dziedziny.

Po zajęciach potrafisz:

- rozpoznaje związek N:M i zamienia go na dwa związki 1:N z tabelą pośredniczącą
- umieszcza atrybuty związku (np. data zapisu) w tabeli pośredniczącej
- rozpoznaje encję słabą i jej klucz złożony
- rysuje związek rekurencyjny (pracownik — przełożony)
- projektuje kompletny diagram ERD dla nowej dziedziny

Sprzęt i materiały

- komputer z XAMPP i bazą sklep
- pliki .drawio z zadania 3

Oprogramowanie

- **draw.io (diagrams.net)** — biblioteka kształtów Entity Relation
drawio.com
- **phpMyAdmin** lub **konsola mysql**
phpmyadmin.net

Przebieg zadania

1 N:M w bazie sklep — jak to zrobiono

Zamówienie zawiera wiele produktów, a produkt występuje w wielu zamówieniach. Zapisz zapytania w pliku `zad04_nazwisko.sql`.

1. Z1 — wyświetl polecenie tworzące tabelę `pozycje_zamowienia`: `SHOW CREATE TABLE pozycje_zamowienia;`. Zapisz, z jakich kolumn składa się klucz główny i jakie są klucze obce.
2. Z2 — wyświetl wszystkie pozycje zamówienia nr 5 (`SELECT * FROM pozycje_zamowienia WHERE id_zamowienia = 5;`). Z3 — w ten sam sposób wyświetl wszystkie pozycje z produktem nr 7. Wyjaśnij na tych wynikach, dlaczego związek zamówienia–produkty jest N:M.
3. Odpowiedz: dlaczego kolumny `ilosc` i `cena_jedn` są w tabeli pośredniczącej, a nie w tabeli produkty ani zamówienia?

2 Związek rekurencyjny — pracownicy

Tabela `pracownicy` ma klucz obcy `id_przezonego` wskazujący na tę samą tabelę.

1. Z4 — wyświetl całą tabelę `pracownicy`. Na kartce narysuj schemat organizacyjny (kto komu podlega).
2. Z5 — wyświetl imię, nazwisko i stanowisko pracowników, których przełożonym jest pracownik nr 2. Z6 — wyświetl pracownika, który nie ma przełożonego (warunek `IS NULL`).
3. Dodaj encję `Pracownik` do diagramu sklep z zadania 3 i narysuj związek rekurencyjny (linia z encji do niej samej). Ustal krotności po obu stronach i uzasadnij je wynikami Z5 i Z6.

3 Biblioteka — autorzy i inne N:M

Otwórz diagram biblioteki z zadania 3.

1. Zastąp linię „N:M — zad. 4” między Książką i Autorem tabelą pośredniczącą Książka_Autor. Ustal jej klucz główny i dodaj atrybut kolejnosc (pierwszy, drugi autor...).
2. Na osobnym, małym diagramie narysuj związek Uczeń — Koło zainteresowań (uczeń zapisuje się na wiele kół, koło ma wielu uczniów, zapamiętujemy datę zapisu i ocenę końcową). Zdecyduj, gdzie trafiają data zapisu i ocena.
3. Sprawdź, czy w twoim diagramie biblioteki są inne ukryte związki N:M (np. Czytelnik — Egzemplarz przez Wypożyczenie). Zapisz, którą encję pośredniczącą już masz.

4 Nowa dziedzina — kino

Narysuj kompletny diagram ERD na podstawie opisu klienta.

Kino ma 3 sale. Każda sala ma numer i miejsca opisane rzędem i numerem miejsca w rzędzie. Film ma tytuł, czas trwania i kategorię wiekową; może należeć do kilku gatunków. Seans to wyświetlenie filmu w danej sali, w danym dniu i o danej godzinie, w wersji z dubbingiem albo z napisami. Bilet jest na konkretny seans i konkretne miejsce; ma rodzaj (normalny, ulgowy) i cenę. Na jedno miejsce na danym seansie można sprzedać tylko jeden bilet. Klient kupuje w jednej transakcji kilka biletów: online (podaje e-mail) albo w kasie (anonimowo).

1. Wypisz encje i atrybuty. Zaznacz, które związki są N:M (film — gatunek, seans — miejsce) i rozbij je.
2. Rozpoznaj encję słabą: miejsce nie istnieje bez sali i jest identyfikowane przez salę, rząd i numer. Narysuj jej klucz złożony.
3. Ustal, jak zapewnić regułę „jeden bilet na miejsce na seansie” (który zestaw kolumn musi być unikalny).
4. Zaznacz uczestnictwo związku Transakcja — Klient tak, żeby pasowało do zakupu anonimowego w kasie.
5. Zapisz diagram jako `zad04_kino_nazwisko.drawio` i wyeksportuj PNG.

W sprawozdaniu umieść

- ☐ tabela zapytań Z1–Z6 z wynikami
- ☐ schemat organizacyjny pracowników i diagram z dodanym związkiem rekurencyjnym
- ☐ poprawiony diagram biblioteki z tabelą Książka_Autor oraz diagram Uczeń — Koło
- ☐ diagram ERD kina (PNG) z opisem encji słabej i reguły unikalności biletu
- ☐ odpowiedzi na pytania kontrolne

Pytania kontrolne

1. Dlaczego w relacyjnej bazie danych nie da się zapisać związku N:M bez dodatkowej tabeli?
2. Co to jest encja słaba? Podaj przykład z bazy sklep i z kina.
3. Dlaczego kolumna id_przełożonego musi dopuszczać wartość NULL?
4. Jaki klucz główny ma tabela pośrednicząca i kiedy warto dodać jej osobny identyfikator (np. id_biletu)?

Kryteria oceny

Kryterium	Punkty
Zapytania Z1–Z6 i wyjaśnienie N:M na danych sklep	4
Związek rekurencyjny z krotnościami	3
Biblioteka: Książka_Autor i diagram Uczeń — Koło	3
Diagram kina: encje, atrybuty, rozbite N:M	4
Kino: encja słaba, reguła unikalności, uczestnictwo klienta	3
Odpowiedzi na pytania kontrolne	3
Razem	20

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

Model relacyjny — z diagramu ERD do schematu tabel

Czas: 2 × 45 min

praca indywidualna

Cel: Zapisanie schematu relacyjnego w notacji Tabela(atrybuty) z kluczami głównymi i obcymi, świadomy wybór między kluczem naturalnym a sztucznym oraz odczyt modelu istniejącej bazy w phpMyAdmin.

Po zajęciach potrafisz:

- zapisuje schemat relacyjny w notacji Tabela(atrybuty) z oznaczeniem PK, FK i UNIQUE
- przekształca związki 1:N, N:M i rekurencyjne na klucze obce i tabele
- porównuje klucze naturalne i sztuczne i uzasadnia wybór
- odczytuje model relacyjny bazy w widoku Projektant phpMyAdmin i z information_schema

Sprzęt i materiały

- komputer z XAMPP i bazą sklep
- diagramy biblioteki i kina z zadań 3–4

Oprogramowanie

- **phpMyAdmin (widok Projektant / Designer)** — baza sklep → Więcej → Projektant
docs.phpmyadmin.net/en/latest
- **draw.io (diagrams.net)** — do poprawek diagramów
drawio.com

Przygotowanie przed zajęciami

- Otwórz swoje diagramy biblioteki i kina z zadań 3–4 (pliki .drawio).

Bezpieczeństwo

- Nie używaj jako klucza danych osobowych wrażliwych lub zbędnych (PESEL, numer dowodu) — klucz trafia do wielu tabel i raportów, a takie dane wymagają szczególnej ochrony.

Przebieg zadania

1 Notacja schematu relacyjnego

W tym zadaniu stosujemy jednolity zapis. Klucz złożony oznaczamy PK przy każdej jego kolumnie.

```
Produkty(id_produktu PK, nazwa, id_kategorii FK→Kategorie NULL, cena, stan, data_dodania)
Pozycje_zamowienia(id_zamowienia PK FK→Zamowienia, id_produktu PK FK→Produkty, ilosc,
cena_jedn)
```

1. Zapisz w tej notacji schemat wszystkich sześciu tabel bazy sklep. Korzystaj z DESCRIBE albo zakładki Struktura w phpMyAdmin.
2. Oznacz UNIQUE przy kolumnach z indeksem unikalnym i NULL przy kolumnach, które mogą być puste.

2 Odczyt modelu bazy sklep

Zapytania wykonuj w zakładce SQL bazy sklep lub w konsoli i zapisuj w pliku `zad05_nazwisko.sql`.

1. Otwórz bazę sklep w phpMyAdmin → Więcej → Projektant. Rozmieść tabele czytelnie, włącz pokazywanie linii relacji i zrób zrzut ekranu. Porównaj z diagramem ERD z zadania 3.
2. Z1 — wyświetl kolumny wszystkich kluczy głównych bazy: z `information_schema.KEY_COLUMN_USAGE` wybierz `TABLE_NAME`, `COLUMN_NAME`, `ORDINAL_POSITION` dla bazy sklep i `CONSTRAINT_NAME = 'PRIMARY'`. Która tabela ma klucz złożony?
3. Z2 — wyświetl indeksy tabeli klienci: `SHOW INDEX FROM klienci;`. Odczytaj, który indeks jest unikalny (`Non_unique = 0`) i dlaczego email nie został kluczem głównym.
4. Z3 — wyświetl wszystkie ograniczenia bazy: z `information_schema.TABLE_CONSTRAINTS` wybierz `TABLE_NAME`, `CONSTRAINT_NAME`, `CONSTRAINT_TYPE` dla bazy sklep. Policz, ile jest ograniczeń każdego typu.
5. Sprawdź swój schemat z kroku 1 z wynikami Z1–Z3 i popraw błędy.

3 Klucz naturalny czy sztuczny?

Klucz naturalny to atrybut istniejący w świecie (ISBN, numer inwentarzowy, e-mail), sztuczny — liczba nadana przez bazę (`AUTO_INCREMENT`).

1. Dla kandydatów: ISBN książki, numer inwentarzowy egzemplarza, numer karty czytelnika, e-mail klienta, PESEL ucznia, para (sala, rząd, miejsce) w kinie — oceń w tabeli: czy jest unikalny, czy zawsze istnieje, czy może się zmienić, czy jest krótki.
2. Na tej podstawie zdecyduj dla każdej tabeli biblioteki, jakiego klucza użyjesz. Tam, gdzie wybierzesz klucz sztuczny, a kandydat naturalny istnieje, oznacz go `UNIQUE`.

4 Schematy biblioteki i kina

Zamieniasz diagramy z zadań 3–4 na schematy relacyjne.

1. Zapisz schemat relacyjny biblioteki: każda encja → tabela, każdy związek 1:N → klucz obcy po stronie „wiele”, związek N:M → tabela pośrednicząca z kluczem złożonym.
2. Zapisz schemat relacyjny kina. Zadbaj o encję słabą Miejsce i o regułę „jeden bilet na miejsce na seansie” (`UNIQUE` na odpowiednim zestawie kolumn).
3. Dla każdego klucza obcego określ, czy może być `NULL`, i porównaj z uczestnictwem na diagramie (kółko po stronie „jeden” = `FK NULL`).
4. Zachowaj schemat biblioteki — w zadaniu 9 utworzysz z niego bazę w SQL.

W sprawozdaniu umieść

- ☐ schemat relacyjny bazy sklep w notacji z zadania
- ☐ zrzut widoku Projektant bazy sklep
- ☐ tabela zapytań Z1–Z3 z wynikami
- ☐ tabela oceny kandydatów na klucz
- ☐ schematy relacyjne biblioteki i kina
- ☐ odpowiedzi na pytania kontrolne

Pytania kontrolne

1. Czym różni się klucz główny od klucza kandydującego? Podaj przykład z bazy sklep.
2. Dlaczego adres e-mail, mimo że jest unikalny, rzadko bywa kluczem głównym?
3. Jak w schemacie relacyjnym zapisuje się związek N:M, a jak związek rekurencyjny?
4. Co oznacza, że kolumna klucza obcego dopuszcza `NULL`? Podaj przykład z bazy sklep.

Kryteria oceny

Kryterium	Punkty
Poprawny schemat relacyjny bazy sklep	3
Zrzut Projektanta i zapytania Z1–Z3	3
Ocena kandydatów na klucz i uzasadniony wybór	3
Schemat relacyjny biblioteki (PK, FK, UNIQUE, NULL)	4
Schemat relacyjny kina z encją słabą i unikalnością biletu	4
Odpowiedzi na pytania kontrolne	3
Razem	20

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

Normalizacja 1NF–3NF — anomalie i zależności

Czas: 2 × 45 min

praca indywidualna

Cel: Rozpoznanie anomalii w tabeli nieznormalizowanej, wyznaczenie zależności funkcyjnych i rozłożenie danych krok po kroku do 1NF, 2NF i 3NF.

Po zajęciach potrafisz:

- wskazuje w danych anomalie wstawiania, modyfikacji i usuwania i demonstruje je w SQL
- sprowadza tabelę do 1NF (wartości atomowe, brak grup powtarzających się)
- zapisuje zależności funkcyjne i rozpoznaje zależności częściowe i przechodnie
- rozkłada tabelę do 2NF i 3NF i zapisuje wynikowy schemat relacyjny

Sprzęt i materiały

- komputer z XAMPP
- plik `r1_06_zamowienia_plaskie.sql` z folderu Pliki_do_zadan

Oprogramowanie

- **phpMyAdmin** lub **konsola mysql** — import pliku i zapytania [phpmyadmin.net](https://www.phpmyadmin.net)
- **Edytor tekstu** lub **arkusz kalkulacyjny** — rozpisywanie tabel 1NF–3NF

Przygotowanie przed zajęciami

- Powtórz z prezentacji 1.05 definicje 1NF, 2NF, 3NF i pojęcie zależności funkcyjnej.

Bezpieczeństwo

- Polecenia UPDATE i DELETE z kroku 3 wykonuj wyłącznie na tabeli `kopia` — tabela zamówienia_plaskie będzie potrzebna w zadaniu 7. Jeśli się pomylisz, zimportuj plik ponownie.

Przebieg zadania

1 Import i przegląd danych

Zimportuj plik `r1_06_zamowienia_plaskie.sql` (phpMyAdmin → Import albo SOURCE w konsoli). Powstanie baza normalizacja z dwiema tabelami. Zapytania zapisuj w pliku `zad06_nazwisko.sql`.

1. Z1 — wyświetl całą tabelę zamówienia_surowe. Tak sklep zapisywał zamówienia w arkuszu. Wypisz, które kolumny łamią zasadę atomowości (1NF) i dlaczego.
2. Z2 — wyświetl całą tabelę zamówienia_plaskie. Porównaj z Z1: ile wierszy ma każda tabela i co odpowiada jednemu wierszowi?
3. Zapisz, jaki jest klucz główny tabeli zamówienia_plaskie (DESCRIBE albo zakładka Struktura).

2 Szukanie powtórzeń i zależności

Słowo DISTINCT usuwa z wyniku powtarzające się wiersze.

1. Z3 — wyświetl wszystkie wiersze zamówień klientki o e-mailu anna.nowak@example.pl. Ile razy powtarza się jej imię, nazwisko i miasto? Co jest nie tak z miastem?
2. Z4 — wyświetl bez powtórzeń pary klient_email, klient_miasto. Ilu jest klientów, a ile wierszy zwróciło zapytanie? Wyjaśnij różnicę.
3. Z5 — wyświetl bez powtórzeń kod_produktu, nazwa_produktu, kategoria, stawka_vat. Z6 — wyświetl bez powtórzeń pary kategoria, stawka_vat.
4. Z7 — wyświetl bez powtórzeń pary kod_produktu, cena_jedn. Czy cena zależy tylko od produktu? Od czego jeszcze?
5. Z8 — policz wiersze, w których występuje produkt P01. Tyle razy baza przechowuje jego nazwę, kategorię i stawkę VAT.

3 Anomalie — demonstracja na kopii

Wykonaj polecenia z ramki po kolei (Z9–Z12). Po Z10 i Z11 sprawdź skutki własnymi zapytaniami SELECT.

```
-- Z9: kopia tabeli do eksperymentów
CREATE TABLE kopia AS SELECT * FROM zamowienia_plaskie;
-- Z10: Anna zmienia e-mail, ale poprawiono tylko zamówienie 1001
UPDATE kopia SET klient_email = 'anna.nowak2@example.pl' WHERE nr_zam = 1001;
-- Z11: anulowano i usunięto zamówienie 1005
DELETE FROM kopia WHERE nr_zam = 1005;
-- Z12: chcemy dodać nowy produkt, którego nikt jeszcze nie kupił
INSERT INTO kopia (kod_produktu, nazwa_produktu, kategoria, stawka_vat, cena_jedn)
VALUES ('P09', 'Linijka 30 cm', 'Przybory do pisanie', 23, 3.50);
```

1. Po Z10 wyświetl bez powtórzeń imię, nazwisko i e-mail osób o nazwisku Nowak w tabeli kopia. Nazwij anomalię.
2. Po Z11 sprawdź, czy w tabeli kopia jest jeszcze produkt P08 i klientka Ola Sowa. Nazwij anomalię.
3. Zapisz komunikat błędu Z12 (numer i treść). Wyjaśnij, dlaczego nie da się dodać produktu bez zamówienia. Nazwij anomalię.

4 Normalizacja krok po kroku

Pracuj w edytorze lub arkuszu. Wyniki zapisz w notacji Tabela(atrybuty) z zadania 5.

1. 1NF: opisz, jak z tabeli zamowienia_surowe powstaje tabela w 1NF (rozdzielenie kolumny klient, rozbicie listy produktów na wiersze). Porównaj swój wynik z tabelą zamowienia_plaskie.
2. Zapisz wszystkie zależności funkcyjne tabeli zamowienia_plaskie w postaci $X \rightarrow Y$ (np. nr_zam $\rightarrow$ data_zam). Oznacz zależności częściowe (od części klucza) i przechodnie.
3. 2NF: usuń zależności częściowe — rozdziel tabelę tak, aby każdy atrybut niekluczowy zależał od całego klucza. Zapisz powstałe tabele.
4. 3NF: usuń zależności przechodnie (np. przez e-mail klienta, przez kategorię). Zapisz końcowy schemat z kluczami PK i FK. Zaproponuj, gdzie warto dodać klucz sztuczny.
5. Sprawdź, czy w schemacie 3NF trzy anomalie z kroku 3 są już niemożliwe. Zachowaj schemat do zadania 7.

W sprawozdaniu umieść

- ☐ tabela zapytań Z1–Z12 z wynikami
- ☐ opis trzech anomalii z dowodem (wynik zapytania lub komunikat błędu)
- ☐ lista zależności funkcyjnych z oznaczeniem częściowych i przechodnich
- ☐ tabele w 1NF, 2NF i 3NF oraz końcowy schemat relacyjny

☐ odpowiedzi na pytania kontrolne

Pytania kontrolne

1. Dlaczego kolumna produkty w tabeli zamówienia_surowe łamie 1NF? Jakie zapytania byłyby przez nią trudne?
2. Czym różni się zależność częściowa od przechodniej? Podaj po jednym przykładzie z tego zadania.
3. Dlaczego cena jednostkowa pozostaje w tabeli pozycji, a nie trafia do tabeli produktów?
4. Dlaczego niespójność Kraków / Krakow nie mogłaby powstać w bazie w 3NF?

Kryteria oceny

Kryterium	Punkty
Import i zapytania Z1–Z8 z poprawnymi wnioskami	4
Demonstracja i nazwanie trzech anomalii (Z9–Z12)	4
Zależności funkcyjne z podziałem na częściowe i przechodnie	3
Poprawne tabele 1NF, 2NF i 3NF	6
Odpowiedzi na pytania kontrolne	3
Razem	20

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

Normalizacja — implementacja i przeniesienie danych

Czas: 2 × 45 min

praca indywidualna

Cel: Utworzenie w MySQL/MariaDB tabel w 3NF zaprojektowanych w zadaniu 6, przeniesienie do nich danych z tabeli płaskiej poleceniem INSERT ... SELECT i sprawdzenie, że żadne dane nie zginęły.

Po zajęciach potrafisz:

- tworzy tabele z kluczami głównymi i obcymi we właściwej kolejności
- przenosi dane poleceniem INSERT ... SELECT DISTINCT
- wykrywa niespójne dane dzięki ograniczeniu UNIQUE i poprawia je przed migracją
- weryfikuje kompletność migracji liczbą wierszy i sumą kontrolną
- sprawdza, że anomalie z zadania 6 nie występują w bazie znormalizowanej

Sprzęt i materiały

- komputer z XAMPP
- plik `r1_06_zamowienia_plaskie.sql` z folderu Pliki_do_zadan i schemat 3NF z zadania 6

Oprogramowanie

- **phpMyAdmin** lub **konsola mysql**
phpmyadmin.net
- **Visual Studio Code (opcjonalnie)** —
pisanie skryptu `zad07_nazwisko.sql`
code.visualstudio.com

Przygotowanie przed zajęciami

- Zaimportuj ponownie plik `r1_06_zamowienia_plaskie.sql`, aby zacząć od czystych danych (usunie to tabelę kopia z zadania 6).

Bezpieczeństwo

- Polecenie UPDATE w kroku 3 zmienia dane źródłowe — przed nim wykonaj SELECT z tym samym warunkiem i sprawdź, ile wierszy zostanie zmienionych.
- Gdy coś pójdzie nie tak, nie poprawiaj na ślepo: zaimportuj plik ponownie i uruchom swój skrypt od początku.

Przebieg zadania

1 Schemat docelowy i tworzenie tabel

Wszyscy implementują ten sam schemat (porównaj go ze swoim z zadania 6). Pisz polecenia w pliku `zad07_nazwisko.sql` i wykonuj je w bazie normalizacja.

```
Kategorie(id_kategorii PK, nazwa UNIQUE, stawka_vat)
Klienci(id_klienta PK, imie, nazwisko, email UNIQUE, miasto)
Produkty(kod_produkту PK, nazwa, id_kategorii FK→Kategorie)
Zamowienia(nr_zam PK, data_zam, id_klienta FK→Klienci)
Pozycje(nr_zam PK FK→Zamowienia, kod_produkту PK FK→Produkty, ilosc, cena_jedn)

-- Z1: wzór
CREATE TABLE kategorie (
  id_kategorii INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
  nazwa          VARCHAR(50) NOT NULL UNIQUE,
  stawka_vat     TINYINT UNSIGNED NOT NULL
) ENGINE=InnoDB;
```

1. Z1 — wykonaj polecenie z ramki.
2. Z2–Z5 — utwórz tabele klienci, produkty, zamowienia i pozycje. Typy kolumn przepisz z tabeli `zamowienia_plaskie` (`DESCRIBE`); `id_klienta` zrób jak `id_kategorii`. Klucz obcy zapisz w `CREATE TABLE` np. tak: `FOREIGN KEY (id_kategorii) REFERENCES kategorie(id_kategorii)`.
3. Zwróć uwagę na kolejność: tabela, na którą wskazuje klucz obcy, musi istnieć wcześniej. Typ kolumny klucza obcego musi być taki sam jak typ kolumny, na którą wskazuje (np. `INT UNSIGNED`).
4. Sprawdź wynik w phpMyAdmin → normalizacja → Więcej → Projektant: czy widać 4 linie relacji?

2 Przeniesienie kategorii i klientów

`INSERT ... SELECT` wstawia do tabeli wynik zapytania. Kolumna `AUTO_INCREMENT` nada identyfikatory sama.

```
-- Z6: wzór
INSERT INTO kategorie (nazwa, stawka_vat)
SELECT DISTINCT kategoria, stawka_vat FROM zamowienia_plaskie;
```

1. Z6 — wykonaj polecenie z ramki i wyświetl tabelę `kategorie`. Ile wierszy wstawiono?
2. Z7 — napisz analogiczne polecenie przenoszące klientów (imię, nazwisko, e-mail, miasto bez powtórzeń). Zapisz komunikat, który otrzymasz, i wyjaśnij go — pomoże ci Z4 z zadania 6.

3 Poprawienie danych i dokończenie migracji

Klucz obcy w tabeli płaskiej nie istnieje — identyfikator kategorii lub klienta trzeba odnaleźć po nazwie lub e-mailu. Służą do tego złączenie JOIN (dokładnie poznasz je w zadaniu 21).

```
-- Z8: poprawka danych źródłowych (najpierw SELECT z tym samym WHERE!)
UPDATE zamowienia_plaskie SET klient_miasto = 'Kraków' WHERE klient_miasto = 'Krakow';

-- Z10: wzór z JOIN
INSERT INTO produkty (kod_produkту, nazwa, id_kategorii)
SELECT DISTINCT p.kod_produkту, p.nazwa_produkту, k.id_kategorii
FROM zamowienia_plaskie AS p
JOIN kategorie AS k ON k.nazwa = p.kategoria;
```

1. Z8 — wyświetl wiersze z miastem 'Krakow', a potem wykonaj UPDATE z ramki. Zapisz liczbę zmienionych wierszy.
2. Z9 — wykonaj ponownie polecenie z Z7. Wyświetl tabelę klienci. Od jakiego numeru zaczynają się identyfikatory i dlaczego nie od 1?
3. Z10 — wykonaj polecenie z ramki. Z11 — napisz na tej samej zasadzie przeniesienie zamówień (nr_zam, data_zam, id_klienta) — klienta odnajdź po e-mailu.
4. Z12 — przenieś pozycje zamówień. Tu DISTINCT ani JOIN nie są potrzebne — dlaczego?

4 Kontrola — czy nic nie zginęło

```
-- Z14: odtworzenie tabeli płaskiej z tabel 3NF
SELECT COUNT(*)
FROM pozycje AS po
JOIN zamowienia AS z ON z.nr_zam = po.nr_zam
JOIN klienci AS k ON k.id_klienta = z.id_klienta
JOIN produkty AS p ON p.kod_produkту = po.kod_produkту
JOIN kategorie AS kat ON kat.id_kategorii = p.id_kategorii;
```

1. Z13 — policz wiersze w każdej z pięciu nowych tabel (`SELECT COUNT(*) FROM ...`) i porównaj z liczbą różnych wartości w tabeli płaskiej (np. `SELECT COUNT(DISTINCT kod_produkту) FROM zamowienia_plaskie;`). Wyniki wpisz do tabeli kontrolnej.
2. Z14 — wykonaj zapytanie z ramki. Wynik musi równać się liczbie wierszy tabeli płaskiej.
3. Z15 — oblicz sumę kontrolną `SUM(ilosc * cena_jedn)` w tabeli zamowienia_plaskie i w tabeli pozycje. Wartości muszą być identyczne.

5 Anomalie — czy zniknęły?

Powtórz eksperymenty z zadania 6 na tabelach 3NF.

1. Z16 — dodaj produkt P09 „Linijka 30 cm” do kategorii Przybory do pisania (sprawdź jej id w tabeli kategorie). Czy potrzebne jest zamówienie?
2. Z17 — zmień e-mail Anny Nowak na anna.nowak2@example.pl. Ile wierszy zmieniono?
3. Z18 — spróbuj usunąć zamówienie 1006 z tabeli zamowienia. Zapisz błąd. Następnie usuń najpierw jego pozycje, potem zamówienie, i sprawdź, czy klient Marek Wrona i produkt P07 nadal są w bazie.

W sprawozdaniu umieść

- ☐ skrypt `zad07_nazwisko.sql` z poleceniami Z1–Z18
- ☐ tabela zapytań z wynikami i komunikatami błędów
- ☐ wyjaśnienie błędu z Z7 i luki w numeracji klientów
- ☐ tabela kontrolna migracji (liczby wierszy, Z14, sumy kontrolne)
- ☐ wyniki eksperymentów z anomaliami (Z16–Z18)

☐ odpowiedzi na pytania kontrolne

Pytania kontrolne

1. Dlaczego tabele trzeba tworzyć i wypełniać w określonej kolejności? Jaka jest poprawna kolejność w tym zadaniu?
2. Jak ograniczenie UNIQUE na kolumnie email pomogło wykryć błąd w danych?
3. Dlaczego po nieudanym INSERT identyfikatory AUTO_INCREMENT nie zaczynają się od 1 i czy to jest problem?
4. Dlaczego porównanie samych liczb wierszy nie wystarcza do sprawdzenia migracji i po co liczy się sumę kontrolną?

Kryteria oceny

Kryterium	Punkty
Poprawne tabele Z1–Z5 z kluczami głównymi i obcymi	5
Migracja danych Z6–Z12 z wyjaśnieniem błędu 1062	5
Kontrola kompletności Z13–Z15	3
Eksperymenty z anomaliami Z16–Z18	3
Odpowiedzi na pytania kontrolne	4
Razem	20

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

Typy danych — dobór i testy

Czas: 2 × 45 min

praca indywidualna

Cel: Sprawdzenie doświadczalnie, jak serwer zapisuje liczby, teksty i daty różnych typów, co dzieje się przy przekroczeniu zakresu w trybie ścisłym i nieścisłym, oraz dobranie typów kolumn dla bazy biblioteki.

Po zajęciach potrafisz:

- dobiera typ liczbowy (TINYINT, INT, UNSIGNED, DECIMAL) do zakresu i rodzaju danych
- wyjaśnia, dlaczego kwot nie przechowuje się w FLOAT ani DOUBLE
- rozróżnia CHAR, VARCHAR i TEXT oraz LENGTH i CHAR_LENGTH dla polskich znaków
- dobiera typ DATE, DATETIME lub TIMESTAMP i zna ograniczenia ENUM i BOOLEAN
- sprawdza i zmienia tryb sql_mode oraz opisuje skutki trybu nieściśłego

Sprzęt i materiały

- komputer z XAMPP
- schemat relacyjny biblioteki z zadania 5

Oprogramowanie

- **Konsola mysql (klient MariaDB)** — w tym zadaniu pracuj w konsoli — ustawienie SET SESSION w phpMyAdmin obowiązuje tylko w jednym wykonaniu zapytań
mariadb.com/kb/en/mariadb-command-line-client
- **Dokumentacja typów danych MariaDB** — zakresy typów; dla MySQL:
<https://dev.mysql.com/doc/refman/8.4/en/data-types.html>
mariadb.com/kb/en/data-types

Przygotowanie przed zajęciami

- Przy każdym teście najpierw wpisz do sprawozdania przewidywany wynik, dopiero potem wykonaj polecenie.

Bezpieczeństwo

- Tryb sql_mode zmieniaj tylko dla swojej sesji (SET SESSION). Nie zmieniaj ustawień globalnych serwera ani pliku my.ini.

Przebieg zadania

1 Baza testowa i tryb SQL

Zaloguj się do konsoli z opcją `--default-character-set=utf8mb4`. Wszystkie polecenia zapisuj w pliku `zad08_nazwisko.sql`.

```
CREATE DATABASE typy CHARACTER SET utf8mb4 COLLATE utf8mb4_polish_ci;
USE typy;
SET SESSION sql_mode =
'STRICT_TRANS_TABLES,ERROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION';

CREATE TABLE liczby (
  id      INT AUTO_INCREMENT PRIMARY KEY,
  maly    TINYINT,
  maly_u  TINYINT UNSIGNED,
  kwota   DECIMAL(6,2),
  f       FLOAT,
  d       DOUBLE
);
```

1. Z1 — przed zmianą trybu odczytaj domyślny tryb serwera: `SELECT @@GLOBAL.sql_mode;`. Zapisz, czy zawiera `STRICT_TRANS_TABLES`.
2. Wykonaj polecenia z ramki. Od teraz sesja działa w trybie ścisłym niezależnie od ustawień serwera.

2 Liczby całkowite i DECIMAL

Przy każdym poleceniu zapisz: przewidywanie, komunikat (numer błędu lub ostrzeżenia) i to, co zostało zapisane.

1. Z2 — wstaw wiersz z wartościami `maly = 127`, `maly_u = 255`, `kwota = 9999.99`, `f = 0.1`, `d = 0.1`. Wyświetl tabelę.
2. Z3 — wstaw `maly = 128`. Z4 — wstaw `maly_u = -1`. Z5 — wstaw `kwota = 12345.67`. Odczytaj z komunikatów i dokumentacji zakresy `TINYINT`, `TINYINT UNSIGNED` i `DECIMAL(6,2)`.
3. Z6 — wstaw `kwota = 12.345`, a potem wykonaj `SHOW WARNINGS;`. Co zapisano w tabeli?

3 Test 0,1 + 0,2

Liczby zmiennoprzecinkowe (`FLOAT`, `DOUBLE`) są zapisywane w systemie dwójkowym w przybliżeniu.

1. Z7 — wstaw drugi wiersz z `f = 0.2` i `d = 0.2`. Oblicz `SUM(f)`, `SUM(d)` oraz porównania `SUM(f) = 0.3` i `SUM(d) = 0.3` dla wierszy, w których `f` nie jest `NULL`.
2. Z8 — wykonaj `SELECT 0.1 + 0.2, 0.1 + 0.2 = 0.3, 0.1e0 + 0.2e0;`. Liczby zapisane z kropką są typu `DECIMAL`, a z literą `e` — `DOUBLE`. Wyjaśnij różnicę wyników.
3. Zapisz wniosek: jakiego typu użyjesz do cen i sum pieniędzy, a jakiego do pomiarów (np. temperatury)?

4 Teksty

```
CREATE TABLE teksty (c CHAR(5), v VARCHAR(5), t TEXT);
```

1. Z9 — utwórz tabelę z ramki i wstaw do wszystkich trzech kolumn tekst `'ab '` (dwie spacje na końcu). Wyświetl `CONCAT('[', c, ' ]')`, `CONCAT('[', v, ' ]')`, `LENGTH(c)` i `LENGTH(v)`. Co stało się ze spacjami w `CHAR`?
2. Z10 — wstaw do kolumny `v` tekst `'Łódź!'` i wyświetl dla niego `LENGTH(v)` i `CHAR_LENGTH(v)`. Wyjaśnij różnicę (bajty i znaki w `utf8mb4`).
3. Z11 — wstaw do kolumny `v` tekst `'Szczecin'`. Zapisz komunikat.

5 Daty, ENUM i BOOLEAN

```
CREATE TABLE daty (d DATE, dt DATETIME, ts TIMESTAMP NULL,  
e ENUM('nowe', 'opłacone', 'wysłane'), b BOOLEAN);
```

- Z12 — utwórz tabelę z ramki i wstaw do kolumny d datę '2026-02-30', a potem '2028-02-29'. Który rok jest przestępny?
- Z13 — wstaw do ts wartość '2038-01-01 00:00:00', potem '2040-01-01 00:00:00'; tę drugą wstaw też do dt. Sprawdź w dokumentacji zakres typu TIMESTAMP i wyjaśnij „problem roku 2038”.
- Z14 — wstaw do e wartość 'wysłane' (z polską literą), potem 'wyslane'. Z15 — wstaw do b wartości TRUE, FALSE i 5 jednym poleceniem INSERT, wyświetl tabelę i wykonaj `SHOW CREATE TABLE daty;`. Jakim typem jest naprawdę BOOLEAN?

6 Tryb nieściśły i dobór typów

- Z16 — wyłącz tryb ściśły: `SET SESSION sql_mode = '';`. Powtórz Z3 (z wartością mały = 300), Z11 i Z12 (data '2026-02-30'). Po każdym poleceniu wykonaj `SHOW WARNINGS`, a na końcu wyświetl zapisane wartości. Dlaczego tryb nieściśły jest groźny?
- Przywróć tryb serwera: `SET SESSION sql_mode = DEFAULT;`.
- Dobierz typy SQL dla wszystkich kolumn schematu biblioteki z zadania 5 (tabela w sprawozdaniu): długość tekstów, UNSIGNED, DECIMAL, typ daty, ENUM dla stanu egzemplarza, NULL / NOT NULL. Zachowaj wynik do zadania 9.

W sprawozdaniu umieść

- ☐ tabela testów Z1–Z16: przewidywanie, komunikat, zapisana wartość
- ☐ wnioski z testu 0,1 + 0,2
- ☐ porównanie trybu ściśłego i nieściśłego
- ☐ dobór typów dla tabel biblioteki z uzasadnieniem
- ☐ odpowiedzi na pytania kontrolne
- ☐ plik `zad08_nazwisko.sql`

Pytania kontrolne

- Dlaczego ceny przechowuje się w DECIMAL, a nie w FLOAT? Co oznacza zapis DECIMAL(10,2)?
- Czym różni się VARCHAR(50) od CHAR(50)? Kiedy CHAR ma sens (podaj przykład)?
- Dlaczego LENGTH('Łódź!') jest większe niż CHAR_LENGTH('Łódź!')? Które z nich odpowiada liczbie w VARCHAR(n)?
- Kiedy użyjesz DATE, kiedy DATETIME, a kiedy TIMESTAMP?
- Co może się stać z danymi w aplikacji, jeśli serwer działa w trybie nieściśłym?

Kryteria oceny

Kryterium	Punkty
Testy liczbowe i DECIMAL (Z2–Z8) z przewidywaniami i wnioskami	5
Testy tekstów (Z9–Z11)	3
Testy dat, ENUM i BOOLEAN (Z12–Z15)	3
Tryb nieściśły (Z16) i wnioski	2
Dobór typów dla biblioteki z uzasadnieniem	4
Odpowiedzi na pytania kontrolne	3
Razem	20

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

DDL — tworzenie bazy i tabel biblioteki

Czas: 2 × 45 min

praca indywidualna

Cel: Napisanie skryptu .sql, który tworzy bazę biblioteka z poprawnym kodowaniem i wszystkie jej tabele z kluczami, wartościami domyślnymi i ograniczeniami NOT NULL, oraz sprawdzenie go poleceniami SHOW.

Po zajęciach potrafisz:

- tworzy bazę z kodowaniem utf8mb4 i porównywaniem utf8mb4_polish_ci
- pisze polecenia CREATE TABLE z PRIMARY KEY, AUTO_INCREMENT, NOT NULL, DEFAULT, UNIQUE i FOREIGN KEY
- tworzy klucz złożony tabeli pośredniczącej
- zapisuje strukturę jako skrypt, który można uruchomić wielokrotnie
- sprawdza strukturę poleceniami SHOW CREATE DATABASE / TABLE

Sprzęt i materiały

- komputer z XAMPP
- schemat relacyjny biblioteki (zad. 5) i dobór typów (zad. 8)

Oprogramowanie

- **Visual Studio Code** — lub Notatnik++ —
pisanie skryptu `biblioteka.sql`
(kodowanie pliku UTF-8)
code.visualstudio.com
- **Konsola mysql lub phpMyAdmin** —
dokumentacja CREATE TABLE w MariaDB
mariadb.com/kb/en/create-table

Przygotowanie przed zajęciami

- Miej pod ręką swój schemat biblioteki i tabelę typów z zadania 8. Jeśli ich nie masz, poproś nauczyciela o schemat wzorcowy.

Bezpieczeństwo

- Polecenie DROP DATABASE IF EXISTS na początku skryptu usuwa bazę bez pytania — sprawdź, czy nazwa bazy w skrypcie to na pewno biblioteka, a nie np. sklep.
- Zapisuj skrypt w pliku na bieżąco — to on jest twoją kopią zapasową struktury.

Przebieg zadania

1 Baza danych

Piszesz skrypt w edytorze i uruchamiasz go w konsoli poleceniem SOURCE (albo wklejasz do zakładki SQL phpMyAdmin).

```
-- początek pliku biblioteka.sql
SET NAMES utf8mb4;
DROP DATABASE IF EXISTS biblioteka;
-- Z1: ...
USE biblioteka;
```

1. Z1 — dopisz polecenie tworzące bazę biblioteka z kodowaniem utf8mb4 i porównywaniem utf8mb4_polish_ci.
2. Z2 — po uruchomieniu wykonaj `SHOW CREATE DATABASE biblioteka;` i sprawdź kodowanie.
3. Z3 — porównaj działanie porównywań: `SELECT 'Kraków' = 'Krakow' COLLATE utf8mb4_polish_ci, 'Kraków' = 'Krakow' COLLATE utf8mb4_general_ci;`. Dlaczego dla polskiej bazy wybieramy polish_ci?

2 Tabele słownikowe i książki

Tabele twórz w kolejności zależności: najpierw te, na które wskazują klucze obce.

```
CREATE TABLE dzialy (
  id_dzialu INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
  nazwa      VARCHAR(50) NOT NULL UNIQUE
) ENGINE=InnoDB;
```

1. Z4 — dopisz do skryptu tabelę dzialy z ramki oraz analogiczne tabele wydawnictwa i autorzy (imię i nazwisko osobno).
2. Z5 — utwórz tabelę ksiazki: identyfikator, ISBN (13 znaków, unikalny, może być pusty), tytuł, rok wydania, wydawnictwo (może być puste) i dział (wymagany). Dodaj dwa klucze obce.
3. Z6 — utwórz tabelę ksiazki_autorzy z kluczem złożonym `PRIMARY KEY (id_ksiazki, id_autora)`, kolumną kolejnosc z wartością domyślną 1 i dwoma kluczami obcymi.

3 Egzemplarze, czytelnicy, wypożyczenia

1. Z7 — utwórz tabelę egzemplarze: numer inwentarzowy jako klucz naturalny (bez AUTO_INCREMENT), książka, stan jako ENUM('dobry','zniszczony','wycofany') z wartością domyślną 'dobry'.
2. Z8 — utwórz tabelę czytelnicy: numer karty (AUTO_INCREMENT), imię, nazwisko, typ ENUM('uczen','nauczyciel') z wartością domyślną, klasa (może być pusta), e-mail (unikalny, może być pusty).
3. Z9 — utwórz tabelę wypozyczenia: identyfikator, egzemplarz, czytelnik, data wypożyczenia, termin zwrotu, data zwrotu (pusta = książka u czytelnika), liczba przedłużeń z wartością domyślną 0.
4. Dla chętnych: dodaj tabelę rezerwacje (tytuł, czytelnik, data rezerwacji, status).

4 Sprawdzenie struktury i skryptu

```
-- Z12: testy (wykonaj po kolei, zapisz komunikaty)
INSERT INTO dzialy (nazwa) VALUES ('Lektury');
INSERT INTO ksiazki (tytul, id_dzialu) VALUES ('Lalka', 1), ('Pan Tadeusz', 1);
INSERT INTO egzemplarze (nr_inwentarzowy, id_ksiazki) VALUES (4501, 1);
SELECT * FROM egzemplarze;
INSERT INTO egzemplarze (nr_inwentarzowy, id_ksiazki) VALUES (4502, 99);
INSERT INTO egzemplarze (nr_inwentarzowy, id_ksiazki, stan) VALUES (4503, 1, 'zgubiony');
```

1. Z10 — `SHOW TABLES;` — ile tabel ma baza? Z11 — `SHOW CREATE TABLE ksiazki;` — jakie nazwy nadał serwer kluczom obcym, których nie nazwałeś?
2. Z12 — wykonaj testy z ramki. Wyjaśnij: jaką wartość dostała kolumna stan, dlaczego dwie książki bez ISBN nie naruszyły UNIQUE, co oznaczają dwa ostatnie błędy.
3. Z13 — uruchom cały skrypt jeszcze raz (`SOURCE C:/.../biblioteka.sql`). Musi wykonać się bez żadnego błędu, a baza po nim jest pusta i gotowa do pracy.
4. Zrób zrzut widoku Projektant bazy biblioteka i porównaj z diagramem ERD z zadań 3–4. Zachowaj skrypt — w zadaniu 10 będziesz zmieniać strukturę.

W sprawozdaniu umieść

- ☐ pełny skrypt `biblioteka.sql` (wydruk lub plik)
- ☐ tabela poleceń Z1–Z13 z wynikami i komunikatami
- ☐ zrzut widoku Projektant bazy biblioteka
- ☐ wyjaśnienie wyników testów z Z12
- ☐ odpowiedzi na pytania kontrolne

Pytania kontrolne

1. Czym różni się CHARACTER SET od COLLATE? Co by się stało, gdyby baza miała kodowanie latin1?
2. Dlaczego kolejność poleceń CREATE TABLE w skrypcie ma znaczenie?
3. Po co w skrypcie DROP DATABASE IF EXISTS i dlaczego jest to zarazem niebezpieczne?
4. Dlaczego kolumna z ograniczeniem UNIQUE może zawierać wiele wartości NULL?

Kryteria oceny

Kryterium	Punkty
Baza z poprawnym kodowaniem i porównywaniem (Z1–Z3)	2
Tabele z poprawnymi typami, PK, NOT NULL i DEFAULT (Z4–Z9)	6
Klucze obce i klucz złożony tabeli pośredniczącej	4
Sprawdzenie struktury i testy Z10–Z12 z wyjaśnieniem	3
Skrypt uruchamiający się wielokrotnie bez błędów (Z13)	2
Odpowiedzi na pytania kontrolne	3
Razem	20

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

DDL — zmiany struktury: ALTER, RENAME, TRUNCATE i DROP

Czas: 2 × 45 min

praca indywidualna

Cel: Bezpieczne zmienianie struktury istniejących tabel poleceniami ALTER TABLE i RENAME TABLE oraz doświadczalne porównanie DELETE, TRUNCATE i DROP na kopiach tabel.

Po zajęciach potrafisz:

- tworzy kopię tabeli poleceniami CREATE TABLE ... LIKE oraz CREATE TABLE ... AS SELECT i porównuje ich wynik
- dodaje, modyfikuje, zmienia nazwę i usuwa kolumny (ADD, MODIFY, CHANGE, DROP COLUMN)
- dodaje i usuwa klucze główne, unikalne i obce
- zmienia nazwę tabeli i sprawdza, co dzieje się z kluczami obcymi
- wyjaśnia różnice między DELETE, TRUNCATE i DROP

Sprzęt i materiały

- komputer z XAMPP i bazą sklep
- plik `r1_10_biblioteka_wzor.sql` z folderu Pliki_do_zadan

Oprogramowanie

- **Konsola mysql lub phpMyAdmin** — dokumentacja ALTER TABLE w MariaDB mariadb.com/kb/en/alter-table

Przygotowanie przed zajęciami

- Zaimportuj plik `r1_10_biblioteka_wzor.sql` — powstanie baza biblioteka_wzor z przykładowymi danymi.

Bezpieczeństwo

- W części A zmieniasz wyłącznie tabele produkty_kopia i produkty_as — nigdy oryginalnej tabeli produkty. Przed każdym DROP i TRUNCATE przeczytaj dwa razy nazwę tabeli.
- ALTER TABLE, TRUNCATE i DROP nie dają się cofnąć poleceniem ROLLBACK. Na prawdziwym serwerze przed zmianą struktury wykonuje się kopię zapasową.
- Po zajęciach zaimportuj ponownie `sklep.sql`, jeśli zmieniłeś coś w bazie sklep przez pomyłkę.

Przebieg zadania

1 Część A — dwie kopie tabeli

Pracuj w bazie sklep. Polecenia zapisuj w pliku `zad10_nazwisko.sql`.

```
CREATE TABLE produkty_kopia LIKE produkty;
INSERT INTO produkty_kopia SELECT * FROM produkty;

CREATE TABLE produkty_as AS SELECT * FROM produkty;
```

1. Z1 — wykonaj polecenia z ramki. Ile wierszy ma każda kopia?
2. Z2 — wyświetl `SHOW CREATE TABLE` dla produkty, produkty_kopia i produkty_as. W tabeli porównawczej zaznacz, która kopia zachowała: klucz główny, AUTO_INCREMENT, indeks na id_kategorii, klucz obcy do kategorii.

2 Część A — zmiany kolumn i kluczy

Po każdym poleceniu sprawdź wynik poleceniem DESCRIBE.

1. Z3 — dodaj do produkty_kopia kolumnę opis typu TEXT (może być pusta) zaraz po kolumnie nazwa (`ADD COLUMN ... AFTER nazwa`).
2. Z4 — wydłuż kolumnę nazwa do 150 znaków, zachowując NOT NULL (MODIFY). Z5 — zmień nazwę kolumny stan na ilosc_magazyn, zachowując typ i wartość domyślną (CHANGE). Uwaga: krótszy zapis `RENAME COLUMN` działa w MySQL 8 i MariaDB od 10.5, ale nie w MariaDB 10.4 z XAMPP.
3. Z6 — usuń kolumnę opis.
4. Z7 — w tabeli produkty_as dodaj klucz główny na id_produktu, potem ustaw tej kolumnie AUTO_INCREMENT (MODIFY z pełną definicją) i dodaj indeks unikalny na kolumnie nazwa. Sprawdź `SHOW INDEX FROM produkty_as` ;.
5. Z8 — zmień nazwę tabeli produkty_as na produkty_archiwum (RENAME TABLE).

3 Część A — DELETE, TRUNCATE, DROP

Eksperyment na tabeli produkty_kopia.

```
INSERT INTO produkty_kopia (nazwa, cena, data_dodania)
VALUES ('Test', 10.00, '2026-10-20');
SELECT id_produktu, nazwa FROM produkty_kopia;
```

1. Z9 — usuń wszystkie wiersze poleceniem `DELETE FROM produkty_kopia`; (zapisz liczbę usuniętych wierszy), potem wstaw wiersz testowy z ramki. Jaki identyfikator dostał?
2. Z10 — wykonaj `TRUNCATE TABLE produkty_kopia`; i znów wstaw wiersz testowy. Jaki identyfikator dostał teraz?
3. Z11 — spróbuj wykonać `TRUNCATE TABLE kategorie`; na oryginalnej tabeli. Zapisz numer błędu i wyjaśnij, dlaczego serwer odmówił.
4. Z12 — usuń tabele produkty_kopia i produkty_archiwum (DROP TABLE) i sprawdź SHOW TABLES, że baza sklep ma znów 6 tabel.

4 Część B — rozbudowa bazy biblioteki

Pracuj w bazie biblioteka_wzor. Klient zgłosił nowe wymagania.

1. Z13 — dodaj do tabeli czytelnicy kolumnę rok_ukonczenia (SMALLINT UNSIGNED, może być pusta) po kolumnie klasa. Dodaj do tabeli ksiazki kolumnę liczba_stron. Sprawdź, jakie wartości mają nowe kolumny w istniejących wierszach.
2. Z14 — rozszerz listę stanów egzemplarza o wartość 'zagubiony' (MODIFY z całą nową listą ENUM). Potem spróbuj zawęzić listę do ('dobry','wycofany'). Zapisz komunikat i wyjaśnij, dlaczego zmiana się nie udała.
3. Z15 — spróbuj skrócić kolumnę imie w tabeli czytelnicy do VARCHAR(3). Zapisz komunikat.
4. Z16 — wyświetl `SHOW CREATE TABLE wypozyczenia`; , usuń klucz obcy do tabeli czytelnicy (DROP FOREIGN KEY z nazwą odczytaną z wyniku) i dodaj go ponownie pod własną nazwą fk_wyp_czytelnik (ADD CONSTRAINT ... FOREIGN KEY ...).
5. Z17 — zmień nazwę tabeli wydawnictwa na wydawcy. Wyświetl `SHOW CREATE TABLE ksiazki`; — co stało się z kluczem obcym?
6. Dopisz zmiany z Z13–Z17 jako osobny plik migracji `biblioteka_zmiany_v2.sql` — tak w praktyce zmienia się strukturę działającej bazy.

W sprawozdaniu umieść

- ☐ tabela porównawcza kopii (Z2)
- ☐ tabela poleceń Z1–Z17 z wynikami i komunikatami

- ☐ wnioski z eksperymentu DELETE / TRUNCATE / DROP
- ☐ plik migracji `biblioteka_zmiany_v2.sql`
- ☐ odpowiedzi na pytania kontrolne

Pytania kontrolne

1. Czym różni się CREATE TABLE ... LIKE od CREATE TABLE ... AS SELECT? Której użyjesz do kopii zapasowej danych przed eksperymentem, a której do pustej tabeli o tej samej strukturze?
2. Czym różnią się MODIFY i CHANGE w ALTER TABLE?
3. Podaj trzy różnice między DELETE FROM tabela a TRUNCATE TABLE tabela.
4. Dlaczego ALTER TABLE może się nie udać na tabeli z danymi, choć na pustej tabeli działa (Z14, Z15)?

Kryteria oceny

Kryterium	Punkty
Kopie tabel i tabela porównawcza (Z1–Z2)	3
Zmiany kolumn i kluczy w części A (Z3–Z8)	5
Eksperyment DELETE / TRUNCATE / DROP z wnioskami (Z9–Z12)	4
Zmiany bazy biblioteki i plik migracji (Z13–Z17)	5
Odpowiedzi na pytania kontrolne	3
Razem	20

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

Integralność danych — klucze i ograniczenia

Czas: 2 × 45 min

praca indywidualna

Cel: Sprawdzenie w praktyce, jak klucze główne, obce, UNIQUE, CHECK, NOT NULL i DEFAULT chronią poprawność danych, oraz zaprojektowanie własnych ograniczeń z akcjami ON DELETE CASCADE, SET NULL i RESTRICT.

Po zajęciach potrafisz:

- przewiduje i odczytuje skutki naruszenia klucza obcego, klucza głównego i UNIQUE
- rozpoznaje błędy 1451, 1452, 1062, 1364 i 4025 (w MySQL 3819) i wyjaśnia ich przyczynę
- dobiera akcję ON DELETE (CASCADE, SET NULL, RESTRICT) do reguły biznesowej
- tworzy tabele z nazwanymi ograniczeniami CHECK, UNIQUE i FOREIGN KEY

Sprzęt i materiały

- komputer z XAMPP i bazą sklep
- plik `sklep.sql` (do ponownego importu na końcu zajęć)

Oprogramowanie

- **Konsola mysql lub phpMyAdmin** — dokumentacja kluczy obcych MariaDB; ograniczenia CHECK:
<https://mariadb.com/kb/en/constraint/>
mariadb.com/kb/en/foreign-keys

Przygotowanie przed zajęciami

- Zaimportuj ponownie `sklep.sql`, żeby zacząć od danych wzorcowych.
- Przygotuj w sprawozdaniu kolumnę „Przewidywanie” — wpisz ją przed wykonaniem każdego polecenia.

Bezpieczeństwo

- Polecenia z kroków 1–2 celowo zmieniają i usuwają dane w bazie sklep. Na koniec zajęć zaimportuj `sklep.sql` ponownie, bo w kolejnych zadaniach potrzebne są dane wzorcowe.
- Na prawdziwej bazie ON DELETE CASCADE może jednym poleceniem usunąć setki powiązanych wierszy — zawsze najpierw sprawdź, co zostanie usunięte.

Przebieg zadania

1 Naruszenia ograniczeń w bazie sklep

Wykonaj polecenia po kolei w bazie sklep. Każde powinno zakończyć się błędem — zapisz jego numer i jednym zdaniem przyczynę. Zapisuj polecenia w pliku `zad11_nazwisko.sql`.

```
-- Z1
DELETE FROM klienci WHERE id_klienta = 1;
-- Z2
INSERT INTO zamowienia (id_klienta, data_zamowienia) VALUES (99, '2026-10-20 10:00:00');
-- Z3
INSERT INTO klienci (imie, nazwisko, email, data_rejestracji)
VALUES ('Anna', 'Nowacka', 'anna.nowak@example.pl', '2026-10-20');
-- Z4
INSERT INTO produkty (nazwa, id_kategorii, cena, data_dodania)
VALUES ('Gratis', 3, 0, '2026-10-20');
-- Z5
DELETE FROM produkty WHERE id_produktu = 7;
-- Z6
INSERT INTO klienci (imie, nazwisko, data_rejestracji) VALUES ('Jan', 'Bez', '2026-10-20');
```

1. Przed każdym poleceniem zapisz przewidywanie, potem wykonaj je i zapisz komunikat.
2. Przy Z1 i Z5 odczytaj z komunikatu, który klucz obcy (nazwa ograniczenia) zablokował operację. Sprawdź w `SHOW CREATE TABLE`, jaką ma akcję ON DELETE.
3. Przy Z4 zwróć uwagę na numer błędu: MariaDB zgłasza 4025, MySQL 8.4 — 3819 (Check constraint is violated). Znaczenie jest to samo.

2 Akcje kluczy obcych w bazie sklep

Teraz polecenia się udadzą — sprawdzaj, co stało się z wierszami w tabelach podrzędnych.

1. Z7 — wyświetl produkty z kategorii 5, usuń kategorię 5 (Drukarki) i ponownie wyświetl produkt 13. Jaką wartość ma `id_kategorii` i która akcja (ON DELETE ...) to spowodowała?
2. Z8 — policz wiersze w `pozycje_zamowienia`, usuń zamówienie nr 4 (anulowane) i policz ponownie. Która akcja zadziałała?
3. Z9 — zmień identyfikator kategorii 6 na 60 (`UPDATE kategorie SET id_kategorii = 60 WHERE id_kategorii = 6;`) i sprawdź `id_kategorii` produktu 14. Następnie spróbuj zmienić `id_klienta` 2 na 100. Dlaczego jedna zmiana się udała, a druga nie?
4. Z10 — spróbuj dodać ograniczenie `ALTER TABLE produkty ADD CONSTRAINT chk_stan CHECK (stan <= 50);`. Dlaczego serwer odmówił? Z11 — spróbuj wstawić produkt ze stanem -3. Które ograniczenie zadziałało, choć nie jest to CHECK?
5. Wypełnij tabelę akcji kluczy obcych bazy sklep (która tabela, ON DELETE, ON UPDATE, przykład skutku).

3 Własna baza z ograniczeniami

Utwórz bazę szkola (utf8mb4, utf8mb4_polish_ci) i zaprojektuj ograniczenia według reguł. Nazwij każde ograniczenie (CONSTRAINT nazwa ...).

```
nauczyciele(id_nauczyciela, imie, nazwisko)
klasy(id_klasy, symbol, id_wychowawcy)
  - symbol klasy jest unikalny; odejście wychowawcy nie usuwa klasy (pole staje się puste)
uczniowie(id_ucznia, imie, nazwisko, id_klasy)
  - uczeń musi mieć klasę; nie wolno usunąć klasy, w której są uczniowie
oceny(id_oceny, id_ucznia, przedmiot, ocena, data_wystawienia)
  - ocena od 1 do 6; przedmiot domyślnie 'bazy danych'
  - usunięcie ucznia usuwa jego oceny

INSERT INTO nauczyciele (imie, nazwisko) VALUES ('Jan', 'Kos'), ('Ewa', 'Lis');
INSERT INTO klasy (symbol, id_wychowawcy) VALUES ('1TI', 1), ('1TP', 2);
INSERT INTO uczniowie (imie, nazwisko, id_klasy) VALUES ('Ola', 'Mak', 1), ('Igor', 'Ryś', 1), ('Kuba', 'Wilk', 2);
INSERT INTO oceny (id_ucznia, ocena, data_wystawienia)
VALUES (1, 5, '2026-10-01'), (1, 4, '2026-10-08'), (2, 3, '2026-10-08');
```

1. Z12 — napisz polecenia CREATE TABLE dla czterech tabel tak, aby baza sama pilnowała wszystkich reguł z ramki. Wstaw dane z ramki.
2. Z13 — sprawdź każdą regułę osobnym poleceniem: ocena 7, druga klasa o symbolu 1TI, usunięcie klasy 1 (są w niej uczniowie). Zapisz numery błędów.
3. Z14 — usuń ucznia nr 1 i policz oceny. Usuń nauczyciela nr 2 i wyświetl tabelę klasy. Czy wynik zgadza się z regułami?
4. Z15 — wyświetl `SHOW CREATE TABLE oceny;` i sprawdź, czy ograniczenia mają nadane przez ciebie nazwy.
5. Na koniec zaimportuj ponownie `sklep.sql` i sprawdź, że kategorii jest znów 6.

W sprawozdaniu umieść

- ☐ tabela testów Z1–Z11: przewidywanie, numer błędu, przyczyna
- ☐ tabela akcji kluczy obcych bazy sklep
- ☐ polecenia CREATE TABLE bazy szkola (Z12)
- ☐ wyniki testów reguł Z13–Z15
- ☐ odpowiedzi na pytania kontrolne

Pytania kontrolne

1. Czym różni się błąd 1451 od 1452? Który dotyczy tabeli nadrzędnej, a który podrzędnej?
2. Kiedy wybierzesz ON DELETE CASCADE, a kiedy RESTRICT? Podaj przykład z bazy sklep, w którym CASCADE byłoby błędem.
3. Dlaczego ON DELETE SET NULL wymaga, żeby kolumna klucza obcego dopuszczała NULL?
4. Dlaczego warto nazywać ograniczenia (CONSTRAINT nazwa), zamiast zostawić nazwy nadane przez serwer?

Kryteria oceny

Kryterium	Punkty
Testy naruszeń Z1–Z6 z przewidywaniem i wyjaśnieniem błędów	5
Akcje kluczy obcych Z7–Z11 i tabela akcji	4
Baza szkola: poprawne ograniczenia realizujące wszystkie reguły (Z12)	5
Testy reguł Z13–Z15	3
Odpowiedzi na pytania kontrolne	3
Razem	20

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

DML — INSERT, UPDATE i DELETE

Czas: 2 × 45 min

praca indywidualna

Cel: Dodawanie, zmienianie i usuwanie danych poleceniami INSERT, UPDATE i DELETE oraz nawyk bezpiecznej pracy: najpierw SELECT z tym samym warunkiem, a przy ryzykownych zmianach — transakcja z możliwością ROLLBACK.

Po zajęciach potrafisz:

- wstawia wiele wierszy jednym poleceniem INSERT i kopiuje dane poleceniem INSERT ... SELECT
- zmienia dane poleceniem UPDATE z warunkiem, także z wyrażeniem (podwyżka o 10%)
- usuwa wybrane wiersze poleceniem DELETE i przewiduje wpływ kluczy obcych
- sprawdza zakres zmian zapytaniem SELECT przed UPDATE i DELETE
- używa transakcji (START TRANSACTION, COMMIT, ROLLBACK) jako zabezpieczenia

Sprzęt i materiały

- komputer z XAMPP
- plik `sklep.sql` z folderu Pliki_do_zadan

Oprogramowanie

- **Konsola mysql (klient MariaDB)** — transakcje wykonuj w konsoli — w phpMyAdmin każde „Wykonaj” to osobne połączenie
mariadb.com/kb/en/mariadb-command-line-client
- **phpMyAdmin** — podgląd danych
phpmyadmin.net

Przygotowanie przed zajęciami

- Zaimportuj ponownie `sklep.sql`, aby zacząć od danych wzorcowych — wyniki zależą od kolejności poleceń.

Bezpieczeństwo

- Zasada na całe życie: przed każdym UPDATE i DELETE wykonaj SELECT z dokładnie tym samym WHERE i sprawdź liczbę wierszy.
- UPDATE lub DELETE bez WHERE zmienia całą tabelę. Na serwerze produkcyjnym przed zmianą danych wykonaj kopię zapasową albo pracuj w transakcji.
- Po zajęciach zaimportuj `sklep.sql` ponownie, żeby kolejne zadania zaczynały się od tych samych danych.

Przebieg zadania

1 INSERT — nowe dane

Zaloguj się do konsoli (`mysql -u root --default-character-set=utf8mb4`), wybierz bazę sklep. Polecenia zapisuj w pliku `zad12_nazwisko.sql`.

1. Z1 — jednym poleceniem INSERT dodaj trzech klientów: Oskar Bielecki (oskar.b@example.pl, Lublin), Hanna Czarnecka (hanna.cz@example.pl, Kraków), Filip Duda (filip.duda@example.pl, bez miasta); data rejestracji 2026-10-20. Wymień kolumny jawnie, bez id_klienta. Jakie identyfikatory dostali?
2. Z2 — dodaj kategorię Sieci i sprawdź jej identyfikator (`SELECT LAST_INSERT_ID();`). Jednym poleceniem dodaj do niej dwa produkty: Router Wi-Fi 7 (899.00 zł, 6 szt.) i Switch 8-portowy (159.00 zł, 15 szt.), data dodania 2026-10-20.
3. Z3 — utwórz tabelę archiwum_zamowien o strukturze tabeli zamowienia (`CREATE TABLE ... LIKE`) i skopiuj do niej poleceniem `INSERT ... SELECT` wszystkie zamówienia o statusie wyslane. Ile wierszy skopiowano?

2 UPDATE — zmiany z warunkiem

Przy każdym poleceniu: najpierw SELECT z tym samym WHERE (zapisz liczbę wierszy), potem UPDATE, potem sprawdzenie.

1. Z4 — podnieś o 10% ceny wszystkich produktów z kategorii Akcesoria (id 3), zaokrąglając do grosza: `SET cena = ROUND(cena * 1.10, 2)`. Zapisz ceny przed i po zmianie.
2. Z5 — zamówienia o statusie nowe złożone przed 1 lipca 2026 r. zmień na anulowane (dwa warunki połączone AND).
3. Z6 — klientom bez podanego miasta wpisz miasto Lublin (pamiętaj: porównanie z NULL wymaga `IS NULL`). Ilu klientów to objęło i dlaczego tylu?
4. Z7 — produktom o stanie 0 zwiększ stan o 10 sztuk (`SET stan = stan + 10`).

3 DELETE — usuwanie

1. Z8 — usuń klientów dodanych w Z1 (warunek na id_klienta). Najpierw SELECT!
2. Z9 — spróbuj usunąć klienta nr 2. Zapisz błąd i wyjaśnij, dlaczego klienta z zamówieniami nie da się usunąć (zadanie 11).
3. Z10 — usuń tabelę archiwum_zamowien (to DDL, nie DML — czym różni się od DELETE FROM archiwum_zamowien?).

4 Transakcje jako zabezpieczenie

Domyślnie każde polecenie jest od razu zatwierdzane (autocommit = 1). Po START TRANSACTION zmiany są widoczne tylko w twojej sesji, dopóki nie wykonasz COMMIT; ROLLBACK je cofa.

```
-- Z11: „pomyłka” pod kontrolą
START TRANSACTION;
DELETE FROM pozycje_zamowienia;          -- zapomniany WHERE!
SELECT COUNT(*) FROM pozycje_zamowienia;
ROLLBACK;
SELECT COUNT(*) FROM pozycje_zamowienia;
```

1. Sprawdź `SELECT @@autocommit;`. Wykonaj Z11 z ramki i zapisz obie liczby.
2. Z12 — w jednej transakcji zapisz sprzedaż: nowe zamówienie klientki nr 3 (data 2026-10-20 12:00:00, status nowe), pozycja: produkt 10, 2 sztuki po 699.00 (id zamówienia weź z `LAST_INSERT_ID()`), zmniejszenie stanu produktu 10 o 2. Zatwierdź COMMIT i sprawdź wszystkie trzy zmiany.
3. Z13 — rozpocznij transakcję, zwiększ stan produktu 1 o 5, a potem spróbuj zmniejszyć stan produktu 2 o 5 (na stanie są 4 sztuki). Zapisz błąd. Czy pierwsza zmiana nadal obowiązuje w transakcji? Wykonaj ROLLBACK i sprawdź stan obu produktów.
4. Z14 — wyloguj się i zaloguj ponownie z opcją `--safe-updates`. Spróbuj wykonać `UPDATE produkty SET cena = cena * 2;`. Zapisz komunikat — ten tryb chroni przed UPDATE i DELETE bez warunku na kluczu.

W sprawozdaniu umieść

- ☐ plik `zad12_nazwisko.sql` z poleceniami Z1–Z14 (z zapytaniami kontrolnymi SELECT)
- ☐ tabela poleceń z liczbą wierszy w SELECT kontrolnym i w poleceniu zmieniającym
- ☐ ceny akcesoriów przed i po podwyżce (Z4)
- ☐ wyniki eksperymentów z transakcjami (Z11–Z13) i komunikat trybu safe-updates (Z14)
- ☐ odpowiedzi na pytania kontrolne

Pytania kontrolne

1. Dlaczego w poleceniu INSERT warto wymieniać nazwy kolumn, nawet gdy podajemy wartości dla wszystkich?
2. Co się stanie po wykonaniu `UPDATE produkty SET cena = cena * 1.10;` bez WHERE? Jak można to odwrócić, a jak — zabezpieczyć się przed tym?
3. Dlaczego warunek `WHERE miasto = NULL` nie znajduje żadnego wiersza?
4. Dlaczego sprzedaż (zamówienie, pozycja, zmiana stanu) powinna być jedną transakcją?

Kryteria oceny

Kryterium	Punkty
INSERT wielu wierszy i INSERT ... SELECT (Z1–Z3)	4
UPDATE z warunkami, poprzedzone SELECT (Z4–Z7)	5
DELETE i rozumienie kluczy obcych (Z8–Z10)	3
Transakcje i safe-updates (Z11–Z14) z wnioskami	5
Odpowiedzi na pytania kontrolne	3
Razem	20

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

Import i eksport danych (CSV)

Czas: 2 × 45 min

praca indywidualna

Cel: Importowanie danych z plików CSV do tabel (phpMyAdmin i LOAD DATA) z poprawnym kodowaniem, separatorem i formatem liczb oraz eksportowanie danych do CSV.

Po zajęciach potrafisz:

- rozpoznaje w pliku CSV kodowanie, separator, nagłówek i znak końca wiersza
- importuje plik CSV w phpMyAdmin i poleceniem LOAD DATA LOCAL INFILE
- przekształca dane podczas importu (przecinek dziesiętny, format daty, kodowanie Windows-1250)
- eksportuje dane do CSV w phpMyAdmin i poleceniem SELECT ... INTO OUTFILE, rozumie rolę secure_file_priv
- sprawdza kompletność importu zapytaniem kontrolnym (COUNT, SUM)

Sprzęt i materiały

- komputer z Windows 11 i zainstalowanym XAMPP (Apache + MariaDB/MySQL)
- pliki `r1_13_import.sql`, `r1_13_klienci_utf8.csv` i `r1_13_cennik_excel.csv` z folderu Pliki_do_zadan (skopiuj folder np. do `C:\Pliki_do_zadan`)
- baza `sklep` (plik `sklep.sql`)

Oprogramowanie

- **XAMPP** — phpMyAdmin i konsola `mysql` [apachefriends.org/download.html](https://www.apachefriends.org/download.html)
- **Visual Studio Code** — podgląd CSV: kodowanie i znak końca wiersza widać na pasku stanu code.visualstudio.com
- **LibreOffice Calc lub Excel** — otwarcie wyeksportowanego pliku pl.libreoffice.org/pobierz

Przygotowanie przed zajęciami

- Skopiuj folder Pliki_do_zadan na dysk lokalny tak, by ścieżka nie zawierała spacji ani polskich liter (np. `C:\Pliki_do_zadan`).

Bezpieczeństwo

- Importuj tylko do bazy `import_danych` (tworzy ją plik z zadania). Bazy `sklep` w tym zadaniu nie zmieniasz — tylko z niej eksportujesz.
- Opcja `local_infile` pozwala serwerowi czytać pliki z komputera klienta. Jeśli włączasz ją w MySQL 8.4, rób to tylko na własnym komputerze i wyłącz po zajęciach.
- Nie eksportuj danych osobowych (np. tabeli klientów) na pendrive'y ani do chmury — w zadaniu eksportujesz tylko tabelę produktów.

Przebieg zadania

1 Przygotowanie i oglądnięcie plików

1. Uruchom XAMPP (Apache i MySQL) i zaimportuj plik `r1_13_import.sql` (phpMyAdmin → Import). Sprawdź, że powstała baza `import_danych` z pustymi tabelami `klienci_nowi` i `cennik`.
2. Otwórz oba pliki CSV w VS Code. Dla każdego zapisz w tabeli sprawozdania: separator pól, czy jest nagłówek, kodowanie i znak końca wiersza (prawy dolny róg okna: np. UTF-8, LF/CRLF).
3. Jeśli w `r1_13_cennik_excel.csv` polskie litery wyglądają źle, kliknij kodowanie na pasku stanu → Reopen with Encoding → Central European (Windows 1250). Nie zapisuj pliku.

2 Import w phpMyAdmin

Tabela `klienci_nowi` ma kolumnę `id_klienta` (AUTO_INCREMENT), której w pliku nie ma — dlatego podajesz listę kolumn.

1. Z1. Zaimportuj `r1_13_klienci_utf8.csv`: baza `import_danych` → tabela `klienci_nowi` → Import. Ustaw: Kodowanie znaków pliku: utf-8; Format: CSV; Kolumny oddzielone: `;`; Kolumny dołączyć: `"`; Nazwy kolumn: `imie,nazwisko,email,miasto,data_rejestracji`; w części „Częściowy import” pole „Pomiń tę liczbę zapytań (dla SQL), zaczynając od pierwszej” ustaw na 1 (pominie wiersz nagłówka). Kliknij Import.
2. Z2. Napisz zapytanie, które policzy zaimportowane wiersze, oraz zapytanie wyświetlające klientów z miast zaczynających się od „Ł” lub „Bi” — sprawdź polskie znaki.
3. Z3. Jedna klientka nie podała miasta. Sprawdź dwoma zapytaniami, czy w kolumnie `miasto` jest NULL, czy pusty tekst `''`. Jeśli pusty tekst — napisz UPDATE, który zamieni puste miasta na NULL (najpierw SELECT z tym samym WHERE).

3 Import poleceniem LOAD DATA

Uruchom konsolę z obsługą plików lokalnych i polskich znaków. W MySQL 8.4 opcja serwera `local_infile` jest domyślnie wyłączona — sprawdź ją i w razie potrzeby włącz.

```
chcp 65001
mysql -u root --default-character-set=utf8mb4 --local-infile=1

SHOW GLOBAL VARIABLES LIKE 'local_infile';  -- OFF? wtedy: SET GLOBAL local_infile = 1;

-- szablon polecenia (uzupełnij):
LOAD DATA LOCAL INFILE 'C:/Pliki_do_zadan/plik.csv'  -- ukośniki /, nie \
INTO TABLE tabela
CHARACTER SET ...                                -- kodowanie pliku
FIELDS TERMINATED BY '...' OPTIONALLY ENCLOSED BY '"'
LINES TERMINATED BY '...'                        -- '\n' albo '\r\n'
IGNORE 1 LINES
(kolumna1, kolumna2, @zmienna)
SET kolumna3 = funkcja(@zmienna);
```

1. Z4. Wyczyść tabelę `klienci_nowi` (TRUNCATE) i zaimportuj ten sam plik poleceniem LOAD DATA LOCAL INFILE. Sprawdź liczbę wierszy i komunikat `Records/Skipped/Warnings`.
2. Z5. Zaimportuj `r1_13_cennik_excel.csv` do tabeli `cennik` „naiwnie”: bez CHARACTER SET, kolumny wprost (kod, nazwa, cena, data_od). Wykonaj `SHOW WARNINGS`; i obejrzyj dane: ceny, daty i polskie litery. Zapisz, co jest nie tak.
3. Z6. Wyczyść tabelę `cennik` i zaimportuj plik poprawnie: CHARACTER SET cp1250, cena i data przez zmienne `@cena`, `@data`, a w SET zamiana przecinka na kropkę (`REPLACE`) i zamiana daty funkcją `STR_TO_DATE(@data, '%d.%m.%Y')`.
4. Z7. Zapytanie kontrolne: liczba wierszy, suma cen, najwcześniejsza i najpóźniejsza data w tabeli `cennik`. Porównaj z plikiem (liczba wierszy bez nagłówka, przykładowe ceny).

4 Eksport danych

SELECT ... INTO OUTFILE zapisuje plik na komputerze serwera (u Ciebie — ten sam komputer). Zmienna `secure_file_priv` określa, w jakim folderze wolno zapisywać: pusta — bez ograniczeń, NULL — eksport wyłączony, ścieżka — tylko ten folder.

1. Z8. W phpMyAdmin: baza `sklep` → tabela `produkty` → Eksport → Metoda eksportu: „Dostosuj – wyświetl wszystkie możliwe opcje” → Format: „CSV dla MS Excel” → zaznacz „Umieść nazwy kolumn w pierwszym wierszu” → Eksport. Otwórz plik w arkuszu i sprawdź kolumny oraz polskie znaki.
2. Z9. Sprawdź wartość `secure_file_priv` (SHOW VARIABLES LIKE ...). Ustal folder, do którego możesz eksportować (w XAMPP np. `C:/xampp/tmp/`, w MySQL 8.4 folder podany w zmiennej).
3. Z10. Wyeksportuj wszystkie kolumny tabeli `sklep.produkty` poleceniem `SELECT * INTO OUTFILE '...' ... FROM sklep.produkty` (opcje pliku przed FROM) do pliku `produkty_eksport.csv`: CHARACTER SET utf8mb4, separator `;`, pola tekstowe w cudzysłowie, wiersze zakończone `\r\n`. Obejrzyj plik w VS Code: jak zapisano NULL w kolumnie kategorii huba USB-C i cudzysłów w nazwach monitorów?
4. Z11. Wykonaj Z10 jeszcze raz bez zmiany nazwy pliku i zanotuj numer błędu.
5. Z12. Wyeksportuj do pliku `produkty_naglowek.csv` te same dane z wierszem nagłówka. Wskazówka: `SELECT 'id_produktu', 'nazwa', ... UNION ALL SELECT ... FROM sklep.produkty INTO OUTFILE ...`.
6. Z13. Test „w obie strony”: w bazie `import_danych` utwórz tabelę `produkty_kopia` o strukturze jak `sklep.produkty` (`CREATE TABLE ... LIKE ...`) i wczytaj do niej plik z Z10 poleceniem LOAD DATA INFILE (bez LOCAL — plik leży na serwerze) z tymi samymi opcjami.
7. Z14. Zapytanie kontrolne: porównaj w obu tabelach liczbę wierszy, sumę wartości magazynu (`cena × stan`) i liczbę produktów bez kategorii. Zrób zrzut ekranu. Jeśli w Z4 włączyłeś `local_infile` w MySQL 8.4, wyłącz ją (`SET GLOBAL local_infile = 0;`).

W sprawozdaniu umieść

- ☐ tabela parametrów obu plików CSV (separator, nagłówki, kodowanie, koniec wiersza)
- ☐ tabela Z1–Z14 z poleceniami SQL (dla Z1 i Z8 — opis ustawień) i wynikami
- ☐ zrzut ustawień importu w phpMyAdmin (Z1) i wyniku Z3
- ☐ opis problemów z importu naiwnego (Z5) — ostrzeżenia i błędne dane — oraz poprawione polecenie (Z6)
- ☐ zrzut wyników zapytań kontrolnych Z7 i Z14
- ☐ odpowiedzi na pytania kontrolne

Pytania kontrolne

1. Dlaczego plik CSV zapisany w polskim Excelu ma zwykle średnik zamiast przecinka jako separator?
2. Co się stało z ceną 49,90 przy imporcie naiwnym i dlaczego serwer nie przerwał importu błędem? (Podpowiedź: słowo LOCAL.)
3. Czym różni się LOAD DATA LOCAL INFILE od LOAD DATA INFILE? Który plik czyta serwer w każdym z przypadków?
4. Do czego służy zmienna `secure_file_priv` i dlaczego dobrze, że w MySQL 8.4 domyślnie ogranicza eksport do jednego folderu?
5. Jak w pliku CSV zapisać wartość NULL, żeby LOAD DATA wczytało ją jako NULL, a nie jako pusty tekst?

Kryteria oceny

Kryterium	Punkty
Oględziny plików i import w phpMyAdmin (Z1–Z3)	4
LOAD DATA: poprawny import obu plików z przekształceniami (Z4–Z7)	6
Eksport phpMyAdmin i INTO OUTFILE, błąd 1086, nagłówki (Z8–Z12)	4
Test w obie strony i zapytania kontrolne (Z13–Z14)	3
Odpowiedzi na pytania kontrolne	3
Razem	20

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

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 zapytaniami; 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).

WHERE dla zaawansowanych: LIKE, IN, BETWEEN, NULL, ORDER BY, LIMIT

Czas: 2 × 45 min

praca indywidualna

Cel: Stosowanie wzorców LIKE, list IN, zakresów BETWEEN i warunków IS NULL oraz sortowanie i stronicowanie wyników ORDER BY i LIMIT.

Po zajęciach potrafisz:

- wyszukuje teksty według wzorca z symbolami % i _
- stosuje IN, BETWEEN i IS NULL / IS NOT NULL zamiast długich warunków OR
- unika pułapek: BETWEEN dla daty z godziną, porównanie z NULL, NOT IN przy brakujących danych
- sortuje wyniki według kilku kolumn i ogranicza ich liczbę LIMIT/OFFSET
- oblicza parametry LIMIT dla wybranej strony wyników

Sprzęt i materiały

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

Oprogramowanie

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

Przygotowanie przed zajęciami

- Miej pod ręką plik z zadania 14 — tę samą konwencję (komentarz `-- Zn.` przed zapytaniem) stosujesz w pliku `zad15_nazwisko.sql`.

Bezpieczeństwo

- Wykonujesz tylko zapytania SELECT na własnej bazie `sklep`; nie zmieniasz danych i nie usuwasz baz.

Przebieg zadania

1 Wzorce LIKE

`%` — dowolny ciąg znaków (także pusty), `_` — dokładnie jeden znak. Przy porównaniu `utf8mb4_polish_ci` wielkość liter nie ma znaczenia.

1. Uruchom XAMPP, otwórz konsolę lub phpMyAdmin i wybierz bazę `sklep`. Jeśli dane były zmieniane, zaimportuj ponownie `sklep.sql`.
2. Z1. Produkty, których nazwa zaczyna się od słowa „Laptop”. Sprawdź, czy wzorzec `'laptop%'` (małe litery) daje ten sam wynik.
3. Z2. Klienci, których nazwisko kończy się na „ska”.
4. Z3. Klienci, których adres e-mail zaczyna się od jednej litery i kropki (np. p.kowalski@...). Użyj symbolu `_`.
5. Z4. Produkty, w których nazwie występuje znak cała `.`.

2 IN, BETWEEN i daty

1. Z5. Klienci z Krakowa lub Gdańska — użyj IN.
2. Z6. Klienci spoza Krakowa i Warszawy — użyj NOT IN. Policz: ilu klientów jest razem w Z6 oraz w Krakowie i Warszawie? Porównaj z liczbą wszystkich klientów.
3. Z7. Produkty w cenie od 300 do 700 zł (BETWEEN). Sprawdź na danych, czy granice zakresu są włączone.
4. Z8. Zamówienia złożone od 10 do 14 lutego 2026 r. — zapisz warunek `data_zamowienia BETWEEN '2026-02-10' AND '2026-02-14'`. Obejrzyj tabelę `zamowienia` i sprawdź, czy wynik jest kompletny.
5. Z9. Popraw Z8 tak, aby obejmował cały dzień 14 lutego (warunek `>=` dla początku i `<` dla dnia następnego).

3 NULL — brak wartości

NULL to „nieznane”. Każde porównanie z NULL operatorem `=` lub `<>` daje NULL, a nie prawdę.

```
SELECT NULL = NULL, NULL <> 1, NULL IS NULL, NULL <=> NULL;
```

1. Z10. a) Klienci, którzy nie podali miasta. b) Produkty bez przypisanej kategorii.
2. Z11. Napisz Z10a z warunkiem `miasto = NULL` i zanotuj liczbę wierszy. Wykonaj zapytanie z ramki powyżej i wyjaśnij każdą z czterech wartości.

4 ORDER BY i LIMIT

ASC — rosnąco (domyślnie), DESC — malejąco. `LIMIT n OFFSET k` pomija k wierszy i zwraca n; zapis skrócony: `LIMIT k, n`.

1. Z12. Nazwa i cena wszystkich produktów od najdroższego do najtańszego.
2. Z13. Klienci posortowani według miasta rosnąco, a w obrębie miasta według nazwiska. Gdzie znalazł się klient bez miasta?
3. Z14. Trzy najdroższe produkty (nazwa, cena).
4. Z15. Stronicowanie: lista produktów posortowana według nazwy, po 5 na stronę. Napisz zapytania dla strony 2 (Z15a) i strony 3 (Z15b). Zapisz wzór na OFFSET dla strony numer s.
5. Z16. Produkty dostępne w magazynie (`stan > 0`) z kategorii 1, 2 lub 4, w cenie od 500 do 3500 zł, posortowane według kategorii rosnąco, a w kategorii według ceny malejąco.
6. Zapisz plik `zad15_nazwisko.sql`. Zrób zrzuty ekranu wyników Z8, Z9, Z11 (ramka NULL), Z13 i Z15a.

W sprawozdaniu umieść

- ☐ tabela Z1–Z16 z treścią zapytań i liczbą wierszy
- ☐ zrzuty ekranu Z8, Z9, Z11, Z13 i Z15a

- ☐ wyjaśnienie różnicy między Z8 i Z9
- ☐ wyjaśnienie wyników Z6 i Z11
- ☐ wzór na OFFSET dla strony s przy n wierszach na stronę
- ☐ odpowiedzi na pytania kontrolne

Pytania kontrolne

1. Czym różni się symbol `%` od `_` we wzorcu LIKE? Podaj przykład wzorca, który znajdzie kody pocztowe z Krakowa (zaczynające się od 30- lub 31-).
2. Dlaczego warunek BETWEEN '2026-02-10' AND '2026-02-14' dla kolumny DATETIME pomija zamówienia z popołudnia 14 lutego?
3. Dlaczego `WHERE miasto = NULL` nigdy nie zwraca wierszy? Jak poprawnie sprawdzić brak wartości?
4. Dlaczego LIMIT bez ORDER BY nie gwarantuje, że uczeń i nauczyciel zobaczą te same wiersze?

Kryteria oceny

Kryterium	Punkty
Z1-Z4 — poprawne wzorce LIKE	3
Z5-Z9 — IN, BETWEEN, poprawka zakresu dat i wyjaśnienia	5
Z10-Z11 — obsługa NULL i wyjaśnienie ramki	3
Z12-Z16 — sortowanie, LIMIT, stronicowanie, zapytanie złożone	5
Plik .sql, zrzuty ekranu i odpowiedzi na pytania	4
Razem	20

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

Zadanie kontrolne 1: DDL, DML i SELECT

Czas: 2 × 45 min

praca samodzielna (sprawdzian), 90 min

Cel: Samodzielne utworzenie bazy danych według opisu, wprowadzenie i modyfikacja danych oraz napisanie zapytań SELECT — sprawdzian umiejętności z zadań 9–15.

Po zajęciach potrafisz:

- tworzy bazę i tabele z kluczami, ograniczeniami i poprawnymi typami według opisu
- wstawia, zmienia i usuwa dane poleceniami INSERT, UPDATE i DELETE
- sprawdza działanie ograniczeń integralności i odczytuje komunikaty błędów
- pisze zapytania SELECT z warunkami, sortowaniem, limitem i wyrażeniami
- dokumentuje pracę w ustalonym formacie (plik z kwerendami i zrzuty ekranu)

Sprzęt i materiały

- komputer z Windows 11 i zainstalowanym XAMPP (Apache + MariaDB/MySQL)
- folder na wyniki na pulpicie, nazwany numerem z dziennika i nazwiskiem, np. `07_Kowalski`

Oprogramowanie

- **XAMPP** — phpMyAdmin i konsola `mysql` apachefriends.org/download.html
- **Notatnik, Notepad++ lub VS Code** — plik `kwerendy.txt` code.visualstudio.com
- **Narzędzie Wycinanie (Win + Shift + S)** — zrzuty ekranu

Przygotowanie przed zajęciami

- Powtórz zadania 9, 11, 12, 14 i 15 (CREATE TABLE, klucze obce, INSERT/UPDATE/DELETE, SELECT z WHERE, ORDER BY i LIMIT).
- Podczas sprawdzianu możesz korzystać wyłącznie z dokumentacji MariaDB/MySQL w sieci (mariadb.com/kb, dev.mysql.com/doc) — bez czatów i komunikatorów.

Bezpieczeństwo

- Pracuj tylko w nowej bazie `planszowki`; nie usuwaj bazy `sklep` ani innych baz.
- Przed UPDATE i DELETE wykonaj SELECT z tym samym warunkiem WHERE i sprawdź, których wierszy dotyczy zmiana.

Przebieg zadania

1 Opis bazy

Świetlica szkolna prowadzi wypożyczalnię gier planszowych. Utwórz bazę `planszowki` (kodowanie utf8mb4, porównywanie utf8mb4_polish_ci) z dwiema tabelami InnoDB:

```
wydawcy
  id_wydawcy  liczba całkowita bez znaku, klucz główny, numerowana automatycznie
  nazwa       tekst do 60 znaków, wymagany, unikalny
  kraj        tekst do 40 znaków, wymagany, domyślnie 'Polska'

gry
  id_gry      liczba całkowita bez znaku, klucz główny, numerowana automatycznie
  tytuł       tekst do 80 znaków, wymagany
  id_wydawcy  wymagany, klucz obcy do wydawcy (usunięcie wydawcy z grami ma być zablokowane)
  rok_wydania liczba całkowita bez znaku (rok, np. 2025)
  min_graczy  mała liczba całkowita bez znaku, wymagana
  max_graczy  mała liczba całkowita bez znaku, wymagana; nie mniejsza niż min_graczy (CHECK)
  czas_gry    czas rozgrywki w minutach, liczba całkowita bez znaku
  cena        kwota z dokładnością do grosza, wymagana, większa od 0 (CHECK)
```

1. Z1. Utwórz bazę `planszowki`.
2. Z2. Utwórz tabelę `wydawcy`.
3. Z3. Utwórz tabelę `gry` z kluczem obcym i dwoma ograniczeniami CHECK. Wykonaj `SHOW CREATE TABLE gry;` i zrób zrzut `kw1.png`.

2 Dane

Wstawiaj dane wielowierszowymi poleceniami INSERT (wiele wierszy w jednym VALUES). Kolejność wierszy = kolejne identyfikatory.

```
wydawcy (nazwa, kraj):
  Kostka, Polska | Planszownia, Polska | Meeple Wawa, Polska |
  Jeux du Nord, Francja | Spielwerk, Niemcy

gry (tytuł, id_wydawcy, rok_wydania, min_graczy, max_graczy, czas_gry, cena):
  Wyprawa na Biegun      1 2024 2 4 60 149.90
  Kolej Bałtycka         2 2025 2 5 45 129.00
  Smoki i Skarby         4 2023 3 6 90 219.00
  Miasto Ogrodów         1 2026 1 4 40 99.90
  Szybkie Słowa          3 2022 4 10 15 59.90
  Zamek Strachów         4 2025 2 6 75 189.00
  Rejs po Wiśle          2 2024 2 4 30 79.00
  Pszczela Farma         5 2026 1 5 50 169.00
  Kosmiczna Kolonia      5 2023 1 4 120 249.00
  Detektyw z Łodzi      3 2025 1 6 90 139.00
```

1. Z4. Wstaw 5 wydawców. Dla trzech polskich wydawców nie podawaj kraju — niech zadziała wartość domyślna (pomiń kolumnę albo użyj słowa DEFAULT).
2. Z5. Wstaw 10 gier. Sprawdź `SELECT * FROM gry;` i zrób zrzut `kw2.png`.

3 Ograniczenia i modyfikacje

Polecenia Z6 i Z7 mają zakończyć się błędem — to sprawdzenie, czy ograniczenia działają. Zapisz numer i treść błędu.

1. Z6. Spróbuj dodać grę „Wyścig Ślimaków” wydawcy o identyfikatorze 9 (pozostałe dane wymyśl).
2. Z7. Spróbuj dodać grę „Duet” wydawcy 1, w której minimalna liczba graczy wynosi 3, a maksymalna 2.
3. Z8. Wydawca Planszownia podnosi ceny o 10%. Zaktualizuj ceny jego gier (zaokrąglaj do groszy).
4. Z9. Gra „Rejs po Wiśle” została wycofana — usuń ją z tabeli. Zrób zrzut tabeli `gry` po Z8 i Z9 jako `kw3.png`.

4 Zapytania SELECT

Wykonuj po Z8 i Z9. Każde zapytanie zapisz w pliku `kwerendy.txt` (z numerem), a wynik zapisz jako zrzut `kw4.png` ... `kw8.png`.

1. Z10. Tytuł, cena i czas gry dla gier, w które można grać w pojedynkę; sortowanie alfabetyczne według tytułu.
2. Z11. Gry na imprezę: tytuł, minimalna i maksymalna liczba graczy dla gier, w które może grać co najmniej 6 osób; od największej maksymalnej liczby graczy.
3. Z12. Gry wydane w latach 2024–2026, trwające nie dłużej niż godzinę: tytuł, rok, czas; sortowanie według roku malejąco, potem tytułu.
4. Z13. Trzy najdroższe gry: tytuł, cena oraz cena z rabatem 15% (kolumna `cena_rabat`, zaokrąglona do 2 miejsc).
5. Z14. Gry wydawców zagranicznych: najpierw odczytaj ich identyfikatory z tabeli `wydawcy`, potem wyświetl tytuł, id wydawcy i cenę tych gier (użyj IN), od najtańszej.

5 Oddanie pracy

1. Wyeksportuj bazę `planszowki` w phpMyAdmin (Eksport → Format SQL) do pliku `planszowki.sql` w swoim folderze.
2. Sprawdź, że w folderze są: `kwerendy.txt` (Z1–Z14), `kw1.png` ... `kw8.png` i `planszowki.sql`. Zgłoś nauczycielowi zakończenie pracy.

W sprawozdaniu umieść

- ☐ plik `kwerendy.txt` z poleceniami Z1–Z14 i komunikatami błędów Z6, Z7
- ☐ zrzuty `kw1.png` (SHOW CREATE TABLE gry), `kw2.png` (dane po wstawieniu), `kw3.png` (po UPDATE i DELETE), `kw4.png` – `kw8.png` (wyniki Z10–Z14)
- ☐ plik `planszowki.sql` (eksport bazy)
- ☐ w sprawozdaniu .docx: tabela poleceń i krótka samoocena

Pytania kontrolne

1. Dlaczego dla kolumny `cena` wybrałeś DECIMAL, a nie FLOAT?
2. Co oznacza błąd 1452 przy Z6 i który element struktury tabeli go wywołał?
3. Co stałoby się przy Z8, gdybyś zapomniał o warunku WHERE?

Kryteria oceny

Kryterium	Punkty
Z1–Z3: baza i tabele zgodne z opisem (typy, klucze, NOT NULL, DEFAULT, UNIQUE, CHECK, FK)	5
Z4–Z5: dane wstawione kompletnie, wartość domyślna kraju	2
Z6–Z7: testy ograniczeń z poprawnie odczytanymi błędami	2
Z8–Z9: poprawne UPDATE i DELETE z warunkiem	2
Z10–Z14: poprawne zapytania SELECT (po 1,4 pkt, w sumie 7)	7
Dokumentacja w ustalonym formacie (nazwy plików, zrzuty, eksport)	2
Razem	20

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

Funkcje tekstowe

Czas: 2 × 45 min

praca indywidualna

Cel: Stosowanie funkcji tekstowych MySQL/MariaDB do łączenia, wycinania, zamiany i formatowania tekstów oraz zrozumienie różnicy między bajtami a znakami w UTF-8.

Po zajęciach potrafisz:

- łączy teksty funkcjami CONCAT i CONCAT_WS i wie, jak traktują NULL
- porównuje LENGTH i CHAR_LENGTH dla tekstów z polskimi znakami
- wycina fragmenty tekstu funkcjami LEFT, RIGHT, SUBSTRING, SUBSTRING_INDEX i LOCATE
- zmienia teksty funkcjami UPPER, LOWER, REPLACE, TRIM i LPAD
- buduje z kilku funkcji wyrażenie, np. login użytkownika lub numer dokumentu

Sprzęt i materiały

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

Oprogramowanie

- **XAMPP** — phpMyAdmin i konsola `mysql` [apachefriends.org/download.html](https://www.apachefriends.org/download.html)
- **Dokumentacja funkcji tekstowych MariaDB** — MySQL: <https://dev.mysql.com/doc/refman/8.4/en/string-functions.html> mariadb.com/kb/en/string-functions

Bezpieczeństwo

- Wykonujesz tylko zapytania SELECT — funkcje zmieniają wynik zapytania, a nie dane w tabeli. Nie zapisuj wyników poleceniem UPDATE.

Przebieg zadania

1 Łączenie tekstów

Funkcję możesz najpierw sprawdzić bez tabeli, np. `SELECT UPPER('kraków');`. Zapytania zapisuj w pliku `zad17_nazwisko.sql`.

1. Otwórz bazę `sklep` (w razie potrzeby zaimportuj ponownie `sklep.sql`).
2. Z1. Lista klientów w jednej kolumnie `klient` w postaci „Nazwisko Imię”, posortowana alfabetycznie.
3. Z2. Kolumna `opis` w postaci „Imię Nazwisko (Miasto)” zbudowana funkcją CONCAT. Co widzisz dla klientki bez miasta?
4. Z3. To samo co Z2, ale funkcją CONCAT_WS z separatorem `' , '` w postaci „Nazwisko, Imię, Miasto”. Porównaj zachowanie z Z2.

2 Wielkość liter i długość

W UTF-8 litera „ś” zajmuje 2 bajty. LENGTH liczy bajty, CHAR_LENGTH — znaki.

1. Z4. Nazwisko wielkimi literami i e-mail małymi literami dla wszystkich klientów.
2. Z5. Nazwisko, LENGTH(nazwisko) i CHAR_LENGTH(nazwisko). Ilu klientów ma różne wartości obu funkcji? Napisz zapytanie, które wyświetli tylko tych klientów.
3. Z6. Klienci, których nazwisko ma więcej niż 8 znaków, od najdłuższego nazwiska.

3 Wycinanie fragmentów

LEFT(t, n), RIGHT(t, n), SUBSTRING(t, od, ile), LOCATE(szukany, t), SUBSTRING_INDEX(t, separator, n).

1. Z7. Inicjały klientów w postaci „A.N.” (kolumna `inicjaly`) obok imienia i nazwiska.
2. Z8. Z adresu e-mail wydziel nazwę użytkownika (przed @) i domenę (po @) — w dwóch kolumnach. Domenę wytnij za pomocą SUBSTRING i LOCATE, nazwę — dowolną metodą.
3. Z9. Dla monitorów wydziel z nazwy przekątną ekranu (liczba przed znakiem `"`) jako kolumnę `cale`. Wskazówka: dwukrotne SUBSTRING_INDEX.

4 Zamiana i formatowanie

```
SELECT CONCAT('[', '   Hub USB-C   ', ' ]'),  
       CONCAT('[', TRIM('   Hub USB-C   '), ' ]'),  
       CONCAT('[', LTRIM('   Hub USB-C   '), ' ]');
```

1. Z10. Wykonaj zapytanie z ramki i dopisz wersję z RTRIM. Opisz, co usuwa każda funkcja.
2. Z11. Dla laptopów wyświetl nazwę i nazwę po zamianie słowa „Laptop” na „Notebook” (REPLACE).
3. Z12. Numer dokumentu każdego zamówienia w postaci `ZAM/2026/0007` (rok z daty zamówienia, numer uzupełniony zerami do 4 cyfr funkcją LPAD) oraz status.
4. Z13. Login dla klientów: pierwsza litera imienia + nazwisko, małymi literami, bez polskich znaków (zagnieżdżone REPLACE dla liter występujących w nazwiskach: ą, ć, ę, ł, ń, ó, ś, ź, ż). Przykład: Zofia Wiśniewska → `zwisniewska`.
5. Z14. Do loginu z Z13 dopisz kolumnę `email_sluzbowy` w postaci `login@sklep.example.pl`. Sprawdź, czy wszystkie loginy są różne (porównaj liczbę wierszy z `SELECT DISTINCT`).
6. Zrób zrzuty ekranu Z2, Z3, Z5, Z9 i Z13.

W sprawozdaniu umieść

- ☐ tabela Z1–Z14 z treścią zapytań i krótkim opisem wyniku
- ☐ zrzuty ekranu Z2, Z3, Z5, Z9 i Z13
- ☐ porównanie CONCAT i CONCAT_WS wobec NULL
- ☐ wyjaśnienie różnicy LENGTH i CHAR_LENGTH
- ☐ odpowiedzi na pytania kontrolne

Pytania kontrolne

1. Dlaczego CONCAT z wartością NULL daje NULL, a CONCAT_WS nie? Jak w CONCAT zastąpić NULL tekstem (np. „brak”)?
2. Ile bajtów zajmuje w UTF-8 słowo „Łódź”, a ile ma znaków? Sprawdź zapytaniem.
3. Którą funkcję zastosujesz do sprawdzenia, czy pole formularza nie zostało wypełnione samymi spacjami?
4. Dlaczego login generowany z imienia i nazwiska może się powtórzyć i jak temu zapobiec w tabeli?

Kryteria oceny

Kryterium	Punkty
Z1-Z3 — łączenie tekstów i obsługa NULL	3
Z4-Z6 — wielkość liter i długość tekstu	3
Z7-Z9 — wycinanie fragmentów	4
Z10-Z14 — zamiana, formatowanie, login	5
Zrzuty, plik .sql i wyjaśnienia	2
Odpowiedzi na pytania kontrolne	3
Razem	20

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

Funkcje daty, liczbowe i CASE

Czas: 2 × 45 min

praca indywidualna

Cel: Stosowanie funkcji daty i czasu, funkcji zaokrąglających oraz wyrażenia CASE i funkcji IFNULL do przygotowania czytelnych raportów.

Po zajęciach potrafisz:

- odczytuje bieżącą datę i wydziela z daty rok, miesiąc, dzień i godzinę
- formatuje daty po polsku funkcją DATE_FORMAT z `lc_time_names`
- oblicza różnice dat i terminy (DATEDIFF, TIMESTAMPDIFF, DATE_ADD)
- zaokrągla liczby funkcjami ROUND, TRUNCATE, CEIL i FLOOR i dobiera właściwą do sytuacji
- kategoryzuje dane wyrażeniem CASE i zastępuje NULL funkcją IFNULL

Sprzęt i materiały

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

Oprogramowanie

- **XAMPP** — phpMyAdmin i konsola `mysql` [apachefriends.org/download.html](https://www.apachefriends.org/download.html)
- **Dokumentacja funkcji daty (MariaDB)** — formaty DATE_FORMAT: <https://mariadb.com/kb/en/date-format/>
mariadb.com/kb/en/date-time-functions

Bezpieczeństwo

- Wykonujesz tylko zapytania SELECT na własnej bazie `sklep`; nie zmieniasz danych (wyniki „podwyżki” i „rat” są tylko obliczeniami w zapytaniu).

Przebieg zadania

1 Bieżąca data i części daty

Ustaw polskie nazwy dni i miesięcy dla bieżącej sesji (w phpMyAdmin wpisz to polecenie w tym samym oknie SQL, przed zapytaniem):

```
SET lc_time_names = 'pl_PL';
```

1. Z1. Wyświetl bieżącą datę i godzinę (NOW), samą datę (CURDATE), samą godzinę (CURTIME) i nazwę dzisiejszego dnia tygodnia (DAYNAME).
2. Z2. Dla każdego zamówienia: numer, data oraz osobno rok, miesiąc, dzień i godzina złożenia.
3. Z3. Zamówienia złożone w maju 2026 r. — użyj funkcji YEAR i MONTH w warunku.
4. Z4. Numer zamówienia i data w formacie „10.02.2026, godz. 09:15” (DATE_FORMAT z `%d`, `%m`, `%Y`, `%H`, `%i`).
5. Z5. Data zamówienia w formacie „wtorek, 10 ... 2026” (`%W`, `%e`, `%M`, `%Y`). Czy nazwa miesiąca jest w poprawnej formie gramatycznej?

2 Obliczenia na datach

Żeby wyniki dało się porównać, Z6–Z8 liczą od stałej daty 1 września 2026 r. (w Z6b — od dzisiaj).

1. Z6. a) Liczba dni od rejestracji każdego klienta do 2026-09-01 (DATEDIFF), od najstarszego konta.
b) To samo względem dnia dzisiejszego (CURDATE).
2. Z7. Termin płatności dla zamówień o statusie `nowe`: 7 dni od daty złożenia (DATE_ADD), bez godziny (funkcja DATE).
3. Z8. Produkty, które są w ofercie co najmniej 9 pełnych miesięcy na dzień 2026-09-01 (TIMESTAMPDIFF z jednostką MONTH); pokaż nazwę, datę dodania i liczbę miesięcy.
4. Z9. Zamówienia złożone w weekend (DAYOFWEEK: 1 = niedziela, 7 = sobota) z nazwą dnia tygodnia.

3 Funkcje liczbowe

```
SELECT cena, cena / 1.23 FROM produkty WHERE id_produktu = 1;
```

1. Z10. Dla produktu nr 1 wyświetl cenę netto (cena / 1,23) bez zaokrąglenia oraz: ROUND do 2 miejsc, TRUNCATE do 1 miejsca, CEIL, FLOOR, a także cenę brutto zaokrągloną do pełnych setek (ROUND z argumentem -2). Wyniki wpisz do tabeli w sprawozdaniu.
2. Z11. Raty: dla produktów droższych niż 2000 zł wyświetl nazwę, cenę i wysokość jednej z 10 rat zaokrągloną w górę do pełnych złotych. Którą funkcję wybrałeś i dlaczego?
3. Z12. Imię, nazwisko, pensja i pensja po podwyżce o 3,5% zaokrąglona do groszy oraz kwota podwyżki.

4 CASE i IFNULL

```
CASE  
  WHEN warunek1 THEN wynik1  
  WHEN warunek2 THEN wynik2  
  ELSE wynik_domyślny  
END AS nazwa_kolumny
```

1. Z13. Nazwa, cena i przedział cenowy produktu: poniżej 300 zł — „tani”, od 300 do 1500 zł — „średni”, powyżej — „drogi”; sortowanie według ceny.
2. Z14. Nazwa, stan i dostępność: 0 — „brak”, 1–5 — „kończy się”, więcej — „dostępny”.
3. Z15. Imię, nazwisko i miasto klienta, a gdy miasta brak — tekst „nie podano” (IFNULL).
4. Z16. Numer zamówienia, nazwa dnia tygodnia i pora dnia złożenia: przed 12:00 — „rano”, 12:00–17:59 — „po południu”, od 18:00 — „wieczorem” (CASE z funkcją HOUR).
5. Zapisz plik `zad18_nazwisko.sql`. Zrób zrzuty ekranu Z5, Z8, Z10, Z13 i Z16.

W sprawozdaniu umieść

- ☐ tabela Z1–Z16 z treścią zapytań i wynikami
- ☐ tabela porównania funkcji zaokrąglających (Z10)
- ☐ zrzuty ekranu Z5, Z8, Z10, Z13 i Z16
- ☐ uzasadnienie wyboru funkcji w Z11 i uwaga o formie miesiąca w Z5
- ☐ odpowiedzi na pytania kontrolne

Pytania kontrolne

1. Czym różni się ROUND od TRUNCATE? Podaj przykład, w którym dają różne wyniki.
2. Dlaczego warunek `WHERE YEAR(data_zamowienia) = 2026` może działać wolniej na dużej tabeli niż `WHERE data_zamowienia >= '2026-01-01' AND data_zamowienia < '2027-01-01' ?`
3. Co zwróci wyrażenie CASE, gdy żaden warunek WHEN nie jest spełniony, a nie ma części ELSE?

4. Dlaczego wyniki Z6b będą inne jutro, a Z6a — nie?

Kryteria oceny

Kryterium	Punkty
Z1–Z5 — części daty i formatowanie	4
Z6–Z9 — obliczenia na datach	4
Z10–Z12 — funkcje liczbowe i tabela porównania	4
Z13–Z16 — CASE i IFNULL	4
Zrzuty, plik .sql i odpowiedzi na pytania	4
Razem	20

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

Funkcje agregujące

Czas: 2 × 45 min

praca indywidualna

Cel: Obliczanie podsumowań danych funkcjami COUNT, SUM, AVG, MIN i MAX oraz zrozumienie, jak agregaty traktują wartości NULL i puste zbiory.

Po zajęciach potrafisz:

- rozróżnia COUNT(*), COUNT(kolumna) i COUNT(DISTINCT kolumna)
- oblicza sumy, średnie, wartości najmniejsze i największe, także z warunkiem WHERE
- wyjaśnia, dlaczego agregaty pomijają NULL i co zwraca SUM dla pustego zbioru
- rozpoznaje błąd łączenia kolumny zwykłej z agregatem bez GROUP BY
- sprawdza wynik agregatu ręcznie na małym zbiorze danych

Sprzęt i materiały

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

Oprogramowanie

- **XAMPP** — phpMyAdmin i konsola `mysql`
apachefriends.org/download.html
- **Dokumentacja funkcji agregujących (MariaDB)**
mariadb.com/kb/en/aggregate-functions

Bezpieczeństwo

- Wykonujesz tylko zapytania SELECT na własnej bazie `sklep`; nie zmieniasz danych ani ustawień serwera na stałe (SET SESSION działa tylko do rozłączenia).

Przebieg zadania

1 Liczenie wierszy

Zapytanie z samymi funkcjami agregującymi zwraca jeden wiersz podsumowania. Nadawaj wynikom aliasy.

1. Otwórz bazę `sklep` (w razie potrzeby zaimportuj ponownie `sklep.sql`). Zapytania zapisuj w pliku `zad19_nazwisko.sql`.
2. Z1. Ile jest produktów w tabeli `produkty`?
3. Z2. W jednym zapytaniu: COUNT(*) i COUNT(id_kategorii) dla produktów. Dlaczego wyniki się różnią?
4. Z3. W jednym zapytaniu dla klientów: liczba klientów, liczba klientów z podanym miastem i liczba różnych miast (COUNT(DISTINCT ...)).
5. Z4. Ile zamówień ma status `wyslane`, a ile produktów jest niedostępnych (stan 0)? Dwa zapytania.

2 Sumy, średnie, minimum i maksimum

1. Z5. Łączna liczba sztuk towaru w magazynie i łączna wartość magazynu (suma cena × stan).
2. Z6. Średnia cena produktu — bez zaokrąglenia i zaokrąglona do 2 miejsc.
3. Z7. Najniższa i najwyższa cena produktu oraz rozpiętość cen (różnica).
4. Z8. Data pierwszego i ostatniego zamówienia oraz liczba dni między nimi (DATEDIFF na wynikach MIN i MAX).
5. Z9. Pracownicy: suma pensji (miesięczny koszt wynagrodzeń), średnia pensja i średnia pensja tylko sprzedawców.
6. Z10. Wartość zamówienia nr 5 (suma ilość × cena jednostkowa w tabeli `pozycje_zamowienia`). Sprawdź wynik ręcznie: wypisz pozycje tego zamówienia i policz na kartce.

3 NULL i puste zbiory

1. Z11. Porównaj średnią liczbę sztuk na produkt liczoną dla wszystkich produktów (AVG(stan)) ze średnią tylko dla produktów dostępnych. Drugą policz bez WHERE — wyrażeniem `AVG(CASE WHEN stan > 0 THEN stan END)`. Dlaczego CASE bez ELSE daje tu poprawny wynik?
2. Z12. Suma cen produktów droższych niż 10 000 zł — co zwraca serwer? Popraw zapytanie tak, by zamiast NULL zwracało 0 (IFNULL). Ile wynosi COUNT(*) dla tego samego warunku?

4 Pułapka: kolumna zwykła obok agregatu

Tryb ONLY_FULL_GROUP_BY jest domyślnie włączony w MySQL 8.4, a w MariaDB (XAMPP) — wyłączony. Włącz go na czas sesji:

```
SELECT @@SESSION.sql_mode;  
SET SESSION sql_mode = CONCAT(@@SESSION.sql_mode, ',ONLY_FULL_GROUP_BY');
```

1. Z13. Bez zmiany trybu wykonaj zapytanie „najdroższy produkt”: `SELECT nazwa, MAX(cena) FROM produkty;` Porównaj nazwę z wynikiem Z7 i z tabelą — czy to naprawdę najdroższy produkt?
2. Z14. Włącz ONLY_FULL_GROUP_BY (ramka powyżej) i wykonaj Z13 ponownie. Zapisz numer i treść błędu.
3. Z15. Napisz poprawne zapytanie zwracające nazwę i cenę najdroższego produktu bez funkcji agregującej (ORDER BY i LIMIT).
4. Zrób zrzuty ekranu Z2, Z3, Z11, Z13 i Z14.

W sprawozdaniu umieść

- ☐ tabela Z1–Z15 z treścią zapytań i wynikami
- ☐ ręczne obliczenie wartości zamówienia nr 5 (Z10)
- ☐ zrzuty ekranu Z2, Z3, Z11, Z13 i Z14
- ☐ wyjaśnienie różnic w Z2, Z11 i Z12 oraz błędu w Z13–Z14
- ☐ odpowiedzi na pytania kontrolne

Pytania kontrolne

1. Czym różni się COUNT(*) od COUNT(miasto) i od COUNT(DISTINCT miasto)?
2. Dlaczego AVG(stan) i SUM(stan) / COUNT(*) dają ten sam wynik, a AVG(kolumna z NULL) i SUM(kolumna) / COUNT(*) — nie?
3. Dlaczego w Z13 MariaDB zwraca wynik, który wygląda poprawnie, ale jest błędny?
4. Jaki wynik zwraca COUNT, a jaki SUM, gdy żaden wiersz nie spełnia warunku WHERE?

Kryteria oceny

Kryterium	Punkty
Z1-Z4 — liczenie wierszy i wartości	4
Z5-Z10 — SUM, AVG, MIN, MAX, ręczne sprawdzenie Z10	6
Z11-Z12 — NULL i pusty zbiór, wyjaśnienia	3
Z13-Z15 — pułapka kolumny nieagregowanej i poprawka	3
Zrzuty, plik .sql i odpowiedzi na pytania	4
Razem	20

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

GROUP BY i HAVING

Czas: 2 × 45 min

praca indywidualna

Cel: Tworzenie raportów z podsumowaniami w grupach (GROUP BY), filtrowanie grup (HAVING) i rozumienie kolejności wykonywania części zapytania.

Po zajęciach potrafisz:

- grupuje wiersze według jednej i kilku kolumn oraz według wyrażenia
- rozróżnia filtrowanie wierszy (WHERE) od filtrowania grup (HAVING)
- zna logiczną kolejność wykonywania FROM → WHERE → GROUP BY → HAVING → SELECT → ORDER BY → LIMIT
- dodaje wiersze sum częściowych i sumy ogólnej (WITH ROLLUP)
- pisze zapytania zgodne z trybem ONLY_FULL_GROUP_BY

Sprzęt i materiały

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

Oprogramowanie

- **XAMPP** — phpMyAdmin i konsola `mysql` apachefriends.org/download.html
- **Dokumentacja GROUP BY (MariaDB)** — WITH ROLLUP: <https://mariadb.com/kb/en/select-with-rollup/> mariadb.com/kb/en/group-by

Przygotowanie przed zajęciami

- Powtórz z zadania 19 różnicę między COUNT(*) i COUNT(kolumna) oraz błąd 1140.

Bezpieczeństwo

- Wykonujesz tylko zapytania SELECT na własnej bazie `sklep`; nie zmieniasz danych.

Przebieg zadania

1 Przygotowanie

Pracuj w trybie zgodnym z MySQL 8.4 — wtedy serwer od razu pokaże błędne grupowanie. W phpMyAdmin wpisuj tę linię na początku każdego okna SQL.

```
SET SESSION sql_mode = CONCAT(@@SESSION.sql_mode, ',ONLY_FULL_GROUP_BY');
```

1. Otwórz bazę `sklep` (w razie potrzeby zaimportuj ponownie `sklep.sql`). Zapytania zapisuj w pliku `zad20_nazwisko.sql`.
2. Rozpisz na kartce lub w sprawozdaniu kolejność wykonywania części zapytania SELECT (FROM, WHERE, GROUP BY, HAVING, SELECT, ORDER BY, LIMIT) — wróć do niej w Z8.

2 Grupowanie według jednej kolumny

W liście SELECT mogą być tylko kolumny z GROUP BY i funkcje agregujące.

1. Z1. Liczba produktów w każdej kategorii (według `id_kategorii`). Jak wygląda grupa produktów bez kategorii?
2. Z2. Liczba zamówień według statusu, od najliczniejszego.
3. Z3. Wartość magazynu (suma cena × stan) oraz liczba sztuk według kategorii, od największej wartości.
4. Z4. Dla każdej kategorii: najniższa, najwyższa i średnia cena (zaokrąglona do 2 miejsc).
5. Z5. Liczba klientów w każdym mieście, od najliczniejszego. Następnie spróbuj dodać do listy SELECT kolumnę `nazwisko` i zanotuj błąd.

3 HAVING a WHERE

WHERE odrzuca wiersze przed grupowaniem, HAVING — całe grupy po obliczeniu agregatów.

1. Z6. Miasta, w których mieszka co najmniej dwóch klientów.
2. Z7. Klienci (`id_klienta`), którzy złożyli co najmniej 2 zamówienia, z liczbą zamówień.
3. Z8. Kategorie, w których średnia cena produktów dostępnych w magazynie (`stan > 0`) przekracza 1000 zł. Wskaż, która część warunku musi być w WHERE, a która w HAVING, i uzasadnij to kolejnością wykonywania.
4. Z9. Wartość każdego zamówienia z tabeli `pozycje_zamowienia` (suma ilość × cena jednostkowa) — trzy zamówienia o największej wartości.
5. Z10. Zamówienia o wartości powyżej 3000 zł z liczbą pozycji i liczbą sztuk.

4 Kilka kolumn i wyrażenia

1. Z11. Liczba zamówień w każdym miesiącu (grupowanie według YEAR i MONTH daty zamówienia), chronologicznie.
2. Z12. Pracownicy: dla każdego przełożonego (`id_przełożonego`) liczba podwładnych i suma ich pensji. Co oznacza wiersz z NULL?
3. Z13. Liczba produktów w przedziałach cenowych z zadania 18 (tani / średni / drogi) — grupowanie według wyrażenia CASE (lub jego aliasu).

5 WITH ROLLUP

W MariaDB nie można łączyć WITH ROLLUP z ORDER BY w tym samym zapytaniu — sortowanie wyniku z grupowania.

1. Z14. Liczba zamówień według statusu z wierszem sumy (WITH ROLLUP). Zamień NULL w wierszu sumy na tekst „RAZEM” (IFNULL) i nadaj kolumnie alias inny niż `status` (np. `status_zam`) — inaczej serwer zgłosi niejednoznaczność.
2. Z15. Liczba zamówień według roku i miesiąca z sumami częściowymi (WITH ROLLUP). Opisz, co oznaczają wiersze z NULL w kolumnie miesiąca i w obu kolumnach.
3. Zrób zrzuty ekranu Z1, Z5 (błąd), Z8, Z13 i Z15.

W sprawozdaniu umieść

- ☐ kolejność wykonywania części zapytania SELECT
- ☐ tabela Z1–Z15 z treścią zapytań i wynikami
- ☐ zrzuty ekranu Z1, Z5 (komunikat błędu), Z8, Z13 i Z15
- ☐ uzasadnienie podziału warunków na WHERE i HAVING w Z8
- ☐ opis wierszy NULL w Z12 i Z15
- ☐ odpowiedzi na pytania kontrolne

Pytania kontrolne

1. Dlaczego warunku `COUNT(*) >= 2` nie można zapisać w WHERE?
2. Czy w HAVING można użyć aliasu z listy SELECT? Sprawdź na Z6 i porównaj z wynikiem testu aliasu w WHERE z zadania 14.
3. Czym różni się NULL w Z1 (grupa produktów bez kategorii) od NULL w wierszu podsumowania WITH ROLLUP?
4. Jak w raporcie Z3 pokazać nazwę kategorii zamiast jej numeru? (Odpowiedź poznasz w zadaniu 21.)

Kryteria oceny

Kryterium	Punkty
Z1–Z5 — grupowanie według jednej kolumny, błąd przy kolumnie spoza GROUP BY	5
Z6–Z10 — HAVING, podział WHERE/HAVING w Z8	5
Z11–Z13 — grupowanie według kilku kolumn i wyrażenia	3
Z14–Z15 — WITH ROLLUP i opis wierszy sum	3
Kolejność wykonywania, zrzuty, plik .sql, odpowiedzi na pytania	4
Razem	20

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

Złączenia — INNER JOIN

Czas: 2 × 45 min

praca indywidualna

Cel: Łączenie danych z kilku tabel złączeniem wewnętrznym na podstawie kluczy obcych oraz rozpoznawanie błędnych złączeń.

Po zajęciach potrafisz:

- łączy dwie i więcej tabel przez INNER JOIN ... ON na kluczu głównym i obcym
- stosuje aliasy tabel i kwalifikuje nazwy kolumn
- zapisuje to samo złączenie przez USING i w starym stylu (warunek w WHERE)
- rozpoznaje iloczyn kartezjański i błędne złączenie po liczbie wierszy
- łączy złączenia z warunkami WHERE i sortowaniem

Sprzęt i materiały

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

Oprogramowanie

- **XAMPP** — phpMyAdmin (zakładka Projektant pokazuje relacje) i konsola `mysql`
[apachefriends.org/download.html](https://httpd.apache.org/download.html)
- **Dokumentacja JOIN (MariaDB)**
mariadb.com/kb/en/join-syntax

Przygotowanie przed zajęciami

- Narysuj lub wydrukuj schemat relacji bazy `sklep` (z zadania 5 albo z phpMyAdmin → baza sklep → Projektant) — będziesz z niego odczytywać, po których kolumnach łączyć tabele.

Bezpieczeństwo

- Wykonujesz tylko zapytania SELECT na własnej bazie `sklep`. Iloczyn kartezjański na małych tabelach jest nieszkodliwy, ale na dużych potrafi obciążyć serwer — nie uruchamiaj go na cudzych bazach.

Przebieg zadania

1 Dwie tabele

Schemat: `tabela1 AS a JOIN tabela2 AS b ON a.klucz_obcy = b.klucz_główny`. INNER JOIN zwraca tylko pary wierszy, które spełniają warunek ON.

1. Otwórz bazę `sklep` (w razie potrzeby zaimportuj ponownie `sklep.sql`). Zapytania zapisuj w pliku `zad21_nazwisko.sql`.
2. Z1. Lista zamówień: numer, data, status oraz imię i nazwisko klienta.
3. Z2. Lista produktów z nazwą kategorii (kolumny `produkt`, `kategoria`, `cena`). Ile wierszy zwraca zapytanie, a ile jest produktów? Którego produktu brakuje i dlaczego?
4. Z3. Napisz Z1 z aliasami tabel (`z`, `k`) i z kolumnami poprzedzonymi aliasami. Następnie usuń z jednej kolumny `id_klienta` alias tabeli w liście SELECT i zanotuj błąd.
5. Z4. Zapisz Z1 przy użyciu `JOIN ... USING (id_klienta)`. Kiedy można użyć USING?

2 Trzy i cztery tabele

Każda kolejna tabela to kolejne `JOIN ... ON ...`. Odczytaj ze schematu drogę między tabelami.

1. Z5. Pozycje zamówień: numer zamówienia, data zamówienia, nazwa produktu, ilość, cena jednostkowa i wartość pozycji (ilość × cena) — tabele `pozycje_zamowienia`, `zamowienia`, `produkty`.
2. Z6. Rozszerz Z5 o imię i nazwisko klienta (czwarta tabela) i pokaż tylko zamówienia o statusie `wyslane`, posortowane według daty.
3. Z7. Produkty kupione przez Annę Nowak (bez powtórzeń): nazwa produktu i nazwa kategorii.
4. Z8. Klienci z Krakowa i ich zamówienia: imię, nazwisko, numer i data zamówienia, status.

3 Inne zapisy i błędy złączeń

1. Z9. Zapisz Z1 w starym stylu: tabele po przecinku w FROM, warunek złączenia w WHERE. Porównaj liczbę wierszy z Z1.
2. Z10. Usuń z Z9 warunek złączenia (zostaw `FROM zamowienia, klienci`). Ile wierszy zwraca zapytanie? Pokaż obliczeniem, skąd ta liczba (iloczyn kartezjański).
3. Z11. Wykonaj `SELECT * FROM produkty NATURAL JOIN kategorie;`. Ile wierszy otrzymałeś? Sprawdź, po których kolumnach serwer złączył tabele (nazwy wspólne w obu tabelach) i wyjaśnij wynik.
4. Z12. Pięć pozycji zamówień o największej wartości: nazwa produktu, kategoria, numer zamówienia, wartość pozycji.
5. Z13. Zamówienia, w których kupiono monitor (kategoria „Monitory” — warunek po nazwie kategorii, nie po numerze): numer zamówienia, klient, produkt, ilość.
6. Zrób zrzuty ekranu Z2, Z6, Z10, Z11 i Z13.

W sprawozdaniu umieść

- ☐ tabela Z1–Z13 z treścią zapytań i liczbą wierszy
- ☐ zrzuty ekranu Z2, Z6, Z10, Z11 i Z13
- ☐ wyjaśnienie brakującego produktu w Z2 i błędu w Z3
- ☐ obliczenie liczby wierszy iloczynu kartezjańskiego (Z10) i wyjaśnienie wyniku NATURAL JOIN (Z11)
- ☐ odpowiedzi na pytania kontrolne

Pytania kontrolne

1. Dlaczego INNER JOIN produktów z kategoriami „gubi” produkt bez kategorii? Jakie złączenie go pokaże? (Zadanie 22.)
2. Dlaczego zapis JOIN ... ON jest czytelniejszy i bezpieczniejszy niż warunek złączenia w WHERE?
3. Ile warunków złączenia potrzeba, żeby połączyć 4 tabele? Podaj ogólną regułę.
4. Dlaczego NATURAL JOIN jest ryzykowny, gdy tabele mają kolumny o tej samej nazwie, ale innym znaczeniu?

Kryteria oceny

Kryterium	Punkty
Z1–Z4 — złączenia dwóch tabel, aliasy, USING	5
Z5–Z8 — złączenia trzech i czterech tabel z warunkami	5
Z9–Z11 — stary zapis, iloczyn kartezjański, NATURAL JOIN z wyjaśnieniem	3
Z12–Z13 — złączenia z sortowaniem i warunkiem po nazwie	3
Zrzuty, plik .sql i odpowiedzi na pytania	4
Razem	20

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

Złączenia zewnętrzne i SELF JOIN

Czas: 2 × 45 min

praca indywidualna

Cel: Stosowanie złączeń zewnętrznych do pokazywania wierszy bez pary i wyszukiwania braków danych, łączenie tabeli z samą sobą oraz emulacja pełnego złączenia zewnętrznego.

Po zajęciach potrafisz:

- dobiera INNER, LEFT lub RIGHT JOIN do pytania o dane
- wyszukuje brakujące powiązania wzorcem LEFT JOIN ... WHERE ... IS NULL
- wyjaśnia różnicę między warunkiem w ON a warunkiem w WHERE przy złączeniu zewnętrznym
- łączy tabelę z samą sobą (SELF JOIN) na przykładzie hierarchii pracowników
- emuluje FULL OUTER JOIN przez UNION dwóch złączeń

Sprzęt i materiały

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

Oprogramowanie

- **XAMPP** — phpMyAdmin i konsola `mysql`
apachefriends.org/download.html
- **Dokumentacja JOIN (MariaDB)**
mariadb.com/kb/en/join-syntax

Przygotowanie przed zajęciami

- Powtórz zadanie 21 — zwłaszcza to, dlaczego INNER JOIN produktów z kategoriami zwraca 14 wierszy.

Bezpieczeństwo

- W Z13 dodajesz do własnej bazy `sklep` jedną kategorię testową, a w Z15 ją usuwasz. Przed DELETE wykonaj SELECT z tym samym warunkiem. Jeśli coś pójdzie nie tak, zaimportuj ponownie `sklep.sql`.
- Nie wykonuj tych poleceń na bazie, z której korzysta ktoś inny.

Przebieg zadania

1 LEFT i RIGHT JOIN

LEFT JOIN zwraca wszystkie wiersze tabeli z lewej strony; gdy brak pary, kolumny tabeli z prawej mają NULL.

1. Otwórz bazę `sklep` (najlepiej świeżo zaimportowaną). Zapytania zapisuj w pliku `zad22_nazwisko.sql`.
2. Z1. Wszyscy klienci i numery ich zamówień (imię, nazwisko, numer, data). Porównaj liczbę wierszy z INNER JOIN z zadania 21 i wyjaśnij różnicę.
3. Z2. Wszystkie produkty z nazwą kategorii; dla produktu bez kategorii wyświetl „bez kategorii” (IFNULL).
4. Z3. Zapisz Z2 przy użyciu RIGHT JOIN (zamień kolejność tabel). Czy wynik jest taki sam?
5. Z4. Wszystkie kategorie z liczbą produktów (COUNT po kolumnie z tabeli `produkty`) — także te bez produktów.

2 Szukanie braków

Wzorzec: `A LEFT JOIN B ON ... WHERE B.klucz IS NULL` — wiersze z A, które nie mają pary w B.

1. Z5. Klienci, którzy nie złożyli żadnego zamówienia.
2. Z6. Produkty, które nigdy nie zostały sprzedane (nie występują w żadnej pozycji zamówienia).
3. Z7. Wszyscy klienci z liczbą zamówień i datą ostatniego zamówienia — klienci bez zamówień z liczbą 0. Dlaczego trzeba liczyć `COUNT(z.id_zamowienia)`, a nie `COUNT(*)`?
4. Z8. Wszystkie produkty z łączną sprzedaną liczbą sztuk; dla niesprzedanych 0 (`IFNULL` na `SUM`), od najlepiej sprzedającego się.

3 Pułapka: warunek w ON czy w WHERE?

1. Z9a. Wszyscy klienci i ich zamówienia o statusie `nowe` — warunek statusu w `WHERE`. Ilu klientów widać?
2. Z9b. To samo, ale warunek statusu dopisany do `ON` (`ON ... AND z.status = 'nowe'`). Porównaj liczbę wierszy z Z9a i wyjaśnij, które zapytanie odpowiada na pytanie „wszyscy klienci, a przy nich zamówienia nowe, jeśli są”.

4 SELF JOIN — pracownik i przełożony

Tabela `pracownicy` występuje w zapytaniu dwa razy, pod różnymi aliasami (np. `p` — pracownik, `s` — szef).

1. Z10. Pracownik (imię i nazwisko) i jego przełożony (imię i nazwisko) — `INNER JOIN`. Ile wierszy i kogo brakuje?
2. Z11. To samo z `LEFT JOIN`; dla osoby bez przełożonego wyświetl „—”.
3. Z12. Przełożeni z liczbą podwładnych i sumą pensji podwładnych, od największej liczby podwładnych. Wskaż różnicę pensji między każdym pracownikiem a jego przełożonym w osobnym zapytaniu (Z12b).

5 FULL OUTER JOIN przez UNION

MariaDB i MySQL nie mają `FULL OUTER JOIN`. Pełne złączenie to suma `LEFT JOIN` i `RIGHT JOIN` połączona `UNION` (`UNION` usuwa powtórzone wiersze).

1. Z13. Dodaj do tabeli `kategorie` testową kategorię „Tablety” (bez produktów). Sprawdź wynik Z4 — czy pojawiła się z liczbą 0?
2. Z14. Wyświetl wszystkie produkty i wszystkie kategorie w jednym wyniku (kolumny: produkt, kategoria): produkt bez kategorii i kategoria bez produktów też mają się pojawić. Ile wierszy ma wynik i które mają `NULL`?
3. Z15. Usuń kategorię „Tablety” (najpierw `SELECT` z tym samym warunkiem) i sprawdź, że w tabeli jest znowu 6 kategorii.
4. Zrób zrzuty ekranu Z1, Z5, Z6, Z9a, Z9b, Z11 i Z14.

W sprawozdaniu umieść

- ☐ tabela Z1–Z15 z treścią zapytań i liczbą wierszy
- ☐ zrzuty ekranu Z1, Z5, Z6, Z9a, Z9b, Z11 i Z14
- ☐ wyjaśnienie różnicy między Z9a i Z9b
- ☐ uzasadnienie `COUNT(kolumna)` w Z7
- ☐ odpowiedzi na pytania kontrolne

Pytania kontrolne

1. Jakim złączeniem i jakim warunkiem znajdziesz produkty, których nikt nie kupił? Dlaczego `INNER JOIN` się do tego nie nadaje?

2. Dlaczego warunek dotyczący prawej tabeli zapisany w WHERE „zamienia” LEFT JOIN w INNER JOIN?
3. Do czego w SELF JOIN potrzebne są dwa różne aliasy tej samej tabeli?
4. Kiedy przy emulacji FULL OUTER JOIN użyjesz UNION, a kiedy UNION ALL?

Kryteria oceny

Kryterium	Punkty
Z1–Z4 — LEFT/RIGHT JOIN, IFNULL, liczenie z zerami	4
Z5–Z8 — szukanie braków i podsumowania z zerami	5
Z9a–Z9b — porównanie warunku w ON i WHERE z wyjaśnieniem	3
Z10–Z12 — SELF JOIN	3
Z13–Z15 — FULL OUTER przez UNION, bezpieczne dodanie i usunięcie danych	2
Zrzuty, plik .sql i odpowiedzi na pytania	3
Razem	20

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

Raporty: JOIN i GROUP BY razem

Czas: 2 × 45 min

praca indywidualna lub w parach

Cel: Przygotowanie raportów sprzedażowych łączących dane z kilku tabel z grupowaniem oraz sprawdzanie ich poprawności zapytaniem kontrolnym i obliczeniem ręcznym.

Po zajęciach potrafisz:

- buduje raporty łączące JOIN, WHERE, GROUP BY, HAVING, ORDER BY i LIMIT
- rozpoznaje zwielokrotnienie wierszy po złączeniu i stosuje COUNT(DISTINCT ...)
- decyduje, które zamówienia wliczać do sprzedaży (status anulowane)
- sprawdza raport zapytaniem kontrolnym (sumy muszą się zgadzać) i obliczeniem ręcznym
- opisuje wnioski z raportu w kilku zdaniach

Sprzęt i materiały

- komputer z Windows 11 i zainstalowanym XAMPP (Apache + MariaDB/MySQL)
- baza `sklep` (plik `sklep.sql` z folderu `Pliki_do_zadan`) — świeżo zaimportowana
- kalkulator lub arkusz do obliczeń kontrolnych

Oprogramowanie

- **XAMPP** — phpMyAdmin i konsola `mysql`
apachefriends.org/download.html
- **LibreOffice Calc lub Excel** — opcjonalnie — obliczenie kontrolne
pl.libreoffice.org/pobierz

Przygotowanie przed zajęciami

- Powtórz zadania 20–22: GROUP BY/HAVING, INNER i LEFT JOIN.

Bezpieczeństwo

- Wykonujesz tylko zapytania SELECT na własnej bazie `sklep`; nie zmieniasz danych.

Przebieg zadania

1 Ustalenia i tryb pracy

Reguła raportów: **sprzedaż** to zamówienia o statusie innym niż `anulowane`. Wartość pozycji = ilość × cena jednostkowa (`cena_jedn` z chwili zakupu, a nie bieżąca cena produktu).

```
SET SESSION sql_mode = CONCAT(@@SESSION.sql_mode, ',ONLY_FULL_GROUP_BY');
```

1. Zaimportuj ponownie `sklep.sql`, żeby wyniki zgadzały się z wynikami innych osób. Zapytania zapisuj w pliku `zad23_nazwisko.sql`; w phpMyAdmin wpisuj linię z ramki na początku okna SQL.
2. Wypisz w sprawozdaniu, które tabele i po których kolumnach trzeba połączyć, żeby z pozycji zamówienia dojść do klienta i do nazwy kategorii.

2 Raporty zamówień i klientów

1. R1. Wartość każdego zamówienia: numer, data, imię i nazwisko klienta, status, liczba pozycji i wartość — wszystkie 12 zamówień, od najnowszego.
2. R2. Obliczenie ręczne: wybierz dwa zamówienia z R1 (jedno z kilkoma pozycjami), wypisz ich pozycje zapytaniem i policz wartość na kartce lub w arkuszu. Wpisz obliczenie do sprawozdania.
3. R3. Raport statusów: dla każdego statusu liczba zamówień i ich łączna wartość. Zwróć uwagę, czym różni się COUNT(*) od COUNT(DISTINCT z.id_zamowienia) po złączeniu z pozycjami — sprawdź obie wersje.
4. R4. Najlepszy klient: imię, nazwisko, liczba zamówień i łączna wartość sprzedaży (bez anulowanych) — jeden klient o największej wartości.
5. R5. Klienci, którzy kupili za ponad 5000 zł (bez anulowanych), od największej kwoty (HAVING).

3 Raporty produktów i czasu

1. R6. Sprzedaż według kategorii (bez anulowanych): nazwa kategorii, liczba sprzedanych sztuk i przychód, od największego przychodu. Produkt bez kategorii pokaż jako „bez kategorii” — wybierz właściwe złączenie.
2. R7. Trzy produkty o największym przychodzie (bez anulowanych): nazwa, sprzedane sztuki, przychód.
3. R8. Raport miesięczny sprzedaży (bez anulowanych): rok-miesiąc w postaci 2026-02 (DATE_FORMAT), liczba zamówień, liczba sztuk, przychód; chronologicznie.
4. R9. Średnia wartość zamówienia (bez anulowanych) — oblicz jako łączny przychód podzielony przez liczbę różnych zamówień, zaokrąglij do groszy.
5. R10. Do skompletowania: produkty z zamówień jeszcze niewysłanych (statusy nowe i opłacone) — nazwa, łączna zamówiona ilość, stan magazynowy i informacja (CASE): „wystarczy” albo „brakuje N szt.”.

4 Kontrola i wnioski

Raport jest wiarygodny dopiero, gdy jego sumy zgadzają się z danymi źródłowymi.

1. K1. Policz łączny przychód z wszystkich pozycji zamówień nieanulowanych jednym prostym zapytaniem (JOIN pozycji z zamówieniami, bez grupowania).
2. K2. Porównaj K1 z sumą kolumny przychodu w R6 i w R8 (zsumuj wyniki ręcznie lub zapytaniem). Wszystkie trzy liczby muszą być równe — jeśli nie są, znajdź błąd (np. INNER JOIN gubiący produkt bez kategorii).
3. Napisz 3–4 zdania wniosków dla właściciela sklepu (najlepsza kategoria, najlepszy miesiąc, czego brakuje w magazynie).
4. Zrób zrzuty ekranu R1, R3 (obie wersje), R6, R8, R10 i K1.

W sprawozdaniu umieść

- ☐ droga złączeń od pozycji zamówienia do klienta i kategorii
- ☐ tabela R1–R10 i K1–K2 z treścią zapytań i kluczowymi wynikami
- ☐ obliczenie ręczne dwóch zamówień (R2)
- ☐ zrzuty ekranu R1, R3 (obie wersje), R6, R8, R10 i K1
- ☐ porównanie sum kontrolnych (K2) i wnioski dla właściciela sklepu
- ☐ odpowiedzi na pytania kontrolne

Pytania kontrolne

1. Dlaczego po złączeniu zamówień z pozycjami COUNT(*) nie podaje liczby zamówień?

2. Dlaczego w raportach sprzedaży używa się `cena_jedn` z pozycji, a nie bieżącej ceny z tabeli `produkty`?
3. Co by się stało z sumą w R6, gdyby użyć INNER JOIN z tabelą `kategorie`? Jak wykazałoby to zapytanie kontrolne?
4. Gdzie umieścisz warunek „bez anulowanych”: w WHERE czy w HAVING? Uzasadnij.

Kryteria oceny

Kryterium	Punkty
R1–R5 — raporty zamówień i klientów, obliczenie ręczne R2	6
R6–R10 — raporty produktów i czasu	6
K1–K2 — zgodne sumy kontrolne i wnioski	3
Zrzuty, plik .sql, czytelność raportów (aliasy, zaokrąglenia, sortowanie)	2
Odpowiedzi na pytania kontrolne	3
Razem	20

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

Podzapytania w WHERE i SELECT

Czas: 2 × 45 min

praca indywidualna lub w parach

Cel: Nauczenie się zapisywania zapytań z podzapytaniem skalarnym, listowym (IN / NOT IN) i podzapytaniem w liście SELECT oraz porównania ich z rozwiązaniem przez złączenie.

Po zajęciach potrafisz:

- zapisuje podzapytanie skalarne w warunku WHERE (np. porównanie ze średnią lub maksimum)
- stosuje IN i NOT IN z podzapytaniem i rozpoznaje pułapkę wartości NULL
- umieszcza podzapytanie w liście SELECT, aby wyliczyć wartość porównawczą
- rozpoznaje błąd 1242 i dobiera właściwy operator
- przepisuje zapytanie z podzapytaniem na złączenie i porównuje wyniki

Sprzęt i materiały

- komputer z XAMPP (MariaDB) lub MySQL 8.4
- baza `sklep` zaimportowana z pliku `sklep.sql` (folder Pliki_do_zadan)

Oprogramowanie

- **XAMPP (Apache + MariaDB + phpMyAdmin)** — phpMyAdmin: <http://localhost/phpmyadmin>, zakładka SQL apachefriends.org
- **Konsola mysql** — w XAMPP: przycisk Shell w panelu, potem `mysql -u root sklep`
- **Edytor tekstu (VS Code lub Notatnik)** — plik `zad24.sql` z zapytaniami code.visualstudio.com

Przygotowanie przed zajęciami

- Przypomnij sobie funkcje agregujące (zadanie 19) i złączenia (zadania 21–22).
- Jeśli baza `sklep` była modyfikowana na wcześniejszych zajęciach, zaimportuj ją ponownie z pliku `sklep.sql` — wyniki muszą się zgadzać z danymi wzorcowymi.

Bezpieczeństwo

- W tym zadaniu wykonujesz tylko zapytania SELECT — nie modyfikuj danych w bazie `sklep`.
- Pracuj wyłącznie na swoim stanowisku i swojej bazie; nie usuwaj baz innych osób.

Przebieg zadania

1 Podzapytanie skalarne w WHERE

Podzapytanie skalarne zwraca dokładnie jedną wartość (jeden wiersz, jedna kolumna). Można go użyć wszędzie tam, gdzie pasowałaby stała. Każde zapytanie zapisuj w pliku `zad24.sql` z komentarzem `-- Z1` itd.

```
-- schemat: zapytanie zewnętrzne korzysta z wyniku zapytania wewnętrznego
SELECT kolumny
FROM tabela
WHERE kolumna > (SELECT AGREGAT(kolumna) FROM tabela);
```

1. Z1. Napisz zapytanie, które wyświetli średnią cenę produktów zaokrągloną do 2 miejsc (bez podzapytania). Zapamiętaj wynik — przyda się do sprawdzenia Z2.
2. Z2. Wyświetl nazwę i cenę produktów droższych od średniej ceny wszystkich produktów. Średnią oblicz w podzapytaniu. Posortuj malejąco po cenie.
3. Z3. Wyświetl nazwę i cenę najdroższego produktu, używając podzapytania z MAX (bez ORDER BY i LIMIT). Dlaczego takie zapytanie jest lepsze od `ORDER BY cena DESC LIMIT 1`, gdy dwa produkty mają tę samą najwyższą cenę?
4. Z4. Wyświetl nazwy i ceny produktów z kategorii „Akcesoria”, nie wpisując ręcznie numeru kategorii — numer pobierz podzapytaniem z tabeli `kategorie`.

2 Podzapytanie zwracające listę: IN i NOT IN

Gdy podzapytanie zwraca wiele wierszy, operator `=` nie wystarczy.

1. Z5. Napisz zapytanie jak w Z4, ale dla kategorii „Laptopy” i „Monitory” (`WHERE nazwa IN ('Laptopy', 'Monitory')` w podzapytaniu), zostawiając w zapytaniu zewnętrznym operator `=`. Zapisz numer i treść błędu, a następnie popraw zapytanie, zamieniając `=` na `IN`.
2. Z6. Wyświetl imię, nazwisko i miasto klientów, którzy złożyli co najmniej jedno zamówienie (`IN` z podzapytaniem na tabeli `zamowienia`).
3. Z7. Wyświetl imię i nazwisko klientów, którzy nie złożyli żadnego zamówienia (`NOT IN`).
4. Z8. Wyświetl nazwy produktów, które nigdy nie zostały zamówione (`NOT IN` z podzapytaniem na tabeli `pozycje_zamowienia`).
5. Z9. Wyświetl nazwy produktów zamówionych przez klientów z Krakowa. Użyj dwóch zagnieżdżonych podzapytań: produkty → pozycje zamówień → zamówienia → klienci.

3 Pułapka: NOT IN i NULL

Jeśli podzapytanie po `NOT IN` zwróci choć jeden NULL, warunek dla każdego wiersza daje NULL (nieznane) i wynik jest pusty.

1. Z10. Wyświetl imię, nazwisko i stanowisko pracowników, którzy nie są niczymi przełożonymi: `WHERE id_pracownika NOT IN (SELECT id_przełożonego FROM pracownicy)`. Zapisz liczbę wierszy. Uruchom samo podzapytanie i znajdź przyczynę. Popraw zapytanie, dopisując w podzapytaniu warunek `IS NOT NULL`, i zapisz nową liczbę wierszy.
2. Z11. Wyświetl nazwy produktów, które nie należą do kategorii „Laptopy” ani „Monitory”: `WHERE id_kategorii NOT IN (SELECT id_kategorii FROM kategorie WHERE nazwa IN ('Laptopy', 'Monitory'))`. Policz produkty w wyniku i porównaj z liczbą wszystkich produktów minus laptopy i monitory. Którego produktu brakuje i dlaczego? Dopisz warunek, który go uwzględni.

4 Podzapytanie w liście SELECT

Podzapytanie skalarne może tworzyć dodatkową kolumnę wyniku.

1. Z12. Dla każdego produktu wyświetl nazwę, cenę, średnią cenę wszystkich produktów (kolumna `srednia`, 2 miejsca po przecinku) i różnicę ceny od średniej (kolumna `roznica`). Posortuj po różnicy malejąco.
2. Z13. Dla każdego pracownika wyświetl imię, nazwisko, pensję i jej udział procentowy w sumie wszystkich pensji (kolumna `udzial_proc`, 1 miejsce po przecinku).

5 Podzapytanie z grupowaniem i porównanie z JOIN

1. Z14. Wyświetl numer, datę i status zamówień, których łączna wartość (suma `ilosc` × `cena_jedn`) przekracza 3000 zł. Numery takich zamówień wyznacz w podzapytaniu z GROUP BY i HAVING.
2. Z15. Przepisz zapytanie Z6 na złączenie INNER JOIN klientów z zamówieniami. Porównaj liczbę wierszy z wynikiem Z6, a potem dodaj DISTINCT. Opisz w sprawozdaniu, skąd bierze się różnica i kiedy wolisz podzapytanie, a kiedy JOIN.
3. Zapisz plik `zad24.sql` ze wszystkimi zapytaniem i wykonaj zrzuty ekranu wyników Z2, Z5 (błąd i poprawka), Z10 (przed i po poprawce), Z12 i Z14.

W sprawozdaniu umieść

- ☐ tabela zapytań Z1–Z15 z treścią SQL i liczbą wierszy wyniku
- ☐ zrzuty ekranu wyników Z2, Z5 (błąd i poprawka), Z10 (przed i po), Z12, Z14
- ☐ wyjaśnienie pułapki NOT IN z NULL na podstawie Z10 i Z11
- ☐ porównanie podzapytania i JOIN z Z15
- ☐ plik `zad24.sql` dołączony do sprawozdania
- ☐ odpowiedzi na pytania kontrolne

Pytania kontrolne

1. Czym różni się podzapytanie skalarne od podzapytania zwracającego listę? Podaj przykład operatora dla każdego z nich.
2. Co oznacza błąd 1242 i na jakie dwa sposoby można go usunąć?
3. Dlaczego warunek `x NOT IN (1, 2, NULL)` nigdy nie jest prawdziwy?
4. W jakiej kolejności serwer logicznie wykonuje zapytanie z nieskorelowanym podzapytaniem w WHERE?
5. Kiedy złączenie (JOIN) zwraca duplikaty, których nie zwraca podzapytanie z IN?

Kryteria oceny

Kryterium	Punkty
Zapytania Z1–Z9 poprawne (wynik zgodny z danymi)	6
Z10–Z11: wykrycie i poprawienie pułapki NULL z wyjaśnieniem	4
Z12–Z14: podzapytanie w SELECT i z grupowaniem	4
Z15: porównanie z JOIN i wnioski	2
Dokumentacja: plik .sql, zrzuty, tabela zapytań	2
Odpowiedzi na pytania kontrolne	2
Razem	20

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

Podzapytania skorelowane, EXISTS i tabele pochodne

Czas: 2 × 45 min

praca indywidualna lub w parach

Cel: Nauczenie się pisania podzapytań skorelowanych, warunków EXISTS / NOT EXISTS i tabel pochodnych w klauzuli FROM oraz oceny, które rozwiązanie jest czytelniejsze i wydajniejsze.

Po zajęciach potrafisz:

- stosuje EXISTS i NOT EXISTS do sprawdzania istnienia powiązanych wierszy
- pisze podzapytanie skorelowane odwołujące się do aliasu tabeli zewnętrznej
- buduje tabelę pochodną w FROM i nadaje jej alias
- wybiera najdroższy element w każdej grupie dwiema metodami
- odczytuje kolumnę select_type w wyniku EXPLAIN i porównuje warianty zapytania

Sprzęt i materiały

- komputer z XAMPP (MariaDB) lub MySQL 8.4
- baza `sklep` zaimportowana z pliku `sklep.sql` (folder Pliki_do_zadan)

Oprogramowanie

- **XAMPP (Apache + MariaDB + phpMyAdmin)** — phpMyAdmin: <http://localhost/phpmyadmin>, zakładka SQL apachefriends.org
- **Konsola mysql** — `mysql -u root sklep` (Shell w panelu XAMPP)
- **Edytor tekstu (VS Code lub Notatnik)** — plik `zad25.sql` code.visualstudio.com

Przygotowanie przed zajęciami

- Miej pod ręką swoje rozwiązania z zadania 24 (Z7, Z8 i Z10) — porównasz je z wersjami z NOT EXISTS.
- Zaimportuj ponownie `sklep.sql`, jeśli dane były zmieniane.

Bezpieczeństwo

- Wykonujesz tylko zapytania SELECT i EXPLAIN — nie zmieniasz danych w bazie `sklep`.
- Nie usuwaj i nie nadpisuj baz innych osób na wspólnym komputerze.

Przebieg zadania

1 EXISTS i NOT EXISTS

EXISTS jest prawdziwe, gdy podzapytanie zwróci choć jeden wiersz; to, co jest na liście SELECT podzapytania, nie ma znaczenia (zwykle pisze się `SELECT 1`). Podzapytanie łączy się z zapytaniem zewnętrznym przez alias.

```
SELECT k.imie, k.nazwisko
FROM klienci AS k
WHERE EXISTS (SELECT 1 FROM zamowienia AS z
              WHERE z.id_klienta = k.id_klienta);
```

1. Z1. Uruchom zapytanie z ramki i policz wiersze. Opisz własnymi słowami, co sprawdza podzapytanie dla każdego klienta.
2. Z2. Wyświetl imię i nazwisko klientów bez żadnego zamówienia, używając NOT EXISTS. Porównaj wynik z Z7 z zadania 24.
3. Z3. Wyświetl nazwy produktów, które nigdy nie zostały zamówione (NOT EXISTS na tabeli `pozycje_zamowienia`).
4. Z4. Wyświetl pracowników, którzy nie są niczymi przełożonymi, używając NOT EXISTS. Porównaj liczbę wierszy z pierwszą (błędą) wersją Z10 z zadania 24 i wyjaśnij, dlaczego NOT EXISTS nie ma problemu z NULL.
5. Z5. Wyświetl klientów, którzy mają co najmniej jedno zamówienie o statusie „nowe” (EXISTS z dodatkowym warunkiem w podzapytaniu).

2 Podzapytania skorelowane

Podzapytanie skorelowane wykonuje się logicznie osobno dla każdego wiersza zapytania zewnętrznego, bo korzysta z jego kolumn.

1. Z6. Wyświetl nazwę, cenę i numer kategorii produktów droższych od średniej ceny w ich własnej kategorii.
2. Z7. Wyświetl najdroższy produkt w każdej kategorii (nazwa kategorii, nazwa produktu, cena) — podzapytanie skorelowane z MAX. Posortuj po nazwie kategorii.
3. Z8. Dla każdego klienta wyświetl imię, nazwisko i liczbę jego zamówień (kolumna `liczba_zamowien`) obliczoną podzapytaniem skorelowanym w liście SELECT. Posortuj malejąco po liczbie zamówień.
4. Z9. Dla każdego produktu wyświetl nazwę i łączną sprzedaną ilość (kolumna `sprzedano`). Produkty, których nie sprzedano, mają mieć 0, a nie NULL (użyj IFNULL lub COALESCE).
5. Z10. Wyświetl imię, nazwisko i pensję pracowników, którzy zarabiają więcej niż średnia pensja osób mających tego samego przełożonego co oni.

3 Tabele pochodne w FROM

Wynik podzapytania w FROM traktuje się jak tymczasową tabelę. Musi mieć alias — sprawdź, co się stanie bez niego.

```
SELECT AVG(w.wartosc)
FROM (SELECT id_zamowienia, SUM(ilosc * cena_jedn) AS wartosc
      FROM pozycje_zamowienia
      GROUP BY id_zamowienia) AS w;
```

1. Z11. Uruchom zapytanie z ramki, a potem to samo bez `AS w`. Zapisz numer i treść błędu. Uzupełnij zapytanie tak, aby zwracało średnią wartość zamówienia zaokrągloną do 2 miejsc (kolumna `srednia_wartosc`).
2. Z12. Wyświetl numer i wartość zamówień, których wartość jest większa od średniej wartości zamówienia. Użyj tabeli pochodnej z wartościami zamówień.
3. Z13. Zbuduj zestawienie: ilu klientów złożyło 0, 1, 2, 3... zamówień (kolumny `liczba_zamowien`, `liczba_klientow`). Wewnętrzne zapytanie: klienci LEFT JOIN zamówienia z GROUP BY; zewnętrzne: grupowanie po liczbie zamówień.
4. Z14. Wyświetl klienta (imię, nazwisko) o największej łącznej wartości zamówień, pomijając zamówienia anulowane. Użyj tabeli pochodnej i porównania z MAX, tak aby przy remisie pokazali się wszyscy.

4 Porównanie wariantów

To samo pytanie można zadać na kilka sposobów. Porównaj czytelność i plan wykonania.

1. Z15. Przepisz Z7 na złączenie produktów z tabelą pochodną (`SELECT id_kategorii, MAX(cena) AS maks FROM produkty GROUP BY id_kategorii`). Sprawdź, czy wynik jest taki sam.
2. Poprzedź Z7 i Z15 słowem EXPLAIN i zapisz w tabeli sprawozdania wartości kolumny `select_type` dla każdego wiersza planu (np. PRIMARY, DEPENDENT SUBQUERY, DERIVED). Wersje serwera mogą przepisać zapytanie inaczej — zapisz to, co pokazuje Twój serwer, i jego wersję (`SELECT VERSION();`).
3. Dla chętnych (Z16): wyświetl drugi najdroższy produkt w każdej kategorii — podzapytanie skorelowane liczące, ile produktów tej samej kategorii jest droższych.
4. Zapisz plik `zad25.sql` i wykonaj zrzuty ekranu wyników Z4, Z7, Z11 (błąd i poprawka), Z13 oraz obu planów EXPLAIN.

W sprawozdaniu umieść

- ☐ tabela zapytań Z1–Z15 (i Z16, jeśli wykonane) z treścią SQL i liczbą wierszy
- ☐ zrzuty ekranu wyników Z4, Z7, Z11 (błąd i poprawka), Z13 i planów EXPLAIN
- ☐ porównanie NOT IN i NOT EXISTS na przykładzie Z4
- ☐ tabela `select_type` dla Z7 i Z15 z wersją serwera oraz wnioski o czytelności i wydajności
- ☐ plik `zad25.sql`
- ☐ odpowiedzi na pytania kontrolne

Pytania kontrolne

1. Czym różni się podzapytanie skorelowane od nieskorelowanego? Po czym je rozpoznać w treści zapytania?
2. Dlaczego NOT EXISTS daje poprawny wynik tam, gdzie NOT IN z kolumną zawierającą NULL zwraca pusty zbiór?
3. Po co tabeli pochodnej alias i jaki błąd pojawia się bez niego?

4. Podaj dwa sposoby znalezienia najdroższego produktu w każdej kategorii. Który wybrałbyś przy bardzo dużej tabeli i dlaczego?
5. Dlaczego w podzapytaniu po EXISTS zwykle pisze się `SELECT 1` ?

Kryteria oceny

Kryterium	Punkty
Z1–Z5: poprawne EXISTS / NOT EXISTS i porównanie z NOT IN	5
Z6–Z10: poprawne podzapytania skorelowane	5
Z11–Z14: tabele pochodne	4
Z15 i EXPLAIN: porównanie wariantów z wnioskami	2
Dokumentacja: plik .sql, zrzuty, tabela zapytań	2
Odpowiedzi na pytania kontrolne	2
Razem	20

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

Operacje na zbiorach i raport końcowy z SELECT

Czas: 2 × 45 min

praca indywidualna lub w parach

Cel: Nauczenie się łączenia wyników zapytań operatorami zbiorowymi oraz przygotowanie raportu podsumowującego, który wykorzystuje filtrowanie, złączenia, grupowanie i podzapytania.

Po zajęciach potrafisz:

- łączy wyniki zapytań operatorami UNION i UNION ALL i wyjaśnia różnicę
- stosuje INTERSECT i EXCEPT oraz zapisuje ich odpowiedniki przez IN, NOT IN lub EXISTS
- sortuje i ogranicza wynik całego zbioru oraz pojedynczych zapytań składowych
- buduje raport z kilku zapytań, łącząc JOIN, GROUP BY i podzapytania
- sprawdza wiarygodność wyniku raportu drugim zapytaniem kontrolnym

Sprzęt i materiały

- komputer 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)** — phpMyAdmin: <http://localhost/phpmyadmin>, zakładka SQL apachefriends.org
- **Konsola mysql** — `mysql -u root sklep;`
`SELECT VERSION();` pokazuje wersję serwera
- **Edytor tekstu (VS Code lub Notatnik)** — plik `zad26.sql`
code.visualstudio.com

Przygotowanie przed zajęciami

- Sprawdź wersję swojego serwera (`SELECT VERSION();`). INTERSECT i EXCEPT działają w MariaDB od 10.3 i w MySQL od 8.0.31 — w XAMPP (MariaDB 10.4) i MySQL 8.4 są dostępne.
- Przejrzyj swoje rozwiązania z zadań 23–25 — w raporcie możesz wykorzystać gotowe zapytania.

Bezpieczeństwo

- Wykonujesz tylko zapytania SELECT — nie modyfikuj danych w bazie `sklep`.
- Nie usuwaj baz innych osób na wspólnym komputerze.

Przebieg zadania

1 UNION i UNION ALL

Zapytania składowe muszą mieć tę samą liczbę kolumn; nazwy kolumn wyniku pochodzą z pierwszego zapytania. UNION usuwa duplikaty, UNION ALL nie.

1. Z1. Utwórz listę kontaktów: imię, nazwisko i kolumna `typ` z wartością 'klient' (z tabeli `klienci`) albo 'pracownik' (z tabeli `pracownicy`). Posortuj całość po nazwisku.
2. Z2. Wyświetl miasta klientów zarejestrowanych przed 1 stycznia 2026 r. połączone z miastami klientów zarejestrowanych od tej daty — raz z UNION, raz z UNION ALL. Zapisz liczbę wierszy w obu przypadkach. Jak UNION potraktował wartości NULL?
3. Z3. Spróbuj połączyć `SELECT imie, nazwisko FROM klienci` z `SELECT imie FROM pracownicy`. Zapisz numer i treść błędu, a potem popraw zapytanie.
4. Z4. W jednym wyniku pokaż 3 najdroższe i 3 najtańsze produkty (nazwa, cena, kolumna `grupa` z wartością 'najdroższe' lub 'najtańsze'). Każde zapytanie składowe ujmij w nawiasy z własnym ORDER BY i LIMIT.

2 INTERSECT i EXCEPT

INTERSECT zwraca wiersze wspólne obu zapytaniom, EXCEPT — wiersze pierwszego zapytania, których nie ma w drugim. Na starszym serwerze (MySQL < 8.0.31) użyj wersji z IN / NOT IN / EXISTS.

1. Z5. Wyświetl miasta, z których pochodzą zarówno klienci zarejestrowani przed 2026 r., jak i klienci zarejestrowani w 2026 r. (INTERSECT).
2. Z6. Wyświetl miasta klientów zarejestrowanych w 2026 r., z których nie było nikogo przed 2026 r. (EXCEPT). Sprawdź, czy w wyniku jest NULL.
3. Z7. Wyświetl nazwy produktów zamówionych zarówno w pierwszym półroczu 2026 r. (do 30 czerwca), jak i od 1 lipca 2026 r. — najpierw z INTERSECT, potem bez niego (IN z podzapytaniem). Porównaj wyniki.

3 Raport końcowy — zestawienie dla kierownika sklepu

Kierownik prosi o raport w ośmiu punktach. Każde zapytanie zapisz w pliku `zad26.sql` z komentarzem. Do co najmniej dwóch punktów dopisz zapytanie kontrolne, które innym sposobem potwierdza wynik (np. suma wartości kategorii = suma wszystkich pozycji).

1. Z8. Podsumowanie bazy: jedna tabela z kolumnami `obiekt` i `liczba` — liczba produktów, klientów, zamówień i pracowników (cztery zapytania połączone UNION ALL).
2. Z9. Liczba zamówień w każdym statusie oraz wiersz końcowy 'RAZEM' z liczbą wszystkich zamówień (GROUP BY i UNION ALL).
3. Z10. Wartość sprzedaży w każdej kategorii (bez zamówień anulowanych), z produktami bez kategorii pokazanymi jako 'bez kategorii'. Posortuj malejąco po wartości.
4. Z11. Sprzedaż w poszczególnych miesiącach 2026 r. (bez anulowanych): miesiąc w formacie RRRR-MM, liczba zamówień, wartość.
5. Z12. Trzech klientów o największej wartości zamówień (bez anulowanych): imię, nazwisko, miasto, wartość.
6. Z13. Alerty magazynowe: produkty ze stanem 0 z opisem 'brak' oraz produkty ze stanem od 1 do 5 z opisem 'niski stan' — dwa zapytania połączone UNION ALL, posortowane po stanie.
7. Z14. Produkty, których nikt nigdy nie zamówił, oraz klienci, którzy niczego nie zamówili — jedna lista z kolumnami `rodzaj` ('produkt' / 'klient') i `opis` (nazwa produktu lub imię i nazwisko klienta).
8. Z15. Zamówienia (numer, klient, wartość), których wartość przekracza średnią wartość zamówienia — z podzapytaniem lub tabelą pochodną.

W sprawozdaniu umieść

- ☐ tabela zapytań Z1–Z15 z treścią SQL i liczbą wierszy
- ☐ wersja serwera i informacja, czy INTERSECT/EXCEPT zadziałały
- ☐ zrzuty ekranu wyników Z2 (UNION i UNION ALL), Z3 (błąd), Z7 (obie wersje) oraz wszystkich punktów raportu Z8–Z15
- ☐ co najmniej dwa zapytania kontrolne do raportu z opisem, co potwierdzają
- ☐ plik `zad26.sql`
- ☐ odpowiedzi na pytania kontrolne

Pytania kontrolne

1. Czy różni się UNION od UNION ALL? Które jest szybsze i dlaczego?
2. Skąd biorą się nazwy kolumn w wyniku zapytania z UNION i jakie warunki muszą spełniać zapytania składowe?
3. Jak zapisać INTERSECT i EXCEPT na serwerze, który ich nie obsługuje?
4. Dlaczego w Z4 każde zapytanie składowe trzeba ująć w nawiasy?
5. Do czego służy zapytanie kontrolne w raporcie i jak je dobrać?

Kryteria oceny

Kryterium	Punkty
Z1–Z4: poprawne UNION / UNION ALL, błąd i sortowanie składowych	4
Z5–Z7: INTERSECT, EXCEPT i ich alternatywy	4
Z8–Z15: raport — poprawne wyniki	6
Zapytania kontrolne i ich opis	2
Dokumentacja: plik .sql, zrzuty, tabela zapytań	2
Odpowiedzi na pytania kontrolne	2
Razem	20

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

Użytkownicy, uprawnienia i role

Czas: 2 × 45 min

praca indywidualna

Cel: Nauczenie się tworzenia kont z minimalnymi niezbędnymi uprawnieniami, nadawania i odbierania uprawnień oraz sprawdzania ich działania przez logowanie się na ograniczone konto.

Po zajęciach potrafisz:

- tworzy konta użytkowników z określonym hostem i hasłem
- nadaje i odbiera uprawnienia na poziomie bazy, tabeli i kolumny
- sprawdza uprawnienia poleceniem `SHOW GRANTS` i testem logowania
- tworzy rolę, przypisuje ją użytkownikowi i aktywuje
- zmienia hasło konta root w XAMPP i dostosowuje konfigurację phpMyAdmin

Sprzęt i materiały

- komputer 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)** — phpMyAdmin → zakładka Konta użytkowników apachefriends.org
- **Konsola mysql** — dwa okna: jedno jako root, drugie jako testowany użytkownik
- **Edytor tekstu (VS Code lub Notatnik)** — plik `zad27.sql` oraz edycja `config.inc.php` code.visualstudio.com

Przygotowanie przed zajęciami

- Przypomnij sobie strukturę bazy `sklep` (tabele i kolumny tabeli `pracownicy`).
- Sprawdź, jak otworzyć drugie okno konsoli: w panelu XAMPP przycisk Shell (otwiera się w folderze `C:\xampp`), polecenie `mysql -u nazwa -p`.

Bezpieczeństwo

- Hasła w zadaniu są przykładowe i służą tylko do ćwiczeń — w prawdziwych systemach używaj długich, unikalnych haseł i nie zapisuj ich w sprawozdaniu ani na zrzutach ekranu.
- Krok 5 (hasło root) wykonuj tylko za zgodą nauczyciela. Zapisz nowe hasło na kartce na czas zajęć — bez niego nie zalogujesz się do serwera. Na koniec przywróć stan wyjściowy zgodnie z poleceniem nauczyciela, bo z komputera korzystają inne klasy.
- Nie nadawaj kontom uprawnień `ALL PRIVILEGES ON *.*` ani `WITH GRANT OPTION` — zasada najmniejszych uprawnień.
- Nie usuwaj kont `root`, `pma` ani kont systemowych (np. `mariadb.sys`). Usuwać tylko konta utworzone przez siebie.

Przebieg zadania

1 Konta na serwerze

Konto w MariaDB/MySQL to para nazwa@host. 'jan'@'localhost' i 'jan'@'%' to dwa różne konta.

1. Z1. Jako root wyświetl listę kont (`SELECT user, host FROM mysql.user;`), swoje bieżące konto (`SELECT CURRENT_USER();`) i swoje uprawnienia (`SHOW GRANTS;`). Zapisz, jakie konta root istnieją i z jakich hostów.
2. Z2. Utwórz konto `sprzedawca`, które może logować się tylko z tego komputera, z hasłem `Sprzed#2026`. Nadaj mu prawo odczytu wszystkich tabel bazy `sklep` (poziom bazy). Wyświetl jego uprawnienia poleceniem `SHOW GRANTS FOR`.
3. Z3. W drugim oknie zaloguj się jako `sprzedawca` (`mysql -u sprzedawca -p sklep`). Wykonaj `SELECT COUNT(*) FROM produkty;`, a potem spróbuj zmienić cenę dowolnego produktu. Zapisz numer i treść błędu. Spróbuj też zalogować się z błędnym hasłem i zapisz kod błędu.

2 Uprawnienia na poziomie tabeli i kolumny

1. Z4. Utwórz konto `magazynier@localhost` (hasło `Magaz#2026`), które może tylko odczytywać i zmieniać dane w tabeli `produkty` (`SELECT`, `UPDATE` na jednej tabeli).
2. Z5. Zaloguj się jako `magazynier` i sprawdź: `SHOW TABLES;` (ile tabel widzi?), zmianę stanu magazynowego produktu nr 7 o +5 sztuk, odczyt tabeli `klienci` i usunięcie produktu (`DELETE` z warunkiem na nieistniejący numer, np. 999). Zapisz, co się udało, a co zakończyło się błędem.
3. Z6. Utwórz konto `recepcja@localhost` (hasło `Recep#2026`), które w tabeli `pracownicy` może odczytać tylko kolumny `imie`, `nazwisko` i `stanowisko` (uprawnienie na poziomie kolumn).
4. Z7. Zaloguj się jako `recepcja` i wykonaj kolejno: `SELECT` dozwolonych kolumn, `SELECT * FROM pracownicy;` oraz `SELECT` kolumny `pensja`. Zapisz kody błędów i wyjaśnij, dlaczego `SELECT *` też jest zabroniony.
5. Z8. Rozszerz uprawnienia konta `sprzedawca` o dodawanie wierszy do tabel `zamowienia` i `pozycje_zamowienia` (dwa polecenia `GRANT` na poziomie tabeli). Sprawdź `SHOW GRANTS FOR`.

3 Odbieranie uprawnień i zmiana hasła

1. Z9. Odbierz kontu `magazynier` prawo `UPDATE` do tabeli `produkty`. Sprawdź `SHOW GRANTS FOR`, zaloguj się ponownie i spróbuj zmienić stan produktu — zapisz wynik.
2. Z10. Zmień hasło konta `sprzedawca` na `Nowe#Haslo2026` (`ALTER USER`). Sprawdź logowanie starym i nowym hasłem.

4 Role

Rola to nazwany zestaw uprawnień, który przypisuje się wielu użytkownikom. Rola przypisana użytkownikowi musi być aktywna — inaczej jej uprawnienia nie działają. Składnia roli domyślnej różni się w MariaDB i MySQL.

```
-- aktywacja roli w bieżącej sesji (MariaDB i MySQL):
SET ROLE nazwa_rol;
SELECT CURRENT_ROLE();

-- rola domyślna (aktywna od zalogowania):
SET DEFAULT ROLE nazwa_rol FOR 'uzytkownik'@'localhost'; -- MariaDB (XAMPP)
SET DEFAULT ROLE nazwa_rol TO 'uzytkownik'@'localhost'; -- MySQL 8.4
```

1. Z11. Utwórz rolę `odczyt_sklep` z prawem SELECT do całej bazy `sklep`. Utwórz konto `stazysta@localhost` (hasło `Staz#2026`) i przypisz mu tę rolę.
2. Z12. Zaloguj się jako `stazysta` (bez podawania bazy: `mysql -u stazysta -p`). Sprawdź `SELECT CURRENT_ROLE();` i spróbuj odczytać `SELECT COUNT(*) FROM sklep.klienci;`. Następnie aktywuj rolę poleceniem SET ROLE i powtórz zapytanie. Zapisz oba wyniki.
3. Z13. Jako root ustaw rolę `odczyt_sklep` jako domyślną dla konta `stazysta` (składnia właściwa dla Twojego serwera), zaloguj się ponownie i sprawdź `CURRENT_ROLE()`.

5 Hasło root w XAMPP i porządki

W XAMPP konto root nie ma hasła, a phpMyAdmin loguje się automatycznie danymi zapisanymi w pliku `C:\xampp\phpMyAdmin\config.inc.php`. Wykonaj ten krok tylko za zgodą nauczyciela.

```
/* fragment C:\xampp\phpMyAdmin\config.inc.php */
$config['Servers'][$i]['auth_type'] = 'config';
$config['Servers'][$i]['user'] = 'root';
$config['Servers'][$i]['password'] = '';
```

1. Z14. Ustaw hasło dla konta `root@localhost` (ALTER USER). Odśwież phpMyAdmin i zapisz komunikat błędu. Zrób kopię pliku `config.inc.php`, a następnie zmień w nim `auth_type` z `config` na `cookie` (phpMyAdmin zacznie pytać o login i hasło). Zaloguj się jako root i jako `sprzedawca` — porównaj, co widzi każde konto. Wyjaśnij, dlaczego `cookie` jest bezpieczniejsze niż wpisanie hasła w pliku.
2. Z15. Porządki: usuń konta `sprzedawca`, `magazynier`, `repcja`, `stazysta` i rolę `odczyt_sklep`. Sprawdź listę kont. Jeśli nauczyciel tak zdecyduje, przywróć puste hasło root i oryginalny plik `config.inc.php` z kopii.
3. Zapisz plik `zad27.sql` z poleceniami (bez haseł z kroku 5) i zrzuty ekranu: SHOW GRANTS z Z2, Z6 i Z8, błędy z Z3, Z5, Z7, wynik Z12 przed i po SET ROLE oraz okno logowania phpMyAdmin.

W sprawozdaniu umieść

- ☐ tabela poleceń Z1–Z15 z treścią SQL i wynikiem
- ☐ tabela testów uprawnień: konto, operacja, oczekiwany wynik, faktyczny wynik / kod błędu
- ☐ zrzuty ekranu SHOW GRANTS (Z2, Z6, Z8), błędów (Z3, Z5, Z7) i ról (Z12)
- ☐ opis zmian w `config.inc.php` i porównanie `auth_type` config i cookie
- ☐ plik `zad27.sql`
- ☐ odpowiedzi na pytania kontrolne

Pytania kontrolne

1. Czym różnią się konta `'jan'@'localhost'` i `'jan'@'%'`? Które jest bezpieczniejsze dla aplikacji działającej na tym samym serwerze?

2. Co oznaczają kody błędów 1044, 1045, 1142 i 1143?
3. Na czym polega zasada najmniejszych uprawnień? Podaj przykład z tego zadania.
4. Dlaczego po przypisaniu roli użytkownik nadal mógł nie mieć dostępu do danych i jak to naprawić na stałe?
5. Jakie ryzyko niesie puste hasło root w XAMPP i kiedy jest ono szczególnie groźne?

Kryteria oceny

Kryterium	Punkty
Z1–Z3: konto z uprawnieniami na poziomie bazy i test logowania	3
Z4–Z8: uprawnienia na poziomie tabeli i kolumny z testami	5
Z9–Z10: REVOKE i zmiana hasła	2
Z11–Z13: role (utworzenie, aktywacja, rola domyślna)	3
Z14–Z15: hasło root, config.inc.php, porządki	3
Dokumentacja: tabela testów, zrzuty, plik .sql	2
Odpowiedzi na pytania kontrolne	2
Razem	20

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

Kopie zapasowe i odtwarzanie bazy

Czas: 2 × 45 min

praca indywidualna lub w parach

Cel: Nauczenie się wykonywania kopii zapasowej bazy i pojedynczej tabeli narzędziem mysqldump i w phpMyAdmin, odtwarzania kopii do nowej bazy, sprawdzania jej kompletności oraz zaplanowania automatycznych kopii.

Po zajęciach potrafisz:

- wykonuje kopię bazy i pojedynczej tabeli poleceniem mysqldump z odpowiednimi opcjami
- odtwarza kopię do nowej bazy w konsoli i w phpMyAdmin
- porównuje zawartość bazy oryginalnej i odtworzonej (COUNT, CHECKSUM TABLE, lista obiektów)
- przywraca pojedynczą tabelę po błędnej modyfikacji
- przygotowuje skrypt .bat tworzący kopię z datą w nazwie pliku i plan kopii zapasowych

Sprzęt i materiały

- komputer z XAMPP (MariaDB 10.4) lub MySQL 8.4 na Windows 11
- baza `sklep` zaimportowana z pliku `sklep.sql` (folder Pliki_do_zadan)
- pendrive (opcjonalnie — kopia poza komputerem)

Oprogramowanie

- **XAMPP (MariaDB, phpMyAdmin)** — narzędzia w `C:\xampp\mysql\bin` (mysqldump.exe, mysql.exe) apachefriends.org
- **Wiersz polecenia (Shell w panelu XAMPP lub cmd)** — przekierowania `>` i `<` działają w cmd; w PowerShell `<` nie działa — używaj cmd
- **Dokumentacja mysqldump (MariaDB)** — w MySQL 8.4: <https://dev.mysql.com/doc/refman/8.4/en/mysqldump.html> mariadb.com/kb/en/mariadb-dump
- **Edytor tekstu (VS Code lub Notatnik)** — podgląd plików .sql, skrypt `kopia.bat` code.visualstudio.com

Przygotowanie przed zajęciami

- Utwórz folder `C:\kopie` na pliki kopii.
- Zaimportuj świeżą bazę `sklep` z pliku `sklep.sql`.

Bezpieczeństwo

- Kopie odtwarzają wyłącznie do nowej bazy `sklep_kopia` (lub `sklep_kopia2`). Nigdy nie wczytuj pliku do bazy, której nie chcesz nadpisać — kopia zawiera polecenia `DROP TABLE`.
- Plik utworzony z opcją `--databases` zawiera `CREATE DATABASE` i `USE sklep` — jego wczytanie nadpisze oryginalną bazę niezależnie od tego, jaką bazę podasz w poleceniu.
- Kopia bazy zawiera wszystkie dane (w prawdziwym sklepie — dane osobowe klientów). Nie zostawiaj kopii w miejscach publicznych i nie wysyłaj jej do chmury bez zgody właściciela danych.
- Nie zapisuj haseł w skrypcie `.bat`. W pracowni root nie ma hasła; na prawdziwym serwerze użyj osobnego konta do kopii i pliku opcji z ograniczonym dostępem.
- Przed zmianami w Z9 upewnij się, że jesteś połączony z bazą `sklep_kopia` (`SELECT DATABASE();`).

Przebieg zadania

1 Przygotowanie obiektów do kopii

Dodaj do swojej bazy `sklep` widok, wyzwalacz i procedurę (poznasz je dokładnie w roku 2). Wykonaj poniższy kod w konsoli `mysql -u root sklep` — w phpMyAdmin ustaw ogranicznik (Delimiter) na `//` i pomiń wiersze `DELIMITER`.

```
CREATE VIEW v_stan_magazynu AS SELECT nazwa, stan FROM produkty;
CREATE TRIGGER klienci_email_male BEFORE INSERT ON klienci
  FOR EACH ROW SET NEW.email = LOWER(NEW.email);
DELIMITER //
CREATE PROCEDURE produkty_kategorii(IN kat INT)
BEGIN
  SELECT nazwa, cena FROM produkty WHERE id_kategorii = kat;
END //
DELIMITER ;
CALL produkty_kategorii(2);
```

1. Z1. Wykonaj kod z ramki i zapisz wynik wywołania procedury. Sprawdź, że widok działa (`SELECT * FROM v_stan_magazynu;`).

2 Kopie poleceniem mysqldump

Otwórz Shell z panelu XAMPP (bieżący folder `C:\xampp`, programy `mysql` i `mysqldump` są dostępne). Zapisz każde polecenie w sprawozdaniu.

```
mysqldump -u root [opcje] nazwa_bazy [tabela ...] > C:\kopie\plik.sql
```

1. Z2. Wykonaj kopię bazy `sklep` bez dodatkowych opcji do pliku `C:\kopie\sklep_bez_procedur.sql`.
2. Z3. Wykonaj kopię bazy `sklep` z opcjami `--routines --triggers --single-transaction` do pliku `C:\kopie\sklep_pelna.sql`. Porównaj rozmiary obu plików i sprawdź poleceniem `findstr /C:"PROCEDURE" C:\kopie\*.sql`, w którym pliku jest procedura. Sprawdź tak samo słowo `TRIGGER`. Co jest domyślnie dołączane, a co nie?
3. Z4. Wykonaj kopię samej tabeli `produkty` do pliku `C:\kopie\produkty.sql` oraz kopię samej struktury bazy (bez danych, opcja `--no-data`) do pliku `C:\kopie\sklep_struktura.sql`. Otwórz pliki w edytorze i znajdź polecenia `DROP TABLE`, `CREATE TABLE` i `INSERT`.
4. Z5. Wykonaj kopię z opcją `--databases sklep` do pliku `C:\kopie\sklep_z_baza.sql` (tylko utwórz — **nie wczytuj go**). Znajdź w nim polecenia `CREATE DATABASE` i `USE` i wyjaśnij, dlaczego taki plik jest niebezpieczny przy odtwarzaniu pod inną nazwą.

3 Odtwarzanie i porównanie

1. Z6. Utwórz pustą bazę `sklep_kopia` (utf8mb4, utf8mb4_polish_ci) i wczytaj do niej plik `sklep_pelna.sql` poleceniem `mysql` z przekierowaniem `<`.
2. Z7. Napisz jedno zapytanie (UNION ALL), które dla tabel `produkty`, `klienci`, `zamowienia` i `pozycje_zamowienia` pokaże liczbę wierszy w bazie `sklep` i w bazie `sklep_kopia` obok siebie. Wykonaj też `CHECKSUM TABLE` dla tabel `produkty` i `pozycje_zamowienia` w obu bazach.
3. Z8. Sprawdź, czy w `sklep_kopia` są widoki, wyzwalacz i procedura (`SHOW FULL TABLES FROM sklep_kopia WHERE Table_type = 'VIEW';`, `SHOW TRIGGERS FROM sklep_kopia;`, `SHOW PROCEDURE STATUS WHERE Db = 'sklep_kopia';`).
4. Z9. Symulacja awarii w bazie `sklep_kopia`: przez pomyłkę zwiększ ceny wszystkich produktów dziesięciokrotnie (UPDATE bez WHERE). Zapisz sumę cen. Przywróć tylko tabelę `produkty` z pliku `produkty.sql` i sprawdź sumę cen oraz `CHECKSUM TABLE` z oryginałem. Czy pozycje zamówień przetrwały odtworzenie tabeli powiązanej kluczem obcym?
5. Z10. W phpMyAdmin wyeksportuj bazę `sklep` (Eksport → metoda Niestandardowa, format SQL; sprawdź, gdzie włącza się dołączanie procedur i wyzwalaczy). Utwórz bazę `sklep_kopia2` i zaimportuj do niej plik (zakładka Import). Sprawdź liczbę tabel i procedur.

4 Automatyzacja i plan kopii

Uzupełnij skrypt. Pętla `for /f` pobiera z PowerShell datę w formacie niezależnym od ustawień regionalnych Windows.

```
@echo off
rem kopia.bat - kopia bazy sklep z data w nazwie pliku
set BIN=C:\xampp\mysql\bin
set KAT=C:\kopie
for /f %d in ('powershell -NoProfile -Command "Get-Date -Format yyyy-MM-dd_HH:mm"'') do set DATA=%d
rem Z11: dopisz wywołanie %BIN%\mysqldump.exe zapisujące kopie do %KAT%\sklep_%DATA%.sql
rem Z11: wypisz komunikat z nazwa utworzonego pliku
```

1. Z11. Uzupełnij skrypt `kopia.bat`, uruchom go dwa razy w odstępie minuty i sprawdź, że powstały dwa pliki z różną datą w nazwie. Dla chętnych: dopisz usuwanie kopii starszych niż 7 dni poleceniem `forfiles` i opisz, jak uruchamiać skrypt codziennie w Harmonogramie zadań Windows.
2. Z12. Przygotuj plan kopii dla sklepu internetowego (tabela w sprawozdaniu): co kopiować, jak często, gdzie przechowywać (zasada 3-2-1), jak długo, kto odpowiada i jak często testować odtwarzanie.
3. Porządki: usuń bazy `sklep_kopia` i `sklep_kopia2`, skopiuj wybrany plik kopii na pendrive (jeśli nauczyciel pozwoli), a pozostałe pliki usuń z `C:\kopie`.

W sprawozdaniu umieść

- ☐ tabela poleceń Z1–Z12 z treścią polecenia i wynikiem
- ☐ porównanie plików kopii (rozmiar, czy zawiera procedurę, wyzwalacz, CREATE DATABASE)
- ☐ wynik porównania baz z Z7 i Z8 oraz przebieg odtworzenia tabeli z Z9
- ☐ treść skryptu `kopia.bat` i zrzut folderu z dwoma plikami kopii
- ☐ plan kopii zapasowych w tabeli
- ☐ odpowiedzi na pytania kontrolne

Pytania kontrolne

1. Co zawiera plik utworzony przez `mysqldump` i dlaczego można go otworzyć w Notatniku?

2. Czym różni się kopia utworzona z opcją `--databases` od kopii bez niej? Kiedy każda z nich jest wygodniejsza?
3. Dlaczego odtworzenie tabeli `produkty` nie zostało zablokowane przez klucz obcy z tabeli `pozycje_zamowienia`? Znajdź odpowiedni wiersz na początku pliku kopii.
4. Na czym polega zasada 3-2-1 i dlaczego kopia na tym samym dysku co baza nie wystarcza?
5. Dlaczego kopia, której nigdy nie odtworzono próbnie, jest niepewna?

Kryteria oceny

Kryterium	Punkty
Z2–Z5: kopie mysqldump z właściwymi opcjami i analiza plików	5
Z6–Z8: odtworzenie do nowej bazy i porównanie	4
Z9–Z10: przywrócenie tabeli po awarii, eksport/import w phpMyAdmin	3
Z11: działający skrypt .bat z datą w nazwie pliku	3
Z12: plan kopii zapasowych	2
Odpowiedzi na pytania kontrolne	3
Razem	20

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

Próba egzaminacyjna INF.03 nr 1 — część bazodanowa

Czas: 2 × 45 min

praca samodzielna

Cel: Przećwiczenie w warunkach zbliżonych do egzaminu INF.03 importu bazy z pliku, pisania czterech zapytań (wybieranie, złączenie, agregacja, modyfikacja lub uprawnienia) oraz dokumentowania wyników zrzutami ekranu i plikiem z kwerendami.

Po zajęciach potrafisz:

- importuje bazę z pliku SQL i sprawdza jej strukturę
- pisze zapytania SELECT z warunkiem, sortowaniem, złączeniem i agregacją zgodnie z treścią polecenia
- tworzy konto użytkownika i nadaje mu uprawnienia lub modyfikuje dane i strukturę tabeli
- dokumentuje wyniki w ustalonym formacie (zrzuty kw1..., plik kwerendy.txt)
- ocenia swoją pracę według kryteriów zbliżonych do egzaminacyjnych

Sprzęt i materiały

- komputer z XAMPP (MariaDB 10.4) lub MySQL 8.4
- plik `r1_29_schronisko.sql` z folderu Pliki_do_zadan
- narzędzie do zrzutów ekranu (Win + Shift + S lub Narzędzie Wycinanie)

Oprogramowanie

- **XAMPP (Apache + MariaDB + phpMyAdmin)** — phpMyAdmin: <http://localhost/phpmyadmin>
[apachefriends.org](http://www.apachefriends.org)
- **Notatnik lub VS Code** — plik `kwerendy.txt`
code.visualstudio.com
- **Centralna Komisja Egzaminacyjna — egzamin zawodowy** — zakładka Egzamin zawodowy: przykładowe zadania i zasady oceniania
cke.gov.pl

Przygotowanie przed zajęciami

- Powtórz składnię SELECT z JOIN i GROUP BY (zadania 20–23) oraz CREATE USER i GRANT (zadanie 27).
- Przećwicz robienie zrzutu fragmentu ekranu i zapisywanie go jako PNG pod wskazaną nazwą.

Bezpieczeństwo

- Plik `r1_29_schronisko.sql` usuwa i tworzy od nowa bazę `schronisko` — nie zmieniaj w nim nazwy bazy na nazwę innej istniejącej bazy.
- Konto `wolontariusz` z kw4 usuń po zakończeniu zajęć (`DROP USER`), chyba że nauczyciel poleci inaczej.
- Pracuj samodzielnie i nie korzystaj z internetu w części A — tak jak na egzaminie.

Przebieg zadania

1 Przygotowanie stanowiska

Na egzaminie folder ma nazwę Twojego numeru PESEL — tutaj użyj nazwiska i imienia.

1. Utwórz na pulpicie folder `nazwisko_imie` (bez polskich znaków). Wszystkie zrzuty i plik `kwerendy.txt` zapisuj w nim.
2. Zimportuj plik `r1_29_schronisko.sql` w phpMyAdmin (Import) — plik sam tworzy bazę `schronisko` z kodowaniem utf8mb4. Przejrzyj strukturę tabel `gatunki`, `zwierzeta`, `adoptujacy`, `adopcje` i relacje między nimi (Widok projektu / Designer).
3. Zanotuj godzinę rozpoczęcia części A. Masz na nią 45 minut.

2 Zestaw A — zadanie egzaminacyjne (45 min)

Wykonaj zapytania w bazie `schronisko`. Po każdym zapytaniu zrób zrzut ekranu wyniku tak, aby było widać treść zapytania i cały wynik, i zapisz go jako `kw1.png`, `kw2.png` ... (format PNG lub JPEG).

Wszystkie zapytania zapisz kolejno w pliku `kwerendy.txt`.

1. kw1. Wybierz imię, płeć i wagę zwierząt o statusie „do adopcji”, które urodziły się w 2023 roku lub później. Posortuj wynik malejąco według wagi.
2. kw2. Wybierz imię zwierzęcia, nazwę gatunku, imię i nazwisko osoby adoptującej oraz datę adopcji dla adopcji przeprowadzonych w 2026 roku. Posortuj wynik rosnąco według daty adopcji.
3. kw3. Dla każdego gatunku wybierz jego nazwę i liczbę zwierząt tego gatunku, które nadal przebywają w schronisku (status inny niż „adoptowany”). Kolumnę z liczbą nazwij `liczba`. Posortuj malejąco według liczby.
4. kw4. Utwórz użytkownika `wolontariusz` logującego się z serwera lokalnego (localhost) z hasłem `Schron!26`. Nadaj mu uprawnienia do wybierania i aktualizowania danych wyłącznie w tabeli `zwierzeta`. Zrzut ekranu ma pokazywać wynik wykonania obu poleceń oraz wynik `SHOW GRANTS` dla tego użytkownika.

3 Zestaw B — trening dodatkowy (25 min)

Te same zasady dokumentowania: zrzuty `kw5.png` – `kw8.png`, zapytania dopisane do `kwerendy.txt`.

1. kw5. Zmień status zwierząt „zarezerwowanych”, które przyjęto do schroniska przed 1 maja 2026 r., na „do adopcji”. Na zrzucie ma być widoczna liczba zmienionych wierszy.
2. kw6. Dodaj do tabeli `adoptujacy` kolumnę `email` typu napisowego o długości 80 znaków, dopuszczającą wartość NULL, umieszczoną po kolumnie `telefon`. Zrzut ma pokazywać strukturę tabeli po zmianie.
3. kw7. Wybierz miasto i sumę opłat adopcyjnych wpłaconych przez osoby z tego miasta (kolumna `suma_oplat`). Pokaż tylko miasta, w których suma wynosi co najmniej 200 zł.
4. kw8. Wybierz imię, nazwę gatunku i liczbę dni od przyjęcia do schroniska do 1 października 2026 r. (kolumna `dni`) dla trzech zwierząt o statusie „do adopcji”, które czekają najdłużej.

4 Samoocena

Nauczyciel udostępni wyniki wzorcowe (liczbę wierszy i kluczowe wartości).

1. Porównaj swoje wyniki z wzorcowymi i wypełnij tabelę samooceny w sprawozdaniu. Dla każdego błędnego zapytania napisz, co było złe i jak to poprawić.
2. Sprawdź format oddanej pracy: nazwa folderu, nazwy plików, kompletność `kwerendy.txt` (każde zapytanie zakończone średnikiem, w kolejności kw1-kw8).
3. Usuń konto `wolontariusz` poleceniem `DROP USER`.

W sprawozdaniu umieść

- ☐ folder `nazwisko_imie` z plikami `kw1.png` – `kw4.png` (oraz `kw5.png` – `kw8.png`) i `kwerendy.txt`

- ☐ godzina rozpoczęcia i zakończenia części A
- ☐ tabela samooceny z punktacją i opisem błędów
- ☐ odpowiedzi na pytania kontrolne

Pytania kontrolne

1. Które elementy zapytania (kolumny, warunek, sortowanie, alias) najłatwiej pominąć przy czytaniu polecenia egzaminacyjnego? Jak się przed tym zabezpieczyć?
2. Dlaczego w kw3 trzeba połączyć tabelę `zwierzeta` z tabelą `gatunki`, choć liczba zwierząt jest w jednej tabeli?
3. Czym różni się warunek w WHERE od warunku w HAVING na przykładzie kw7?
4. Jak sprawdzić, że użytkownik z kw4 ma dokładnie te uprawnienia, których wymaga polecenie?

Kryteria oceny

Kryterium	Punkty
Import bazy i poprawna organizacja folderu i nazw plików	2
kw1 — kolumny, warunek, sortowanie	2
kw2 — złączenie trzech/czterech tabel, warunek, sortowanie	3
kw3 — grupowanie, alias, warunek, sortowanie	3
kw4 — użytkownik i uprawnienia zgodne z treścią	2
kw5–kw8 — zestaw B (po 1 pkt)	4
Plik kwerendy.txt kompletny, zrzuty czytelne	2
Samoocena i odpowiedzi na pytania kontrolne	2
Razem	20

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

Próba egzaminacyjna INF.03 nr 2 — wypożyczalnia rowerów

Czas: 2 × 45 min

praca samodzielna

Cel: Sprawdzenie w warunkach egzaminacyjnych (limit 60 minut) umiejętności importu bazy, pisania zapytań wybierających, złączeń, agregacji i modyfikacji danych oraz rzetelnej samooceny według kryteriów.

Po zajęciach potrafisz:

- wykonuje część bazodanową zadania egzaminacyjnego w wyznaczonym czasie
- dobiera warunek IS NULL / IS NOT NULL do danych z wartościami nieznanymi
- łączy grupowanie z funkcjami AVG i ROUND oraz poprawnie sortuje wynik
- modyfikuje wiele kolumn jednym poleceniem UPDATE z warunkiem
- ocenia własne rozwiązania według kryteriów i planuje poprawę

Sprzęt i materiały

- komputer z XAMPP (MariaDB 10.4) lub MySQL 8.4
- plik `r1_30_wypożyczalnia.sql` z folderu Pliki_do_zadan
- narzędzie do zrzutów ekranu (Win + Shift + S lub Narzędzie Wycinanie)

Oprogramowanie

- **XAMPP (Apache + MariaDB + phpMyAdmin)** — phpMyAdmin: <http://localhost/phpmyadmin>
apachefriends.org
- **Notatnik lub VS Code** — plik `kwerendy.txt`
code.visualstudio.com

Przygotowanie przed zajęciami

- Przejrzyj swoją samoocenę z zadania 29 i zapisz na kartce dwa błędy, których chcesz dziś uniknąć.
- Powtórz warunki z NULL (zadanie 15) i UPDATE z kilkoma kolumnami (zadanie 12).

Bezpieczeństwo

- Plik `r1_30_wypożyczalnia.sql` usuwa i tworzy od nowa bazę `wypożyczalnia` — nie zmieniaj w nim nazwy bazy na nazwę innej istniejącej bazy.
- Przed wykonaniem kw4 uruchom SELECT z tym samym warunkiem WHERE i sprawdź, które wiersze zostaną zmienione.
- Konto `serwisant` z zadania dodatkowego usuń na koniec zajęć (`DROP USER`).
- Część egzaminacyjną wykonuj samodzielnie, bez internetu i bez notatek.

Przebieg zadania

1 Start (5 min)

Na egzaminie folder ma nazwę numeru PESEL — tutaj użyj nazwiska i imienia.

1. Utwórz na pulpicie folder `nazwisko_imie_2`. Zapisz w nim zrzuty `kw1.png` ... i plik `kwerendy.txt`.
2. Zaimportuj plik `r1_30_wypożyczalnia.sql` (tworzy bazę `wypożyczalnia` z tabelami `stacje`, `rowery`, `klienci`, `wypozycczenia`). Zapisz godzinę rozpoczęcia — od tej chwili masz 60 minut.

2 Zadanie egzaminacyjne (limit 60 min)

Po każdym zapytaniu zrób zrzut ekranu z treścią zapytania i całym wynikiem (PNG lub JPEG) o nazwie `kw1`, `kw2`... Zapytania zapisuj kolejno w pliku `kwerendy.txt`. W tabeli `wypozyczenia` wypożyczenie trwające ma pustą (NULL) datę zakończenia `czas_do` i koszt.

- kw1. Wybierz numer i rok zakupu sprawnych rowerów elektrycznych (kolumna `sprawny` ma wartość 1). Posortuj rosnąco według roku zakupu.
- kw2. Wybierz imię i nazwisko klienta, numer i typ roweru oraz czas rozpoczęcia dla wypożyczeń, które jeszcze trwają. Posortuj rosnąco według czasu rozpoczęcia.
- kw3. Dla każdego typu roweru wybierz typ, liczbę zakończonych wypożyczeń (kolumna `liczba`) i średni koszt zakończonego wypożyczenia zaokrąglony do dwóch miejsc po przecinku (kolumna `sredni_koszt`). Posortuj malejąco według średniego kosztu.
- kw4. Rowery zakupione przed 2020 rokiem przenieś do stacji „Serwis” (identyfikator 5) i oznacz jako niesprawne (`sprawny` = 0) — jednym poleceniem UPDATE. Na zrzucie ma być widoczna liczba zmienionych wierszy.
5. Zapisz godzinę zakończenia. Jeśli zostało Ci czasu, wykonaj zadania dodatkowe.

3 Zadania dodatkowe (dla osób, które skończą przed czasem)

Te same zasady dokumentowania (zrzuty `kw5`, `kw6`).

- kw5. Utwórz użytkownika `serwisant@localhost` z hasłem `Rower#26`. Nadaj mu prawo odczytu tabeli `rowery` oraz prawo zmiany wyłącznie kolumn `sprawny` i `stacja_id` w tej tabeli. Pokaż SHOW GRANTS.
- kw6. Wybierz imię i nazwisko klientów, którzy nigdy nie wypożyczyli roweru (złączenie zewnętrzne).

4 Samoocena i poprawa (ok. 20 min)

Po upływie 60 minut nauczyciel pokaże wyniki wzorcowe i kryteria.

- Wypełnij tabelę samooceny: dla każdego zapytania przyznaj sobie punkty według kryteriów (kolumny, warunek, sortowanie/alias, wynik) i opisz błąd.
- Popraw błędne zapytania w nowym pliku `kwerendy_poprawione.txt` (nie zmieniaj oryginalnego `kwerendy.txt`).
- Porównaj wynik z próbą nr 1 (zadanie 29): czas, liczbę błędów i ich rodzaj. Usuń konto `serwisant`, jeśli je utworzyłeś.

W sprawozdaniu umieść

- ☐ folder `nazwisko_imie_2` ze zrzutami `kw1` – `kw4` (i `kw5` – `kw6`), plikiem `kwerendy.txt` oraz `kwerendy_poprawione.txt`
- ☐ godzina rozpoczęcia i zakończenia części egzaminacyjnej
- ☐ tabela samooceny z punktami i opisem błędów
- ☐ porównanie z próbą nr 1 (zadanie 29)
- ☐ odpowiedzi na pytania kontrolne

Pytania kontrolne

- Dlaczego w kw2 nie można napisać warunku `czas_do = NULL`?
- Co zmieniłoby się w kw3, gdyby pominąć warunek na zakończone wypożyczenia? Która kolumna by się zmieniła, a która nie — i dlaczego?
- Jak bezpiecznie przygotować polecenie UPDATE z kw4, żeby nie zmienić przypadkiem wszystkich rowerów?
- Jaką strategię kolejności rozwiązywania zadań przyjmiesz na egzaminie i dlaczego?

Kryteria oceny

Kryterium	Punkty
Import, folder i nazwy plików zgodne z poleceniem, praca w limicie 60 min	2
kw1 — kolumny, warunek złożony, sortowanie	3
kw2 — złączenie trzech tabel, warunek IS NULL, sortowanie	3
kw3 — grupowanie, aliasy, ROUND(AVG), warunek, sortowanie	4
kw4 — UPDATE dwóch kolumn z poprawnym warunkiem	3
Plik kwerendy.txt i zrzuty czytelne i kompletne	1
Samooocena, poprawki i porównanie z próbą nr 1	2
Odpowiedzi na pytania kontrolne	2
Razem	20

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

Projekt roczny cz. 1 — projekt bazy danych

Czas: 2 × 45 min

zespoły 2-osobowe

Cel: Zaprojektowanie w zespole kompletnej bazy danych dla wybranej dziedziny: od wymagań i reguł biznesowych, przez diagram ERD, po schemat relacyjny w 3NF, oraz ocena projektu innego zespołu.

Po zajęciach potrafisz:

- formułuje wymagania i reguły biznesowe dla wybranej dziedziny
- rysuje diagram ERD w notacji krótkiej stopki z krotnościami i tabelą pośredniczącą dla związku N:M
- zamienia diagram na schemat relacyjny z kluczami głównymi i obcymi
- sprawdza i uzasadnia, że schemat jest w 3NF
- recenzuje projekt innego zespołu według kryteriów i wprowadza poprawki po recenzji

Sprzęt i materiały

- komputer z dostępem do internetu (draw.io) lub zainstalowanym draw.io Desktop
- kartka A4 do szkicu

Oprogramowanie

- **draw.io (diagrams.net)** — biblioteka Entity Relation; wersja desktop: <https://www.drawio.com/app.diagrams.net>
- **Edytor tekstu lub arkusz kalkulacyjny** — słownik danych i schemat relacyjny

Przygotowanie przed zajęciami

- Uzgodnij w zespole temat z listy w kroku 1 (lub własny, zatwierdzony przez nauczyciela). Każdy zespół w klasie ma inny temat.
- Przejrzyj swoje prace z zadań 2–7 (wymagania, ERD, model relacyjny, normalizacja).

Bezpieczeństwo

- W opisie dziedziny i przykładowych danych nie używaj prawdziwych danych osobowych (nazwisk kolegów, numerów telefonów, adresów).
- Pliki projektu zapisuj w folderze zespołu i rób kopię (np. na pendrive lub w szkolnej chmurze) — będą potrzebne w zadaniach 32 i 33.

Przebieg zadania

1 Temat i wymagania

Przykładowe tematy: warsztat samochodowy, przychodnia weterynaryjna, liga e-sportowa, szkoła jazdy, siłownia (karnety i zajęcia), wypożyczalnia gier planszowych, serwis komputerowy, festiwal muzyczny (artyści, sceny, koncerty, bilety).

1. Napisz opis dziedziny (5–8 zdań): kto korzysta z bazy, jakie dane są przechowywane i jakie czynności wykonują użytkownicy.
2. Wypisz co najmniej 8 reguł biznesowych, np. „Samochód ma dokładnie jednego właściciela”, „Zlecenie może obejmować wiele usług”. Każda reguła ma dać się sprawdzić w projekcie.
3. Wypisz co najmniej 10 pytań, na które baza ma odpowiadać (np. „Ile zarobił każdy mechanik w miesiącu?”). Z nich powstaną zapytania w zadaniu 33 — co najmniej 2 pytania mają wymagać złączenia, 2 grupowania i 1 podzapytania.
4. Zaplanuj co najmniej dwie role użytkowników bazy i to, co każda z nich może robić (wykorzystasz to w zadaniu 33).

2 Encje i słownik danych

1. Wypisz encje (co najmniej 5) i ich atrybuty. Dla każdej encji wskaż identyfikator.
2. Przygotuj słownik danych dla wszystkich atrybutów: nazwa, opis, typ danych MariaDB/MySQL, wymagany (NOT NULL), unikalny, przykład wartości, ograniczenie (np. cena > 0).

3 Diagram ERD

1. Narysuj w draw.io diagram ERD w notacji kurzej stopki. Zaznacz krotności (1, N) i uczestnictwo (obowiązkowe / opcjonalne) każdego związku.
2. Projekt musi zawierać co najmniej jeden związek N:M rozbitý tabelą pośredniczącą z własnymi atrybutami (np. ilość, cena w chwili sprzedaży).
3. Wyeksportuj diagram do PNG (Plik → Eksportuj jako → PNG) i zapisz plik źródłowy `.drawio`.

4 Schemat relacyjny i 3NF

Zapis: `tabela(klucz PK, atrybut, #klucz_obcy → tabela)`.

1. Zapisz schemat relacyjny wszystkich tabel z kluczami głównymi, obcymi i unikalnymi. Zdecyduj, gdzie stosujesz klucz sztuczny (AUTO_INCREMENT), a gdzie naturalny, i krótko to uzasadnij.
2. Dla każdej tabeli sprawdź 1NF, 2NF i 3NF: wypisz zależności funkcyjne i wyjaśnij, dlaczego nie ma zależności częściowych ani przechodnich. Jeśli znajdziesz problem — popraw schemat.
3. Przypisz każdej regule biznesowej z kroku 1 element projektu, który ją realizuje (klucz obcy, UNIQUE, NOT NULL, CHECK, krotność).

5 Recenzja krzyżowa i poprawki

Zespoły wymieniają się projektami (para wskazana przez nauczyciela).

1. Wypełnij kartę recenzji projektu drugiego zespołu (tabela w sprawozdaniu): oceń każde kryterium w skali 0–2 i wpisz co najmniej trzy konkretne uwagi.
2. Na podstawie otrzymanej recenzji popraw swój projekt i opisz w sprawozdaniu, co zmieniliście (albo dlaczego nie przyjęliście uwagi).

W sprawozdaniu umieść

- ☐ opis dziedziny, reguły biznesowe (min. 8) i pytania do bazy (min. 10)
- ☐ planowane role użytkowników
- ☐ słownik danych
- ☐ diagram ERD (PNG) i plik `.drawio`

- ☐ schemat relacyjny z uzasadnieniem 3NF i przypisaniem reguł
- ☐ karta recenzji wypełniona dla innego zespołu oraz lista poprawek po recenzji własnego projektu
- ☐ odpowiedzi na pytania kontrolne

Pytania kontrolne

1. Dlaczego związku N:M nie da się zapisać bezpośrednio w modelu relacyjnym?
2. Kiedy lepszy jest klucz naturalny (np. numer rejestracyjny), a kiedy sztuczny? Podaj przykład ze swojego projektu.
3. Dlaczego w tabeli pośredniczącej przechowuje się cenę, skoro cena jest też w tabeli usług lub produktów?
4. Jaką anomalię spowodowałaby zależność przechodnia w Waszym projekcie, gdyby jej nie usunąć?

Kryteria oceny

Kryterium	Punkty
Wymagania: opis, reguły biznesowe, pytania do bazy, role	4
Słownik danych (typy, ograniczenia)	3
Diagram ERD: poprawna notacja, krotności, N:M z tabelą pośredniczącą	5
Schemat relacyjny w 3NF z uzasadnieniem	4
Recenzja innego zespołu i poprawki po recenzji	2
Odpowiedzi na pytania kontrolne	2
Razem	20

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

Projekt roczny cz. 2 — implementacja i dane testowe

Czas: 2 × 45 min

zespoły 2-osobowe (te same co w zadaniu 31)

Cel: Zaimplementowanie zaprojektowanej bazy w skrypcie SQL z pełnym zestawem ograniczeń integralności, wypełnienie jej danymi testowymi i udowodnienie testami, że ograniczenia działają.

Po zajęciach potrafisz:

- pisze skrypt DDL tworzący bazę z kodowaniem utf8mb4 i tabele we właściwej kolejności
- dobiera typy danych, klucze, UNIQUE, CHECK, DEFAULT oraz akcje ON DELETE / ON UPDATE do reguł biznesowych
- przygotowuje spójne dane testowe wieloma poleceniami INSERT
- planuje i wykonuje testy ograniczeń, interpretując kody błędów
- sprawdza, że skrypt wykonuje się bez błędów na czystym serwerze

Sprzęt i materiały

- komputer z XAMPP (MariaDB 10.4) lub MySQL 8.4
- projekt bazy z zadania 31 (ERD, schemat relacyjny, słownik danych)

Oprogramowanie

- **XAMPP (Apache + MariaDB + phpMyAdmin)** — phpMyAdmin → Widok projektu (Designer) do porównania z ERD apachefriends.org
- **VS Code** — plik `projekt.sql` (kodowanie UTF-8) code.visualstudio.com
- **Konsola mysql** — `mysql -u root < projekt.sql` — test na czystym serwerze

Przygotowanie przed zajęciami

- Wprowadź poprawki z recenzji z zadania 31.
- Przygotuj (np. w arkuszu) fikcyjne dane: co najmniej 10 wierszy w każdej tabeli głównej.

Bezpieczeństwo

- Skrypt zaczyna się od `DROP DATABASE IF EXISTS` — upewnij się, że nazwa bazy jest unikalna i nie jest nazwą istniejącej bazy (np. `sklep`).
- Dane testowe mają być fikcyjne — bez prawdziwych nazwisk, telefonów i adresów e-mail (używaj domeny `example.pl`).
- Trzymaj aktualną wersję `projekt.sql` w dwóch miejscach (folder zespołu i pendrive/chmura szkolna).

Przebieg zadania

1 Skrypt DDL

Cały projekt powstaje w jednym pliku `projekt.sql`, który można wielokrotnie uruchomić od zera.

```
SET NAMES utf8mb4;
DROP DATABASE IF EXISTS nazwa_projektu;
CREATE DATABASE nazwa_projektu CHARACTER SET utf8mb4 COLLATE utf8mb4_polish_ci;
USE nazwa_projektu;
-- najpierw tabele bez kluczy obcych, potem tabele, które się do nich odwołują
-- CREATE TABLE ... ENGINE=InnoDB;
```

1. Z1. Napisz polecenia CREATE TABLE dla wszystkich tabel ze schematu z zadania 31: typy ze słownika danych, PRIMARY KEY (w tabeli pośredniczącej — złożony lub sztuczny z UNIQUE), AUTO_INCREMENT, NOT NULL, DEFAULT.
2. Z2. Dodaj klucze obce z nazwami (CONSTRAINT fk_...) i świadomie wybierz akcję ON DELETE (RESTRICT, CASCADE lub SET NULL) dla każdego z nich. Zapisz w tabeli sprawozdania uzasadnienie każdego wyboru.
3. Z3. Dodaj co najmniej trzy ograniczenia UNIQUE i trzy CHECK, w tym jedno CHECK porównujące dwie kolumny (np. data zakończenia $\geq$ data rozpoczęcia).
4. Z4. Uruchom skrypt i porównaj diagram w phpMyAdmin (Widok projektu) ze swoim ERD. Zrób zrzut ekranu. Wykonaj SHOW CREATE TABLE dla tabeli pośredniczącej.

2 Dane testowe

1. Z5. Dopisz do skryptu polecenia INSERT (jedno polecenie z wieloma wierszami na tabelę): co najmniej 10 wierszy w każdej tabeli głównej i co najmniej 15 w tabeli pośredniczącej. Dane mają pozwolić odpowiedzieć na pytania z zadania 31 (np. różne miesiące, wiersze z NULL tam, gdzie to dozwolone, rekordy bez powiązań).
2. Z6. Napisz jedno zapytanie (UNION ALL) zwracające liczbę wierszy w każdej tabeli. Zapisz wynik.
3. Z7. Uruchom cały skrypt dwa razy z konsoli (`mysql -u root < projekt.sql`). Drugi przebieg też musi zakończyć się bez błędów — popraw skrypt, jeśli trzeba.

3 Testy ograniczeń integralności

Każdy test to jedno polecenie, które celowo łamie regułę. Zapisz testy w osobnym pliku `testy.sql` (nie w `projekt.sql`), bo część z nich zmienia dane.

1. Z8. Przygotuj i wykonaj co najmniej 8 testów: klucz obcy przy wstawianiu (1452), usuwanie rodzica przy RESTRICT (1451), duplikat UNIQUE (1062), naruszenie CHECK (MariaDB 4025, MySQL 3819), brak wartości w kolumnie NOT NULL (1364), działanie ON DELETE CASCADE, działanie ON DELETE SET NULL oraz jeden test własnej reguły biznesowej.
2. Z9. Dla każdego testu wpisz do tabeli: regułę, polecenie, oczekiwany wynik i faktyczny wynik (kod błędu lub liczbę zmienionych wierszy). Jeśli test nie dał oczekiwanego wyniku, popraw skrypt DDL i powtórz test.
3. Z10. Po testach odtwórz bazę ze skryptu `projekt.sql`, żeby wrócić do kompletnych danych.

W sprawozdaniu umieść

- ☐ plik `projekt.sql` (DDL + dane) wykonujący się bez błędów dwukrotnie
- ☐ plik `testy.sql` i tabela testów ograniczeń z wynikami
- ☐ uzasadnienie akcji ON DELETE dla każdego klucza obcego
- ☐ zrzut Widoku projektu z phpMyAdmin i wynik SHOW CREATE TABLE tabeli pośredniczącej
- ☐ wynik zapytania z liczbą wierszy w tabelach
- ☐ odpowiedzi na pytania kontrolne

Pytania kontrolne

1. Dlaczego kolejność poleceń CREATE TABLE i INSERT w skrypcie ma znaczenie? Jak inaczej można rozwiązać problem kolejności?
2. Kiedy wybierzesz ON DELETE CASCADE, a kiedy RESTRICT? Podaj przykład z własnego projektu.
3. Co się stanie z ograniczeniem CHECK w MariaDB starszej niż 10.2 lub MySQL starszym niż 8.0.16?
4. Po co dane testowe zawierają rekordy bez powiązań (np. klienta bez zamówień)?

Kryteria oceny

Kryterium	Punkty
Skrypt DDL: typy, klucze, NOT NULL, DEFAULT, kolejność, utf8mb4	5
Klucze obce z uzasadnionymi akcjami, UNIQUE i CHECK	4
Dane testowe: liczba wierszy, spójność, przydatność do zapytań	4
Testy ograniczeń (min. 8) z tabelą wyników	4
Skrypt działa dwukrotnie na czystym serwerze	1
Odpowiedzi na pytania kontrolne	2
Razem	20

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

Projekt roczny cz. 3 — zapytania, użytkownicy, kopia i prezentacja

Czas: 2 × 45 min

zespoły 2-osobowe (te same co w zadaniach 31–32)

Cel: Dokończenie projektu rocznego: zapytania raportowe odpowiadające na pytania z zadania 31, konta z różnymi uprawnieniami, kopia zapasowa z próbnym odtworzeniem, dokumentacja i krótka prezentacja.

Po zajęciach potrafisz:

- pisze zapytania raportowe łączące JOIN, GROUP BY, HAVING i podzapytania
- sprawdza poprawność wyniku zapytania na danych testowych
- tworzy konta odpowiadające rolom w projekcie i testuje ich uprawnienia
- wykonuje kopię zapasową i próbną odtworzenie projektu
- dokumentuje i prezentuje projekt bazy w 3 minuty

Sprzęt i materiały

- komputer z XAMPP (MariaDB 10.4) lub MySQL 8.4
- pliki projektu z zadań 31–32 (`projekt.sql`, ERD, schemat)
- rzutnik do prezentacji

Oprogramowanie

- **XAMPP (Apache + MariaDB + phpMyAdmin)** — phpMyAdmin i konsola mysql, mysqldump
apachefriends.org
- **VS Code** — pliki `zapytania.sql`, `konta.sql`
code.visualstudio.com
- **Program do prezentacji (PowerPoint, LibreOffice Impress lub Google Slides)** — 3–4 slajdy
libreoffice.org

Przygotowanie przed zajęciami

- Odtwórz bazę projektu ze skryptu `projekt.sql` (zadanie 32).
- Przejrzyj listę pytań do bazy z zadania 31 i podziel zapytania między członków zespołu.

Bezpieczeństwo

- Konta testowe twórz tylko dla bazy projektu; nie nadawaj uprawnień `ON *.*` ani `WITH GRANT OPTION`. Na koniec zajęć usuń konta (DROP USER), chyba że nauczyciel chce je sprawdzić.
- Kopię odtwarzaj wyłącznie do nowej bazy `nazwa_projektu_test`, nigdy do bazy projektu.
- W prezentacji i na zrzutach nie pokazuj haseł kont.

Przebieg zadania

1 Zapytania raportowe

Zapytania zapisz w pliku `zapytania.sql`: przed każdym komentarz z numerem i pytaniem z zadania 31 (np. `-- Z3. Jaka jest wartość każdego zlecenia?`).

1. Z1–Z10. Napisz 10 zapytań odpowiadających na pytania z zadania 31. Wymagania: co najmniej 3 zapytania ze złączeniem (w tym jedno LEFT JOIN), co najmniej 3 z GROUP BY (w tym jedno z HAVING), co najmniej 1 z podzapytaniem lub NOT EXISTS, co najmniej 1 z funkcją daty i 1 z ORDER BY ... LIMIT.
2. Dla dwóch wybranych zapytań z grupowaniem sprawdź wynik ręcznie: wybierz jedną grupę, wypisz jej wiersze prostym SELECT i policz sumę lub liczbę samodzielnie. Zapisz obliczenia w sprawozdaniu.
3. Jeśli dane testowe nie wystarczają, żeby zapytanie zwróciło sensowny wynik (np. pusty wynik LEFT JOIN ... IS NULL), dopisz dane do `projekt.sql` i odnotuj zmianę.

2 Konta użytkowników

Utwórz konta dla dwóch ról zaplanowanych w zadaniu 31.

1. Z11. Utwórz pierwsze konto (np. pracownik obsługi) z uprawnieniami do odczytu całej bazy i dopisywania wierszy w wybranych tabelach.
2. Z12. Utwórz drugie konto z węższymi uprawnieniami: dostęp tylko do wybranych tabel i co najmniej jedno uprawnienie na poziomie kolumny (np. UPDATE tylko kolumny `status`).
3. Dla każdego konta wykonaj co najmniej dwa testy dozwolonych i dwa testy zabronionych operacji. Zapisz polecenia w `konta.sql`, a wyniki w tabeli testów.

3 Kopia zapasowa

1. Z13. Wykonaj kopię bazy projektu poleceniem mysqldump (z opcjami `--routines --triggers --single-transaction`) do pliku z datą w nazwie. Odtwórz ją do nowej bazy `nazwa_projektu_test`, porównaj liczbę wierszy i CHECKSUM TABLE dwóch wybranych tabel, a następnie usuń bazę testową.

4 Dokumentacja i prezentacja

Prezentacja: 3 minuty na zespół, potem 1–2 pytania od klasy.

1. Przygotuj dokumentację projektu (sprawozdanie): opis dziedziny, ERD, schemat relacyjny, lista plików, opis kont i instrukcja uruchomienia (`mysql -u root < projekt.sql`).
2. Przygotuj 3–4 slajdy: dziedzina i reguły, ERD, dwa najciekawsze zapytania z wynikiem, konta i kopia. Podziel prezentację między członków zespołu.
3. Spakuj do archiwum ZIP: `projekt.sql`, `testy.sql`, `zapytania.sql`, `konta.sql`, plik kopii, ERD (PNG i `.drawio`), sprawozdanie i prezentację. Oddaj zgodnie z poleceniem nauczyciela.

W sprawozdaniu umieść

- ☐ plik `zapytania.sql` (Z1–Z10) i tabela zapytań z pytaniem i liczbą wierszy
- ☐ ręczne sprawdzenie dwóch zapytań z grupowaniem
- ☐ plik `konta.sql` i tabela testów uprawnień
- ☐ przebieg kopii i próbnego odtworzenia (polecenia, wynik porównania)
- ☐ dokumentacja projektu i prezentacja
- ☐ archiwum ZIP z kompletem plików
- ☐ odpowiedzi na pytania kontrolne

Pytania kontrolne

1. Jak sprawdzić, czy zapytanie z GROUP BY zwraca poprawny wynik, nie znając go wcześniej?
2. Dlaczego jedno z kont ma uprawnienie na poziomie kolumny zamiast całej tabeli? Jakie ryzyko to ogranicza?
3. Co zmieniłbyś w projekcie, gdyby baza miała obsługiwać kilka oddziałów firmy?
4. Dlaczego kopia zapasowa projektu jest niepełna, jeśli nie zawiera kont użytkowników, i jak można je odtworzyć?

Kryteria oceny

Kryterium	Punkty
Zapytania Z1–Z10: poprawność i spełnienie wymagań (JOIN, GROUP BY, podzapytanie...)	7
Ręczne sprawdzenie wyników dwóch zapytań	1
Dwa konta z różnymi uprawnieniami i testy	4
Kopia i próbne odtworzenie	2
Dokumentacja i archiwum ZIP kompletne	2
Prezentacja (3 min, udział obu osób, odpowiedzi na pytania)	2
Odpowiedzi na pytania kontrolne	2
Razem	20

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

Zadanie zaliczeniowe roku 1 — szkoła językowa

Czas: 2 × 45 min

praca samodzielna

Cel: Samodzielne wykonanie w 90 minut pełnego cyklu pracy z bazą: utworzenie bazy i tabel z ograniczeniami według opisu, wprowadzenie danych, zapytania, konto z ograniczonymi uprawnieniami i kopia zapasowa.

Po zajęciach potrafisz:

- tworzy bazę i tabele z kluczami, ograniczeniami i właściwymi typami na podstawie opisu słownego
- wprowadza dane i sprawdza działanie ograniczeń
- pisze zapytania z warunkami, złączeniami, grupowaniem i podzapytaniami
- tworzy konto z minimalnymi uprawnieniami i sprawdza je
- wykonuje i weryfikuje kopię zapasową bazy

Sprzęt i materiały

- komputer z XAMPP (MariaDB 10.4) lub MySQL 8.4
- narzędzie do zrzutów ekranu

Oprogramowanie

- **XAMPP (Apache + MariaDB + phpMyAdmin)** — phpMyAdmin, konsola mysql, mysqldump
apachefriends.org
- **Notatnik lub VS Code** — pliki `baza.sql` i `zapytania.sql`
code.visualstudio.com

Przygotowanie przed zajęciami

- Powtórz materiał z zadań 9–12 (DDL i DML), 19–25 (zapytania), 27 (uprawnienia) i 28 (kopie).
- Utwórz folder `C:\kopie`, jeśli go nie ma.

Bezpieczeństwo

- Pracuj wyłącznie na bazie `szkola_jezykowa` i bazie testowej `szkola_test`; nie zmieniaj innych baz.
- Przed poleceniem UPDATE (Z11) wykonaj SELECT z tym samym warunkiem WHERE.
- Na koniec usuń konto `sekretariat` i bazę `szkola_test`.
- Pracuj samodzielnie, bez internetu.

Przebieg zadania

1 Baza i tabele według opisu (ok. 25 min)

Szkoła językowa prowadzi kursy. Każdy kurs ma język (do 30 znaków), poziom (jedna z wartości A1, A2, B1, B2, C1), cenę w zł z groszami (większą od zera), limit miejsc (liczba całkowita 1–255, domyślnie 12) i datę startu. Kursant ma imię, nazwisko, unikalny adres e-mail i opcjonalny rok urodzenia. Zapis łączy kursanta z kursem i ma datę zapisu oraz informację, czy kurs został opłacony (domyślnie nie). Kursant może zapisać się na ten sam kurs tylko raz. Nie można usunąć kursu, na który ktoś jest zapisany; usunięcie kursanta usuwa jego zapisy.

1. Z1. Utwórz bazę `szkoła_językowa` z kodowaniem `utf8mb4` i porównywaniem `utf8mb4_polish_ci`. Utwórz tabele `kursy`, `kursanci` i `zapisy` zgodnie z opisem: klucze główne z `AUTO_INCREMENT`, klucze obce z właściwymi akcjami `ON DELETE`, `NOT NULL`, `UNIQUE`, `CHECK` i `DEFAULT`. Zapisz polecenia w pliku `baza.sql`.
2. Z2. Wprowadź dane z ramki w kroku 2 (numery w ramce to identyfikatory nadawane przez `AUTO_INCREMENT` — wstawiaj wiersze w podanej kolejności). Dopisz polecenia `INSERT` do `baza.sql`.
3. Z3. Sprawdź trzy ograniczenia: drugi zapis tego samego kursanta na ten sam kurs, zapis na nieistniejący kurs nr 9 i kurs z ceną 0. Zapisz kody błędów.

2 Dane

```
kursy (jezyk, poziom, cena, limit_miejsc, data_startu):
1 angielski B1 1200.00 12 2026-10-05    2 angielski B2 1350.00 10 2026-10-12
3 niemiecki A1 990.00 12 2026-10-05    4 hiszpański A2 1100.00 8 2026-11-02
5 włoski    A1 950.00 10 2026-11-09

kursanci (imie, nazwisko, email, rok_urodzenia):
1 Alicja Mazur alicja.m@example.pl 2004    2 Bartek Lis bartek.l@example.pl 1998
3 Celina Nowak celina.n@example.pl 2007    4 Dawid Kot dawid.k@example.pl 1985
5 Emilia Wrona emilia.w@example.pl 2001    6 Franek Sowa franek.s@example.pl 2009
7 Gabriela Pawlak gabriela.p@example.pl 1979 8 Henryk Zając henryk.z@example.pl (brak roku)

zapisy (id_kursu, id_kursanta, data_zapisu, oplacony):
1,1,2026-09-10,1  1,2,2026-09-12,1  1,3,2026-09-15,0  2,4,2026-09-20,1
2,1,2026-09-21,1  3,5,2026-09-01,1  3,6,2026-09-03,0  3,2,2026-09-25,1
4,7,2026-10-01,0  4,3,2026-10-02,1  1,5,2026-09-28,1  4,4,2026-10-06,0
```

1. Po wprowadzeniu danych sprawdź liczbę wierszy w każdej tabeli (5, 8, 12).

3 Zapytania (ok. 40 min)

Każde zapytanie zapisz w `zapytania.sql` z komentarzem, a wynik udokumentuj zrzutem ekranu.

1. Z4. Język, poziom, cena i data startu kursów rozpoczynających się w listopadzie 2026 r., posortowane według daty startu.
2. Z5. Imię i nazwisko kursantów zapisanych na dowolny kurs języka angielskiego — każda osoba tylko raz, posortowane według nazwiska.
3. Z6. Dla każdego kursu (także bez zapisów): język, poziom, liczba zapisów (`liczba_zapisow`) i liczba wolnych miejsc (`wolne_miejsca`). Posortuj rosnąco według wolnych miejsc.
4. Z7. Suma wpłat za opłacone zapisy w podziale na języki (`wplaty`) — tylko języki, dla których suma przekracza 1500 zł, malejąco.
5. Z8. Imię i nazwisko kursantów, którzy nie są zapisani na żaden kurs (`NOT EXISTS` lub `LEFT JOIN`).
6. Z9. Kursy (język, poziom, cena) droższe od średniej ceny wszystkich kursów (podzapytanie).
7. Z10. Kursanci zapisani na więcej niż jeden kurs: imię, nazwisko i liczba kursów (`kursow`).
8. Z11. Podnieś o 5% (z zaokrągleniem do groszy) cenę kursów rozpoczynających się od 1 listopada 2026 r. Zrzut ma pokazać liczbę zmienionych wierszy.

4 Użytkownik i kopia (ok. 20 min)

1. Z12. Utwórz konto `sekretariat@localhost` (hasło `Sekr#2026`) z prawem odczytu całej bazy `szkola_jezykowa` oraz dodawania i zmieniania wierszy tylko w tabeli `zapisy`. Zaloguj się na nie i wykonaj jedną dozwoloną i jedną zabronioną operację.
2. Z13. Wykonaj kopię bazy poleceniem `mysqldump` do `C:\kopie\szkola_jezykowa.sql`, odtwórz ją do bazy `szkola_test` i porównaj liczbę wierszy tabeli `zapisy` w obu bazach jednym zapytaniem.
3. Usuń bazę `szkola_test` i konto `sekretariat`. Oddaj pliki `baza.sql`, `zapytania.sql`, plik kopii i sprawozdanie ze zrzutami.

W sprawozdaniu umieść

- ☐ plik `baza.sql` (DDL i dane) oraz wynik testów ograniczeń Z3
- ☐ plik `zapytania.sql` (Z4–Z11) i zrzuty wyników
- ☐ polecenia i zrzuty z Z12 (SHOW GRANTS, test dozwolony i zabroniony)
- ☐ polecenia kopii i porównanie z Z13
- ☐ odpowiedzi na pytania kontrolne

Pytania kontrolne

1. Dlaczego w tabeli `zapisy` potrzebne jest ograniczenie UNIQUE na parze kolumn, skoro jest już klucz główny?
2. Dlaczego w Z5 bez DISTINCT jedna osoba pojawia się dwa razy?
3. Dlaczego w Z6 trzeba użyć LEFT JOIN i COUNT(kolumna), a nie COUNT(*)?
4. Jak dobrać akcje ON DELETE do reguł „nie można usunąć kursu z zapisami” i „usunięcie kursanta usuwa zapisy”?

Kryteria oceny

Kryterium	Punkty
Z1: baza i tabele zgodne z opisem (typy, klucze, ograniczenia, ON DELETE)	5
Z2–Z3: dane i testy ograniczeń	2
Z4–Z11: zapytania (po 1 pkt)	8
Z12: konto z minimalnymi uprawnieniami i test	2
Z13: kopia, odtworzenie i porównanie	1
Odpowiedzi na pytania kontrolne	2
Razem	20

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