

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).