EF Core — relacje: dziennik ocen

Czas: 2 × 45 min | praca indywidualna | Wykład 2.14

Cel: Zbudujesz dziennik ocen z trzema powiązanymi encjami: listą uczniów ze średnimi, szczegółami ocen wybranego ucznia i średnimi przedmiotów liczonymi w bazie.

Po zajęciach potrafisz:

- modeluje relacje 1:N z kluczami obcymi i właściwościami nawigacyjnymi
- wczytuje dane powiązane Include/ThenInclude
- pisze zapytania LINQ z grupowaniem tłumaczone na SQL
- nakłada ograniczenia w Fluent API

Sprzęt i materiały

- komputer z Windows 11 i dostępem do internetu
- pendrive lub konto w repozytorium do zapisania projektu

Oprogramowanie

- **Visual Studio Community 2026 (obciążenie „Programowanie aplikacji klasycznych platformy .NET")** — darmowe dla uczniów i nauczycieli; w pracowni zainstalowane visualstudio.microsoft.com/pl/downloads
- **.NET 10 SDK (LTS)** — instalowany razem z Visual Studio; sprawdzenie: `dotnet --version` dotnet.microsoft.com/pl-pl/download/dotnet/10.0
- **Pakiet NuGet CommunityToolkit.Mvvm 8.4** — instalacja: Zarządzaj pakietami NuGet nuget.org/packages/CommunityToolkit.Mvvm
- **Pakiety Microsoft.EntityFrameworkCore.Sqlite i .Tools 10.x** — wersja zgodna z .NET 10 nuget.org/packages/Microsoft.EntityFrameworkCore.Sqlite
- **DB Browser for SQLite** — podgląd pliku bazy; wersja przenośna nie wymaga instalacji sqlitebrowser.org/dl

Bezpieczeństwo

- Zapisuj projekt w swoim folderze (np. `C:\Users\<login>\source\repos`), a na koniec zajęć skopiuj go na pendrive lub wyślij do repozytorium.
- Nie uruchamiaj programów i pakietów z nieznanych źródeł; pakiety NuGet instaluj tylko te, które wskazuje zadanie.
- Rób przerwy dla oczu (zasada 20-20-20) — praca z projektantem okien wymaga długiego patrzenia w ekran.

Przebieg zadania

1 Model

Projekt WPF `Dziennik`.

1. Encje `Uczen` (Imie, Nazwisko, Klasa, Oceny), `Przedmiot` (Nazwa, Oceny), `Ocena` (Wartosc, Data, Opis, UczenId, Uczen, PrzedmiotId, Przedmiot).
2. W Uczen właściwości pomocnicze `PelneImie` i `Srednia` (null, gdy brak ocen).
3. `DziennikContext`: ograniczenie CHECK `Wartosc BETWEEN 1 AND 6`, unikalna nazwa przedmiotu, dane startowe (3 uczniów, 3 przedmioty, 7 ocen).
4. Migracja i `Migrate()` przy starcie.

2 Zapytania

`DziennikViewModel`.

```
foreach (var s in db.Oceny
    .GroupBy(o => o.Przedmiot.Nazwa)
    .Select(g => new SredniaPrzedmiotu(g.Key, g.Average(o => o.Wartosc), g.Count()))
    .OrderBy(s => s.Przedmiot))
    SredniePrzedmiotow.Add(s);
```

1. Uczniowie z ocenami i przedmiotami: `Include(u => u.Oceny).ThenInclude(o => o.Przedmiot).AsNoTracking()`.
2. Średnie przedmiotów liczone w bazie: `GroupBy(o => o.Przedmiot.Nazwa).Select(g => new SredniaPrzedmiotu(...))`.
3. Podejrzyj SQL: `Debug.WriteLine(zapytanie.ToQueryString())` i zapisz, czy jest w nim GROUP BY.
4. Sprawdź, co pokazuje lista ocen, gdy usuniesz Include.

3 Interfejs

Okno z listą i szczegółami.

1. DataGrid uczniów: uczeń, klasa, liczba ocen, średnia (dla braku ocen „—”: `TargetNullValue`).
2. Panel szczegółów: oceny wybranego ucznia (`Run` z przedmiotem, wartością i datą).
3. Dodawanie oceny: przedmiot (ComboBox z DisplayMemberPath), wartość 1–6, przycisk aktywny po wyborze ucznia i przedmiotu.
4. Lista średnich przedmiotów odświeżana po dodaniu oceny.

4 Sprawdzenie

Integralność danych.

1. Spróbuj w DB Browser wpisać ręcznie ocenę 7 — co się dzieje?
2. Porównaj średnie z aplikacji z obliczeniem ręcznym.
3. Zatwierdź pracę w Git.

W sprawozdaniu umieść

- ☐ kod modelu i kontekstu
- ☐ wygenerowany SQL zapytania średnich
- ☐ zrzuty okna
- ☐ wynik próby wpisania oceny 7

Pytania kontrolne

1. Jak EF rozpoznaje klucz obcy UczenId?
2. Dlaczego bez Include lista ocen jest pusta?
3. Gdzie liczona jest średnia przedmiotu — w bazie czy w pamięci? Skąd wiesz?
4. Do czego służy ograniczenie CHECK?

Kryteria oceny

Kryterium	Punkty
Model i relacje 1:N	5
Fluent API i dane startowe	3
Zapytania Include i GroupBy	5
Interfejs z dodawaniem ocen	4
Pytania	3
Razem	**20**

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