



## Bazy danych w C# cz. II

Celem lekcji jest dalsze omówienie tematu baz danych z użyciem języka C#.



# Przypomnienie



## 1. Co to jest SQL?

- **SQL** (Structured Query Language) jest językiem zapytań służącym do zarządzania i manipulacji danymi w relacyjnych bazach danych. Używany jest do tworzenia, modyfikowania, zarządzania i przeszukiwania baz danych.

## 2. Do czego służy komenda SELECT?

- **Select** jest poleceniem używanych do wybierania z bazy danych.

## 3. Jak się usuwa elementy z bazy danych?

- Używamy do tego komendy **Delete**.

## 4. Kiedy warto używać ORDER BY?

- Kiedy chcemy posortować wyniki według określonej kolumny. Na przykład posortować tabelę z danymi pracowników alfabetycznie po nazwisku.





**ORM** (Object-Relational Mapping) to technika programowania, która umożliwia:

- konwersję danych między niekompatybilnymi systemami, w szczególności pomiędzy obiektowo zorientowanymi językami programowania (czyli na przykład C#), a relacyjnymi bazami danych.
- na manipulowanie danymi w bazie danych za pomocą obiektów w kodzie, zamiast bezpośredniego pisania zapytań SQL.



# Entity Framework



**Entity Framework** (EF) to Object Related Mapping (ORM) framework czyli zestaw narzędzi do mapowania obiektowo relacyjnego opracowany przez Microsoft dla .NET.

Zastosowania:

- Mapowanie obiektowo-relacyjne: EF automatycznie mapuje klasy i właściwości na odpowiednie tabele i kolumny w bazie danych.
- Ułatwienie zarządzania zmianami: EF ułatwia zarządzanie migracjami schematu bazy danych w miarę rozwoju aplikacji.
- Wsparcie dla wielu baz danych: EF obsługuje różne systemy zarządzania bazami danych (na przykład SQL Server, MySQL, PostgreSQL).



# Model Builder



**Model Builder** w Entity Framework Core to:

- Narzędzie, które pomaga zdefiniować, jak nasze klasy (modele) są mapowane na tabele w bazie danych.
- Umożliwia konfigurowanie tych mapowań w bardziej zaawansowany sposób niż za pomocą atrybutów bezpośrednio w klasach.



# Programowanie asynchroniczne



**Programowanie asynchroniczne** umożliwia wykonywanie operacji w tle, co pozwala na responsywność aplikacji nawet podczas wykonywania długotrwałych zadań. Dzięki temu aplikacje mogą reagować na interakcje użytkownika lub inne zdarzenia w czasie rzeczywistym.

Kluczowymi słowami w języku C# do programowania asynchronicznego są właśnie `async` i `await`. Pomagają one w zarządzaniu operacjami, które mogą zająć dużo czasu (np. operacje sieciowe, odczyt/zapis plików, operacje na bazie danych), bez blokowania głównego wątku aplikacji.

- **async:** Słowo kluczowe używane do deklarowania metody asynchronicznej. Metoda oznaczona jako `async` może zawierać słowo kluczowe `await`.
- **await:** Słowo kluczowe używane do wstrzymania wykonywania metody asynchronicznej do momentu zakończenia zadania (Task).  
Gdy zadanie się zakończy, metoda kontynuuje wykonywanie od miejsca, w którym zostało użyte `await`.  
W praktyce, `await` powoduje, że wykonanie metody zostaje wstrzymane, **dopóki** zadanie się nie zakończy.



# Chcesz dowiedzieć się więcej ?



Chcesz dowiedzieć się więcej o programowaniu asynchronicznym ? Zajrzyj do książek:

- *C Sharp 10. Programowanie. Tworzenie aplikacji Windows, internetowych i biurowych*  
- Ian Griffiths
- *Concurrency in C# Cookbook: Asynchronous, Parallel, and Multithreaded Programming*  
- Stephen Cleary.

