

# Bazy danych



**Analiza rzeczywistości jako podstawa projektowania bazy danych**

**Dr inż. Arkadiusz Rzucidło, prof. PRz**

# Informacja w świecie cyfrowym

- ◉ Na poprzednim wykładzie rozmawialiśmy o tym, **dlaczego istnieją bazy danych i dlaczego współczesne systemy informatyczne muszą przechowywać ogromne ilości informacji.**
- ◉ Pokazaliśmy również, że baza danych jest **cyfrowym modelem fragmentu rzeczywistości**. System informatyczny nie tworzy nowych zjawisk — on jedynie **opisuje rzeczywisty świat w postaci danych.**
- ◉ Na dzisiejszym wykładzie skupimy się na pytaniu:
  - ◉ Jak przejść od rzeczywistości do modelu danych?
- ◉ Innymi słowy: zanim stworzymy bazę danych, musimy **zrozumieć system, który chcemy opisać**. Ten etap nazywamy **analizą systemu**.
- ◉ Analiza jest niezwykle ważna, ponieważ błędy popełnione na tym etapie bardzo trudno naprawić później, gdy system jest już zaprojektowany lub wdrożony.
- ◉ Dlatego w praktyce projektowania systemów informatycznych analiza często zajmuje **bardzo dużą część całego projektu.**

# Informacja w świecie cyfrowym

- ◉ Na poprzednim wykładzie rozmawialiśmy o tym, **dlaczego istnieją bazy danych i dlaczego współczesne systemy informatyczne muszą przechowywać ogromne ilości informacji.**
- ◉ Pokazaliśmy również, że baza danych jest **cyfrowym modelem fragmentu rzeczywistości**. System informatyczny nie tworzy nowych zjawisk — on jedynie **opisuje rzeczywisty świat w postaci danych.**
- ◉ Na dzisiejszym wykładzie skupimy się na pytaniu:
  - ◉ Jak przejść od rzeczywistości do modelu danych?
- ◉ Innymi słowy: zanim stworzymy bazę danych, musimy **zrozumieć system, który chcemy opisać**. Ten etap nazywamy **analizą systemu**.
- ◉ Analiza jest niezwykle ważna, ponieważ błędy popełnione na tym etapie bardzo trudno naprawić później, gdy system jest już zaprojektowany lub wdrożony.
- ◉ Dlatego w praktyce projektowania systemów informatycznych analiza często zajmuje **bardzo dużą część całego projektu.**

# Informacja w świecie cyfrowym

Co jest pierwszym krokiem przy projektowaniu systemu bazodanowego?

- A. analiza rzeczywistości
- B. instalacja bazy danych
- C. napisanie programu
- D. testowanie systemu

# Cele wykładu



- ◉ Zrozumieć czym jest **analiza systemu**
- ◉ Potrafić **identyfikować obiekty systemu**
- ◉ Potrafić **określać relacje między obiektami**
- ◉ Zrozumieć różnicę między **analizą strukturalną i obiektową**
- ◉ Potrafić stworzyć **prosty model systemu**

# Cele wykładu



- ◉ Celem dzisiejszego wykładu jest pokazanie, **jak analizować rzeczywistość z punktu widzenia systemu informatycznego.**
- ◉ Projektując bazę danych, nie zaczynamy od tworzenia tabel. Najpierw musimy odpowiedzieć na kilka podstawowych pytań:
  - ◉ jakie obiekty istnieją w systemie?
  - ◉ jakie dane opisują te obiekty?
  - ◉ jakie relacje występują między nimi?

# Cele wykładu



- ◉ Na przykład w systemie uczelnianym mamy obiekty takie jak:
  - ◉ student
  - ◉ wykładowca
  - ◉ przedmiot
- ◉ Natomiast w systemie medycznym mogą to być:
  - ◉ pacjent
  - ◉ lekarz
  - ◉ wizyta
  - ◉ recepta
- ◉ Rozeznanie w tym temacie pozwala spojrzeć na dowolny system i spróbować **zidentyfikować jego podstawowe elementy informacyjne**.
- ◉ To jest pierwszy krok do stworzenia **modelu danych**, a później **bazy danych**.

# Cele wykładu



Co jest najważniejszym celem analizy systemu?

- A. projekt interfejsu użytkownika
- B. instalacja oprogramowania bazy danych
- C. zrozumienie struktury danych systemu
- D. wybór języka programowania

# Dlaczego analiza systemu jest ważna

- ⦿ Błędy analizy prowadzą do:
  - ⦿ błędnego modelu danych
  - ⦿ problemów w działaniu systemu
  - ⦿ trudności w rozwoju systemu
  - ⦿ niespójności informacji

# Dlaczego analiza systemu jest ważna

- ⦿ W projektach informatycznych istnieje zasada:
  - ⦿ Najdroższe błędy to błędy popełnione na etapie analizy.
- ⦿ Jeżeli projektant systemu źle zrozumie rzeczywistość, system będzie przechowywał dane w nieodpowiedni sposób.

Przykład:

- ⦿ Jeżeli w systemie medycznym nie przewidzimy, że pacjent może mieć **wiele wizyt**, baza danych może zostać zaprojektowana błędnie.

# Dlaczego analiza systemu jest ważna

- ◉ W efekcie później pojawią się problemy:
  - ◉ trudność w przechowywaniu danych
  - ◉ konieczność przebudowy systemu
  - ◉ utrata spójności danych
- ◉ Dlatego analiza systemu ma kluczowe znaczenie dla jakości całego projektu informatycznego.

# Dlaczego analiza systemu jest ważna

Na którym etapie projektu najłatwiej poprawić błędy systemu?

- A analiza systemu
- B implementacja systemu
- C wdrożenie systemu
- D eksploatacja systemu

# Etapy tworzenia systemu informatycznego



- ◉ Proces projektowania systemu:

1. analiza rzeczywistości
2. model systemu
3. projekt bazy danych
4. implementacja systemu
5. testowanie systemu

# Etapy tworzenia systemu informatycznego



- ◉ Proces tworzenia systemu informatycznego przebiega zazwyczaj w kilku etapach.
  - ◉ Pierwszym etapem jest **analiza rzeczywistości**, czyli zrozumienie problemu, który system ma rozwiązać.
  - ◉ Następnie tworzymy **model systemu**, który opisuje obiekty i relacje między nimi.
  - ◉ Na podstawie tego modelu projektujemy **strukturę bazy danych**, czyli tabele i ich powiązania.

# Etapy tworzenia systemu informatycznego



- ◉ Dopiero później rozpoczyna się właściwa **implementacja systemu**, czyli pisanie programu.
- ◉ Na końcu system musi zostać **przetestowany**, aby upewnić się, że działa zgodnie z założeniami.
- ◉ Warto zauważyć, że **baza danych pojawia się dopiero po etapie analizy i modelowania**.

# Etapy tworzenia systemu informatycznego



Na którym etapie powstaje model danych systemu?

- A implementacji
- B testowania
- C modelowania systemu
- D instalacji serwera

# Metody analizy systemów

- ◉ Dwie główne metody analizy:
  - ◉ **analiza strukturalna**
  - ◉ **analiza obiektowa**
- ◉ Analiza strukturalna koncentruje się przede wszystkim na:
  - ◉ przepływie informacji w systemie
  - ◉ procesach przetwarzania danych
- ◉ Natomiast analiza obiektowa skupia się na:
  - ◉ obiektach systemu
  - ◉ ich właściwościach
  - ◉ relacjach między obiektami
- ◉ W projektowaniu baz danych szczególnie przydatna jest **analiza obiektowa**, ponieważ pozwala łatwo przejść od obiektów rzeczywistości do **encji w bazie danych**.

# Metody analizy systemów

Która metoda analizy koncentruje się na obiektach i ich właściwościach?

- A analiza strukturalna
- B analiza obiektowa
- C analiza sieciowa
- D analiza statystyczna

# System do analizy: przychodnia

- ◉ Analizowany system:
  - ◉ **system rejestracji pacjentów w przychodni**
- ◉ Scenariusz:
  - ◉ pacjent
    - rejestracja wizyty
    - wizyta lekarska
    - diagnoza
    - recepta
    - realizacja recepty

# System do analizy: przychodnia

- ⦿ Aby lepiej zrozumieć proces analizy systemu, przeanalizujemy **konkretny przykład**.
- ⦿ Będzie to system informatyczny obsługujący **przychodnię lekarską**.
  - ⦿ Pacjent chce skorzystać z wizyty lekarskiej. Może zadzwonić do rejestracji lub zarejestrować się przez Internet.
  - ⦿ Rejestracja zapisuje pacjenta na wizytę u konkretnego lekarza.
  - ⦿ Podczas wizyty lekarz bada pacjenta i może wystawić receptę.
  - ⦿ Pacjent udaje się z receptą do apteki, gdzie otrzymuje leki.

# System do analizy: przychodnia

- ⦿ Ten prosty scenariusz zawiera już wiele elementów informacyjnych, które możemy analizować:
  - ⦿ pacjent
  - ⦿ lekarz
  - ⦿ wizyta
  - ⦿ recepta
  - ⦿ lek
  - ⦿ apteka
- ⦿ Na kolejnych slajdach spróbujemy przekształcić ten scenariusz w **model systemu informacyjnego**.

# System do analizy: przychodnia

- ⦿ Ten prosty scenariusz zawiera już wiele elementów informacyjnych, które możemy analizować:
  - ⦿ pacjent
  - ⦿ lekarz
  - ⦿ wizyta
  - ⦿ recepta
  - ⦿ lek
  - ⦿ apteka
- ⦿ Na kolejnych slajdach spróbujemy przekształcić ten scenariusz w **model systemu informacyjnego**.

# System do analizy: przychodnia

Który element jest uczestnikiem analizowanego systemu?

A pacjent

B lekarz

C apteka

D wszystkie wymienione

# Scenariusz działania systemu.

## Opis scenariusza systemu.

- ⦿ Pacjent chce skorzystać z wizyty lekarskiej.
  - ⦿ zadzwonić do rejestracji
  - ⦿ zarejestrować się online
- ⦿ System musi obsłużyć proces rejestracji.
- ⦿ Zaczynamy od bardzo prostego pytania:
  - ⦿ Co właściwie dzieje się w tym systemie?
- ⦿ Każdy system informatyczny obsługuje pewien **proces rzeczywisty**.

# Scenariusz działania systemu.

## Opis scenariusza systemu.

- ◉ W naszym przypadku jest to proces:
  - ◉ **rejestracji pacjenta na wizytę lekarską i dalszej obsługi tej wizyty**
- ◉ Pierwszym krokiem jest sytuacja wyjściowa:
  - ◉ Pacjent chce skorzystać z usługi medycznej.
- ◉ Może to zrobić na dwa sposoby:
  - ◉ telefonicznie
  - ◉ przez Internet
- ◉ Już na tym etapie widzimy, że system musi obsługiwać:
  - ◉ dane pacjenta
  - ◉ dane lekarzy
  - ◉ dostępność terminów
- ◉ To pierwszy krok do identyfikacji elementów systemu.

# Scenariusz działania systemu. Opis scenariusza systemu.

Który element rozpoczyna działanie systemu?

- A lekarz
- B system informatyczny
- C pacjent
- D apteka

# Rejestracja wizyty



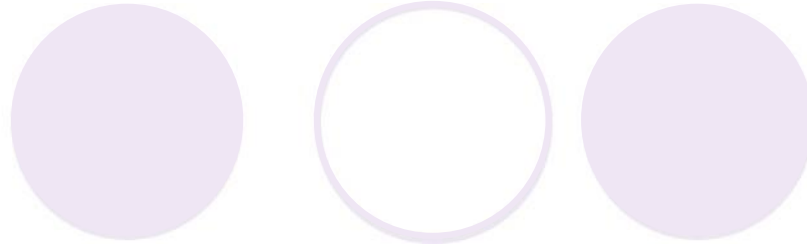
- ◉ Rejestracja:
  - ◉ wyszukuje lekarza
  - ◉ sprawdza dostępne terminy
  - ◉ zapisuje pacjenta na wizytę
- ◉ Tworzony jest z
- ◉ Po zgłoszeniu pacjenta następuje etap **rejestracji wizyty**.
- ◉ Może to być:
  - ◉ pracownik rejestracji
  - ◉ system informatyczny (rejestracja online)
  - ◉ zapis wizyty.

# Rejestracja wizyty



- ◉ System musi:
  - ◉ znaleźć odpowiedniego lekarza
  - ◉ sprawdzić dostępne terminy
  - ◉ przypisać pacjenta do konkretnej wizyty
- ◉ W tym momencie powstaje bardzo ważny element systemu:
  - ◉ **wizyta**
- ◉ To jest pierwszy obiekt, który naturalnie wynika z analizy procesu.

# Rejestracja wizyty



Jaki element powstaje w wyniku rejestracji?

- A pacjent
- B lekarz
- C wizyta
- D lek

# Wizyta u lekarza



- ◉ Podczas wizyty:
  - ◉ lekarz bada pacjenta
  - ◉ wprowadza rozpoznanie
  - ◉ zapisuje zalecenia
- ◉ Powstają nowe dane medyczne.
- ◉ Kolejnym etapem procesu jest **wizyta lekarska**.
- ◉ To moment, w którym system zaczyna gromadzić **nowe informacje**.

# Wizyta u lekarza



- ◉ Podczas wizyty:
  - ◉ lekarz analizuje stan pacjenta
  - ◉ podejmuje decyzję medyczną
  - ◉ zapisuje rozpoznanie
- ◉ Te dane są bardzo ważne, ponieważ:
  - ◉ są powiązane z pacjentem
  - ◉ są powiązane z konkretną wizytą
  - ◉ mogą być wykorzystywane w przyszłości
- ◉ Widzimy więc, że dane w systemie zaczynają tworzyć **relacje**.

# Wizyta u lekarza



Który element jest tworzony podczas wizyty?

- A tabela danych
- B struktura systemu
- C informacje medyczne o pacjencie
- D program komputerowy

# Wystawienie recepty



- ◉ Lekarz może wystawić:
  - ◉ receptę
  - ◉ skierowanie
  - ◉ zalecenia
- ◉ Recepta zawiera informacje o lekach.
- ◉ Po zakończeniu wizyty lekarz może podjąć decyzję o leczeniu.
- ◉ Najczęściej oznacza to wystawienie:
  - ◉ recepty
  - ◉ skierowania na badania
  - ◉ zaleceń

# Wystawienie recepty



- ◉ W naszym systemie skupimy się na **receptcie**, ponieważ jest to bardzo dobry przykład obiektu systemu.
- ◉ Recepta:
  - ◉ ma swój identyfikator
  - ◉ jest powiązana z wizytą
  - ◉ zawiera listę leków
- ◉ To kolejny element, który będzie później reprezentowany w bazie danych.

# Wystawienie recepty



Z czym bezpośrednio powiązana jest recepta?

- A apteką
- B pacjentem
- C lekarzem
- D wizytą lekarską

# Realizacja recepty



- ⦿ Pacjent realizuje receptę w aptece.
- ⦿ Apteka:
  - ⦿ identyfikuje receptę
  - ⦿ wydaje lek
  - ⦿ rejestruje realizację
- ⦿ Ostatnim etapem naszego scenariusza jest **realizacja recepty w aptece.**
- ⦿ Pacjent udaje się do apteki, gdzie:
  - ⦿ recepta jest identyfikowana
  - ⦿ farmaceuta wydaje odpowiednie leki

# Realizacja recepty



- ◉ System może zapisywać:
  - ◉ czy recepta została zrealizowana
  - ◉ jakie leki zostały wydane
  - ◉ kiedy nastąpiła realizacja
- ◉ Widzimy więc, że system nie kończy się na przychodni - obejmuje także **inne instytucje**, takie jak apteka.
- ◉ To pokazuje, że systemy informatyczne często obejmują **wiele powiązanych elementów rzeczywistości**.

# Realizacja recepty



Który element realizuje ostatni etap procesu?

- A lekarz
- B rejestracja
- C pacjent
- D apteka

# Elementy systemu



- ⦿ W systemie występują:
  - ⦿ pacjent
  - ⦿ lekarz
  - ⦿ wizyta
  - ⦿ recepta
  - ⦿ lek
  - ⦿ apteka
- ⦿ To są podstawowe **obiekty systemu**.

# Elementy systemu



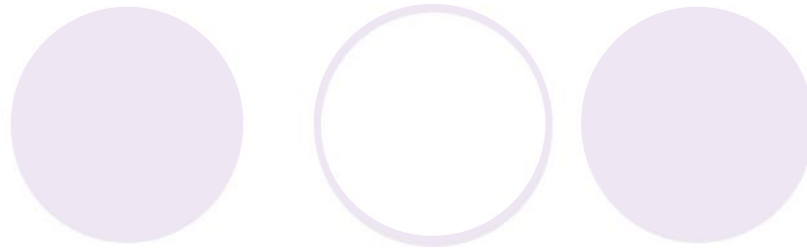
- ◉ Podsumujmy nasz scenariusz.
- ◉ Z prostego opisu procesu udało nam się wyodrębnić kluczowe elementy systemu:
  - ◉ pacjent
  - ◉ lekarz
  - ◉ wizyta
  - ◉ recepta
  - ◉ lek
  - ◉ apteka
- ◉ To właśnie te elementy nazywamy:
  - ◉ **obiektami systemu**
- ◉ To bardzo ważny moment wykładu.

# Elementy systemu



- ◉ Co trzeba zrozumieć i zapamiętać:
  - ◉ Obiekty systemu nie są wymyślane — one wynikają bezpośrednio z analizy rzeczywistości.
- ◉ To dokładnie ten moment, w którym przechodzimy od opisu słownego do **modelu danych**.

# Elementy systemu



Jak nazywamy elementy takie jak pacjent, lekarz, wizyta?

A rekordy

B zapytania

C tabele

D obiekty systemu

# Analiza obiektowa.

## Obiekt w systemie informatycznym.

Cel: przejście od scenariusza → do **obiektów i ich atrybutów**

- ◉ **Obiekt** to element rzeczywistości opisany w systemie informatycznym.

Przykłady:

- ◉ pacjent
- ◉ lekarz
- ◉ wizyta
- ◉ recepta

# Analiza obiektowa.

## Obiekt w systemie informatycznym.

- ◉ W analizie systemów jednym z najważniejszych pojęć jest **obiekt**.
- ◉ Obiekt to coś, co istnieje w rzeczywistości i co chcemy opisać w systemie informatycznym.
- ◉ W naszym przykładzie przychodni obiektami są:
  - ◉ pacjent
  - ◉ lekarz
  - ◉ wizyta
  - ◉ recepta

# Analiza obiektowa.

## Obiekt w systemie informatycznym.

- ◉ Bardzo ważne:
  - ◉ Obiekty nie są tworzone sztucznie — one wynikają z analizy rzeczywistości.
  - ◉ Każdy obiekt będzie później reprezentowany w bazie danych jako:
    - ◉ **tabela (opisane encje)**

# Analiza obiektowa. Obiekt w systemie informatycznym.

Który z poniższych elementów jest obiektem systemu?

A pacjent

B lekarz

C wizyta

D **wszystkie wymienione**

# Identyfikacja obiektów systemu

- ◉ Obiekty w systemie przychodni:
  - ◉ Pacjent
  - ◉ Lekarz
  - ◉ Wizyta
  - ◉ Recepta
  - ◉ Lek
  - ◉ Apteka
- ◉ Na podstawie naszego scenariusza wyodrębniliśmy listę obiektów.

# Identyfikacja obiektów systemu

- ◉ To bardzo ważny moment:
  - ◉ tworzymy „listę rzeczy”, które będą zapisane w systemie
- ◉ Jeżeli zapomnimy o jakimś obiekcie:
  - ◉ system będzie niekompletny
  - ◉ baza danych będzie błędna
- ◉ Dlatego analiza musi być dokładna.

# Identyfikacja obiektów systemu

Który element NIE jest obiektem w naszym systemie?

- A pacjent
- B lekarz
- C recepta
- D system operacyjny

# Atrybuty obiektu (Pacjent)

- ◉ Pacjent posiada atrybuty:
  - ◉ imię
  - ◉ nazwisko
  - ◉ PESEL
  - ◉ adres
  - ◉ Telefon
- ◉ Każdy obiekt musi być **opisany danymi**.
- ◉ Te dane nazywamy:
  - ◉ **atrybutami**

# Atrybuty obiektu (Pacjent)

Przykład:

- ◉ Obiekt pacjent - atrybuty:
  - ◉ imię
  - ◉ nazwisko
  - ◉ PESEL
  - ◉ adres
- ◉ Bez atrybutów obiekt nie ma wartości informacyjnej.
- ◉ W bazie danych atrybuty staną się:
  - ◉ **kolumnami tabeli**

# Atrybuty obiektu (Pacjent)



Jak nazywamy cechy opisujące obiekt?

- A relacje
- B rekordy
- C atrybuty
- D zapytania

# Atrybuty obiektu (Lekarz)

- ◉ Lekarz posiada atrybuty:
  - ◉ imię
  - ◉ nazwisko
  - ◉ specjalizacja
  - ◉ numer PWZ
  - ◉ Telefon
- ◉ Analogicznie jak pacjent — każdy obiekt ma swoje atrybuty.

# Atrybuty obiektu (Lekarz)

- ◉ Lekarz:
  - ◉ ma dane osobowe
  - ◉ ma specjalizację
  - ◉ posiada numer prawa wykonywania zawodu
- ◉ To bardzo ważne:
  - ◉ różne obiekty mają różne zestawy atrybutów
- ◉ Nie można mieszać danych różnych obiektów w jednej strukturze.
- ◉ To pozwala zrozumieć:
  - ◉ dlaczego tworzymy osobne tabele

# Atrybuty obiektu (Lekarz)



Który atrybut najlepiej opisuje lekarza?

- A PESEL pacjenta
- B data wizyty
- C nazwa leku
- D specjalizacja

# Atrybuty obiektu (Wizyta)

- ◉ Wizyta posiada atrybuty:
  - ◉ data
  - ◉ godzina
  - ◉ pacjent
  - ◉ lekarz
  - ◉ Rozpoznanie
- ◉ Wizyta jest **bardzo ciekawym obiektem.**

# Atrybuty obiektu (Wizyta)

- ◉ Bo łączy inne obiekty:
  - ◉ pacjenta
  - ◉ lekarza
- ◉ To oznacza, że:
  - ◉ wizyta jest **punktem połączenia danych**
- ◉ W bazie danych będzie to bardzo ważna tabela, ponieważ:
  - ◉ pozwala powiązać pacjenta z lekarzem
  - ◉ przechowuje dane o zdarzeniu
- ◉ To jest pierwszy moment, gdzie studenci zaczynają rozumieć:
  - ◉ relacje między danymi

# Atrybuty obiektu (Wizyta)



Dlaczego wizyta jest ważnym obiektem?

A bo jest tabelą

B bo jest raportem

C bo jest programem

D bo łączy pacjenta i lekarza

# Atrybuty obiektu (Wizyta)



Dlaczego wizyta jest ważnym obiektem?

A bo jest tabelą

B bo jest raportem

C bo jest programem

D bo łączy pacjenta i lekarza

# Atrybuty obiektu (Recepta)

- ◉ Recepta posiada atrybuty:
  - ◉ numer recepty
  - ◉ data wystawienia
  - ◉ lekarz
  - ◉ lista leków

# Atrybuty obiektu (Recepta)

- ◉ Recepta to kolejny bardzo ważny obiekt.
- ◉ Jest powiązana z:
  - ◉ wizytą
  - ◉ lekarzem
  - ◉ pacjentem
- ◉ Dodatkowo zawiera:
  - ◉ **listę leków**

# Atrybuty obiektu (Recepta)

- ◉ I tu pojawia się bardzo ważne zagadnienie:
  - ◉ jeden obiekt może być powiązany z wieloma innymi obiektami
- ◉ To będzie później prowadzić do:
  - ◉ relacji jeden-do-wielu
  - ◉ i osobnych tabel (np. lista leków na recepcie)

# Atrybuty obiektu (Recepta)

Co może zawierać jedna recepta?

- A jeden lek
- B dwóch pacjentów
- C jednego lekarza
- D wiele leków

# Efekt dydaktyczny tego bloku

- ◉ Zrozumienie czym jest obiekt
  - ◉ Umiejętność wskazania obiektu systemu
  - ◉ Określenie czym są atrybuty
  - ◉ Zrozumienie powiązania między obiektami
- 
- ◉ Bardzo istotne umiejętności ponieważ to **must have** do zrozumienia relacji między obiektami (encjami)

# Efekt dydaktyczny tego bloku

- ◉ Zrozumienie czym jest obiekt
  - ◉ Umiejętność wskazania obiektu systemu
  - ◉ Określenie czym są atrybuty
  - ◉ Zrozumienie powiązania między obiektami
- 
- ◉ Bardzo istotne umiejętności ponieważ to **must have** do zrozumienia relacji między obiektami (encjami)

# Relacje między obiektami

## Czym jest relacja



- **Relacja** to powiązanie między obiektami w systemie.

Przykład:

pacjent → ma → wizyty

# Relacje między obiektami

## Czym jest relacja



- ◉ Do tej pory zidentyfikowaliśmy:
  - ◉ obiekty
  - ◉ atrybuty
- ◉ Ale system to nie tylko lista obiektów.
- ◉ Najważniejsze są **powiązania między nimi**.
- ◉ Te powiązania nazywamy:
  - ◉ **relacjami**

# Relacje między obiektami

## Czym jest relacja



- ◉ Relacje opisują, jak elementy systemu są ze sobą powiązane.
- ◉ Bez relacji dane nie mają sensu.

Przykład:

- ◉ sam pacjent → to tylko dane
- ◉ pacjent + wizyty → to już historia leczenia

# Relacje między obiektami

## Czym jest relacja



Co opisuje relacja w systemie?

A strukturę tabeli

B wygląd programu

C powiązania między obiektami

D szybkość działania systemu

# Relacja: Pacjent – Wizyta

- ◉ Pacjent → może mieć wiele wizyt
- ◉ Relacja: **1 : N (jeden do wielu)**
- ◉ Rozważmy pierwszą relację:
  - ◉ pacjent – wizyta
- ◉ Jeden pacjent:
  - ◉ może mieć jedną wizytę
  - ◉ może mieć wiele wizyt

# Relacja: Pacjent – Wizyta



- ◉ Ale każda wizyta dotyczy:
  - ◉ jednego konkretnego pacjenta
- ◉ To oznacza relację:
  - ◉ **jeden do wielu (1:N)**
- ◉ To jedna z najważniejszych relacji w bazach danych.

# Relacja: Pacjent – Wizyta



Ilu wizyt może mieć jeden pacjent?

- A jedną
- B dwie
- C żadnej
- D wiele

# Relacja: Lekarz – Wizyta

- ◉ Lekarz → przyjmuje wielu pacjentów
- ◉ Relacja: **1 : N**
- ◉ Druga relacja:
  - ◉ lekarz – wizyta
- ◉ Jeden lekarz:
  - ◉ przyjmuje wielu pacjentów
  - ◉ prowadzi wiele wizyt

# Relacja: Lekarz – Wizyta



- ◉ Ale jedna wizyta:
  - ◉ dotyczy jednego lekarza
- ◉ Znowu mamy relację:
  - ◉ **1 : N**
- ◉ Widzimy, że w systemach informatycznych takie relacje pojawiają się bardzo często.

# Relacja: Lekarz – Wizyta



Jedna wizyta może mieć:

- A wielu lekarzy
- B wielu pacjentów
- C wielu administratorów
- D jednego lekarza

# Relacja: Wizyta – Recepta

- ◉ Jedna wizyta:
  - ◉ może mieć receptę
  - ◉ może nie mieć recepty
- ◉ Relacja: **1 : 0..1**
- ◉ Teraz mamy ciekawszy przypadek.
  - ◉ wizyta – recepta
- ◉ Nie każda wizyta kończy się receptą.



# Relacja: Wizyta – Recepta

Czy każda wizyta musi mieć receptę?

A tak

B nie

C tylko w szpitalu

D tylko prywatnie

# Relacja: Recepta – Lek



- ◉ Jedna recepta: → może zawierać wiele leków
- ◉ Relacja: **1 : N**
- ◉ Kolejna relacja:
  - ◉ recepta – lek
- ◉ Jedna recepta:
  - ◉ może zawierać jeden lek
  - ◉ może zawierać wiele leków

# Relacja: Recepta – Lek



- ◉ To bardzo ważny przypadek, bo:
  - ◉ mamy listę elementów w jednym obiekcie
- ◉ W praktyce oznacza to:
  - ◉ potrzebujemy osobnej struktury danych (tabeli)
- ◉ To będzie bardzo ważne w następnym wykładzie.

# Relacja: Recepta – Lek



Ile leków może być na jednej receptce?

A jeden

B dwa

C maksymalnie trzy

D dowolnie wiele

# Relacja: Recepta – Lek



Ile leków może być na jednej receptce?

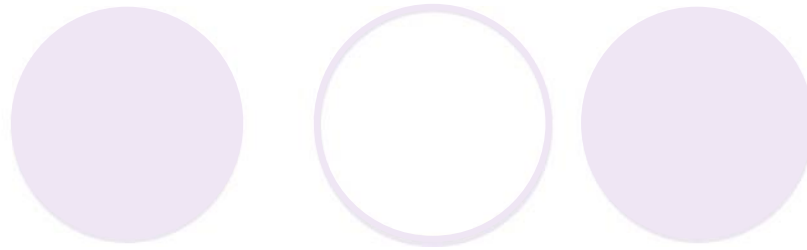
A jeden

B dwa

C maksymalnie trzy – w sumie zależy od lekarza

D dowolnie wiele

# Diagram relacji



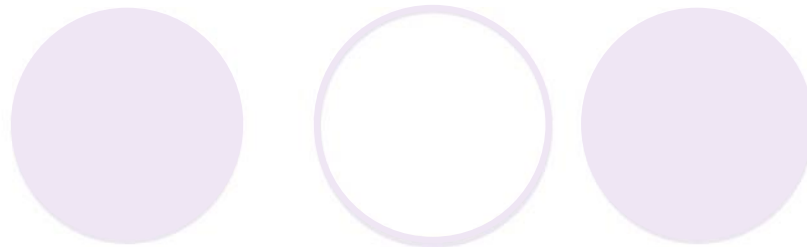
- ◉ Relacje w systemie:
  - ◉ Pacjent → Wizyta
  - ◉ Lekarz → Wizyta
  - ◉ Wizyta → Recepta
  - ◉ Recepta → Lek
- ◉ To moment, w którym warto popatrzeć na **całość systemu jako model**.
- ◉ Mamy:
  - ◉ obiekty
  - ◉ relacje między nimi

# Diagram relacji



- ◉ I zaczyna się pojawiać coś bardzo ważnego:
  - ◉ **struktura systemu danych**
- ◉ Można to zobrazować w postaci układu:
  - ◉ Pacjent → Wizyta ← Lekarz
  - ◉ Wizyta → Recepta → Lek
- ◉ To jest już prawie:
  - ◉ **model danych (ERD)**

# Diagram relacji



Który element łączy pacjenta i lekarza?

A recepta

B lek

C wizyta

D apteka

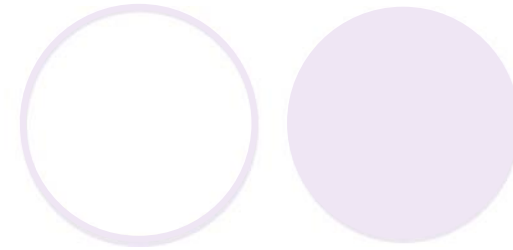
# Efekt dydaktyczny



- ◉ Zrozumienie czym jest relacja
- ◉ Identyfikacja i zrozumienie relacji 1:N
- ◉ Zrozumienie idei relacji opcjonalnej
- ◉ Obserwacja powiązań w systemie

# Model danych systemu

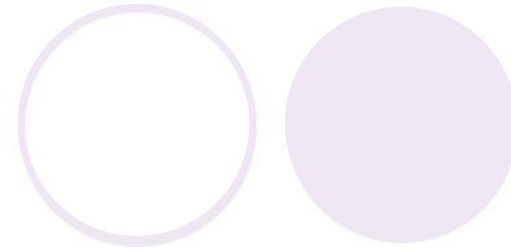
## Encje systemu



- ◉ Encje systemu:
  - ◉ Pacjent
  - ◉ Lekarz
  - ◉ Wizyta
  - ◉ Recepta
  - ◉ Lek
- ◉ Encja = obiekt zapisany w bazie danych

# Model danych systemu

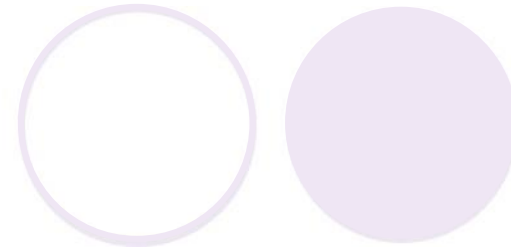
## Encje systemu



- ◉ W analizie używaliśmy pojęcia:
  - ◉ **obiekt**
- ◉ W bazach danych używamy pojęcia:
  - ◉ **encja**
- ◉ To w praktyce oznacza:
  - ◉ obiekt z analizy → staje się encją w bazie danych

# Model danych systemu

## Encje systemu

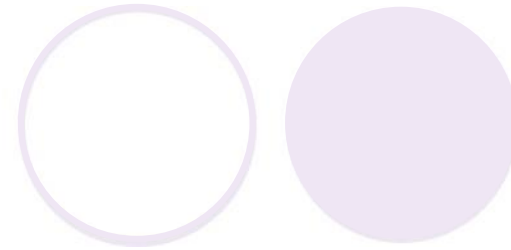


Przykład:

- obiekt: pacjent → encja: PACJENT (tabela)
- obiekt: lekarz → encja: LEKARZ (tabela)
  
- To bardzo ważne przejście:
  - zaczynamy myśleć w kategoriach bazy danych

# Model danych systemu

## Encje systemu



Czym jest encja w bazie danych?

A zapytaniem SQL

B relacją między tabelami

C raportem

D obiektem zapisanym w bazie danych

# Atrybuty encji



- ◉ Encja posiada:
  - ◉ identyfikator (ID)
  - ◉ zestaw atrybutów

## Przykład PACJENT:

- ◉ ID\_pacjenta
- ◉ imię
- ◉ nazwisko
- ◉ PESEL

# Atrybuty encji



- ◉ Każda encja musi mieć:
  - ◉ **identyfikator**
    - ◉ czyli coś, co jednoznacznie ją identyfikuje.
- ◉ To będzie później:
  - ◉ **klucz główny (Primary Key)**

# Atrybuty encji



Przykład:

- ◉ PACJENT:

- ◉ ID\_pacjenta
- ◉ imię
- ◉ nazwisko

- ◉ Dlaczego ID jest ważne?

- ◉ imię i nazwisko mogą się powtarzać
- ◉ ID jest unikalne

- ◉ To jest fundament projektowania baz danych.

# Atrybuty encji



Dlaczego używamy identyfikatora (ID)?

- A żeby skrócić nazwę tabeli
- B żeby przyspieszyć komputer
- C aby jednoznacznie zidentyfikować wpis w tabeli
- D aby zmniejszyć ilość danych

# Relacje między encjami



- ◉ Relacje między encjami:
  - ◉ Pacjent → Wizyta
  - ◉ Lekarz → Wizyta
  - ◉ Wizyta → Recepta
  - ◉ Recepta → Lek
- ◉ Relacje będą realizowane przez:
  - ◉ klucze obce

# Relacje między encjami

- ◉ To bardzo ważny moment.
- ◉ Relacje, które analizowaliśmy wcześniej, muszą zostać zapisane w bazie danych.
- ◉ Jak to robimy?
  - ◉ za pomocą **kluczy obcych (Foreign Keys)**
  - ◉ pozwalają łączyć encje ze sobą korzystając z definicji relacji

# Relacje między encjami



Przykład:

- ◉ Tabela WIZYTA zawiera:
  - ◉ ID\_pacjenta
  - ◉ ID\_lekarza
- ◉ To pozwala połączyć dane między tabelami.
- ◉ Czyli:
  - ◉ relacja z analizy → staje się relacją w bazie danych

# Relacje między encjami



Jak realizujemy relacje w bazie danych?

- A poprzez raporty
- B poprzez formularze
- C poprzez zapytania
- D poprzez klucze obce

# Model danych systemu



- ◉ Model danych zawiera:
  - ◉ encje
  - ◉ atrybuty
  - ◉ relacje
- ◉ Model danych = struktura bazy danych
- ◉ W tym momencie mamy już wszystko:
  - ◉ obiekty → encje
  - ◉ atrybuty → kolumny
  - ◉ relacje → klucze obce

# Model danych systemu



- ◉ To razem tworzy:
  - ◉ **model danych**
- ◉ Model danych to:
  - ◉ plan bazy danych
  - ◉ schemat systemu
- ◉ Bez modelu nie da się poprawnie stworzyć bazy danych.

# Model danych systemu



Z czego składa się model danych?

- A tylko z tabel
- B tylko z zapytań
- C tylko z relacji
- D z encji, atrybutów i relacji



# Przejdźcie do implementacji

- ◉ Na podstawie modelu danych tworzymy:
  - ◉ tabele
  - ◉ relacje
  - ◉ zapytania
  - ◉ formularze
  - ◉ raporty

# Przejs̄cie do implementacji

- ◉ To jest moment przej̄scia:
  - ◉ z analizy → do praktyki
- ◉ Na podstawie modelu danych możemy stworzȳc:
  - ◉ tabele w bazie danych
  - ◉ relacje między tabelami
  - ◉ zapytania (QBE, SQL)
  - ◉ formularze do wprowadzania danych
  - ◉ raporty
- ◉ To b̄dzie tematem kolejnego wykł̄adu.
  - ◉ analiza nie jest celem — jest podstaw̄ą implementacji

# Przejdźcie do implementacji

Co stworzymy bezpośrednio na podstawie modelu danych?

- A układ scenariuszowy
- B dobre wrażenie, dobrze jest zrobić analizę
- C niezbędną dokumentację
- D tabele i relacje w bazie danych