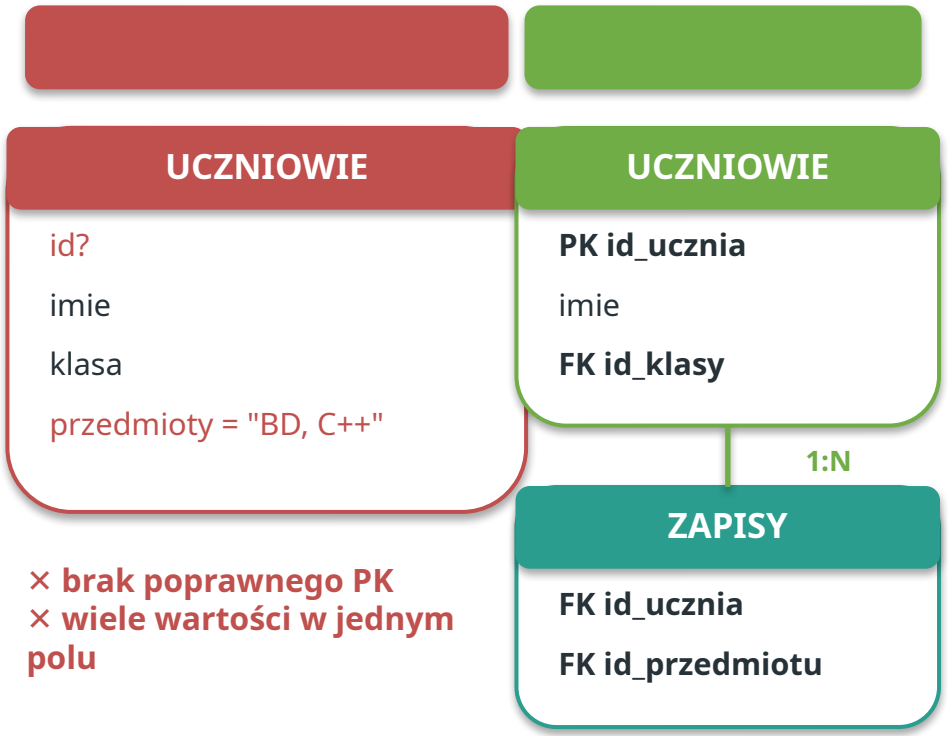
TEMAT: MODEL DANYCH

Audyt modelu danych

wyszukiwanie błędów projektowych

Jak sprawdzić, czy tabele, klucze i relacje naprawdę opisują problem - zanim zaczniemy pisać dużo SQL.



Bazy danych • 45 minut • teoria + analiza schematów + ćwiczenia na kartce

💬 **Audyt nie pyta: „czy działa?”, tylko: „czy model jest poprawny i logiczny?”**

1. Co to jest audyt modelu danych?

Audyt to spokojne sprawdzenie projektu przed budową bazy lub przed jej dalszym rozwojem.

1. Czy każda tabela ma sens?
Czy opisuje jeden typ obiektu lub zdarzenia?

2. Czy dane są we właściwym miejscu?
Czy nazwisko autora jest przy autorze, a nie powielone w każdym rekordzie książki?

3. Czy relacje są poprawne?
Czy związek jest 1:1, 1:N czy N:M? Czy klucz obcy stoi po właściwej stronie?

4. Czy baza broni spójności?
Czy PK, FK, UNIQUE i NOT NULL zapobiegają błędom, które da się przewidzieć?

Dobra baza danych nie tylko przechowuje informacje. Ona również utrudnia zapisanie informacji nielogicznych.

Przykład

Jeśli w tabeli ZAMÓWIENIA wpiszemy id_klienta = 99, a klient 99 nie istnieje, klucz obcy powinien taki zapis zablokować.

💬 **Czy audyt wykonujemy dopiero wtedy, gdy baza zaczyna zgłaszać błędy? Dlaczego wcześniej jest lepiej?**

2. Sześć punktów kontroli modelu

To nasza checklista na całą lekcję.

ENCJE / TABELE

Czy każda tabela opisuje jedną rzecz?

ATRYBUTY

Czy kolumny są potrzebne i atomowe?

PRIMARY KEY

Czy każdy rekord można jednoznacznie wskazać?

FOREIGN KEY

Czy powiązania są zapisane i chronione?

KROTNOŚCI

Czy 1:1, 1:N i N:M odpowiadają rzeczywistości?

REDUNDANCJA

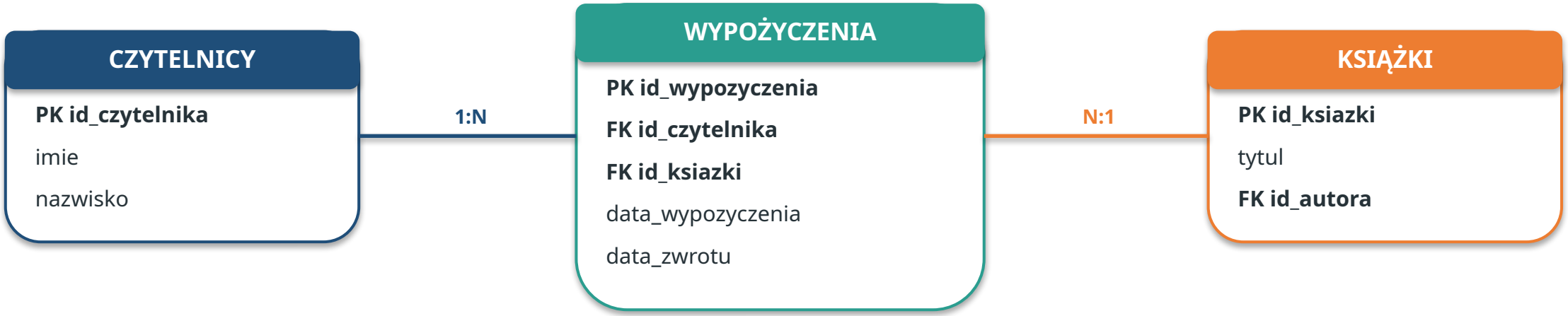
Czy tych samych danych nie wpisujemy wiele razy?

AUDYT = czytanie modelu + zadawanie pytań + próba znalezienia sytuacji, w której model zachowa się źle.

💬 **Który punkt kontroli sprawdza, czy dwóch uczniów nie dostanie tego samego identyfikatora?**

3. Najpierw zobaczmy prosty, poprawny model

Przykład: biblioteka. Trzy encje, dwa związki 1:N.



Jak to czytać?

Jeden czytelnik może mieć wiele wypożyczeń. Jedna książka może pojawić się w wielu wypożyczeniach w różnych terminach.

Co kontrolujemy?

Każde wypożyczenie wskazuje konkretnego czytelnika i konkretną książkę przez klucze obce.

💬 **Gdzie powinien znajdować się id_czytelnika: w CZYTELNICY czy w WYPOŻYCZENIA?**

4. Błąd nr 1 - brak dobrego klucza podstawowego

Bez jednoznacznego identyfikatora nie wiemy, który rekord naprawdę modyfikujemy.

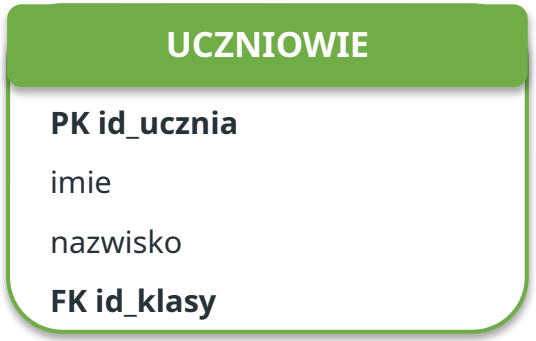
UCZNIOWIE - wersja błędna

imie	nazwisko	klasa
Jan	Kowalski	3A
Jan	Kowalski	3A
Ola	Nowak	3B

Problem

- dwa rekordy mogą wyglądać identycznie
- nazwisko nie jest dobrym PK - może się powtarzać
- trudno bezpiecznie wskazać jeden konkretny wiersz

Naprawa



Reguła

Dobry PK jest jednoznaczny, stabilny i nie zależy od cechy, która może się zmienić.

💬 **Czy PESEL zawsze byłby dobrym kluczem podstawowym? Jakie mogą być argumenty za i przeciw?**

5. Błąd nr 2 - kilka wartości w jednej komórce

Jedna komórka powinna przechowywać jedną logiczną wartość.

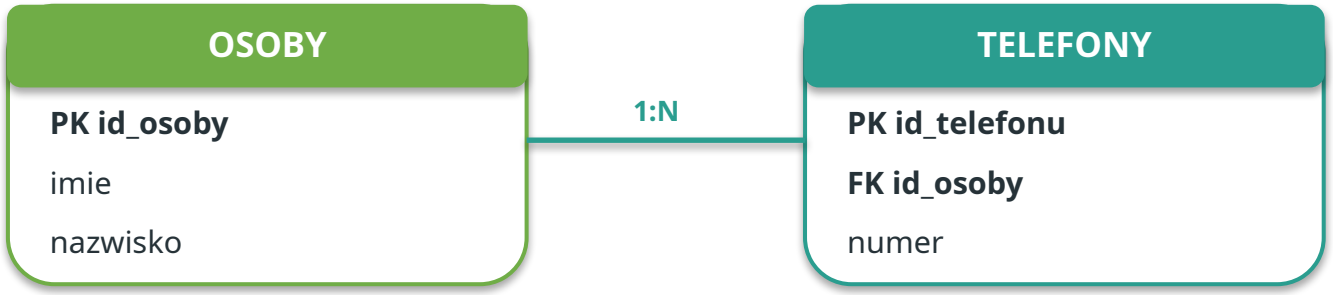
KONTAKTY - źle

id	osoba	telefon
1	Anna	600111222, 601222333
2	Tomek	602333444

Dlaczego to przeszkadza?

- trudniej wyszukiwać po jednym numerze
- trudniej sprawdzić unikalność numeru
- nie wiadomo, ile numerów ma rekord
- aktualizacja jednej wartości staje się niewygodna

Lepszy pomysł



Wniosek

Powtarzalną listę wartości często warto zamienić w osobną tabelę.

💬 **Jak przechować trzy adresy e-mail jednego klienta: trzy kolumny czy osobna tabela?**

6. Błąd nr 3 - powtarzanie tych samych danych

Redundancja = ta sama informacja zapisana wiele razy.

KSIĄŻKI - wersja z redundancją

id	tytul	autor	rok
1	Solaris	Stanisław Lem	1961
2	Cyberiada	Stanisław Lem	1965
3	Dune	Frank Herbert	1965

Co się stanie przy zmianie?

Jeśli poprawimy nazwisko autora w jednym rekordzie, a zapomnimy w drugim, baza zacznie zawierać dwie wersje tej samej informacji.

Korzyść

Dane autora zapisujemy raz. Książki tylko wskazują autora.

Rozdzielenie danych



💬 **Która zmiana jest bezpieczniejsza: poprawić nazwisko autora w 12 rekordach czy w jednym?**

7. Błąd nr 4 - brak lub zły klucz obcy

Klucz obcy powinien chronić relację, a nie być tylko „liczbą, która podobno coś oznacza”.

KLIENCI

id_klienta	nazwa
1	Alfa
2	Beta

ZAMÓWIENIA

id_zam	id_klienta	kwota
101	1	250
102	9	120

id_klienta = 9

Nie ma takiego klienta!

Jak baza może się bronić?

```
FOREIGN KEY (id_klienta)
REFERENCES
klienci(id_klienta);
```

Rekord osierocony

To rekord, który wskazuje na nieistniejący rekord nadrzędny. Poprawny FK powinien mu zapobiec.

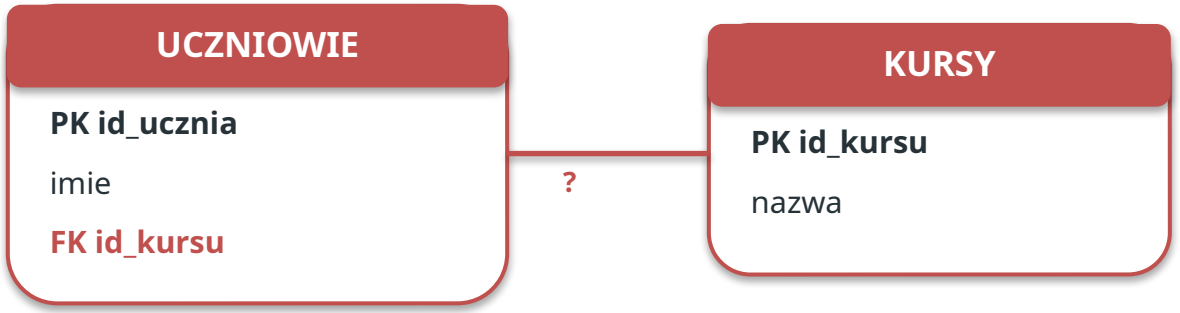
Audyt

Sprawdź nie tylko obecność kolumny id_klienta, ale czy naprawdę jest ona zdefiniowana jako FOREIGN KEY.

💬 **Czy sama nazwa kolumny id_klienta gwarantuje, że relacja jest chroniona?**

8. Błąd nr 5 - zła krotność relacji

Najczęstsza pułapka: modelujemy N:M tak, jakby było 1:N.



Co jest nie tak?

Kolumna id_kursu pozwala jednemu uczniowi wskazać tylko jeden kurs. A przecież uczeń może uczęszczać na kilka kursów.

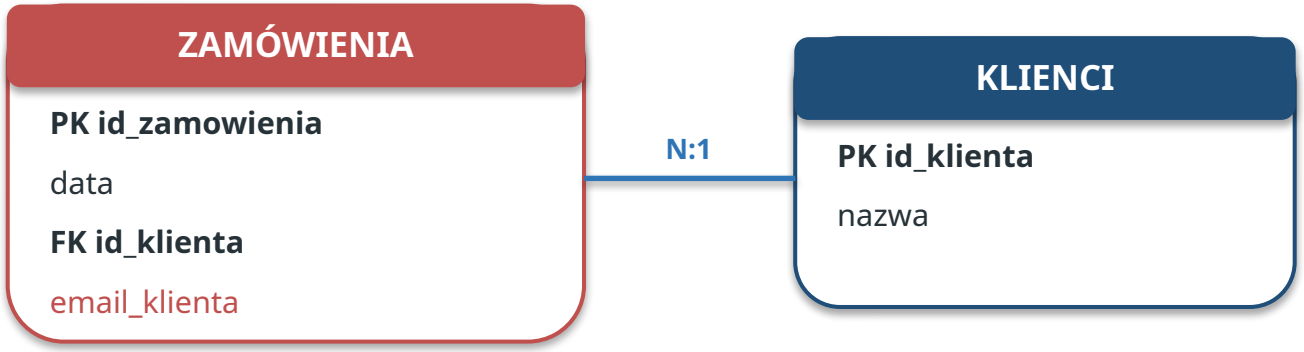
Poprawny model N:M



💬 **Jaką tabelę pośrednią zaproponujesz dla relacji UCZEŃ ↔️ PRZEDMIOT?**

9. Błąd nr 6 - atrybut w niewłaściwej tabeli

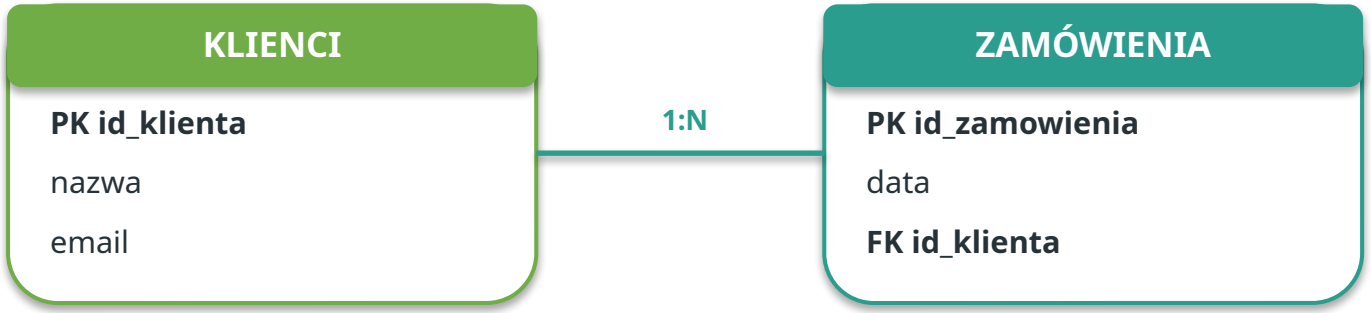
Pytamy: „do jakiego obiektu ta informacja naprawdę należy?”



Dlaczego email_klienta jest podejrzany?

Ten sam klient może mieć 20 zamówień. Email zostałby powtórzony 20 razy. Po zmianie adresu trzeba byłoby poprawiać wiele rekordów.

Po audycie



Prosta reguła

Atrybut umieszczamy przy encji, którą opisuje.

💬 **Gdzie powinien znajdować się adres szkoły: w UCZNIOWIE czy w SZKOŁY?**

10. Błąd nr 7 - nieprzemyślane NULL i obowiązkowość danych

Diagram może mówić, że informacja jest obowiązkowa albo opcjonalna. Tabela powinna to odzwierciedlać.

Przykład

PRACOWNIK może nie mieć jeszcze służbowego telefonu, ale musi mieć nazwisko.

PRACOWNICY

kolumna	NULL?	sens
nazwisko	NIE	obowiązkowe
telefon	TAK	może być brak
data_zatrudnienia	NIE	obowiązkowe

Jak zapisujemy to w tabeli?

```
nazwisko VARCHAR(60) NOT NULL,
telefon VARCHAR(20) NULL,
data_zatrudnienia DATE NOT NULL
```

Błąd projektowy

Ustawienie NOT NULL „wszędzie na wszelki wypadek” może uniemożliwić zapis poprawnego rekordu. Brak NOT NULL tam, gdzie dane są wymagane, osłabia kontrolę.

💬 **Czy data_zwrotu w tabeli WYPOŻYCZENIA powinna być NOT NULL w chwili wypożyczenia książki?**

11. Błąd nr 8 - zły typ danych lub zbyt szeroka domena

Typ danych powinien odpowiadać temu, co naprawdę zapisujemy.

Porównanie

kolumna	podejrzany typ	lepszy typ	dlaczego
wiek	VARCHAR(100)	INT / data_urodzenia	tekst utrudnia liczenie
cena	VARCHAR	DECIMAL(10,2)	trudne sumowanie i porównania
data	VARCHAR	DATE	trudne sortowanie po dacie
aktywny	VARCHAR	BOOLEAN	za dużo możliwych wartości

Domena

Typ to nie wszystko. Czasem warto dodatkowo ograniczyć dozwolone wartości, np. ocena od 1 do 6 albo przebieg >= 0.

Pytanie audytora

„Jakie wartości są poprawne, a jakie powinny zostać odrzucone?”

💬 **Czy numer telefonu powinien mieć typ INT? Co stanie się z zerem na początku albo znakiem +?**

12. Audyt przykładowego modelu - ile błędów znajdziesz?

Nie poprawiaj od razu. Najpierw nazwij problem i wyjaśnij, dlaczego jest problemem.

UCZNIOWIE
nazwisko
imie
klasa
przedmioty = "BD,C++"

KLASY
PK nazwa_klasy
wychowawca
email_wychowawcy

OCENY
PK id_oceny
uczen_nazwisko
przedmiot
ocena VARCHAR

NAUCZYCIELE
PK id_nauczyciela
imie
nazwisko
przedmioty

Podejrzane miejsca

- brak PK w UCZNIOWIE
- lista przedmiotów w jednej kolumnie
- powiązania zapisane tekstem zamiast FK
- ocena jako tekst
- dane wychowawcy powielane przy klasie
- niejasna relacja nauczyciel-przedmiot

Zadanie na 3 minuty

W parach wybierzcie 3 błędy. Dla każdego zapiszcie: 1) co jest źle, 2) jaki może być skutek, 3) jak to naprawić.

💬 **Który błąd w tym modelu najbardziej utrudniłby późniejsze tworzenie relacji?**

13. Prosty SQL może pomóc znaleźć błędy

Na lekcji nie wykonujemy zapytań - uczymy się przewidywać, co mogą ujawnić.

Zapytanie 1 - duplikaty

```
-- Czy są powtarzające się numery rejestracyjne?
SELECT nr_rejestracyjny, COUNT(*)
FROM pojazdy
GROUP BY nr_rejestracyjny
HAVING COUNT(*) > 1;
```

Zapytanie 2 - rekordy osierocone

```
-- Czy istnieją zamówienia bez klienta?
SELECT z.id_zamowienia
FROM zamowienia z
LEFT JOIN klienci k
  ON z.id_klienta = k.id_klienta
WHERE k.id_klienta IS NULL;
```

Jak czytać wynik 1?

Jeśli pojawią się wiersze, warto sprawdzić, czy numer rejestracyjny powinien mieć UNIQUE.

Jak czytać wynik 2?

Jeśli pojawią się zamówienia, relacja klient-zamówienie może być niespójna albo brakowało FK.

💬 **Czy puste wyniki tych zapytań zawsze oznaczają, że cały model jest poprawny?**

14. Przewidujemy wynik zapytania - bez komputera

Dane pomagają zobaczyć, jak błąd modelu ujawnia się w wyniku.

KLIENCI

id	nazwa
1	Alfa
2	Beta

ZAMÓWIENIA

id	id_klienta	kwota
101	1	200
102	9	300
103	2	150

Który wiersz zostanie?

```
SELECT z.id, z.id_klienta
FROM zamowienia z
LEFT JOIN klienci k
 ON z.id_klienta = k.id
WHERE k.id IS NULL;
```

101

102

103

żaden

Wyjaśnienie

Szukamy zamówienia, którego id_klienta nie ma w tabeli KLIENCI. To rekord 102 z id_klienta = 9.

💬 **Co powinno zostać zmienione w modelu, aby taki rekord nie mógł powstać?**

15. Krótkie ćwiczenie - audyt w 5 krokach

Pracujemy na kartce. Nie trzeba pisać pełnego SQL.

REZERWACJE

id_rezerwacji

klient_email

pokoj_nr

data_od

data_do

uslugi = "spa,parking"

Krok 1

Wskaż, co w tej tabeli wygląda podejrzanie.

Krok 2

Zaproponuj brakujące encje / tabele.

Krok 3

Wskaż PK każdej tabeli.

Krok 4

Dodaj relacje i klucze obce.

Krok 5

Określ krotności: 1:N czy N:M?

Bonus

Które pola powinny być UNIQUE lub NOT NULL?

Możliwy kierunek rozwiązania:

KLIENCI → REZERWACJE ← POKOJE

REZERWACJE ↔ USŁUGI przez tabelę REZERWACJE_USŁUGI

💬 **Który element tego modelu wymaga tabeli pośredniej i dlaczego?**

16. Podsumowanie - checklista audytora

Zanim zaakceptujesz model, przejdź po tej liście od góry do dołu.

1	Każda tabela opisuje jeden obiekt lub zdarzenie	✓
2	**Każdy rekord ma poprawny PRIMARY KEY**	✓
3	W jednej komórce jest jedna logiczna wartość	✓
4	Powtarzające się dane są wydzielone do właściwych tabel	✓
5	**Relacje mają poprawne FOREIGN KEY i krotności**	✓
6	NULL, UNIQUE, typy danych i zakres wartości są przemyślane	✓

Najważniejsza myśl: model nie jest poprawny dlatego, że wygląda ładnie. Musi poprawnie odwzorowywać dane i ich zależności.

Exit ticket - 1 minuta

Napisz jedną rzecz, którą zawsze sprawdzisz w modelu danych, zanim uznasz go za poprawny.