TEMAT 10

Trzecia postać normalna (3NF).

Bazy danych • klasa 5A • 45 minut

Cel lekcji

EFEKT	Rozpoznaje zależności przechodnie i stosuje 3NF.

Po lekcji potrafisz wyjaśnić pojęcie, odczytać schemat i uzasadnić swoją odpowiedź na przykładzie.

Rozgrzewka: co już pamiętamy?

Odpowiedz ustnie. Nie korzystamy z komputerów.

PYTANIE 1	Co w tabeli oznacza jeden wiersz?
PYTANIE 2	Po czym rozpoznajemy jeden konkretny rekord?
PYTANIE 3	Jak baza łączy dane z dwóch tabel?

Najważniejsze pojęcia

1. 3NF wymaga 2NF i braku zależności przechodnich.
2. Atrybut niekluczowy powinien zależeć bezpośrednio od klucza.
3. Dane miasta oddzielamy, gdy zależą od kodu pocztowego.

Prosty przykład danych

klient_id PK	kod_pocztowy	miasto
1	60-101	Poznań
2	60-101	Poznań

Czytamy tabelę od nagłówków kolumn, potem analizujemy pojedyncze rekordy.

Jak czytać ten przykład

KROK 1	Nazwij obiekt opisywany przez tabelę.
KROK 2	Sprawdź znaczenie każdej kolumny.
KROK 3	Przeczytaj jeden rekord jako pełną informację.
KROK 4	Poszukaj klucza, zależności lub błędu.

Schemat pojęcia

3NF wymaga 2NF i braku zależności przechodnich.

Atrybut niekluczowy powinien zależeć bezpośrednio od klucza.

Dane miasta oddzielamy, gdy zależą od kodu pocztowego.

Przykład zapisu SQL lub modelu

```
SELECT miasto FROM klienci WHERE klient_id=1;
```

Najpierw rozpoznaj słowo kluczowe. Następnie wskaż tabelę, kolumnę lub warunek. Na końcu przewidź skutek.

Co oznacza zapis?

1	Rozpoznaj rodzaj operacji.
2	Wskaż obiekt, którego dotyczy zapis.
3	Ustal, które dane zostaną odczytane lub zmienione.
4	Sprawdź, czy zapis nie łamie reguł bazy.

Przewidujemy wynik

```
SELECT miasto FROM klienci WHERE klient_id=1;
```

Zanim zobaczysz odpowiedź, wskaż rekordy lub opisz zmianę, którą spowoduje ten zapis.

Uzasadnij odpowiedź jednym pełnym zdaniem.

Typowe błędy

BŁĄD 1	Mylenie rekordu z kolumną.
BŁĄD 2	Odpowiedź bez wskazania klucza lub zależności.
BŁĄD 3	Czytanie SQL od końca zamiast od słów kluczowych.
SPRAWDŹ	Czy wynik pasuje do danych w tabeli?

Pytanie do klasy

Czy miasto zależy od klient_id bezpośrednio?

Najpierw odpowiedź. Potem uzasadnienie na podstawie tabeli lub schematu.

Krótkie ćwiczenie

Zaproponuj tabele Klienci i Kody_pocztowe.

Czas: 4 minuty • forma: samodzielnie na kartce • potem porównanie w parach

W odpowiedzi użyj poprawnych pojęć bazodanowych.

Omówienie ćwiczenia

SPRAWDŹ	Czy odpowiedź wskazuje właściwy obiekt?
SPRAWDŹ	Czy użyto właściwej nazwy pojęcia?
SPRAWDŹ	Czy uzasadnienie wynika z danych?
POPRAW	Zmień jedno zdanie tak, aby było precyzyjne.

Podsumowanie

1. 3NF wymaga 2NF i braku zależności przechodnich.
2. Atrybut niekluczowy powinien zależeć bezpośrednio od klucza.
3. Dane miasta oddzielamy, gdy zależą od kodu pocztowego.

Zdanie kończące: „Najważniejsze w tym temacie jest…”

Bilet wyjścia

1 MINUTA	Zapisz jedno pojęcie i własny przykład.
ODPOWIEDŹ	Wyjaśnij, dlaczego przykład jest poprawny.
PYTANIE	Co wymaga jeszcze powtórzenia?

Następna lekcja zacznie się od krótkiego powrotu do odpowiedzi.

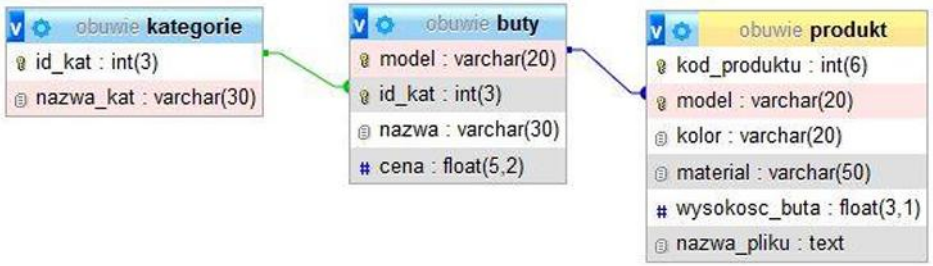
Ćwiczenie praktyczne: baza obuwie

Po części teoretycznej przechodzimy do pracy w phpMyAdmin.

CEL	Utworzysz znormalizowaną bazę, sprawdzisz relacje między tabelami i wykonasz zapytania SQL.
CZAS	Praca w parach lub samodzielnie • około 45–60 minut
EFEKT	Na koniec oddajesz plik kwerendy.txt oraz cztery zrzuty ekranu wyników.

Etap 1: baza i tabele

W phpMyAdmin otwórz kartę SQL. Utwórz bazę obuwie z kodowaniem polskich znaków, a następnie tabele zgodnie ze schematem.



Ilustracja 1. Baza danych

```
CREATE DATABASE obuwie
  CHARACTER SET utf8
  COLLATE utf8_unicode_ci;

USE obuwie;
```

Zwróć uwagę na 3NF: kategoria znajduje się w osobnej tabeli, a produkt łączy się z butem przez model.

Jeżeli tworzysz tabele ręcznie, najpierw utwórz kategorie, potem buty, a na końcu produkt.

Etap 2: tabele i kwerendy SELECT

STRUKTURA

```
CREATE TABLE kategorie (
  id_kat INT PRIMARY KEY,
  nazwa_kat VARCHAR(30) NOT NULL
);

CREATE TABLE buty (
  model VARCHAR(20) PRIMARY KEY,
  id_kat INT NOT NULL,
  nazwa VARCHAR(30) NOT NULL,
  cena DECIMAL(5,2) NOT NULL,
  FOREIGN KEY (id_kat) REFERENCES kategorie(id_kat)
);
```

KWERENDY

```
-- kw1: kolor i materiał
SELECT kolor, material
FROM produkt
WHERE wysokosc_buta > 10;

-- kw2: nazwa, cena i kategoria
SELECT b.nazwa, b.cena, k.nazwa_kat
FROM buty b
JOIN kategorie k ON k.id_kat = b.id_kat;
```

Do tabeli produkt dodaj pola z rysunku, w tym klucz kod_produktu oraz klucz obcy model.

Zapisz oba zapytania w pliku kwerendy.txt jako kw1 i kw2.

Etap 3: konto użytkownika i uprawnienia

Wykonaj zapytania jako administrator serwera MySQL. Konto Marek ma pracować tylko na tabeli produkt w bazie obuwie.

KW3

```
CREATE USER 'Marek'@'localhost'
IDENTIFIED BY 'M@reK';
```

KW4

```
GRANT SELECT, UPDATE
ON obuwie.produkt
TO 'Marek'@'localhost';
```

SELECT pozwala przeglądać dane. UPDATE pozwala je aktualizować. Konto nie otrzymuje praw do innych tabel ani do tworzenia struktury.

Jeżeli nauczyciel wymaga odświeżenia uprawnień, można wykonać: FLUSH PRIVILEGES;

Wynik pracy i kryteria oddania

PLIKI

kwerendy.txt
kw1.png
kw2.png
kw3.png
kw4.png

ZDJĘCIA

Każdy zrzut obejmuje cały ekran, pasek zadań oraz widoczny wynik działania odpowiedniej kwerendy.

SPRAWDŹ

Czy baza ma nazwę obuwie? Czy relacje wskazują właściwe klucze? Czy kw2 korzysta z JOIN? Czy Marek ma tylko SELECT i UPDATE na produkt?

Po wykonaniu zadania pokaż nauczycielowi bazę oraz komplet plików.