TEMAT 7

Anomalie danych i powtórka podstaw baz danych oraz SQL

Każde pojęcie: wyjaśnienie + przykład + zastosowanie

Bazy danych • klasa 5A • 45 minut

Cel lekcji i plan pracy

CEL	Rozpoznasz trzy anomalie i wyjaśnisz, dlaczego zły projekt tabeli powoduje błędy.
POWTÓRKA	Przypomnisz tabelę, rekord, pole, klucz, relację oraz polecenia SELECT, INSERT, UPDATE i DELETE.
METODA	Całą lekcję analizujemy na jednym przykładzie zapisów uczniów na kursy.

Najważniejsza zasada: odpowiedź zawsze uzasadniamy danymi widocznymi w tabeli.

Powtórka: tabela, kolumna i rekord

TABELA	KOLUMNA / POLE	WIERSZ / REKORD
Uporządkowany zbiór danych zapisany w wierszach i kolumnach.	Jedna cecha opisywanego obiektu. Nagłówek mówi, jakie dane zapisujemy.	Komplet danych o jednym obiekcie lub zdarzeniu.
PRZYKŁAD: UCZNIOWIE przechowują dane uczniów.	**PRZYKŁAD: email zawiera adres ucznia.**	**PRZYKŁAD: Wiersz 2 opisuje jeden zapis Piotra.**

Tabela, którą będziemy analizować

id_zapisu	uczen	email	kurs	nauczyciel	sala
1	Anna Kowalska	anna@szkola.pl	SQL	Jan Nowak	12
2	Piotr Lis	piotr@szkola.pl	SQL	Jan Nowak	12
3	Anna Kowalska	anna@szkola.pl	Python	Ewa Maj	15

Jeden rekord opisuje zapis ucznia na kurs. Dane ucznia i kursu powtarzają się w wielu rekordach.

PYTANIE: Które wartości widzisz więcej niż raz?

Klucz podstawowy i klucz obcy

KLUCZ PODSTAWOWY (PK)

Kolumna, której wartość jednoznacznie wskazuje jeden rekord. Nie może się powtarzać ani być pusta.

PRZYKŁAD: id_zapisu = 2 wskazuje tylko zapis Piotra na SQL.

KLUCZ OBCY (FK)

Kolumna, która przechowuje klucz podstawowy z innej tabeli i dzięki temu łączy rekordy.

PRZYKŁAD: id_ucznia = 1 w ZAPISACH wskazuje Annę w UCZNIACH.

W jednej dużej tabeli nie mamy jeszcze poprawnych kluczy obcych. To sygnał, że projekt wymaga podziału.

Encja, atrybut i relacja

ENCJA

Rodzaj obiektu, o którym przechowujemy dane.

PRZYKŁAD: UCZEŃ i KURS to dwie różne encje.

ATRYBUT

Cecha encji, która staje się kolumną tabeli.

PRZYKŁAD: email jest atrybutem ucznia, sala jest atrybutem kursu.

RELACJA

Powiązanie rekordów z dwóch tabel za pomocą kluczy.

PRZYKŁAD: Uczeń może zapisać się na wiele kursów.

Wniosek: jedna tabela miesza dane trzech obiektów: ucznia, kursu i zapisu.

Redundancja: ta sama informacja wiele razy

REDUNDANCJA

Niepotrzebne powtarzanie tej samej informacji w wielu rekordach.

uczen	kurs	nauczyciel	sala
Anna	SQL	Jan Nowak	12
Piotr	SQL	Jan Nowak	12

PRZYKŁAD: nauczyciel „Jan Nowak” i sala „12” występują dwa razy, chociaż opisują jeden kurs SQL.

Skutek: każdą zmianę trzeba wykonać w wielu miejscach.

Anomalia wstawiania

WYJAŚNIENIE

Nie możemy dodać informacji o jednym obiekcie bez podawania niepotrzebnych lub nieznanych danych o innym obiekcie.

PRZYKŁAD: Chcemy dodać kurs Java, ale nie ma jeszcze zapisanego ucznia.

```
INSERT INTO zapisy
(uczen, email, kurs, nauczyciel, sala)
VALUES (NULL, NULL, 'Java', 'Adam Kot', 18);
```

NULL oznacza brak znanej wartości. Tutaj puste dane ucznia wymusza zła konstrukcja tabeli.

PYTANIE: Dlaczego kurs powinien istnieć niezależnie od zapisów uczniów?

Anomalia aktualizacji

WYJAŚNIENIE

Tę samą informację zapisano w wielu rekordach. Jeżeli zmienimy tylko część z nich, baza pokaże sprzeczne dane.

PRZYKŁAD: Kurs SQL zmienia salę z 12 na 20, ale poprawiono tylko rekord Anny.

uczen	kurs	sala
Anna	SQL	20
Piotr	SQL	12

```
UPDATE zapisy SET sala = 20
WHERE uczen = 'Anna';
```

BŁĄD: ten sam kurs ma jednocześnie salę 20 i 12.

Anomalia usuwania

WYJAŚNIENIE

Usunięcie rekordu o jednym zdarzeniu powoduje utratę jedynej informacji o innym obiekcie.

PRZYKŁAD: Usuwamy ostatni zapis na Python i tracimy dane o kursie, nauczycielu oraz sali.

```
DELETE FROM zapisy
WHERE id_zapisu = 3;
```

Usuwamy zapis Anny na Python.

Razem z rekordem znika jedyna informacja: Python prowadzi Ewa Maj w sali 15.

Powtórka SQL: SELECT i WHERE

SELECT

Polecenie odczytuje dane. Po SELECT podajemy kolumny, a po FROM nazwę tabeli.

PRZYKŁAD: SELECT uczen, kurs FROM zapisy;

WHERE

Warunek wybiera tylko rekordy spełniające podane kryterium.

PRZYKŁAD: WHERE kurs = 'SQL' pozostawia zapisy na SQL.

```
SELECT uczen, kurs
FROM zapisy
WHERE kurs = 'SQL';
```

Powtórka SQL: INSERT, UPDATE i DELETE

INSERT	UPDATE	DELETE
Dodaje nowy rekord do tabeli.	Zmienia wartości w istniejących rekordach. WHERE wskazuje, które rekordy zmienić.	Usuwa rekordy. Bez WHERE polecenie usunęłoby wszystkie rekordy tabeli.
PRZYKŁAD: INSERT INTO kursy (nazwa) VALUES ('Java');	**PRZYKŁAD: UPDATE kursy SET sala=20 WHERE id_kursu=1;**	**PRZYKŁAD: DELETE FROM zapisy WHERE id_zapisu=3;**

Zasada bezpieczeństwa: przed UPDATE lub DELETE najpierw wykonaj SELECT z takim samym warunkiem WHERE.

Naprawa: rozdzielamy różne encje

UCZNIOWIE	KURSY	ZAPISY
id_ucznia PK	id_kursu PK	id_zapisu PK
imie_nazwisko	nazwa	id_ucznia FK
email	nauczyciel	id_kursu FK
	sala	

Każdy fakt zapisujemy tylko raz: dane ucznia w UCZNIACH, dane kursu w KURSACH, powiązanie w ZAPISACH.

PRZYKŁAD: zmiana sali kursu SQL wymaga teraz poprawienia jednego rekordu w tabeli KURSY.

Jak relacje łączą poprawne tabele

UCZNIOWIE
id_ucznia PK
imie_nazwisko
email

ZAPISY
id_zapisu PK
id_ucznia FK
id_kursu FK

KURSY
id_kursu PK
nazwa
nauczyciel, sala

1 uczeń może mieć wiele zapisów

1 kurs może mieć wiele zapisów

Tabela ZAPISY rozwiązuje relację wiele-do-wielu między uczniami i kursami.

Ćwiczenie: znajdź problem w tabeli

zamowienie_id	klient	telefon	produkt	cena
101	Adam Nowak	500 111 222	Laptop	3000
102	Adam Nowak	500 111 222	Mysz	80
103	Ewa Lis	600 222 333	Laptop	3000

1. Wskaż dane, które się powtarzają. 2. Podaj przykład każdej anomalii. 3. Zaproponuj tabele KLIENCI, PRODUKTY i ZAMÓWIENIA.

Podpowiedź: klient i produkt to różne encje. Zamówienie powinno łączyć ich kluczami obcymi.

Odpowiedzi do ćwiczenia

REDUNDANCJA	Adam i jego telefon występują dwa razy. Laptop i cena 3000 także występują dwa razy.
AKTUALIZACJA	Zmiana telefonu Adama w tylko jednym wierszu stworzy dwie różne wartości.
USUWANIE	Usunięcie ostatniego zamówienia produktu może usunąć jedyną informację o jego cenie.
WSTAWIANIE	Nie można dodać nowego produktu, dopóki nie istnieje zamówienie.
NAPRAWA	KLIENCI, PRODUKTY oraz ZAMÓWIENIA z kluczami PK i FK.

Słownik pojęć do odpowiedzi

HASŁO	WYJAŚNIENIE	PRZYKŁAD
rekord	jeden wiersz danych	zapis Anny na SQL
PK	jednoznacznie wskazuje rekord	id_zapisu = 1
FK	łączy rekord z inną tabelą	id_kursu w ZAPISACH
redundancja	powtarzanie tego samego faktu	sala 12 w wielu wierszach
anomalia	błąd wynikający ze złej struktury	sprzeczne sale kursu SQL

Jeśli nauczyciel pyta „dlaczego?”, wskaż konkretny rekord, powtórzoną wartość albo skutek polecenia SQL.

Podsumowanie i bilet wyjścia

1. Redundancja powtarza ten sam fakt w wielu rekordach.
2. Anomalie wstawiania, aktualizacji i usuwania powstają przez złą strukturę tabeli.
3. Podział encji na osobne tabele ogranicza błędy.
4. Klucze podstawowe i obce pozwalają bezpiecznie łączyć dane.

BILET WYJŚCIA: Wybierz jedną anomalię, wyjaśnij ją własnymi słowami i podaj przykład.