

Podsumowanie wykładu BD

Plan wykładu

- Modelowanie koncepcyjne – diagramy ER
- Relacyjny model danych – relacja, krotka, atrybut, ograniczenia integralnościowe
- Elementy języka SQL i T-SQL
- Model fizyczny danych, indeksy, B⁺-drzewa
- Optymalizacja zapytań i strojenie bazy danych
- Przetwarzanie transakcyjne – własności ACID, poziomy izolacji, algorytm 2PL
- Odtwarzanie po awarii - UNDO/REDO
- Bezpieczeństwo baz danych

Baza danych ma za zadanie:

Składować

- **Model konceptualny - ERD**
- Model relacyjny: relacje, krotki, atrybuty
- Ograniczenia integralnościowe: klucz podstawowy, obcy, typu CHECK
- **Normalizacja schematu**
- Język DDL – create, alter, drop
- Organizacja fizyczna bazy danych

Przetwarzać

- Podstawy języka SQL – AR, RRK, RRD
- Język DML – select, insert, update, delete
- T-SQL – procedury, funkcje, triggery

Udostępniać

- **Optymalizacja zapytań!**
 - Indeksy
 - Statystyki
- Transakcje – **ACID**
- Operacje konfliktowe
- **Poziomy izolacji, anomalie**
- Blokady, **uszeregowalność**, 2PL, blokady intencjonalne

Chronić

- **Reakcja na awarie: UNDO/REDO**
- Uwierzytelnianie, uprawnienia, role

ERD -> model relacyjny

- Sklep internetowy:
 - Klient
 - Sprzedawca
 - Koszyk
 - Produkt
 - Kategorie produktu (hierarchia)
- Testy
 - Użytkownicy
 - Pytania
 - Wielokrotnego wyboru
 - Jednokrotnego wyboru
 - Odpowiedzi

Normalizacja

- Po co się normalizuje?
 - jakie anomalie wiążą się z nieznormalizowanym schematem?

Normalizacja

- Po co się normalizuje?
 - jakie anomalie wiążą się z nieznormalizowanym schematem?
- Który schemat jest znormalizowany?

VER. 1

imie	nazwisko	miasto
-----	-----	-----
alan	nowicki	poznań
zuza	galicka	poznań
tomek	byczkowski	kraków
karol	baranski	poznań
jola	lojalna	kraków

VER. 2

imie	nazwisko	miasto
-----	-----	-----
alan	nowicki	1
zuza	galicka	1
tomek	byczkowski	2
karol	baranski	1
jola	lojalna	2

id	miasto
----	-----
1	poznań
2	kraków

Normalizacja c.d.

- Zależności funkcyjne

- Znormalizuj schemat relacji:

$U = (A, B, C, D, E)$

$F = (AB \rightarrow C, A \rightarrow DE, B \rightarrow D, A \rightarrow B, E \rightarrow C)$

$U = (A, B, C, D)$

$F = (AB \rightarrow C, B \rightarrow C, A \rightarrow CD)$

- wyznacz klucz podstawowy
- czy schemat jest w 2PN? 3PN?
- sprowadź do 2PN a następnie do 3PN – twierdzenie o rozkładalności

Normalizacja c.d. - rozwiązanie metoda 1

- Znormalizuj schemat relacji:

$U = (A, B, C, D, E)$

$F = (AB \rightarrow C, A \rightarrow DE, B \rightarrow D, A \rightarrow B, E \rightarrow C)$

- Krok 1 – znajdź klucz

- Sprawdzam $A^+ = \{A, D, E, B, C\}$ – jest kluczem

- Jaka to postać normalna?

- 2PN – tak, bo klucz jednoatrybutowy (nie ma części)
- 3PN – nie, bo $B \rightarrow D, E \rightarrow C$

- Normalizuję do 3PN

- Wybierz zależność, która psuje 3PN: $B \rightarrow D$
- Według niej podziel na dwa schematy:

Normalizacja c.d. – rozwiązanie metoda 1

- $U1 = \{B, D\}$, $F1 = \{B \rightarrow D\}$, jest 3PN
- $U2 = \{B, A, C, E\}$, $F2 = \{AB \rightarrow C, A \rightarrow BE, E \rightarrow C\}$ – nie jest 3PN bo klucz: A, i istnieje zależność przejściowa $E \rightarrow C$
 - $U21 = \{E, C\}$, $F21 = \{E \rightarrow C\}$ – jest 3PN
 - $U22 = \{E, A, B\}$ $F22 = \{A \rightarrow BE\}$ - jest 3PN
- A więc schemat w 3PN: $U1, U21, U22$

Normalizacja c.d. – rozwiązanie metoda 1

- Znormalizuj schemat relacji:

$U = (A, B, C, D)$

$F = (AB \rightarrow C, B \rightarrow C, A \rightarrow CD)$

- Krok 1 – znajdź klucz

- $A^+ = \{A, C, D\}$

- $B^+ = \{B, C\}$

- $AB^+ = \{A, B, C, D\}$ – klucz

- Jaka to postać normalna?

- 2PN – nie, bo atrybuty niekluczowe zależą od *części* klucza, np. C zależy od B, D zależy od A

- 3PN – nie, bo nie jest w 2PN

Normalizacja c.d. – rozwiązanie metoda 1

- Sprowadź do 2PN
 - Wybierz zależność, która psuje 2PN i według niej podziel, np.: $A \rightarrow CD$
 - $U1 = \{A, C, D\}$, $F1 = \{A \rightarrow CD\}$ – jest 2PN, 3PN
 - $U2 = \{A, B\}$, $F2 = \emptyset$, więc kluczem jest AB, jest 2PN, 3PN
- Uwaga, powyższe rozwiązania wykorzystywały **twierdzenie o rozkładalności** (omawiane na wykładzie i ćwiczeniach)
- Można zadanie wykonać też **metodą syntezy** ale do tego potrzebujemy bazy minimalnej – c.d. następne slajdy

Normalizacja c.d. – rozwiązanie metoda 2

- Rozwiązanie powyższych zadań **metodą syntezy**:
- Znormalizuj schemat relacji:
 $U = (A, B, C, D, E)$
 $F = (AB \rightarrow C, A \rightarrow DE, B \rightarrow D, A \rightarrow B, E \rightarrow C)$
- Krok 1 – znajdź bazę minimalną (poza programem)
 - Baza minimalna: $F_{\min} = \{ A \rightarrow BE; B \rightarrow D; E \rightarrow C \}$
- Sprawdź postać normalną:
 - Klucz: A
 - 2PN – tak, bo klucz jednoatrybutowy (nie ma części)
 - 3PN – nie, bo $B \rightarrow D, E \rightarrow C$
- Sprowadź do 3PN:
 - Każda zależność generuje relację:
 - $U1 = \{A, B, E\}, U2 = \{B, D\}, U3 = \{E, C\}$

Normalizacja c.d. – rozwiązanie metoda 2

- Znormalizuj schemat relacji:

$U = (A, B, C, D)$

$F = (AB \rightarrow C, B \rightarrow C, A \rightarrow CD)$

- Krok 1 – znajdź bazę minimalną;

- $F_{\min} = \{ A \rightarrow CD; B \rightarrow C \}$

- Jaka postać normalna?

- Klucz: AB

- 2PN – nie, bo $A \rightarrow CD; B \rightarrow C$

- 3PN – nie, bo nie jest 2PN

- Znormalizuj do 3PN

- Uwaga – algorytm syntezy nie umożliwia normalizacji tylko do 2PN

Normalizacja c.d. – rozwiązanie metoda 2

- Schemat w 3PN:
- Każda zależność generuje relację:
 - $U1=\{A,C,D\}$, $U2=\{B,C\}$
- Jeśli żadna z relacji nie zawiera pierwotnego klucza (tutaj: AB) to dodaj dodatkową relację z tym kluczem:
 - $U3 = \{A,B\}$

SQL

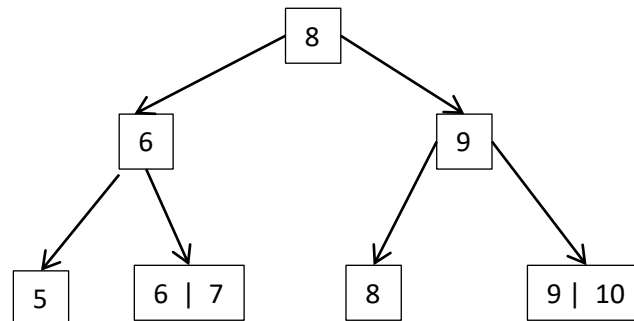
- Jakie jest typowe wykorzystanie
 - Procedur – operacje CRUD (tworzenie, zapis, modyfikacja, usuwanie danych; realizacja funkcjonalności); również – realizacja reguł biznesowych; raportowanie; redukcja przesyłanych danych
 - Funkcji – odczyt danych, wyznaczanie wartości wyliczanych;
 - Wyzwalaczy – realizacja złożonych ograniczeń na bazie; realizacja operacji kaskadowych;
 - Widoków – upraszczanie korzystania z bazy, zapewnienie bezpieczeństwa poprzez ukrycie struktury bazy, nadawanie uprawnień

Optymalizacja

- Optymalizacja:
 - Regułowa
 - Kosztowa
- Statystyki
 - Co zawierają?
- Plan wykonania zapytania
- Jakie struktury mogą być wykorzystane w procesie optymalizacji zapytań?

Indeksy

- Struktura fizyczna bazy danych
- Cel stosowania indeksów
- Ogólna budowa indeksu
- Rodzaje indeksów
- Budowa B+drzewa, wstawianie do B+drzewa
 - Wstaw do B+drzewa $p=3$ wartości: 7, 8, 9, 6, 5, 10
 - Wynik:



Indeksy c.d.

- Jaki powinien być rząd indeksu o strukturze B+drzewa, jeżeli rozmiar bloku wynosi 2kB, pole klucza ma długość 16B a pole wskaźnika ma długość 8B
- Rozwiązanie:
 - p – rząd indeksu
 - Tzn. w węźle zmieści się $(p-1)$ kluczy i p wskaźników
 - Chcemy żeby węzeł maksymalnie wypełnił blok
 - Zatem: $(p-1) \cdot 16B + p \cdot 8B \leq 2048 B$
 - Z tego: $p = 86$

Indeksy c.d.

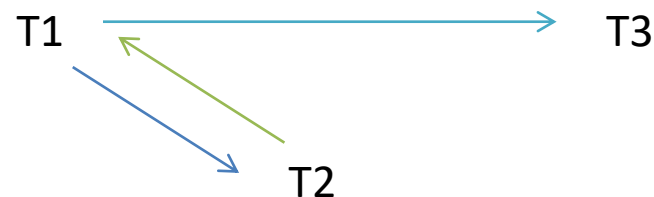
- Jaka jest maksymalna liczba odwołań do pamięci zewnętrznej, jeżeli dla pliku liczącego 14 000 rekordów korzystamy z indeksu w postaci B+drzewa rzędu 11?
- Rozwiązanie: przeszukując drzewo musimy ściągnąć po jednym bloku z każdego jego poziomu plus 1 blok z danymi; musimy zatem wyliczyć wysokość drzewa
- Wysokość drzewa rzędu p indeksującego n rekordów to $\log_p n$
- Zatem liczba odwołań to $\log_{11} 14000 + 1$

Transakcje

- Pojęcie transakcji
- Własność ACID
- Poziomy izolacji, anomalie

Transakcje c.d.

- Czy ta historia przetwarzania jest uszeregowalna?
- H: r1[x] w1[x] r2[y] r3[x] w2[y] r2[x] r1[y]
- Szukamy i zaznaczamy operacje konfliktowe, czyli takie, które pochodzą z dwóch niezatwierdzonych transakcji, z czego jedna jest operacją zapisu, na tych samych danych
- Sprawdzamy, że jest cykl, więc historia nie jest uszeregowalna



Transakcje c.d.

- H: r1[x] w1[x] r2[y] w3[y] r2[x] r3[x] c1 c2 c3
- Jak H zostanie zrealizowana przez algorytm 2PL na poszczególnych poziomach izolacji?
- Poziom O – nie ma blokady do odczytu
 - H: r1[x] w1[x] r2[y] w3[y] r2[x] r3[x] c1 c2 c3
- Poziom 1 – blokada do odczytu od razu zdejmowana
 - H: r1[x] w1[x] r2[y] w3[y] c1 r2[x] r3[x] c2 c3
- Poziom 2 i 3 – blokada do odczytu trzymana do końca
 - H: r1[x] w1[x] r2[y] c1 r2[x] c2 w3[y] r3[x] c3

UNDO/REDO

- Jak zostanie zrealizowany algorytm UNDO/REDO dla poniższej historii przetwarzania 2 transakcji, jeśli w buforze danych mieści się tylko jeden rekord
- Wartości początkowe:
a=20
b=50
- H: r1[a] w1[a=a*2] r2[b] w2[b=b-20] c2 *awaria!*

UNDO/REDO



Faza UNDO

Wycofane są zmiany wprowadzone przez niezatwierdzone transakcje

Lista zatwierdzonych: T2

Wycofana zmiana wprowadzona przez T1:

a = 20

operacja	plik logu
r1[a]	S1
w1[a=a*2]	(W,1,a,20,40)
r2[b]	S2
w2[b=b-20]	(W,2,b,50,30)
c2	C2 (zapis bufora na dysk)



Faza REDO

Wprowadzane są ponownie zmiany wprowadzone przez zatwierdzone transakcje

Zatwierdzona: T2

Przywracamy b:

b = 30

Po fazie UNDO/REDO w bazie są wartości: a = 20, b = 30

Bezpieczeństwo

- Konta, role, uprawnienia
- Nadawanie uprawnień
- Rola procedur i perspektyw