

# **Dazy Banych**

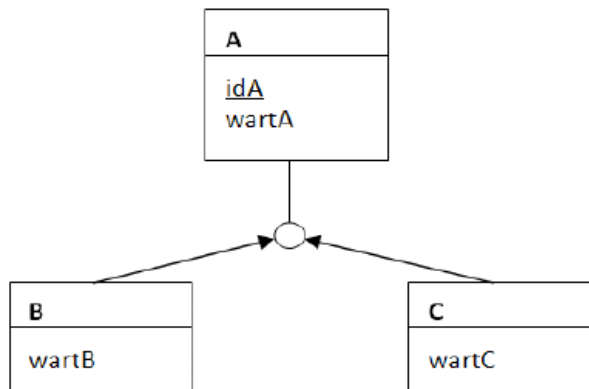
Michał Rusnarczyk

1. Co reprezentuje encja słaba na diagramie ER?

- (a) Obiekt niepowiązany z żadnym innym obiektem
- (b) Obiekt zależny od innego obiektu i poprzez niego identyfikowany
- (c) Obiekt istniejący tylko tymczasowo

Encją słabą (ang. weak entities) nazywamy taką encję, której istnienie zależy od istnienia innej encji (jej właściciela). Na przykład: istnienie jednostki

2. Dany jest poniższy fragment diagramu ER, reprezentujący specjalizację całkowitą rozłączną.



Rodzaje specjalizacji:

Całkowita lub częściowa – zależnie od tego czy każda instancja encji nadtypu musi być instancją którejś z encji podtypu (całkowita) czy nie (częściowa)

Rozłączna lub nierozłączna – zależnie od tego czy podtypy są parami rozłączne, czy nie

Wskaż odpowiadający mu model relacyjny

- (a) A (idA, wartA, wartB, wartC, typ)
- (b) A (idA, wartA); B(wartB); C(wartC)
- (c) AB (idA, wartA, wartB); AC (idA, wartA, wartC)

### Zad3

Dane są następujące tabele A i B. Jaki będzie wynik wykonania zapytania SQL?

| A    |      |
|------|------|
| Kol1 | Kol2 |
| X    | 1    |
| Y    | 2    |
| Z    | 3    |

| B    |      |
|------|------|
| Kol1 | Kol3 |
| X    | 10   |
| Y    | 20   |
| W    | 30   |

```
select A.kol2
from A
where not exists (select *
                  from B
                  where B.kol1 = A.kol1)
```

- (a) {3}
- (b) {1, 2}
- (c) {1, 2, 3}

4. Dane są następujące tabele A i B:

| A    |      |
|------|------|
| Kol1 | Kol2 |
| X    | 1    |
| Y    | 2    |
| Z    | 3    |

| B    |      |
|------|------|
| Kol1 | Kol3 |
| X    | 10   |
| Y    | 10   |
| X    | 30   |

Jaki będzie wynik wykonania operacji w RRD:

$\{\text{wynik} \mid \exists \text{kol1 } A(\text{kol1}, \text{wynik}) \wedge \exists \text{kol3 } B(\text{kol1}, \text{kol3}) \wedge \text{kol3} = 10\}$

- (a) {X, Y}  
(NIE)
- (b) {1, 2}  
(TAK)
- (c) {10}  
(NIE)

## 5. W jakim podstawowym celu stosuje się kursory w bazie danych?

(a) Do realizacji zapytań rekurencyjnych

(b) Do symulowania wskaźników na zmienne

(c) Do iteracji po zbiorze danych

Kursor (ang. cursor) jest to tymczasowa struktura służąca do pobierania wyniku zapytania SQL, dostarcza mechanizmu dostępu do pojedynczych wierszy wg zasady "jeden po drugim".

Przykład:

Aby przetwarzać dane za pomocą kursora należy:

|                                                                |                                                                                                                                                        |
|----------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>declare @nazwisko varchar(20)</code>                     | zadeklarować kursor DECLARE ... CURSOR i zdefiniować jego zakres FOR SELECT                                                                            |
| <code>declare cursor kursor</code>                             | otworzyć kursor, OPEN - wykonywana jest operacja SELECT, kursor wskazuje na miejsce bezpośrednio przed pierwszym wierszem w utworzonym zbiorze wierszy |
| <code>for select nazwisko from pracownicy</code>               |                                                                                                                                                        |
| <code>open kursor</code>                                       |                                                                                                                                                        |
| <code>fetch next from kursor into @nazwisko</code>             | wykonać operację pobrania danych na kolejnym wierszu FETCH NEXT FROM, pobierane dane można zapamiętać pod podanymi zmiennymi                           |
| <code>while @@fetch_status = 0</code>                          |                                                                                                                                                        |
| <code>Begin</code>                                             |                                                                                                                                                        |
| <code>    update pracownicy</code>                             |                                                                                                                                                        |
| <code>    set placa = placa*1.1 where current of kursor</code> |                                                                                                                                                        |
| <code>    print @nazwisko</code>                               |                                                                                                                                                        |
| <code>    fetch next from kursor into @nazwisko</code>         |                                                                                                                                                        |
| <code>end</code>                                               | pobieranie danych może być realizowane w pętli, różna od zera wartość zmiennej (funkcji) globalnej @@FETCH_STATUS                                      |
| <code>close kursor</code>                                      | może kończyć wykonywanie operacji pobierania danych                                                                                                    |
| <code>deallocate kursor</code>                                 |                                                                                                                                                        |
| w końcu kursor należy zamknąć CLOSE i dealokować DEALLOCATE    |                                                                                                                                                        |

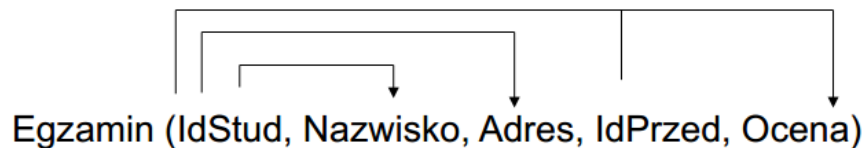
6. Na czym polega anomalia modyfikacji występująca w relacji nie będącej w 2PN?
- (a) Modyfikacja jednej informacji może wymagać modyfikacji wielu wierszy bazy
  - (b) Modyfikacja danych może spowodować utratę innych danych
  - (c) Modyfikacja danych może doprowadzić do niespójności bazy danych

## 2 Postać normalna (2PN)

DEFINICJA Niech  $X$  i  $Y$  będą rozłącznymi zbiorami atrybutów. Mówimy, że  $Y$  jest *w pełni funkcyjnie zależny* od  $X$  lub, że istnieje *pełna zależność funkcyjna* z  $X$  do  $Y$ , jeśli  $Y$  jest zależny funkcyjnie od  $X$  i nie jest zależny od żadnego właściwego podzbioru zbioru  $X$ .

Przykład

Schemat relacyjny, który **nie jest** w 2PN



Klucz główny: {IdStud, IdPrzed}

Atrybuty kluczowe: IdStud, IdPrzed

Atrybuty niekluczowe: Nazwisko, Adres, Ocena

Nazwisko i Adres zależą funkcyjnie od IdStud (tj. od części klucza głównego)

Ocena zależy funkcyjnie od całego klucza głównego.

Taka niejednorodność prowadzi do wielu anomalii.

6. Na czym polega anomalia modyfikacji występująca w relacji nie będącej w 2PN?

- (a) Modyfikacja jednej informacji może wymagać modyfikacji wielu wierszy bazy
- (b) Modyfikacja danych może spowodować utratę innych danych
- (c) Modyfikacja danych może doprowadzić do niespójności bazy danych

**Anomalia dołączania** – nie można dołączyć informacji o studencie dopóki nie zda co najmniej jednego egzaminu (atrybut kluczowy *IdPrzed* nie może być NULL-em).

**Redundancja** – pewne informacje (Nazwisko i Adres) pamiętane są wielokrotnie, problemy z wprowadzaniem i modyfikacją danych.

**Anomalia usuwania** – usunięcie informacji o zdanym egzaminie (np. Nowaka) powoduje usunięcie wszystkich informacji o studencie (jeśli to był jedyny jego egzamin).

7. Jakiego typu problemy mogą wiązać się z relacją nie będącą w 3PN?

- (a) Anomalia wstawiania i usuwania
- (b) Redundancja (nadmiarowość) danych
- (c) Utrata zależności funkcyjnych

# 3 Postać normalna (3PN)

DEFINICJA Schemat relacyjny  $R = (U, F)$  jest w 3PN, jeśli jest w 2PN i żaden niekluczowy atrybut  $A \in U$  nie jest tranzytywnie zależny od żadnego klucza (głównego lub nie) schematu  $R$ .

Relacja (tabela) jest w 3 postaci jeśli :

- jest w II postaci normalnej
- każdy atrybut jest funkcjonalnie zależny jedynie od klucza głównego, nie mogą więc istnieć jakiejkolwiek zależności przechodnie

# Przykład (3PN)

| imię    |        | nazwisko |               | miejsce urodzenia |            |
|---------|--------|----------|---------------|-------------------|------------|
| Emil    | Zajac  | Zajac    | Pszczyna      | Pszczyna          | pszczyński |
| Zofia   | Zima   | Zima     | Pszczyna      | Pszczyna          | pszczyński |
| Eulalia | Jańska | Jańska   | Szczebrzeszyn | Szczebrzeszyn     | zamojski   |

| miejsowość    | powiat     |
|---------------|------------|
| Pszczyna      | pszczyński |
| Szczebrzeszyn | zamojski   |

## tabela `Osoby` przed normalizacją

W powyższej tabeli pole powiat jest tylko częściowo zależny od klucza głównego, którym jest imię i nazwisko. Nazwa powiatu jest funkcjonalnie zależna od kolumny z nazwą miejsca urodzenia danej osoby. Z tego powodu tabela nie jest w III PN. Należy zauważyć, że istnieje tu nadmiarowość po normalizacji w 3PN: (a) z powodu

powtarzającej się nazwy powiatu pszczyński. Może to być przyczyną **pod-tabela `Miejscowości` po normalizacji** 3PN. Normalizacja tabeli z imionami i nazwiskami nie może się w 3PN zapożyczyć nazwy powiatu (a) w tabeli `miejsowości`, w której jest informacja o tym do jakiego powiatu należy dane miejsce.

7. Jakiego typu problemy mogą wiązać się z relacją nie będącą w 3PN?

(a) Anomalia wstawiania i usuwania

(TAK)

(b) Redundancja (nadmiarowość) danych

(TAK)

(c) Utrata zależności funkcyjnych

(NIE)

8. Dany jest schemat  $R(U, F)$  gdzie  $U = \{A, B, C, D\}$ ,  $F = \{AB \rightarrow C, B \rightarrow D, B \rightarrow A\}$ . Wskaż wszystkie klucze kandydujące.

(a)  $\{A, B\}$

(b)  $\{B, C\}$

(c)  $\{B, D\}$

# Klucz kandydujący

W bazach danych jest to dowolna kolumna, która może pełnić rolę klucza głównego (jej wartości jednoznacznie identyfikują rekord). Kandyduje on na klucz główny tabeli, ale czy zostanie wybrany zależy od projektu bazy i decyzji programisty.

## Definicja

Niech dana będzie tabela  $R$  typu  $U$  i niech  $X, Y \subseteq U$ .

Mówimy, że w  $R$  **spełniona jest zależność funkcyjna**  $X \rightarrow Y$ , co zapisujemy

$$R \models X \rightarrow Y,$$

jeśli dla każdych dwóch krotek  $r_1, r_2 \in R$ , zachodzi

$$r_1[X] = r_2[X] \Rightarrow r_1[Y] = r_2[Y],$$

tzn. z równości krotek na atrybutach ze zbioru  $X$  wynika równość tych krotek na atrybutach ze zbioru  $Y$ .

Uwaga:

$r[X]$  – projekcja krotki  $r$  na zbiór atrybutów  $X$ .

8. Dany jest schemat  $R(U, F)$  gdzie  $U = \{A, B, C, D\}$ ,  $F = \{AB \rightarrow C, B \rightarrow D, B \rightarrow A\}$ . Wskaż wszystkie klucze kandydujące.

(a)  $\{A, B\}$   
(TAK)

(b)  $\{B, C\}$   
(TAK)

(c)  $\{B, D\}$   
(NIE)

$\{A, B\}$

Z  $\{A\}$  nie otrzymujemy żadnej zależności funkcyjnej  
Z  $\{B\}$  otrzymujemy zależność funkcyjną  $D$  i  $A$   
Z  $\{AB\}$  otrzymujemy zależność funkcyjną  $C$   
W sumie  $\{A, B\}$  wyznacza nam jednoznacznie  $A, B, C, D$  więc może być kluczem kandydującym.

$\{B, D\}$

Z  $\{B\}$  otrzymujemy zależność funkcyjną  $D$  i  $A$   
Z  $\{D\}$  nie otrzymujemy żadnej zależności funkcyjnej  
W sumie  $\{A, B\}$  wyznacza nam jednoznacznie  $A, B, D$  więc NIE może być kluczem kandydującym.

$\{B, C\}$

Z  $\{B\}$  otrzymujemy zależność funkcyjną  $D, A$   
Z  $\{C\}$  nie otrzymujemy żadnej zależności funkcyjnej  
W sumie  $\{B, C\}$  wyznacza nam jednoznacznie  $A, B, C, D$  więc może być kluczem kandydującym.

9. Dany jest zbiór zależności funkcyjnych  $F = \{AB \rightarrow C, C \rightarrow D, BD \rightarrow A, A \rightarrow B\}$ . Wskaż zależności, które można wyprowadzić z tego zbioru.

(a)  $\{A \rightarrow C\}$

(b)  $\{AB \rightarrow D\}$

(c)  $\{CB \rightarrow A\}$

(a)  $A \rightarrow B, AB \rightarrow C$

(b)  $AB \rightarrow C, C \rightarrow D$

(c)  $C \rightarrow D, BD \rightarrow A$

10. Dany jest schemat relacyjny  $R = (U, F)$ , gdzie:  $U = \{A, B, C, D\}, F = \{A \rightarrow B, BC \rightarrow D, D \rightarrow B, D \rightarrow C\}$

oraz rozkład:  $R1 = (U1, F1), R2 = (U2, F2)$ , gdzie:  $U1 = \{A, B\}, F1 = \{A \rightarrow B\}, U2 = \{B, C, D\}, F2 = \{BC \rightarrow D, D \rightarrow B, D \rightarrow C\}$ .

Rozkład ten jest rozkładem:

- (a) bez straty zależności
- (b) bez straty danych
- (c) na składowe niezależne

# Rozkładalność schematów relacyjnych

Wyróżnimy następujące pojęcia rozkładalności:

- 1) rozkładalność bez straty danych,
- 2) rozkładalność bez straty zależności,
- 3) rozkładalność na składowe niezależne (jednocześnie bez straty danych i bez straty zależności).

# Rozkładalność bez straty danych

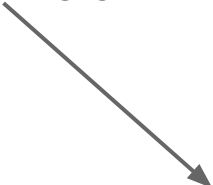
DEFINICJA 8. Mówimy, że schemat relacyjny

$$R = (U, F)$$

jest *rozkładalny bez straty danych* na dwa schematy  $R[X]$  i  $R[Y]$ , wtedy i tylko wtedy, gdy

a)  $X \cup Y = U$ ,

b)  $R = R[X] \bowtie R[Y]$ .



**Definicja 1.5.4** (Złączenie schematów). Niech  $\mathcal{R} = (X, F)$ ,  $\mathcal{S} = (Y, G)$ . Schemat  $\mathcal{T} = (X \cup Y, (F \cup G)^+)$  nazywamy złączeniem schematów relacyjnych.

# Rozkładalność bez straty zależności

**Definicja 1.6.2** (Rozkładalność bez straty zależności funkcyjnych). Mówimy, że schemat relacyjny  $\mathcal{R} = (U, F)$  jest rozkładalny bez straty zależności funkcyjnych na schematy  $\mathcal{R}_1 = (X, G)$  oraz  $\mathcal{R}_2 = (Y, H)$  gdy

- 1)  $X \cup Y = U$
- 2)  $(G \cup H)^+ = F^+$

**Przykład.** Dla schematu relacyjnego  $R := ( U, F )$

$$U := \{ A, B, C, D \},$$

$$F := \{ A \rightarrow B, BC \rightarrow D, D \rightarrow B, D \rightarrow C \}$$

rozważmy następujące schematy:

$$R_1 := ( \{ A, B \}, \{ A \rightarrow B \} ),$$

$$R_2 := ( \{ B, C, D \}, \{ BC \rightarrow D, D \rightarrow B, D \rightarrow C \} ),$$

będące rozkładami schematu  $R$  bez straty zależności.  
Rozkład ten nie jest jednak rozkładem bez straty danych.

***R:***

| <i>A</i>              | <i>B</i> | <i>C</i>              | <i>D</i>              |
|-----------------------|----------|-----------------------|-----------------------|
| <i>a</i>              | <i>b</i> | <i>c</i>              | <i>d</i>              |
| <i>a</i> <sub>1</sub> | <i>b</i> | <i>c</i> <sub>1</sub> | <i>d</i> <sub>1</sub> |
| <i>a</i> <sub>2</sub> | <i>b</i> | <i>c</i> <sub>1</sub> | <i>d</i> <sub>1</sub> |

**Wówczas relacje:**

$$***R*<sub>1</sub> := R[AB] \quad \text{ i } \quad R_2 := R[BCD]**$$

**mają postać:**

$R_1$ :

| $A$   | $B$ |
|-------|-----|
| $a$   | $b$ |
| $a_1$ | $b$ |
| $a_2$ | $b$ |

$R_2$ :

| $B$ | $C$   | $D$   |
|-----|-------|-------|
| $b$ | $c$   | $d$   |
| $b$ | $c_1$ | $d_1$ |

i  $R \neq R_1 \triangleright \triangleleft R_2$  (następny slajd).

Zauważmy, że zależności  $B \rightarrow A$  i  $B \rightarrow CD$  nie należą do  $F^+$ , tzn. nie są spełnione założenia twierdzenia o warunku koniecznym i dostatecznym rozkładalności bez straty danych.

# Złączenie relacji R1 i R2

$R_1 \triangleright \triangleleft R_2$

| <i>A</i>              | <i>B</i> | <i>C</i>              | <i>D</i>              |
|-----------------------|----------|-----------------------|-----------------------|
| <i>a</i>              | <i>b</i> | <i>c</i>              | <i>d</i>              |
| <i>a</i>              | <i>b</i> | <i>c</i> <sub>1</sub> | <i>d</i> <sub>1</sub> |
| <i>a</i> <sub>1</sub> | <i>b</i> | <i>c</i>              | <i>d</i>              |
| <i>a</i> <sub>1</sub> | <i>b</i> | <i>c</i> <sub>1</sub> | <i>d</i> <sub>1</sub> |
| <i>a</i> <sub>2</sub> | <i>b</i> | <i>c</i>              | <i>d</i>              |
| <i>a</i> <sub>2</sub> | <i>b</i> | <i>c</i> <sub>1</sub> | <i>d</i> <sub>1</sub> |

$R$ :

| <i>A</i>              | <i>B</i> | <i>C</i>              | <i>D</i>              |
|-----------------------|----------|-----------------------|-----------------------|
| <i>a</i>              | <i>b</i> | <i>c</i>              | <i>d</i>              |
| <i>a</i> <sub>1</sub> | <i>b</i> | <i>c</i> <sub>1</sub> | <i>d</i> <sub>1</sub> |
| <i>a</i> <sub>2</sub> | <i>b</i> | <i>c</i> <sub>1</sub> | <i>d</i> <sub>1</sub> |

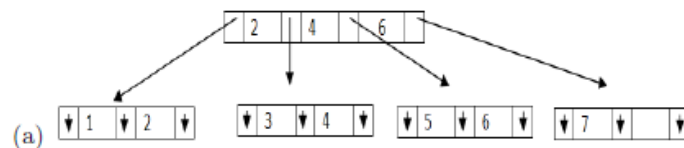
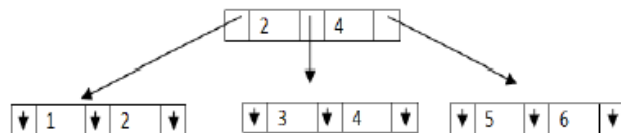
10. Dany jest schemat relacyjny  $R = (U, F)$ , gdzie:  $U = \{A, B, C, D\}, F = \{A \rightarrow B, BC \rightarrow D, D \rightarrow B, D \rightarrow C\}$

oraz rozkład:  $R1 = (U1, F1), R2 = (U2, F2)$ , gdzie:  $U1 = \{A, B\}, F1 = \{A \rightarrow B\}, U2 = \{B, C, D\}, F2 = \{BC \rightarrow D, D \rightarrow B, D \rightarrow C\}$ .

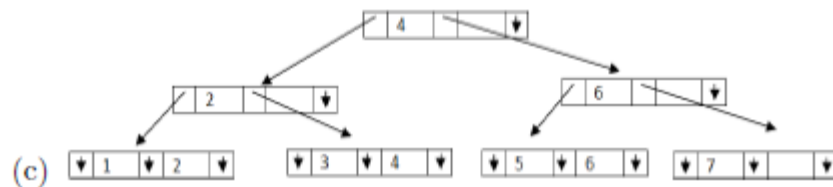
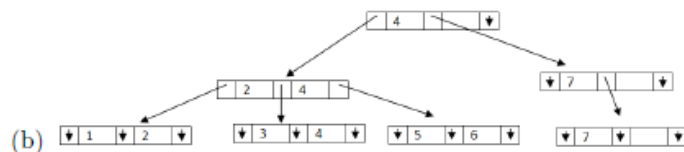
Rozkład ten jest rozkładem:

- (a) bez straty zależności  
(TAK)
- (b) bez straty danych  
(NIE)
- (c) na składowe niezależne  
(NIE)

11. Jaka będzie postać następującego  $B^+$ -drzewa rzędu  $p=3$  (tzn. posiadającego maksymalnie 2 klucze i 3 wskaźniki w węźle) po wstawieniu do niego wartości 7?



(Na tablicy)



# Definicja indeksu oraz B-drzewa

## Definicja indeksu:

$I = \{(x, a) : r \in F(R) \mid x = \text{key}(r) \wedge a = \text{kbd}(r)\}$   $F(R)$  – plik rekordów typu  $R$ ,  $F(R) = \{r_1, \dots, r_m\}$ ,  $x = \text{key}(r)$  – wartość klucza głównego w rekordzie  $r$ ,  $a = \text{kbd}(r)$  – klucz bazy danych (adres rekordu  $r$ ).

## Definicja B-drzewa:

Drzewo skierowane  $T$  nazywamy B-drzewem klasy  $t(h, m)$ , jeśli  $h = 0$  (drzewo puste) lub Wszystkie drogi od korzenia do liści są długości  $h$ , Każdy wierzchołek z wyjątkiem korzenia ma, co najmniej  $m$  kluczy (elementów) i  $m+1$  synów, Każdy wierzchołek ma, co najwyżej  $2m$  kluczy (i  $2m+1$  synów), Korzeń ma, co najmniej jeden klucz.

# Dołączanie elementu do B-drzewa

Chcemy dołączyć element indeksu o kluczu  $x$ , tak aby nie naruszyć struktury B-drzewa. Dołączanie poprzedzone jest procedurą SZUKAJ, w wyniku której, albo znajdziemy wierzchołek zawierający klucz  $x$  (koniec) albo znajdziemy adres wierzchołka (liścia) do którego należy dołączyć klucz  $x$ . Jeśli wierzchołek ma mniej niż  $2m$  elementów to dołączamy nowy klucz. Jeśli wierzchołek ma  $2m$  elementów to następuje tzw. kolizja (przepełnienie, nadmiar), którą rozwiązujemy albo metodą podziału albo metodą kompensacji.

## Metoda podziału

$2m+1$  elementów dzielimy na trzy części. Elementy  $1, \dots, m$  umieszczamy w wierzchołku 1 Element  $m+1$  przenosimy do strony ojca Elementy  $m+2, \dots, 2m+1$  umieszczamy w wierzchołku 2

## Metoda kompensacji

Można ją stosować jeżeli sąsiednia strona zawiera  $j < 2m$  elementów. Stan wyjściowy –  $j=2 < 2m$ , obliczamy  $i = \text{entier}((2m+j+3)/2) = ((4+2+3)/2) = 4$ . Elementy  $1, 2, \dots, i-1$  umieszczamy w wierzchołku 1, Element  $i$  przenosimy do strony ojca, Elementy  $i+1, \dots, 2m+j+2$  umieszczamy w wierzchołku 2.

12. Dana jest historia wykonywania transakcji współbieżnych:

H: w1[x] r2[x] w2[u] c2 w1[z] a1

Historia ta jest:

- (a) nieodtworzalna
- (b) dopuszczalna przy poziomie izolacji 0
- (c) dopuszczalna przy poziomie izolacji 1

# Transakcja

- Stosuje się pojęcie transakcji jako jednostki operowania na bazie danych podlegającej sterowaniu i kontroli.
- Transakcja T jest sekwencją logicznie powiązanych operacji na bazie danych, przeprowadzających bazę danych z jednego stanu spójnego w inny stan spójny.

Te operacje to:

- $rT[x]$  – czytanie (read) danej x przez transakcję T,
  - $wT[x]$  – zapisanie (write) (aktualizacja) danej x przez transakcję T,
  - $aT$  – odrzucenie (abort, rollback) (wycofanie) transakcji T,
  - $cT$  – zatwierdzenie (commit) transakcji T.
- Mówiąc o danej x mamy na myśli jednostki danych na różnych poziomach granulacji – może to być dana elementarna, krotka (rekord), zbiór krotek wyznaczony przez warunek, tabela bazy danych, itp.

# Przykład ogólny nieodtworzalnej historii przetwarzania

Przypuśćmy, że transakcja T1 zmieniła wartość danej x.

- Następnie transakcja T2 wczytała x i na podstawie jej wartości zmieniła wartość danej y w bazie danych
- Przypuśćmy dalej, że transakcja T2 została zatwierdzona, a po tym zdarzeniu powstała konieczność odrzucenia transakcji T1.
- Należy więc wycofać wszystkie zmiany, jakie wprowadziła w bazie danych transakcja T1, a także wszystkie konsekwencje tych zmian – w szczególności więc zmianę wartości danej y.
- Ta ostatnia operacja jest jednak niemożliwa, gdyż transakcja T2, która zmianę tę wykonała, jest już zatwierdzona.
- Zaistniała sytuacja, w której baza danych jest nieodtworzalna(!)

12. Dana jest historia wykonywania transakcji współbieżnych:

H: w1[x] r2[x] w2[u] c2 w1[z] a1

Historia ta jest:

(a) nieodtworzalna

(b) dopuszczalna przy poziomie izolacji 0

# Przykład szczegółowy

- Rozważmy historię przetwarzania:  $H1 : w1[x] \ r2[x] \ w2[y] \ c2 \ w1[z] \ a1$
- $H1$  opisuje przetwarzanie nieodtworzalne.
- Transakcja  $T1$  zmodyfikowała  $x$ :  $w1[x]$ .
- Transakcja  $T2$  czyta tą daną:  $r2^*x$ , i na podstawie jej wartości modyfikuje daną  $y$ :  $w2[y]$ ;  $T2$  jest zatwierdzana
- Teraz transakcja  $T1$  jest wycofywana, a wraz z nią wszystkie zmiany, jakie wykonała, a więc również zmiana danej  $x$
- Nie można jednak wycofać zmiany danej  $y$  gdyż transakcja  $T2$  została wcześniej zatwierdzona.

# Poziomy Izolacji transakcji (nazwy stand. SQL2)

## Poziom izolacji 0 (READ UNCOMMITTED)

- Poziom izolacji 0 dopuszcza czytanie przez transakcję danych nie zatwierdzonych tj. danych które zostały zmienione przez transakcję jeszcze nie zatwierdzoną. (12 b)
- Za operacje konfliktowe uważa się tylko parę operacji zapisu, a dwie operacje, z których jedna jest operacją odczytu nie są operacjami konfliktowymi.

## Poziom izolacji 1 (READ COMMITTED)

- Poziom izolacji 1 wprowadza zakaz czytania danych z transakcji niezatwierdzonych, a więc czytać można tylko dane zatwierdzone (14 c)
- Przy tym poziomie izolacji dopuszczalne jest jednak zapisywanie danych w transakcjach nie zatwierdzonych. (12 c), (14 b)
- Za konfliktowe uważa się wówczas takie pary operacji, gdzie pierwsza jest operacją zapisu, a druga czytania, lub obydwie są operacjami zapisu. Dwie operacje, z których pierwsza jest operacją czytania, a druga operacją zapisu nie są więc konfliktowe.

# Poziomy Izolacji transakcji (nazwy stand. SQL2) c.d.2

## Poziom izolacji 2 (REPEATABLE READ)

- Poziom izolacji 2 wprowadza zakaz zapisywania w transakcjach nie zatwierdzonych. Jeśli więc transakcja nie zatwierdzona przeczytała jakąś daną, to dana ta może być tylko czytana przez inną transakcję. Jeśli natomiast transakcja nie zatwierdzona zapisała jakąś daną, to nie można jej ani odczytać ani tym bardziej zapisać dopóty, dopóki transakcja ta nie zostanie zatwierdzona.
- Za konfliktowe uważa się takie pary operacji, gdzie co najmniej jedna jest operacją zapisu. Za niekonfliktowe uważa się tylko operacje czytania.
- Na tym poziomie izolacji mogą pojawiać się fantomy.

## Poziom izolacji 3 (SERIALIZABLE)

Jeśli w obrębie jednej transakcji wykonujemy właśnie polecenie SELECT wówczas z poziomu dowolnej innej transakcji nie możemy wykonać zmiany danych, które są właśnie wybierane poleceniem SELECT. Inaczej mówiąc zapytania w obrębie transakcji są wykonywane tak, jakby automatycznie była do nich dołączana klauzula blokowania danych na których się wykonują

12. Dana jest historia wykonywania transakcji współbieżnych:

H: w1[x] r2[x] w2[u] c2 w1[z] a1

Historia ta jest:

- (a) nieodtworzalna  
(TAK)
- (b) dopuszczalna przy poziomie izolacji 0  
(TAK)
- (c) dopuszczalna przy poziomie izolacji 1  
(NIE)

# Anomalia powtórnego czytania

- Przypuśćmy, że transakcja T1 czyta daną x, a następnie transakcja T2 zapisuje nową wartość danej x i jest zatwierdzana.
- Jeśli teraz transakcja T1 ponownie przeczyta daną x, to może się okazać, że dana ta ma inną wartość.
- Transakcja T1 dysponuje więc dwiema różnymi wartościami tej samej danej.
- Może zdarzyć się też sytuacja, że transakcja T2 usunie daną x. Wówczas przy probie ponownego czytania, transakcja T1 ma informację, że danej x nie ma w bazie danych.

# Fantom

- Przypuśćmy, że transakcja T1 wczytała z tabeli R zbiór rekordów spełniających warunek  $\phi$ .
- Następnie inna transakcja, T2, dołączyła do R nowy rekord r spełniający warunek  $\phi$  i została zatwierdzona.
- Jeśli T1 ponownie odwoła się do rekordów tabeli R spełniających warunek  $\phi$ , to okaże się, że tym razem zbiór ten jest inny (większy).
- Podobna sytuacja wystąpi, jeśli transakcja T2 dokona takiej modyfikacji rekordu r' nie spełniającego warunku  $\phi$ , że po jej wykonaniu rekord r' warunek ten będzie spełniał.
- Ten nowy rekord pojawiający się w obszarze zainteresowań transakcji T1 nazywany jest fantomem lub zjawą.

# Podsumowanie

- Przyjęcie określonego poziomu izolacji może być źródłem problemów (anomalii). Może też eliminować te problemy. Jednocześnie zwiększa się lub zmniejsza współbieżność transakcji.
- W poniższej tabelicy symbol T oznacza, że dany problem występuje przy rozważanym poziomie izolacji, N – że nie występuje.

Związek poziomów izolacji z problemami przetwarzania transakcji

| Poziom<br>Izolacji   | Brak<br>odtworzał<br>ności | Anomalia<br>powtórnego<br>czytania | Fantomy |
|----------------------|----------------------------|------------------------------------|---------|
| 0 : READ UNCOMMITTED | T                          | T                                  | T       |
| 1 : READ COMMITED    | N                          | T                                  | T       |
| 2 : REPEATABLE READ  | N                          | N                                  | T       |
| 3 : SERIALIZABLE     | N                          | N                                  | N       |

13. Przy poziomie izolacji 2 (REPEATABLE READ) może wystąpić anomalia:

- (a) "brudnego czytania"  
(NIE)
- (b) powtórnego czytania  
(NIE)
- (c) fantomów  
(TAK)

14. Następujące uporządkowane pary operacji mogą być wykonywane współbieżnie:

- (a)  $w1[x]$ ,  $r2[x]$ , przy poziomie izolacji 0  
(TAK)
- (b)  $r1[x]$ ,  $w2[x]$ , przy poziomie izolacji 1  
(TAK)
- (c)  $w1[x]$ ,  $r2[x]$ , przy poziomie izolacji 1  
(NIE)

15. Protokół blokowania dwufazowego (B2F):

- (a) zabezpiecza przed zakleszczeniami (dead locks)
- (b) zabezpiecza przed powstaniem cykli w grafie szeregowalności
- (c) realizowany jest przez menedżera transakcji

# Protokół blokowania dwufazowego (B2F)

## Algorytmy synchronizacji współbieżnych transakcji

- Problem poprawnego uporządkowania operacji współbieżnie wykonywanych transakcji jest rozwiązywany np. za pomocą algorytmów blokowania dostępu do danych – uszeregowanie transakcji wynika z kolejności uzyskiwania blokad.
- W komercyjnych systemach zarządzania bazą danych najczęściej realizowanym algorytmem blokowania dostępu do danych jest algorytm blokowania dwufazowego B2F (two-phase locking, 2PL)
- Zarządzaniem transakcjami zajmuje się wyspecjalizowany moduł planisty.
- Planista związany jest z każdym menedżerem danych (MD) zarządzającym danymi określonej lokalnej bazy danych na określonym węźle sieci.
- Tyle jest planistów ile lokalnych baz danych w systemie.
- Nie ma natomiast żadnego centralnego planisty, który koordynowałby dostęp do lokalnych baz danych (taki centralny nadzorca jest natomiast potrzebny do rozwiązywania globalnych zakleszczeń).
- Każda transakcja może kierować swoje operacje odczytu/zapisu do wielu lokalnych baz danych

# Przetwarzanie transakcji

- **Menadżer transakcji (MT) koordynuje wykonywanie transakcji. Decyduje, do jakiego menadżera danych (MD) przekazać operację transakcji.**
- Operacje przejmuje planista związany z MD i zarządza ich wykonywaniem posługując się protokołem B2F (Blokowanie 2-Fazowe)
- Algorytm blokowania dwufazowego
- Algorytm blokowania dwufazowego składa się z dwóch faz: fazy wzrostu (ang. growing phase ) oraz fazy zmniejszania (ang. shrinking phase ). W pierwszej fazie transakcja może blokować zasoby, ale nie wolno jej zwalniać zasobów, które wcześniej już zablokowała. W drugiej fazie transakcja może zwalniać zasoby, lecz nie wolno jej blokować żadnych nowych zasobów. Jeżeli proces nie chce modyfikować danych dopóki nie osiągnie momentu przed fazą zmniejszania, to w razie niepowodzenia przy zakładaniu jakiegś blokady może on zwolnić wszystkie blokady i rozpocząć ponownie algorytm.
- Podstawową cechą algorytmu B2F jest to, że każda utworzona przez niego historia przetwarzania transakcji jest poprawna.
- **Wadą algorytmu B2F jest to, że może prowadzić do powstania zakleszczeń (ang. deadlock).**

# Poprawność B2F

- Definicja: Historię  $H$  nazywamy poprawną, jeśli dla każdej pary transakcji  $T, T'$  spełniony jest warunek: jeśli  $T < T'$ , to nieprawda, że  $T' < T$ .
- Jeśli historia  $H$  jest poprawna, to określona na niej relacja „ $<$ ” wprowadza częściowy porządek w zbiorze transakcji. Zbiór transakcji możemy wówczas przedstawić w postaci acyklicznego grafu skierowanego zwanego grafem uszeregowalności (GU) transakcji.

15. Protokół blokowania dwufazowego (B2F):

- (a) zabezpiecza przed zakleszczeniami (dead locks)  
(NIE)
- (b) zabezpiecza przed powstaniem cykli w grafie szeregowalności  
(TAK)
- (c) realizowany jest przez menedżera transakcji  
(NIE)