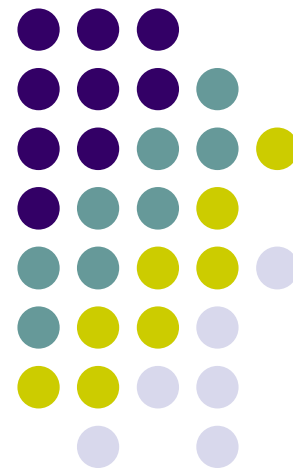


wykład

Przetwarzanie transakcyjne Cz.1



Technologia baz danych (2)



2. Przetwarzanie transakcyjne (spójność bazy danych)

- dostęp do bazy danych za pomocą transakcji o własnościach ACID
- metody synchronizacji transakcji (2PL, znaczniki czasowe, wielowersyjność danych)
- metody odtwarzania spójności bazy danych (plik logu, odtwarzanie i wycofywanie operacji, punkty kontrolne)
- archiwizacja bazy danych i odtwarzanie po awarii



Plan wykładu

Celem niniejszego wykładu jest omówienie problematyki związanej z transakcjami w bazie danych.

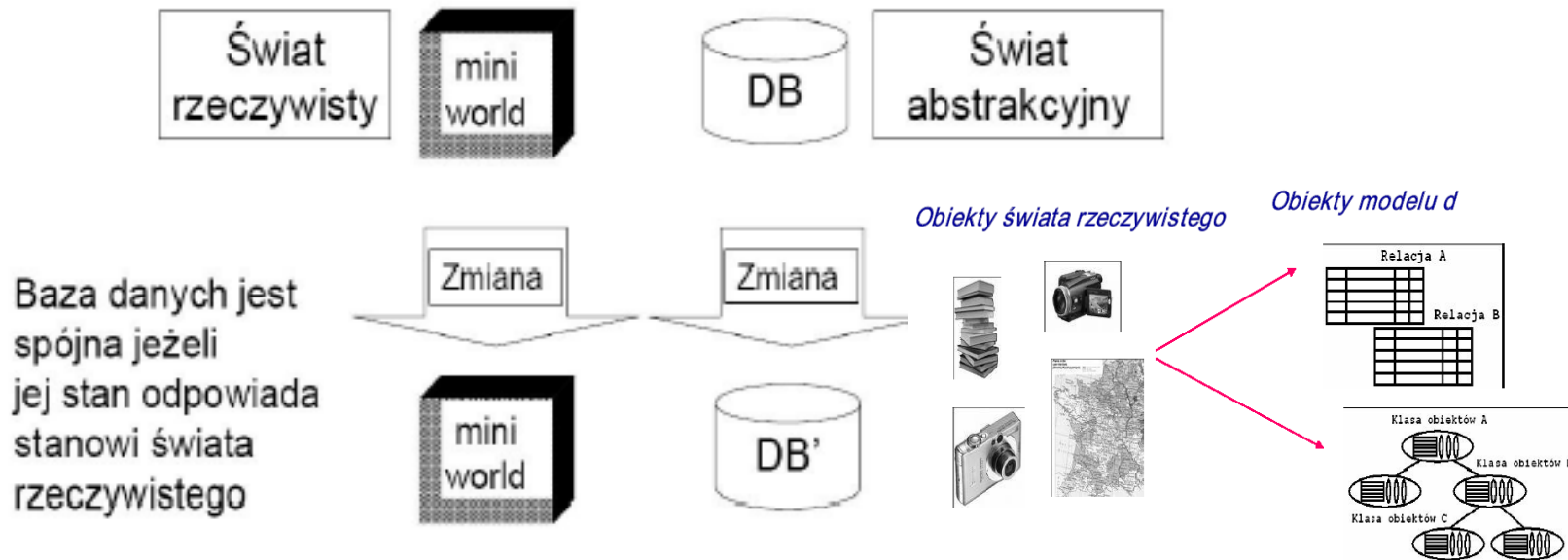
W szczególności zostaną omówione:

- transakcja i jej własności,
- formalny model transakcji,
- sekwencyjne i współbieżne realizacje zbioru transakcji,
- uszeregowalność transakcji.

wprowadzenie (1)



- Baza danych – jest abstrakcyjnym odzwierciedleniem wybranego fragmentu rzeczywistości (ang. *miniworld*)



Baza danych jest spójna, jeżeli jej stan odpowiada stanowi świata rzeczywistego.

wprowadzenie (2)



1. Zmiany zachodzące w świecie rzeczywistym muszą być zakodowane w postaci programu, który będzie transformował bazę danych z jednego stanu spójnego do innego stanu spójnego
2. Niebezpieczeństwa związane z realizacją programu transformującego bazę danych
 1. Awaryjność środowiska sprzętowo-programowego
 2. Współbieżny dostęp do danych
 3. Rozproszenie baz danych

1. spójności BD → transformacja BD z jednego stanu spójnego do innego stanu spójnego.
2. Odporność na awarie sprzętowo-programowe.
3. Obsługa **równoległa**.

problemy przygotowania aplikacji



Przykład: Napisać aplikację przelewu kwoty N z konta A na konto B

Problem 1 – awaria systemu

Po pobraniu kwoty N z konta A, i zapisaniu tej aktualizacji do bazy danych, wystąpiła awaria systemu. W wyniku awarii systemu wykonana została jedynie część operacji składających się na daną aplikację

1

Problem 2 – współbieżny dostęp do danych

Operacje współbieżnie wykonywanych transakcji mogą naruszać spójność bazy danych, lub generować niepoprawne wyniki

2

Przykład: system bankowy i aplikacja przelewu kwoty N z konta A na konto B.



transakcja (1)



Problem 3 - utrata danych w wyniku awarii

Wyniki zakończonych aplikacji, buforowane w pamięci operacyjnej, mogą zostać utracone w wyniku awarii systemu

Rozwiązaniem problemu awaryjności, rozproszenia i wielodostępności środowiska systemu bazy danych – koncepcja transakcji

3

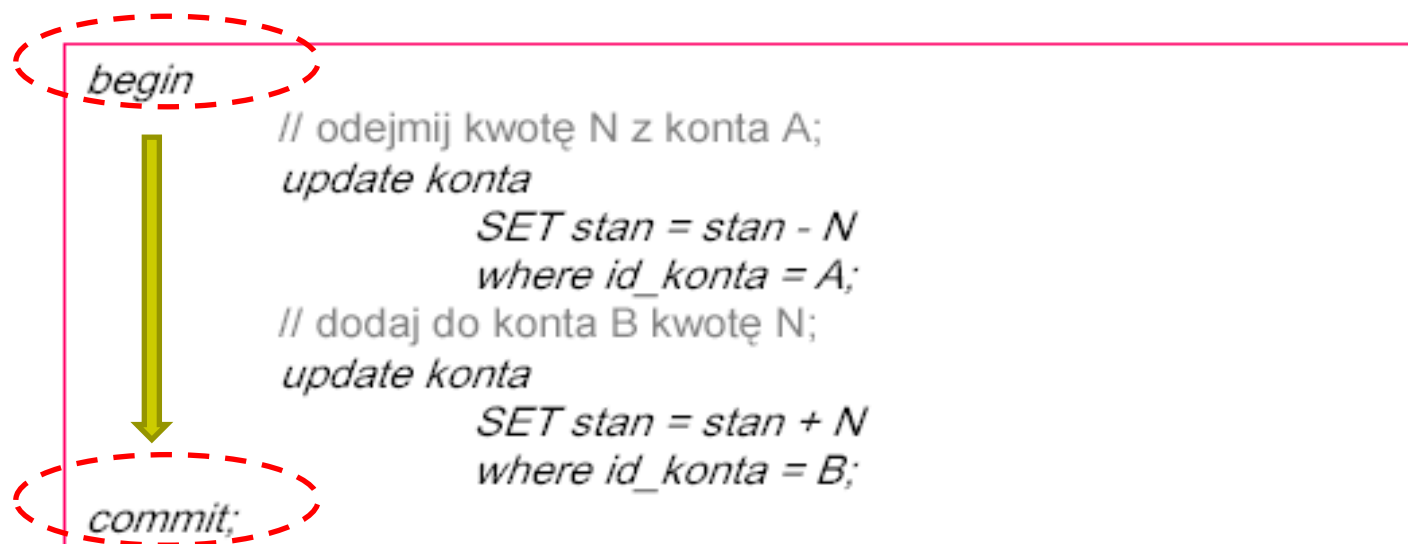
Transakcja jest sekwencją logicznie powiązanych operacji na bazie danych, która przeprowadza bazę danych z jednego stanu spójnego w inny stan spójny. Typy operacji na bazie danych obejmują: odczyt i zapis danych oraz zakończenie i akceptację (zatwierdzenie), lub wycofanie transakcji

DEFINICJA

transakcja (2)



Transakcja przelewu kwoty N z konta A na konto B:



własności transakcji (1)



A(tomicity)C(onsistency)I(solation)D(urability)

ATOMOWOŚĆ - Atomicity

SPÓJNOŚĆ - Consistency

ACID

własności transakcji (2)



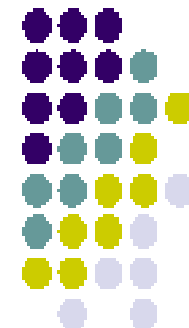
A(atomicity)C(onsistency)I(solation)D(urability)

IZOLACJA - Isolation

TRWAŁOŚĆ - Durability

ACID

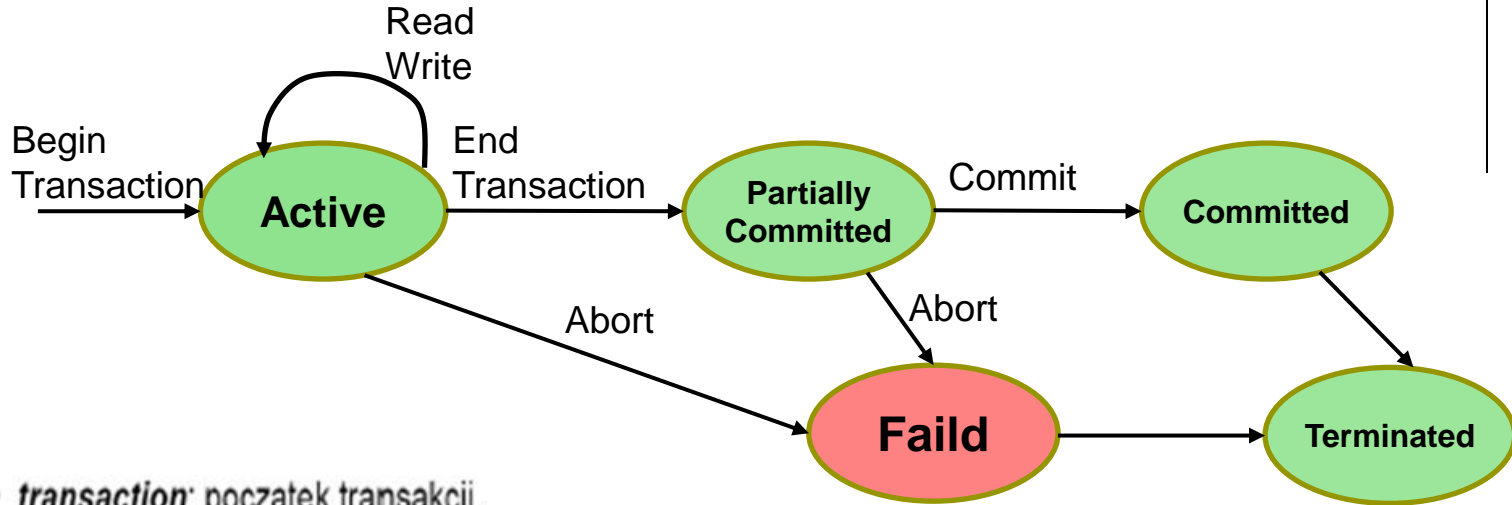
transakcja (3)



Transakcja jest: Cechy transakcji

1. Atomowa: jeżeli pieniądze zostaną poprawnie przetransferowane z konta A do B
2. Spójna: jeżeli kwota odejta z konta A jest równa kwocie dodanej do konta B
3. Izolowana: jeżeli inne transakcje wykonywane współbieżnie, czytające i modyfikujące konta A i B, nie mają wpływu na transakcję
4. Trwała: jeżeli po zakończeniu transakcji, baza danych trwale odzwierciedla nowe stany kont A i B

diagram stanów transakcji



Begin_transaction: początek transakcji.

Read, Write: operacje odczytu i zapisu danych w bazie danych.

End_transaction: koniec transakcji:

Commit: zatwierdzenie (akceptacja) wyników transakcji.

Rollback: wycofanie wyników transakcji

Stany transakcji

Active

Partially Committed

Committed

Failed

Terminated

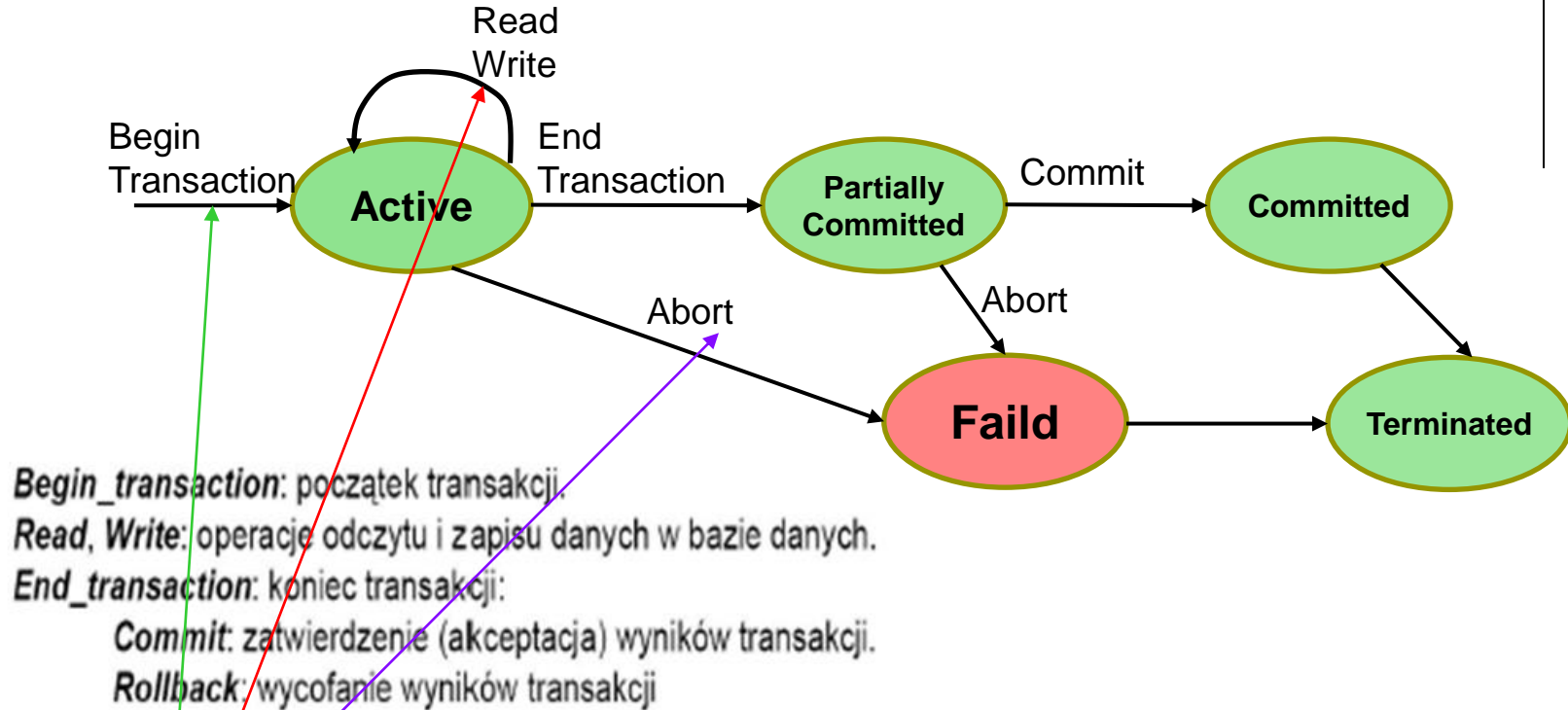
Stany transakcji

Save point:

Release savepoint

Rollback to savepoint

diagram stanów transakcji



1. (**Begin Transaction**) uruchamia transakcję, która jest aktywna.
2. (**Read, Write**) dokonuje się w stanie aktywnym transakcji.
3. (**Abort**) przeprowadza transakcję ze stanu **Active** do stanu **Failed**, a następnie **Terminate**.
4. Kończenie transakcji z jej zatwierdzeniem przeprowadza ją ze stanu **Active** do **Partially committed** transakcja jest gotowa do zatwierdzenia.

zakończenie transakcji



End_transaction: koniec transakcji oznacza, że wszystkie operacje odczytu i/lub zapisu transakcji zostały wykonane. W tym momencie, zachodzi konieczność podjęcia decyzji, czy zmiany wprowadzone przez transakcję mają być wprowadzone do bazy danych (zatwierdzenie transakcji) czy też mają być wycofane z bazy danych

Commit: zatwierdzenie (akceptacja transakcji) oznacza pomyślne zakończenie transakcji - zmiany wprowadzone przez transakcję mają być wprowadzone do bazy danych

Rollback: wycofanie transakcji oznacza niepoprawne zakończenie transakcji i konieczność wycofania z bazy danych wszystkich ewentualnych zmian wprowadzonych przez transakcję

transakcja logiczna a fizyczna



Transakcja
logiczna

```
UPDATE employee  
SET salary = 1.15 * salary  
WHERE work_period > 5;
```

```
COMMIT;
```

Begin_transaction;

```
Read (A);  
Write (A);  
...  
Read (Z);  
Write (Z);
```

```
Commit;
```

Transakcja
fizyczna



Poziom użytkownika: zbiór poleceń języka SQL, tj. **select, insert, update, delete, commit, rollback** - tzw. transakcja logiczna.

Poziom SZBD - transakcja fizyczna, która jest zarządzana przez odpowiedni moduł **SZBD**.



model transakcji (1)

Transakcją T_i nazywamy uporządkowaną parę:

$$T_i = (\overline{T_i}, < T_i)$$

gdzie:

$\overline{T_i} = \{o_j : 1 \leq j \leq n_j\}$ - oznacza zbiór operacji na bazie danych:

{ R - odczyt, W - zapis, C - zatwierdzenie transakcji, A - wycofanie }

$< T_i$ - jest relacją częściowego porządku na zbiorze T_i

Zbiór operacji zawiera:

- odczyt (**R**),
- zapis (**W**),
- zatwierdzenie transakcji (**C**),
- wycofanie transakcji (**A**).



model transakcji (1)

Transakcją T_i nazywamy uporządkowaną parę:

$$T_i = (\overline{T_i}, < T_i)$$

gdzie:

$\overline{T_i} = \{o_j : 1 \leq j \leq n_j\}$ - oznacza zbiór operacji na bazie danych:

{ R - odczyt, W - zapis, C - zatwierdzenie transakcji, A - wycofanie }

$< T_i$ - jest relacją częściowego porządku na zbiorze $\overline{T_i}$

Dalsza notacja: Przyjmiemy następującą notację:

- $r_i(x)$ lub $r_i(x, \text{wartość})$
- $w_i(x)$ lub $w_i(x, \text{wartość})$
- c_i lub a_i

model transakcji (2)



Każda transakcja może być reprezentowana przez graf skierowany:

$G = (V, A)$, gdzie:

V - jest zbiorem węzłów odpowiadających operacjom transakcji T_i

A - jest zbiorem krawędzi reprezentujących porządek na zbiorze operacji

Przykład:

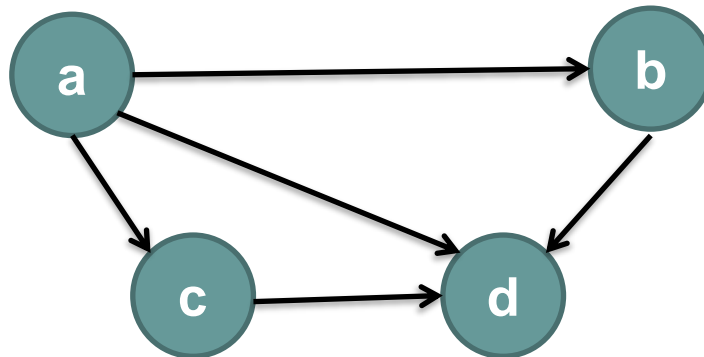
$r_1(x) \rightarrow w_1(x) \rightarrow r_2(y) \rightarrow w_2(y) \rightarrow c_1$

sekwencyjne wykonywanie
transakcji

$r_1(x) \rightarrow w_1(x) \rightarrow w_2(y) \rightarrow c_1$

$r_2(y) \rightarrow$

współbieżne wykonywanie
transakcji



Graf skierowany, digraf - DG

klasyfikacja transakcji



Ze względu na porządek operacji:

- transakcja sekwencyjna
- transakcja współbieżna

Ze względu na zależność operacji:

- transakcja zależna od danych
- transakcja niezależna od danych

Ze względu na typy operacji:

- zapytania lub transakcja odczytu (read only)
- transakcja aktualizująca - transakcja (read/write)

Trzy kryteria podziału transakcji: **porządek operacji**, **zależność operacji**, **typ operacji**.



KONIEC WYKŁADU

Cz. 1