

BAZY DANYCH, WYKŁAD NR 5

Uniwersytet Gdański

Agenda

2

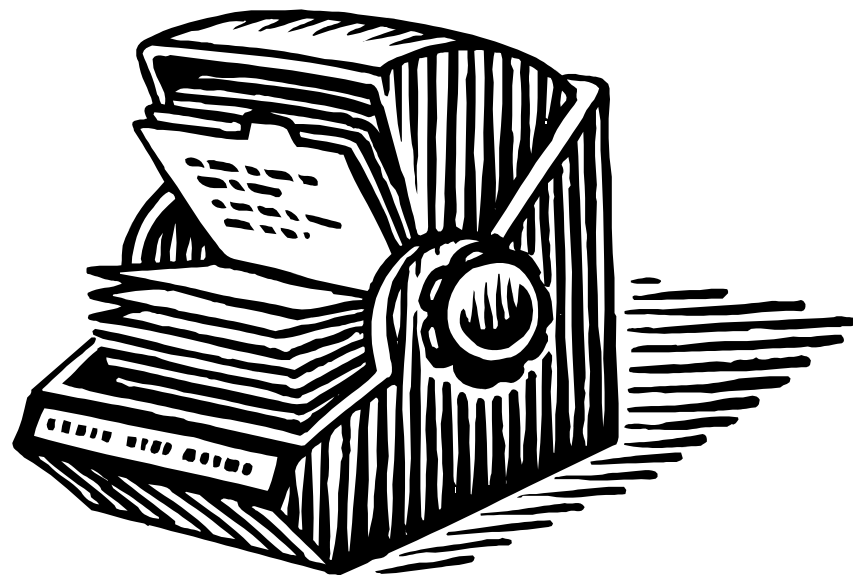
Teoria baz danych

- Transakcyjność

Język definiowania struktur danych

- Instrukcje modyfikujące
- Wyzwalacze
- Funkcje

Teoria baz danych



Transakcyjność w bazie danych

4



- Transakcja jest zbiorem powiązanych logicznie operacji, występujących w określonej sekwencji.
- Tymi operacjami mogą być:
 - ▣ odczyt i zapis danych
 - ▣ zatwierdzenie operacji (commit)
 - ▣ wycofanie operacji (rollback)

Właściwości transakcji



5

- ACID – Atomicity, Consistency, Isolation, Durability
- czyli
 - ▣ Atomowość
 - ▣ Spójność
 - ▣ Izolacja
 - ▣ Trwałość

Atomowość transakcji

6



- Inaczej niepodzielność.
- Oznacza, że wszystkie operacje zaplanowane w sekwencji muszą zostać wykonane.
- Przykład:
 - ▣ Najpierw następuje pobranie aktualnego numeru sekwencji, jego powiększenie o 1, wstawienie tego numeru do nowego wiersza.

Spójność transakcji



7

- Transakcja może wprowadzać bazę danych w chwilowy stan niespójności.
- Przykład:
 - ▣ Może to być zwiększanie ceny produktu dla dużej liczby produktów. W danym momencie nie wszystkie produkty mogą zostać zmodyfikowane.

Izolacja transakcji



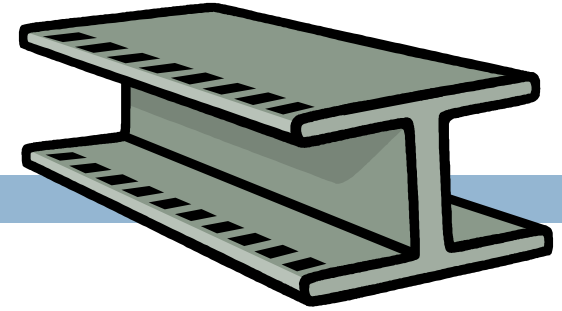
8

- Transakcje wykonują się zawsze w sposób sekwencyjny.

- Przykład:
 - ▣ Oznacza to, że w tym samym momencie nie ma możliwości wprowadzania nowych wierszy przez dwa zapytania. Dzięki temu nie nastąpi powtórzenie wartości klucza głównego.

Trwałość

9



- Dane zatwierdzone nie są tracone.
- Przykład:
 - ▣ Dane wprowadzone do bazy danych po wykonaniu instrukcji commit są zatwierdzane.

Instrukcje transakcji

10



□ commit

- ▣ zatwierdza transakcję
(np. wprowadzanie wierszy)

□ rollback

- ▣ anuluje wcześniej wykonaną transakcję
(np. usuwanie wierszy)

Część II

Język SQL





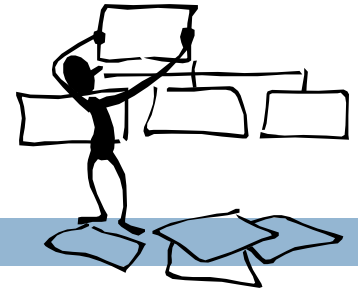
Instrukcja UPDATE

12

- W celu aktualizacji danych w języku sql używa się instrukcji UPDATE. Składnia tej instrukcji jest następująca:
 - ▣ UPDATE nazwaTabeli
 - ▣ SET nazwaKolumny=wyrażenie [, kolumna=wyrażenie]
 - ▣ [WHERE warunek]

- Przykład:
 - ▣ update uzytkownik set haslo='K0wAlSk1' where nazwisko='Kowalski';
 - ▣ update produkt set cena=cena*0.5;
 - ▣ update klient set imie='Anna' where id_klienta=10;

Instrukcja ALTER



13

- Instrukcja modyfikująca strukturę tabel.

- Ogólna postać instrukcji:
 - ▣ ALTER TABLE nazwa_tabeli
ADD | DROP | ALTER...

Instrukcja ALTER

– fraza ADD



14

- Dodaje do tabeli kolumny.
- ALTER TABLE nazwa_tabeli ADD (definicja kolumny [definicja ograniczenia] [,definicja kolumny [definicja ograniczenia]]);

Instrukcja ALTER

– fraza DROP

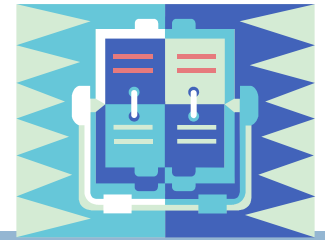


15

- ❑ Usuwa z tabeli kolumny.
- ❑ ALTER TABLE nazwa_tabeli
DROP COLUMN
(nazwa_kolumny [,nazwa_kolumny])

Instrukcja ALTER

– fraza ALTER

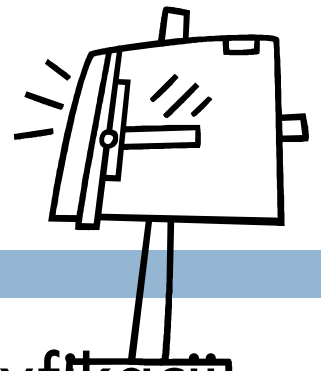


16

- Zmienia nazwę kolumny w tabeli.
- ALTER TABLE nazwa_tabeli
ALTER COLUMN nazwa_kolumny typ;

Procedura sp_rename

– zmiana nazwy kolumny tabeli



17

- Aby zmienić nazwę tabeli należy użyć specyfikatora polecenia RENAME.
 - ▣ EXEC sp_rename 'nazwaTabeli.nazwaKolumny', 'NowaNazwa', 'COLUMN';
- Przykład:
 - ▣ EXEC sp_rename , 'dbo.a.b', 'c', 'COLUMN';

Instrukcja ALTER

– dodanie kolumny do tabeli

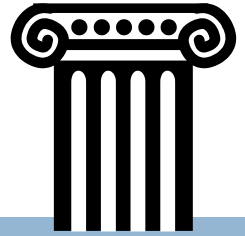


18

- Dodanie nowej kolumny do tabeli jest możliwe poprzez użycie specyfikacji polecenia ADD.
 - ▣ ALTER TABLE nazwaTabeli ADD nowaKolumna typKolumny;

- Przykład:
 - ▣ ALTER TABLE uzytkownicy ADD numerTelefonu varchar(15);

Usuwanie kolumn



19

- Do usunięcia istniejącej kolumny z tabeli służy polecenie DROP COLUMN.
 - ▣ ALTER TABLE nazwaTabeli DROP COLUMN nazwaKolumny;

- Przykład:
 - ▣ ALTER TABLE a DROP COLUMN c;

Modyfikacje kolumn w tabelach



20

- Do zmiany charakterystyki istniejącej kolumny używa się polecenia ALTER.
 - ▣ ALTER TABLE nazwaTabeli ALTER nazwaKolumny typKolumny [opcje];

- Przykłady:
 - ▣ alter table a.test alter column b int;

Tworzenie indeksów



21

- Indeksy pozwalają na szybsze działanie bazy danych. Zazwyczaj są tworzone podczas tworzenia tabeli, jednak istnieje możliwość tworzenia indeksów po jej utworzeniu.
- Do tworzenia indeksów używana jest instrukcja CREATE INDEX.
 - CREATE [UNIQUE] INDEX nazwaIndeksu [typIndeksu] ON nazwaTabeli(nazwaKolumnyIndeksu, ...)
- gdzie
 - nazwaKolumnyIndeksu: nazwaKolumny [ASC | DESC]
 - UNIQUE – tworzy indeksy unikalne
 - ASC oraz DESC – tworzenie indeksów z rosnącym lub malejącym porządkiem sortowania

Polecenia przydatne podczas indeksowania

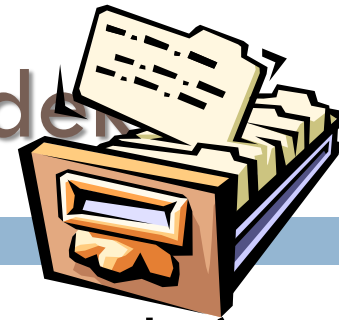
22



- Indeksy należy zakładać zgodnie z kryteriami najczęstszego wyszukiwania danych.

- Usunięcie indeksu:
 - ▣ `DROP INDEX nazwa_indeksu;`

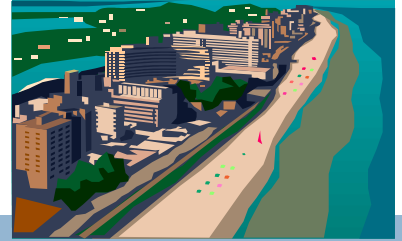
Przykłady poleceń tworzących indeksy



23

- ❑ `create index nazwiskoInd on uzytkownicy(nazwisko);`
- ❑ `create index fragmentNazwiskoInd on uzytkownicy(nazwisko);`
- ❑ `create index nazwiskoImie on uzytkownicy(nazwisko asc, imie desc);`

Perspektywy



24

- Pozwalają na tworzenie wirtualnych tabel, które bazują na istniejących tabelach.
- Celem tworzenia perspektyw jest ograniczenie liczby wyświetlanych kolumn i danych, umożliwienie przeglądania tylko wybranych danych, uproszczenie dostępu do wyników pochodzących z wielu tabel.
- Inna nazwa to widok.

Ogólna składnia zapytania tworzącego perspektywę

25



- Tworzenie perspektywy
 - ▣ CREATE VIEW nazwa_widoku AS SELECT zapytanie [WITH READ ONLY];

- Usuwanie perspektywy
 - ▣ DROP VIEW nazwa_widoku;

Przykłady pracy z perspektywami

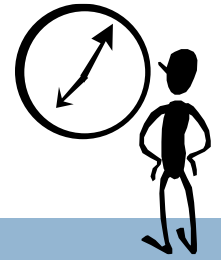
26



- ❑ `CREATE VIEW nazwiskaKlientow
AS SELECT DISTINCT nazwisko
FROM klient;`
- ❑ `SELECT * FROM nazwiskaKlientow;`
- ❑ `DROP VIEW nazwiskaKlientow;`



Wyzwalacze



28

Małe programy zbliżone konstrukcją do procedur i funkcji, najczęściej wykonujące się w momencie wystąpienia określonego zdarzenia, np. dodania kolejnego wiersza do tabeli

Wyzwalacze są kompilowane i umieszczane w bazie danych



Wyzwalacze

29

- ❑ **CREATE TRIGGER** nazwa
- ❑ **ON** nazwa_tabeli
- ❑ **FOR [INSERT], [UPDATE], [DELETE]**
- ❑ **AS**
 - ❑ polecenia_triggera

Przykład nr 1



30

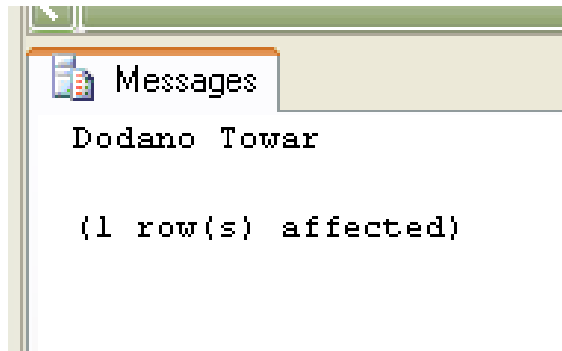
1. CREATE TRIGGER dodaj_towar ON towar
2. FOR INSERT
3. AS
4. BEGIN
5. PRINT 'Dodano Towar'
6. END

Działanie wyzwalacza dla przykład nr 1

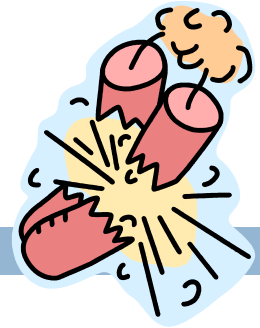


31

- insert into Towar (nazwa_towaru, cena, data_dodania) values ('Ołówek',2.32,GETDATE());



Przykład nr 2



32

1. **CREATE TRIGGER usun_towar ON towar**
2. **FOR DELETE**
3. **AS**
4. **BEGIN**
5. **ROLLBACK TRANSACTION**
6. **RAISERROR('Produkt nie może zostać usunięty',1,1)**
7. **END**

Działanie wyzwalacza dla przykładu nr 2



33

❑ delete from Towar where id=2;

```
Messages
Produkt nie może zostać usunięty
Msg 50000, Level 1, State 1
Msg 3609, Level 16, State 1, Line 1
The transaction ended in the trigger. The batch has been aborted.
```

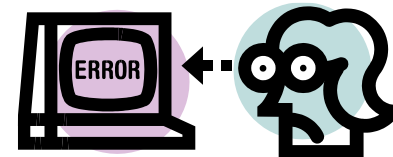
Instrukcje ROLLBACK TRANSACTION i RAISERROR



34

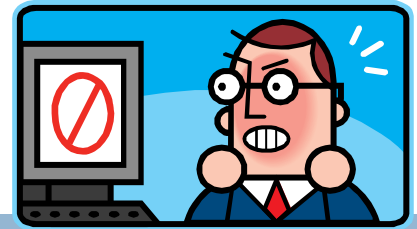
- ❑ **ROLLBACK TRANSACTION** – anuluje wszystkie zmiany wprowadzone przez wykonywaną transakcję
- ❑ **RAISERROR** – przerywa wykonywanie transakcji i wyświetla komunikat o błędzie

Instrukcja RAISERROR



35

- **RAISERROR('Produkt nie może zostać usunięty', 1, 1)**
 - **'Produkt nie może zostać usunięty'** – skutkuje kodem błędu 50000
 - **1** – surowość błędu (ang. severity), przyjmuje wartość 1-18 oraz 19-25 (tylko SYSADMIN)
 - **1** – zdefiniowany przez użytkownika numer z zakresu od 0 do 255, pozwala na znalezienie fragmentu kodu odpowiedzialnego za błąd



Przykład nr 3

36

1. CREATE TRIGGER CenaPowyzejZakresu ON Towar
2. AFTER INSERT, UPDATE
3. AS
4. DECLARE
5. @cena NUMERIC(5,2)
6. BEGIN
7. -- wyłączenie opcji wypisywania dodatkowego
8. -- zestawu danych po instrukcji select
9. SET NOCOUNT ON;
10. SELECT @cena = cena FROM towar
11. IF @cena > 200
12. BEGIN
13. **ROLLBACK TRANSACTION**
14. **RAISERROR('Cena przekracza dozwoloną wartość',16,1)**
15. END
16. END

Działanie wyzwalacza dla przykładu nr 3



37

- insert into Towar (nazwa_towaru, cena, data_dodania) values ('Temperówka',299.31,GETDATE());

Messages

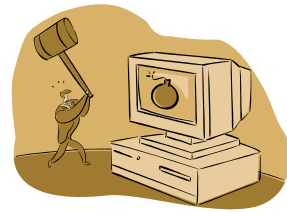
Dodano Towar

Msg 50000, Level 16, State 1, Procedure CenaPowyzejZakresu, Line 14
Cena przekracza dozwoloną wartość

Msg 3609, Level 16, State 1, Line 1
The transaction ended in the trigger. The batch has been aborted.

	id	nazwa_towaru	cena	data_dodania
1	6	Ołówek	2,32	2011-04-18
2	2	Kredka	0,52	2011-04-11
3	3	Długopis	1,32	2011-04-11
4	4	Mazak czerwony	NULL	2011-04-11
5	5	Mazak zielony	NULL	2011-04-11

Porównanie surowości błędów



38

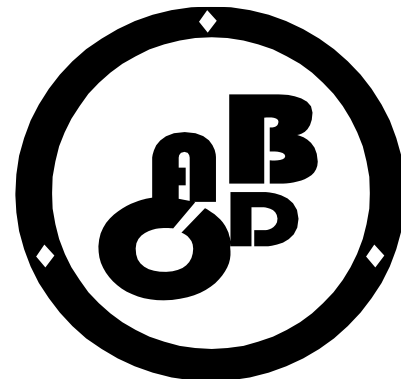
1. **ROLLBACK TRANSACTION**
2. **RAISERROR('Produkt nie może zostać usunięty',1,1)**

```
Messages
Produkt nie może zostać usunięty
Msg 50000, Level 1, State 1
Msg 3609, Level 16, State 1, Line 1
The transaction ended in the trigger. The batch has been aborted.
```

3. **ROLLBACK TRANSACTION**
4. **RAISERROR('Cena przekracza dozwoloną wartość',16,1)**

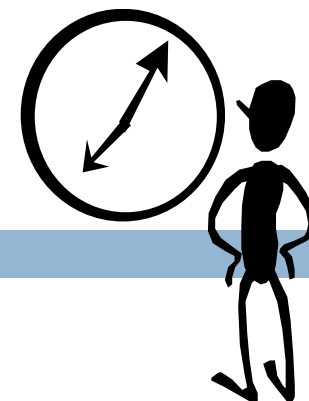
```
Messages
Dodano Towar
Msg 50000, Level 16, State 1, Procedure CenaPowyzejZakresu, Line 14
Cena przekracza dozwoloną wartość
Msg 3609, Level 16, State 1, Line 1
The transaction ended in the trigger. The batch has been aborted.
```

Typowe funkcje języka SQL



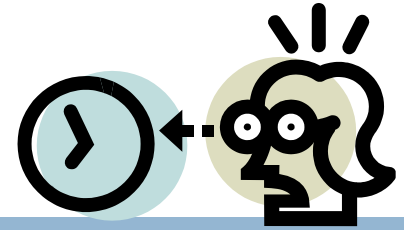
Funkcje daty (1 / 3)

40



- GETDATE() – zwraca bieżącą datę
- DATEADD(interwał, liczba, data) – zwiększenie daty o odpowiednio: dd – dni, mm – miesiące, yy – lata

Funkcje daty (2/3)



41

- DATENAME(jednostka czasu, data),
gdzie jednostka to dd, dw, mm, yy

- DAY(data)
- MONTH(data)
- YEAR(data)

Funkcje daty (3/3)



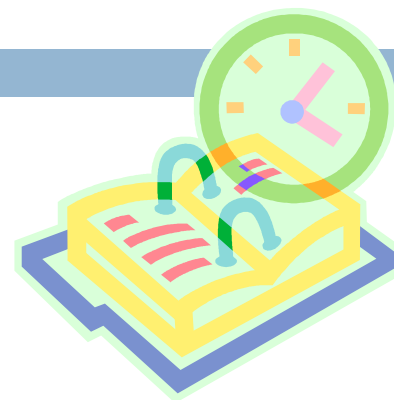
42

- DATEDIFF(jednostka czasu, data1, data2) – zwraca różnicę pomiędzy dwoma datami
- ISNULL(kolumna, 'wyświetlany tekst') – wypisuje wartość, jeżeli na pozycji kolumna jest ona pusta

Przykład – jaki będzie efekt działania?

43

1. DECLARE
2. @data Date
3. BEGIN
4. SET @data='2011-04-05';
5. SELECT GETDATE();
6. SELECT DAY(@data) AS Dzień, MONTH(@data) AS Miesiąc,
YEAR(@data) AS Rok;
7. SELECT DATEADD(mm, 14, '2016-04-05') AS Za_14_miesięcy;
8. SELECT DATEDIFF(dd, @data, '2016-06-10') AS Interwał;
9. SELECT DATENAME(MONTH, @data) AS Miesiąc;
10. SELECT ISNULL(NULL, @Data) AS Z_NULLem, ISNULL('2016-04-03',@Data) AS Bez_NULLa;
11. END



Rozwiązanie



44

5
6
7
8
9
10

	(No column name)		
1	2016-11-03 16:20:16.923		
	Dzień	Miesiąc	Rok
1	5	4	2016
	Za_14_miesięcy		
1	2017-07-04 00:00:00.000		
	Interwał		
1	66		
	Miesiąc		
1	kwiecień		
	Z_NULLem	Bez_NULLa	
1	2016-04-05	2016-04-03	

Pytanie?

45

- Które z poniższych słów kluczowych powodują zmianę struktury tabeli (wielokrotny wybór)?
 - ▣ DELETE
 - ▣ ALTER
 - ▣ UPDATE
 - ▣ DROP

