

# **Administracja i programowanie pod Microsoft SQL Server 2000**

Paweł Rajba

[pawel@ii.uni.wroc.pl](mailto:pawel@ii.uni.wroc.pl)  
<http://www.kursy24.eu/>

# Zawartość modułu 3

---

- Podstawy programowania w T-SQL
  - Zmienne i operatory
  - Instrukcje sterujące
  - Komunikaty
  - Format daty
  - Funkcje
    - Daty i czasu, matematyczne, napisowe
    - Systemowe, bezparametrowe, globalne

# Zmienne i operatory

---

- Zmienne lokalne
  - deklaracja: declare @nazwa typ [, @nazwa typ] ...
  - ustawienie: set @zmienna=wartosc
  - wypisanie: print @zmienna
- Operatory
  - Arytmetyczne: \*, /, %, +, -
  - Łączenie napisów: +
  - Logiczne: not, and, or
- Przykład
  - sqlserver-p03-01.sql

# Instrukcje sterujące

---

- BEGIN..END
- IF..ELSE
- RETURN [n]
- WAITFOR { DELAY 'time' | TIME 'time' }
- WHILE
- ..BREAK
- ..CONTINUE

# Instrukcje sterujące

---

- CASE
  - jest skrótem na if..else if..else ...
  - jest także wyrażeniem, stąd możemy użyć np. w
    - instrukcji SELECT
      - na liście wybierającej
      - w klauzulach: WHERE, GROUP BY, ORDER BY
    - instrukcji INSERT
      - na liście wartości INSERT
    - instrukcji UPDATE
      - w klauzulach: SET i WHERE
    - instrukcji DELETE
      - w klauzuli WHERE

# Wyrażenia bazujące na CASE

---

- COALESCE(w1, w2, w3, ..., wN)
  - zwraca pierwsze różne od NULL wyrażenie
  - jeśli wszystkie są NULL, zwraca NULL
  - równoważne

CASE

WHEN w1 IS NOT NULL THEN w1

...

WHEN wN IS NOT NULL THEN wN

ELSE wN

END

# Wyrażenia bazujące na CASE

---

- NULLIF(w1, w2)
  - równoważne  
CASE  
    WHEN w1=w2 THEN NULL  
    ELSE w1  
END
- ISNULL(w1, w2)
  - równoważne  
CASE  
    WHEN w1 IS NULL THEN w2  
    ELSE w1  
END

# Instrukcje sterujące

---

- Przykład
  - sqlserver-p03-02.sql



# Komunikaty

---

- Do tworzenia komunikatów mamy dwie funkcje: PRINT i RAISERROR
- PRINT służy do prostych komunikatów
- Powinno się raczej używać RAISERROR

- Składnia

```
RAISERROR ( { msg_id | msg_str }  
           { , waga , stan }  
           [ , argument [ ,...n ] ] )  
WITH LOG, NOWAIT, SETERROR
```

# Komunikaty

---

- RAISERROR

- Znaczenie parametrów

- msgid, msgstr – podajemy id komunikatu z tabeli master..sysmessages lub tekst komunikatu
  - tekst można formatować podobnie jak w funkcji printf
- waga komunikatu
  - 0 lub 10 – informacyjny
  - 11-16 – błędy, które może naprawić użytkownik
  - 17-18 – poważniejsze błędy
  - 19 – wewnętrzny błąd związany z zasobami
    - może go wygenerować tylko administrator (rola sysadmin)
    - musi być używane z opcją WITH LOG

# Komunikaty

---

- RAISERROR
  - Znaczenie parametrów
    - waga komunikatu c.d.
      - 20-25 – błędy krytyczne
        - może je wygenerować tylko administrator (rola sysadmin)
        - najczęściej przerywane są wsady, zrywane są połączenia
        - muszą być używane z opcją WITH LOG
    - stan komunikatu
      - dodatkowa informacja o komunikacie, np. nr wiersza, w którym pojawił się błąd

# Komunikaty

---

- RAISERROR
  - Znaczenie parametrów
    - LOG – komunikat zostanie zapisane w dzienniku zdarzeń systemu Windows
      - komunikaty o wadze  $\leq 14$  – jako informacyjne
      - komunikaty o wadze  $= 15$  – jako ostrzeżenia
      - komunikaty o wadze  $\geq 16$  – jako błędy
    - NOWAIT – sprawia, że wiadomość zostanie przesłana od razu do klienta (istotne w przypadku użycia funkcji WAITFOR)
    - SETERROR – sprawia, że nawet dla komunikatów informacyjnych funkcja @@ERROR będzie zwracała numer komunikatu (lub 50000 w przypadku podania tekstu)

# Komunikaty

---

- RAISERROR
  - Wersje językowe
    - Każdy komunikat w kolumnie msglangid ma zdefiniowany język
    - Można zatem dodać kilka wersji językowych tego samego komunikatu
  - Komunikaty użytkownika
    - mają numery od 50000
    - do obsługi mamy funkcje
      - sp\_addmessage
      - sp\_altermessage
      - sp\_dropmessage

# Komunikaty

---

- Przykład
  - sqlserver-p03-03.sql

# Fomat daty

---

- Format daty
  - SET DATEFORMAT { mdy|dmy|ymd|ydm|dym|myd }
    - określa sposób traktowania napisów np. '7/4/2005'
  - Format ISO
    - jest traktowane jednoznacznie niezależnie od ustawienia dateformat
- Przykład
  - sqlserver-p03-04.sql

# Funkcje

---

- Konwersja typów
  - CAST(wyrażenie AS typ\_danych)
  - CONVERT(typ\_danych[ (len) ], wyrażenie [, styl] )
    - styl można określić dla dat i liczb
- Do konwersji z typów zmiennoprzecinkowych
  - STR(wyrażenie [, dlugosc [, miejsca\_dziesietne ] ] )
- Przykład
  - sqlserver-p03-05.sql



# Funkcje

---

- Daty i czasu
  - DATEADD( składnik, liczba, datagodzina )
    - dodaje ustalony składnik do datagodzina
    - wartości składnika:
      - year, month, day, dayofyear, quarter
      - hour, minute, second, milisecond
      - week, weekday
  - DATEDIFF(składnik, datagodzina1, datagodzina2)
    - zwraca różnicę pomiędzy datami
    - jednostką będzie składnik

# Funkcje

---

- Daty i czasu
  - DATENAME(składnik, datagodzina)
    - zwraca nazwę reprezentującą składnik z datagodzina
  - DATEPART(składnik, datagodzina)
    - zwraca liczbę reprezentującą składnik z datagodzina
  - DAY(dg), MONTH(dg), YEAR(dg)
    - zwracają odpowiednio dzień, miesiąc i rok z datagodzina

# Funkcje

---

- Daty i czasu
  - GETDATE
    - zwraca bieżący czas i datę
  - GETUTCDATE
    - zwraca bieżący czas GMT
- Przykład
  - sqlserver-p03-06.sql

# Funkcje

---

- Matematyczne
  - ABS, CEILING, FLOOR, ROUND
  - SIN, COS, TAN
  - PI, EXP, LOG, LOG10
  - POWER, SQRT, SQUARE
  - SIGN, RAND

# Funkcje

---

- Napisowe
  - UPPER, LOWER
  - TRIM, LTRIM, RTRIM
  - REPLACE(t1, t2, t3) – wymienia w t1, t2 na t3
  - REPLICATE(t, n)
    - wypisuje napis t, n razy, jeśli  $n < 0$ , zwraca NULL
  - REVERSE(napis)
  - SPACE(liczba) – zwraca napis będący pewną liczbą spacji lub NULL, gdy  $liczba < 0$

# Funkcje

---

- Napisowe
  - STUFF – usuwa fragment napisu
  - SUBSTRING – zwraca fragment napisu
  - LEN – zwraca długość napisu
  - LEFT, RIGHT – zwraca pewien fragment z lewej/prawej strony napisu

# Funkcje

---

- Systemowe
  - `col_length('tabela', 'kolumna')`
    - maksymalna ilość znaków w kolumnie
  - `datalength(wyrażenie)`
    - rozmiar typu danych wyniku
  - `db_id('nazwa_bazy')`
    - id podanej bazy danych lub bazy bieżącej
  - `db_name(1)`
    - nazwa podanej bazy danych lub bazy bieżącej

# Funkcje

---

- Systemowe
  - `host_id()` – id procesu programu, z którego łączymy się do SQL Servera
  - `host_name()` – nazwa stacji roboczej
  - `user_id()` – id użytkownika bazy danych
  - `user_name()` – nazwa użytkownika bazy
  - `object_id(nazwa_obiektu)` – zwraca id obiektu
  - `object_name(id_obiektu)` – zwraca nazwę obiektu



# Funkcje

---

- Bezparametrowe
  - typu niladic – są po to, żeby SQLServer był zgodny z ANSI SQL-92
  - funkcje i ich odpowiedniki są następujące
    - CURRENT\_TIMESTAMP      GETDATE
    - SYSTEM\_USER              SUSER\_SNAME
    - CURRENT\_USER             USER\_NAME
    - SESSION\_USER             USER\_NAME
    - USER                        USER\_NAME

# Funkcje

---

- Globalne
  - Globalne funkcje typu @@
  - Przykładowe:
    - @@ERROR – numer błędu ostatniego polecenia
    - @@ROWCOUNT – liczba wierszy przetworzonych przez ostatnie polecenie
    - @@VERSION – wersja SQL Servera
    - @@TRANCOUNT – liczba aktywnych transakcji w bieżącej sesji

# Funkcje

---

- Przykład
  - sqlserver-p03-07.sql