


Vysoká škola báňská – Technická univerzita Ostrava
Fakulta strojní, Katedra automatizační techniky a řízení

8

Informační systémy

2008/2009


Radim Farana



1

Obsah

- MS SQL Server 2005,
 - Jazyk Transact-SQL, syntaxe, proměnné, struktury, funkce.
- Doporučená literatura:
Andrew J. Brust, Stephen Forte. *Mistrovství v programování SQL Serveru 2005*. Brno: Computer Press, a.s., 2007. ISBN 978-80-251-1607-4.




Informační systémy

2

Skripty a dávky

- Nastavení aktuální databáze.
- USE DatabaseName
 - USE Prace
- Deklarace proměnných:
- DECLARE @name datatype
 - DECLARE @prom DateTime



Informační systémy

3

Skripty a dávky

- Přřazení hodnoty do proměnné
- SET @name = value
 - SET @prom = @p1 + 1
SELECT @prom
 - SET @prom = (SELECT MAX(p_kusu) FROM prace) -- možný ale nedoporučovaný postup
SELECT @prom
- SELECT @name = value
 - SELECT @prom = MAX(p_kusu) FROM prace
SELECT @prom



Systémové funkce

- Vestavěné funkce
- @@CURSOR_ROWS - počet řádků aktuálního kurzoru
- @@CPU_BUSY - počet milisekund práce od posledního spuštění SQL Serveru
- @@DATEFIRST - vrátí nastavený první den týdne 1 - pondělí, 7 - neděle, je možno nastavit SET DATEFIRST number
- @@DBTS - vrátí aktuální časové razítko (unikátní v rámci databáze)
- @@ERROR - číslo chyby po posledním příkazu T-SQL
- @@FETCH_STATUS - indikuje výsledek posledního použití FETCH (0 - OK, -1 - konec, -2 - záznam není)
- @@IDENTITY - poslední hodnota identity vložená INSERT
- @@IDLE - počet milisekund nečinnosti od posledního spuštění SQL Serveru
- @@IO_BUSY - počet milisekund vstupních a výstupních operací od posledního spuštění SQL Serveru
- @@LANGID - vrátí číslo aktuálního použitého jazyka
- @@LOCK_TIMEOUT - počet milisekund po které bude systém čekat na zablokovaný prostředek
- @@NESTLEVEL - aktuální úroveň vnoření při provádění vnořených procedur
- @@OPTIONS - informace o volbách nastavených příkazem SET
- @@PACK_RECEIVED - počet vstupních paketů od posledního spuštění serveru
- @@REMSERVER - vrací číslo serveru, který vyvolal vnořenou proceduru
- @@ROWCOUNT - počet řádků dotčených posledním příkazem
- @@SERVERNAME - jméno serveru
- @@SERVICENAME - jméno služby SQL Serveru (MSSQLServer)
- @@TIMETICKS - počet milisekund jednoho tiků (31,25)
- @@TOTAL_ERRORS - počet chyb vstupu/výstupu od posledního spuštění
- @@TOTAL_READ - počet členů z disku od posledního spuštění
- @@TOTAL_WRITE - počet zápisů na disk od posledního spuštění
- @@TRANSCOUNT - počet aktivních transakcí v aktuálním spojení
- @@VERSION - verze SQL serveru



Odeslání dávky

- Dávky se odesílají a zpracovávají odděleně.
- Každá dávka musí být plně autonomní.
- Rozpoznání konce dávky (SQL Analyzer)
- GO
 - vždy na samostatném řádku.



Chyby v dávkách

- Syntaktické chyby – zastavení běhu, neprovede se nic.
- Chyby běhu – zastavení běhu, všechny provedené příkazy zůstávají v platnosti.



Využití dávek

- Oddělení operací, které musí proběhnout v určitém pořadí (zajištění referenční integrity apod.)
- Některé příkazy vyžadují samostatné dávky
 - CREATE DEFAULT
 - CREATE PROCEDURE
 - CREATE RULE
 - CREATE TRIGGER
 - CREATE VIEW
- Pozor na správné nastavení implicitní databáze.



Příklad využití dávek

- CREATE DATABASE Test
- GO
- USE Test
- CREATE TABLE TestTable (col1 INT, col2 INT)
- GO
- Databáze musí být nejprve vytvořena, jinak do ní není možno nic vložit.



Generování programového kódu za běhu

- EXEC string definition
- EXECUTE string definition
 - DECLARE @name VarChar(128)
 - SET @name = 'MyTable'
 - EXEC ('SELECT * FROM ' + @name)
- Příkaz pracuje v samostatném oboru platnosti. Proměnné použité v definičním řetězci v něm musí být deklarovány.



Uložení procedury

- Vytvoření uložené procedury
- CREATE PROCEDURE procname
parmame datatype VARYING = default OUTPUT
...
[WITH RECOMPILE|ENCRYPTION]
[FOR REPLICATION]
AS
 program code
GO



Úprava uložené procedury

- Změna uložené procedury:
 - ALTER PROC procname
- Očekává existenci procedury.
- Zachovává všechna existující oprávnění.
- Odstranění uložené procedury:
 - DELETE PROC procname



Parametrizace procedury

- @name [AS] datatype [=default | NULL] [VARYING] [OUTPUT]
- @name – jméno proměnné,
- datatype – datový typ,
- default – výchozí hodnota (proměnná vždy NULL), není-li určena je parametr povinný,
- VARYING - výstupní parametr kurzoru,
- OUTPUT – výstupní parametr.



Příklad – vložení záznamu

- USE prace
GO
- CREATE PROCEDURE VlozPraci
@pracovnik INT,
@typ VARCHAR(50),
@kusu FLOAT,
@datum DATETIME = NULL
AS
INSERT INTO prace
(p_pracovnik, p_typ, p_kusu, p_datum)
VALUES
(@pracovnik, @typ, @kusu, @datum)
GO



Použití procedury

- USE prace
GO
EXEC VlozPraci 153, 'praní', 10,
'04/10/2006'
GO
 - Vloží nový záznam se všemi údaji
- EXEC VlozPraci 153, 'praní', 10
GO
 - Vloží nový záznam, datum bude NULL

Parametry jsou předány
pozičně, nepovinné parametry
mohou být jen na konci seznamu



Vrácení hodnoty parametru

- USE prace
GO
- ALTER PROCEDURE VlozPraci
@pracovnik INT,
@typ VARCHAR(50),
@kusu FLOAT,
@datum DATETIME = NULL,
@ID INT OUTPUT
AS
INSERT INTO prace
(p_pracovnik, p_typ, p_kusu, p_datum)
VALUES
(@pracovnik, @typ, @kusu, @datum)
SELECT @ID = @@identity
GO



Vrácení hodnoty parametru

- USE prace
GO
- DECLARE @MyID INT
EXEC VlozPraci @pracovnik=153,
@typ='praní',
@kusu=2,
@datum='04/11/2006',
@ID=@MyID OUTPUT
PRINT @MyID
GO

Parametry jsou předány odkazem, není nutno určit všechny hodnoty



Návratové hodnoty

- Je možno určit příkazem
- RETURN hodnota
 - Ukončí okamžitě proceduru a vrátí hodnotu jejím názvem
- RETURN
 - Ukončí proceduru a vrátí 0.
- Volání provedury
- DECLARE @MyRet INT
EXEC @MyRet=VlozPraci ...



Příkazy pro řízení toku

- IF .. ELSE
- GOTO
- WHILE
- WAITFOR



IF .. ELSE

- Podmíněná činnost, resp. Rozhodování
- IF podmínka příkaz | BEGIN posloupnost END
ELSE příkaz | BEGIN posloupnost END
- Pozor na testování existence hodnoty
@prom = NULL -- chyba!
@prom IS NULL -- správně



Kontrola hodnoty cizího klíče

- IF EXISTS (SELECT * FROM typyprace
WHERE tp_nazev=@typ)
BEGIN Posloupnost příkazů
INSERT INTO prace
(p_pracovnik, p_typ, p_kusu, p_datum)
VALUES
(@pracovnik, @typ, @kusu, @datum)
SELECT @ID = @@IDENTITY
END
ELSE Jeden příkaz
SELECT @ID = 0



Kontrola hodnoty cizího klíče

- DECLARE @MyID INT
EXEC VlozPraci @pracovnik=153,
@typ='prani',
@kusu=2,
@datum='04/12/2006',
@ID=@MyID OUTPUT
PRINT @MyID
GO Vloženo
- DECLARE @MyID INT
EXEC VlozPraci @pracovnik=153,
@typ='zehleni',
@kusu=2,
@datum='04/12/2006',
@ID=@MyID OUTPUT
PRINT @MyID
GO Bez efektu



Kontrola hodnoty parametru

- IF @datum IS NULL
BEGIN
DECLARE @actdate as datetime
SET @actdate=getdate()
SET @datum=CAST(CAST(DATEPART(m, @actdate) AS
VARCHAR) + '/' + CAST(DATEPART(d, @actdate) AS VARCHAR)+
'/' + CAST(DATEPART(yy, @actdate) AS VARCHAR) AS
DATETIME) CONVERT(datetime, CONVERT(varchar, getdate(), 112))
END 112 – ANSI formát, 101 – US formát obojí plný letopočet
- DECLARE @MyID INT
EXEC VlozPraci @pracovnik=153,
@typ='prani',
@kusu=12,
@ID=@MyID OUTPUT
PRINT @MyID
GO Hodnotu nepovinného parametru
nastaví procedura na dnešní datum



GOTO

- Přejech na určené návěští
- GOTO label
...
label: příkaz
- IF @@ERROR != 0 GOTO err_handler
...
err_handler:



Ošetření chyby

```
USE prace
GO
DECLARE @MyID INT, @MyRet INT
EXEC @MyRet=VlozPraci
@pracovnik=1530,
@typ='prani',
@kusu=12,
@ID=@MyID OUTPUT
PRINT @MyID
PRINT @MyRet
GO
```

Server: Msg 547, Level 16, State 1, Procedure VlozPraci, Line 17
INSERT statement conflicted with COLUMN FOREIGN KEY constraint 'FK_prace_Pracovnici'. The conflict occurred in database 'prace', table 'Pracovnici', column 'pr_osc'.
The statement has been terminated.
-1
547

```
ALTER PROCEDURE VlozPraci
@pracovnik INT,
@typ VARCHAR(50),
@kusu FLOAT,
@datum DATETIME = NULL,
@ID INT OUTPUT
AS
DECLARE @MYERR INT
IF @datum IS NULL
BEGIN
DECLARE @actdate as datetime
SET @actdate=getdate()
SET @datum=CONVERT(datetime, CONVERT(varchar, getdate(), 112))
END
IF EXISTS (SELECT * FROM typyprace WHERE tp_nazev=@typ)
BEGIN
INSERT INTO prace
(p_pracovnik, p_typ, p_kusu, p_datum)
VALUES
(@pracovnik, @typ, @kusu, @datum)
SELECT @MYERR = @@ERROR
IF @MYERR!=0 GOTO err_handler
SELECT @ID = @@IDENTITY
END
ELSE
SELECT @ID = 0
RETURN 0
err_handler:
SELECT @ID = -1
RETURN @MYERR
GO
```



Vynucení hlášení chyby

- Např. pro kontrolu správnosti její obsluhy.
- RAISERROR (ID zprávy | text zprávy, významnost, stav, [argument] [WITH volba])
- ID zprávy – z tabulky (SELECT * FROM master..SysMessages), nové ID vygeneruje nový záznam v této tabulce.



WHILE

- Cyklus
- WHILE podmínka příkaz
- Nebo
- WHILE podmínka
BEGIN
posloupnost příkazů
[BREAK, CONTINUE]
END



Použití cyklů

- Obvykle ve spolupráci s kurzory
- **BREAK** – násilné ukončení cyklu (skok za konec cyklu)
- **CONTINUE** – restart cyklu (skok na začátek cyklu a vyhodnocení podmínky)



Použití cyklu

```
DECLARE @nazev varchar(50), @popis varchar, @cena money

DECLARE listtypyprace CURSOR FORWARD_ONLY
FOR
SELECT [tp_nazev], [tp_popis], [tp_cena] FROM [prace].[dbo].[typyprace]

OPEN listtypyprace

FETCH NEXT FROM listtypyprace
INTO @nazev, @popis, @cena

WHILE @@FETCH_STATUS = 0
BEGIN
PRINT @nazev
FETCH NEXT FROM listtypyprace
INTO @nazev, @popis, @cena
END

CLOSE listtypyprace
DEALLOCATE listtypyprace
GO
```



WAITFOR

- Spuštění operace v určeném čase
- **WAITFOR**
DELAY čas | TIME čas
- **DELAY** – doba čekání, maximálně 24 hod.
- **TIME** – konkrétní časový okamžik dne