



**XMW4 / IW4**

**Modifikace dat, UDF, USP**

Štefan Pataky

# Agenda

- INSERT
- UPDATE
- MERGE
- DELETE
- PROCEDURE
- FUNCTION

# INSERT

- INSERT INTO ... VALUES() vkládá řádek, více řádků se oddělují čárkami

```
INSERT INTO Sales.OrderDetails(  
orderid, productid, unitprice, qty, discount)  
VALUES  
    (12001,39,18,2,0.05),  
    (12002,39,18,5,0.10);
```

- INSERT nemůže vynechat sloupce které jsou NOT NULL bez DEFAULT hodnoty
- V INSERT je možné použít DEFAULT, které vloží DEFAULT hodnotu nebo NULL

```
INSERT INTO Sales.OrderDetails  
    (orderid, productid, unitprice, qty, discount)  
VALUES (11077, 72, 34.80, DEFAULT, DEFAULT);
```

# INSERT SELECT

- INSERT ... SELECT je používáný pro vložení výsledku dotazu do existující tabulky

```
INSERT INTO Sales.OrderHist(  
   orderid,custid,empid,orderdate)  
SELECT orderid,custid,empid,orderdate  
FROM Sales.Orders  
WHERE orderdate < '20080101';
```

- INSERT ... EXEC je používáný pro vložení výsledku uložené procedury nebo výsledku dynamického dotazu

```
INSERT INTO dbo.T1 (productid, productname, unitprice)  
EXEC Production.ProdsByCategory  
    @numrows = 5, @catid=1;
```

# SELECT INTO

- SELECT ... INTO se podobá na INSERT ... SELECT, ale vkládá výsledek do nové tabulky, kterou vytvoří
- Při vytváření kopíruje jména sloupců, datový typ a NULL podmínky

```
SELECT orderid, custid, empid, orderdate, shippeddate  
INTO Sales.OrderArchive  
FROM Sales.Orders  
WHERE orderdate < '20080101';
```

# UPDATE

- UPDATE aktualizuje všechny řádky tabulky nebo pohledu
  - Aktualizace může být filtrovaná pomocí WHERE
  - Aktualizace může být filtrovaná pomocí JOIN
- Aktualizují se jenom sloupce uvedené v SET

```
UPDATE Production.Products  
SET unitprice = (unitprice * 1.04)  
WHERE categoryid = 1 AND discontinued = 0;
```

# MERGE

- MERGE upravuje data na základe podmínek
  - Když se zdroj shoduje s cílem
  - Když se zdroj neshoduje s cílem
  - Když se cíl neshoduje se zdrojem

```
MERGE INTO schema_name.table_name AS TargetTbl
  USING (SELECT <select_list>) AS SourceTbl
  ON (TargetTbl.col1 = SourceTbl.col1)
  WHEN MATCHED THEN
    UPDATE SET col2 = SourceTbl.col2
  WHEN NOT MATCHED THEN
    INSERT (<column_list>)
    VALUES (<value_list>);
```

# DELETE TRUNCATE

- DELETE maže všechny řádky tabulky nebo pohledu
  - Mazání může být filtrované pomocí WHERE
  - Mazání může být filtrované pomocí JOIN
- TRUNCATE vyčistí tabulku, na disku se dealokuje místo
  - Minimální logování

```
TRUNCATE TABLE dbo.Nums;
```



# Uložené procedury

- Uložená procedura – kolekce T-SQL příkazů uložených v databáze
- Může vracet tabulku, upravovat data, a provázet admin dotazy na serveri
- Vykonávání příkazů
  - EXECUTE / EXEC
  - Předání parametrů @jmeno = hodnota

# Uložené procedury

- Vytvoření procedury pomocí CREATE PROCEDURE
- Parametry jsou definovány v hlavičce společně s datovým typem a směrem
- Výchozí směr parametrů je INPUT, lze však definovat i OUTPUT (pro prodaní hodnoty z procedury jinak než výsledkem dotazu)

```
CREATE PROCEDURE <proc_name>  
(@<input_param> AS <type>,  
 @<output_param> AS <type> OUTPUT)  
AS ...
```

```
DECLARE @<output_param> AS <type>;  
EXEC <proc_name> <input_parameter_list>,  
@<output_param> OUTPUT;  
SELECT @output_param;
```

# Funkce

- Rozdělení:
  - Build-in
  - User defined
- Podle vráceného výsledku
  - Skalární
  - tabulkové

# Skalární funkce

- Vrací skalární hodnotu pro zadané parametry
- Nevolat často pokud pracuje s velkým množstvím dat

```
CREATE FUNCTION Add  
(@a AS int ,  
  @b AS int )  
RETURNS INT  
AS  
BEGIN  
    RETURN @a + @b  
END
```

# Vlastní tabulkové funkce

- Vracejí tabulku dat
- Možnost použít v dotazech v části FROM
- Možnost využít parametry

```
CREATE FUNCTION ALL_Customers  
(@COUNTRYID AS int)  
RETURNS TABLE  
AS  
RETURN SELECT ...
```