

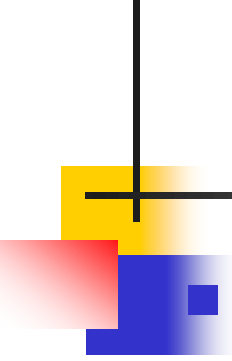
МОДИФИЦИРАНЕ НА ДАННИТЕ



Добавяне на редове в таблици -
INSERT

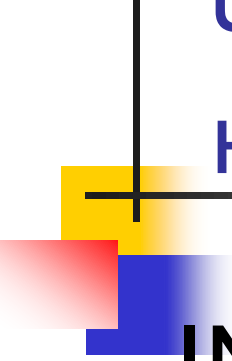
Изтриване на редове в таблици –
DELETE, TRUNCATE TABLE

Обновяване на данните - UPDATE

- 
- При всички операции за модифициране на данни (INSERT, UPDATE, DELETE) се прави контрол върху дефинирания интегритет на данните и ако се нарушава, то операторът се прекратява и се извежда съобщение за грешка.

Добавяне на редове в таблици

- Синтаксис на оператора
- Примери
- Особености при използването на Insert
- Добавяне на стойности по подразбиране
- Добавяне на редове, получени като резултат от SELECT



Частичен синтаксис на оператора Insert

```
INSERT [INTO]  
{table_name | view_name}  
[(column_list)]  
VALUES (  
{expression|DEFAULT |NULL }  
[,....])
```

Пример на Insert

INSERT INTO [Order Details]

(OrderId, ProductId
, UnitPrice
, Quantity, Discount)

VALUES

(10324,6,25.5,125,0.05)
,(10324,7,35,200,0.1)

Някои правила при използване на Insert

1

В списъка се задават имената на колоните, които ще приемат стойности.

- Ако всички колони ще приемат стойности, то задаването на списъка е по желание, но е препоръчително.
- В списъка от колони, в които ще се въвеждат стойности, не трябва да се специфицират колони със свойство **IDENTITY**, тъй като техните стойности се генерират автоматично.

Някои правила при използване на Insert

1

В клаузата **VALUES** се задават стойности за всяка колона от приемащия списък и е необходимо между двата списъка да има съответствие в последователността и типа на елементите от данни.

- При добавяне на повече от един ред с помощта на оператор INSERT отделните списъци със стойности се ограждат в скоби и се разделят със запетая.



```
USE Northwind
```

```
GO
```

```
INSERT [Order Details]
```

```
VALUES (10300, 19, 7.5, 15, 0.05)
```

```
, (10300, 40, 14.7, 10, 0)
```

```
GO
```

Результат ?

Ключова дума Default

Добавя стойността по подразбиране за специфицираните колони

- Ако колоната, за която се използва DEFAULT няма зададена стойност по подразбиране и разрешава неопределена стойност, то се добавя стойност NULL
- Колоните, за които се използва DEFAULT, трябва да имат стойности по подразбиране или да позволяват неопределени стойности

Пример на използване на DEFAULT

USE Northwind

GO

/*В таблицата Order Details се добавя нов ред за поръчка номер 10500, продукт с код 26 на цена 25.5, в количество 1 (зададено по подразбиране) и търговска отстъпка 5% */

INSERT [Order Details]

VALUES (10500, 26, 25.5, DEFAULT, 0.05)

/* показване на вмъкнатия ред */

SELECT * FROM [Order Details]

WHERE OrderID = 10500 AND ProductId = 26

Добавяне на редове, получени
като резултат от SELECT

♦ **INSERT table_name**
(column_list)

(SELECT column_list
FROM table_list
[WHERE search_conditions]
[GROUP BY...[ORDER BY..])

- Създаване на таблица EmployeeOrders, която съдържа колони EmployeeID, LastName, FirstName, Total, Count_Orders, CreateDate

```
CREATE TABLE EmployeeOrders  
(EmployeeID INT, LastName nvarchar(25)  
, FirstName nvarchar(10)  
, Total Money, Count_Orders INT  
, CreateDate Date DEFAULT GetDate())
```

- В таблицата EmployeeOrders да се добавят данни за служителите и общата стойност на приетите от тях поръчки и общия брой на поръчките, които се извличат от съответните таблици на Northwind.

```
INSERT INTO EmployeeOrders
(EmployeeID, LastName, FirstName
, Total, Count_Orders)
(SELECT E.EmployeeID, LastName, FirstName
, SUM(UnitPrice*Quantity), COUNT(O.OrderID)
FROM Employees E JOIN Orders O
ON E.EmployeeID = O.EmployeeID
JOIN [Order Details] OD
ON O.OrderID = OD.OrderID
GROUP BY E.EmployeeID, LastName, FirstName)
/* Показване добавените в EmployeeOrders редове */
SELECT * FROM EmployeeOrders ORDER BY 1
```

Добавяне на данни през изглед

INSERT view_name

Добавят се нови редове към таблица, участваща в изгледа.

- Данни могат да се добавят само към една от таблиците в изгледа.
- Добавяните данни не трябва да нарушават дефинираните свойства за интегритет на данните.
- Всички колони от таблицата, които не позволяват стойност null трябва да приемат стойности, специфицирани в оператора INSERT или от дефиницията DEFAULT.

Добавяне на данни през изглед

USE Northwind

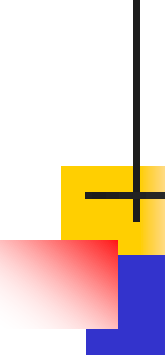
GO

INSERT CustomerOrders

(CustomerId, CompanyName, ContactName)

VALUES

('12347', 'InFocus Company', 'Ivan Petrov')



Пример за добавяне на данни през изглед с използване на колони от две таблици, което води до прекратяване на изпълнението и извеждане на съобщение за грешка:

```
USE Northwind
```

```
GO
```

```
INSERT CustomerOrders (CustomerId, CompanyName,  
ContactName, OrderID)
```

```
VALUES ('12350', 'NetGate Company', 'Jonne Vele',  
10450)
```

Msg 4405, Level 16, State 1, Line 1

View or function 'CustomerOrders' is not updatable
because the modification affects multiple base tables.

Изтриване на редове от таблици

◆ Оператор DELETE

- Действие
- Особености
- Ограничаване броя на изтриваните редове
- Изтриване на редове от таблица въз основа на други таблици

■ Оператор TRUNCATE TABLE

Оператор DELETE

- Използва се за премахване на един или повече редове от таблица

- Синтаксис:

DELETE [FROM] {table_name|view_name}
WHERE search_conditions

Пример: Изтриване на поръчката с номер 10300 за продукт с код 19

```
DELETE FROM [order details]  
WHERE productid=19 AND orderid=10300
```

Изтриване на редове от таблици



- ♦ Всеки изтрит ред се съхранява в дневника на транзакциите - log file
- ♦ Винаги включвайте клауза **WHERE** за да ограничите броя на изтриваните редове

Оператор DELETE

■ Пример: Изтриване на всички поръчки, направени през м.януари 1998



```
DELETE FROM Orders  
WHERE  
MONTH(OrderDate)=1 AND YEAR(OrderDate)=1998
```

Изтриване на редове от таблица въз основа на други таблици

1

- Редове от дадена таблица могат да се изтриват въз основа на условие, поставено върху колони от друга (свързана) таблица
- В този случай в клаузата FROM трябва да се зададе съединяване с другата таблица, за която с WHERE се задава условието като се използва втора клауза FROM.

Изтриване на редове, основано на други таблици - пример

2

- Изтриване на всички редове от таблицата Order Details, които са за поръчки, направени на определена дата (12/10/2002)

```
DELETE FROM [Order details]  
FROM Orders  
INNER JOIN [Order details] OD  
ON Orders.OrderID = OD.OrderID  
WHERE OrderDate = 12/10/2002
```

Използване на допълнителна клауза **FROM**

◆ **DELETE [FROM] table_name**
FROM <table_source> [,...n]
WHERE search_conditions

- ◆ Първата клауза **FROM** индикира таблицата за модификация
- ◆ Втората клауза **FROM** специфицира ограничаващ критерий за оператора **DELETE**

Использование подзапроса в WHERE клаузуле на DELETE

Пример:

```
DELETE FROM [Order details]
WHERE OrderID IN
(SELECT OrderID
FROM Orders
WHERE OrderDate='12/10/2002')
```




Изтриване на данни през изглед

**Използва се само за изгледи върху
една таблица!**

USE Northwind

CREATE CustomerView

AS

SELECT CustomerId, CompanyName **FROM**
Customers

**Изтриване на данните за клиент с код
12345**

DELETE CustomerView

WHERE CustomerID = '12345'

Оператор TRUNCATE TABLE

■ Синтаксис:

■ **TRUNCATE TABLE** [[database.]owner]table_name

- Изтриват се всички редове от таблицата, но се запазва структурата и и асоциираните с нея обекти
- Изтритите редове не се записват в дневника на транзакциите
- Изпълнява се по бързо от DELETE
- Ако има колона IDENTIFY началната и стойност се възстановява

Обновяване на данни

Оператор UPDATE

Синтаксис:

```
UPDATE {table_name|view_name}  
SET {column_name=expression  
    |DEFAULT |NULL} [,...]  
WHERE {search_conditions}
```

Пример: UPDATE Products
SET UnitPrice = (UnitPrice*1.1)

Обновяване на редове

- Клауза WHERE специфицира редовете, в които се прави промяна
- Ключовата дума SET специфицира новите данни
- Входните данни трябва да са от същия тип като колоните, които се обновяват
- SQL Server не обновява редовете, които нарушават ограниченията за интегритет
- С един оператор се обновява само една таблица

Препоръки за ускоряване на обновяването

Използвайте само положителни условия за търсене, т.е. не използвайте NOT IN, NOT BETWEEN, NOT NULL и т.н.

- Избягвайте използването на LIKE
- Използвайте точно съвпадение или интервали
- Клаузата ORDER BY забавя обръщението към данните и не трябва да се използва.

Обновяване, основано на други таблици



```
UPDATE {table_name | view_name}
SET {column_name={expression
|DEFAULT |NULL}
[,...n]
[FROM <table_source>]
[WHERE search_conditions]
```

Обновяване, основано на други таблици

- Използване на клаузата FROM и JOIN на обновяваната таблица с другата таблица или
- Използване на подзаявка; тя се изпълнява по един път за всеки ред от таблицата, която се обновява, като за всеки ред се връща единична стойност.
- Даден ред се обновява само един път.

Пример

- Обновяване на таблицата Products чрез увеличаване с 2% цената на всички продукти от категория Seafood

UPDATE Products

SET UnitPrice=UnitPrice*1.02

FROM Products PR JOIN Categories G

ON PR.CategoryID=G.CategoryID

WHERE G.CategoryName = 'Seafood'

Пример на UPDATE с използване на подзаявка

Обновяване на таблицата Products
чрез намаляване с 2% цената на
всички продукти от категория **Seafood**

UPDATE Products

SET UnitPrice=UnitPrice*0.98

WHERE CategoryID =

(SELECT CategoryID FROM Categories
WHERE CategoryName ='Seafood')

Пример на UPDATE с използване на корелационна подзаявка

- Обновяване на колона total в таблицата Orders със сумарната стойност на поръчаните стоки за всяка поръчка

UPDATE Orders

**SET Total = (SELECT
SUM(UnitPrice*Quantity)
FROM [Order Details] OD
WHERE Orders.OrderID=OD.OrderID)**



Използване на подзаявка с EXISTS

Намаляване с 5% цената на всички продукти, които не са поръчвани досега

USE Northwind

UPDATE Products

SET UnitPrice = UnitPrice*0.95

WHERE NOT EXISTS

(SELECT * FROM [Order Details] AS OD

WHERE Products.ProductId =
OD.ProductId)

Обновяване на данни през изгледи

- ✓ Оператор UPDATE view_name.....
- ✓ Ако изгледът е върху две или повече таблици, то могат да се правят промени само в една таблица.
- ✓ Колоните, изброени в оператора UPDATE, трябва да принадлежат на една и съща таблица от дефиницията на изгледа.
- ✓ **Ако изгледът е върху свързани таблици, то кои редове ще се обновяват се определя от връзката между таблиците.**

Модифициране на данни през изгледи

- ✓ Всички колони от таблицата, които се актуализират и не позволяват стойност null приемат стойности, специфицирани в оператора INSERT или от дефиницията DEFAULT.
- ✓ Модифицираните данни в колоните трябва да удовлетворяват ограниченията за тези колони, като например тези за FOREIGN KEY.



Промяна на данни през изглед

```
USE Northwind
```

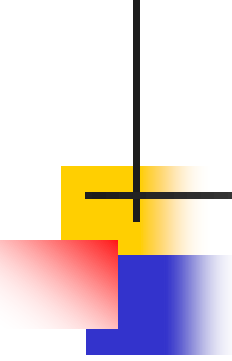
```
GO
```

```
UPDATE CustomersOrders
```

```
SET OrderDate = OrderDate - 5
```

```
WHERE CompanyName LIKE 'Bon%'
```

- Променя се датата на поръчките за клиентите с име, започващо с Bon



Промяна на цената на стоките в таблицата Products, за които са били направени поръчки, през изгледа ProductsOrders

USE Northwind

GO

UPDATE ProductsOrders

SET UnitPrice = [UnitPrice]*1.02

- В таблицата Products ще се увеличи цената с 2% само на тези стоки, за които има данни за поръчки в таблицата Order Details.

Клауза OUTPUT

INSERT INTO [Order Details]

(OrderId, ProductId, UnitPrice, Quantity,
Discount)

OUTPUT INSERTED.* -- показва
добавените записи

VALUES(10324,6,25.5,1,0.05)
, (10324,7,35,2,0.1)

Клауза OUTPUT

```
USE Northwind
```

```
GO
```

```
DELETE FROM [Order Details]
```

```
OUTPUT DELETED.* -- показва
```

```
изтритите записи
```

```
WHERE OrderID IN
```

```
(SELECT OrderID FROM Orders
```

```
WHERE OrderDate='05/23/2013')
```

Клауза OUTPUT

USE Northwind

GO

UPDATE Products

SET UnitPrice=UnitPrice*1.05

OUTPUT DELETED.* --показва записите,
които са обновени преди обновяването им
FROM Products Pr INNER JOIN Categories Ca
ON Pr.CategoryID=Ca.CategoryID
WHERE CategoryName = 'Seafood'