

Язык SQL

Инструкции манипулирования
данными

Инструкции манипулирования данными (Data Manipulation Language, DML)

В состав DML входят инструкции:

- ✓ **INSERT**
- ✓ **UPDATE**
- ✓ **DELETE**
- ✓ **TRUNCATE**
- ✓ **MERGE**
- ✓ **SELECT**

Добавление данных в таблицу

Инструкция **INSERT** применяется для добавления в таблицу строк, формируемых из заданных значений

```
INSERT INTO имя_таблицы(список_столбцов)  
VALUES (список_значений) ;
```

Пример

```
USE testdb;

IF OBJECT_ID('dbo.Orders', 'U') IS NOT NULL
    DROP TABLE dbo.Orders;

CREATE TABLE dbo.Orders
(
    orderid INT          CONSTRAINT PK_ORDERS PRIMARY KEY,
    empid    INT          NOT NULL,
    custid   VARCHAR(10)  NOT NULL,
    orderts  DATETIME     NOT NULL,
    qty      INT          NOT NULL,
    CONSTRAINT FK_Orders_Emp FOREIGN KEY (empid)
    REFERENCES dbo.Employees(empid)
);
```

Пример (продолжение)

```
INSERT INTO dbo.Orders  
        (orderid, empid, custid, orderts, qty)  
VALUES (10001, 3, 'A', '20130930', 1000);
```



```
INSERT INTO dbo.Orders  
VALUES (10001, 3, 'A', '20130930', 1000);
```

Пример (продолжение)

```
INSERT INTO dbo.Orders  
        (orderid, empid, custid, qty)  
VALUES (10002, 3, 'B', 10000);
```

Пример (продолжение)

```
INSERT INTO dbo.Orders
        (orderid, empid, custid, qty)
VALUES  (10003, 3, 'C', 10000),
        (10004, 3, 'D', 20000),
        (10005, 3, 'E', 30000),
        (10006, 3, 'F', 40000);
```

Добавление данных в таблицу из файла

Инструкция **BULK INSERT** применяется для вставки в существующую таблицу данных, хранящихся в файле

```
BULK INSERT dbo.Orders FROM 'C:\Temp\orders.txt'  
WITH (DATAFILETYPE = 'char',  
      FIELDTERMINATOR = ',',  
      ROWTERMINATOR = '\n');
```

```
C:\Temp\orders.txt:  
10007,1,G,20130930,340  
10008,1,Н,20130930,207  
10009,1,I,20130930,123456  
10010,1,J,20130930,654321
```


Пример

```
BULK INSERT dbo.Orders FROM 'C:\Temp\orders.txt'  
WITH (DATAFILETYPE = 'char',  
      FIELDTERMINATOR = ',',  
      ROWTERMINATOR = '\n');
```

C:\Temp\orders.txt:

10007,1,G,,340

10008,1,H,,207

10009,1,I,,123456

10010,1,J,,654321

Свойство IDENTITY

- Может быть определено для столбца с любым целочисленным типом данных
- При выполнении инструкции **INSERT** вызывает автоматическую генерацию значений, исходя из начального значения и величины приращения
- Применяется при создании суррогатных ключей

Пример

```
CREATE TABLE dbo.Orders
(
   orderid INT IDENTITY(1,1) CONSTRAINT PK_ORDERS
                                PRIMARY KEY,
    empid   INT                NOT NULL,
    custid  VARCHAR(10)        NOT NULL,
    orderts DATETIME           NOT NULL,
    qty     INT                NOT NULL,
    CONSTRAINT FK_Orders_Emp FOREIGN KEY (empid)
    REFERENCES dbo.Employees(empid)
);
```

Пример (продолжение)

```
INSERT INTO dbo.Orders (empid, custid, qty)
VALUES (3, 'C', 10000),
       (3, 'D', 20000),
       (3, 'E', 30000),
       (3, 'F', 40000);
```

orderid	empid	custid	orderts	qty
1	3	C	2013-09-30 00:00:00.000	10000
2	3	D	2013-09-30 00:00:00.000	20000
3	3	E	2013-09-30 00:00:00.000	30000
4	3	F	2013-09-30 00:00:00.000	40000

Пример

```
INSERT INTO dbo.Orders (orderid, empid, custid, qty)
VALUES (10030, 1, 'C', 10000);
```

Невозможно вставить явное значение для столбца идентификаторов в таблице "Orders", когда параметр `IDENTITY_INSERT` имеет значение `OFF`.

```
SET IDENTITY_INSERT dbo.Orders ON;
INSERT INTO dbo.Orders (orderid, empid, custid, qty)
VALUES (10030, 1, 'C', 10000);
SET IDENTITY_INSERT dbo.Orders OFF;
```

Удаление данных из таблицы

➤ **DELETE**

Удаляет данные из таблицы на основе условия

➤ **TRUNCATE**

Удаляет все строки из таблицы

Инструкция DELETE

```
USE testdb;
```

```
DELETE FROM dbo.Orders
```

```
WHERE orderts < '20130930';
```

Инструкция TRUNCATE

```
USE testdb;
```

```
TRUNCATE TABLE dbo.Orders;
```


Обновление данных в таблицах

Инструкция **UPDATE** позволяет обновлять подмножество строк в таблице

```
USE testdb;
```

```
UPDATE dbo.Orders
```

```
    SET qty = 1000000
```

```
WHERE orderid = 10001;
```

Обновление данных в таблицах

Поддерживаются составные операторы
присваивания:

`+=, -=, *=, /=, %=`

```
USE testdb;
```

```
UPDATE dbo.Orders
```

```
    SET qty += qty
```

```
WHERE orderid = 10002;
```

Обновление данных в таблицах

```
USE testdb;
```

```
UPDATE dbo.Orders  
    SET qty = qty + 100;
```

Одновременность выполнения операций

```
UPDATE dbo.T1
```

```
SET col1 = col1 + 10, col2 = col1 + 10;
```

col1 = 100 col2 = 200

Обычное выполнение операций	SQL (одновременное выполнение операций)
col1 = 100 + 10 = 110 col2 = 110 + 10 = 120	col1 = 100 + 10 = 110 col2 = 100 + 10 = 110

Одновременность выполнения операций

```
UPDATE dbo.T1
```

```
SET col1 = col2, col2 = col1;
```

```
col1 = 100 col2 = 200
```

Обычное выполнение операций	SQL (одновременное выполнение операций)
col1 = col2 = 200 col2 = col1 = 200	col1 = col2 = 200 col2 = col1 = 100