

# Коллективная работа

## Немного информации

В этой части вы узнаете о:

- Командах `WRITETEXT` и `UPDATETEXT`
- Ограничениях команд `INSERT`, `DELETE` и `UPDATE`
- Определении первичного ключа

## WRITETEXT и UPDATETEXT КОМАНДЫ

### Команда WRITETEXT

Команда WRITETEXT используется для обновления значения в столбцах text, ntext или image. Эта команда используется, когда вы не хотите использовать команду UPDATE. Поскольку команда UPDATE протоколирует изменения, сделанные в таблице, в журнале транзакций.

В отличие от команды UPDATE, при использовании команда WRITETEXT нельзя напрямую обновлять данные в таблице при помощи команды SET. Команда WRITETEXT требует указателя на столбцы text, ntext или image, которые будут обновляться. Для нахождения текстового указателя для конкретных строк и столбцов используется функция TEXTPTR().

# Коллективная работа

## WRITETEXT и UPDATETEXT команды (продолжение..)

### Команда WRITETEXT

#### Синтаксис

```
WRITETEXT {имятабл.столбец текст_ук}  
  [WITH LOG] {данные}
```

где,

имятаблицы.столбец – таблица и столбец, которые обновляются.

тест\_ук – указатель данных text, ntext or image для обновления.

WITH LOG – это опциональное выражение, которое позволяет протоколировать обновление.

Протоколирование находится в зависимости от модели извлечения, применяемой для базы данных. Например, команда WRITETEXT полностью протоколируется, когда в базе данных используется полная модель извлечения и команда не протоколируется, когда в базе данных используется простая

модель извлечения

данные – это обновляемые величины.

# Коллективная работа

## WRITETEXT и UPDATETEXT команды (продолжение..)

### Команда WRITETEXT

Для использования команды WRITETEXT в режиме непротоколирования, необходимо разрешение select into/bulkcopy. Для установки разрешения select into/bulkcopy используется системная процедура sp\_dboption.

# Коллективная работа

## WRITETEXT и UPDATETEXT команды (продолжение..)

### Команда UPDATETEXT

Команда UPDATETEXT используется для обновления значения в полях столбца text, netext или image. Команда UPDATETEXT имеет больше опций, чем WRITETEXT.

# Коллективная работа

## WRITETEXT и UPDATETEXT команды (продолжение..)

### Команда UPDATETEXT

#### Синтаксис

```
UPDATETEXT { имятаблицы.столбец текст_ук }  
  { NULL | вставка_смещения }  
  { NULL | удал_длина }  
  [ WITH LOG ]  
  [ данные  
    | { исх_имятабл.столбец исх_текст_ук } ]
```

где,

имятаблицы.столбец – таблица и столбец, которые обновляются.

текст\_ук – указатель данных text, ntext или image, которые обновляются.

вставка\_смещение – нулевая начальная позиция для обновления. Это позиция, с которой начинается вставка.

# Коллективная работа

## WRITETEXT и UPDATETEXT команды (продолжение..)

### Команда UPDATETEXT

Удал\_длина – длина данных в байтах или символах, которые будут удалены начиная с позиции, заданной как вставка\_смещение.

WITH LOG – это опциональное выражение, которое позволяет протоколировать обновление данных. Протоколирование находится в зависимости от модели извлечения, применяемой для базы данных. Например, команда WRITETEXT полностью протоколируется, когда в базе данных используется полная модель извлечения и команда не протоколируется, когда в базе данных используется простая модель извлечения.

данные – это обновляемые величины.

scr\_tablename.column is the name of the source tablename and column from where data is to be inserted.

src\_text\_ptr is the pointer to scr\_tablename.column.

# Коллективная работа

## Ограничения команд INSERT, DELETE, UPDATE и SELECT INTO

### Ограничения команды INSERT

#### Команда INSERT

- Не может вставлять записи через представление, если представление необновляемое.
- Может вставлять данные только в одну таблицу базы, которая представляется ссылками.
- Пытается вставить значения в каждый столбец таблицы, если имя таблицы не задано.
- Не может вставить в столбец IDENTITY, если не включена (ON) опция SET IDENTITY\_INSERT.
- Вставляет значение NULL, если ключевое слово DEFAULT вложено для столбца, не имеющего умолчания, но нулевого.
- Вызывает ошибку, если ключевое слово DEFAULT вложено для столбца, не имеющего умолчания, и не нулевого.
- Не может вставить строки в таблицу, которая имеет действие INSTEAD OF INSERT. В таком случае, действие выполняется.

# Коллективная работа

## Ограничения команд INSERT, DELETE, UPDATE и SELECT INTO (продолжение..)

### Ограничения команды DELETE

#### Команда DELETE

- Не может удалять записи через представление, если представление необновляемо.
- Не может удалять данные, если представление команды содержит более чем одну таблицу в своем выражении FROM, используемого для представления.
- Удаляет все строки таблицы назначения, если в команде опущено выражение WHERE.
- Не может удалять строки в таблице, которая имеет действие INSTEAD OF DELETE. В таком случае, действие выполняется.
- Не может удалять, если команда нарушает ограничение на таблицу. При удалении нескольких строк, даже если одна строка имеет ограничения, не удалится ни одна из строк.

# Коллективная работа

## Ограничения команд INSERT, DELETE, UPDATE и SELECT INTO (продолжение..)

### Ограничения команды DELETE

#### Команда DELETE

- Не может удалять строки из таблицы на нулевой стороне внешнего соединения. Давайте рассмотрим это ограничение команды DELETE на примере.  
Пусть имеются две таблицы, клиенты и заказы. Есть некоторые клиенты, которые не разместили заказ. В случае внешнего соединения этих двух таблиц, таблица заказов будет содержать значения NULL против клиентов, которые не разместили заказ. Команда DELETE с последующим выражением FROM не сможет использоваться для удаления строк и таблицы заказов.  
FROM clients LEFT JOIN orders ON  
clients.client\_id = orders.client\_id

# Коллективная работа

## Ограничения команд INSERT, DELETE, UPDATE и SELECT INTO (продолжение..)

### Ограничения команды UPDATE

Команда UPDATE не может обновить

- Более чем одну таблицу в представлении.
- таблицы через представление, если представление не обновляемо.
- Представление, основанное на идентичности столбцов таблицы.
- Если несколько значений вложено в выражение команды.
- если команда нарушает ограничение на таблицу. При обновлении нескольких строк, даже если одна строка имеет ограничения, ни одна из строк не обновится.
- Если в команде сгенерирована арифметическая ошибка.
- Несколько строк при обновлении столбца text или image
- строки в таблице, которая имеет действие INSTEAD OF UPDATE. В таком случае, действие выполняется.

# Коллективная работа

## Ограничения команд INSERT, DELETE, UPDATE и SELECT INTO (продолжение..)

### Ограничения оператора SELECT INTO

- Для создания таблицы и ее заполнения данными, команды SELECT INTO не использовать. Из за того, что таблицы, созданные при помощи команд SELECT INTO не поддерживают:
  - индексы
  - Первичные ключи
  - Внешние ключи
  - Значения по умолчанию
  - действия
- Для использования этих функций необходимо применять команду ALTER TABLE для таблицы, созданной при помощи команды SELECT INTO.
- Команда SELECT INTO не может быть исполнена в середине транзакции.
- Команда SELECT INTO берет результат команды SELECT в постоянную таблицу.

## Определение первичного ключа

Первичный ключ, это ключ, уникально определяющий каждую запись таблицы. Уникальность первичного ключа обеспечивает целостность таблицы. Поэтому, определение «хороших» первичных ключей является важной задачей. Однако, если в вашей таблице неттакого поля, в качестве первичного можно использовать свернутый ключ.

### **Свернутый ключ**

Свернутый ключ, это уникальный ключ, определяющий уникальным образом каждую запись таблицы. Однако этот ключ не является частью данных таблицы. Например, таблица SUPPLIERS содержит такие сведения о поставщиках, как имя, адрес и товар. Однако там нет ключа, который может уникальным образом идентифицировать записи в таблице. В такой ситуации, можно определить свернутый ключ, `supplier_id`.

### **Потенциальные ключи**

Можно использовать комбинацию ключей, которая может уникально определить записи в таблице, как первичный ключ. Ключи, которые можно использовать в комбинации вместо первичного ключа, называются потенциальными ключами. Если вернуться к рассматриваемому примеру, то в таблице SUPPLIERS потенциальными первичного ключа может использоваться комбинация имени и адреса поставщика.

## Советы по выбору потенциальных ключей

- 1: Комбинация потенциальных ключей должна быть такова, чтобы не было или сведена к минимуму возможность изменения данных.
- 2: Количество потенциальных ключей должно быть меньше размерности таблицы. Попробуйте свести их к минимуму.
- 3: Потенциальные ключи должны подходить устройству таблицы.

# Коллективная работа

## Слово эксперта

В данном разделе даны ответы на следующие вопросы:

- Наглядные примеры вставки значения в столбец IDENTITY
- Советы по использованию функции NEWID()
- Советы по обновлению данных при помощи выражений FROM и TOP
- FAQ по таблицам

# Коллективная работа

## Наглядные примеры

### Использование WITH TIES в команде TOP

Рассмотрим сценарий. Роберт создает таблицу заказов (Orders). Он определил Order\_Id в качестве столбца IDENTITY. Он хочет указать значение для столбца. Роберт не может просто вставить значения в столбец IDENTITY. Он может сделать так, только если включена (ON) опция SET IDENTITY\_INSERT.

#### Пример

```
SET IDENTITY_INSERT Orders ON
INSERT INTO Orders
(Order_ID, Customer_ID, Order_date)
VALUES (089, C34, '2/21/20004')
```

# Коллективная работа

## Советы

### Использование функции NEWID()

- Вы можете использовать функцию NEWID() для создания уникального значения и назначить его уникально идентифицированному типу данных. Вы можете использовать эту функцию вместе с командой CREATE TABLE для создания уникального идентификатора для каждой строки таблицы.
- Эта функция возвращает GUID. Этот GUID сгенерирован при помощи сложной формулы, зависящей от аппаратных характеристик, случайного числа и информации о дате и времени. В результате, каждый GUID уникальный, без привязки ко времени и компьютеру.

# Коллективная работа

## Советы (Продолжение..)

### Использование функции NEWID()

#### Синтаксис

```
NEWID( )
```

#### Пример

```
CREATE TABLE employees  
(  
  EmpCode uniqueidentifier default NEWID(),  
  EmpName char(25) NOT NULL,  
  DeptName char(25) NOT NULL  
)
```

# Коллективная работа

## Советы (Продолжение..)

### Обновление данных при помощи выражений FROM и TOP

Для обновления данных в таблицах вы можете использовать выражения FROM. Это можно сделать, вытаскивая данные из представлений или других таблиц.

#### **Пример**

Пусть в базе данных издательского дома имеются три таблицы: authors, (авторы), titles (названия) и titleauthor (названиеавтор). Таблица author содержит сведения об авторах, а таблица titles – о названиях. В таблице titleauthor хранятся сведения и о названиях, и об авторах. Пусть требуется обновить таблицу titleauthor номером title\_identification, назначенном книге Net Etiquette в таблице titles.

# Коллективная работа

## Советы (продолжение..)

### Обновление данных при помощи выражений FROM и TOP

```
UPDATE titleauthor
  SET title_id = titles.title_id
  FROM titles INNER JOIN titleauthor
    ON titles.title_id = titleauthor.title_id
    INNER JOIN authors
    ON titleauthor.au_id = authors.au_id
  WHERE titles.title = 'Net Etiquette'
    AND au_lname = 'Jack'
```

Подобным же образом для обновления данных в таблицах вы можете использовать выражения FROM и TOP.

# Коллективная работа

## Советы (Продолжение..)

### Обновление данных при помощи выражений FROM и TOP

#### Пример

Рассмотрим другой пример. Требуется обновить столбец state (штат) для первых 10 авторов таблицы authors.

```
UPDATE authors
SET state = 'ZZ' FROM (SELECT TOP 10 *
FROM authors ORDER BY au_lname) AS t1
WHERE authors.au_id = t1.au_id
```

# Коллективная работа

## FAQ по запросам

- В.* При создании таблицы выдается сообщение об ошибке 2714:  
Общий размер строк таблицы превышает макс. число байт на строку (8060). (8060).  
Строки, превышающие макс. число байт не будут добавлены.  
Однако, таблица успешно создается и можно вставлять данные без проблем. Что означает это сообщение?
- О.* Это сообщение об ошибке показывает, что у вас различная длина столбцов в таблице, как `nvarchar` или `varbinary`. Общая максимальная длина всех этих столбцов в сумме больше 8060 байт. Вы все же можете вставлять данные в таблицу, если общая длина данных в каждой строке не превышает 8060 байт. Однако, если данные не превышают 8060 байт, при вставке будет сообщение об ошибке:  
Server: Msg 511, Level 16, State 1, Line 5  
Cannot create a row of size <rowlength> which is greater than the allowable maximum of 8060.  
The statement has been terminated.

# Коллективная работа

## FAQs (продолжение..)

- В. Что такое столбцы `IDENTITY`?
- О. Столбцы `IDENTITY` – это такие столбцы в таблице, которым необходимы автоматически сгенерированные уникальные системные значения. Например: Код служащего для служащих организации.
  
- В. Как я могу убедиться, что заново добавленное ограничение применено к существующим данным в таблице?
- О. Для указания того, должны ли проверяться существующие данные на введенное или повторное включенное ограничение, необходимо воспользоваться выражением `WITH CHECK` в команде `ALTER TABLE`.

# Коллективная работа

## FAQs (продолжение..)

В. Почему не работает команда SELECT INTO?

О. Прежде всего, проверьте правильность записи команды SELECT INTO. Если она верна, проверьте, включена ли опция базы данных `select into/bulkcopy`.

Чтобы узнать, включена ли опция базы данных `select into/bulkcopy`, можно использовать Enterprise Manager или системную процедуру `sp_dboption`.

Например, чтобы проверить включение опции базы `select into/bulkcopy` для `pubs`, можно использовать команду:

```
EXEC sp_dboption pubs, 'select into/bulkcopy'
```

Включить опцию базы `select into/bulkcopy` можно командой:

```
EXEC sp_dboption 'pubs', 'select into/bulkcopy', 'TRUE'
```

# Коллективная работа

## FAQs (продолжение..)

- В. Как можно получить список системных таблиц и представлений основной базы данных?
- О. `Select type, name From Master..sysObjects`  
где `Type In ('S', 'V')`  
`AND name Not In (Select name From Model..sysObjects) order by type, name`

# Коллективная работа

## СРС 5\_4

1. Какая команда позволяет копировать содержимое одной таблицы в другую?
2. Том Аллен желает проанализировать 400 строк таблицы, содержащей 1000000 строк. Ему не нужны эти данные для постоянного хранения. Он использует команду `SELECT INTO`. Какой символ он использует с командой `SELECT INTO`, чтобы получить желаемый результат?
3. Вы хотите использовать первичный ключ для уникальной идентификации всех студентов в группе и проверить, что введенное значение попадает в особый диапазон и неотрицательно. Какое ограничение(я) поможет это сделать?
4. Какая функция может использоваться для извлечения указателя текста для конкретной строки или столбца таблицы?
5. В таблице допускается максимум \_\_\_\_\_ столбцов.

# Коллективная работа

## СРС 5\_4 (продолжение..)

6. \_\_\_\_\_ начальное значение столбца IDENTITY.
7. В чем разница между ограничениями PRIMARY KEY и UNIQUE?
8. Одно ограничение CHECK может быть применено к нескольким столбцам, когда оно определено на:
  - a. Уровне столбца
  - b. Уровне таблицы
9. Какая системная процедура используется для назначения и снятия правила?
10. Какое ограничение обеспечивает целостность домена налая ограничение на значение, вставляемое в столбец?
11. Какую обцию необходимо установить для работы команды WRITETEXT?