

## Практическая работа 19

### Модификация базы данных

**Цель занятия:** Получить практический опыт выполнения модификации базы данных.

#### Перечень оборудования и программного обеспечения

Персональный компьютер  
Microsoft Office (Word, Visio, Access)  
Microsoft SQL Server Management Studio

#### Краткие теоретические сведения

##### Модификация объектов базы данных

Для модификации (изменения) отдельных объектов базы данных в SQL языке есть команды и основанные на них sql запросы: SQL ALTER TABLE. Применяется эта команда (запрос), когда нужно сохранить данные в таблицы, и лишь исправить (модифицировать) некоторые из них.

Изменить отдельные данные в таблице базы данных можно на основе запроса sql ALTER TABLE.

**Команда ALTER** относится к подмножеству SQL, языку DDL, — языку определения данных. С её помощью можно модифицировать не только таблицы, но и процедуры, пользователей, представления, табличные области. Есть целое семейство таких команд: ALTER TABLE, ALTER VIEW, ALTER PROCEDURE, ALTER TRIGGER, ALTER USER, ALTER ROLE.

С помощью использования команды ALTER можно:

- Добавить столбец в таблицу;
- Добавить ограничение целостности;
- Изменить тип данных в столбце таблицы, его размер, значение по умолчанию (переопределить столбец);
- Удалить столбец;
- Включить, выключить, удалить триггер или ограничение целостности.

Целостность БД- любое отношение должно иметь первичный ключ и для каждого внешнего ключа должен существовать первичный ключ.

Важно! Для использования команды ALTER пользователь должен иметь привилегии ALTER, INSERT и CREATE для этой таблицы.

#### Команда SQL ALTER TABLE, синтаксис команды

ALTER TABLE [IGNORE] имя\_таблицы опции\_модификации\_таблицы  
IGNORE необязательная опция для защиты первичного ключа.

## Примеры использования команды ALTER TABLE

### № 1 Добавляем столбец в таблицу

```
ALTER TABLE client(client_site varchar(10));//Добавить в таблицу client столбец client_site//
```

### № 2 Меняем размер поля столбца

```
ALTER TABLE client(client_passwd varchar(25));//Изменение размера поля столбца client_passwd//
```

### № 3 Меняем размер поля столбца с использованием modify

```
ALTER TABLE client modify client_passwd varchar(25);//Изменение размера поля столбца client_passwd//
```

### № 4 Добавляем столбец, после определенного столбца

```
ALTER TABLE client add client_site varchar(50) after client_telefon;//Добавить в таблицу client столбец client_site, после столбца client_telefon//
```

Чтобы изменить ограничение FOREIGN KEY с помощью Transact-SQL, сначала необходимо удалить существующее ограничение FOREIGN KEY, а затем повторно создать его с новым определением.

```
ALTER TABLE имя_таблицы DROP имя_ключа;
```

```
ALTER TABLE имя_таблицы
```

```
ADD CONSTRAINT имя_ключа FOREIGN KEY(имя_поля) REFERENCES имя_таблицы_для_связи(имя_поля);
```

## Обновление данных

**Операция обновления данных UPDATE** требуется тогда, когда происходят изменения во внешнем мире и их надо адекватно отразить в базе данных, так как надо всегда помнить, что база данных отражает некоторую предметную область.

Операция обновления **UPDATE** имеет следующий формат:

```
UPDATE имя_таблицы
```

```
SET {имя_столбца = новое_значение [...]}
```

```
[WHERE условие_отбора]
```

Выражение, которое находится в фигурных скобках означает, что мы за одну операцию обновления можем изменить сразу несколько столбцов, при этом ключевое слово SET не повторяется, а выражение «имя\_столбца = новое\_значение» может повторяться несколько раз для разных столбцов, и при этом каждая пара отделяется от следующей разделителем — запятой.

Часть WHERE является необязательной, как и в операторе DELETE. Она играет здесь ту же роль, что и в операторе DELETE, — позволяет отобрать строки, к которым будет применена операция модификации. Если условие отбора не задается, то операция модификации будет применена ко всем строкам таблицы.

Пример.

```
UPDATE R1
SET R1.Оценка = 4
WHERE R1.ФИО = «Степанова» AND R1.Дисциплина = «Базы данных»
```

**Оператор удаления данных DELETE** позволяет удалить одну или несколько строк из таблицы в соответствии с условиями, которые задаются для удаляемых строк.

Синтаксис оператора DELETE следующий:

```
DELETE FROM имя_таблицы [WHERE условия_отбора]
```

Если условия отбора не задаются, то из таблицы удаляются все строки. Однако это не означает, что удаляется вся таблица. Исходная таблица остается, но она остается пустой, незаполненной.

Например, если надо удалить все строки таблицы R1:

```
DELETE FROM R1
```

Условия отбора в части WHERE имеют тот же вид, что и условия фильтрации в операторе SELECT. Эти условия определяют, какие строки из исходного отношения будут удалены.

```
DELETE FROM R2 WHERE ФИО = «Миронов»
```

В части WHERE кроме простых условий фильтрации может находиться также встроенный запрос.

Пример

```
DELETE FROM R2
WHERE R2. ФИО IN
(SELECT R1. ФИО
FROM R1
WHERE Оценка = 2 OR Оценка IS NULL
```

GROUP BY R1. ФИО  
HAVING COUNT(\*) >= 2)

## Задания

- 1 Изучить теоретические сведения.
- 2 В соответствии с вариантом задания выполнить операции модификации структуры базы данных:
  - внешние ключи по необходимости;
  - свойства полей;
  - добавление и удаление полей таблиц.
- 3 В соответствии с вариантом задания выполнить операции модификации данных в БД:
  - обновление записей;
  - удаление записей.

## Порядок выполнения работы

### Обновление ключей

Рассмотрим базу данных библиотеки.

Предположим, что в ней есть раздел художественной литературы, и он передается в другую библиотеку или в другой филиал. Тогда из базы данных он должен быть удален, а значит и все книги этого раздела каскадно удаляются. Но эти книги выдавались, если определить каскадное удаление в таблицу выдачи, нарушится порядок работы библиотеки.

Или другой пример, если книга списана и удалена из базы данных, но до этого времени выдавалась читателям, то в статистике работы библиотеки такая запись учитывается. Каскадное удаление данных по смыслу подходит не всегда, здесь при удалении книги нельзя удалять ее из выдачи, т.к. это нарушит статистику деятельности библиотеки, посмотрим другие способы соблюдения целостности данных:

**ON DELETE { NO ACTION | CASCADE | SET NULL | SET DEFAULT }**

Указывает, какое действие производится над строками изменяемой таблицы, если эти строки имеют ссылочную связь, и строка, на которую имеется ссылка, удаляется из родительской таблицы. Параметр по умолчанию — NO ACTION.

**NO ACTION**

Компонент SQL Server Database Engine формирует ошибку, и выполняется откат операции удаления строки из родительской таблицы.

**CASCADE**

Если из родительской таблицы удаляется строка, соответствующие ей строки удаляются и из ссылающейся таблицы.

#### SET NULL

Все значения, составляющие внешний ключ, при удалении соответствующей строки родительской таблицы устанавливаются в значение NULL. Для выполнения этого ограничения внешние ключевые столбцы должны допускать значения NULL.

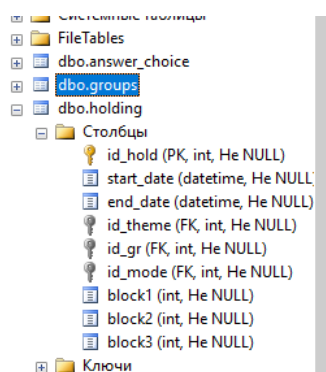
#### SET DEFAULT

Все значения, содержащие внешний ключ, при удалении соответствующей строки родительской таблицы устанавливаются в значения по умолчанию. Для выполнения этого ограничения все внешние ключевые столбцы должны иметь определения по умолчанию. Если столбец допускает значения NULL и значение по умолчанию явно не определено, значением столбца по умолчанию становится NULL.

```
FOREIGN KEY (i_num) REFERENCES izdania (i_num)
ON DELETE SET DEFAULT
ON UPDATE CASCADE
```

Вместо ON DELETE CASCADE лучше использовать ON DELETE SET DEFAULT, но предварительно задать начальное значение в описании поля:

Рассмотрим пример базы данных «Тестирование». В ней имеется таблица «Проведение тестирования» для назначения даты, группы и формы проведения теста.



Здесь в случае удаления группы необходимо, чтобы запись осталась для контроля проведения промежуточной аттестации, аналогично для поля id\_mode (порядок проведения) и поля темы. Значит:

- удаляем ограничение внешнего ключа (имя внешнего ключа можно увидеть в обозревателе объектов у нужной таблицы в папке Ключи);
  - добавляем ограничение значения по умолчанию;
- добавляем внешний ключ снова, но с характеристиками ON DELETE SET DEFAULT, ON UPDATE CASCADE.

```

31
32 ALTER TABLE holding DROP CONSTRAINT FK_holding_id_gr_6C190E8B;
33
34 ALTER TABLE holding
35 ADD CONSTRAINT DF_holding_idgr
36 DEFAULT 1 FOR id_gr;
37
38 ALTER TABLE holding
39 ADD CONSTRAINT FK_holding_groups FOREIGN KEY (id_gr) REFERENCES groups (id_gr)
40 ON DELETE SET DEFAULT
41 ON UPDATE CASCADE;

```

Сообщения

Выполнение команд успешно завершено.

Но в базе данных таблицы заполнены информацией, поэтому необходимо обновить данные. В таблице «Группы» запись, имеющаяся в таблице с кодом 1, добавим снова, т.к. код группы формируется автоматически, это действие не вызовет ошибки, а затем обновим запись 1:

```

22 );
23
24 INSERT INTO groups VALUES ('ПО-31',24,2020,'09.02.03');
25
26 UPDATE dbo.groups
27 SET name_gr='удалены', struct=00, years='****', department='**.*.*'
28 WHERE id_gr=1
29
30 SELECT * FROM groups;

```

Сообщения

id_gr	name_gr	struct	years	department
1	удалены	0	****	**.*.*
2	ПО-32	22	2020	09.02.03
3	ПО-33к	16	2020	09.02.03
4	ПО-31	24	2020	09.02.03

Теперь добавим в таблицу групп запись:

```

42
43 INSERT INTO groups VALUES ('ПО-21',23,2020,'09.02.03');
44
45 INSERT INTO holding VALUES ('2020-12-08','2020-12-08',5,9,1,30,10,0);
46 DELETE FROM groups
47 WHERE name_gr='ПО-21'
48

```

Сообщения

id_gr	name_gr	struct	years	department
1	удалены	0	****	**.*.*
2	ПО-32	22	2020	09.02.03
3	ПО-33к	16	2020	09.02.03
4	ПО-31	24	2020	09.02.03
5	ПО-21	23	2020	09.02.03

Добавим в таблицу проведения тестирования запись, здесь код группы равен 9:

```

45 INSERT INTO holding VALUES ('2020-12-08','2020-12-08',5,9,1,30,10,0);
46 SELECT * FROM holding;
47 DELETE FROM groups
48 WHERE name_gr='ПО-21'
49

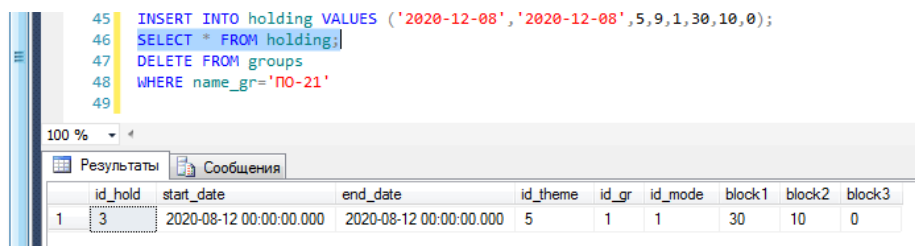
```

Сообщения

id_hold	start_date	end_date	id_theme	id_gr	id_mode	block1	block2	block3
1	2020-08-12 00:00:00.000	2020-08-12 00:00:00.000	5	9	1	30	10	0

И удалим запись из таблицы «Группы», в таблице проведения код группы изменился на значение по умолчанию, сама же запись осталась в таблице:

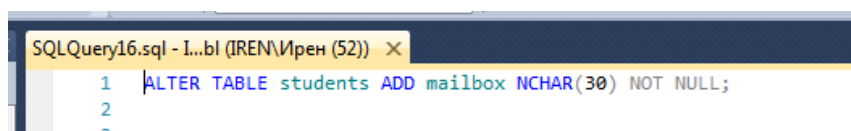
```
45 INSERT INTO holding VALUES ('2020-12-08', '2020-12-08', 5, 9, 1, 30, 10, 0);
46 SELECT * FROM holding;
47 DELETE FROM groups
48 WHERE name_gr='ГО-21'
49
```



	id_hold	start_date	end_date	id_theme	id_gr	id_mode	block1	block2	block3
1	3	2020-08-12 00:00:00.000	2020-08-12 00:00:00.000	5	1	1	30	10	0

### Добавление полей.

Чтобы добавить поле, используем команду ALTER TABLE имя таблицы ADD имя поля характеристики поля;



```
SQLQuery16.sql - I...bl (IREN\Ирен (52)) X
1 ALTER TABLE students ADD mailbox NCHAR(30) NOT NULL;
2
3
```

...

Таким образом, в SQL Server имеются инструменты для ... после создания базы данных.

### Содержание отчета

- 1 Название работы
- 2 Цель работы
- 3 Перечень технических средств обучения
- 4 Порядок выполнения работы
- 5 Вывод

### Варианты заданий

Варианты заданий представлены в практической работе № 13.

### Используемая литература

- Г.Н.Федорова Основы проектирования баз данных. М.: Академия, 2020
- Г.Н.Федорова Разработка, администрирование и защита баз данных. М.: Академия, 2018

– <https://webonto.ru/sql-alter-table-komanda-sql-na-modifikatsiyu-tablitsyi-bazyi-dannyih/>