

Типы данных. Ограничения целостности.

Лекция 4.

Типы данных:

1. числовые,
2. символьные,
3. для хранения даты и времени,
4. денежные,
5. двоичные,
6. специальные.

```
create table AUDITORIUM -- аудитории вуза
(
    AUDITORIUM          char(20)    -- идентификатор аудитории (304-1, 401-4,...)
    constraint AUDITORIUM_PK primary key,
    AUDITORIUM_TYPE     char(10)    -- тип аудитории (ЛК, ЛБ-К, ...)
    constraint AUDITORIUM_AUDITORIUM_TYPE_FK foreign key
    references AUDITORIUM_TYPE (AUDITORIUM_TYPE),
    AUDITORIUM_CAPACITY int         -- вместимость (макс. кол. слушателей)
    constraint AUDITORIUM_CAPACITY_CHECK default 1
    check (AUDITORIUM_CAPACITY between 1 and 300),
    AUDITORIUM_NAME     varchar(50) -- текстовое наименование аудитории (комментарий)
) on FG1; -- в файловой группе FG1
```

Точные числовые типы

Тип данных	Диапазон значений	Количество байт
tinyint	0–255	1
smallint	–32 768–32 768	2
int	$-2^{31}-(2^{31}-1)$	4
bigint	$-2^{63}-(2^{63}-1)$	8
bit	0 или 1	1
decimal(p,s) numeric(p,s) $1 \leq p \leq 38,$ $0 \leq s < p$	$(-10^{38} + 1) - (10^{38} + 1)$	5–17

```
use TEMPDB
go
create table EXACT_NUMBER
```

```
(
  N_BIT          bit,
  N_TINYINT      tinyint,
  N_SMALLINT     smallint,
  N_INT          int,
  N_BIGINT       bigint,
  N_DECIMAL      decimal(10,3)
);
```

```
go
insert into EXACT_NUMBER values (1, 123, 12345, 1234567890, 123456789012345678, 1234567.890);
insert into EXACT_NUMBER values (-1, 123, -12345, -1234567890, -123456789012345678, -1234567.123);
insert into EXACT_NUMBER values (0, 123, 12345, 1234567890, 123456789012345678, 1234567.1234);
go
select * from EXACT_NUMBER;
go
```

N_BIT	N_TINYINT	N_SMALLINT	N_INT	N_BIGINT	N_DECIMAL
1	123	12345	1234567890	123456789012345678	1234567.890
1	123	-12345	-1234567890	-123456789012345678	-1234567.123
0	123	12345	1234567890	123456789012345678	1234567.123

Приближенные числовые типы

Тип данных	Диапазон значений	Количество байт
float(p) $1 \leq p \leq 53$	От $-1,79 \cdot 10^{38}$ до $-2,23 \cdot 10^{-38}$; 0; от $2,23 \cdot 10^{-38}$ до $1,79 \cdot 10^{38}$	4 или 8
real float(24)	От $-3,4 \cdot 10^{38}$ до $-1,18 \cdot 10^{-38}$; 0; от $1,18 \cdot 10^{-38}$ до $3,4 \cdot 10^{38}$	4

```
use TEMPDB
go
create table APPROXIMATE_NUMBER
(
    N_FLOAT      float(53),
    N_REAL       real
);
go
```

N_FLOAT	N_REAL
1,1	2,2
11234567890,12	2,123457E+12
1123456789012	2,123457E+12
1,123E-08	2,12345E-10
-1,123E-08	-2,12345E-10
1,23E-05	0,12345

```
insert into APPROXIMATE_NUMBER values (1.1, 2.2);
insert into APPROXIMATE_NUMBER values (1.123456789012E10, 2.123456789012E12);
insert into APPROXIMATE_NUMBER values (112.3456789012*power(10.0, 10), 2.123456789012*power(10., 12));
insert into APPROXIMATE_NUMBER values (1.123E-8, 2.12345E-10);
insert into APPROXIMATE_NUMBER values (-1.123E-8, -2.12345E-10);
insert into APPROXIMATE_NUMBER values (0.0000123, .12345);
select * from APPROXIMATE_NUMBER;
```

Символьные типы:

- обычные (CHAR, VARCHAR),
- в формате Юникода (NCHAR, NVARCHAR).

Тип данных	Размер в символах	Количество байт
char(n)	1–8000	n
varchar(n)	1–4000	Количество символов + 2
varchar(max)	1–($2^{31} - 1$)	Количество символов + 2
nchar(n)	1–8000	2n
nvarchar(n)	1–4000	2·количество символов + 2
nvarchar(max)	1–($2^{30} - 1$)	2·количество символов + 2

```
use TEMPDB
go
```

```
create table CHARACTER_DATA
(
    C_CHAR      char(100),
    C_VARCHAR   varchar(200),
    C_VARCHARM  varchar(max),
    C_NCHAR     nchar(100),
    C_NVARCHAR  nvarchar(200),
    C_NVARCHARM nvarchar(max)
);
go
```

```
insert into CHARACTER_DATA values ('12345', '12345', '12345', N'12345', N'12345', N'12345')
```

```
select * from CHARACTER_DATA
```

```
select datalength(C_CHAR)      [C_CHAR (байты)], datalength(C_VARCHAR) [C_VARCHAR (байты)],
       datalength(C_VARCHARM) [C_VARCHARM(байты)],
       datalength(C_NCHAR)     [C_NCHAR (байты)], datalength(C_NVARCHAR) [C_NVARCHAR (байты)],
       datalength(C_NVARCHARM) [C_NVARCHARM(байты)] from CHARACTER_DATA;
```

```
select len(C_CHAR)      [C_CHAR (симв)], len(C_VARCHAR) [C_VARCHAR (симв)],
       len(C_VARCHARM) [C_VARCHARM(симв)],
       len(C_NCHAR)     [C_NCHAR (симв)], len(C_NVARCHAR) [C_NVARCHAR (симв)],
       len(C_NVARCHARM) [C_NVARCHARM(симв)] from CHARACTER_DATA;
```

C_CHAR	C_NCHAR	C_VARCHAR	C_VARCHARM	C_NVARCHAR	C_NVARCHARM
12345	12345	12345	12345	12345	12345
C_CHAR (байты)	C_VARCHAR (байты)	C_VARCHARM(байты)	C_NCHAR (байты)	C_NVARCHAR (байты)	C_NVARCHARM(байты)
100	5	5	200	10	10
C_CHAR (симв)	C_VARCHAR (симв)	C_VARCHARM(симв)	C_NCHAR (симв)	C_NVARCHAR (симв)	C_NVARCHARM(симв)
5	5	5	5	5	5

Типы для даты и времени

Тип данных	Диапазон, точность, формат	Количество байт
date	01.01.1753–31.12.9999; 1 день; YYYYMMDD	3
time(p) $0 \leq p \leq 7$	00:00:00.0000000–23:59:59.9999999; 100 нс; hh:mm:ss.nnnnnnnn	3–5
smalldatetime	01.01.1900 00:00–06.06.2079 23:59, 1 мин; YYYYMMDD hh:mm	4
datetime	01.01.1753 00:00:00.000–31.12.9999 23:59:59.999; 0.003 с; YYYYMMDD hh:mm:ss.nnn	8
datetime2(p) $0 \leq p \leq 7$	01.01.0001 00:00:00.00000000– 31.12.9999 23:59:59.99999999; 100 нс; YYYYMMDD hh:mm:ss.nnnnnnnn	6–8
datetimeoffset(p) $0 \leq p \leq 7$	01.01.0001.00:00:00.00000000+00:00–31.12.9999. 23:59:59.99999999+23:59; 100 нс; YYYYMMDDhh:mm:ss.nnnnnnnn ± hh:mm	8–10

```

create table DATE_TIME_DATA
(
    D_DATE          date,
    D_TIME          time(7),
    D_SMALLDATETIME smalldatetime,
    D_DATETIME      datetime,
    D_DATETIME2     datetime2(7),
    D_DATETIMEOFFSET datetimeoffset(7)
);
go
insert into DATE_TIME_DATA
    values (
        '20140107',
        '23:59:55.1234567',
        '20140107 23:59',
        '20140107 23:59:59.123',
        '20140107 23:59:59.1234567',
        '20140107 23:59:59.1234567+02:30'
    );
select * from DATE_TIME_DATA

```

D_DATE	D_TIME	D_SMALLDATETIME	D_DATETIME	D_DATETIME2	D_DATETIMEOFFSET
2014-01-07	23:59:55.1234567	2014-01-07 23:59:00	2014-01-07 23:59:59.123	2014-01-07 23:59:59.1234567	2014-01-07 23:59:59.1234567 +02:30

Типы для хранения двоичных данных

Тип данных	Размер в байтах	Количество байт
binary(n)	1–8000	n
varbinary(n)	1–8000	Количество символов + 2
varbinary(max)	1–($2^{31} - 1$)	Количество символов + 2

```

use TEMPDB
go
create table BINARY_DATA
(
    B_BINARY          binary(400),
    B_VARBINARY       varbinary(400),
    B_VARBINARYM      varbinary(max)
);
go
insert into BINARY_DATA
    values(
        0x123456789012345,
        0x123456789012345,
        (select * from openrowset(bulk N'c:\SQLLoad\picture.jpg', SINGLE_BLOB) pic)
    );
select datalength(B_BINARY), datalength(B_VARBINARY), datalength(B_VARBINARYM) from BINARY_DATA

```

binary(байты)	varbinary(байты)	varbinary(max)(байты)
400	8	8549

Типы данных TIMESTAMP и ROWVERSION

Используются для определения порядка ввода/изменения записей в таблице. Занимают 8 байт.

```
use TEMPDB
go
create table TIMESTAMP_DATA1
(
    ID          int,
               timestamp      -- имя по умолчанию TIMESTAMP
);
go
create table TIMESTAMP_DATA2
(
    ID          int,
    T_ROWVERSION rowversion    -- имя обязательно
);
go
insert into TIMESTAMP_DATA1 (ID) values (1);
insert into TIMESTAMP_DATA2 (ID) values (1);
select * from TIMESTAMP_DATA1
select * from TIMESTAMP_DATA2
```

ID	timestamp
1	0x0000000000000007EF

ID	T_ROWVERSION
1	0x0000000000000007F0

Тип данных UNIQUEIDENTIFIER

Используется для хранения идентификаторов стандарта **UUID** (Universally Unique Identifier). Занимают 128 бит и могут быть получены с помощью встроенной функции **NEWID**.

```
use TEMPDB
go
create table ID_DATA
(
    ID        uniqueidentifier,
    V         varchar(10)
);
go

insert into ID_DATA values(newid(), 'ABCD');

select *, datalength(ID) [datalength(ID)] from ID_DATA
```

ID	V	datalength(ID)
0E947E7A-CEDC-4253-8732-4A52A9024CF7	ABCD	16

Редко используемые типы данных

Тип данных	Назначение
smallmoney, money	Денежный тип, разновидность числового типа с фиксированной точкой, предназначен для хранения значения денежных величин
xml	XML-тип предназначен для хранения данных, представленных в формате XML
hierarhied	Иерархический тип данных, используется для отражения иерархических (древовидных) связей между строками таблицы
geography, geometry	Пространственные данные, предназначенные для хранения географических координат, геометрических координат и геометрических (2D) объектов
sqlvariant	Универсальный тип, способный хранить числовые, символьные данные, а также данные для хранения даты и времени. Непосредственно перед использованием, данные могут быть преобразованы к необходимому типу
text, ntext, image	Устаревшие типы данных, поддерживаются для совместимости с предыдущими версиями сервера, заменены на varchar(max) и varbinary(max)

Преобразование типов (функции CAST и CONVERT)

```
select cast('-123.123' as numeric(7,3)), -- СИМВОЛЫ --> ЧИСЛО
        convert(numeric(7,3), '-123.123')
select cast(-123.123 as varchar(12)), -- ЧИСЛО --> СИМВОЛЫ
        convert(varchar(12), -123.123)
select cast(getdate() as char(30)), -- дата --> СИМВОЛЫ
        convert(char(30), getdate()),
        convert(char(30), getdate(), 112) -- 112 - стиль
select cast('20140107 22:00:01.123' as datetime), -- СИМВОЛЫ --> дата
        convert(datetime, '20140107 22:00:01.123')
```

(No column name)	(No column name)
-123.123	-123.123

(No column name)	(No column name)
-123.123	-123.123

(No column name)	(No column name)	(No column name)
Jan 2 2014 10:55PM	Jan 2 2014 10:55PM	20140102

(No column name)	(No column name)
2014-01-07 22:00:01.123	2014-01-07 22:00:01.123

Ограничения целостности

Условное обозначение ограничения целостности	Действие ограничения целостности
data type тип данных	Предотвращает появление в столбце значений, не соответствующих типу данных
not null запрет значений null	Предотвращает появление в столбце значений null
default значение по умолчанию	Устанавливает значение в столбце по умолчанию при выполнении операции INSERT
primary key первичный ключ	Предотвращает появление в столбце (группе столбцов) повторяющихся значений (комбинации значений) и пустого значения (комбинации пустых значений)
foreign key внешний ключ	Устанавливает связь между таблицей со столбцом, имеющим свойство foreign key (FK) и таблицей, имеющей столбец со свойством primary key (PK); предотвращает не согласованные операции между PK и FK
unique уникальное значение	Аналогично primary key, но допускает пустые значения и не может быть использован для связи с foreign key
check проверка значений	Предотвращается появление в столбце значения, не удовлетворяющего логическому условию

data type

```
create table S_AUDITORIUM  -- аудитории вуза
(
    AUDITORIUM             char(20),
    AUDITORIUM_TYPE        char(10),
    AUDITORIUM_CAPACITY    int,
    AUDITORIUM_NAME        varchar(50)
)
```

```
insert into S_AUDITORIUM(AUDITORIUM, AUDITORIUM_NAME, AUDITORIUM_TYPE, AUDITORIUM_CAPACITY)
values ('206-1', '206-1', 'ЛБ-К', 'XXX');
```

Msg 245, Level 16, State 1, Line 1

Conversion failed when converting the varchar value 'XXX' to data type int.

Первичный ключ

```
use BSTU
```

```
create table FACULTY -- факульт
```

```
(
    FACULTY      char(10) primary key,
    FACULTY_NAME varchar(50)
);
go
```

Msg 2627, Level 14, State 1, Line 4

Violation of PRIMARY KEY constraint 'PK_FACULTY_81E78829173876EA'. Cannot insert duplicate key in object 'dbo.FACULTY'.

The statement has been terminated.

```
insert into FACULTY (FACULTY, FACULTY_NAME )
    values ('ХТнТ', 'Химическая технология и техника');
insert into FACULTY (FACULTY, FACULTY_NAME )
    values ('ХТнТ', 'Химическая технология и техника');
select * from FACULTY;
```

FACULTY	FACULTY_NAME
ХТнТ	Химическая технология и техника

```
use BSTU
```

```
create table FACULTY -- факульт
```

```
(
    FACULTY      char(10) constraint PK_FACULTY_FACULTY primary key,
    FACULTY_NAME varchar(50)
);
go
```

Msg 2627, Level 14, State 1, Line 3

Violation of PRIMARY KEY constraint 'PK_FACULTY_FACULTY'. Cannot insert duplicate key in object 'dbo.FACULTY'.

```
insert into FACULTY (FACULTY, FACULTY_NAME )
    values ('ХТнТ', 'Химическая технология и техника');
insert into FACULTY (FACULTY, FACULTY_NAME )
    values ('ХТнТ', 'Химическая технология и техника');
select * from FACULTY;
```

NOT NULL

```
create table FACULTY -- факульт
(
    FACULTY      char(10),      -- идентификатор факультета
    FACULTY_NAME varchar(50),    -- полное наименование
    constraint PK_FACULTY_FACULTY primary key(FACULTY)
);

create table SHEDULE_TEACHER -- расписание преподавателя
(
    [DATETIME]    smalldatetime, -- дата и время занятий
    TEACHER        char(10),      -- преподаватель
    [SUBJECT]      char(10),      -- дисциплина
    AUDITORIUM     char(10),      -- аудитория
    constraint PK_SHEDULE_TEACHER primary key([DATETIME],TEACHER)
) on FG2
```

FACULTY	FACULTY_NAME
ИДиП	Издательское дело и полиграфия
ХТиТ	Химическая технология и техника

```
use BSTU
go

update FACULTY set FACULTY = NULL where FACULTY = 'ИДиП';
insert into FACULTY (FACULTY, FACULTY_NAME)
values (NULL, 'Лесохозяйственный факультет');
insert into FACULTY (FACULTY_NAME)
values ('Лесохозяйственный факультет');
go
```

```
Msg 515, Level 16, State 2, Line 1
Cannot insert the value NULL into column 'FACULTY', table 'BSTU.dbo.FACULTY'; column does not allow nulls. UPDATE fails.
The statement has been terminated.
Msg 515, Level 16, State 2, Line 2
Cannot insert the value NULL into column 'FACULTY', table 'BSTU.dbo.FACULTY'; column does not allow nulls. INSERT fails.
The statement has been terminated.
Msg 515, Level 16, State 2, Line 4
Cannot insert the value NULL into column 'FACULTY', table 'BSTU.dbo.FACULTY'; column does not allow nulls. INSERT fails.
The statement has been terminated.
```

Значение по умолчанию

```
create table FACULTY -- факультеты
(
    FACULTY      char(10),          -- идентификатор факультета
    FACULTY_NAME varchar(50) default '???' -- полное наименование
    constraint FACULTY_PK primary key(FACULTY)
);
go

insert into FACULTY (FACULTY, FACULTY_NAME)
    values ('ИДиП', 'Издательское дело и полиграфия');
insert into FACULTY (FACULTY)
    values ('ХТиТ'); -- неявный ввод значения default
insert into FACULTY (FACULTY, FACULTY_NAME)
    values ('ЛХФ', NULL); -- явный ввод значения NULL
insert into FACULTY (FACULTY, FACULTY_NAME)
    values ('ТТЛП', NULL); -- явный ввод значения NULL |
insert into FACULTY (FACULTY, FACULTY_NAME)
    values ('ИЭФ', default); -- явный ввод значения default

update FACULTY set FACULTY_NAME = NULL where FACULTY = 'ИДиП';
update FACULTY set FACULTY_NAME = default where FACULTY = 'ЛХФ';

select * from FACULTY;
```

FACULTY	FACULTY_NAME
ИДиП	NULL
ИЭФ	???
ЛХФ	???
ТТЛП	NULL
ХТиТ	???

Внешний ключ FOREIGN KEY

```
use BSTU
go

create table FACULTY -- факультеты
(
    FACULTY      char(10),          -- идентификатор факультета
    FACULTY_NAME varchar(50) not null default '???' , -- полное наименование
    constraint FACULTY_PK primary key(FACULTY)
);
go
```

FACULTY	FACULTY_NAME
ИДиП	Издательское дело и полиграфия
ИЗФ	Инженерно-экономический факультет
ЛХФ	Лесохозяйственный факультет
ТОВ	Технология органических веществ
ТТП	Технология и техника лесной промышленности
ХТиТ	Химическая технология и техника

```
create table PULPIT -- кафедра
(
    PULPIT      char(20)          -- идентификатор кафедры
    constraint PULPIT_PK primary key,
    PULPIT_NAME varchar(100)      -- полное наименование кафедры
    default '???' ,
    FACULTY      char(10)          -- идентификатор факультета
    constraint PULPIT_FACULTY_FK foreign key
    references FACULTY (FACULTY)
);
```

FK-таблица, оператор INSERT

```

insert into PULPIT (PULPIT, PULPIT_NAME, FACULTY) -- OK
    values ('ИСиТ', 'Информационных систем и технологий ', 'ИДиП' );
insert into PULPIT (PULPIT, PULPIT_NAME, FACULTY) -- OK
    values ('ПОиСОИ', 'Полиграфического оборудования и систем обработки информации ', NULL);
insert into PULPIT (PULPIT, PULPIT_NAME, FACULTY) -- OK
    values ('ЛУ', 'Лесоустройства', 'ЛХФ');
insert into PULPIT (PULPIT, PULPIT_NAME, FACULTY) -- error 547
    values ('ЛВ', 'Лесоводства', 'XXXX');
insert into PULPIT (PULPIT, PULPIT_NAME, FACULTY) -- OK
    values ('ЛВ', 'Лесоводства', 'ЛХФ');
insert into PULPIT (PULPIT, PULPIT_NAME, FACULTY) -- OK
    values ('ТЛ', 'Транспорта леса', NULL);
select * from PULPIT order by FACULTY

```

```

(1 row(s) affected)
(1 row(s) affected)
(1 row(s) affected)

```

Msg 547, Level 16, State 0, Line 7

The INSERT statement conflicted with the FOREIGN KEY constraint "PULPIT_FACULTY_FK".
The conflict occurred in database "BSTU", table "dbo.FACULTY", column 'FACULTY'.

The statement has been terminated.

```

(1 row(s) affected)
(1 row(s) affected)
(5 row(s) affected)

```

PULPIT	PULPIT_NAME	FACULTY
ПОиСОИ	Полиграфического оборудования и систем обработки...	NULL
ТЛ	Транспорта леса	NULL
ИСиТ	Информационных систем и технологий	ИДиП
ЛВ	Лесоводства	ЛХФ
ЛУ	Лесоустройства	ЛХФ

1. Оператор INSERT выполнится успешно, если для значения FK-ключа добавляемой строки в РК-таблице есть строка со значением РК-ключа, равным значению FK-ключа.
2. Оператор INSERT выполнится успешно, если значение FK-ключа будет NULL.

FK-таблица, оператор UPDATE

```
update PULPIT set FACULTY = 'ИДиП' where PULPIT = 'ПОиСОИ'; -- OK
update PULPIT set FACULTY = 'ЛХФ' where FACULTY is NULL; -- OK
update PULPIT set FACULTY = NULL where PULPIT = 'ИСиТ'; -- OK
update PULPIT set FACULTY = 'XXX' where FACULTY = 'ЛХФ'; -- error 547
```

```
select * from PULPIT order by FACULTY
```

```
(1 row(s) affected)
(1 row(s) affected)
(1 row(s) affected)
```

Msg 547, Level 16, State 0, Line 4

The UPDATE statement conflicted with the FOREIGN KEY constraint "PULPIT_FACULTY_FK".
The conflict occurred in database "BSTU", table "dbo.FACULTY", column 'FACULTY'.
The statement has been terminated.

PULPIT	PULPIT_NAME	FACULTY
ИСиТ	Информационных систем и технологий	NULL
ПОиСОИ	Полиграфического оборудования и систем обработки...	ИДиП
ТЛ	Транспорта леса	ЛХФ
ЛВ	Лесоводства	ЛХФ
ЛУ	Лесоустройства	ЛХФ

1. Оператор UPDATE выполнится успешно, если новое значение FK-ключа изменяемой строки таково, что в РК-таблице есть строка со значением РК-ключа, равным значению нового FK-ключа.
2. Оператор UPDATE выполнится успешно, если новое значение FK-ключа будет NULL.

FK-таблица, оператор DELETE

```
select count(*) 'количество строк до DELETE' from PULPIT;  
delete PULPIT; -- удаление всех строк PULPIT  
select count(*) 'количество строк после DELETE' from PULPIT;
```

количество строк до DELETE

5

количество строк после DELETE

0

Оператор DELETE для FK-таблицы выполняется успешно всегда.

PK-таблица, оператор INSERT

```
insert into FACULTY (FACULTY, FACULTY_NAME) -- OK
      values ('ИСП', 'Информационных систем и программирования');
insert into FACULTY (FACULTY, FACULTY_NAME) -- error 2627
      values ('ИДиП', 'Изд. дело, полиграфии и сист. обр. инф.');
```

select * from FACULTY order by FACULTY

(1 row(s) affected)
Msg 2627, Level 14, State 1, Line 3
Violation of PRIMARY KEY constraint 'FACULTY_PK'.
Cannot insert duplicate key in object 'dbo.FACULTY'.
The statement has been terminated.

FACULTY	FACULTY_NAME
ИДиП	Издательское дело и полиграфия
ИСП	Информационных систем и программирования
ИЭФ	Инженерно-экономический факультет
ЛХФ	Лесохозяйственный факультет
ТОВ	Технология органических веществ
ТТПП	Технология и техника лесной промышленности
ХТиТ	Химическая технология и техника

Оператор INSERT для PK-таблицы выполняется успешно всегда, если не нарушает ограничение PRIMARY KEY.

РК-таблица, оператор UPDATE

```
update FACULTY set FACULTY = 'ИЭ' where FACULTY = 'ИЭФ' -- OK
update FACULTY set FACULTY = 'ЛХ' where FACULTY = 'ЛХФ' -- error 547
update FACULTY set FACULTY = 'ИДиП' where FACULTY = 'ИСП' -- error 2627
```

```
select * from FACULTY order by FACULTY
```

(1 row(s) affected)

Msg 547, Level 16, State 0, Line 2

The UPDATE statement conflicted with the REFERENCE constraint "PULPIT_FACULTY_FK".

The conflict occurred in database "BSTU", table "dbo.PULPIT", column 'FACULTY'.

The statement has been terminated.

Msg 2627, Level 14, State 1, Line 3

Violation of PRIMARY KEY constraint 'FACULTY_PK'.

Cannot insert duplicate key in object 'dbo.FACULTY'.

The statement has been terminated.

FACULTY	FACULTY_NAME
ИДиП	Издательское дело и полиграфия
ИСП	Информационных систем и программирования
ИЭ	Инженерно-экономический факультет
ЛХФ	Лесохозяйственный факультет
ТОВ	Технология органических веществ
ТТПП	Технология и техника лесной промышленности
ХТиТ	Химическая технология и техника

1. Оператор UPDATE выполняется успешно, если применяется к строкам, на которые не ссылаются строки из FK-таблицы, и не нарушается ограничение PRIMARY KEY;
2. Строки FK-таблицы, не связанные со строками РК-таблицы (имеют значение NULL FK-ключа), никак не влияют на выполнение оператора UPDATE.

РК-таблица, оператор DELETE

```
select count(*) 'количество строк до DELETE' from FACULTY;  
delete FACULTY where FACULTY = 'ИЭ' -- OK  
delete FACULTY where FACULTY = 'ЛХФ' -- error 547  
select count(*) 'количество строк после DELETE' from FACULTY;
```

(1 row(s) affected)

Msg 547, Level 16, State 0, Line 3

The DELETE statement conflicted with the REFERENCE constraint "PULPIT_FACULTY_FK".

The conflict occurred in database "BSTU", table "dbo.PULPIT", column 'FACULTY'.

The statement has been terminated.

количество строк до DELETE

7

количество строк после DELETE

6

1. Оператор DELETE выполняется успешно, если применяется к строкам, на которые не ссылаются строки из FK-таблицы;
2. Строки FK-таблицы, не связанные со строками РК-таблицы (имеют значение NULL FK-ключа), никак не влияют на выполнение оператора DELETE.

Проверка значений CHECK

```
use BSTU
go
create table TEACHER
```

```
(
  TEACHER          char(10)  constraint TEACHER_PK  primary key,
  TEACHER_NAME     varchar(100),
  GENDER           char(1)   constraint TEACHER_GENDER_CH check (GENDER in ('м', 'ж')),
  PULPIT           char(20)  constraint TEACHER_PULPIT_FK foreign key
                        references PULPIT(PULPIT)
);
```

```
go
insert into TEACHER (TEACHER, TEACHER_NAME, PULPIT) -- OK
values ('СМЛВ', 'Смелов Владимир Владиславович', 'ИСИТ');
insert into TEACHER (TEACHER, TEACHER_NAME, PULPIT, GENDER) -- OK
values ('АКНВЧ', 'Акунович Станислав Иванович', 'ИСИТ', 'м');
insert into TEACHER (TEACHER, TEACHER_NAME, PULPIT, GENDER) -- OK
values ('БРС', 'Брусенцова Татьяна Палладьевна', 'ИСИТ', 'м');
insert into TEACHER (TEACHER, TEACHER_NAME, PULPIT, GENDER) -- OK
values ('МРЗ', 'Мороз Елена Станиславовна', 'ИСИТ', 'ж');
insert into TEACHER (TEACHER, TEACHER_NAME, PULPIT, GENDER) -- error 547
values ('РМНК', 'Романенко Дмитрий Михайлович', 'ИСИТ', 'х');
```

```
update TEACHER set GENDER = 'ж' where TEACHER = 'БРС';
update TEACHER set GENDER = NULL where TEACHER = 'МРЗ';
update TEACHER set GENDER = 'у' where TEACHER = 'СМЛВ';
```

```
select * from TEACHER
```

TEACHER	TEACHER_NAME	GENDER	PULPIT
АКНВЧ	Акунович Станислав Иванович	м	ИСИТ
БРС	Брусенцова Татьяна Палладьевна	ж	ИСИТ
МРЗ	Мороз Елена Станиславовна	NULL	ИСИТ
СМЛВ	Смелов Владимир Владиславович	NULL	ИСИТ

```
(1 row(s) affected)
```

```
(1 row(s) affected)
```

```
(1 row(s) affected)
```

```
(1 row(s) affected)
```

```
Msg 547, Level 16, State 0, Line 9
```

The INSERT statement conflicted with the CHECK constraint "TEACHER_GENDER_CH".

The conflict occurred in database "BSTU", table "dbo.TEACHER", column 'GENDER'.

The statement has been terminated.

```
(1 row(s) affected)
```

```
(1 row(s) affected)
```

```
Msg 547, Level 16, State 0, Line 14
```

The UPDATE statement conflicted with the CHECK constraint "TEACHER_GENDER_CH".

The conflict occurred in database "BSTU", table "dbo.TEACHER", column 'GENDER'.

The statement has been terminated.

Совместное использование ограничений CHECK, NOT NULL и DEFAULT

```
use BSTU
go
create table TEACHER
(
  TEACHER      char(10)  constraint TEACHER_PK  primary key,
  TEACHER_NAME varchar(100),
  GENDER       char(1)
               not null
               default 'М'
               constraint TEACHER_GENDER_CH check (GENDER in ('М', 'Ж')),
  PULPIT       char(20)  constraint TEACHER_PULPIT_FK foreign key
                   references PULPIT(PULPIT)
);
```

Уникальные значения UNIQUE

```

use BSTU
go
create table PROFESSION -- специальность
(
    PROFESSION      char(20)      -- специальность
                        constraint PROFESSION_PK primary key,
    FACULTY         char(10)      -- факультет
                        constraint PROFESSION_FACULTY_FK
                        foreign key references FACULTY(FACULTY),
    PROFESSION_NAME  varchar(100) -- наименование специальности
                        constraint PROFESSION_NAME_UQ unique,
    QUALIFICATION    varchar(50)  -- квалификация
);
go
insert into PROFESSION(FACULTY, PROFESSION,PROFESSION_NAME, QUALIFICATION) --OK
values ('ИДиП','1-40 01 02', 'Информационные системы и технологии',
        'инженер-программист-системотехник' );
insert into PROFESSION(FACULTY, PROFESSION,PROFESSION_NAME, QUALIFICATION) --OK
values ('ИДиП','1-47 01 01','Издательское дело', 'редактор-технолог' );
insert into PROFESSION(FACULTY, PROFESSION,PROFESSION_NAME, QUALIFICATION) -- error 2627
values ('ИДиП', '1-36 06 01','Издательское дело', 'инженер-электромеханик' );
insert into PROFESSION(FACULTY, PROFESSION,PROFESSION_NAME, QUALIFICATION) --OK
values ('ХТиТ', '1-36 01 08',NULL,'инженер-механик' );
insert into PROFESSION(FACULTY, PROFESSION,PROFESSION_NAME, QUALIFICATION) -- error 2627
values ('ХТиТ', '1-36 07 01',NULL, 'инженер-механик' );
update PROFESSION set PROFESSION_NAME = 'Информационные системы и технологии' -- error 2627
where PROFESSION = '1-36 01 08';
select * from PROFESSION;
    
```

(1 row(s) affected)

(1 row(s) affected)

Msg 2627, Level 14, State 1, Line 6

Violation of UNIQUE KEY constraint 'PROFESSION_NAME_UQ'.
Cannot insert duplicate key in object 'dbo.PROFESSION'.

The statement has been terminated.

(1 row(s) affected)

Msg 2627, Level 14, State 1, Line 10

Violation of UNIQUE KEY constraint 'PROFESSION_NAME_UQ'.
Cannot insert duplicate key in object 'dbo.PROFESSION'.

The statement has been terminated.

Msg 2627, Level 14, State 1, Line 12

Violation of UNIQUE KEY constraint 'PROFESSION_NAME_UQ'.
Cannot insert duplicate key in object 'dbo.PROFESSION'.

The statement has been terminated.

PROFESSION	FACULTY	PROFESSION_NAME	QUALIFICATION
1-36 01 08	ХТиТ	NULL	инженер-механик
1-40 01 02	ИДиП	Информационные системы и технологии	инженер-программист-системотехник
1-47 01 01	ИДиП	Издательское дело	редактор-технолог

Свойство IDENTITY

```

use BSTU
go
create table GROUPS --
(
  IDGROUP      int          -- идентификатор учебной группы
                identity(1,1)
                constraint GROUP_PK primary key,
  FACULTY      char(10)     -- факультет
                constraint GROUPS_FACULTY_FK foreign key
                references FACULTY(FACULTY),
  PROFESSION    char(20)    -- специальность
                constraint GROUPS_PROFESSION_FK foreign key
                references PROFESSION(PROFESSION),
  YEAR_FIRST   smallint     -- год поступления
                check (YEAR_FIRST<=YEAR(GETDATE())),
);
go
insert into GROUPS (FACULTY, PROFESSION, YEAR_FIRST) -- OK
values ('ИДиП', '1-40 01 02', 2013),
       ('ИДиП', '1-40 01 02', 2012),
       ('ИДиП', '1-40 01 02', 2011),
       ('ИДиП', '1-40 01 02', 2010),
       ('ИДиП', '1-47 01 01', 2013);

insert into GROUPS (IDGROUP, FACULTY, PROFESSION, YEAR_FIRST) -- error 544
values (99, 'ИДиП', '1-47 01 01', 2013);

select * from GROUPS;

```

(5 row(s) affected)

Msg 544, Level 16, State 1, Line 7

Cannot insert explicit value for identity column in table 'GROUPS' when IDENTITY_INSERT is set to OFF.

IDGROUP	FACULTY	PROFESSION	YEAR_FIRST
1	ИДиП	1-40 01 02	2013
2	ИДиП	1-40 01 02	2012
3	ИДиП	1-40 01 02	2011
4	ИДиП	1-40 01 02	2010
5	ИДиП	1-47 01 01	2013

```
update GROUPS set IDGROUP = 99 where IDGROUP = 1;
```

Msg 8102, Level 16, State 1, Line 1

Cannot update identity column 'IDGROUP'.

Вычисляемые столбцы

Столбец таблицы БД, значение которого может быть вычислено как выражение называется **вычисляемым**.

Свойство **PERSISTED** к выражению вычисляемого столбца предъявляет требование **детерминистичности** (значение всегда одинаково для каждой комбинации значений аргументов).

Если для вычисляемого столбца не указано свойство **PERSISTED**, то значения такого столбца физически не хранятся в таблице, а вычисляются каждый раз при обращении к нему. Такие столбцы называются **виртуальными**.

IDGROUP	FACULTY	PROFESSION	YEAR_FIRST	COURSE
1	ИДиП	1-40 01 02	2013	1
2	ИДиП	1-40 01 02	2012	2
3	ИДиП	1-40 01 02	2011	3
4	ИДиП	1-40 01 02	2010	4
5	ИДиП	1-47 01 01	2013	1
6	ИДиП	1-47 01 01	2012	2
7	ИДиП	1-47 01 01	2011	3

```
insert into GROUPS (FACULTY, PROFESSION, YEAR_FIRST, COURSE) -- error 271
values ('ТОБ', '1-54 01 03', 2012, 2);
```

```
update GROUPS set COURSE = 1 where PROFESSION = '1-40 01 02'; -- error 271
```

Msg 271, Level 16, State 1, Line 1

The column "COURSE" cannot be modified because it is either a computed column or is the result of a UNION operator.

```
use BSTU
```

```
go
```

```
create table GROUPS -- учебные группы
```

```
(
```

```
IDGROUP      int          -- идентификатор учебной группы
              identity(1,1)
```

```
              constraint GROUP_PK primary key,
```

```
FACULTY       char(10)     -- факультет
```

```
              constraint GROUPS_FACULTY_FK foreign key
              references FACULTY (FACULTY),
```

```
PROFESSION    char(20)     -- специальность
```

```
              constraint GROUPS_PROFESSION_FK foreign key
              references PROFESSION (PROFESSION),
```

```
YEAR_FIRST    smallint     -- год поступления
```

```
              check (YEAR_FIRST <= YEAR(GETDATE())),
```

```
COURSE        as case      -- курс
```

```
              when month(getdate()) between 8 and 12
```

```
              then year(getdate()) - YEAR_FIRST + 1
```

```
              else year(getdate()) - YEAR_FIRST
```

```
              end
```

```
);
```

```
go
```

```
insert into GROUPS (FACULTY, PROFESSION, YEAR_FIRST )
```

```
values ('ИДиП', '1-40 01 02', 2013), --1
```

```
       ('ИДиП', '1-40 01 02', 2012),
```

```
       ('ИДиП', '1-40 01 02', 2011),
```

```
       ('ИДиП', '1-40 01 02', 2010),
```

```
       ('ИДиП', '1-47 01 01', 2013), ---5 гр
```

```
       ('ИДиП', '1-47 01 01', 2012),
```

```
       ('ИДиП', '1-47 01 01', 2011);
```

```
select * from GROUPS;
```

Временные таблицы

- создаются для временного хранения результатов SELECT-запросов.
- хранятся в системной БД **TEMPDB** (создается при каждом перезапуске сервера, поэтому сохранить или восстановить временную таблицу в случае сбоя невозможно);
- не могут иметь внешние ключи;

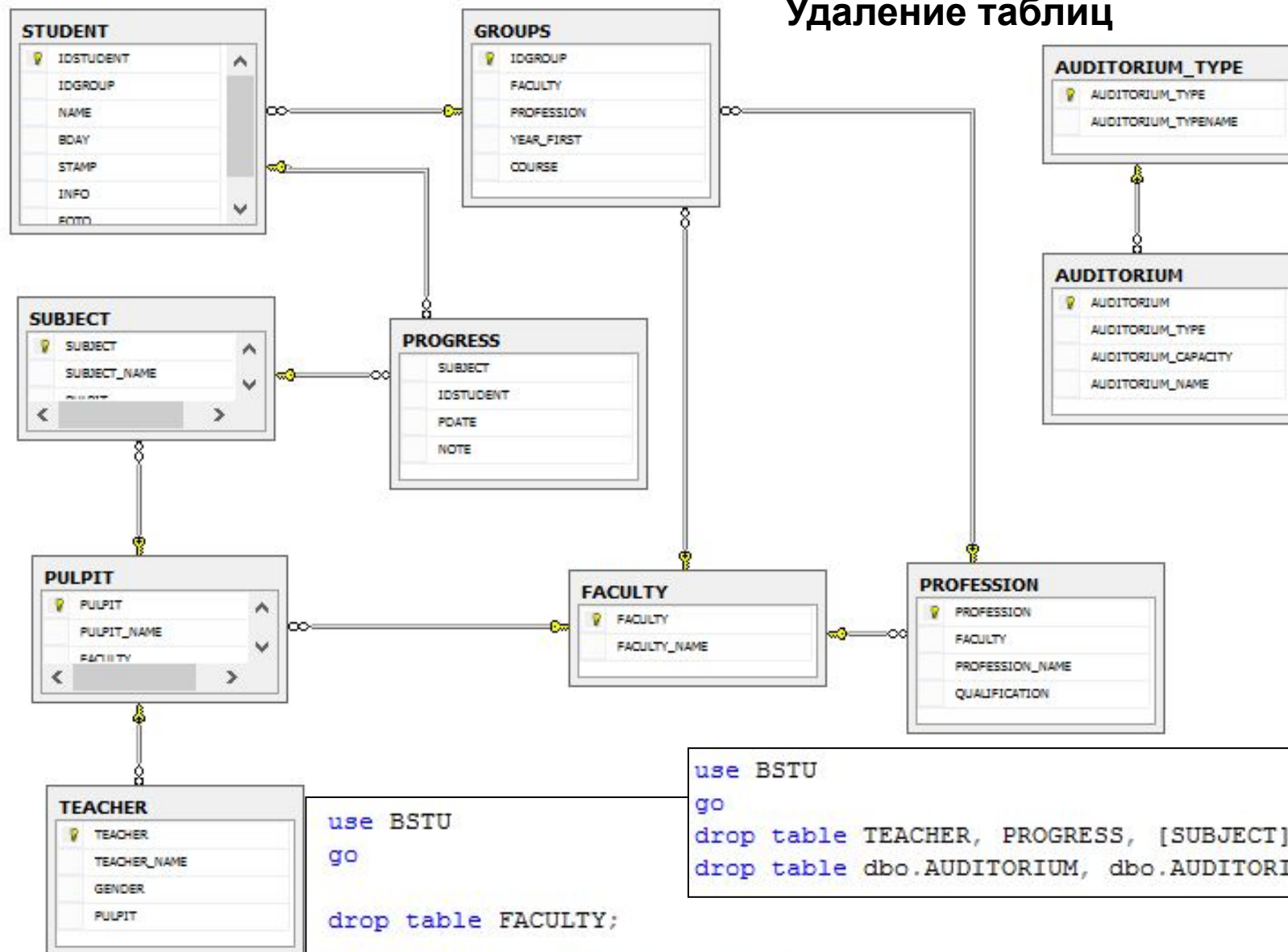
Два вида временных таблиц:

1. **Локальные** (имеют имена, начинающиеся с символа #, доступны только создавшему ее пользователю и могут быть удалены с помощью оператора DROP TABLE или автоматически при отключении пользователя);
2. **Глобальные** (имеют имена, начинающиеся с символа ##, доступны всем пользователям, подключенным к серверу, могут быть удалены с помощью оператора DROP TABLE автоматически при отключении всех пользователей).

TEACHER	TEACHER_NAME	GENDER	PULPIT
АКНВЧ	АКНВЧ	м	ИСиТ
ГРМН	Герман Олег Витольдович	м	ИСиТ
ЛЩНК	Лашенко Анатолий Павлович	м	ИСиТ
СМЛВ	Смелов Владимир Владиславович	м	ИСиТ

```
use BSTU
go
create table #TEACHER
(
    TEACHER          char(10)
                    constraint [T_TEACHER_PK] primary key,
    TEACHER_NAME     varchar(100) default '???' ,
    GENDER           char(1)  default 'м'
                    constraint [T_TEACHER_GENDER_CH] check (GENDER in ('м', 'ж')),
    PULPIT           char(20)
);
go
insert into #TEACHER (TEACHER, TEACHER_NAME, PULPIT )
values ('СМЛВ', 'Смелов Владимир Владиславович', 'ИСиТ'),
       ('АКНВЧ', default, 'ИСиТ'),
       ('КЛСНВ', null, 'ИСиТ'),
       ('ГРМН', 'Герман Олег Витольдович', 'ИСиТ'),
       ('ЛЩНК', 'Лашенко Анатолий Павлович', 'ИСиТ');
update #TEACHER set TEACHER_NAME = TEACHER where TEACHER_NAME = '???' ;
delete #TEACHER where TEACHER_NAME is null;
select * from #TEACHER;
drop table #TEACHER;
```

Удаление таблиц



Таблицу можно удалить с помощью оператора DROP TABLE. Возможные проблемы:

1. Пользователь не имеет достаточных прав на удаление таблицы.
2. Таблица заблокирована транзакцией другого сеанса.
3. На первичный ключ удаляемой таблицы ссылается внешний ключ другой таблицы.

```
use BSTU
```

```
go
```

```
drop table TEACHER, PROGRESS, [SUBJECT], PULPIT, STUDENT, GROUPS, PROFESSION, FACULTY;
```

```
drop table dbo.AUDITORIUM, dbo.AUDITORIUM_TYPE;
```

```
use BSTU
```

```
go
```

```
drop table FACULTY;
```

Msg 3726, Level 16, State 1, Line 2

Could not drop object 'FACULTY' because it is referenced by a FOREIGN KEY constraint.

Модификация таблиц

Для просмотра структуры таблицы применяют системную процедуру **SP_HELP**.

```
use BSTU
go
exec SP_HELP TEACHER;
```

Name	Owner	Type	Created_datetime
TEACHER	dbo	user table	2014-01-09 15:24:54.093

Column_name	Type	Computed	Length	Prec	Scale	Nullable	TrimTrailingBlanks	FixedLenNullInSource
TEACHER	char	no	10			no	no	no
TEACHER_NAME	varchar	no	100			yes	no	yes
GENDER	char	no	1			yes	no	yes
PULPIT	char	no	20			yes	no	yes

constraint_type	constraint_name	delete_action	update_action	status_enabled	status_for_replication	constraint_keys
CHECK on column GENDER	TEACHER_GENDER_CH	(n/a)	(n/a)	Enabled	Is_For_Replication	([GENDER]='ж' OR [GENDER]='м')
PRIMARY KEY (clustered)	TEACHER_PK	(n/a)	(n/a)	(n/a)	(n/a)	TEACHER
FOREIGN KEY	TEACHER_PULPIT_FK	No Action	No Action	Enabled	Is_For_Replication	PULPIT
						REFERENCES BSTU.dbo.PULPI...

```
use BSTU
go
select *, len(TEACHER_NAME) 'len(TEACHER_NAME)' from TEACHER order by TEACHER
```

TEACHER	TEACHER_NAME	GENDER	PULPIT	len(TEACHER_NAME)
АКНВЧ	Акунович Станислав Иванович	NULL	ИСиТ	27
АРС	Арсентьев Виталий Арсентьевич	NULL	ПОиСОИ	29
АТР	Атрощенко Олег Александрович	NULL	ЛУ	28
БЗБРДВ	Безбородов Владимир Степанович	NULL	ОХ	30
БЛВ	Беляев Валерий Павлович	NULL	ПОиСОИ	23
БРГ	Бурганская Татьяна Минаевна	NULL	ЛПиСПС	27
БРКВЧ	Бракович Андрей Игоревич	NULL	ИСиТ	24
БРНВСК	Барановский Станислав Иванович	NULL	ЭТиМ	30
БРС	Брусенцова Татьяна Палладьевна	NULL	ИСиТ	30
БРТШВЧ	Барташевич Святослав Александрович	NULL	ПОиСОИ	34
ГРМН	Герман Олег Витольдович	NULL	ИСиТ	23
ГРН	Гурин Николай Иванович	NULL	ИСиТ	22
ДЛН	Долинина Татьяна Николаевна	NULL	СБУАиА	27
ДМДК	Демидко Марина Николаевна	NULL	ЛПиСПС	25
ДТК	Дятко Александр Аркадьевич	NULL	ИСиТ	26
ЕЩНК	Ещенко Людмила Семеновна	NULL	ТНВиО...	24

```
use BSTU
go
alter table TEACHER alter column GENDER char(1) not null;
Msg 515, Level 16, State 2, Line 1
Cannot insert the value NULL into column 'GENDER', table 'BSTU.dbo.TEACHER';
column does not allow nulls. UPDATE fails.
The statement has been terminated.
```

```
use BSTU
go
update TEACHER set GENDER = 'м' where GENDER is null or GENDER not in ('м', 'ж');
alter table TEACHER alter column GENDER char(1) not null;
```

```
use BSTU
go
alter table TEACHER -- изменение столбца
    alter column TEACHER_NAME varchar(50) not null;
alter table TEACHER -- добавление нового столбца
    add BDATE date not null default '19600101'
```

```
use BSTU
go
select *, len(TEACHER_NAME) 'len(TEACHER_NAME)' from TEACHER order by TEACHER
```

TEACHER	TEACHER_NAME	GENDER	PULPIT	BDATE	len(TEACHER_NAME)
АКНВЧ	Акунович Станислав Иванович	NULL	ИСиТ	1960-01-01	27
АРС	Арсентьев Виталий Арсентьевич	NULL	ПОиСОИ	1960-01-01	29
АТР	Атрощенко Олег Александрович	NULL	ЛУ	1960-01-01	28
БЗБРДВ	Безбородов Владимир Степанович	NULL	ОХ	1960-01-01	30
БЛВ	Беляев Валерий Павлович	NULL	ПОиСОИ	1960-01-01	23
БРГ	Бурганская Татьяна Минаевна	NULL	ЛПиСПС	1960-01-01	27
БРКВЧ	Бракович Андрей Игоревич	NULL	ИСиТ	1960-01-01	24
БРНВСК	Барановский Станислав Иванович	NULL	ЭТиМ	1960-01-01	30
БРС	Брусенцова Татьяна Палладьевна	NULL	ИСиТ	1960-01-01	30
БРТШВЧ	Барташевич Святослав Александрович	NULL	ПОиСОИ	1960-01-01	34
ГРМН	Герман Олег Витольдович	NULL	ИСиТ	1960-01-01	23
ГРН	Гурин Николай Иванович	NULL	ИСиТ	1960-01-01	22
ДЛН	Долинина Татьяна Николаевна	NULL	СБУАиА	1960-01-01	27
ДМДК	Демидко Марина Николаевна	NULL	ЛПиСПС	1960-01-01	25
ДТК	Дятко Александр Аркадьевич	NULL	ИСиТ	1960-01-01	26
ЕЩНК	Ещенко Людмила Семеновна	NULL	ТНВиО...	1960-01-01	24