

Практический материал 4 к РГР2. Изменение таблицы.

Команда ALTER TABLE. Добавить/удалить столбцы, изменить тип столбцов. Добавить/удалить ограничения.. Добавление/удаление ограничений с именами. Добавление внешнего ключа. Добавление первичного ключа

Выполнить задания 1-8, отчет содержащий скрины прикрепить на уч. портал

<https://learn.microsoft.com/ru-ru/sql/t-sql/statements/alter-table-transact-sql?view=sql-server-ver15>

Инструкция ALTER TABLE используется для добавления, удаления или изменения столбцов в существующей таблице.

Инструкция ALTER TABLE также используется для добавления и удаления различных ограничений на существующую таблицу.

Синтаксис команды ALTER TABLE:

ALTER TABLE название_таблицы [WITH CHECK | WITH NOCHECK]
{ ADD название_столбца тип_данных_столбца [атрибуты_столбца] |
DROP COLUMN название_столбца |
ALTER COLUMN название_столбца тип_данных_столбца [NULL|NOT NULL] |
ADD [CONSTRAINT] определение_ограничения |
DROP [CONSTRAINT] имя_ограничения }

Для модифицирования схемы таблицы применяется инструкция ALTER TABLE.

Задание 1.

Создадим таблицу Klient_Alt

!!Примечание: работаете в своей БД у меня это *TestDatabas*, у вас это может быть другая

```
use TestDatabas;
If Object_ID('dbo.klient_alt', 'U') is not Null
Drop Table dbo.klient_alt;
Create table Klient_alt
(
  Id int Primary Key,
  Age int,
  NName NVARCHAR(20),
  Fio NVARCHAR(20),
  Email VARCHAR(30),
  Phone VARCHAR(20)
);
```

Оператор IF вызывает функцию OBJECT_ID, чтобы проверить, существует ли таблица Klient_Alt в текущей БД. В качестве аргументов функция OBJECT_ID принимает имя и тип объекта. Тип 'U' представляет пользовательскую таблицу. Если ответ положительный, функция возвращает идентификатор объекта; в противном случае — значение NULL. В нашей ситуации код удаляет существующую таблицу и создает вместо нее новую. Естественно, можно предусмотреть и другую реакцию — например, оставить имеющийся объект.

Здесь указывается полное имя таблицы 'dbo.klient_alt. Если опустить префикс, SQL Server будет считать, что пользователь запускает код в контексте схемы, применяемой по умолчанию в текущей БД.

Рекомендация: использовать запись NOT NULL и опускать ее только в случаях, когда возникает серьезная необходимость поддерживать значения NULL. Если не указать ограничение NOT NULL, в столбце появятся пустые значения.

Задание 2. Добавление нового столбца

```
ALTER TABLE [schema_name.]table_name
ADD column_name1 data_type constraint,
    column_name2 data_type constraint
...
    column_nameN data_type constraint;
```

Добавим в таблицу новый столбец Address:

```
use TestDatabas;
If Object_ID('dbo.klient_alt', 'U') is not Null
Drop Table dbo.klient_alt;
Create table Klient_alt
(
    Id int Primary Key,
    Age int Not Null,
    NName NVARCHAR(20) Not Null,
    Fio NVARCHAR(20) Not Null,
    Email VARCHAR(30) Not Null,
    Phone VARCHAR(20) Not Null
);
Alter Table Klient_alt
Add Address NVARCHAR(50) Null;
```

В данном случае столбец Address имеет тип NVARCHAR и для него определен атрибут NULL. Но что если нам надо добавить столбец, который не должен принимать значения NULL? Если в таблице есть данные, то следующая команда не будет выполнена:

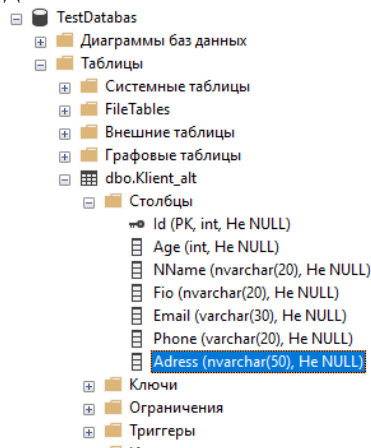
```
Alter Table Klient_alt
Add Address NVARCHAR(50) Not Null;
```

Поэтому в данном случае решение состоит в установке значения по умолчанию через атрибут DEFAULT:

Выполните:

```
use TestDatabas;
If Object_ID('dbo.klient_alt', 'U') is not Null
Drop Table dbo.klient_alt;
Create table Klient_alt
(
    Id int Primary Key,
    Age int Not Null,
    NName NVARCHAR(20) Not Null,
    Fio NVARCHAR(20) Not Null,
    Email VARCHAR(30) Not Null,
    Phone VARCHAR(20) Not Null
);
Alter Table Klient_alt
Add Address NVARCHAR(50) Not Null Default 'нет';
```

В этом случае, если в таблице уже есть данные, то для них для столбца Address будет добавлено значение "Неизвестно".



Задание 3. Удаление столбца

Синтаксис:

ALTER TABLE [schema_name.]table_name

DROP column column_name1, column_name2,... column_nameN;

Ниже удаляется столбец таблицы. AddressEmployee

ALTER TABLE dbo.Employee

DROP COLUMN Address;

В следующем примере удаляются несколько столбцов таблицы. Employee

ALTER TABLE dbo.Employee

DROP COLUMN Address, PhoneNo, Email;

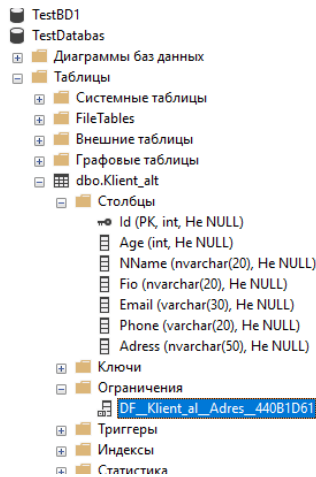
Удалим столбец Address с помощью команды

```
ALTER TABLE dbo.klient_alt
    DROP COLUMN adress;
```

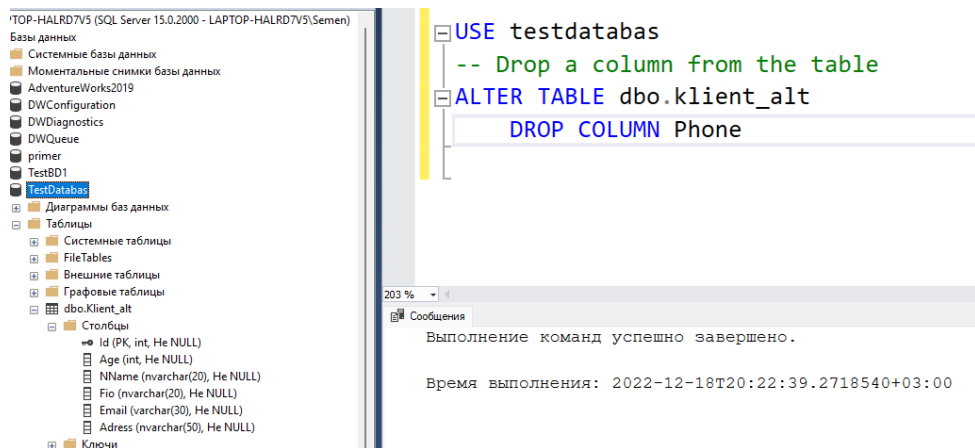
```
USE testdatabas;
-- Drop a column from the table
ALTER TABLE dbo.klient_alt
    DROP COLUMN adress;
```

Сообщение 5074, уровень 16, состояние 1, строка 4
The object 'DF_Klient_al_Adres_440B1D61' is dependent on column 'adress'.
Сообщение 4922, уровень 16, состояние 9, строка 4
ALTER TABLE DROP COLUMN adress failed because one or more objects access this column.

Видим, что ошибка. У нас на столбец Address было ограничение

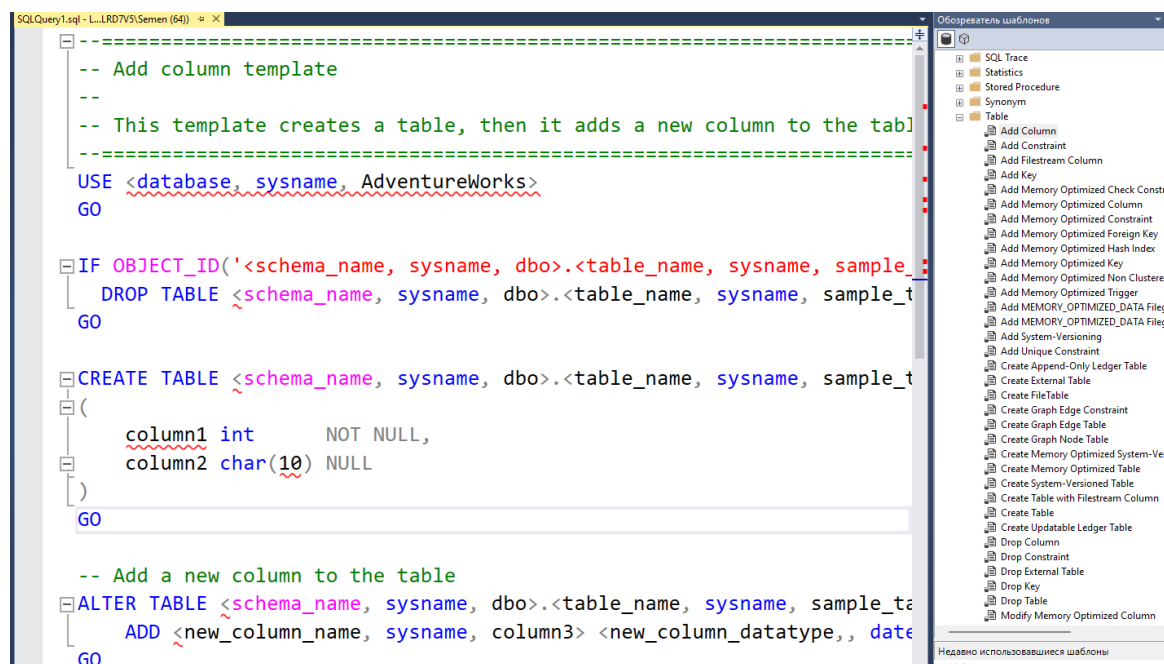


Задание 4.



Задание 5.

Вернем столбец Phone. Добавим столбец Phone. Воспользуемся обозревателем шаблонов в SSMS.

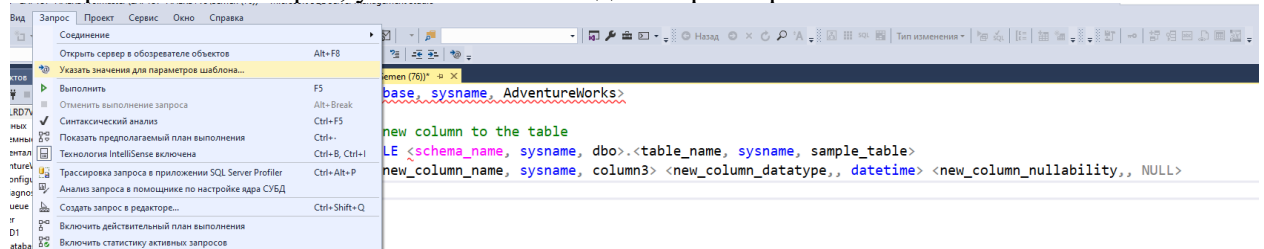


Оставим только следующее

```
USE <database, sysname, AdventureWorks>

-- Add a new column to the table
ALTER TABLE <schema_name, sysname, dbo>.<table_name, sysname, sample_table>
    ADD <new_column_name, sysname, column3> <new_column_datatype,, datetime> <new_column_nullability,, NULL>
```

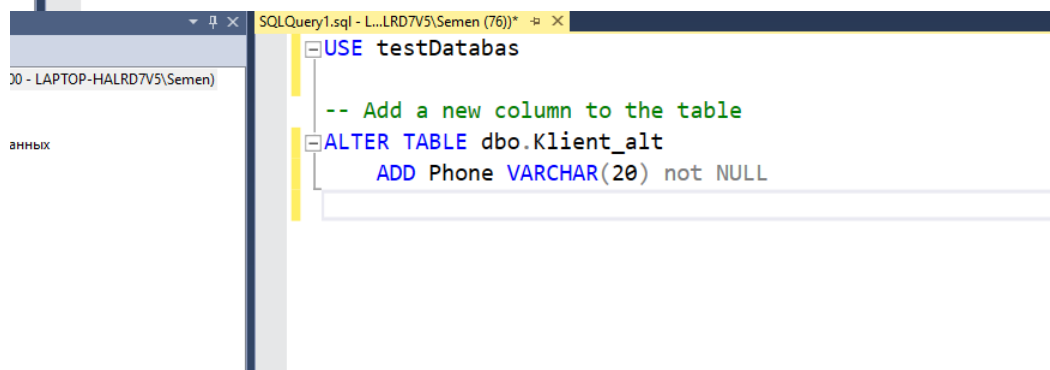
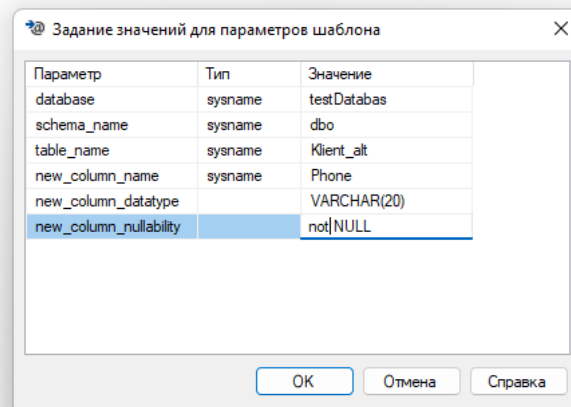
Выберем в меню запрос, указать значения для параметров шаблона

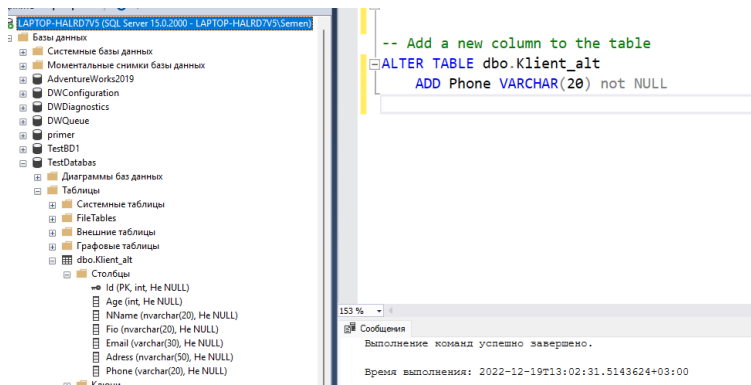


Укажем наши значения

```
USE <database, sysname, AdventureWorks>

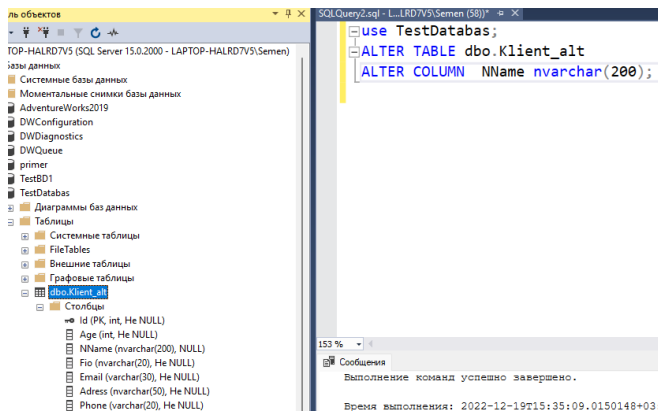
-- Add a new column to the table
ALTER TABLE <schema_name, sysname, dbo>.<table_name, sysname, sample_table>
    ADD <new_column_name, sysname, column3> <new_column_datatype,, datetime> <new_column_nullability,, NULL>
```





Задание 6. Изменение типа столбца

Изменим в таблице Klient_alt тип данных у столбца NName на **NVARCHAR(20)**



Задание 7. Ограничения

SQL ограничения используются для указания правил для данных в таблице.

Ограничения используются для ограничения типа данных, которые могут быть помещены в таблицу. Это обеспечивает точность и достоверность данных в таблице. Если существует какое-либо нарушение между ограничением и действием данных, действие прерывается.

Ограничения могут быть на уровне столбцов или таблицы. Ограничения уровня столбца применяются к столбцу, а ограничения уровня таблицы ко всей таблице.

В SQL обычно используются следующие ограничения:

- **NOT NULL** - Гарантирует, что столбец не может иметь нулевое значение
- **UNIQUE** - Гарантирует, что все значения в столбце будут разными
- **PRIMARY KEY** - Комбинация NOT NULL и UNIQUE. Уникально идентифицирует каждую строку в таблице
- **FOREIGN KEY** - Однозначно идентифицирует строку/запись в другой таблице
- **CHECK** - Гарантирует, что все значения в столбце удовлетворяют определённому условию
- **DEFAULT** - Задаёт значение по умолчанию для столбца, если значение не указано
- **INDEX** - Используется для быстрого создания и извлечения данных из базы данных

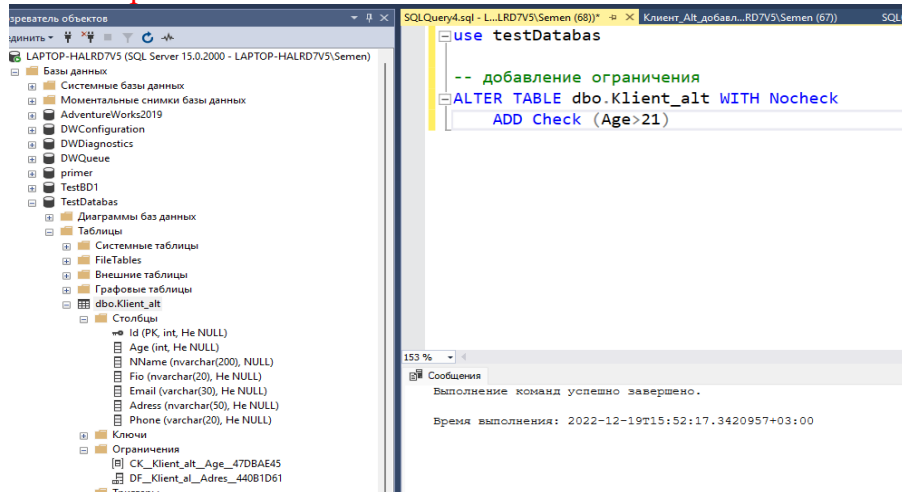
Задание 7.1 Добавление ограничения CHECK

При добавлении ограничений SQL Server автоматически проверяет имеющиеся данные на соответствие добавляемым ограничениям. Если данные не соответствуют ограничениям, то такие ограничения не будут добавлены.

Задание. Установим для столбца Age в таблице Klient_alt ограничение Age > 21.

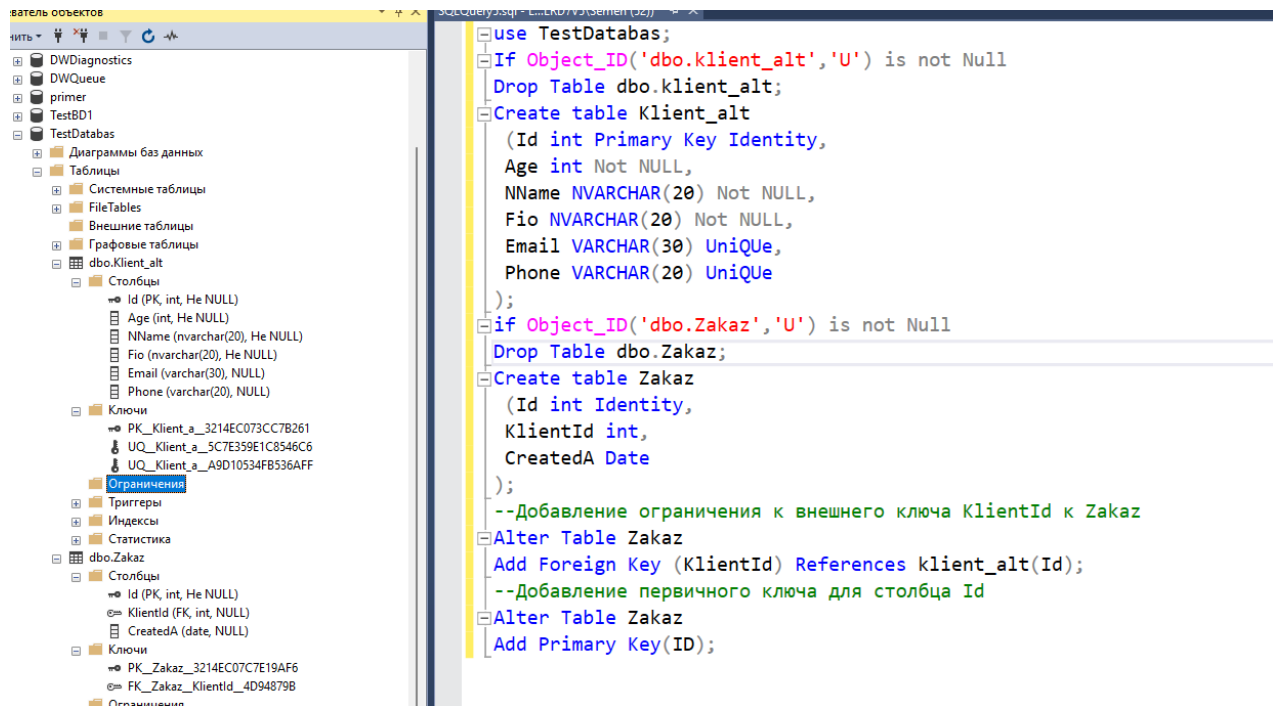
Если в таблице есть строки, в которых в столбце Age есть значения, несоответствующие этому ограничению, то sql-команда завершится с ошибкой. Чтобы избежать подобной проверки на соответствие и все таки добавить ограничение, несмотря на наличие несоответствующих ему данных, используется выражение **WITH NOCHECK**.

По умолчанию используется значение **WITH CHECK**, которое проверяет на соответствие ограничениям.



Задание 7.2 Добавление внешнего ключа. Добавление первичного ключа

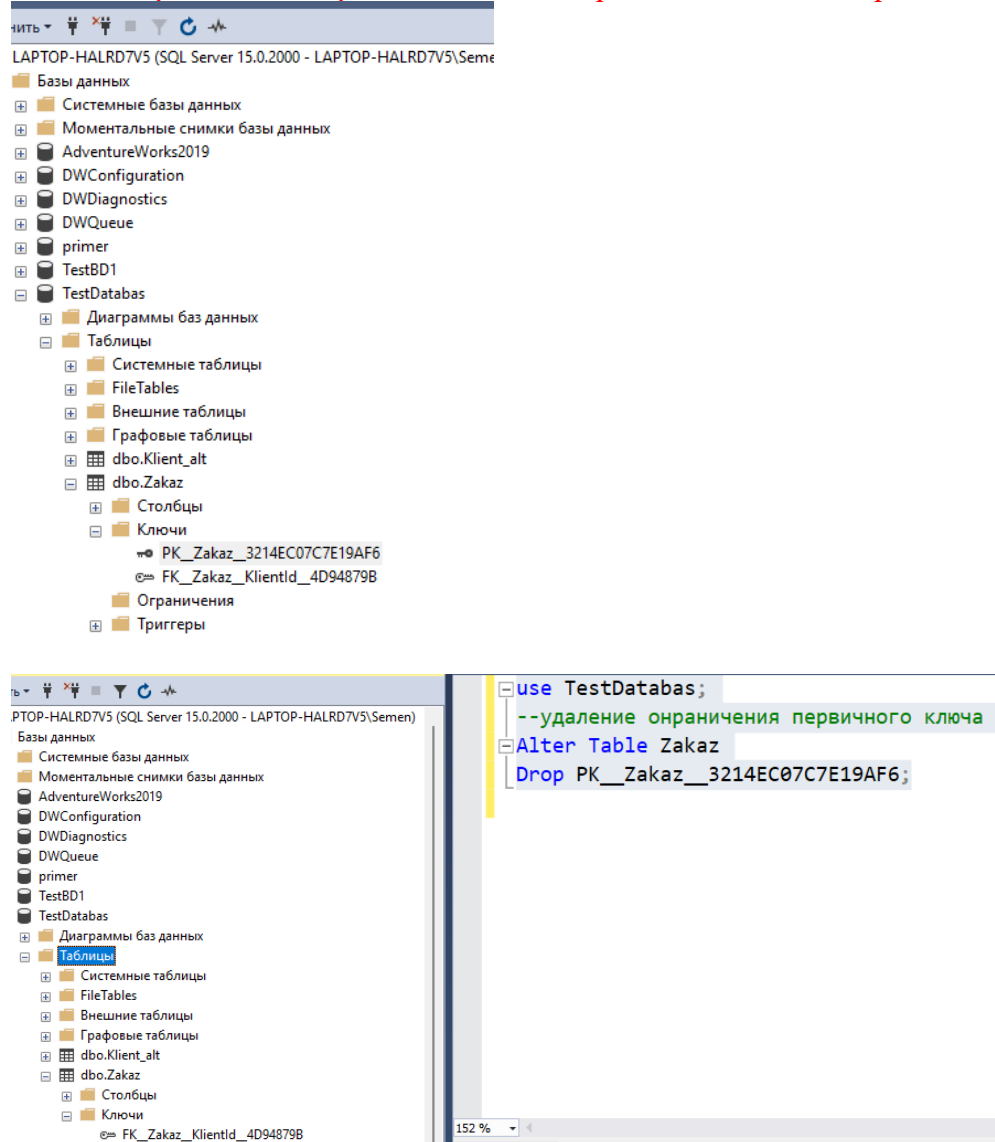
Напишем код, где в БД добавляются две таблицы, никак не связанные между собой, затем далее добавим ограничение внешнего ключа к столбцу KlientId таблицы Zakaz, и добавим к ней первичный ключ для столбца Id:



Задание 7.3 Добавление (Удаление) ограничений с именами

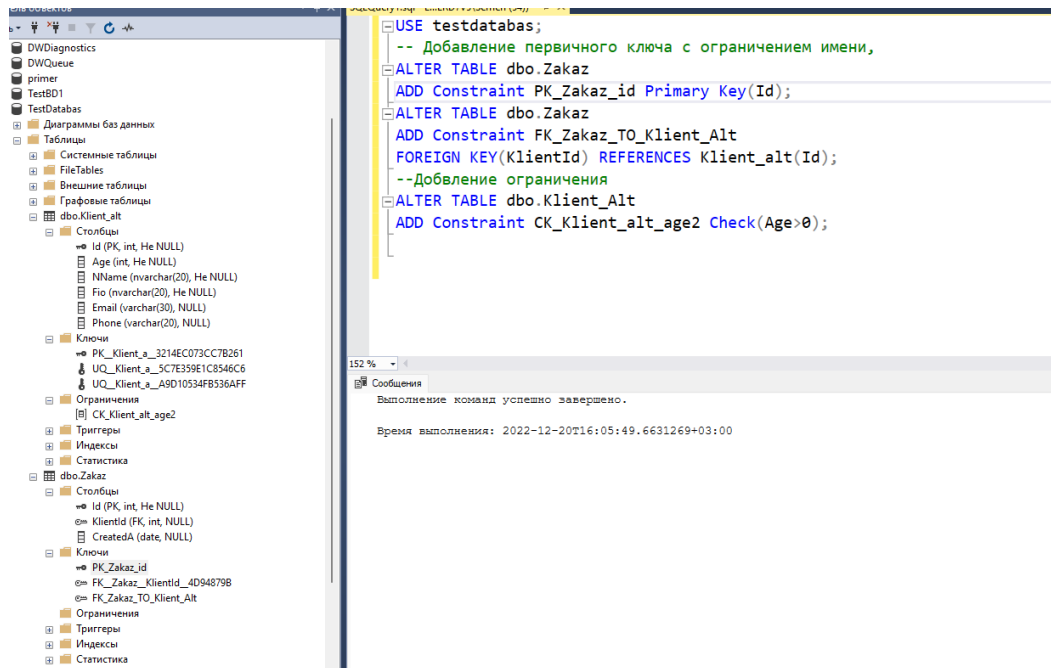
При добавлении ограничений мы можем указать для них имя, используя оператор **CONSTRAINT**, после которого указывается имя ограничения:

Удалим ограничение первичного ключа через автоматически присвоенное имя



Задание 7.4 Добавим ограничение первичного ключа с именем, добавим ограничение внешнего ключа с именем в таблице Zakaz.

Добавим ограничение CHECK для столбца Age (Столбец может содержать любое количество ограничений CHECK).



Задание 7.5 Удаление ограничений

Для удаления ограничений необходимо знать их имя. Если мы точно не знаем имя ограничения, то его можно узнать через SQL Server Management Studio

Напоминание:

Названия ограничений внешних ключей начинаются с "FK".

Названия ограничений CHECK начинаются с "CK", а ограничений DEFAULT - с "DF".

Далее выполним команду

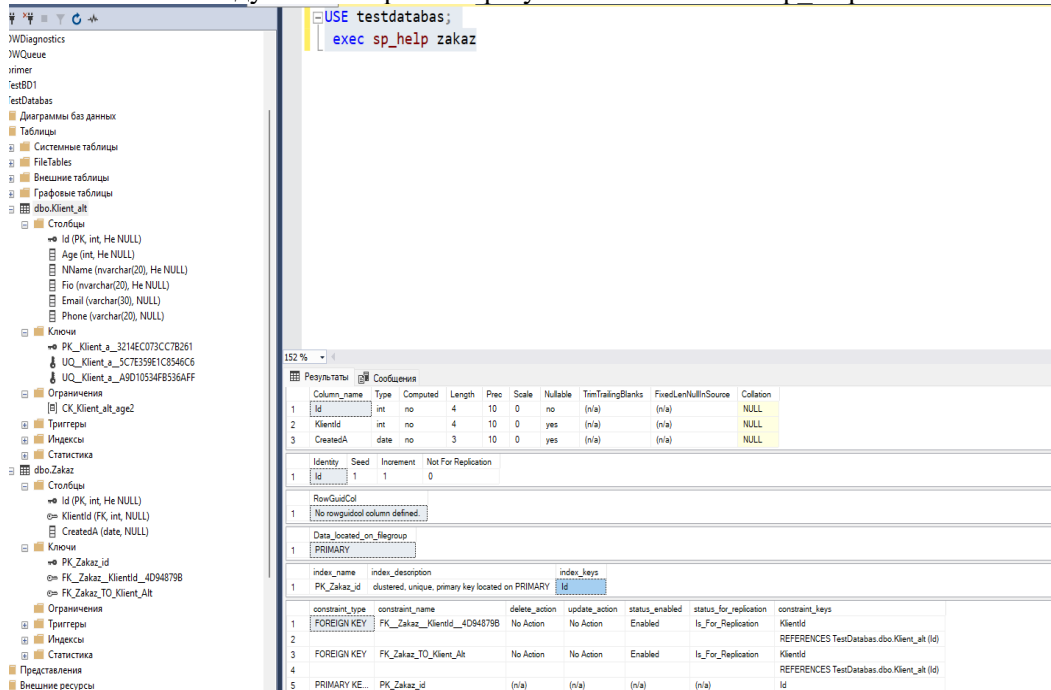
sp_help Names

Команда **sp_help** (точнее это процедура SQL сервера) отображает подробную информацию о указанной таблице, в данном случае это таблица Names.

USE testdatabas;

exec sp_help zakaz

Выполните команду и посмотрите на результат выполнения sp_help.



Обратите внимание на последние строки. Здесь отображается список ограничений и их имена. Чуть выше показаны две строки индексов для этих ограничений.

Задание 8 Переименование таблиц и других объектов баз данных

<https://learn.microsoft.com/ru-ru/sql/relational-databases/system-stored-procedures/sp-rename-transact-sql?view=sql-server-ver16>

Для изменения имени существующей таблицы (и любых других объектов базы данных) применяется системная процедура **sp_rename**.

```
sp_rename [ @objname = ] 'object_name', [ @newname = ] 'new_name'  
[ , [ @objtype = ] 'object_type' ]
```

Аргументы

[@objname =] 'object_name'

Текущее полное или невалифицированное имя объекта пользователя или типа данных. Если объект, который требуется переименовать, является столбцом в таблице, *object_name* должен находиться в формате *table.column* или *schema.table.column*. Если объект, который необходимо переименовать, является индексом, *object_name* должен находиться в форме *table.index* или *schema.table.index*. Если объект, который требуется переименовать, является ограничением, *object_name* должен быть в форме *schema.constraint*.

Кавычки необходимы, только если указан объект с полным именем. Если предоставлено полное имя таблицы, включая имя базы данных, в качестве последнего должно использоваться имя текущей базы данных. *object_name* имеет значение *nvarchar(776)* без значения по умолчанию.

[@newname =] 'new_name'

Новое имя указанного объекта. *new_name* должно быть однокомпонентным именем и соответствовать правилам для идентификаторов. *newname* имеет значение *sysname* без значения по умолчанию.

Переименование столбца

